

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Estimativa da Produtividade no
Desenvolvimento de Software**

por

MARIA ISABEL HAUFE

Dissertação submetida à avaliação
como requisito parcial
para obtenção do grau de
Mestre em Ciência da Computação

Prof. Dra. Ana Maria de Alencar Price
Orientadora

Porto Alegre, janeiro de 2001.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Haufe, Maria Isabel

Estimativa da produtividade no desenvolvimento de software / por Maria Isabel Haufe. – Porto Alegre: PPGC da UFRGS, 2001. 108p.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2001. Orientadora: Price, Ana Maria de Alencar.

1. Produtividade. 2. Métricas. 3. Estimativas. 4. Gerência de Projetos. 5. Qualidade em Software. I. Price, Ana Maria de Alencar. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor de Pós-Graduação: Prof. Philippe Olivier Alexandre Navaux

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alerto Heuser

Bibliotecária – Chefe do Instituto de Informática: Beatriz Haro

Agradecimentos

Para que este trabalho pudesse tornar-se realidade, muitos foram aqueles que me ajudaram, que não mediram esforços colaborando das mais diferentes formas. Por isso, julga-se oportuno destacar aqui, todos que tiveram importância igualmente valorosa.

A meus pais que foram de grande importância, me apoiando e me incentivando para que este trabalho pudesse chegar ao fim.

Aos meus irmãos que me ajudam quando mais precisei, me deram forças para continuar a caminhada.

Aos meus amigos pela compreensão.

A minha orientadora, Ana Price, pela amizade, compreensão e auxílio no desenvolvimento deste trabalho.

A professora Mara Abel pela amizade, e ajuda prestada.

A coordenação e aos funcionários da UFRGS pelas informações e serviços prestados.

Sumário

Lista de Abreviaturas ou Siglas	66
Lista de Figuras.....	7
Lista de Tabelas.....	8
Resumo.....	9
Abstract.....	10
1 Introdução	11
2 Gerência de Projetos de Software.....	14
2.1 Planejamento	14
2.1.1 Elaborar Estimativas	16
2.1.2 Definir a Estrutura da Organização do Projeto	17
2.1.3 Definir Premissa de Gestão.....	17
2.1.4 Elaborar Cronograma.....	18
2.1.5 Orçar o Projeto.....	18
2.2 Controle.....	19
2.3 Observações	20
3 Técnicas de Estimativas.....	21
3.1 Estimativa do Esforço	21
3.2 Estimativa de Putnam.....	22
3.3 Constructive Cost Model - COCOMO.....	24
3.4 Análise de Pontos por Função	26
3.5 Pontos de Particularidade	28
3.6 Personal Software Process - PSP	30
3.7 Comparativo entre as Técnicas.....	32
4 Ferramentas Existentes	35
4.1 ESTIMACS	35
4.2 SLIM.....	36
4.3 SPQR/20	36
4.4 ESTIMATE Professional.....	37
4.5 Resultados	38
5 Ferramenta Desenvolvida	41
5.1 Ambiente	41
5.2 Definições Técnicas para a Ferramenta.....	43

5.3 Estrutura da Ferramenta	45
5.4 Casos de Uso.....	48
5.5 Funcionalidades da Ferramenta	49
5.5.1 Parametrização	50
5.5.2 Cadastramento das Informações Pessoais.....	51
5.5.3 Cadastramento dos Projetos.....	51
5.5.4 Estimativas.....	53
5.5.5 Acompanhamento	55
5.6 Busca de Informações numa Base Histórica.....	59
6 Estudos de Casos Submetidos	64
6.1 Custos do Transporte.....	64
6.1.1 Definições do Custo do Transporte.....	64
6.1.2 Resultados	66
6.2 EDI – Troca de Informações de forma Eletrônica.....	70
6.2.1 Definições do EDI	71
6.2.2 Resultados	72
7 Conclusão	76
Anexo 1 Tabela de Decomposição	78
Anexo 2 Tabela de Multiplicadores (COCOMO).....	79
Anexo 3 Tabelas dos Pontos por Função.....	81
Anexo 4 Tabelas do PSP	84
Anexo 5 Diagrama de Atividades	85
Anexo 6 Diagramas de Interação.....	92
Anexo 7 Diagrama de Estados	98
Anexo 8 Dicionário de Dados	99
Bibliografia	106

Lista de Abreviaturas ou Siglas

APF	Análise de Pontos por Função
CMM	Capability Maturity Model
COCOMO	Constructive Cost Model
IFPUG	International Function Point Users Group
LOC	Número de Linhas de Código
PF	Pontos por Função
PM	Programadores-Mês
PMP	Proposta de Melhoria do Processo
PP	Pontos de Particularidade
PROBE	Proxy-Based-Estimating Method
PSP	Personal Software Process
RBC	Raciocínio Baseado em Casos
SEI	Software Engineering Institute
SPC	Software Productivity Centre
TRC	Transporte Rodoviário de Cargas

Lista de Figuras

FIGURA 2.1 – Ciclo básico de Gerenciamento de Projetos	14
FIGURA 2.1.1 – Processo de Planejamento de Projetos.....	16
FIGURA 5.1.1 – Ambiente necessário para controle do desenvolvimento.....	42
FIGURA 5.3.1 – Diagrama de Classes.....	46
FIGURA 5.4.1 – Diagrama de Casos de Uso	48
FIGURA 5.5.4.1 – Tela de Consulta da Estimativa de um Módulo.....	54
FIGURA 5.5.4.2 – Tela de Consulta da Estimativa de um Função.....	55
FIGURA 5.5.5.1 – Tela de Pontos de Revisão do Projeto	56
FIGURA 5.5.5.2 – Tela de Alocação dos Profissionais no Projeto.....	56
FIGURA 5.5.5.3 – Tela de Acompanhamento do Profissional	57
FIGURA 5.5.5.4 – Tela de Acompanhamento da Estimativa	58
FIGURA 5.5.5.5 – Tela de Comparação entre Estimativas.....	59
FIGURA 5.6.1 – Ciclo do RBC.....	60
FIGURA 5.6.2 – Representação Estruturada - RBC	61
FIGURA 5.6.3 – Modelo de Caso da Ferramenta de Estimativa	61
FIGURA 5.6.4 – Descrição de um de Caso para Ferramenta de Estimativa	62
FIGURA 6.1.2.2 – Comparativo entre Estimativas no projeto “Custos do Transporte”	67
FIGURA 6.1.2.3 – Comparativo da Métrica PF-Fixo modificada para o Projeto de Custos do transporte.	68
FIGURA 6.1.2.4 – Acompanhamento dos Profissionais no projeto Custos do Transporte	69
FIGURA 6.1.2.5 – Alocação dos Profissionais no Projeto Custos do Transporte	69
FIGURA 6.1.2.6 – Acompanhamento do Projeto – Custos do Transporte	70
FIGURA 6.2.2.2 – Comparativo entre Estimativas no projeto “EDI”	73
FIGURA 6.2.2.3 – Acompanhamento dos Profissionais no projeto “EDI”	74
FIGURA 6.2.2.4 – Alocação dos Profissionais no projeto “EDI”	74
FIGURA 6.2.2.5 – Acompanhamento do Projeto no projeto “EDI”	75

Lista de Tabelas

TABELA 3.1.1 – Estimativa do Esforço	22
TABELA 3.3.1 – Coeficientes do COCOMO	25
TABELA 3.5.1 – Cálculo de Pontos de Particularidade	29
TABELA 3.7.1 - Comparação das Técnicas Apresentadas	34
TABELA 4.5.1 – Comparativo entre Ferramentas	40
TABELA 5.2.1 - Comparativo das formas de medir o Tamanho do Projeto	43
TABELA 5.2.2 – Conversão de Pontos por Função em Horas	44
TABELA 6.2.2.1 – Tabela Comparativa das Estimativas - EDI	72
TABELA A.1.1 - Decomposição	78
TABELA A.2.1 - Multiplicadores do COCOMO 81	79
TABELA A.2.2 - Multiplicadores da extensão do COCOMO	80
TABELA A.3.1.1 - Complexidade dos Arquivos Lógicos Internos	81
TABELA A.3.2.1 - Complexidade das Interfaces Externas	81
TABELA A.3.3.1 - Complexidade das Entradas Externas	81
TABELA A.3.4.1 - Complexidade das Saídas Externas	81
TABELA A.3.5.1 - Complexidade das Consultas – Parte de Entrada	82
TABELA A.3.5.2 - Complexidade das Consultas – Parte de Saída	82
TABELA A.3.6.1 - Cálculo dos Pontos por Função	82
TABELA A.3.7.1 - Características dos Pontos por Função	83
TABELA 4.1 – Como aplicar o PSP	84

Resumo

Este trabalho apresenta uma ferramenta para gerenciamento de projetos, priorizando as fases de planejamento e o controle do desenvolvimento de software.

Ao efetuar o planejamento de um projeto é necessário estimar o prazo, o custo e o esforço necessário, aplicando técnicas já aprovadas, existentes na literatura, tais como: Estimativa do Esforço, Estimativa de Putnam, Modelo COCOMO, Análise de Pontos por Função, Pontos de Particularidade e PSP. É necessária a utilização de uma ferramenta que automatizem o processo de estimativa. Hoje no mercado, encontram-se várias ferramentas de estimativas, tais como: ESTIMACS, SLIM, SPQR/20, ESTIMATE Professional.

O controle do desenvolvimento do projeto está relacionado ao acompanhamento do projeto, do profissional e da própria estimativa de custo e esforço de desenvolvimento. Nenhuma das ferramentas estudadas permitiu o controle do projeto por parte da gerência, por isto esta se propondo o desenvolvimento uma nova ferramenta que permita o planejamento e controle do processo de desenvolvimento. Esta ferramenta deve permitir a comparação entre as diversas técnicas de estimativas, desde que baseadas na mesma medida de tamanho: pontos por função. Para exemplificar o uso desta ferramenta, foram aplicados dois estudos de casos desenvolvidos pela empresa Newsoft Consultoria de Informática.

Palavras-Chaves: Produtividade, Métricas, Estimativas, Gerência de Projetos e Qualidade em software.

Title: “Productivity Estimate in Software Development”**Abstract**

This paper presents an environment geared towards for project management, prioritizing planning and development control.

In order to carry out project planning, the schedule, the cost and necessary effort have to be estimated, applying previously tested techniques which are found in the literature such as: Effort Estimate, Putnam’s Estimate, COCOMO Model, Function Point Analysis, Features Points and PSP. This project also presents some features from the ESTIMACS, SLIM, SPQR/20 and Estimate Professional tools that automate the estimate process.

Project control is related to the project follow-up, to the professional and to the development effort and the cost estimate itself. As none of the tools analyzed has allowed this sort of control, a new tool that makes this control possible and also compares several estimate techniques, provided that they are based on size measurement, function points, was developed. To exemplify the use of this tool, two case studies developed by the company Newsoft Consultoria de Informática were applied.

Keywords: Productivity, Metrics, Estimates, Project Management and Quality in software.

1 Introdução

A globalização da economia eleva o nível de concorrência entre as empresas sob todos os aspectos, exigindo uma constante redução de custos, melhora na qualidade dos projetos, racionalização de processos e desenvolvimento rápido de novos produtos e serviços, assim como o aperfeiçoamento dos recursos humanos e técnicos.

Para algumas empresas, a estimativa de custo do software é vital para a tomada de decisão, constituindo-se num aspecto até mesmo de sobrevivência delas no mercado. Assim sendo, não se pode encarar tal atividade meramente como uma sofisticação de algumas organizações, mas como um fator decisivo para um bom planejamento e administração por parte dos elementos responsáveis pelo gerenciamento dos projetos. A utilização de recursos computacionais e humanos consiste nos maiores custos de uma empresa desenvolvedora de software.

De acordo com [PRE 97], o software é medido por muitas razões:

- ✓ Indicar a qualidade do produto
- ✓ Avaliar a produtividade da equipe de desenvolvimento
- ✓ Avaliar os benefícios derivados de novos métodos e ferramentas
- ✓ Formar uma linha básica para estimativas
- ✓ Ajudar a justificar os pedidos de novas ferramentas ou treinamento.

Apesar da existência de técnicas de estimativas de custo e esforço disponíveis, muitas vezes o gerente prefere valer-se somente de sua experiência, utilizando a semelhança entre projetos a fim de realizar a estimativa de custo e esforço do software. Portanto, é desejável que o gerente disponha de modelos e ferramentas automáticas que o motivem à utilização de modelos algorítmicos de estimativa de custos e esforços.

De acordo com [SEL 96], a complexidade de softwares tende a aumentar continuamente, fato que dificulta a garantia da qualidade dos produtos de software ao longo do processo de desenvolvimento. O primeiro passo para controlar e reduzir a complexidade do software é compreender em que consiste essa complexidade e como ela pode ser trabalhada.

Segundo Watts S. Humphrey “Até hoje, os produtos de software têm os seus cronogramas atrasados, custos maiores do que os esperados e apresentam defeitos. Isto tem como resultado uma série de inconvenientes para os usuários e traz consigo uma enorme perda de tempo e de recursos. Estes problemas são causados por práticas não profissionais no desenvolvimento de software, que são ocasionadas pela atual cultura de software” [WEB 99].

Faz-se necessário uma cultura onde:

- ✓ O gerente do projeto deve seguir processos definidos, planejados e medidos.
- ✓ O trabalho deve ser planejado antes de chegar-se a um cronograma.
- ✓ O gerente deve medir e controlar, tanto os dados relativos ao tempo como ao tamanho e defeitos nos produtos.
- ✓ O projeto deve ser documentado e revisado antes de ser codificado.

- ✓ A qualidade deve ser planejada, medida e controlada ao longo do processo de desenvolvimento.
- ✓ O trabalho concluído deve ser analisado e o resultado deve ser utilizado para melhorar o processo.

A estimativa é parte integrante do planejamento e controle do processo de gerência de projetos. Para que um gerente de projetos possa definir o prazo, o custo e o esforço do desenvolvimento, ele deve utilizar técnicas definidas para tal. Exemplo destas são: Estimativa do Esforço, Estimativa de Putnam, Modelo COCOMO – Constructive Cost Model, Análise de Pontos por Função, Pontos de Particularidade e PSP – Personal Software Process.

Cada uma das técnicas possui características específicas. A Estimativa do Esforço baseia-se no número de pessoas-mês que irá desenvolver determinada função, enquanto que a Estimativa de Putnam baseia-se na equação do software, relacionando o esforço com o tempo de desenvolvimento. O Modelo COCOMO utiliza as equações definidas por Barry Boehm, a partir do número de linhas de código. A Análise de Pontos por Função baseia-se na funcionalidade do projeto a ser desenvolvido, a Técnica Pontos de Particularidade é uma extensão da Análise de Pontos por Função e o PSP é um processo de melhoria pessoal.

Este trabalho tem como objetivo apresentar uma ferramenta que possibilite ao gerente de projeto efetuar o planejamento e o controle dos projetos de desenvolvimento do software.

O planejamento envolve a estimativa de custo e esforço do desenvolvimento do projeto, a alocação dos profissionais e dos recursos necessários. O processo de controle refere-se ao acompanhamento do projeto, do profissional e da estimativa, que pretende verificar a produtividade da equipe de desenvolvimento e a qualidade do software desenvolvido.

Este trabalho apresenta no Capítulo 2, a importância da gerência de projetos, priorizando o planejamento e o controle dos projetos de software a serem desenvolvidos.

As técnicas de estimativas encontradas na literatura, tais como: Estimativa do Esforço, Estimativa de Putnam, Modelo COCOMO, Análise de Pontos por Função, Pontos de Particularidade e PSP estão descritas no Capítulo 3. E ao final deste capítulo apresenta-se um estudo comparativo entre essas técnicas.

No Capítulo 4 encontra-se a descrição de algumas ferramentas que automatizam o processo de estimativa, tais como: ESTIMACS, SLIM, SPQR/20 e ESTIMATE Professional. Ao final deste capítulo também é apresentado um estudo comparativo entre essas ferramentas.

Verificou-se que nenhuma das ferramentas descritas no Capítulo 4 atendia completamente as necessidades de planejamento e controle da gerência. Por este motivo está se propondo, no Capítulo 5, uma ferramenta de desenvolvimento adequado ao planejamento e controle de projeto, juntamente com uma ferramenta de estimativa de custo e esforço, que atenda as necessidades dos gerentes dos projetos.

No Capítulo 6, descreve-se dois projetos, Custos para o Transporte e EDI – Troca Eletrônica de Informações, desenvolvidos pela empresa Newsoft Consultoria de Informática. Estes projetos foram submetidos à ferramenta de estimativa de custo e esforço desenvolvida. Neste Capítulo, também descreve-se os resultados obtidos para cada um dos projetos analisados.

Finalmente no Capítulo 7, encontra-se a conclusão deste trabalho, e sugestões para trabalhos futuros.

2 Gerência de Projetos de Software

A complexidade das empresas modernas, fruto do elevado nível de competitividade e de avanços tecnológicos, provocou um aumento na exigência da qualidade e complexidade das decisões administrativas [CAN 99]. Um projeto envolve execuções de diversas atividades independentes e ao mesmo tempo, eleva o grau de risco e incerteza quanto ao sucesso do projeto.

Este ambiente dinâmico e incerto dos projetos requer um gerenciamento eficaz, com procedimentos avançados de planejamento e controle do projeto.

A Figura 2.1 apresenta o ciclo básico de gerenciamento de projetos de software [FER 95]. O controle deve atuar tanto sobre o planejamento quanto sobre o desenvolvimento. Podendo ocasionar desvios que geram ações corretivas. A avaliação ao final do projeto, realimenta o planejamento de novos projetos e assim prossegue num ciclo indefinido. Os padrões são definidos a partir dos controles e das avaliações para o controle e planejamento do projeto.

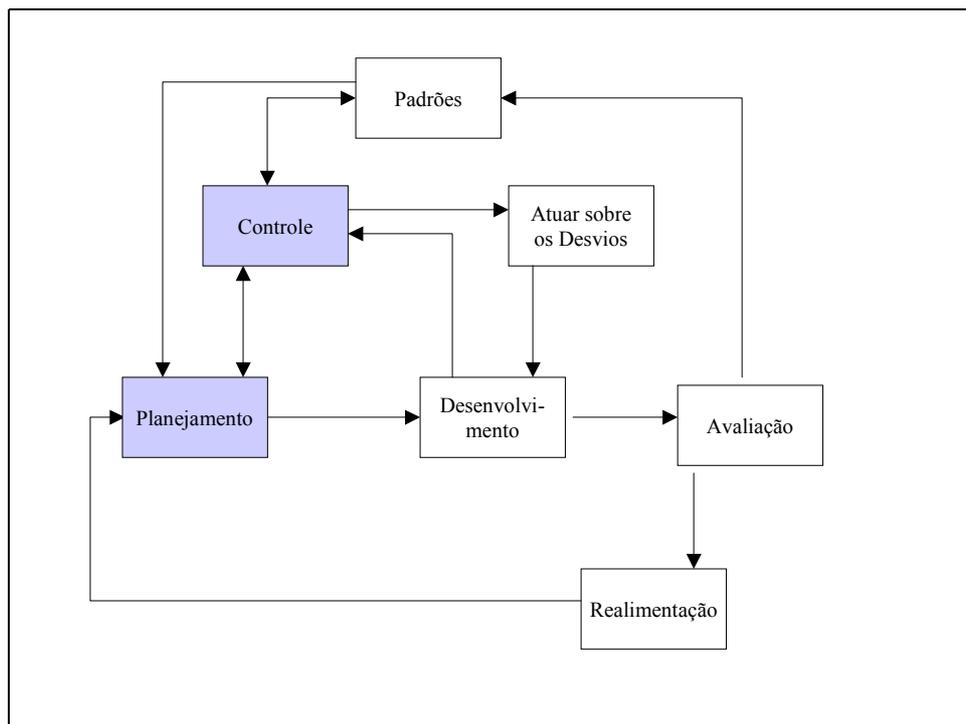


FIGURA 2.1 – Ciclo básico de Gerenciamento de Projetos

2.1 Planejamento

O planejamento é fundamental para o sucesso do projeto [FER 95]. A Figura 2.1.1 apresenta um processo de planejamento de projetos composto por:

Caracterizar Solução: significa definir o escopo do projeto em termos de requisitos do cliente.

Definir o Processo de Desenvolvimento: consiste em determinar o processo que apoiará o desenvolvimento da solução em termos de fases/etapas/atividades, padrões, técnicas a serem utilizadas.

Selecionar Perfil do Pessoal: significa determinar as habilidades técnicas e gerenciais necessárias para o projeto em função da natureza e das características da solução e do processo definido.

Elaborar Estimativas: consiste na elaboração de estimativas de prazo, esforço, custo do projeto bem como de tamanho do software. Pode-se também considerar a produtividade esperada da equipe.

Definir Estrutura da Organização do Projeto: significa definir como o projeto será organizado em termos de coordenação, quem deverá participar da coordenação, definição das atribuições, indicação das funções de apoio ao projeto.

Definir Premissa de Gestão: significa determinar o modelo de gestão a ser adotado em termos de monitoramento do projeto, informações a serem fornecidas para o cliente acerca do progresso do projeto, estabelecimento de pontos de controle, local de desenvolvimento, aspectos relativos à transferência de tecnologia, tratamento de mudanças de escopo.

Elaborar Cronograma: consiste em estabelecer prazos para cada fase / etapa / atividade do projeto, conforme o processo de desenvolvimento definido.

Orçar o Projeto: consiste em elaborar o orçamento do projeto a partir dos resultados das estimativas e da determinação quantitativa dos recursos necessários, requeridos pelo empreendimento.

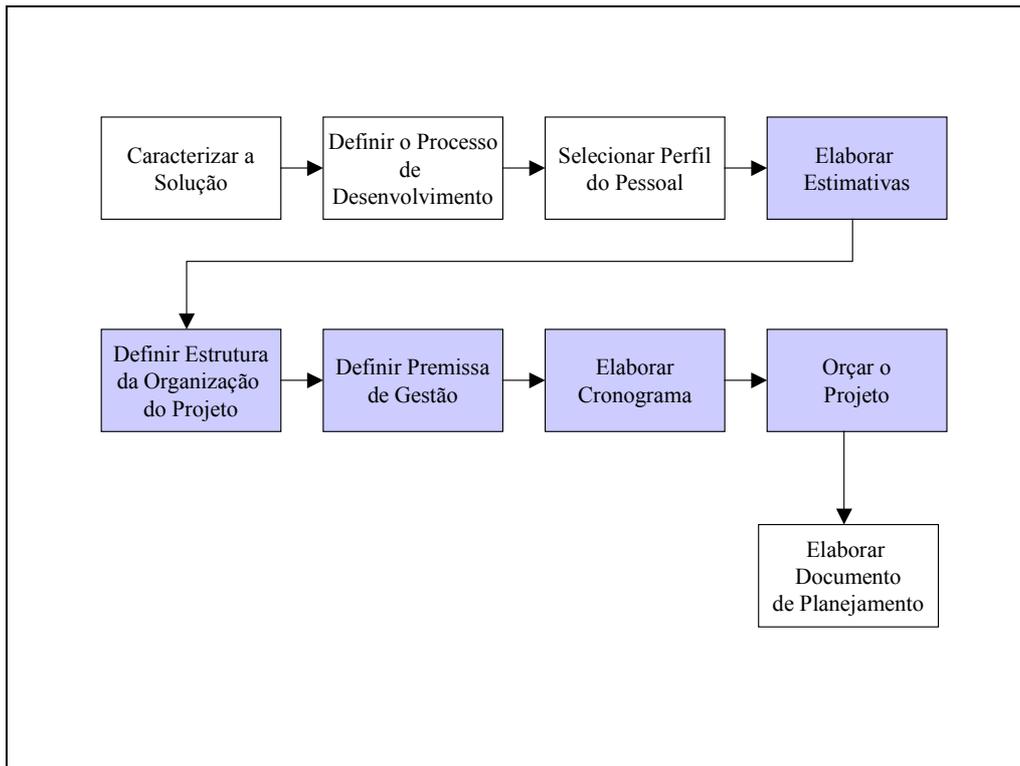


FIGURA 2.1.1 – Processo de Planejamento de Projetos

No estágio de planejamento o gerente deve preocupar-se com [MAF 92]:

- ✓ estabelecer as características básicas da funcionalidade do sistema a ser desenvolvido,
- ✓ demonstrar a viabilidade técnica e financeira do projeto,
- ✓ analisar o custo através de comparações custo-benefício com projetos concorrentes,
- ✓ estabelecer as características básicas da implementação mais econômica,
- ✓ definir os custos materiais e humanos, juntamente com os cronogramas estimados para essa implementação,
- ✓ convencer o cliente / usuário que contratou o desenvolvimento do sistema que a solução proposta efetivamente contribuirá para o objetivo da sua organização.

2.1.1 Elaborar Estimativas

O processo de elaborar estimativa de custo e esforço do desenvolvimento do projeto, é uma etapa importante no processo de planejamento. Esta etapa é composta das seguintes atividades:

- ✓ Definir a medida de tamanho: definir qual das medidas de tamanho será utilizada: número de pessoas-mês, número de linhas de código, pontos por função ou pontos de objetos.
- ✓ Definir a métrica de estimativa: definir qual métrica de estimativa será utilizada: Estimativa do Esforço, Estimativa de Putnam, Modelo COCOMO, Análise de Pontos por Função, Pontos de Particularidade ou PSP.

- ✓ Estimar o tamanho do projeto: aplicar um método, tabela de decomposição, experiência passada, ou contagem de pontos por função, para identificar o tamanho do projeto.
- ✓ Definir as prioridades do desenvolvimento: definir a seqüência das atividades a serem desenvolvidas.
- ✓ Estimar o esforço necessário: a partir do tamanho do projeto e da prioridade estabelecida, definir o número de pessoas-mês necessárias para o desenvolvimento aplicando a métrica de estimativa definida.

Não basta elaborar uma estimativa na fase inicial do desenvolvimento e não efetuar o controle sobre o projeto, porque sem o controle os valores apresentados podem se tornar incorretos.

Caso ocorram modificações no projeto que afetem o seu tamanho, a estimativa deve ser refeita e reanalisada, novos valores devem ser utilizados para efetuar o controle do projeto.

2.1.2 Definir a Estrutura da Organização do Projeto

Esta etapa do planejamento consiste na definição da equipe que trabalhará no desenvolvimento do projeto.

Primeiramente deve ser definido que tipos de serviços prestados pelos profissionais a serem alocados no projeto: analistas, programadores, consultores. Depois desta definição é necessário definir o perfil do profissional que irá executar estes serviços: tempo de empresa, conhecimento técnico, conhecimento do negócio e escolaridade.

A partir destas definições devem ser sugeridos nomes de profissionais disponíveis, capazes de serem alocados para este novo projeto.

Para controle da gerência é interessante que estas informações sobre os profissionais alocados no projeto, possam ser visualizados em gráficos (Ex: Gráfico de Gantt). Na alocação dos profissionais deve ser considerada a data de início da alocação, a data de término da alocação, e o tempo disponível para desenvolvimento do projeto. Um profissional pode ser alocado em dois períodos diferentes de tempo. Também deve ser considerado o tempo improdutivo do profissional: como atendimento de telefones, suporte a outros projetos.

2.1.3 Definir Premissa de Gestão

Esta etapa do planejamento define o relacionamento entre o cliente e a software house. Atualmente, os clientes querem ter informações sobre o andamento do seu projeto, para que ao final não sejam surpreendidos com um projeto que não atenda as suas necessidades, ou que seja entregue após o prazo estimado ou com um custo superior ao definido.

Nesta etapa do planejamento definem-se os pontos de controle ou “milestones”. Pontos de controle são datas de verificação do projeto. Quando se define uma data de verificação define-se também o que deve estar desenvolvido até este ponto, permitindo desta forma fazer uma relação do estimado com o desenvolvido.

Para determinar o que deve estar desenvolvido até a data de verificação, pode-se utilizar as informações referentes à alocação dos profissionais juntamente com a sua capacidade de produção.

Para projetos pequenos, duração de zero a quatro meses, os pontos de controle podem ser definidos como sendo de quinze em quinze dias. Mas para projetos maiores, de quatro meses a um ano, sugere-se que os pontos de controle sejam definidos mês-a-mês. Para projeto de um ano a dois anos, os pontos de controle devem ser bimestrais, e para projetos superiores devem ser trimestrais.

Nesta etapa também devem ser definidos os procedimentos a serem seguidos quando ocorrer uma modificação no escopo do projeto. O cliente deve ser informado dos novos prazos e custos do projeto.

2.1.4 Elaborar Cronograma

A partir das informações das atividades a serem desenvolvidas, e dos profissionais alocados para o projeto, deve-se definir o cronograma do projeto.

Primeiramente as prioridades das atividades do desenvolvimento devem ser revisadas, e aprovadas pelo cliente. Depois de definidas as prioridades, deve-se relacionar as atividades com os profissionais alocados e capacitados para o desenvolvimento da atividade. Esta relação, também chamada de cronograma, deve ser divulgada para toda a equipe de desenvolvimento, pois existem atividades que são dependentes de outras atividades.

Um cronograma pode ser apresentado por meio de uma tabela, relacionando a atividade com o profissional e o período no qual ele estará alocado para o seu desenvolvimento.

2.1.5 Orçar o Projeto

Esta etapa do planejamento é responsável pela definição dos custos e do preço do projeto. Com base nos profissionais alocados, nos cronogramas e observando os custos administrativos pode-se determinar o custo do projeto. Sobre o custo do projeto deve ser definido um percentual de lucro que a software house deseja obter para determinar o preço do projeto.

O preço do projeto deve estar de acordo com as possibilidades do cliente e tornar o projeto viável para a software house.

2.2 Controle

No controle, devem ser empregados alguns procedimentos para medir e acompanhar: o tempo, o custo e o esforço do desenvolvimento de um projeto, durante a sua vida, isto significa coordenar a ação de todas as partes.

Em relação ao tempo, a melhor forma de gerenciar é através de cronogramas, representação gráfica do tempo planejado, ou estimado, para que determinada atividade seja executada. A forma de um escala de tempo, pode ser demonstrada por meio de barras que indicam a duração e o período de tempo para a qual a atividade foi programada, também pode ser demonstrado o confronto entre o tempo estimado e o tempo real gasto no projeto.

Para o controle do custo utiliza-se como ferramenta básica o orçamento. Elaborado ainda na fase preliminar do planejamento, por meio de estimativas de custos. À medida que o projeto vai sendo desenvolvido também o orçamento vai sendo aperfeiçoado, até chegar ao nível operacional de execução das atividades.

Também deve ser efetuado o controle do projeto, do profissional e da estimativa de custo e esforço no desenvolvimento de software.

O projeto deve ser controlado através dos pontos de controle, ou “milestones”, definidos na fase de planejamento. Um ponto de controle significa que em uma determinada data se fará uma verificação do andamento do projeto, se o que foi desenvolvido até o momento está de acordo com o que foi estimado. A partir destes pontos de controle é possível elaborar um gráfico de linhas, visualizando o andamento do projeto. A partir destas informações o gerente do projeto pode tomar ações corretivas para acelerar ou desacelerar o desenvolvimento do projeto.

O controle dos profissionais está relacionado diretamente à produtividade da equipe de desenvolvimento, esta não é uma atividade fácil, pois o desenvolvimento depende do conhecimento e da vontade do profissional. Primeiramente devem ser analisadas as características de cada um dos profissionais que compõem a equipe de desenvolvimento. Depois deve-se definir uma média de produtividade por profissional. O controle está na relação do que o profissional realmente produz com a sua média. De tempos em tempos a média deve ser novamente refeita. O gerente do projeto deve visualizar estas informações através de gráficos, para a tomada de ações corretivas, pois a visualização através de gráficos facilita identificar quais profissionais apresentam problemas quanto à produtividade.

E o controle das estimativas, serve para efetuar os ajustes nas estimativas de acordo com a empresa desenvolvedora do software. Também é interessante efetuar uma comparação entre diversas estimativas sobre um mesmo projeto, para identificar a estimativa que mais se adapta ao projeto a ser desenvolvido.

2.3 Observações

O processo de estimativa é uma pequena parte do processo de planejamento e controle de um projeto, mas é extremamente importante. Pois é a partir destas informações que se consegue:

- ✓ Definir o tamanho previsto do projeto.
- ✓ Definir o esforço necessário para o desenvolvimento do projeto.
- ✓ Definir o custo previsto do projeto.
- ✓ Definir o tempo necessário para o desenvolvimento do projeto.

De acordo com Deming, “85% dos problemas que ocorrem nos processos da empresa é de responsabilidade da gerência” [DEM 86]. Mais do que nunca, o ambiente competitivo das empresas está a exigir um processo constante de mudanças, em qualquer tipo de segmento produtivo.

Para que um processo de estimativa tenha sucesso é necessário:

- ✓ Definir claramente quais métricas deverão ser coletadas e analisadas
- ✓ Definir como os resultados das métricas serão utilizados
- ✓ Definir as métricas que serão empregadas no projeto desde seu planejamento
- ✓ Dar um feedback periódico à equipe do projeto
- ✓ Manter o comprometimento com o esforço da medição

É importante ter em mente que a obtenção da qualidade desejada é de responsabilidade de todos, mas principalmente dos níveis gerenciais, pois são eles que definem os orçamentos e aprovam as aquisições de recursos, entre outros aspectos.

3 Técnicas de Estimativas

Antes que um projeto seja planejado, os objetivos e o escopo devem ser estabelecidos, soluções alternativas devem ser consideradas e as restrições administrativas e técnicas identificadas. Sem essas informações é impossível definir estimativas de custos razoáveis e precisas.

A literatura apresenta uma série de técnicas de estimativas tais como: Estimativa do Esforço [PRE 95], Estimativa de Putnam [PUT 78], Constructive Cost Model (COCOMO) [BOE 81], Análise de Pontos por Função (APF) [IFP 94], Pontos de Particularidade [PRE 95] e Personal Software Process (PSP) [HUM 95]. Essas técnicas apresentam os seguintes atributos em comum:

- ✓ O escopo do projeto deve ser estabelecido antecipadamente,
- ✓ As métricas de software utilizam os históricos de aferições passadas como base sobre as quais as estimativas serão feitas,
- ✓ O projeto é dividido em pequenas partes que são estimadas individualmente.

Hoje, as métricas de estimativa de custo dividem-se em métricas orientadas ao tamanho e métricas orientadas à função. As métricas orientadas ao tamanho são medidas diretas do software e do processo pelo qual ele é desenvolvido [JON 91]. Elas provocam controvérsias e não são universalmente aceitas como a melhor maneira de medir o processo de desenvolvimento de software. A maior parte da controvérsia gira em torno do uso das linhas de código como uma medida chave. Métricas orientadas à função são medidas indiretas do software e do processo pelo qual ele é desenvolvido. Em vez de contar as linhas de código, a métrica orientada à função concentra-se na “funcionalidade” ou “utilidade” do sistema.

3.1 Estimativa do Esforço

Segundo [PRE 95], a estimativa do esforço é a técnica mais comum para levantar os custos de qualquer projeto de desenvolvimento de engenharia. Um número de pessoas-dia, mês ou ano é aplicado à solução de cada tarefa do projeto. Um custo em dólares é associado a cada unidade de esforço e um custo estimado é derivado.

De acordo com a medida LOC (Número de Linhas de Código), a estimativa do esforço inicia com um delineamento das funções do software identificadas a partir do escopo do projeto. As funções e as tarefas de desenvolvimento (análise de requisitos, projeto, codificação, teste, documentação) correlatas podem ser representadas como parte de uma tabela, tal como é ilustrada na Tabela 3.1.1.

O planejador estima o esforço (pessoas-mês) que será exigido para concluir cada tarefa de desenvolvimento para cada função do software. Taxas de mão-de-obra (custo-esforço unitário) são aplicadas em cada uma das tarefas de desenvolvimento. Muito provavelmente a taxa de mão-de-obra variará para cada tarefa. O pessoal de nível sênior deverá envolver-se fortemente na análise de requisitos e nas primeiras tarefas de

realização do projeto; o pessoal de nível junior (programadores) deverá envolver-se nas últimas tarefas de projeto, codificação e nos primeiros testes.

TABELA 3.1.1 – Estimativa do Esforço

	Análise de Requisitos	Projeto	Codificação	Teste	Documentação	Total
Funções específicas do projeto						
				Estimativas em		
				Pessoas-Mês		
Total						
Taxa(\$)						
Custo (\$)						

O custo e o esforço de cada função e tarefa de desenvolvimento são computados como o último passo. Se a estimativa do esforço for realizada independentemente da estimativa LOC (tabela de decomposição por número de linhas de código, apresentadas no Anexo 1) tem-se, então, duas estimativas para o custo e para o esforço que podem ser comparadas e acordadas. Se as duas estimativas demonstrarem razoável concordância, haverá uma boa razão para acreditar que as estimativas são confiáveis. Se por outro lado, os resultados dessas técnicas exibirem pouca concordância, investigação e análise adicionais devem ser feitas.

Esta estimativa é aplicada em empresas sem seguir uma regra definida, pois a sua medida-chave é o número de pessoas-mês que irão trabalhar no projeto. Medida esta que identifica o maior custo de uma software house, e para controle da empresa é necessário saber quais profissionais estão trabalhando em quais projetos.

3.2 Estimativa de Putnam

A estimativa de Putnam [PUT 78] é um modelo dinâmico de múltiplas variáveis que pressupõe uma distribuição de esforço específico ao longo da existência de um projeto de desenvolvimento de software. O modelo foi construído a partir de uma distribuição de mão-de-obra encontrada em grandes projetos (esforço total de 30 pessoas-ano ou mais), porém a extrapolação para projetos de software menores é possível.

A estimativa de Putnam relaciona linhas de código ao tempo e esforço de desenvolvimento, utilizando as seguintes equações:

$$L = C_k * K^{1/3} * T^{4/3} \quad \text{ou, equivalentemente } K = L^3 / (C_k^3 * T^4)$$

Onde,

L = número de linhas de código

K = esforço medido em pessoas-mês ou pessoas-ano

T = duração do projeto em meses ou anos

C_k = constante do estado da tecnologia.

Os valores típicos para a constante do estado da tecnologia são: para um ambiente de desenvolvimento pobre¹ $C_k = 2.000$; para um ambiente de desenvolvimento de bom² nível, $C_k = 8.000$ e para um ambiente excelente³, $C_k = 11.000$. O valor da constante de tecnologia (C_k) também pode ser derivado das condições locais utilizando dados históricos coletados no esforço do desenvolvimento.

A estimativa de Putnam permite a troca do tempo pelo esforço. Ao acrescentar pessoas num projeto, diminui-se o tempo do desenvolvimento. Em contra partida aumenta-se o custo do projeto. Ao ultrapassar o ponto máximo para esta troca o projeto torna-se incontrolável. Em contra partida, os administradores, podem simular outras situações, verificando o tempo, o esforço e o custo do projeto. Quanto menor o tempo, maior o esforço e o custo do desenvolvimento de um projeto.

Esta estimativa também permite o cálculo da produtividade média constante através da fórmula:

$$PR = C_k * D^{-2/3} \quad \text{ou} \quad PR = C_k * (K / T^2)^{-2/3}$$

Onde,

PR = produtividade constante

$D = K / T^2$

Isso significa que a produtividade média constante (PR) está diretamente relacionada ao tempo (T) e esforço do desenvolvimento do sistema (K). Por isto, conclui-se que é difícil medir a produtividade de um projeto com base em outro, por que ambos devem ter o mesmo grau de dificuldade (mesmo número de arquivos, mesmo número de consultas, relatórios...) e o mesmo tempo.

A equação utilizada pela estimativa de Putnam (também chamada de equação do software) apresenta como resultado uma função, fazendo com que o usuário defina valores para determinar a curva (Rayleigh-Norden – Tempo x Esforço) e decidir qual será o esforço e o tempo necessário para o desenvolvimento do software.

Segundo [PIL 97], vários pesquisadores criticam o uso de uma curva de Rayleigh como base para a estimativa de custo e esforço do desenvolvimento de software, pois as observações originais de Norden não foram baseadas em teorias, mas em observação de projetos de hardware.

O método de Putnam é baseado na visão que os projetos de software seguem padrões bem definidos e que podem ser modelados com um conjunto de equações exponenciais [SPC 99].

¹ Ambiente de Desenvolvimento Pobre é um ambiente sem base metodológica, sem documentação e com revisões precárias.

² Ambiente de Desenvolvimento Bom é um ambiente que aplica metodologias, com documentação.

³ Ambiente de Desenvolvimento Excelente é um ambiente com ferramentas e técnicas automatizadas.

3.3 Constructive Cost Model - COCOMO

O modelo COCOMO foi desenvolvido para estimar esforços de desenvolvimento, prazos e tamanho de equipe para projetos de sistemas de informações, a partir de uma amostra de 63 projetos concluídos nas áreas de negócios, controle científico, suporte e sistemas operacionais. É uma técnica já bem documentada e bastante aceita considerando o número de projetos desenvolvidos que a utilizaram.

Barry Boehm [BOE 81] apresenta uma hierarquia de modelos de estimativa de software, da seguinte forma:

Modelo 1. O COCOMO básico é um modelo estático de valor simples que computa o esforço (e custo) de desenvolvimento de software como uma função do tamanho do programa expresso no número de linhas de código estimado.

Modelo 2. O COCOMO intermediário computa o esforço de desenvolvimento de software como uma função do tamanho do projeto e de um conjunto de “direcionadores de custo” que incluem avaliações subjetivas do projeto, do hardware, do pessoal e dos atributos do projeto.

Modelo 3. O COCOMO avançado incorpora todas as características da versão intermediária com uma avaliação do impacto dos direcionadores de custos sobre cada passo (análise, projeto, etc) do processo de engenharia de software.

O COCOMO pode ser aplicado em três tipos de projetos de software. Usando a terminologia de Boehm, são elas:

- ✓ Tipo Orgânico – projetos de software simples, relativamente pequenos, nos quais pequenas equipes com boa experiência em aplicações trabalham num conjunto de requisitos não tão rígidos;
- ✓ Tipo Semidestacado (difuso) – um projeto de software intermediário (em tamanho e complexidade) no qual equipes com níveis de experiência mistos devem atingir uma combinação de requisitos rígidos e não tão rígidos;
- ✓ Tipo Embutido (restrito) – um projeto de software que deve ser desenvolvido dentro de um conjunto rígido de restrições operacionais, de hardware e de software.

Existe um grupo de pesquisa trabalhando para atualizar a técnica COCOMO para os novos processos de desenvolvimento [MAD 97].

O COCOMO utiliza as equações desenvolvidas por Boehm para prever o número de programadores-mês (PM) e o tempo de desenvolvimento (D), os quais são calculados a partir do número de instruções-fonte (Linhas de Código, Pontos por Função, Pontos de Objetos). Os resultados das equações devem ser ajustados a fim de representarem as influências que ocorrem no desenvolvimento do projeto de software no que se refere a atributos do produto, computacionais, de pessoal e de projeto. A técnica utiliza, para caracterizar essas variações, a aplicação de multiplicadores de esforço às estimativas nominais. No Anexo 2 são mostrados os multiplicadores de esforço propostos por Boehm em 81 e na versão estendida do COCOMO, chamado de COCOMO II.

O COCOMO II é adaptado aos ciclos de vida de software modernos. O modelo do COCOMO original teve muito sucesso, mas ele não se aplica às novas práticas de desenvolvimento de software tão bem quanto ele se aplicava às práticas tradicionais. O COCOMO II tem como alvo os projetos de softwares dos anos 1990 e 2000, e vai continuar a evoluir nos próximos anos de acordo com [HON 97].

O COCOMO II incorpora um conjunto de indicadores de custos revisados e indicadores de escala. Essa extensão suporta o planejamento de projeto ao identificar, categorizar, quantificar e priorizar os recursos do mesmo, e detecta também as anomalias de entradas na estimativa de custo, fornecendo aconselhamento no controle do risco além dos cálculos comuns [MAD 97].

As equações do COCOMO básico são:

$$E = a_b (KLOC) \exp(b_b)$$

$$D = c_b (E \exp(d_b))$$

A equação do COCOMO intermediário assume a forma:

$$E = a_i (KLOC) \exp(b_i) \times EAF$$

Onde,

E = é o esforço aplicado em pessoas-mês,

D = é o tempo de desenvolvimento em meses cronológicos,

KLOC = é o número estimado de linhas de código do projeto (expresso em milhares),

EAF = é o fator de ajuste do esforço. Valores típicos para o EAF variam de 0,9 a 1,4.

Os coeficientes a_b , a_i e c_b e os expoentes b_b , b_i e d_b são fornecidos na Tabela 3.3.1.

TABELA 3.3.1 – Coeficientes do COCOMO

Projeto de Software	a_b	b_b	c_b	d_b	a_i	b_i
Orgânico	2.4	1.05	2.5	0.38	3.2	1.05
Semidestacado	3.0	1.12	2.5	0.35	3.0	1.12
Embutido	3.6	1.20	2.5	0.32	2.8	1.20

O COCOMO foi utilizado pela “US Army Organization” como calibrador do modelo de estimativas desenvolvido por essa organização [GRA 99]. O COCOMO possui coeficientes que não são válidos para todas as organizações, mas no estudo foi verificado que os resultados do modelo desenvolvido e do COCOMO foram semelhantes para projetos pequenos.

Barry Boehm está liderando um grupo de acadêmicos que atuam em indústrias para revisar o novo conjunto de equações do COCOMO, e para coletar dados para avaliar hipóteses, selecionando os parâmetros significativos [HON 97].

3.4 Análise de Pontos por Função

A técnica Análise de Pontos por Função foi introduzida por Allan J. Albretch da IBM em 1974. Foram analisadas centenas de softwares para isolar as variáveis críticas e determinar o tamanho do software. Essa medida determina o tamanho do software através das funções executadas pelo software, ao invés de utilizar o volume ou a complexidade do código fonte como base para o cálculo do tamanho.

A técnica de Análise de Pontos por Função (APF) dimensiona uma aplicação na perspectiva do usuário final, ao invés de levar em consideração as características técnicas da linguagem utilizada. APF dimensiona o software quantificando a funcionalidade que ele proporciona aos usuários baseando-se, principalmente, no seu desenho lógico.

Os objetivos da técnica são:

- ✓ Medir o que foi requisitado, e recebido do usuário; independente da tecnologia utilizada para a implementação.
- ✓ Prover uma métrica de medição para apoiar a análise de produtividade e qualidade.
- ✓ Prover uma forma de estimar o tamanho do software e um fator de normalização para comparação de software.
- ✓ Permitir que qualquer usuário ligado a processamento de dados que conheça a aplicação do software possa compreender e utilizar essa técnica.

Além de atingir os objetivos descritos, o processo de contagem de Pontos por Função deve ser simples para minimizar o trabalho adicional do processo de mensuração, e conciso para permitir consistência ao longo do tempo dos projetos e entre os usuários da técnica.

A métrica de Pontos por Função é uma medida padronizada pela Organização Internacional IFPUG que procura ser independente de metodologia, tecnologia e ferramenta de desenvolvimento, preocupando-se com a funcionalidade do ponto de vista do cliente. A contagem do IFPUG inclui cinco etapas [CAS 98]:

- ✓ definição do tipo de sistema a ser contado,
- ✓ identificação das fronteiras do sistema,
- ✓ contagem das bases de dados internas e externas,
- ✓ contagem das funções de entrada, saída e consulta,
- ✓ ajuste da contagem final através dos fatores de ajuste.

O modelo definido por Albretch considera cinco componentes: arquivo, interface, entrada, saída e consulta e atribui uma complexidade e funcionalidade que varia de simples, média e complexa.

Esta técnica possui várias extensões, cada uma com suas características próprias [PRE 95], [FUR 97]. A contagem dos Pontos por Função dentro de uma mesma organização pode também variar dependendo dos aspectos considerados pelas pessoas que fizeram efetivamente a contagem. Isso aparece por causa dos desvios nas avaliações subjetivas [MAT 94]. A forma de minimizar esse problema é definir indicadores padrões para poder encontrar o ponto médio. Outro fator que também influencia na variabilidade da

contagem é a equipe (grupo) do projeto. Para amenizar esse problema é necessário considerar as pessoas como fator de ajuste.

A seguir é mostrado o caminho que deve ser percorrido para se calcular os Pontos por Função:

Passo 1: Contar o número de pontos existentes para cada parte do projeto, contando o número de arquivos, número de interfaces, número de entradas, número de saídas e número de consultas.

Passo 2: Identificar o nível de cada item contado conforme a Tabela de Complexidade do Item (Anexo 3, A.3.1 – A.3.5).

Passo 3: Multiplicar os Pontos por Função de acordo com a classificação encontrada na Tabela de Cálculo dos Pontos por Função (Anexo 3, A.3.6).

Passo 4: Somar todos os Pontos por Função de cada item contado e multiplicado.

Passo 5: Os Pontos por Função devem sofrer um ajuste conforme o ambiente onde serão desenvolvidos. Utilizando a Tabela de Características Gerais (Anexo 3, A.3.7) identifica-se o NI (nível de influência) e aplica-se a fórmula a seguir para encontrar o Fator de Ajuste.

$$\text{Fator de Ajuste} = (\text{NI} * 0,01) + 0,65.$$

Passo 6: Aplicar o Fator de Ajuste nos Pontos por Função calculados a partir da fórmula:

$$\text{PF Ajustados} = \text{PF Brutos} * \text{Fator Ajuste}.$$

Aproximadamente 40 diferentes métodos de Análise de Pontos por Função existem hoje em dia [MAX 00].

Um destes métodos, também muito utilizado, é considerando o componente “interface”, definido por Albretch, como sendo parte do componente “arquivo”, e utilizando níveis de complexidades pré-definidos ao invés de simples, médio e complexo. O cálculo de APF resultante seria:

$$\text{APF} = (4 \text{ x entrada} + 5 \text{ x saída} + 4 \text{ x consulta} + 10 \text{ x arquivos}) * \text{Ck}$$

Onde:

Ck é o Fator de Ajuste (calculado da mesma forma que Albretch definiu)

4, 5, 4 e 10 são os coeficientes utilizados para sistemas de complexidade média.

A métrica de Análise de Pontos por Função foi definida no período do desenvolvimento “Estruturado”. Hoje em dia, as empresas estão optando pelo desenvolvimento “Orientado a Objetos”. Ao desenvolver um projeto orientado a objetos é possível definir o número de pontos de objetos da mesma forma que no desenvolvimento estruturado se define o número de pontos por função. Mas ao invés de verificar as funções a serem desenvolvidas devem ser verificados as classes e eventos a serem programados.

3.5 Pontos de Particularidade

Em 1986, a empresa Software Productivity Research, Inc. desenvolveu um modelo experimental aplicado à lógica de Pontos por Função em processos de software, e sistemas operacionais. Para evitar confusão com o método de Pontos por Função essa alternativa experimental foi chamada de Pontos de Particularidade (Features Points). A técnica de Pontos de Particularidade proposta por Jones [JON 86], é uma extensão da técnica de Pontos por Função.

A métrica de Pontos de Particularidade é um super conjunto da métrica Pontos por Função e introduz mais um parâmetro (número de algoritmos) nos padrões dos Pontos por Função (entrada, saída, consulta, arquivo e interface).

A medida de Pontos de Particularidade é utilizada em aplicações em que a complexidade algorítmica é elevada. Quando se aplica, numa organização, os Pontos por Função e Pontos de Particularidade, os resultados podem ser iguais quando o software desenvolvido não tem uma complexidade muito elevada. Já em casos de desenvolvimento de software complexos (maior número de algoritmos que o número de arquivos) a técnica Pontos de Particularidade apresenta um resultado maior que a de Pontos por Função. Se o número de algoritmos for menor que o número de arquivos, a métrica, Ponto de Particularidade apresentará um resultado menor que a métrica, Pontos por Função.

A maior diferença entre Pontos por Função e os Pontos de Particularidade é o novo parâmetro: número de algoritmos. Algoritmos são definidos dentro do contexto padrão da engenharia de software, como sendo rotinas que executam determinada função dentro do sistema. Alguns exemplos de algoritmos típicos: rotinas de cálculo de datas nos calendários, extração de tabelas e cálculo de salários.

Os parâmetros da métrica “Pontos de Particularidade” são:

Número de Entradas: uma entrada consiste no fornecimento de algum dado do usuário ou da aplicação para o software.

Número de Saídas: uma saída compreende os dados produzidos pela aplicação fornecidos ao usuário, que incluem impressão de relatórios, informações na tela e mensagens de erros. Os componentes individuais dos relatórios ou telas não são contados.

Número de Consultas: uma consulta do usuário é diferente das entradas porque está envolvida com o controle do software, tanto que fornece dados orientados da aplicação.

Número de Arquivos: número de tabelas que estão relacionadas ao projeto.

Número de Interfaces: todas as interfaces através das quais são transmitidas informações para outra parte do sistema são contadas. Estas incluem: discos, impressoras, modems, links, etc.

Número de Algoritmos: abrange todos os algoritmos requeridos para cumprir a especificação.

O método de Pontos de Particularidade é similar aos de Pontos por Função como pode ser visto na Tabela 3.5.1. Cada um dos parâmetros deve ser multiplicado por um fator de complexidade correspondente como mostra a Tabela 3.5.1. Estes fatores podem ser alterados conforme a experiência da organização, a partir de estimativas anteriores.

TABELA 3.5.1 – Cálculo de Pontos de Particularidade

Tipo	Número de Pontos	Complexidade	Pontos X Complexidade
Número de Algoritmos		X 3	
Número de Entradas		X 4	
Número de Saídas		X 5	
Número de Consultas		X 4	
Número de Arquivos		X 7	
Número de Interface		X 7	
Sub Total			
Fator de Ajuste da Complexidade			
Total dos Pontos de Particularidade Ajustados			

A seguir são mostrados quais algoritmos são significativos e devem ser contados:

- ✓ Os algoritmos que estão de acordo com a resolução do problema.
- ✓ Os algoritmos que estão de acordo com a definição do problema.
- ✓ Os algoritmos que possuem entradas ou valores iniciais.
- ✓ Os algoritmos que possuem saídas ou processam um resultado.
- ✓ Os algoritmos que incluem ou chamam algoritmos subordinados a ele.
- ✓ Os algoritmos que representam, via conceitos de programação estruturada padrão as seqüências: if-then-else, do-while, case...

O Total de Pontos por Função e o Total de Pontos de Particularidade podem ser combinados dando o Valor dos Pontos por Função usando a fórmula:

$$VPF = TPF \times [0,65 + (0,01 \times TPP)]$$

Onde:

VPF = Valor de Pontos por Função

TPF = Total dos Pontos por Função

TPP = Total dos Pontos de Particularidade

O Valor de Pontos por Função representa a complexidade do projeto de software. E a organização necessitaria de validações para relacionar os valores de Pontos por Função com o esforço de desenvolvimento do software e o número de defeitos.

Essa técnica acrescenta outros Pontos por Função que também devem ser medidos para facilitar, e tornar mais precisos o cálculo do esforço, do tempo e do custo do projeto.

3.6 Personal Software Process - PSP

O Personal Software Process (PSP), processo derivado do SEI-CMM (Software Engineering Institute – Capability Maturity Model) foi desenvolvido como recurso para capacitação, melhoria e otimização do processo individual de trabalho. Essa nova abordagem é uma grande oportunidade para aplicar conceitos importantes de engenharia de software em nível individual para desenvolver software e não apenas para codificar programas [HUM 95].

Assim como um processo de software é uma seqüência de etapas necessárias para se desenvolver ou fazer manutenções em software, o PSP faz uso de um conjunto de sete etapas seqüenciais e progressivas, como mostra a Figura 3.6.1. Feitas sob medida para desenvolver programas de tamanho modular entre 50 a 5.000 linhas de código. Ou seja, PSP pode ser aplicado desde a construção de um simples programa ao desenvolvimento de um software complexo com um grande número de programas. Cada uma dessas etapas possui um conjunto de roteiros formulários e gabaritos associados (vide Anexo 4).

A Evolução do Processo PSP

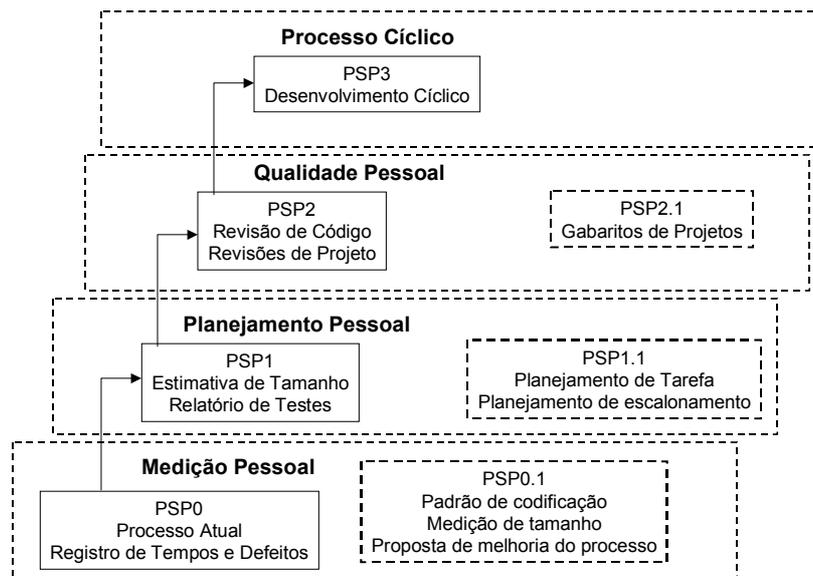


FIGURA 3.6.1 - Evolução do Processo do PSP

O PSP visa demonstrar os princípios do processo pessoal, determinar a situação do processo atual de software, desenvolver um processo de planejamento para desenvolvimento de software, medir o tamanho do software como parte do cronograma e dos recursos necessários para o software, realizar medições apropriadas do processo pessoal, fazer revisões significativas de projeto e código, executar gerenciamento da qualidade do software, estabelecer um padrão de referência para medir as melhorias do processo pessoal e determinar o impacto das mudanças sobre a eficiência profissional.

O PSP introduz os conceitos de processo de software no nível pessoal, em uma série de sete etapas [SIL 97]:

- ✓ Medição Pessoal

PSP0 – Nessa etapa os profissionais de software aprendem a aplicar os formulários e roteiros do PSP aos seus trabalhos pessoais, medindo tempos e defeitos de desenvolvimento, defeitos esses injetados ou removidos. Isso faz com que os engenheiros de software reúnam informações reais e práticas que servirão como ponto de referência contra o qual será medido o progresso enquanto aprendem a praticar o PSP. PSP0 possui três fases: planejamento, desenvolvimento e autópsia, sendo o desenvolvimento composto de projeto, codificação, compilação e teste.

PSP0.1 – Adiciona um padrão de codificação, medição de tamanho e formulário de Proposta de Melhoria do Processo (PMP). Os profissionais de software registram no PMP os problemas, tópicos importantes para discussão e argumentação e as idéias a serem usadas futuramente, aperfeiçoando assim os seus processos pessoais.

✓ Planejamento Pessoal

PSP1 – Introduz o método PROBE (Proxy-Based-Estimating Method), através do qual profissionais de software estimam tamanhos e tempos de desenvolvimento para novos programas, com base nos seus próprios dados pessoais. PROBE é utilizado para indicar a qualidade da estimativa de tamanho e tempo.

PSP1.1 – Adição de escalonamento e de planejamento das tarefas

✓ Qualidade Pessoal

PSP2 – Introduz o gerenciamento de defeitos. Com os dados de defeitos reunidos previamente, os profissionais de software constroem e usam checklists para revisão de produtos e códigos. No treinamento é salientada a importância de focar qualidade desde o início e como revisar eficientemente os programas.

PSP2.1 – Introduz as técnicas de especificação de projeto e análise em adição à prevenção de defeitos, análise e comparação de processos. Medindo os tempos das tarefas realizadas e o número de defeitos injetados ou removidos em cada fase do processo. Os profissionais aprendem a avaliar e melhorar a eficiência pessoal.

✓ Processo Cíclico

PSP3 – É a etapa final do PSP, onde os profissionais de software combinam múltiplos PSP2.1 de uma forma cíclica para construir módulos com milhares de linhas de código. Neste nível, os profissionais exploram os métodos de verificação de projeto, assim como os princípios e métodos de definição de processo.

No Anexo 4, é apresentada uma tabela com as principais tarefas que devem ser seguidas para a implantação do PSP numa empresa.

Um ponto crítico percebido por [SIL 97] é a dependência entre as tabelas do treinamento em relação ao paradigma de programação que está sendo utilizado para implementar (exemplo: procedimentos versus orientação objetos). Conseqüentemente há uma forte dependência dos exercícios em relação à linguagem e ao ambiente de programação.

3.7 Comparativo entre as Técnicas

As estimativas são confiáveis? A única resposta razoável a esta pergunta é: não se pode ter certeza. Qualquer técnica de estimativa, não importa quão sofisticada seja, deve sofrer uma verificação cruzada com outra abordagem. Mesmo assim, o senso comum e a experiência devem prevalecer [PRE 97].

Alguns problemas comuns encontrados com relação às estimativas [JON 91]:

- ✓ As estimativas relativas à precisão de projetos de software são difíceis de serem obtidas, em parte porque os gerentes de projetos não gostam de tornar públicos os seus fracassos.
- ✓ A principal razão, porém, é que a maioria dos gerentes de projeto não fazem as estimativas. Se as fazem, não controlam o objeto estimado. Se controlam o objeto estimado, não fazem os acompanhamentos ou constantemente substituem as estimativas por outras até que as originais sejam esquecidas.

Quando as estimativas são feitas, no entanto, os resultados do controle mostram que geralmente elas tendem a não se concretizar, principalmente porque:

- ✓ As estimativas são realizadas por pessoas que não possuem o conhecimento necessário para aplicá-las.
- ✓ Não se faz as previsões adequadas para contrabalançar os efeitos das distorções.
- ✓ Não se segue exatamente a regra definida na literatura de como se deve estimar um projeto.
- ✓ Não se sabe como lidar com os problemas políticos que dificultam o processo de estimativa, como por exemplo: a inclusão de novas funções em um projeto pode modificar o seu cronograma.
- ✓ Não se baseiam em estimativas de desempenhos passados.

Cada técnica possui as suas próprias características. A Tabela 3.7.1 mostra um comparativo entre as técnicas descritas neste capítulo.

Ao analisar a Tabela 3.7.1, identifica-se que a Estimativa de Putnam, COCOMO, Análise de Pontos por Função e Pontos de Particularidade levam em consideração a tecnologia utilizada pela organização. A Estimativa de Putnam considera a tecnologia para definir a sua constante tecnológica, enquanto as estimativas COCOMO, Análise de Pontos por Função e Pontos de Particularidade utilizam a tecnologia como fator de ajuste de seus cálculos.

O COCOMO é a única técnica que faz distinção em relação ao tipo de projeto. Ele sugere valores diferenciados para cada um deles (orgânico, semidestacado e embutido) obtendo resultados diferenciados.

Análise de Pontos por Função e Pontos de Particularidade são as únicas estimativas onde a linguagem de programação não interfere nos resultados. Elas estimam a funcionalidade do sistema, independentemente da linguagem. Entretanto o projeto físico interfere no cálculo, pois estas técnicas necessitam saber qual o número de arquivos, entradas, saídas, consultas e interfaces para calcular a estimativa. O PSP também necessita das informações sobre o projeto físico do sistema.

O tamanho do projeto interfere em todas as estimativas. Na Estimativa do Esforço, esta interferência é indireta porque ela trabalha com o número de profissionais que executam determinada função do projeto em determinada tarefa do desenvolvimento. Então, quando o projeto é volumoso, o número de profissionais deve aumentar e quando pequeno deve reduzir. Mas o tamanho do projeto não interfere diretamente no cálculo da estimativa. Por este motivo não foi identificado na tabela comparativa que o tamanho do projeto interfira na estimativa.

O COCOMO e a estimativa de Putnam não calculam o custo de um projeto, apenas o esforço e o tempo necessário para o seu desenvolvimento. Enquanto que as demais estimativas permitem acrescentar uma taxa de mão-de-obra definida pela organização sobre algum aspecto do projeto. Por exemplo, a Estimativa do Esforço aplica a taxa de mão-de-obra para cada tarefa do desenvolvimento a ser executada. Já a Análise de Pontos por Função permite que se aplique a taxa de mão-de-obra por pontos.

Os gerentes de projetos desejam controlar o risco do desenvolvimento de um projeto. Ainda não se conhecem todos os riscos dos projetos, mas a estimativa COCOMO já está implementando alguns fatores de risco no cálculo do seu fator de ajuste.

Cada umas das técnicas, aqui apresentadas, possui suas próprias características, mas devem ser modeladas de acordo com a organização que as aplica, pois cada organização trabalha de uma forma diferente.

Pode-se acrescentar nas técnicas aqui apresentadas outros fatores ou elementos para transformar uma técnica de estimativa mais factível para a organização. Por exemplo, uma empresa pode determinar o número de pontos por função ou número de linhas de código que um profissional de determinado grau de escolaridade e tempo de empresa é capaz de produzir. Considerar o tempo gasto em outras atividades, como por exemplo, ligações telefônicas, suporte a clientes e intervalos.

O nível de conhecimento do profissional sobre o projeto, o tempo trabalhado em projetos similares e o custo do profissional também devem ser considerados, bem como o ambiente, a linguagem de programação, a tecnologia de desenvolvimento e as ferramentas de gerenciamento de projeto disponíveis.

TABELA 3.7.1 - Comparação das Técnicas Apresentadas

	Estimativa do Esforço	Putnam	COCOMO	Análise de Pontos por Função	Pontos de Particularidade	PSP
Base da Estimativa	Número de Profissionais para cada tarefa	Tamanho do Software, Constante de Tecnologia.	Tamanho do Software, Fator de Ajuste.	Tamanho do Software, Fator de Ajuste.	Tamanho do Software, Fator de Ajuste.	Tamanho do Software
Tamanho do Produto	—	Linhas de Código	Linhas de Código	Nº Entradas Nº Saídas Nº Consultas Nº Arquivos Nº Interface	Nº Entradas Nº Saídas Nº Consultas Nº Arquivos Nº Interface	Linhas de Código
A tecnologia interfere?	Não ⁴	SIM	SIM	SIM	SIM	Não ⁴
O tipo de projeto interfere?	Não ⁴	Não ⁴	SIM	Não ⁴	Não ⁴	Não ⁴
O tamanho do projeto interfere?	Não ⁴	SIM	SIM	SIM	SIM	SIM
O custo é calculado sobre uma taxa de mão-de-obra definida pela organização?	SIM	Não ⁵	Não ⁵	SIM	SIM	SIM
Possui um fator de ajuste?	Não ⁴	Não ⁴	SIM	SIM	SIM	Não ⁴
Permite o Controle do Risco?	Não ⁴	Não ⁴	SIM	Não ⁴	Não ⁴	Não ⁴
A linguagem de programação interfere?	SIM	SIM	SIM	Não ⁶	Não ⁶	Não ⁶
Projeto Físico interfere?	Não ⁴	Não ⁴	Não ⁴	SIM	SIM	SIM
Leva em consideração a experiência dos profissionais?	SIM	Não ⁴	Não ⁴	Não ⁴	Não ⁴	Não ⁴
Ferramentas	—	SLIM [PRE 95]	BYL WICOMO DECPland [PRE 95]	Estimacs [PRE 95]	SPQR/20 [PRE 95]	PSP Tool [SIL 97]

⁴ Não interfere diretamente no cálculo.

⁵ Na bibliografia consultada não se encontrou caso onde se aplique uma taxa de mão-de-obra sobre o COCOMO ou sobre a Estimativa de Putnam, o que seria possível é aplicá-la sobre o tamanho do projeto, medido em número de Linhas de Código.

⁶ Não interfere diretamente no cálculo, mas ao se definir o número de pontos que um profissional é capaz de fazer, a linguagem de programação interfere.

4 Ferramentas Existentes

Quando se planeja um orçamento e um cronograma para um projeto, e o mesmo não é cumprido, muitas vezes o projeto é forçado a ser abortado ou sacrificar sua qualidade ou ser entregue sem retirar todos os erros [TSO 99].

Para solucionar estes problemas, muitas empresas estão desenvolvendo ferramentas de estimativa de custo para automatizar o processo a fim de medir e acompanhar o andamento do mesmo. Encontram-se várias referências reportando experiências de empresas que utilizam técnicas próprias de estimativa de custo e esforço de desenvolvimentos de software, tais como [GRA 99], [KAU 99], [HAL 97] e [DRA 96].

A seguir serão apresentadas algumas ferramentas de estimativa de custo e esforço que aplicam as técnicas apresentadas no capítulo 3. As ferramentas são as seguintes: ESTIMACS, SLIM, SPQR/20 e ESTIMATE Professional.

4.1 ESTIMACS

ESTIMACS [SAK 98] é um sistema proprietário, projetado para dar estimativas de custo de desenvolvimento em um estágio de concepção do projeto numa visão macro, tendo como base a medida Pontos por Função e os fatores pessoais da equipe de desenvolvimento. Os Pontos por Função são as entradas primárias da estimativa de custo.

O software ESTIMACS usa um questionário contendo 25 questões. Das quais 6 questões se referem ao número de arquivos, número de saídas, número de organizações, número de subsistemas, número de entradas e complexidade lógica (estas 6 questões são as de maior peso para o software). Cada resposta é representada por um valor numérico. As respostas para as questões baseadas nos requisitos do sistema são: entradas para o software obter uma estimativa em horas do desenvolvimento do sistema.

A ferramenta ESTIMACS contém também um conjunto de modelos que habilitam o gerente a estimar:

- ✓ O esforço de desenvolvimento do sistema
- ✓ Definir a equipe e o custo do projeto
- ✓ Definir a configuração de hardware
- ✓ Verificar o risco do projeto

O modelo do esforço de desenvolvimento do sistema combina os dados sobre usuários, desenvolvedores, geografia do projeto, número de funções de negócio, complexidade da aplicação, performance e confiança.

ESTIMACS pode descrever equipes e custos usando como base de dados um ciclo de vida para fornecer a distribuição de trabalho e informação de custo do desenvolvimento.

A configuração de hardware do projeto é medida em tamanho, sendo que o hardware é definido pelas respostas de algumas questões que ajudam o planejador a avaliar o volume de transações, janelas de aplicação e outros dados.

O nível de risco associado com o sucesso da implementação do sistema proposto é determinado basicamente em resposta ao questionário que examina os fatores do projeto tais como tamanho, estrutura e tecnologia.

O software ESTIMACS utiliza-se de um algorítmico heurístico, baseado em milhares de projetos, para apresentar os resultados do novo projeto a ser estimado. A utilização destes projetos históricos dão a ferramenta ESTIMACS segurança e consistência nos resultados apresentados.

4.2 SLIM

SLIM [HON 97] é um sistema de custo automatizado desenvolvido pela Quantitative Software Management. Aplica o modelo de estimativa de Putnam, programação linear, simulação estática, avaliação do programa e a técnica de revisão ou técnica PERT para derivar as estimativas de projetos de software.

O SLIM usa as curvas de Rayleigh separadas para design e código; teste e validação; manutenção e gerenciamento [JOH 98].

O sistema habilita o gerente a desempenhar as seguintes funções:

- ✓ Calibrar o ambiente de desenvolvimento de software local, interpretando dados históricos fornecidos pelo gerente.
- ✓ Criar um modelo de informação do software para ser desenvolvido por extração de características de software básicos, atributos pessoais e considerações ambientais.

Uma vez que o tamanho do software tenha sido estabilizado, o SLIM computa o desvio de tamanho, um perfil de sensibilidade que indica os desvios do custo e esforço em potencial e um checklist da consistência dos dados coletados com os softwares de tamanhos similares. Um gerente pode invocar uma análise de programação linear que considere de difícil desenvolvimento, custos e esforços, solicitando uma distribuição mês-a-mês do esforço e o checklist de consistências dos dados coletados do sistema com os de tamanhos similares.

4.3 SPQR/20

SPQR/20 [WU 97] é uma ferramenta de estimativa de custo desenvolvida pela Software Productivity Research Inc, baseada no modelo pontos de particularidade.

Esta ferramenta faz com que o usuário complete um conjunto simples de questões de escolhas múltiplas que indicam:

- ✓ Objetivo, Escopo, Tipo e Classe do projeto
- ✓ Tipo de aplicação, inovação
- ✓ Facilidade de trabalho
- ✓ Requisitos do programa e do projeto
- ✓ Documentação do usuário
- ✓ Tempo de resposta
- ✓ Experiência da equipe
- ✓ Percentual de reuso do código
- ✓ Linguagem de programação
- ✓ Complexidade lógica do algorítmico
- ✓ Dados de custos relacionados ao projeto.

Além dos dados de saída para outras ferramentas, o SPQR/20 estima também:

- ✓ Número de páginas total de documentação do projeto
- ✓ Potencial de defeitos total para o projeto
- ✓ Eficiência de remoção de defeitos acumulados
- ✓ Número de defeitos total no final (entrega) do projeto
- ✓ Número de defeitos por KLOC.

4.4 ESTIMATE Professional

ESTIMATE Professional é um software desenvolvido pela Software Productivity Centre (SPC), que está sendo utilizado em mais de 1200 companhias [SPC 00]. Esta ferramenta implementa os modelos de Putnam, COCOMO 2.0 e Simulação Monte Carlo.

Para estimar um projeto é necessário:

- ✓ Estimar o tamanho do produto – definir as especificações dos requisitos obtendo o tamanho do produto em número de linhas de código.
- ✓ Definir as prioridades – acertar a lógica da estimativa, definindo o tipo de projeto e as restrições.
- ✓ Criar uma distribuição de probabilidade de cronograma – neste ponto a ferramenta utiliza a Simulação de Monte Carlo para sugerir soluções.
- ✓ Revisar e acertar os resultados – definir o cronograma e custo do desenvolvimento do software o mais próximo do real, baseando-se nas definições do projeto.

O software ESTIMATE Professional permite:

- ✓ Gerar uma estimativa do esforço, cronograma e custo do desenvolvimento,
- ✓ Apresentar uma variedade de opções de estimativas cientificamente,
- ✓ Justificar a sua estimativa em relatórios extensivos,
- ✓ Estimar o tamanho do projeto seja ele pequeno, médio ou grande,
- ✓ Criar um banco de dados histórico,
- ✓ Estimar o tamanho do projeto em linhas de código,
- ✓ Refinar a estimativa inicial conforme o projeto progride,

- ✓ Ter um resumo da estimativa,
- ✓ Exportar as informações para outros aplicativos, tais como Microsoft Project,
- ✓ Customizar qualquer driver de custo (ex: atributos do projeto quanto a confiabilidade, tamanho da base de dados e complexidade).

O software ESTIMATE Professional retira a adivinhação das estimativas dos projetos, ele reconhece a natureza volátil do desenvolvimento de software e leva em consideração esta incerteza e variabilidade usando a Simulação de Monte Carlo.

4.5 Resultados

Cada uma das ferramentas de estimativa automatizadas, apresentadas neste capítulo, conduz para um diálogo com o gerente do projeto para a obtenção das informações sobre o projeto. É a partir destas informações que será efetuada a estimativa do custo e esforço do desenvolvimento de software, gerando tabelas e gráficos a serem analisadas pelo gerente do projeto.

A Tabela 4.5.1 mostra que em todas as ferramentas de estimativa estudadas, os diálogos entre o gerente do projeto e a ferramenta são efetuados através de questionários. Nestes questionários os usuários informam o número de linhas de código ou o número de pontos por função do projeto, informações estas entradas manualmente. Caso o usuário informe um valor incorreto, a estimativa também será incorreta.

A ferramenta ESTIMATE Professional tem uma função que auxilia o cálculo dos pontos por função. O usuário deve informar o número de entradas, saídas, consultas, interfaces e arquivos de acordo com o seu entendimento. Este processo de contagem dos pontos por função pode ser automatizado se a ferramenta de estimativa visualizar o banco de dados do projeto e as funções a serem desenvolvidas.

Tanto a ferramenta ESTIMACS como a ferramenta ESTIMATE Professional relatam o uso de uma base de dados histórica para ajuste das suas estimativas.

As ferramentas estudadas têm como objetivo estimar o esforço e o custo do desenvolvimento do software, permitindo um acompanhamento do projeto. Mas nenhuma delas permite o acompanhamento do profissional, da estimativa e o confronto entre diversas estimativas.

Kemerer reportou que o percentual médio de erros em pessoas-mês estimados foi de 772% no modelo SLIM, 85% no ESTIMACS. Este percentual de erro elevado faz com que a qualidade das estimativas seja questionável. Felizmente ocorre uma melhora significativa a partir das experiências de equipes nas calibrações / ajustes dos modelos [KEM 87].

Uma pesquisa elaborada em 1999 pelo International Journal of the Computer the Internet and Management revelou [TSO 99]:

- ✓ 33,3% das empresas consultadas utilizaram algumas vezes uma ferramenta de estimativa de custo para fazer o planejamento dos seus projetos.

- ✓ 80% das respostas disseram que as empresas têm tempos e orçamentos incorretos aplicando ferramentas de estimativa de custos.
- ✓ 100% sentem que a insuficiência de dados históricos é um fator crítico, para a não utilização de ferramentas de estimativa de custo.

TABELA 4.5.1 – Comparativo entre Ferramentas

	ESTIMACS	SLIM	SPQR/20	ESTIMATE PROFESSIONAL
Base da Ferramenta	Pontos por função. Fatores pessoais da equipe de desenvolvimento.	Estimativa de Putnam. Programação linear. Simulação estática. Avaliação do programa. Técnica de revisão.	Pontos de Particularidade.	Estimativa de Putnam. COCOMO 2.0. Simulação Monte Carlo. Pontos por Função.
Obtenção das Informações	Questionário com 25 questões.	Através do número de linhas de código (LOC).	Questionário.	Questionário.
Permite o gerente	Estimar o esforço e o custo. Definir a equipe e a configuração de hardware. Verificar o risco do projeto.	Calibrar o ambiente de desenvolvimento Criar um modelo de informação do software. Conferir mês-a-mês o andamento do projeto Criar diversos cenários de estimativas.	Identificar o potencial de defeitos do projeto. Identificar a eficiência de remoção dos defeitos. Identificar o número total de defeitos do projeto. Identificar o número de defeitos por KLOC.	Gerar uma estimativa do esforço, cronograma e custo do desenvolvimento de software. Criar um banco de dados histórico. Obter um resumo da estimativa. Exportar qualquer informação para outros aplicativos. Refinar a estimativa inicial de acordo com o progresso do projeto.
Base de dados histórica	Sim	Não	Não	Sim
Fabricante	Proprietário	Quantitative Software Management	Software Productivity Research Inc	Software Productivity Center
Utilização				1200 companhias

5 Ferramenta Desenvolvida

Identificou-se a necessidade de implementar uma ferramenta para estimar custo e o esforço de projetos de desenvolvimento de software, pois as ferramentas existentes (apresentadas no capítulo 4), não atendiam a todas as características, a seguir:

- ✓ Estimativa do custo e esforço do desenvolvimento: estimar um projeto definindo o tempo e o esforço necessário para desenvolvê-lo, de acordo com as características do projeto e de acordo com a disponibilidade da equipe.
- ✓ Acompanhamento dos projetos: possibilita a visualização do projeto durante o seu desenvolvimento, permitindo identificar se o projeto está antecipado ou atrasado, e com estas informações o gerente de projeto pode tomar ações corretivas.
- ✓ Acompanhamento dos profissionais: permitir visualizar a produção da equipe de desenvolvimento individualmente, desta forma identificar os profissionais subestimados e superestimados.
- ✓ Acompanhamento das estimativas: permitir visualizar a estimativa, e modificá-la caso seja necessário; cada estimativa pode ser apropriada para determinado tipo de projeto.
- ✓ Comparativo entre estimativas: permitir comparar diversas estimativas, verificando a que melhor se aplica ao projeto a ser desenvolvido.
- ✓ Armazenamento das bases históricas: permitir que o gerente de projeto tenha conhecimento sobre problemas e soluções utilizados em projetos semelhantes já concluídos.
- ✓ Integração com outros os sistemas: permitir exportar e importar os dados gerados pela ferramenta para outras ferramentas, tais como: Excel, Word, MsProject.

Na verdade, está se propondo uma ferramenta que estime o projeto, mas uma ferramenta que permita o acompanhamento e o controle do desenvolvimento, verificando também a produtividade da equipe, em outras palavras, que permita o planejamento e o controle, por parte do gerente do projeto.

5.1 Ambiente

Para que uma ferramenta de estimativa de esforço e custo do desenvolvimento de software seja utilizada dentro de uma empresa é necessário um ambiente adequado para que a sua utilização não se torne inviável.

A base de uma ferramenta de estimativa é os seus dados, e estes sempre devem estar atualizados. Quanto maior a integração entre a ferramenta de estimativa com o ambiente de desenvolvimento mais fácil se torna a sua utilização.

Uma empresa de desenvolvimento de software possui em seu ambiente de desenvolvimento uma ferramenta de modelagem dos dados e uma ferramenta de desenvolvimento. Para completar este ambiente sugere-se uma ferramenta de estimativa

de custo e esforço, que também permita fazer o acompanhamento e o controle de projetos e da equipe de desenvolvimento.

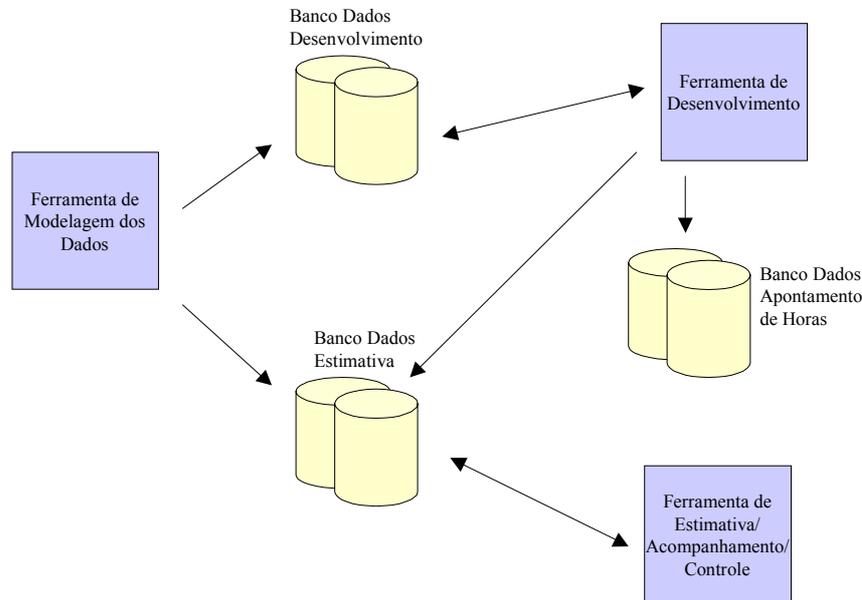


FIGURA 5.1.1 – Ambiente necessário para controle do desenvolvimento

Como a Figura 5.1.1 apresenta, o banco de dados da ferramenta de estimativa deve ser atualizado pelas ferramentas de modelagem dos dados, tais como: Rational Rose ou System Architect e pelas ferramentas de desenvolvimento. Ferramenta de controle dos trabalhos, normalmente são desenvolvidas internamente pela empresa, em conjunto com a ferramenta de desenvolvimento.

A ferramenta de modelagem deve transferir para o banco de dados das estimativas as informações referentes aos projetos, módulos, funções a serem desenvolvidas, e os bancos de dados utilizados pelo projeto, incluindo tabelas e atributos.

Enquanto que a ferramenta de desenvolvimento deve transferir as informações referentes aos trabalhos executados, informações estas sobre o tempo real gasto no desenvolvimento das funções do projeto. Quanto mais próximas às informações de tempo contidas no banco de dados da estimativa estiverem em relação ao tempo real gasto, mais ajustes podem ser feitos nas estimativas, criando uma maior confiança na ferramenta e nos seus resultados.

As informações referentes às horas trabalhadas sobre uma função devem ser obtidas da forma mais automática possível. Pode-se dividir as horas trabalhadas sobre uma função de três formas:

- ✓ Horas efetivamente trabalhadas sobre a função – pode-se considerar o tempo que o programador permaneceu com a função em aberto. Entretanto, quando um programador ficar com uma função em aberto por mais de 15 minutos sem

efetuar nenhuma modificação, a aplicação e o contador de tempo devem ser interrompidos automaticamente.

- ✓ Horas trabalhadas para entendimento da função – devem ser informadas manualmente, pois não é possível definir um processo que automatize a contagem do tempo que o profissional ficou estudando uma função do projeto. Isto porque normalmente na fase de estudo acontecem reuniões, conversas com os usuários/analistas.
- ✓ Horas trabalhadas com testes da função – pode-se definir o mesmo processo utilizado na contagem do tempo trabalhado efetivamente sobre uma função, mas neste caso considerando o tempo de execução da função. Uma rotina de processamento não deve ser cancelada se ultrapassar o tempo de 15 minutos. Somente deve ser cancelada a aplicação se ultrapassar os 15 minutos e nenhum processo sobre a função estiver sendo executado.

O ideal para o ambiente é que esta integração seja automatizada, mas caso isto não seja possível, ela deve ocorrer através de procedimentos internos definidos pela empresa, procedimentos estes de importação e exportação das informações. Deve-se poder garantir que as informações existentes nos bancos de dados sejam reais.

5.2 Definições Técnicas para a Ferramenta

As métricas de estimativa de custo e esforço baseiam-se no tamanho do projeto. Este tamanho pode ser medido através do número de linhas de código, pontos por função ou pontos por objetos.

A Tabela 5.2.1 apresenta um comparativo entre as formas de medir o tamanho de um projeto.

TABELA 5.2.1 - Comparativo das formas de medir o Tamanho do Projeto

	Linhas de Código	Pontos por Função	Pontos por Objetos
A linguagem de programação interfere?	SIM	NÃO	NÃO
Exige o projeto físico definido?	NÃO	SIM	SIM
Independência de Tecnologia?	NÃO	SIM	SIM
Possibilita comparações?	NÃO ⁷	SIM	SIM
Ênfase:	Código	Funcionalidade	Objetos / Eventos
Designado para:	Qualquer Aplicação	Aplicação de Negócios	Aplicação de Negócios
Padrão de Contagem	NÃO	SIM	SIM

Analisando a tabela comparativa (Tabela 5.2.1) percebe-se que Pontos por Função e Pontos por Objetos são semelhantes. A diferença entre eles é uma questão conceitual: ao definir funções utilizando a análise estruturada obtêm-se os pontos por função, enquanto que ao definir os eventos na análise orientada a objetos têm-se os pontos por objetos.

⁷ Se os projetos utilizarem a mesma linguagem de programação é permitida a comparação.

Como medida base para a ferramenta, decidiu-se utilizar a medida de pontos por função para determinar o tamanho do projeto, por permitir uma comparação entre projetos e ser independente da tecnologia utilizada pela empresa desenvolvedora de software.

Tendo a medida de tamanho em pontos por função, decidiu-se implementar as seguintes métricas: Análise de Pontos por Função definida pela IFPUG, Análise de Pontos por Função com níveis de complexidade fixos e Pontos de Particularidade definidos por Capers Jones. Entretanto, outros métodos baseados em pontos por função podem ser implementados pelo próprio usuário da ferramenta. E futuramente pode-se pensar em implementar, métodos baseados em outras formas de medidas de tamanho.

Para que fosse possível um confronto entre a estimativa do projeto e o projeto realmente desenvolvido, deve-se definir uma forma de conversão de pontos por função para horas. Pois o projeto estimado está identificado pelo número de pontos por função, e o projeto realmente desenvolvido está identificado pelo tempo, horas trabalhadas pela equipe de desenvolvimento.

Como cada profissional possui características diferentes em relação ao conhecimento técnico, conhecimento do negócio, tempo de empresa e escolaridade. Optou-se em definir a Tabela 5.2.2 onde a soma dos valores apresentados de acordo com as características define o valor de conversão por profissional (horas x pontos por função). Estes valores podem variar de 8 pontos por hora até 46 pontos por hora.

Não existe uma definição única de conversão de pontos por função para horas, pois isto pode variar dependendo da linguagem de programação utilizada. A Tabela 5.2.2 foi definida com base nos projetos, Custos do Transporte e EDI – Troca eletrônica de Informações, já desenvolvidos da empresa Newsoft Consultoria de Informática, utilizando a linguagem de programação Progress Character.

TABELA 5.2.2 – Conversão de Pontos por Função em Horas

Características	Níveis				
	Muito Fraco	Fraco	Bom	Ótimo	Excelente
Conhecimento Técnico	3	5	7	10	15
Conhecimento do Negócio	3	5	10	15	20
	0 – 6 meses	6 – 18 meses	18 – 36 meses	36 – 72 meses	Acima de 72 meses
Tempo de Empresa	1	2	3	5	1
	2º Grau	Técnico	Superior	Pós-Graduação	Pós-Graduação
Escolaridade	1	2	3	4	4

Por exemplo: Um profissional com dois anos de empresa, com um “bom” conhecimento técnico, com um “ótimo” conhecimento no negócio do projeto e com curso técnico teria como valor de conversão:

Conhecimento Técnico = 7
 Conhecimento do Negócio = 15
 Tempo de Empresa = 3

Escolaridade = 2

Valor de conversão = Soma = 27

Isto significa que a cada hora trabalhada o profissional produz 27 pontos por função. Uma função estimada em 215 pontos por função, este profissional deve desenvolvê-la em torno de 8 horas, na linguagem de programação Progress Character.

A ferramenta desenvolvida utiliza, para armazenamento das informações, o banco de dados Access 2000 da Microsoft, por se tratar de um banco de dados com um custo acessível e com capacidade de armazenamento. Foi utilizada a linguagem de programação JAVA por se tratar de uma linguagem nova e que permite o desenvolvimento de aplicativos para internet.

Na construção da interface se utilizou a ferramenta NetBeans da Sun, por ser uma ferramenta freeware. Para a construção dos gráficos utilizou-se as classes pré-definidas pela Kavachart [KAV 00]

5.3 Estrutura da Ferramenta

Ao analisar as métricas a serem implementadas na ferramenta juntamente com as informações dos projetos e as solicitações de respostas, definiu-se o seguinte diagrama de classes, apresentado na Figura 5.3.1.

Também foi definida uma subdivisão para as funções de um projeto, esta subdivisão classifica a função a ser implementada, a classe referente a esta classificação é a classe Tipo de Função (Ex: cadastro simples, cadastro médio, cadastro complexo, listagem simples, listagem média).

Neste diagrama encontra-se as classes Banco de Dados e Tabela que são referentes aos bancos de dados utilizados pelos projetos a serem desenvolvidos. Estas informações são necessárias para automatizar o processo de contagem dos pontos por função. Como por exemplo, o projeto de Custos do Transporte utiliza-se dos bancos de dados: TRC (Transporte Rodoviários de Cargas), MGADM (Magnus administrativo), CUSTO. Do banco TRC é utilizada as informações referente a Tabela de Veículo, do banco MGADM é utilizada as informações referentes a Tabela de Estabelecimento e as demais informações necessárias são retiradas do banco CUSTOS, específico para este projeto.

A classe Função Tabela serve como elo de ligação entre a função e a tabela do banco de dados, nesta classe se determina o tipo de ligação que está sendo feita: entrada, saída, consulta ou interface. Esta informação interfere no cálculo do número de pontos por função de um projeto, porque para calcular o número de pontos é necessário saber que tipo de ligação ocorre entre a função e a tabela.

As classes Profissional, Profissional Histórico e Profissional Férias referem-se às informações da equipe de desenvolvimento.

A classe Serviço serve para definir que tipo de serviço está sendo executado (Ex: consultoria, programação ou análise).

A métrica de estimativa é definida pelas classes Estimativa, Nível de Complexidade e Fator de Ajuste.

Ao estimar um projeto utilizando uma métrica de estimativa, os resultados são armazenados nas classes: Histórico Projeto, Histórico Módulo, Histórico Função e Fator de Ajuste do Projeto. Da forma como foi definido, caso ocorra alguma modificação na estrutura do projeto, ele pode ser reestimado sem perder as informações da primeira estimativa.

As informações referentes aos tempos reais trabalhados estão armazenadas na classe Apontamento. E ao finalizar uma função a ferramenta agrupa estas informações de tempo por profissional e as armazena na classe Função Profissional.

A classe Histórico Revisão armazena as revisões feitas sobre uma determinada estimativa de projeto. Esta classe serve para verificar o andamento do projeto em relação a uma determinada estimativa do projeto. Um projeto pode ser acompanhado por várias estimativas, isto permite melhor um confronto entre as estimativas.

5.4 Casos de Uso

Analisando os objetivos da ferramenta a ser desenvolvida, definiu-se os seguintes casos de uso: Parametrizar Sistema, Cadastrar Projeto, Apontar Horas Trabalhadas, Estimar Projeto, Estimar Função, Avaliar Projeto, Avaliar Estimativa e Avaliar Profissional (Figura 5.4.1).

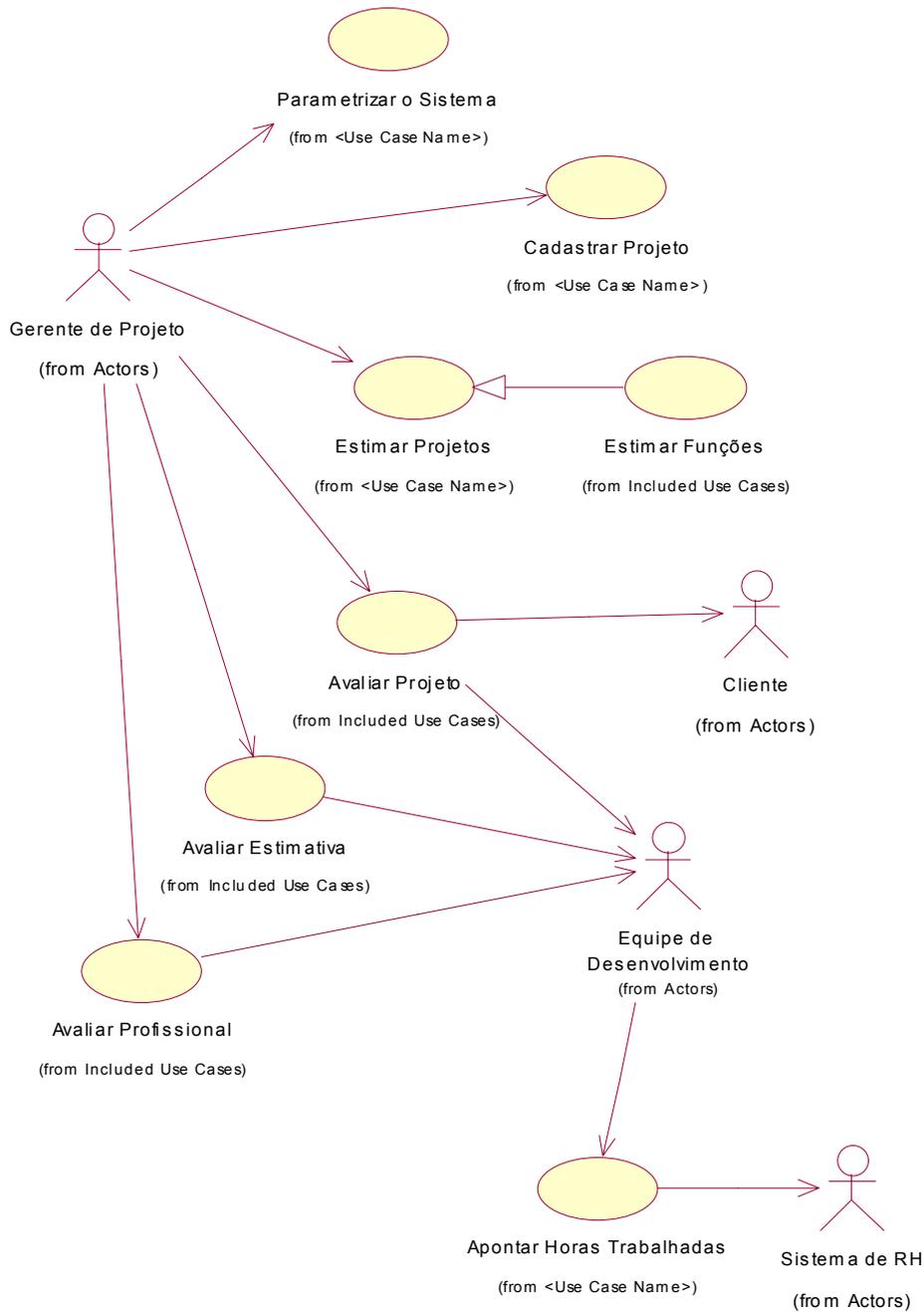


FIGURA 5.4.1 – Diagrama de Casos de Uso

O caso de uso Parametrizar Sistema tem, como objetivo, o cadastramento das classes de apoio, tais como o cadastramento dos Profissionais, Estimativas, Tipos de Funções e Serviços.

Enquanto que o caso de uso Cadastrar Projetos se responsabiliza pelas informações referentes ao projeto, quais os seus módulos, funções, pontos de revisão e profissionais alocados.

O caso de uso Apontar Horas Trabalhadas tem exatamente a função de armazenamento das horas trabalhadas pelos profissionais sobre as funções do projeto.

O caso de uso Estimar Função está contido no caso de uso Estimar Projeto, porque quando se estima um projeto, na verdade, está se estimando todas as funções pertencentes a ele.

Avaliar Projetos, Avaliar Estimativas e Avaliar Profissionais são os casos de uso referentes ao controle do desenvolvimento. Será a partir destas informações que ações corretivas devem ser tomadas.

5.5 Funcionalidades da Ferramenta

De acordo com o diagrama de classes e o diagrama de casos de uso, a ferramenta de estimativa desenvolvida implementa as seguintes funcionalidades: Definição de Parâmetros, Cadastramento das Informações Pessoais, Cadastramento dos Projetos, Estimativas e Acompanhamento. No Anexo 5 encontram-se os diagramas de atividades da ferramenta.

A Definição de Parâmetros implementa os métodos de estimativas utilizados pela ferramenta, tais como Pontos por Função definidos pelo IFPUG, Pontos por Função com nível de complexidade fixo ou Pontos de Particularidade. E os tipos de funções permitidas, tipos de funções é uma subdivisão de acordo com as características e complexidade da função (ex: Cadastro Simples, Cadastro Normal, Cadastro Complexo, Cadastro Muito Complexo). A complexidade varia de acordo com o número de informações de entrada, saída, necessários para o desenvolvimento da função.

Os Cadastros referem-se às informações definidas pela empresa, informações estas sobre os profissionais e serviços executados por eles. Caso a empresa já possua estas informações em um outro banco de dados, por exemplo, no banco de dados de recursos humanos, deve ser feita a integração das informações.

A Definição do Projeto a ser estimado, é uma parte importante no processo de estimativa, pois é a partir esta definição que será calculado o número de pontos por função de acordo com a métrica de estimativa selecionada. A definição de um projeto consiste no cadastramento do cliente que solicitou o projeto, no cadastramento do próprio projeto com seus módulos (subdivisões), funções, pontos de revisão, profissionais alocados, banco de dados utilizados, tabelas dos bancos de dados, relacionamento entre as funções e tabelas, e apontamento de horas.

As informações que puderem vir da integração com a ferramenta de modelagem dos dados e da ferramenta de desenvolvimento devem ser utilizadas. É mandatário, que o gerente do projeto sempre fique sabendo das modificações do projeto, pois estas modificações podem alterar as estimativas.

Hoje, a ferramenta permite fazer, uma estimativa por projeto, ou módulo do projeto ou uma simulação de uma função específica. É permitido estimar um projeto através de várias estimativas. Ao modificar o projeto no decorrer do desenvolvimento é necessário reestimá-lo.

Em relação ao acompanhamento, a ferramenta tem três objetivos:

- ✓ Acompanhamento do projeto – verificar como está o andamento do projeto, se está sendo cumprido o prazo e o custo de acordo com o que foi definido (pontos de revisão).
- ✓ Acompanhamento do profissional – verificar a produtividade da equipe de desenvolvimento individualmente.
- ✓ Acompanhamento da estimativa – verificar a utilização do modelo de estimativas, comparando o previsto com o real e fazendo o confronto de várias estimativas sobre um mesmo projeto.

Os acompanhamentos devem ser visualizados pelos gerentes de projetos de forma gráfica, para facilitar a tomada de decisões. Eles podem ser diários, semanais, quizenais, mensais, bimestrais, isto varia de acordo com a necessidade do projeto e da empresa desenvolvedora.

A partir das informações de acompanhamento o gerente do projeto deve conseguir concluir, se o projeto está no prazo, se o problema é com a equipe de desenvolvimento ou com a estimativa, ou se o projeto deve ser reestimado. Ações corretivas devem ser tomadas a partir destas informações, o gerente do projeto também deve considerar a sua experiência nas decisões.

5.5.1 Parametrização

Nos Parâmetros encontram-se os programas para Parametrização das Estimativas, e o Cadastro de Tipos de Funções.

A Parametrização das Estimativas tem como objetivo o cadastramento das fórmulas de cálculo das estimativas. As estimativas são baseadas em valores, níveis de complexidade e fatores de ajustes. Estas informações podem e devem ser modificadas, de acordo com a necessidade de cada empresa. Ao cadastrar os fatores de ajuste de uma estimativa, o usuário deve informar o valor mínimo e máximo para o respectivo fator. Quando forem definidos os valores dos fatores de ajuste de um projeto, somente serão válidos os valores que estiverem neste intervalo. Pode-se estimar um projeto sobre várias técnicas de estimativas diferentes, permitindo desta forma um confronto entre as estimativas, verificando a que melhor se adapta ao projeto.

O cadastro de Tipos de Funções tem como objetivo criar uma subdivisão das possíveis funções do projeto. (Ex: Cadastros Simples, Cadastros Médios, Cadastros Complexos, Listagens, Atualizações, Relatórios, etc) A partir desta subdivisão, pode-se verificar melhor o desenvolvimento do sistema, pois a ferramenta de estimativa desenvolvida permite a visualização das informações de acordo com esta subdivisão, isto significa que o gerente de projeto pode verificar que tipo de função está causando erros na estimativa do projeto. Neste programa de cadastramento de Tipos de Funções também são informados os valores padrões a serem acrescidos, como entrada, saída, consulta, interface e algoritmo. Estes devem ser definidos pela empresa desenvolvedora, analisando os padrões de programação (ex: o número de mensagens de erro que deve ser considerado como saída). Estas informações serão repassadas para o cadastro de funções do projeto.

5.5.2 Cadastramento das Informações Pessoais

Nos Cadastros das Informações Pessoais encontram-se os programas para cadastramento dos Profissionais e dos Serviços executados pela empresa desenvolvedora do software.

O cadastro de Profissionais tem como objetivo o cadastramento das pessoas envolvidas no processo de desenvolvimento do software. Neste cadastro são informados os períodos de férias do profissional como as mudanças salariais, portanto está relacionado diretamente ao custo do projeto. Também é identificado o valor do ponto por função padrão do profissional, isto é, significa quantos pontos o profissional é capaz de desenvolver em 1 hora de trabalho. Esta informação será sugerida posteriormente na alocação do profissional ao projeto.

O cadastramento dos Serviços executados pela empresa tem como objetivo definir quais os tipos de serviços que a empresa desenvolvedora oferece para os seus clientes (Ex: consultoria, programação, análise, etc). Neste cadastro encontram-se os valores padrões cobrados pela empresa pela execução do serviço, valor do salário padrão do profissional que o executa, como também o valor do ponto por função. Caso o profissional não possua valor do ponto por função o valor informado no serviço, será utilizado como sugestão na alocação do profissional ao projeto.

5.2.3 Cadastramento dos Projetos

No Cadastramento dos Projetos encontram-se os programas para cadastramento de Clientes, Projetos, Funções do Projeto, Banco de Dados, Tabelas dos Bancos de Dados, Relacionamento entre as Funções e as Tabelas como também o cadastramento dos Apontamentos de Horas (horas trabalhadas realmente no projeto).

O cadastro de Clientes tem como objetivo o cadastramento das empresas que solicitaram os projetos a serem desenvolvidos e estimados. O cliente pode negociar um valor de serviço diferenciado, desta forma é necessário que se faça o cadastramento

destes valores, para que a obtenção do custo do projeto estimado seja próximo do custo real.

No cadastro de Projetos encontram-se as informações sobre o projeto a ser desenvolvido e estimado. Ao cadastrar um projeto também é necessário que se faça o cadastramento dos módulos. Eles podem ser entendidos como sendo uma subdivisão do projeto. Também deve ser determinada a alocação dos profissionais que trabalharão no projeto, juntamente com as suas características. Desta forma será possível identificar o custo do projeto mais precisamente. Neste cadastro também é possível definir quais serão os pontos de revisão do projeto, para relacionar o estimado com o produzido.

Já, o cadastro de Funções de um Projeto tem como objetivo o cadastramento das funcionalidades que o projeto irá ter. Uma função pode ser comparada com um programa ou evento. Ao informar qual o tipo de função com a qual está relacionada, de acordo com a finalidade da função e da sua complexidade, a ferramenta herda as informações armazenadas no cadastramento do Tipo de Função. Informações estas referentes ao número de interfaces, número de algoritmos, número de entradas, número de saídas e de consultas a serem acrescentadas no cálculo da estimativa. Uma função pode ser cancelada, isto significa ser ignorada nos cálculos das estimativas ou finalizada significando que a função já foi desenvolvida totalmente e é possível fazer uma comparação entre o custo previsto e o real.

O cadastro de Banco de Dados tem como objetivo o cadastramento dos bancos de dados que serão utilizados pelos projetos a serem estimados.

O cadastro das Tabelas dos Bancos de Dados tem como objetivo o cadastramento das classes implementadas no banco de dados, que serão utilizadas pelas funções do projeto a ser desenvolvido. Ao cadastrar as tabelas o usuário também deve informar quais são seus atributos. Estas informações serão utilizadas posteriormente nos cálculos das estimativas.

O cadastro de Relacionamentos entre as Funções e as Tabelas é prioritário para a estimativa. Pois é a partir deste relacionamento, que se consegue obter as informações referentes ao número de entradas, número de saídas, número de consultas, número de interfaces e número de arquivos com que a função se relaciona. Ao relacionar uma função a uma tabela, o usuário deve informar que tipo de relacionamento existe entre eles (entrada, saída, consulta ou interface). Uma tabela pode estar relacionada por vários tipos, é a partir desta informação que se consegue definir o número de entradas, saídas, consultas, interfaces e arquivos relacionados.

Já o Apontamento de Horas tem como objetivo armazenar as informações reais do projeto quanto ao profissional que trabalhou e o tempo trabalhado. Com estas informações será possível fazer um relacionamento entre o custo previsto e o real.

5.5.4 Estimativas

Nas Estimativas encontram-se os programas para estimar um projeto, um módulo do projeto, e o programa para estimar uma função específica do projeto (simulação).

Estimar um projeto significa estimar o seu tamanho considerando todos os seus módulos e conseqüentemente as suas funções. Estimar significa definir o número de pontos por função de um projeto / módulo / função. Ao estimar um projeto, o usuário deve definir os fatores de ajustes do projeto, fatores estes que interferem no cálculo da estimativa. Toda estimativa é armazenada em um banco de dados histórico para poder ser utilizada posteriormente. Será a partir das informações calculadas, que o gerente do projeto poderá verificar o andamento e definir novos cronogramas.

Estimar um módulo de um projeto significa estimar todas as funções que o compõem. Na verdade a ferramenta não estima um projeto ou um módulo, mas sim uma função, a menor parte de um projeto. A soma das estimativas das funções que compõem um módulo, é a estimativa do módulo. Conseqüentemente a soma das estimativas dos módulos resulta na estimativa do projeto.

Estimar uma função significa identificar o número de entradas, saídas, consultas, interfaces, arquivos e algoritmos relacionados a ela. Cada valor encontrado (número de entradas, número de saídas, número de consultas, número de interfaces, número de arquivos e número de algoritmos) é multiplicado por um fator de complexidade definido pela estimativa. A soma dos valores multiplicados resulta no número de pontos por função brutos de uma função. Para se conseguir o número de pontos por função líquido da função, é necessário aplicar um fator de ajuste determinado por um questionário sobre o ambiente e sobre as características do projeto, previamente definido pelo usuário.

O gerente de projeto deve conseguir visualizar os resultados destas estimativas de forma numérica ou gráfica. A Figura 5.5.4.1 apresenta um exemplo da consulta numérica dos pontos por função do módulo 1 - Cadastros do projeto “Custos do Transporte”. Pode-se observar que nesta figura são apresentados os números de pontos por função estimados para cada uma das funções que compõem o módulo 1 – Cadastros. Na figura são apresentados também os números de pontos por função já concluídos. Isto ocorre porque o projeto já foi desenvolvido e finalizado. O botão “gráfico” permite que o gerente de projeto visualize as mesmas informações, mas de forma gráfica através de um gráfico de barras.

Dados Estimados da Fase

Cliente:

Projeto:

Módulo:

Estimativa:

Sequencia:

Função	Estimados	Concluidos
CUM - Cadastrar Combustivel	71.98	50.0
FUN - Cadastrar Funcoes de Transport	90.28	25.0
LIC - Cadastrar Licenciamento	114.679999	40.0
LUB - Cadastrar Lubrificantes	137.859999	40.0
MAR - Cadastrar Marcas dos veiculos	71.98	20.0
MOD - Cadastrar modelos dos veiculos	231.799999	60.0
PAR - Cadastrar Parametros do Módulo	126.88	40.0
PNE - Cadastrar Medida de Pneu	71.98	30.0
TLUB - Cadastra Tipo de Lubrificante	71.98	43.5
TOP - Cadastrar Tipo de Operação	96.38	30.0
VEI - Cadastra Veiculo	498.979999	70.0
TOTAL	1584.78	448.5

OK Gráfico

Pontos

Tempo

FIGURA 5.5.4.1 – Tela de Consulta da Estimativa de um Módulo.

O gerente de projeto também deve conseguir visualizar os valores utilizados nos cálculos da estimativa de uma função, pois desta forma o gerente consegue identificar se o cálculo da estimativa está correto. A Figura 5.5.4.2 apresenta um exemplo do cálculo da função SCS – Simulação do Custo do projeto Custos do Transporte. Pode-se observar que nesta figura são apresentados os números de pontos de entrada, saída, consulta, interface, arquivos e algoritmos, como também os fatores de complexidades utilizados como multiplicadores. Ao final são apresentados os números de pontos por função brutos, qual o fator de ajuste utilizado e o número de pontos por função líquidos.

Detalhes da Estimativa de uma Função

Cliente:
 Projeto:

Módulo:
 Função:

Estimativa:
 Sequencia:

Profissional Alocado:

Tipos	Pontos	Compl	Pt x CL
Entrada:	<input type="text" value="91"/>	<input type="text" value="5"/>	<input type="text" value="455"/>
Saída:	<input type="text" value="25"/>	<input type="text" value="5"/>	<input type="text" value="125"/>
Consulta:	<input type="text" value="20"/>	<input type="text" value="4"/>	<input type="text" value="80"/>
Interface:	<input type="text" value="5"/>	<input type="text" value="10"/>	<input type="text" value="50"/>
Arquivos:	<input type="text" value="16"/>	<input type="text" value="10"/>	<input type="text" value="160"/>
Algoritmos:	<input type="text" value="5"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Pontos Brutos:

Fator de Ajuste:

Pontos Líquidos:

FIGURA 5.5.4.2 – Tela de Consulta da Estimativa de um Função

5.5.5 Acompanhamento

No Acompanhamento encontram-se os programas para efetuar as revisões do projeto, visualizar a alocação dos profissionais no projeto, a produtividade do profissional, o método de estimativa e fazer um comparativo entre as estimativas com o real.

O gerente pode efetuar o acompanhamento do projeto através de pontos de revisão, “milestones”, previamente definidos no cadastro de projetos. Ao efetuar uma revisão a ferramenta calcula todos pontos que até aquele momento foram desenvolvidos, e os armazena na classe Histórico Revisão. A partir das informações dos “milestones” e dos pontos desenvolvidos, pode ser gerado o gráfico apresentado na Figura 5.5.5.1.

Analisando o gráfico apresentado (Figura 5.5.5.1), o gerente do projeto pode visualizar o andamento do projeto. Se ele está atrasado ou adiantado em relação ao que foi programado, caso seja necessário ações corretivas devem ser tomadas. O cliente sempre deve ser informado sobre o andamento do projeto.

A Figura 5.5.5.1 apresenta um exemplo do gráfico de andamento do projeto Custos do Transporte, sobre a métrica de estimativa de Pontos de Particularidade. A linha tracejada apresenta o projeto estimado e a linha contínua apresenta os dados reais do projeto.

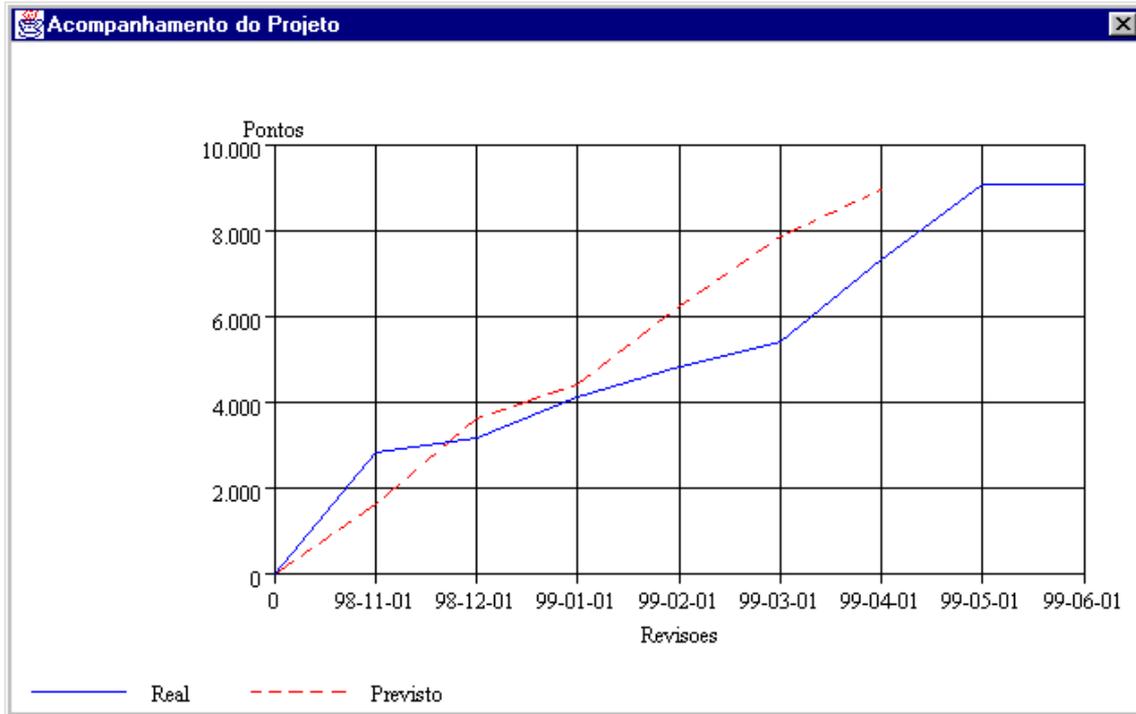


FIGURA 5.5.5.1 – Tela de Pontos de Revisão do Projeto

O programa de acompanhamento da alocação dos profissionais no projeto, é demonstrado através de um gráfico de Gantt (Figura 5.5.5.2). Este gráfico pode auxiliar o gerente de projeto na criação dos “milestones”, da seguinte maneira: ao definir as datas de revisão, o gerente de projeto pode emitir o gráfico de Gantt visualizando a alocação dos profissionais em determinado projeto e desta forma ter uma idéia melhor de quanto do projeto já deve estar concluído. Caso o projeto se prolongue, novas alocações podem surgir, como novos pontos de revisão ou “milestones”.

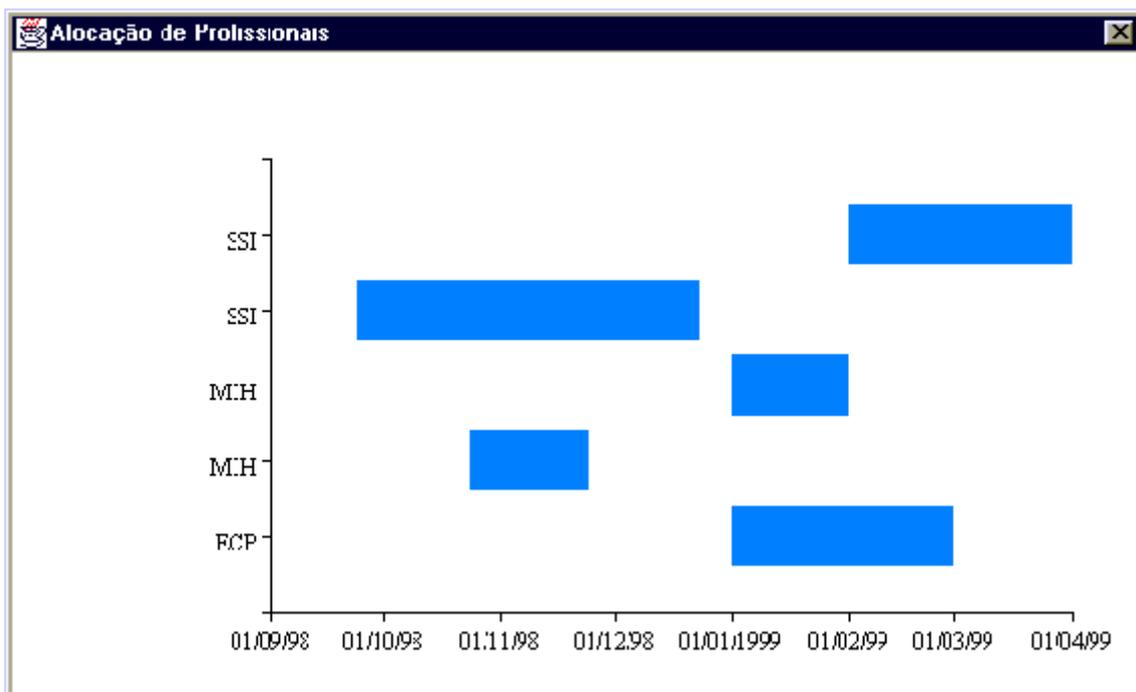


FIGURA 5.5.5.2 – Tela de Alocação dos Profissionais no Projeto

O gráfico de Gantt é formado por barras horizontais que representam o período de tempo trabalhado ou previsto. Na Figura 5.5.5.2 apresenta a alocação prevista dos profissionais, SSI, MIH e ECP no projeto Custos do Transporte. O profissional SSI foi alocado em dois períodos diferentes de tempo, 24/09/1998 até 24/12/1998 e 10/02/1999 até 01/04/1999, cada alocação é indicada por uma barra, no gráfico. A profissional MIH também foi alocada em dois períodos e a profissional ECP em apenas um período. O nome do profissional é apresentado no eixo y do gráfico, e no eixo x são apresentadas as datas de alocação dos profissionais.

A produtividade do profissional pode ser observada num confronto das informações sobre os dados estimados e das informações reais (Figura 5.5.5.3). A produtividade estimada de um profissional pode estar incorreta por dois motivos: a ferramenta está subestimando o profissional ou, realmente, ele não está produzindo o esperado. O gerente do projeto deve sempre levar em consideração a sua experiência, caso o problema esteja na ferramenta, o gerente deve reavaliar os parâmetros do profissional (número de pontos).

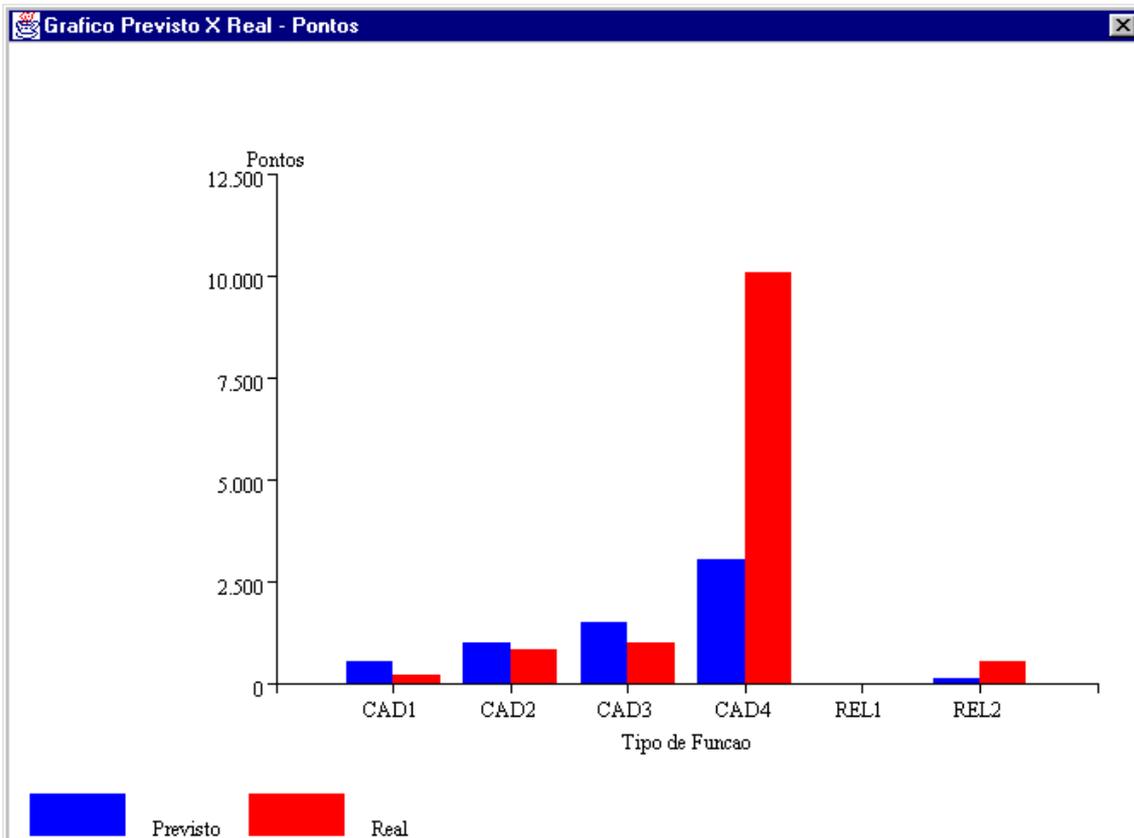


FIGURA 5.5.5.3 – Tela de Acompanhamento do Profissional

A Figura 5.5.5.3 apresenta um exemplo do gráfico de acompanhamento do profissional SSI, para o projeto Custos do Transporte, este gráfico é apresentado sobre os tipos de funções desenvolvidos pelo profissional (CAD1 = Cadastros Simples, CAD2 = Cadastros Médios, CAD3 = Cadastros Complexos, CAD4 = Cadastros Muito Complexos, REL1 = Relatórios Simples, REL2 = Relatórios Médios). Pode-se observar no gráfico, que ocorreu uma diferença significativa na função CAD4 - Cadastros Muito Complexos, isto porque a métrica de estimativa utilizada subestimou as funções deste tipo (CAD4 – Cadastros Muito Complexos).

O programa que acompanha a estimativa, é um programa que confronta as informações estimadas com as informações reais do projeto (Figura 5.5.5.4). O gerente pode selecionar visualizar os dados do projeto pelos módulos ou pelos tipos de funções, de acordo com a sua necessidade.

A Figura 5.5.5.4 apresenta o acompanhamento da estimativa de Pontos de Particularidade para o projeto Custos do Transporte. Neste exemplo estão sendo demonstradas as informações de acordo com os tipos de funções existentes no projeto (CAD1 = Cadastros Simples, CAD2 = Médios, CAD3 = Complexos, CAD4 = Muito Complexos, LIS1 = Listagem Simples, LIS2 = Média, LIS3 = Complexa, REL1 = Relatórios Simples e REL2 = Médios). Novamente pode ser observado que a métrica de estimativa subestimou as funções do tipo CAD4 – Cadastros Muito Complexos.

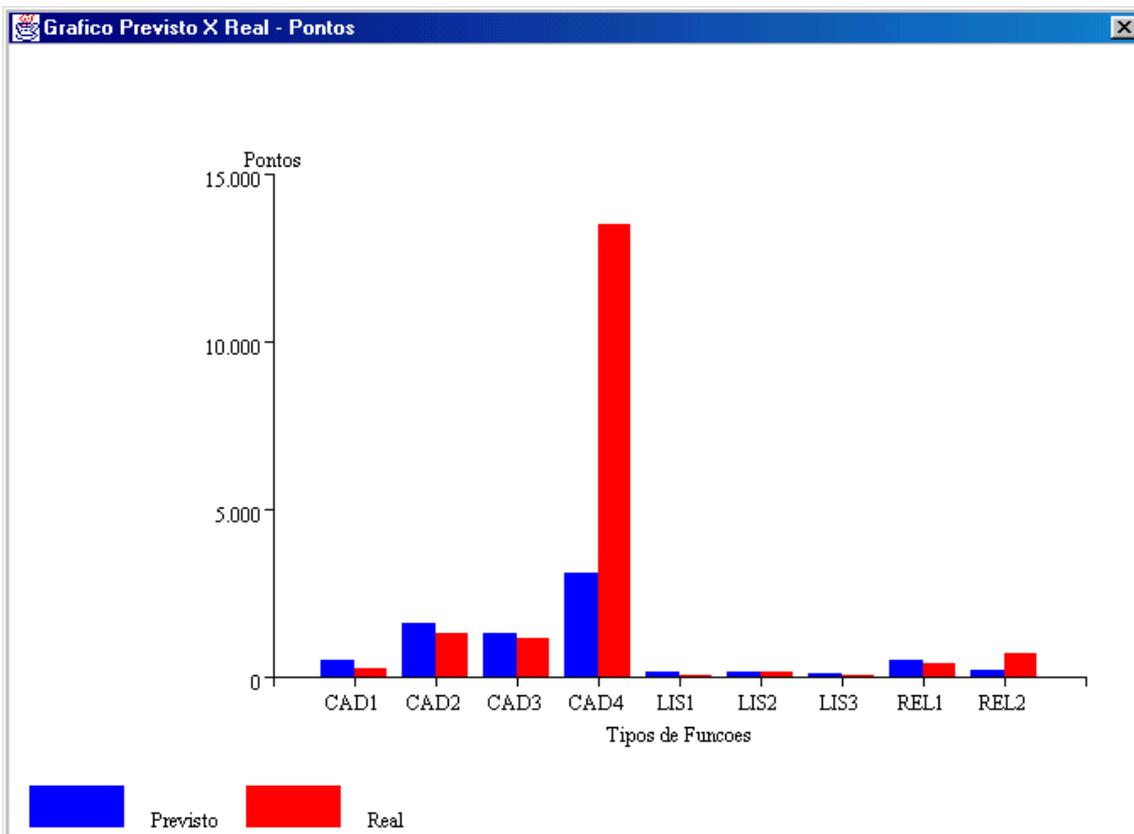


FIGURA 5.5.5.4 – Tela de Acompanhamento da Estimativa

O comparativo entre estimativas sobre um mesmo projeto tem como objetivo, identificar o melhor método de estimativa para este tipo de projeto. A melhor forma de visualizar este comparativo é através de um gráfico de linhas, se o projeto já estiver concluído é interessante que as informações reais do projeto também sejam avaliadas.

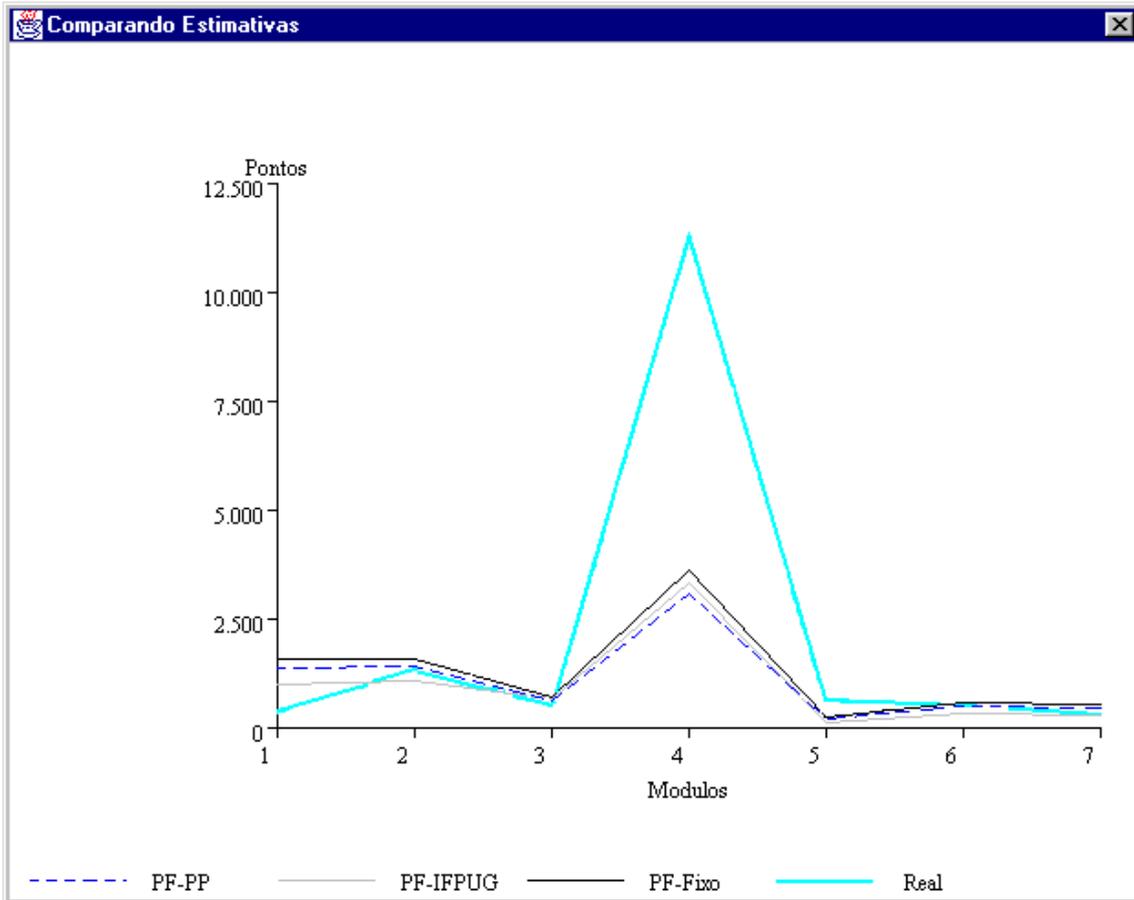


FIGURA 5.5.5.5 – Tela de Comparação entre Estimativas

A Figura 5.5.5.5 apresenta um exemplo do comparativo entre as métricas PF-PP (Pontos de Particularidade), PF-IFPUG (Pontos por Função definidos pelo IFPUG) e PF-FIXO (Pontos por Função com Nível de complexidade Fixo) para o projeto Custos do Transporte. No eixo X estão descritos os módulos que compõem o projeto (1 – Cadastros, 2 – Atualização, 3 – Operação, 4 – Simulação, 5 – Relatórios das Operações, 6 – Listagem das Atualizações, 7 – Listagem dos Cadastros). E no eixo y são identificados os números de pontos por função calculados para cada módulo do projeto.

5.6 Busca de Informações numa Base Histórica

As ferramentas de estimativa devem manter uma base de dados histórica dos projetos estimados para que posteriormente as estimativas possam ser utilizadas / visualizadas, auxiliando o gerente de projeto na escolha do melhor método de estimativa.

Para a busca de uma estimativa de um projeto numa base de dados histórica, pode-se utilizar o conceito de Raciocínio Baseado em Casos - RBC.

Raciocínio Baseado em Casos é uma técnica que se prova ter sucesso em uma grande variedade de áreas de aplicação [KRA 97]. Um sistema de RBC resolve novos problemas adaptando soluções que foram utilizadas para resolver problemas anteriores [KOL93].

Num primeiro estágio deve-se permitir que o gerente visualize as estimativas de projetos semelhantes, representados como casos e armazenados em um repositório. Futuramente, pode-se implementar procedimentos que, a partir de projetos já estimados, se solucione estimativas de projetos novos utilizando raciocínio baseado em casos propriamente dito.

Um sistema de RBC funciona comparando a descrição de um problema a ser resolvido com os casos descritos em uma base de casos. O sistema recupera então o caso mais parecido segundo critérios de similaridade previamente definidos e apresenta a estimativa associada ao usuário. Pela descrição do caso recuperado e solução associada, o usuário decide como fazer a adaptação da solução ao novo problema. Esse procedimento pode ser representado através de um ciclo (Figura 5.6.1, modificada a partir de [ABE 96]).

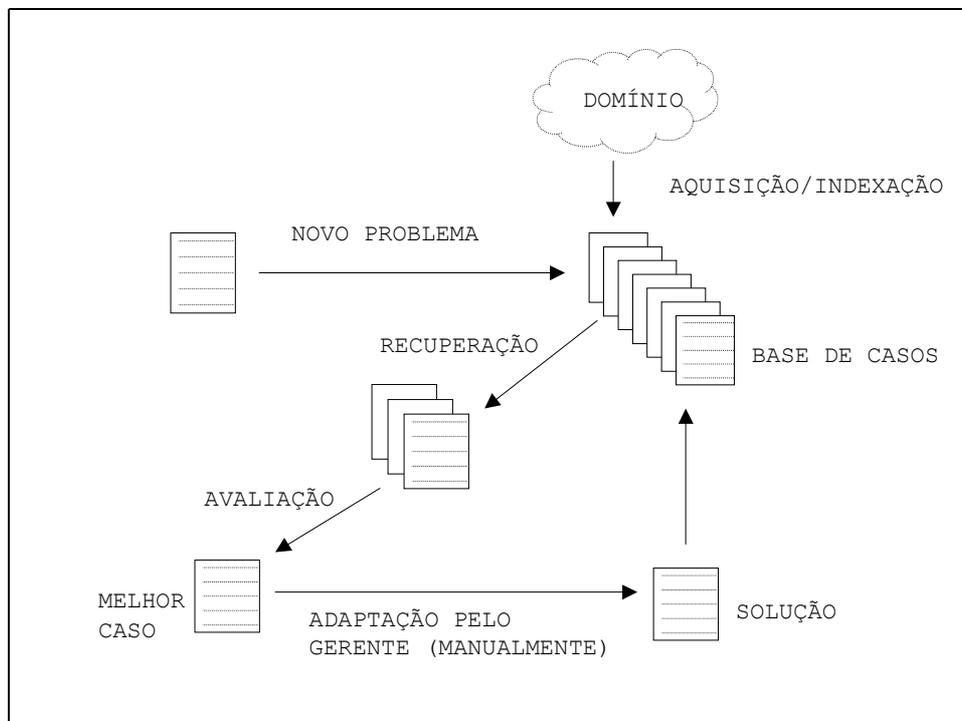


FIGURA 5.6.1 – Ciclo do RBC

A base de casos é construída utilizando-se casos significativos, ou seja, que apresentem descrições e soluções distintas, e descritos apenas pelos atributos relevantes na definição da estimativa de cada projeto. Normalmente a base de casos resulta da replicação de uma base de dados histórica, cujos erros, informações não relevantes e casos duplicados são eliminados.

O modelo de casos descreve os parâmetros de comparação entre os casos de uso. O modelo de casos da ferramenta de estimativa de custo e esforço de desenvolvimento utiliza uma representação estruturada, conforme a Figura 5.6.2.

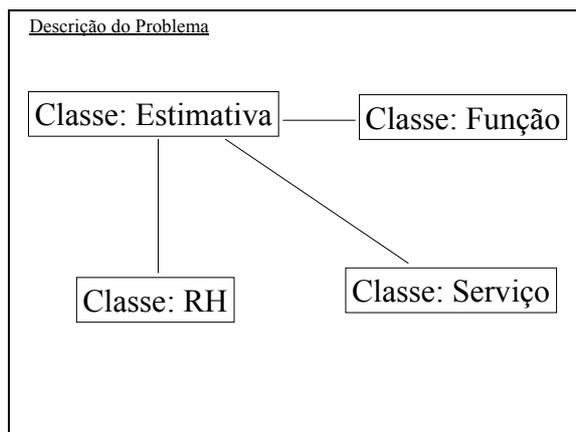


FIGURA 5.6.2 – Representação Estruturada - RBC

O modelo de casos está descrito na Figura 5.6.3, na qual encontram-se as classes:

Classe Estimativa: é a super classe do caso, está relacionada com as demais classes e nela também é armazenada a solução do problema. Como não está se propondo uma adaptação automatizada ao caso mais semelhante à solução também deve fazer parte do problema.

Classe Função: é utilizada para identificar o número de funções para cada tipo de função predeterminada pelo sistema

Classe Serviço: é utilizada para identificar o número de profissionais de acordo com o serviço por ele executado.

Classe RH: é utilizada para identificar o perfil dos profissionais

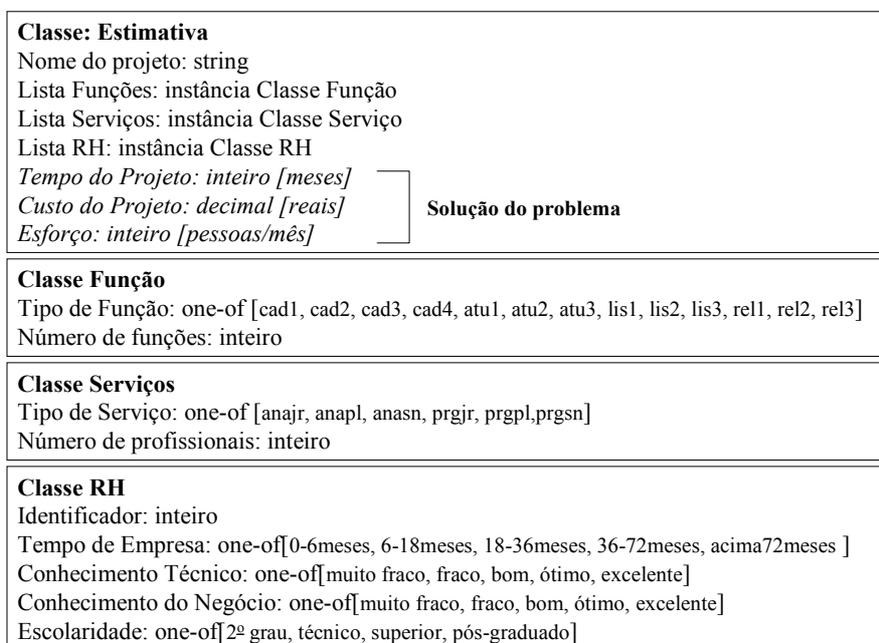


FIGURA 5.6.3 – Modelo de Caso da Ferramenta de Estimativa

Para recuperação do melhor caso, ou o caso mais semelhante, utiliza-se o método de comparação “Algoritmo de vizinhança”, que mede a diferença de valores de cada um

dos atributos definidos no modelo de casos com os existentes na base de dados histórica (ou base de casos).

Para os tipos numéricos (inteiros), utiliza-se o seguinte algoritmo de similaridade, considerando uma distribuição uniforme para os valores dos atributos.

$$\text{Similaridade}(A1P, A1C) = \frac{1 - |A1C - A1P|}{(\text{val max} - \text{val min})}$$

Onde:

A1P = Atributo 1 do Projeto a ser estimado.

A1C = Atributo 1 do Caso constante na base de casos.

Val max = valor máximo que o atributo pode ter.

Val min = valor mínimo que o atributo pode ter.

E para os tipos simbólicos utiliza-se o seguinte algoritmo.

Se atributos possuem valores diferentes, similaridade = 0, se atributos com valores iguais, similaridade = 1.

O caso que obtiver a maior soma das similaridades é escolhido como o melhor caso, e este deve ser apresentado para o gerente, para que ele possa analisar. O gerente pode ou não aceitá-lo como sendo um caso semelhante.

A Figura 5.6.4 apresenta uma descrição de um caso que deve ser “preenchida” pelo gerente de projeto para a busca de um caso semelhante.

Modelo de Casos – Ferramenta de Estimativa

Nome do Cliente: _____
 Nome do Projeto: _____

<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">Tipo da Função</td> <td style="width: 50%; text-align: center;">No. Funções</td> </tr> <tr> <td style="border-bottom: 1px solid black;"> </td> <td style="border-bottom: 1px solid black;"> </td> </tr> <tr> <td style="border-bottom: 1px solid black;"> </td> <td style="border-bottom: 1px solid black;"> </td> </tr> <tr> <td style="border-bottom: 1px solid black;"> </td> <td style="border-bottom: 1px solid black;"> </td> </tr> </table>	Tipo da Função	No. Funções							<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">Serviço</td> <td style="width: 50%; text-align: center;">No. Profissionais</td> </tr> <tr> <td style="border-bottom: 1px solid black;"> </td> <td style="border-bottom: 1px solid black;"> </td> </tr> <tr> <td style="border-bottom: 1px solid black;"> </td> <td style="border-bottom: 1px solid black;"> </td> </tr> <tr> <td style="border-bottom: 1px solid black;"> </td> <td style="border-bottom: 1px solid black;"> </td> </tr> </table>	Serviço	No. Profissionais						
Tipo da Função	No. Funções																
Serviço	No. Profissionais																

No. Profissional	Tempo/Empresa	Conhec/Técnico	Conhec/Área	Escolaridade

Tempo do Projeto: _____
 Custo do Projeto: _____
 Esforço do Projeto: _____

FIGURA 5.6.4 – Descrição de um de Caso para Ferramenta de Estimativa

Um ponto muito importante em um sistema de RBC, que não pode ser esquecido, é em relação à confiança nos casos existentes na base de dados. Devem ser instâncias de casos reais, de modo a que a base de casos seja representativa da solução do problema.

6 Estudos de Casos Submetidos

Após a ferramenta de estimativa de custo e esforço do desenvolvimento de software ter sido concluída, foram escolhidos dois projetos, desenvolvidos pela empresa Newsoft Consultoria de Informática Ltda, para mostrar a utilidade da ferramenta desenvolvida, apresentando os resultados obtidos.

A empresa Newsoft desenvolve um sistema chamado de TRC (Transporte Rodoviário de Cargas) para empresas transportadoras de mercadorias. A linguagem de desenvolvimento é a linguagem de programação Progress Character, em ambiente Unix.

Os projetos analisados foram “Custos do Transporte” e “EDI – Troca de Informações de forma Eletrônica”, ambos os projetos serão incorporados ao sistema TRC.

6.1 Custos do Transporte

O projeto “Custos para o Transporte” tem como objetivo identificar todos os custos envolvidos diretamente e indiretamente na transferência de mercadorias. Este sistema deve ser integrado com o sistema geral TRC. A partir do custo, o sistema deve ser capaz de apresentar o valor ideal de preço a ser cobrado do cliente partindo de um percentual de lucro sobre os custos.

Este projeto foi desenvolvido em parceria com a Transportadora Condor de São Bento do Sul, utilizando o banco de dados Progress para armazenamento dos dados, e foi desenvolvido na linguagem de programação Progress Character.

O sistema de “Custo do Transporte” divide-se basicamente em três partes principais:

- ✓ Cadastros – Veículos, Marca, Modelo, Combustível.
- ✓ Atualização de Valores – Despesas administrativas, valores do combustível, licenciamento.
- ✓ Simulações – Preço, Custo, Viagem e Previsão de Viagem

Sendo que a terceira parte “Simulações” é a mais complexa, exigindo bastante conhecimento sobre a área.

6.1.1 Definições do Custo do Transporte

Para o projeto “Custo do Transporte”, primeiramente foi alocado um analista MIH, um programador SSI e um estagiário ECP.

O analista MIH possuía as seguintes características:

- ✓ 36 a 72 meses de trabalho na empresa
- ✓ Ótimo conhecimento na tecnologia
- ✓ Bom conhecimento sobre o negócio

- ✓ Com curso superior completo

O programador SSI possuía as seguintes características:

- ✓ 18 a 36 meses de trabalho na empresa
- ✓ Ótimo conhecimento sobre a tecnologia
- ✓ Bom conhecimento sobre o negócio
- ✓ Com curso técnico completo

O estagiário ECP possuía as seguintes características:

- ✓ 6 a 18 meses de trabalho na empresa
- ✓ Pouco conhecimento sobre a tecnologia
- ✓ Pouco conhecimento sobre o negócio
- ✓ Com curso superior completo

A partir destas características definiu-se que o analista MIH desenvolveria 25 pontos por hora trabalhada o programador SSI 23 pontos por hora trabalhada e o estagiário ECP 12 pontos por hora trabalhada, conforme a tabela apresentada no Capítulo 5.2.

O projeto “Custos do Transporte” foi dividido nos seguintes módulos e funções:

- ✓ Cadastros: Parâmetros do Módulo, Marca, Modelo, Veículo, Combustível, Tipo de Lubrificante, Lubrificantes, Medida dos Pneus, Funções de Transporte, Licenciamento e Tipo de Operação.
- ✓ Atualizações: Despesas Administrativas, Ficha Técnica, Salários dos Funcionários, Valor do Combustível, Valor do Lubrificante, Valor do Licenciamento, Valor da Lavações e Valor dos Pneus.
- ✓ Operações: Operações com e sem Veículo
- ✓ Simulações: Simulação do preço, do custo, da viagem e da viagem prevista.
- ✓ Relatórios das Operações: Relatório das operações com e sem veículo.
- ✓ Listagem das Atualizações: Listagem das informações referentes aos valores cadastrados na rotina Atualizações.
- ✓ Listagem dos Cadastros: Listagem das informações cadastrais da rotina Cadastros.

As funções de simulação do custo, do preço, da viagem e da viagem previstas foram consideradas programas com alta complexidade, porque internamente apresentam fórmulas matemáticas complexas e informações de diversas tabelas.

No desenvolvimento deste projeto, houve preocupação com relação ao desempenho do projeto, reutilização do código, facilidade a mudanças no sistema e facilidades operacionais. O usuário do sistema deve ser uma pessoa com conhecimento sobre a área de Custos e com conhecimento sobre a tecnologia (fórmulas de cálculo) para conseguir analisar corretamente as informações produzidas pelo sistema, e efetuar simulações úteis para a transportadora.

6.1.2 Resultados

Este projeto implementa funções com alto nível de complexidade, principalmente aquelas que se referem às simulações (módulo 4 do projeto de Custos do Transporte).

A ferramenta de estimativa desenvolvida permite uma comparação das estimativas Pontos de Particularidade (PF-PP), Pontos por Função definidos pelo IFPUG (PF-IFPUG), Pontos por Função com nível de complexidade Fixo (PF-Fixo) com o tempo real gasto no desenvolvimento do projeto. A Tabela 6.1.2.1 foi montada com base nas informações apresentadas pela ferramenta para o projeto de Custos do Transporte.

Ao analisar estas informações, concluindo que todas as métricas (PF-PP, PF-IFPUG e PF-Fixo) subestimaram o projeto. A métrica PF-PP apresentou um percentual de erro em torno de 40%, enquanto que a métrica PF-IFPUG apresentou 44% de erro e a métrica PF-Fixo 30% de erro.

TABELA 6.1.2.1 – Tabela Comparativa das Estimativas – Custo do Transporte

Módulo	PF - PP		PF - IFPUG		PF - Fixo		Real
	Pontos	% erro	Pontos	% erro	Pontos	% erro	
1 - Cadastros	1.440,56	(221,20)	1.001,62	(123,33)	1.584,78	(253,35)	448,50
2 - Atualizações	1.440,82	(6,77)	1.111,42	17,64	1.623,82	(20,33)	1.349,50
3 - Operações	627,08	(11,98)	675,88	(20,69)	719,80	(28,54)	560,00
4 - Simulações	4.362,72	61,75	4.860,48	57,38	5.292,36	53,59	11.404,50
5 - Relatório das Operações	248,88	63,40	161,04	76,32	287,92	57,66	680,00
6 - Listagem das Atualizações	536,80	2,75	351,36	36,35	624,64	(13,16)	552,00
7 - Listagem dos Cadastros	491,65	(31,28)	319,64	14,65	583,16	(55,72)	374,50
Total	9.148,51	40,47	8.481,44	44,81	10.716,48	30,27	15.369,00

Se o módulo 4 – Simulações (módulo responsável pelo maior desvio das estimativas) fosse desconsiderado, os percentuais de erro das estimativas seriam os seguintes: para a métrica PF-PP 19% de erro e a métrica PF-Fixo 36% de erro, ambas super estimaram o projeto. Enquanto que a métrica PF-IFPUG apresentou 8,66% mas subestimou o projeto.

A métrica PF-Fixo aplica um fator de complexidade igual a três para os algoritmos, e um fator de complexidade dez para arquivos e interfaces. Se o fator de complexidade aplicado aos algoritmos fosse modificado para dez, igual ao fator de complexidade dos arquivos e interfaces, a métrica de estimativa apresentaria um percentual de erro igual a 25%.

Para aperfeiçoar a ferramenta de estimativa, deve-se analisar os valores apresentados na Tabela 6.1.2.1 de acordo com o tipo da função a ser desenvolvido, pois existem grandes variações de percentual de erro, como pode ser observado no módulo de 1- Cadastros 220% de erro na estimativa de Pontos de Particularidade. 120% de erro na estimativa de Ponto por Função do IFPUG e 250% na estimativa de Pontos por Função Fixo.

A ferramenta de estimativa de custo e esforço permite também com que o gerente do projeto, visualize as informações contidas na Tabela 6.1.2.1 de forma gráfica, como apresentado na Figura 6.1.2.2. Graficamente é possível perceber o grande desvio causado pelo módulo 4 – Simulações.

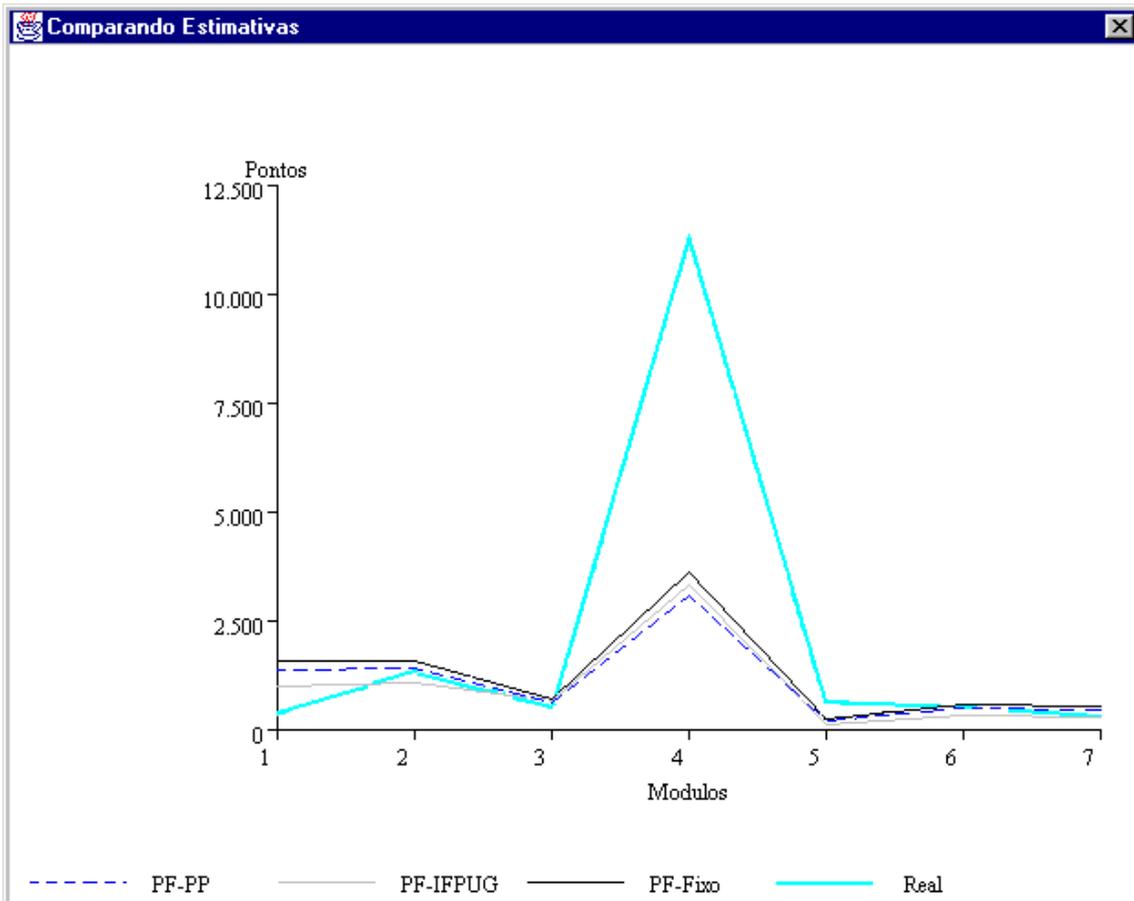


FIGURA 6.1.2.2 – Comparativo entre Estimativas no projeto “Custos do Transporte”

Outra forma do gerente de projeto visualizar a estimativa do projeto é a partir de um gráfico de barras, que compara somente as informações de uma métrica selecionada com as informações reais do projeto. A Figura 6.1.2.3 apresenta o gráfico de barras da estimativa PF-Fixo no projeto de Custos do Transporte. Da mesma forma pode-se visualizar o grande desvio da estimativa em relação ao módulo 4 – Simulações.

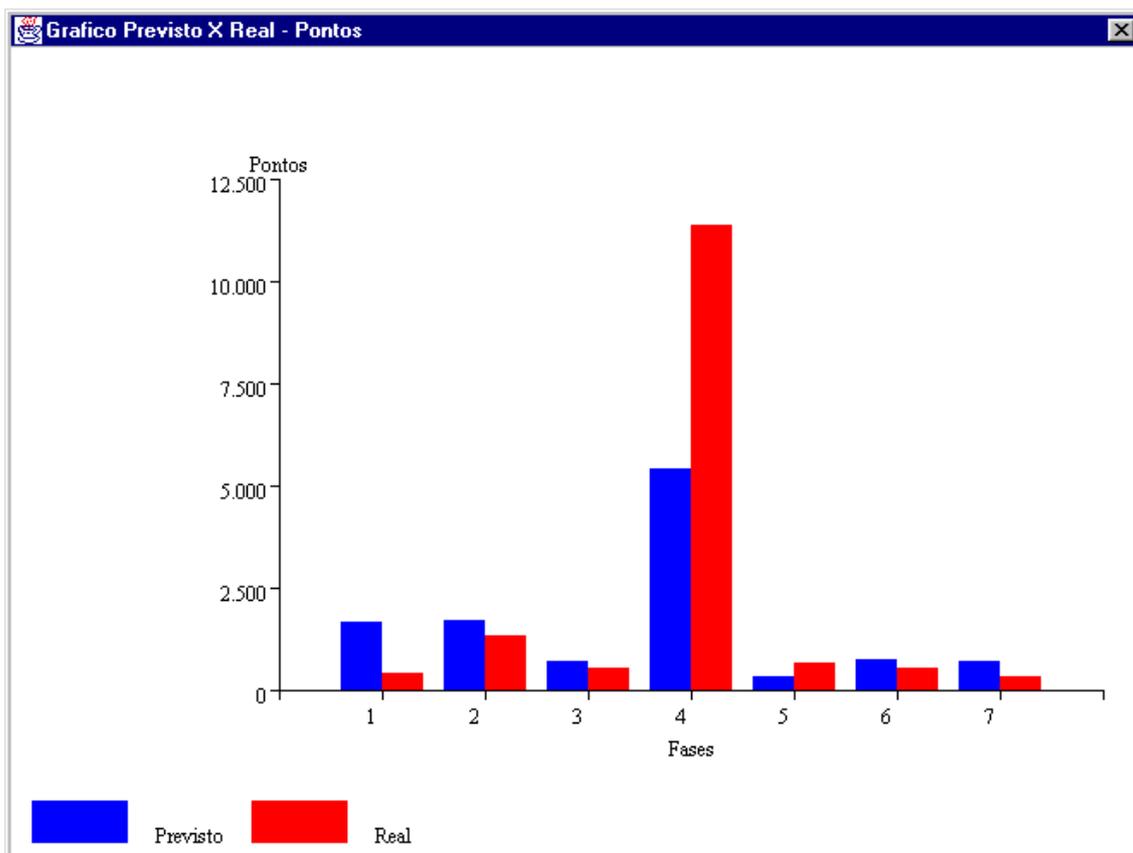
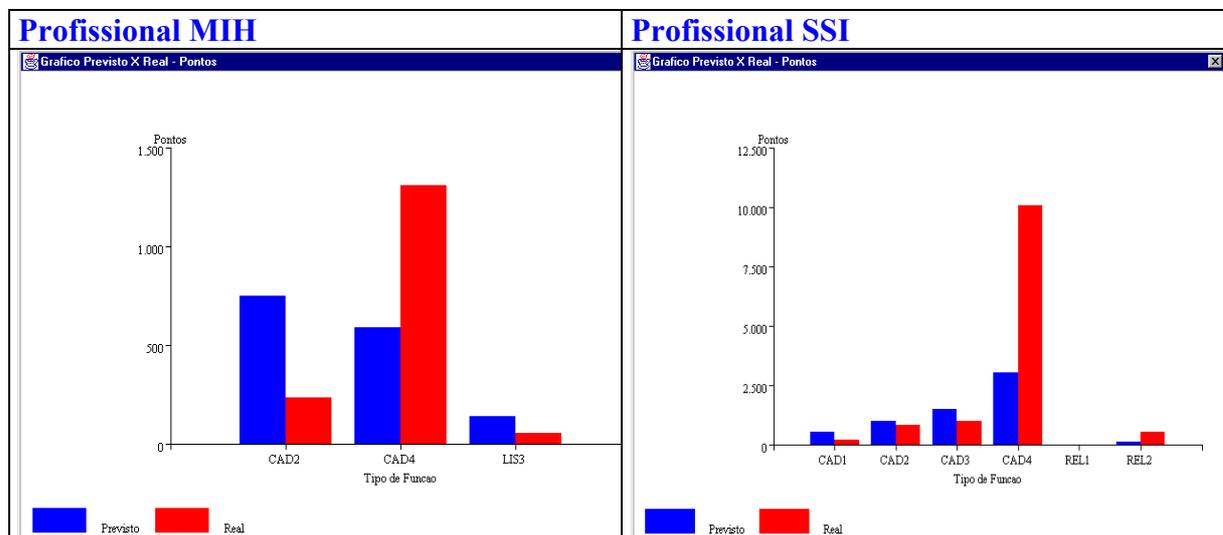


FIGURA 6.1.2.3 – Comparativo da Métrica PF-Fixo modificada para o Projeto de Custos do transporte.

Analisando a produtividade dos profissionais envolvidos (MIH, SSI e ECP) verificou-se que o profissional SSI, produz normalmente o que foi estimado, ocorrendo desvios nas funções com alto nível de complexidade. O profissional MIH foi subestimado para as funções de cadastros médios, mas também apresentou um desvio quanto as funções de alto nível de complexidade. Enquanto que o profissional ECP foi subestimado em relação às funções de nível simples e médio de complexidade. Conforme apresentado nos gráficos da Figura 6.1.2.4.



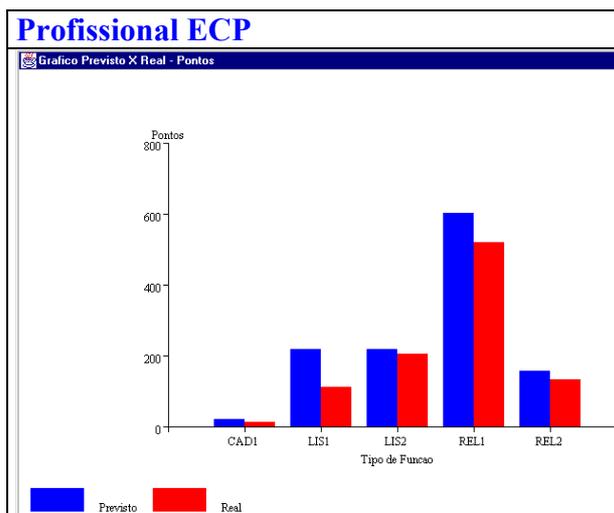


FIGURA 6.1.2.4 – Acompanhamento dos Profissionais no projeto Custos do Transporte

A princípio, o projeto foi estimado para ser desenvolvido em seis meses, com a alocação dos profissionais MIH, SSI e ECP, conforme a Figura 6.1.2.5. As barras consideram o tempo que o profissional foi alocado para o projeto.

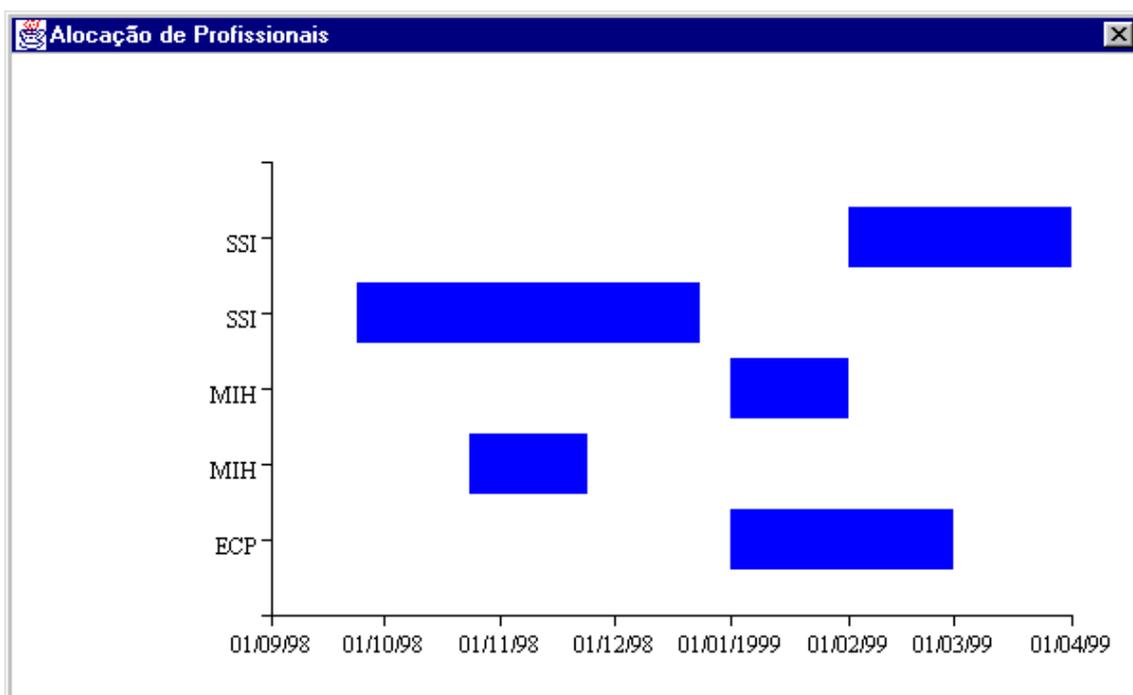


FIGURA 6.1.2.5 – Alocação dos Profissionais no Projeto Custos do Transporte

Os pontos de revisão ou “milestones” foram definidos mensalmente, por se tratar de um projeto com um cronograma de seis meses de desenvolvimento. A Figura 6.1.2.6 apresenta o andamento do projeto, utilizando a métrica de estimativa PF-Fixo. A linha tracejada significa o estimado, e a linha corrente significa as informações reais do desenvolvimento do projeto. Pode-se observar que do período 01/11/1998 até 01/12/1998 não foi produzido o que foi estimado, mas os atrasos no projeto realmente ocorreram porque no período de 01/01/1999 até 01/03/1999 não ocorreu o desempenho esperado, o que causou um atraso no projeto em torno de um mês.

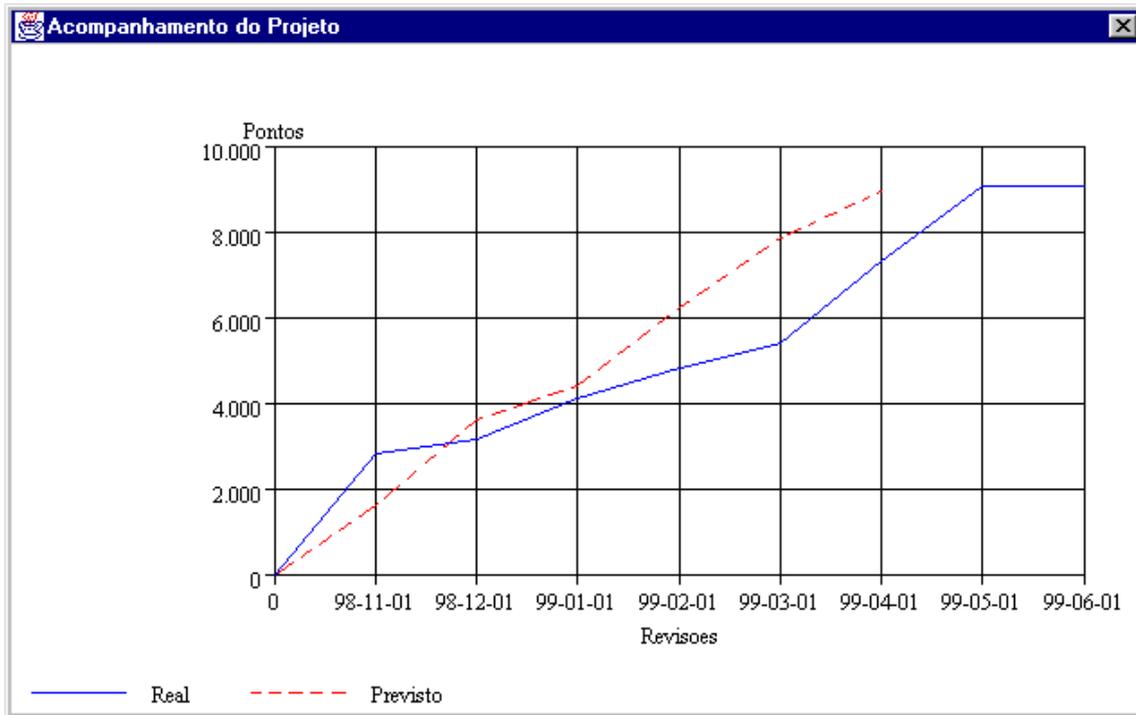


FIGURA 6.1.2.6 – Acompanhamento do Projeto – Custos do Transporte

6.2 EDI – Troca de Informações de forma Eletrônica

O projeto EDI – Troca de Informações de forma Eletrônica tem como objetivo permitir a troca de informações entre as empresas transportadoras e seus clientes, seguindo um padrão predefinido.

Esta troca de informações serve para agilizar o processo de transporte de mercadorias como também serve para o gerenciamento das informações.

Este projeto também foi desenvolvido utilizando a linguagem de programação Progress Character, acessando um banco de dados Progress.

O projeto EDI se dividia em três partes principais:

- ✓ Cadastramento dos layouts (como os arquivos de comunicação estão definidos, esta definição pode ser obtida através de um padrão de comunicação, exemplo Proceda ou ser uma definição proprietária do cliente)
- ✓ Importação dos arquivos
- ✓ Exportação dos arquivos

As funções deste projeto foram trabalhosas, por serem genéricas, tentando atender a diversos tipos de layouts diferentes.

6.2.1 Definições do EDI

Para o projeto “EDI – Troca de informações de forma eletrônica”, foram alocados um analista LA, e um estagiário ECP.

O analista LA possuía as seguintes características:

- ✓ 18 a 36 meses de trabalho na empresa
- ✓ Ótimo conhecimento na tecnologia
- ✓ Bom conhecimento sobre o negócio
- ✓ Com curso superior completo

O estagiário ECP possuía as seguintes características:

- ✓ 6 a 18 meses de trabalho na empresa
- ✓ Pouco conhecimento sobre a tecnologia
- ✓ Pouco conhecimento sobre o negócio
- ✓ Com curso superior completo

A partir destas características definiu-se que o analista desenvolveria 23 pontos por hora trabalhada e o estagiário 12 pontos por hora trabalhada, conforme a tabela apresentada no capítulo 5.2.

O projeto “Custos do Transporte” foi dividido nos seguintes módulos:

- ✓ Cadastros: Cadastro de clientes, Eventos de Entregas, Status de Entrega, Eventos do TRC, Layout e Caminho de Busca.
- ✓ Listagens: Listagem dos cadastros.
- ✓ Movimentação: Importar arquivos e exportar arquivos de acordo com os layouts gerados.
- ✓ Consulta: Consultar a movimentação importada e exportada.
- ✓ Especiais: Gerar os programas a serem executados pelas rotinas de importação e exportação.

Depois de alguns estudos, o módulo de consulta foi cancelado, pois com as consultas básicas já existentes no sistema TRC era possível repassar as informações necessárias para os usuários, não havendo a necessidade de se implementar novas funções de consulta.

As funções especiais de geração de programas de importação e exportação são programas com complexidade considerável, porque internamente apresentam muitas validações / verificações. Estes programas têm como objetivo auxiliar o desenvolvimento de um novo layout, definido pelo cliente.

No desenvolvimento deste projeto, houve a preocupação com relação ao desempenho do projeto, reutilização do código, facilidade a mudanças e facilidades operacionais. O usuário final do software deve ser uma pessoa com conhecimento sobre a área técnica, transferência de arquivos, para conseguir receber e enviar os arquivos para os clientes de forma a não causar problemas, tais como: arquivos incorretos ou arquivos incompletos.

6.2.2 Resultados

Este projeto possui um nível de complexidade média, possui algumas funções (exportação / importação e geração de programas) com nível elevado de complexidade, mas apresenta várias funções (cadastramento) de nível baixo e médio.

Verificando a tabela comparativa (Tabela 6.2.2.1), obtida através das informações apresentadas pela ferramenta de estimativa, entre as métricas, PF-PP (Pontos de Particularidade), PF-IFPUG (Pontos por Função definidos pelo IFPUG), PF-Fixo com nível de complexidade fixo e o tempo real gasto no desenvolvimento do projeto “EDI – Troca de informações de forma Eletrônica”, identificou-se que a métrica PF-PP (Pontos de Particularidade) apresentou um resultado bem próximo do real (3% de erro).

TABELA 6.2.2.1 – Tabela Comparativa das Estimativas - EDI.

Módulo	PF - PP		PF - IFPUG		PF - Fixo		Real
	Pontos	% erro	Pontos	% erro	Pontos	% erro	Pontos
1 - Cadastros	1.380,12	21,98	1.331,76	24,72	1.587,20	10,28	1.769,00
2 - Consulta	0,00		0,00		0,00		0,00
3 - Especial	899,00	(30,67)	1.062,68	(54,46)	987,04	(43,47)	688,00
4 - Listagem	1.068,87	(186,18)	918,84	(146,01)	1.216,43	(225,68)	373,50
5 - Movimentação	1.044,08	26,71	1.173,04	17,65	1.160,63	18,52	1.424,50
Total	4.392,07	(3,22)	4.486,32	(5,44)	4.951,30	(16,36)	4.255,00

A Figura 6.2.2.1 apresenta os dados da Tabela 6.2.2.1 – Tabela Comparativa das Estimativas de forma gráfica. Normalmente, o gerente do projeto prefere visualizar as informações de forma gráfica para identificar com mais rapidez os pontos de discrepância, e caso seja necessário pode recorrer às informações específicas que geraram o gráfico.

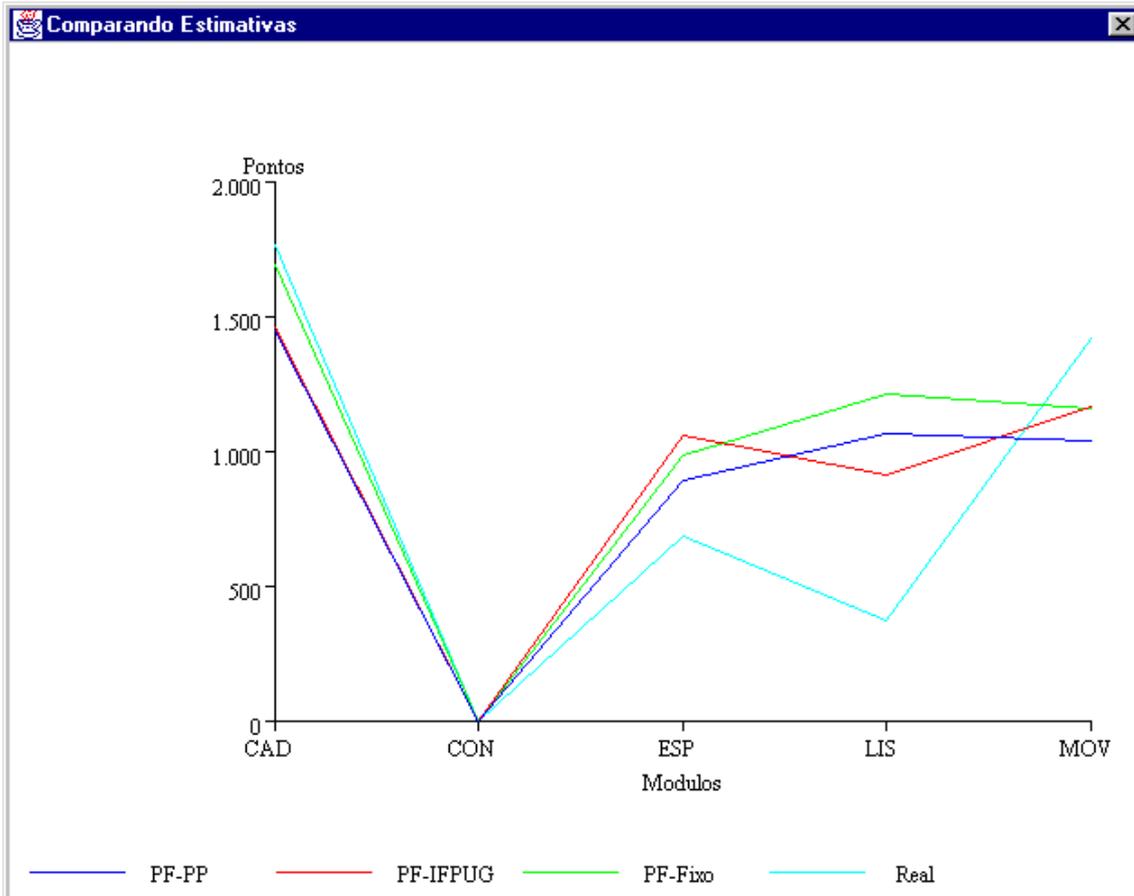


FIGURA 6.2.2.2 – Comparativo entre Estimativas no projeto “EDI”

Analisando a produtividade dos profissionais envolvidos (LA e ECP) verificou-se que o profissional LA ao desenvolver funções com baixo nível de complexidade leva mais tempo do que o estimado, mas para funções com médio ou alto nível de complexidade, desenvolve no tempo estimado.

Enquanto que o profissional ECP foi subestimado para as funções do tipo de listagem, enquanto que para o desenvolvimento das funções de cadastrados atingiu o esperado. Quanto às funções de atualização foi superestimada para as de média complexidade e subestimada para as de alto nível de complexidade. Quando isto ocorre, deve ser verificado se o problema está com o profissional ou com a estimativa do projeto. Analisando ambas as situações verificou-se que o profissional ECP, juntamente no período de desenvolvimento das atualizações de média complexidade também estava adquirindo conhecimento sobre o assunto o que causou uma improdutividade.

As informações sobre os profissionais, pode ser visualizada pelos gráficos apresentados na Figura 6.2.2.3.

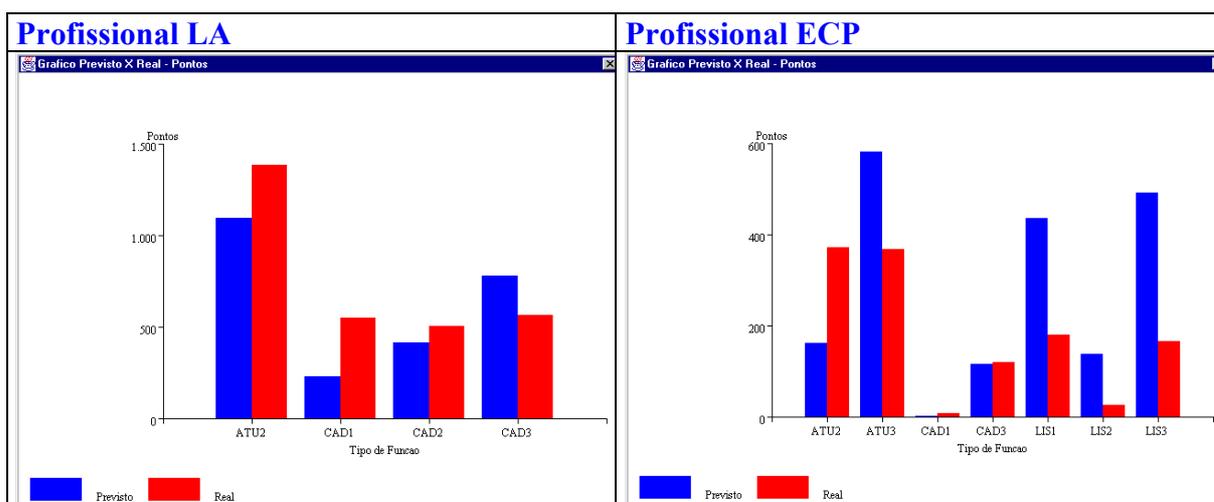


FIGURA 6.2.2.3 – Acompanhamento dos Profissionais no projeto “EDI”

A princípio, o projeto EDI foi planejado para ser desenvolvido quando os profissionais estivessem disponíveis. Por este motivo o projeto prolongou-se durante oito meses. O projeto foi iniciado em novembro/1998, foi colocado de lado no período de janeiro/março de 1999 quando foi novamente reiniciado e concluído em julho de 1999.

Por ter sido um projeto sem prazo definido de entrega, o acompanhamento do projeto ocorreu conforme a alocação dos profissionais. A Figura 6.2.2.4 apresenta o gráfico de Gantt da alocação dos profissionais e com base neste gráfico definiram-se os pontos de controle. Isto significa que os pontos de controle foram sendo definidos durante o desenvolvimento do projeto, isto não é uma boa prática.

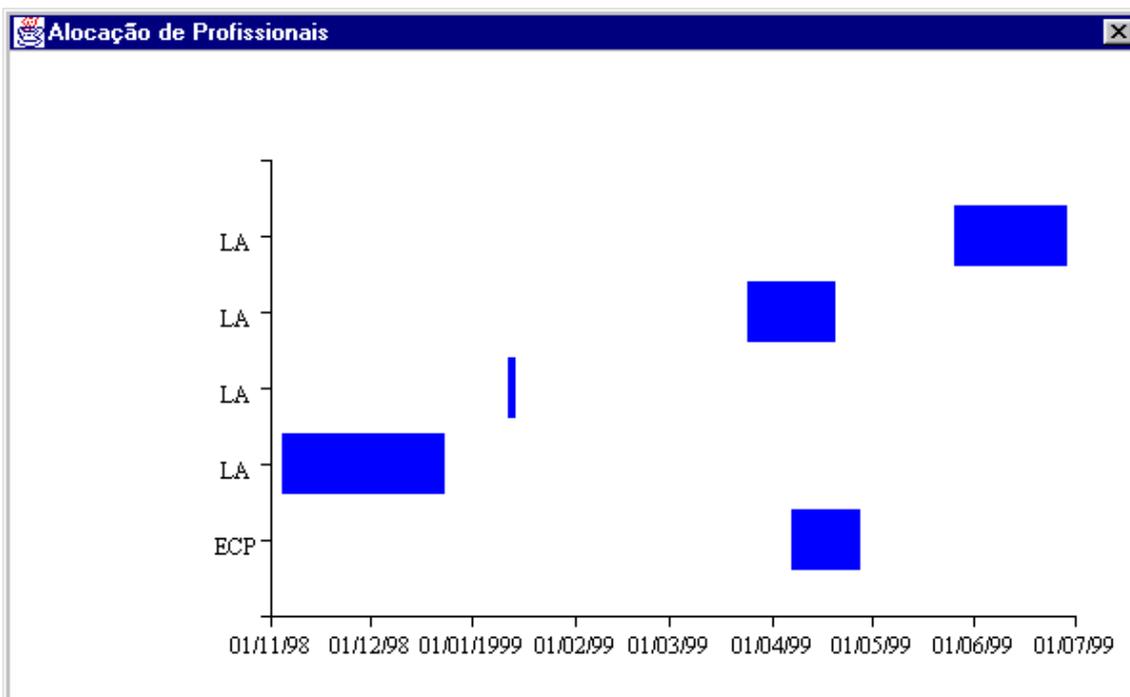


FIGURA 6.2.2.4 – Alocação dos Profissionais no projeto “EDI”

Não se pode determinar que neste projeto ocorreram atrasos ou antecipações com relação ao projeto, porque nenhuma data de término do projeto foi definida

inicialmente. Portanto a Figura 6.2.2.5 apresenta apenas um cenário, um acompanhamento fictício do projeto.

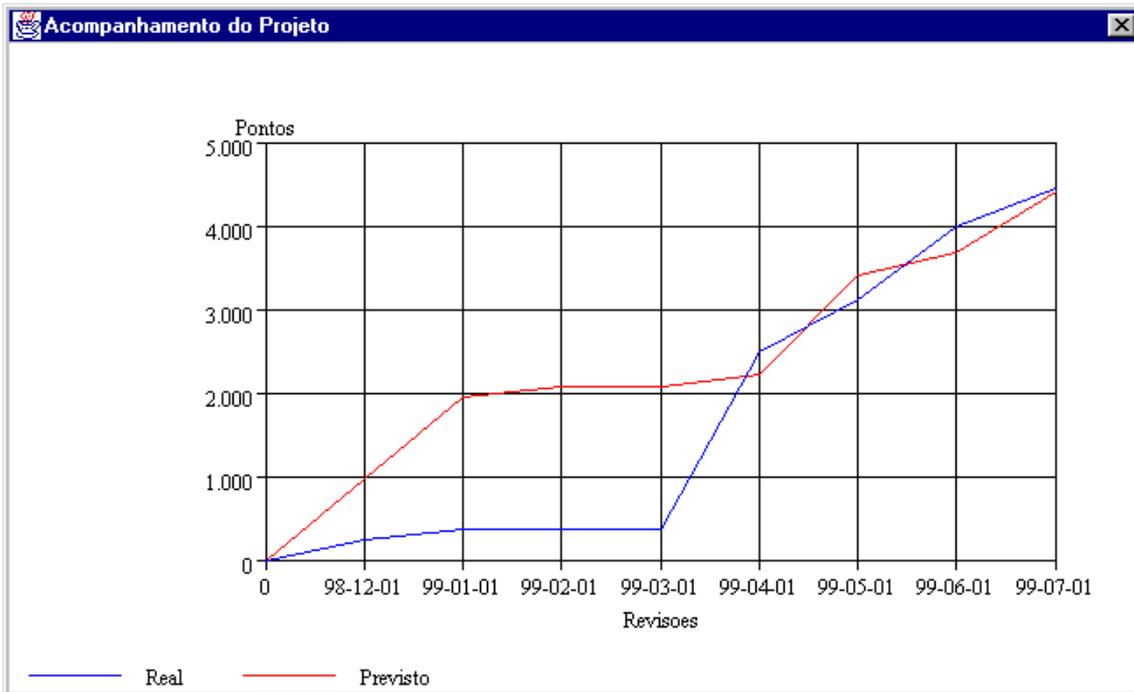


FIGURA 6.2.2.5 – Acompanhamento do Projeto no projeto “EDI”

7 Conclusão

Neste trabalho foi apresentada a importância do gerenciamento de projetos no desenvolvimento de software. O gerenciamento de projetos é dividido em duas partes, Planejamento e Controle das atividades a serem executadas pela equipe de desenvolvimento do projeto.

Uma etapa do processo de Planejamento de projeto é estimar o tamanho, o custo e o esforço no processo de desenvolvimento do sistema de software em questão. Para estimar um projeto deve ser aplicada alguma métrica de estimativa. Exemplos: Estimativa do Esforço, Estimativa de Putnam, Modelo COCOMO, Análise de Pontos por Função, Pontos de Particularidade ou PSP.

Cada métrica possui características próprias. A métrica Estimativa do Esforço determina o esforço e o custo do desenvolvimento de um projeto através do número de profissionais que executaram cada tarefa de desenvolvimento. A métrica de Estimativa de Putnam determina o esforço e o tempo necessário para o desenvolvimento de um projeto através das curvas de Rayleigh. O modelo COCOMO baseia-se nas equações definidas por Barry Boehm para determinar o esforço, o tempo e o número de profissionais necessários para desenvolver o projeto. A métrica de Análise de Pontos por Função baseia-se na funcionalidade do sistema para determinar o tamanho do projeto. Pontos de Particularidade é uma métrica estendida da métrica de Análise de Pontos por Função, pois considera também o cálculo do número de algoritmos a ser desenvolvido. O PSP não pode ser considerado somente uma estimativa para um processo de melhoria profissional pessoal, onde se inclui uma etapa de estimativa.

Todas as métricas aqui apresentadas são baseadas no tamanho do projeto. E o tamanho do projeto pode ser definido pelo número de linhas de código, pontos por função ou pontos por objetos.

Para aplicar uma métrica de estimativa em uma empresa de desenvolvimento de software fez-se necessária uma ferramenta que automatize este processo, caso contrário torna-se inviável.

Ao analisar as ferramentas, ESTIMACS, SLIM, SPQR/20 e ESTIMATE Professional, verificou-se que nenhuma delas atendia totalmente às necessidades do gerente de projeto. Um gerente de projeto deve ser capaz de acompanhar o projeto, a estimativa e os profissionais que desenvolvem o sistema. Por este motivo optou-se em desenvolver uma nova ferramenta para estimativa e controle de projetos de software.

Antes de desenvolver uma ferramenta de estimativa investigou-se um ambiente ideal para que o gerenciamento do projeto (planejamento e controle de projetos). Quanto melhor o ambiente, mais integrado, melhores são as condições de gerenciamento, criando desta forma um ambiente de trabalho ideal. Este ambiente deveria conter ferramentas de desenvolvimento, modelagem dos dados e de estimativa. As informações existentes para cada uma das ferramentas devem ser compartilhadas com as demais, de forma automatizada se possível.

A ferramenta de estimativa deve permitir um acompanhamento do projeto, do profissional e da estimativa. Deve ser possível estimar um projeto através de várias técnicas de estimativas, permitindo um comparativo entre as estimativas obtidas, verificando desta forma a melhor métrica de estimativa para determinado projeto.

Dois projetos de software foram submetidos à ferramenta de estimativa desenvolvida. Dois projetos é pouco, para que conclusões possam ser tiradas, mas pode se observar que a métrica de estimativa de Pontos Fixos (considerando o projeto como um todo) é mais recomendada para projetos onde exista uma complexidade algorítmica bem elevada. E que a métrica de Pontos de Particularidade é a melhor métrica aplicada em projeto com complexidade algorítmica média, sem muitas fórmulas matemáticas. Deve-se considerar também que ao aplicar uma métrica de estimativa sobre um projeto, algumas funções podem apresentar variações de erro consideráveis, que ao final se complementam.

Conclui-se que para ter um ambiente de desenvolvimento com que gere produtos de alta qualidade e produtividade é necessário um bom gerenciamento - planejamento e controle do processo de desenvolvimento. O gerente de projeto deve ter condições de obter informações, de forma automática, sobre o andamento do projeto e a produtividade da equipe de desenvolvimento, identificando os problemas, para que ações corretivas possam ser tomadas. No entanto o gerente de projeto ainda deve valer-se também da sua experiência, pois cada projeto é um projeto, com características semelhantes e características diferenciadas de outros projetos.

A ferramenta de estimativa desenvolvida ainda pode ser aperfeiçoada e melhorada. O processo de RBC pode ser implementado, como também novos projetos devem ser introduzidos para que conclusões mais realistas possam ser retiradas.

A ferramenta de estimativa hoje implementa somente métricas de estimativa baseadas em pontos por função ou pontos por objeto, mas futuramente podem ser implementadas métricas baseadas em linhas de código. Deve-se tomar cuidado quando se define uma métrica baseada em linhas de código, porque a contagem de linha deve ser muito bem especificada - quais tipos de linhas de código devem ser contadas (comentários, linhas em branco).

As empresas desenvolvedoras de software devem tomar consciência que para melhorar a qualidade e a produtividade do processo de desenvolvimento é necessário mensurar este processo. A partir do conhecimento das medidas, é possível definir metas para aumentar a qualidade e produtividade do desenvolvimento. E com o conhecimento sobre a produtividade da equipe de desenvolvimento é possível estabelecer prazos e custos mais próximos da realidade.

Anexo 1 Tabela de Decomposição

A Tabela de Decomposição pode ser utilizada para ajudar a estimativa a encontrar o número de linhas de código de um projeto. Esta técnica determina o tamanho do software, pode ser utilizada em conjunto com as métricas COCOMO e Putnam determinando o número total de linhas de código do projeto. No entanto pode estimar o custo e o esforço do projeto, desde que se aplique a taxa de mão-de-obra e a média produzida pelos profissionais.

Para utilizar-se a tabela de decomposição é necessário definir o número de Linhas de Código ou Pontos por Função ou Pontos de Particularidade, por uma visão otimista, pessimista e a mais provável. A partir destas informações consegue-se cálculo esperado a partir da fórmula:

$$\text{Esperado} = (\text{otimista} + (4 * \text{mais provável}) + \text{pessimista}) / 6$$

Deve ser informados, também, o valor do ponto ou da linha e a quantidade de linhas ou pontos produzidos por mês, para calcular o custo e o tempo, segundo as fórmulas abaixo.

$$\text{Custo} = \text{Esperado} * \$\text{Linha}$$

$$\text{Meses} = \text{Esperado} / \text{Linha-mês}$$

Desta forma consegue-se construir a seguinte tabela:

TABELA A.1.1 - Decomposição

Função	Otimista	Mais Provável	Pessimista	Esperado	\$ Linha	Linha-Mês	Custo	Meses
Funções do projeto definidas no escopo.								
Total								

Anexo 2 Tabela de Multiplicadores (COCOMO)

TABELA A.2.1 - Multiplicadores do COCOMO 81

Cód	Descrição	Muito Baixo	Baixo	Normal	Alto	Muito Alto	Extra Alto
01	Atributos do Produto						
01	Confiabilidade desejada	0,75	0,88	1,00	1,15	1,40	-
02	Tamanho do banco de dados	-	0,94	1,00	1,08	1,16	-
03	Complexidade do produto	0,70	0,85	1,00	1,15	1,30	1,65
02	Atributos do Ambiente						
01	Limitação de tempo de execução	-	-	1,00	1,11	1,30	1,66
02	Limitação de memória	-	-	1,00	1,06	1,21	1,56
03	Volatibilidade do equipamento	-	0,87	1,00	1,15	1,30	-
04	Tempo de "Turnaround" do computador	-	0,87	1,00	1,07	1,15	-
03	Atributos de Pessoal						
01	Capacidade dos analistas	1,46	1,19	1,00	0,86	0,71	-
02	Capacidade dos programadores	1,42	1,17	1,00	0,86	0,70	-
03	Experiência na aplicação	1,29	1,13	1,00	0,91	0,82	-
04	Experiência com o sistema operacional	1,21	1,10	1,00	0,9	-	-
05	Experiência na linguagem de programação	1,14	1,07	1,00	0,95	-	-
04	Atributos do Projeto						
01	Uso de práticas modernas de programação	1,24	1,10	1,00	0,91	0,82	-
02	Uso de ferramentas de desenvolvimento	1,24	1,10	1,00	0,91	0,83	-
03	Tempo disponível para o projeto	1,23	1,08	1,00	1,04	1,10	-

TABELA A.2.2 - Multiplicadores da extensão do COCOMO

Cód		Muito Baixo	Baixo	Normal	Alto	Muito Alto	Extra Alto
01	Escala de Drives						
01	Precedência	6,20	4,96	3,72	2,48	1,24	0,00
02	Flexibilidade de desenvolvimento	5,07	4,05	3,04	2,03	1,01	0,00
03	Arquitetura / Resolução de risco	7,07	5,65	4,24	2,83	1,41	0,00
04	Coesão da equipe	5,48	4,38	3,29	2,19	1,10	0,00
05	Maturidade do processo	7,80	6,24	4,68	3,12	1,56	0,00
02	Atributos do Produto						
01	Confiabilidade desejada	0,82	0,92	1,00	1,10	1,26	-
02	Tamanho do banco de dados	-	0,90	1,00	1,14	1,28	-
03	Complexidade do produto	0,73	0,87	1,00	1,17	1,34	1,74
04	Documentação	0,81	0,91	1,00	1,11	1,23	-
03	Atributos do Ambiente						
01	Limitação de tempo de execução	-	-	1,00	1,11	1,29	1,63
02	Limitação de memória	-	-	1,00	1,05	1,17	1,46
03	Volatilidade do equipamento	-	0,87	1,00	1,15	1,30	-
04	Atributos de Pessoal						
01	Capacidade dos analistas	1,42	1,19	1,00	0,85	0,71	-
02	Capacidade dos programadores	1,34	1,15	1,00	0,88	0,76	-
03	Continuidade do Pessoal	1,20	1,09	1,00	0,91	0,84	-
04	Experiência dos analistas	1,22	1,10	1,00	0,88	0,81	-
05	Experiência dos programadores	1,19	1,09	1,00	0,91	0,85	-
06	Experiência na linguagem de programação	1,29	1,12	1,00	0,90	0,81	-
05	Atributos do Projeto						
01	Uso de práticas modernas de programação	1,17	1,09	1,00	0,90	0,78	-
02	Uso de ferramentas de desenvolvimento	1,22	1,09	1,00	0,93	0,86	0,8

Anexo 3 Tabelas dos Pontos por Função

A.3.1 Arquivos Lógicos Internos

Os Arquivos Lógicos Internos representam os requerimentos de armazenamento de dados, cuja manutenção é feita pela própria aplicação.

TABELA A.3.1.1 - Complexidade dos Arquivos Lógicos Internos

	1 a 19 itens de dados referenciados	20 a 50 itens de dados referenciados	51 ou mais itens de dados referenciados
1 registro lógico	SIMPLES	SIMPLES	MÉDIA
2 a 5 registros lógicos	SIMPLES	MÉDIA	COMPLEXA
6 ou mais registros lógicos	MÉDIA	COMPLEXA	COMPLEXA

A.3.2 Arquivo Interface Externas

Os Arquivos de Interface Externos representam as necessidades dos dados, originadas externamente à aplicação.

TABELA A.3.2.1 - Complexidade das Interfaces Externas

	1 a 19 itens de dados referenciados	20 a 50 itens de dados referenciados	51 ou mais itens de dados referenciados
1 registro lógico	SIMPLES	SIMPLES	MÉDIA
2 a 5 registros lógicos	SIMPLES	MÉDIA	COMPLEXA
6 ou mais registros lógicos	MÉDIA	COMPLEXA	COMPLEXA

A.3.3 Entradas Externas

As Entradas Externas representam as atividades de manutenção de dados

TABELA A.3.3.1 - Complexidade das Entradas Externas

	1 a 4 itens de dados referenciados	5 a 15 itens de dados referenciados	16 ou mais itens de dados referenciados
1 arquivo referenciado	SIMPLES	SIMPLES	MÉDIA
2 arquivos referenciados	SIMPLES	MÉDIA	COMPLEXA
3 ou mais arquivos referenciados	MÉDIA	COMPLEXA	COMPLEXA

A.3.4 Saídas Externas

As Saídas Externas representam as atividades da aplicação que tem como resultado a saída dos dados.

TABELA A.3.4.1 - Complexidade das Saídas Externas

	1 a 5 itens de dados referenciados	6 a 19 itens de dados referenciados	20 ou mais itens de dados referenciados
1 arquivo referenciado	SIMPLES	SIMPLES	MÉDIA
2 a 3 arquivos referenciados	SIMPLES	MÉDIA	COMPLEXA
4 ou mais arquivos referenciados	MÉDIA	COMPLEXA	COMPLEXA

A.3.5 Consultas Externas

As Consultas Externas são uma combinação de Entradas / Saídas de dados, onde uma entrada ocasiona a recuperação imediata de dados.

Para classificar uma Consulta Externa:

- ✓ Calcula-se a complexidade funcional da parte de entrada das consultas externas
- ✓ Calcula-se a complexidade funcional da parte de saída das consultas externas
- ✓ Escolhe a maior complexidade encontrada

TABELA A.3.5.1 - Complexidade das Consultas – Parte de Entrada

	1 a 4 itens de dados referenciados	5 a 15 itens de dados referenciados	16 ou mais itens de dados referenciados
1 arquivo referenciado	SIMPLES	SIMPLES	MÉDIA
2 arquivos referenciados	SIMPLES	MÉDIA	COMPLEXA
3 ou mais arquivos referenciados	MÉDIA	COMPLEXA	COMPLEXA

TABELA A.3.5.2 - Complexidade das Consultas – Parte de Saída

	1 a 5 itens de dados referenciados	6 a 19 itens de dados referenciados	20 ou mais itens de dados referenciados
1 arquivo referenciado	SIMPLES	SIMPLES	MÉDIA
2 a 3 arquivos referenciados	SIMPLES	MÉDIA	COMPLEXA
4 ou mais arquivos referenciados	MÉDIA	COMPLEXA	COMPLEXA

A.3.6 Tabela de Cálculo dos Pontos por Função

TABELA A.3.6.1 - Cálculo dos Pontos por Função

Tipo de Função	Complexidade e Funcionalidade	Total de Pontos	Pontos X Complexidade
Arquivo	Simples X 7 = Média X 10 = Complexa X 15 =		
Interface	Simples X 7 = Média X 10 = Complexa X 15 =		
Entrada	Simples X 3 = Média X 4 = Complexa X 6 =		
Saída	Simples X 4 = Média X 5 = Complexa X 7 =		
Consulta	Simples X 3 = Média X 4 = Complexa X 6 =		
Total de Pontos por Função Brutos			

A.3.7 Características Gerais dos Sistemas

Atribui-se um peso de 0 a 5 para cada uma destas características, de acordo com o seu nível na aplicação:

- 0 – Nenhuma influência
- 1 – Influência Mínima
- 2 – Influência Moderada
- 3 – Influência Média
- 4 – Influência Significativa
- 5 – Grande Influência

TABELA A.3.7.1 - Características dos Pontos por Função

Características
01 – Comunicação de Dados
02 – Processamento Distribuído
03 – Performance
04 – Utilização do Equipamento
05 – Volume de Transações
06 – Entrada de Dados “on-line”
07 – Eficiência do Usuário Final
08 – Atualização “on-line”
09 – Processamento Complexo
10 – Reutilização de Código
11 – Facilidade de Implantação
12 – Facilidade Operacional
13 – Múltiplos Locais
14 – Facilidade de Mudanças

Anexo 4 Tabelas do PSP

Ao aplicar o PSP, a companhia deve seguir os seguintes tópicos:

TABELA 4.1 – Como aplicar o PSP

Tópico	Nível do PSP	Descrição
A estratégia do PSP e a sua Linha Básica	PSP0	Calcule a média e o desvio padrão de números reais guardados em uma lista vinculada
O Processo de Planejamento	PSP0.1	Conte o LOC de um programa Produza o padrão de contagem LOC Produza o padrão de código
Medindo o Tamanho do Software	PSP0.1	Aumente o programa para contar o LOC do objeto ou o LOC da função / procedimento Reporte análise de defeitos
Estimando o Tamanho do Software	PSP1	Calcule os parâmetros de regressão linear de pares de número reais guardados em uma lista vinculada
Estimativa de recursos e cronograma	PSP1.1	Integração numérica usando a regra de Simpson.
Medições no PSP	PSP1.1	Aumente o programa para calcular um intervalo de previsão de 90% e 70%
Revisão do Código e do Design		Relatório intermediário de análise do processo
Gerenciamento da Qualidade do Software	PSP2	Calcule a correção de pares de números reais guardados em uma lista vinculada
Design do Software	PSP2 ou PSP2.1	Organize uma lista vinculada
Verificação do Design do Software	PSP2.1	Teste o Qui-quadrado para normalidade
Escalonando o PSP	PSP3	Calcule os parâmetros de regressão linear múltipla para conjuntos de quatro números reais guardados em uma lista vinculada
Definindo o Processo de Software e usando o PSP		Relatório final de análise do processo.

Anexo 5 Diagrama de Atividades

O Diagrama de Atividade representa como a ferramenta de estimativa de custo e esforço está desenvolvida. O processo de estimar um projeto está subdividido em Parametrizar a ferramenta, Cadastrar as Informações referentes à empresa, Cadastrar os projetos, Estimar os projetos, e Gerenciar os projetos. Esta divisão pode ser visualizada no Anexo A.5.1 – Visão Geral.

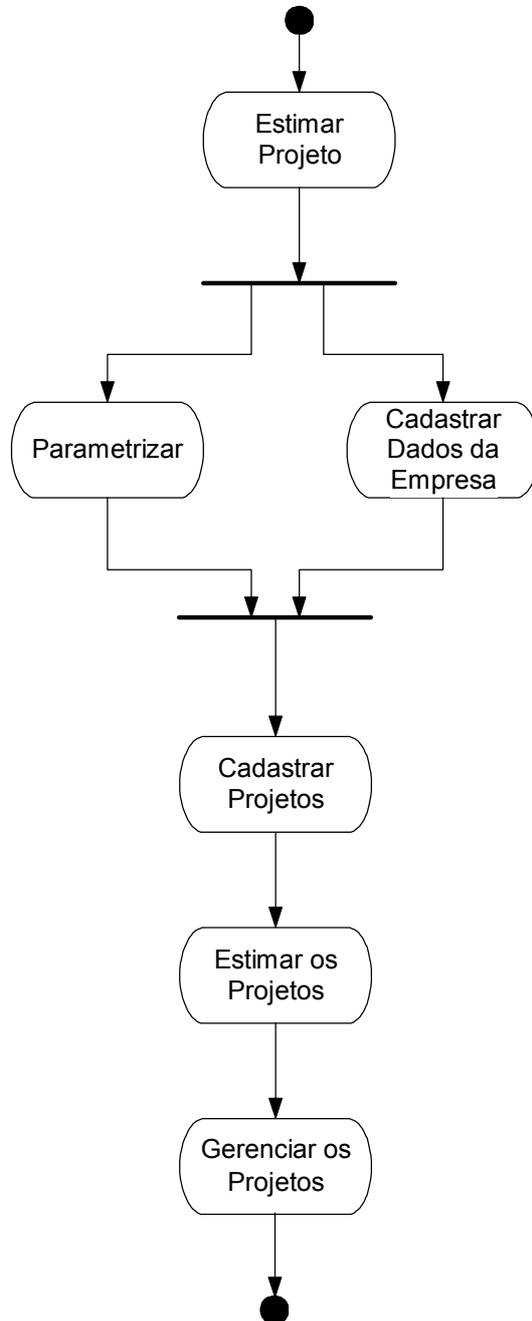
O processo de Parametrizar a ferramenta é composto por definir as métricas de estimativa que a ferramenta implementa e os tipos de funções. O Cadastramento das Informações referentes à empresa são subdivididos em Cadastrar os profissionais que desenvolvem os projetos e os serviços prestados pela empresa desenvolvedora. Ao cadastrar um profissional as informações referentes a férias e históricos salariais também devem ser informadas. Anexo A.5.2 – Cadastros e Parâmetros.

O processo de Cadastrar Projetos engloba um conjunto de atividades, tais como: Cadastrar Cliente, Cadastrar Projeto, Cadastrar Módulos, Cadastrar Funções, Alocar Profissionais, Definir Pontos de Controle, Cadastrar Banco de Dados, Cadastrar as Tabelas dos bancos de dados, Relacionar as Tabelas com as Funções do Projeto e Apontar o número de horas trabalhadas. Anexo A.5.3 – Projetos.

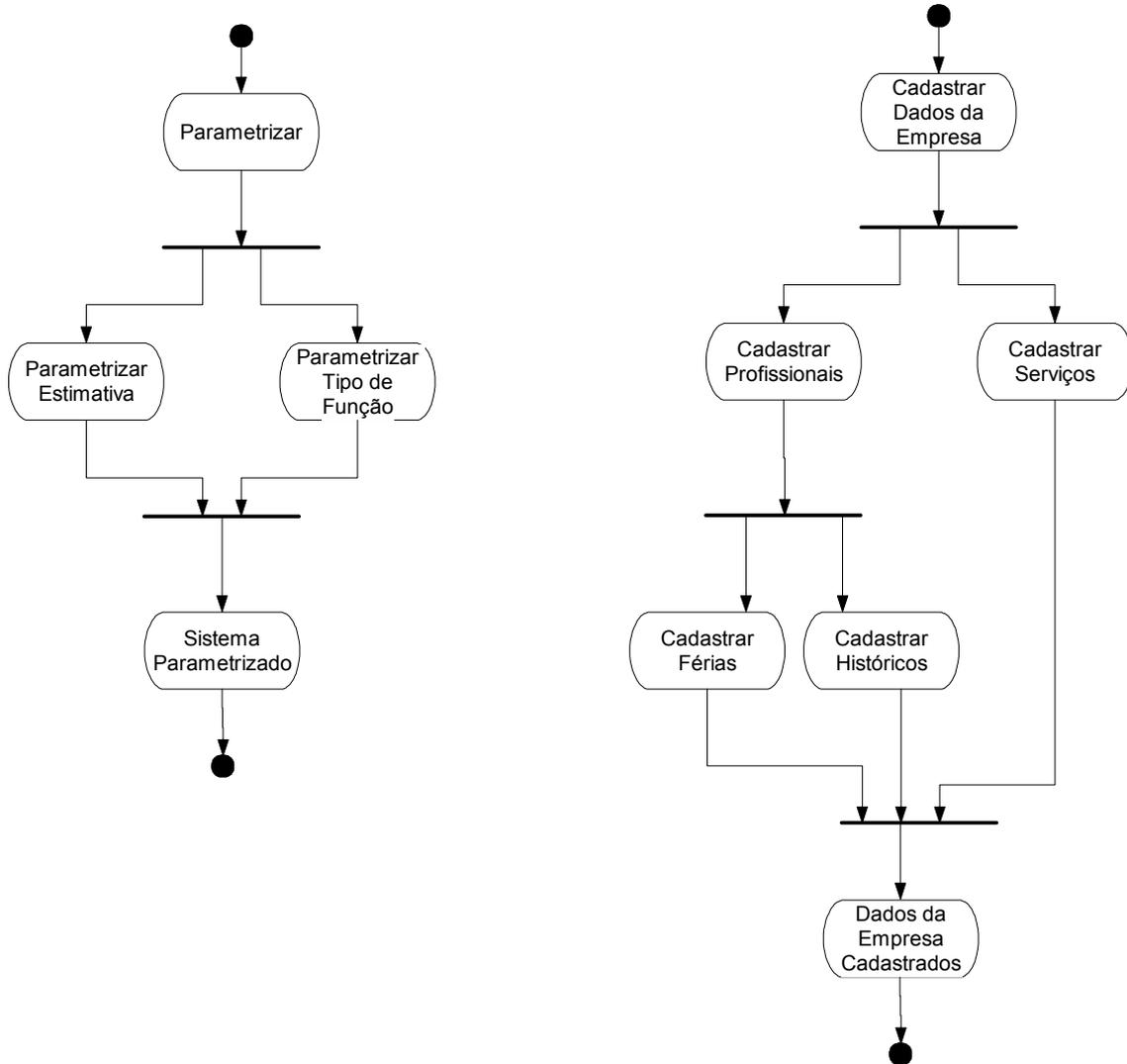
Estimar um Projeto significa estimar os seus módulos e conseqüentemente estimar as funções, a ferramenta desenvolvida permite também uma simulação da função do projeto, Anexo A.5.4 – Estimar Projetos. O Anexo A.5.4.1 detalha como a ferramenta de estimativa calcula a função do projeto.

O processo de Acompanhamento foi subdividido em Gerenciar Projetos, Gerenciar Profissionais e Gerenciar Estimativas, para facilitar o entendimento das informações apresentadas pela ferramenta desenvolvida. Anexo A.5.5 – Acompanhamento.

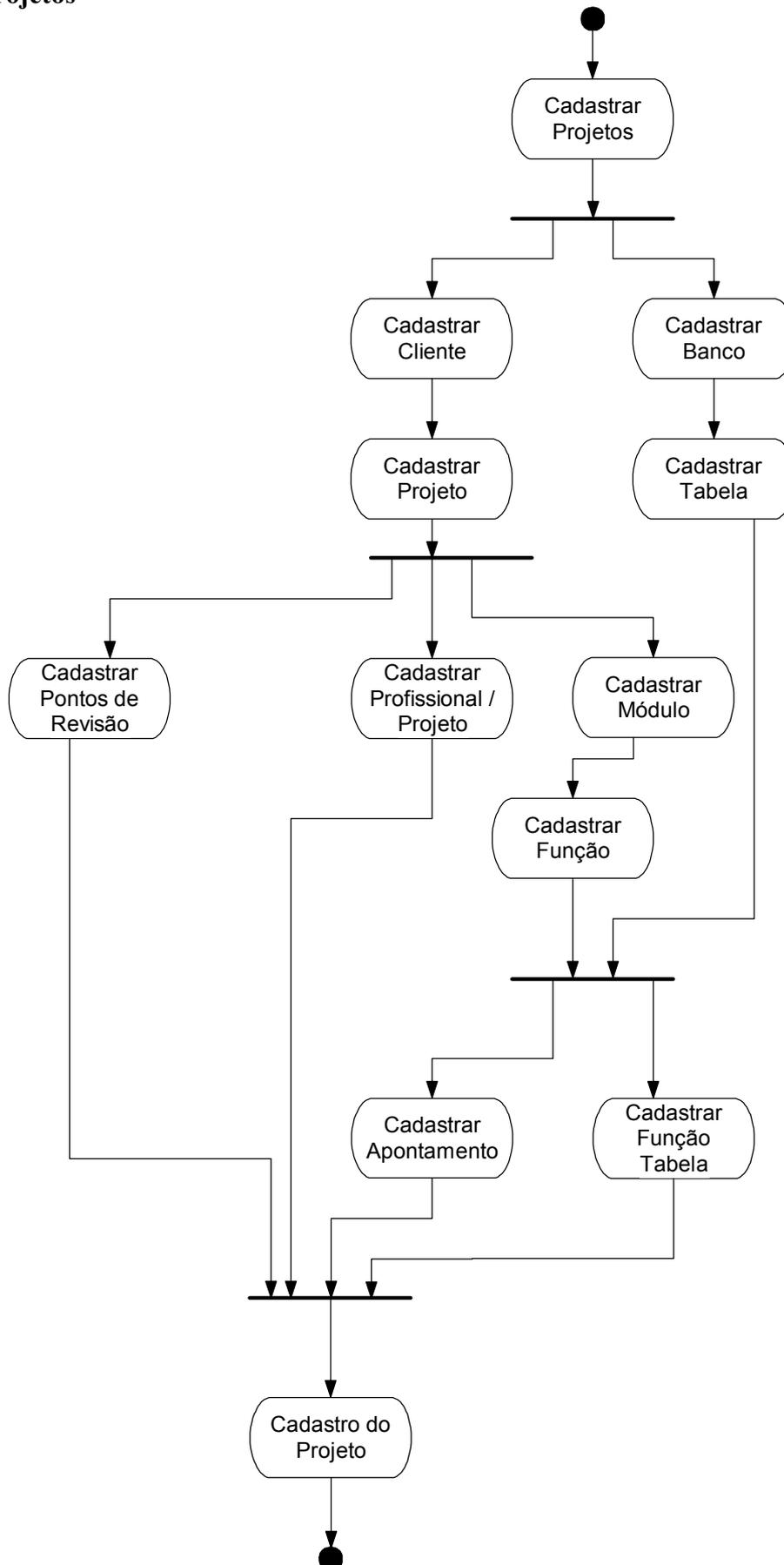
A.5.1 Visão Geral



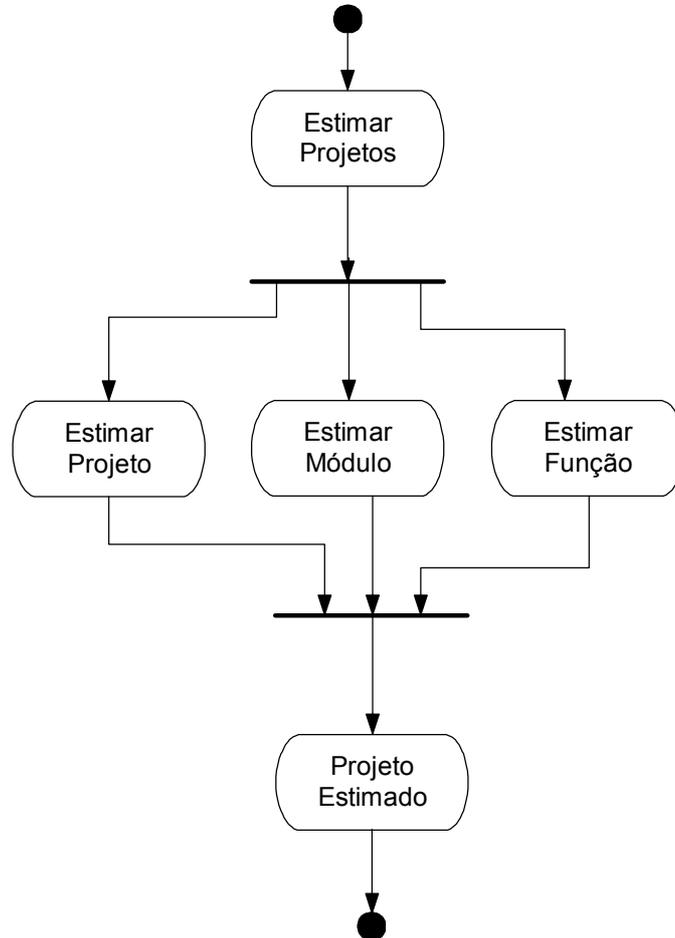
A.5.2 Parâmetros e Cadastros

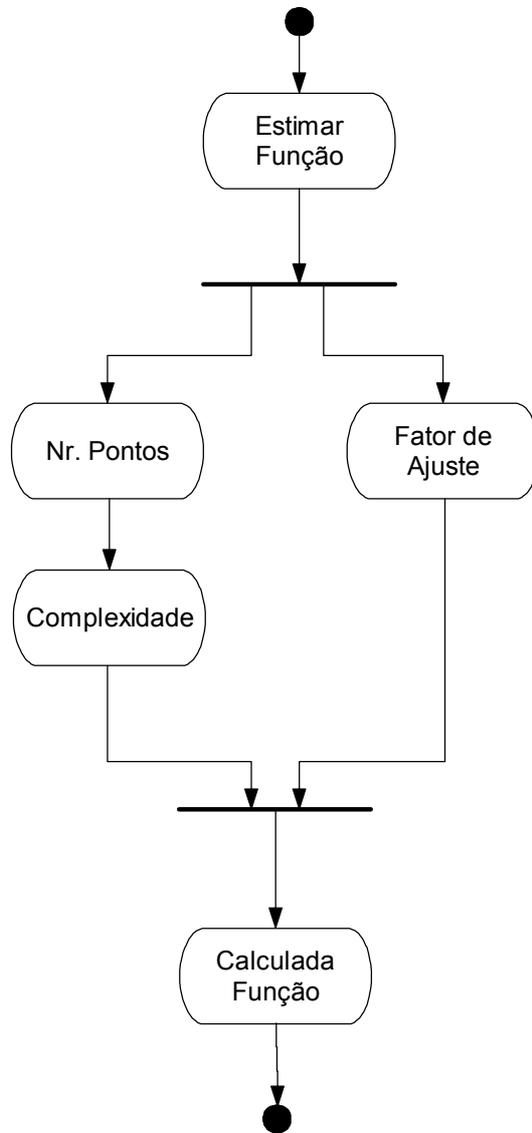


A.5.3 Projetos

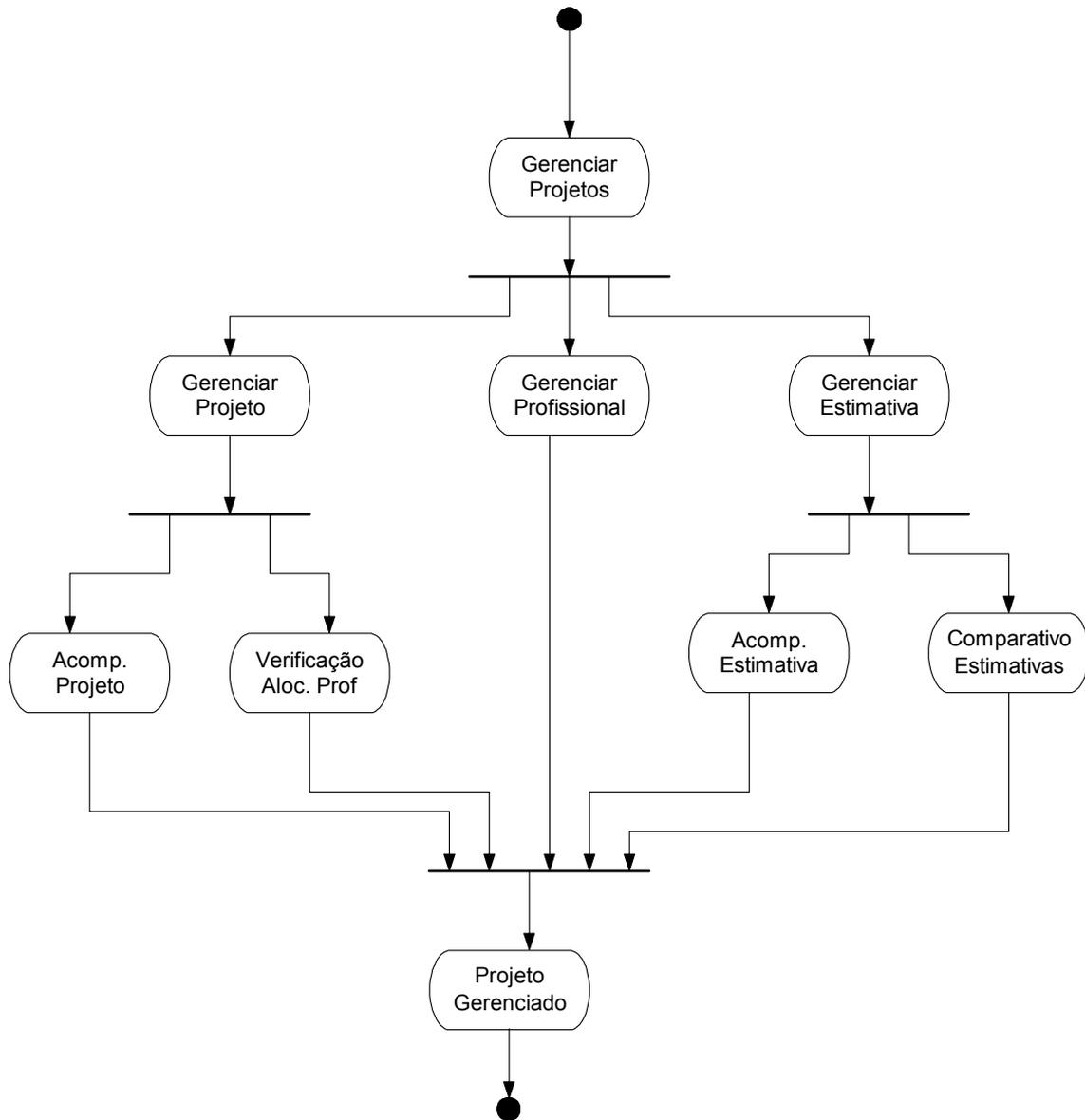


A.5.4 Estimar Projetos



A.5.4.1 Estimar Função

A.5.5 Acompanhamento

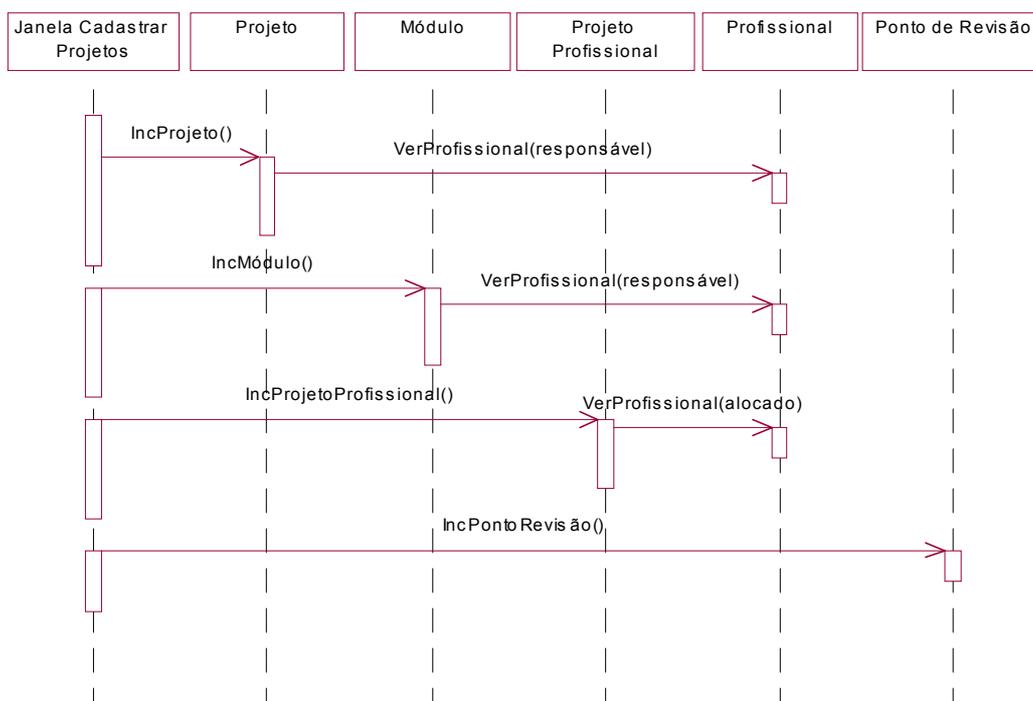


Anexo 6 Diagramas de Interação

Para facilitar o entendimento do processo de estimativa de custo e esforço de um projeto optou-se em detalhar, através do uso de diagramas de interações, os métodos mais complexos existentes nas classes da ferramenta de estimativa desenvolvida.

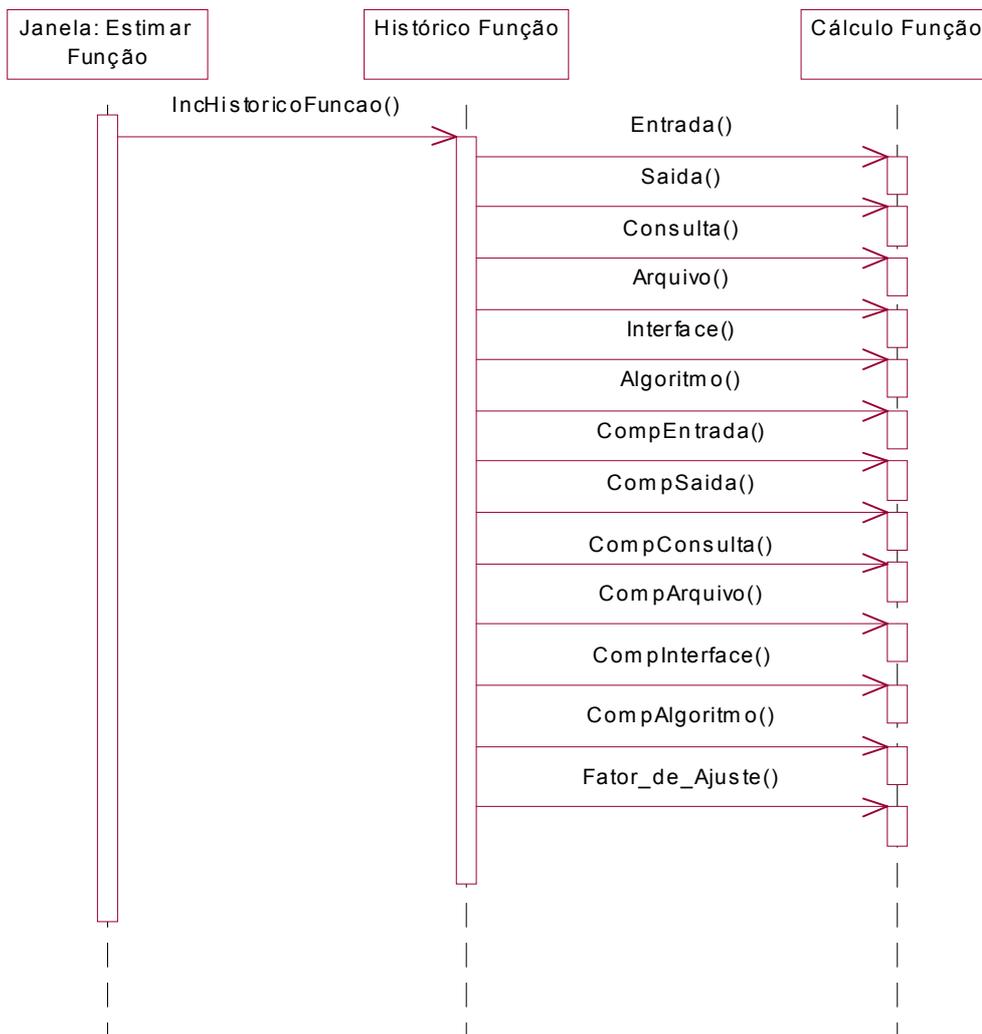
A.6.1 Cadastrar Projetos

Definir como um projeto deve ser incluído na ferramenta de estimativa desenvolvida.

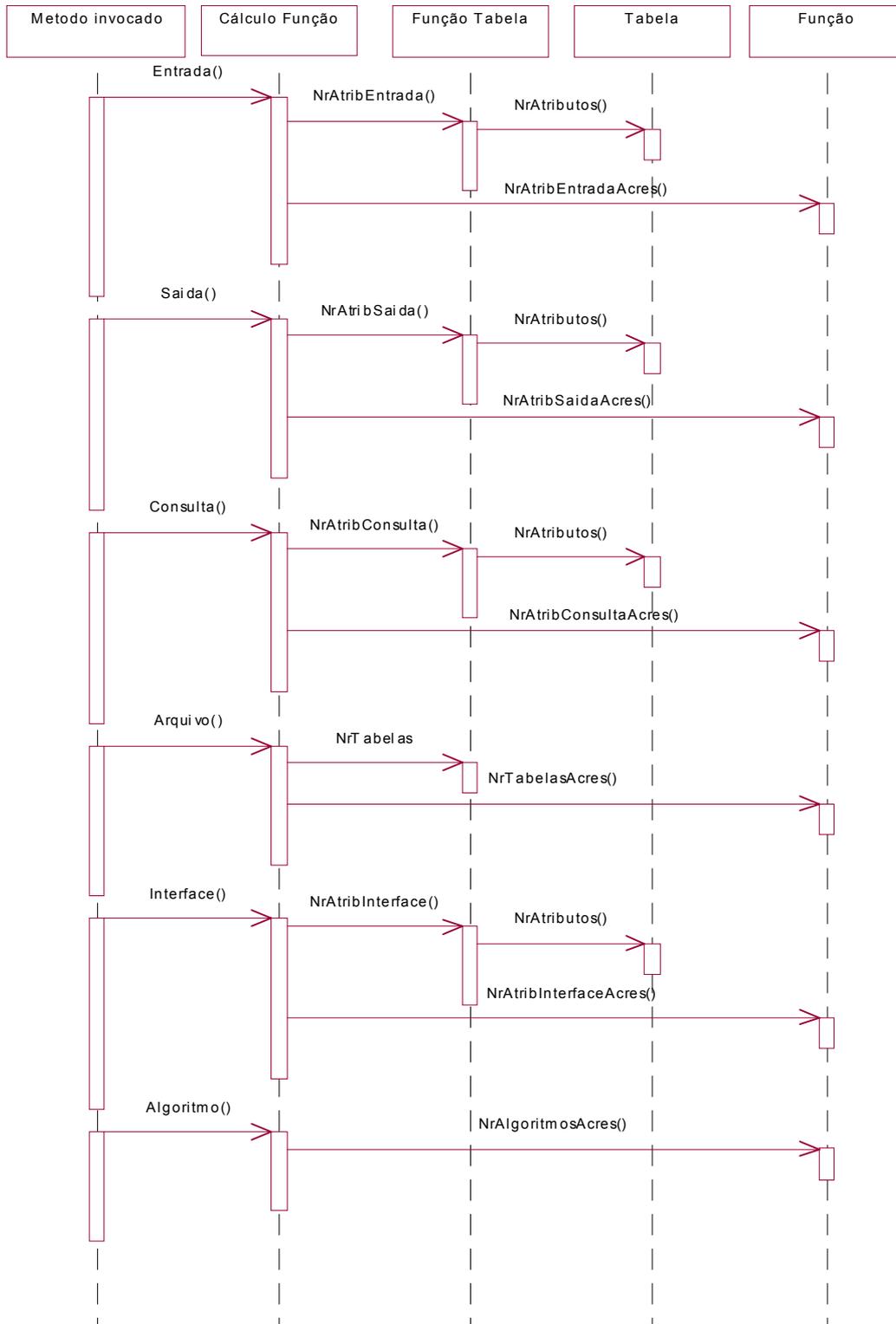


A.6.2 Estimar uma Função

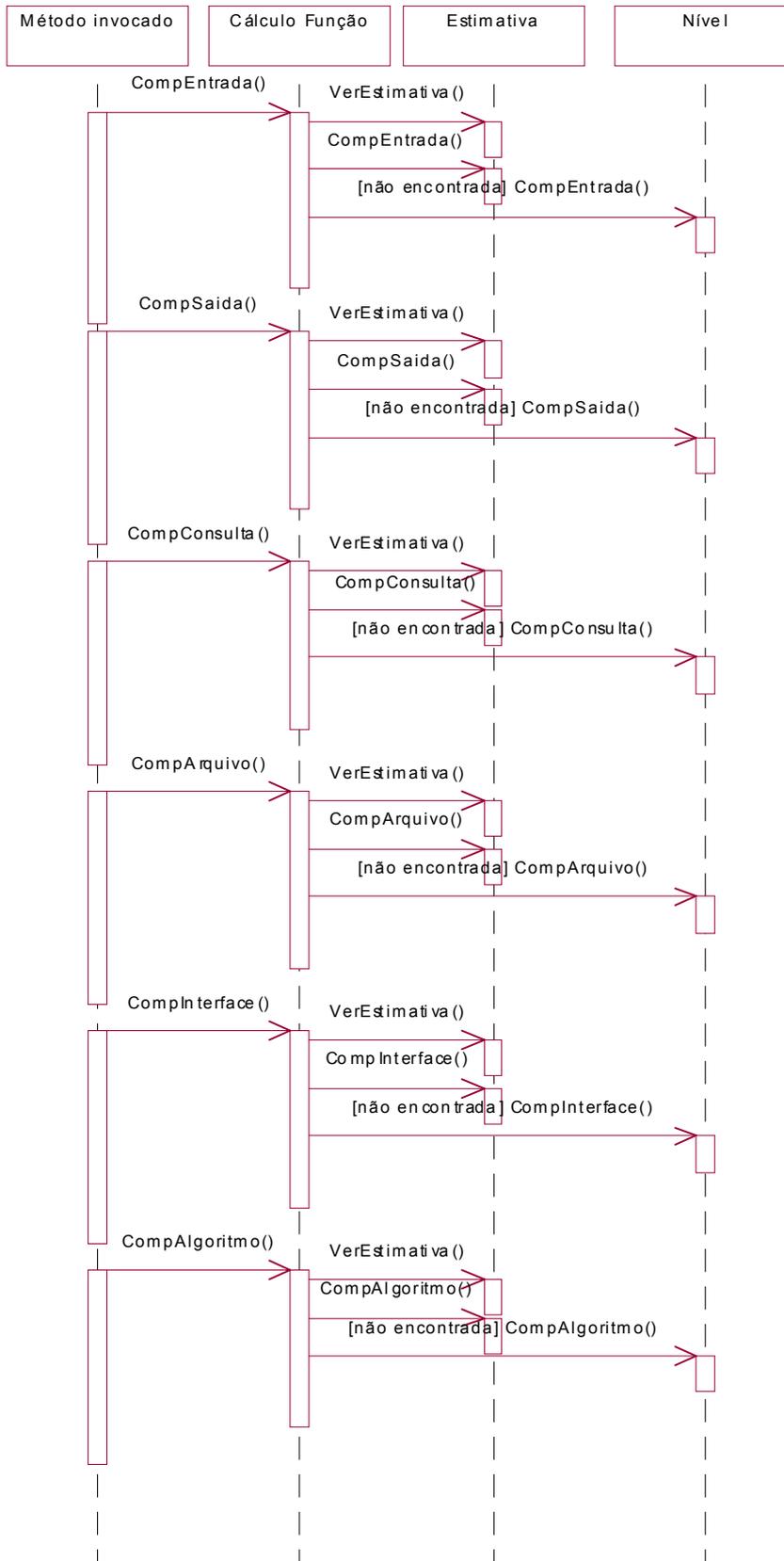
Define como uma função deve ser estimada. Os métodos Entrada, Saída, Consulta, Arquivo, Interface e Algoritmos foram detalhados no Anexo A.6.2.1. Os métodos CompEntrada (complexidade de entrada), CompSaída (complexidade de saída), CompConsulta (complexidade de consulta), CompArquivo (complexidade de arquivo), CompInterface (complexidade de Interface), CompAlgoritmo (complexidade do algoritmo) foram detalhados no Anexo A.6.2.2. E o anexo A.6.2.3 detalha o método para cálculo do fator de ajuste.



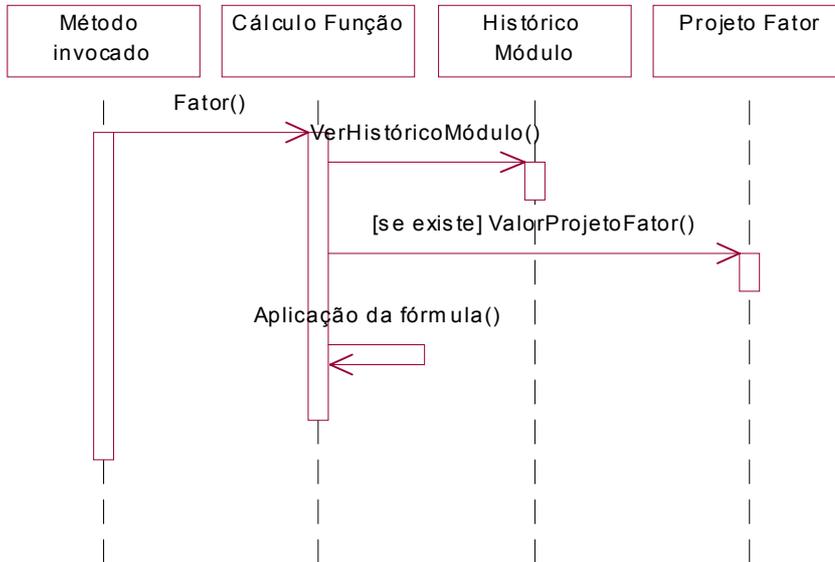
A.6.2.1 Calcular o Número de Pontos



A.6.2.2 Calcular os Níveis de Complexidade

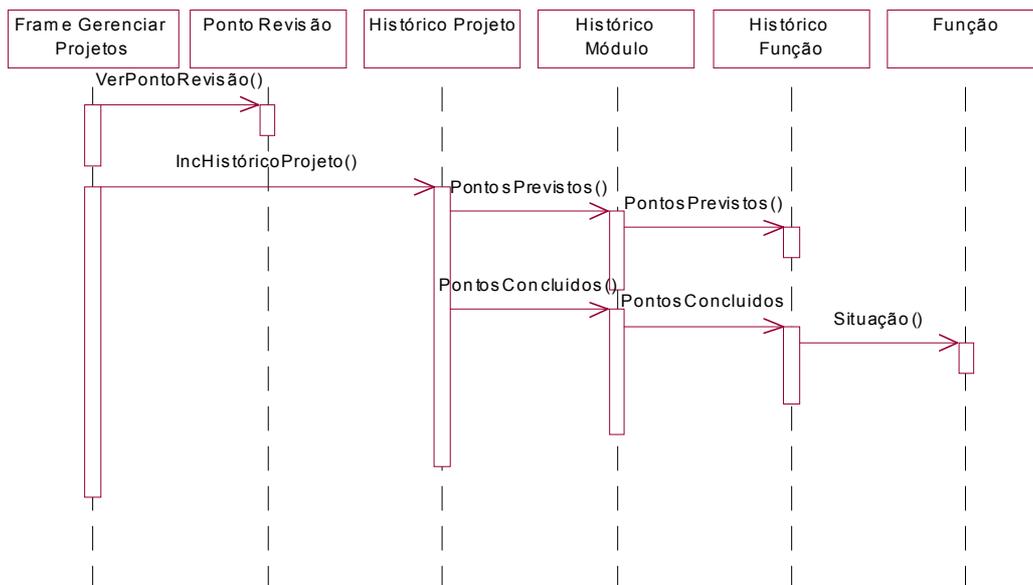


A.6.2.3 Calcular o Fator de Ajuste



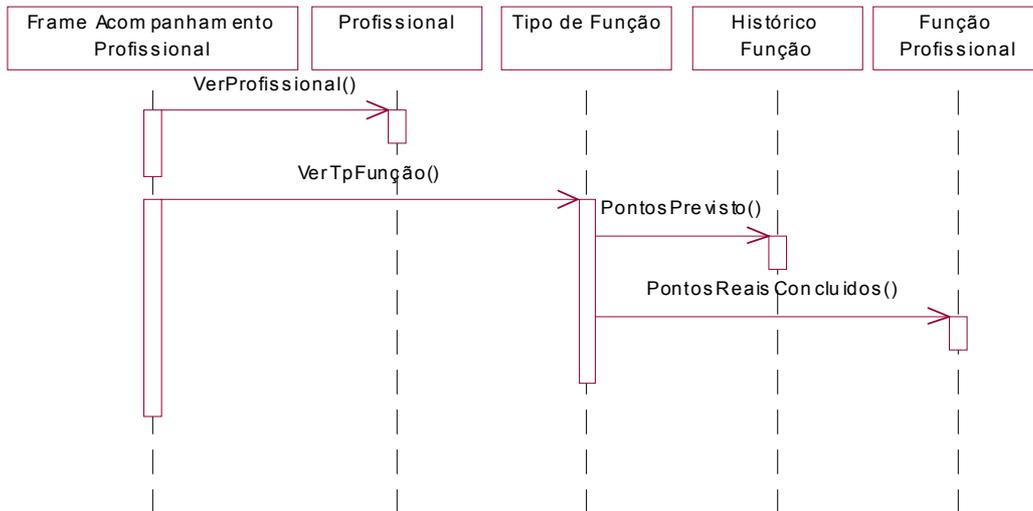
A.6.3 Acompanhamento de Projetos

O processo de acompanhamento do projeto ocorre através dos pontos de controle determinados na definição do projeto. O método de acompanhamento de projetos está baseado na revisão do projeto. A revisão do projeto significa a verificação dos pontos já concluídos, em relação aos pontos estimados.



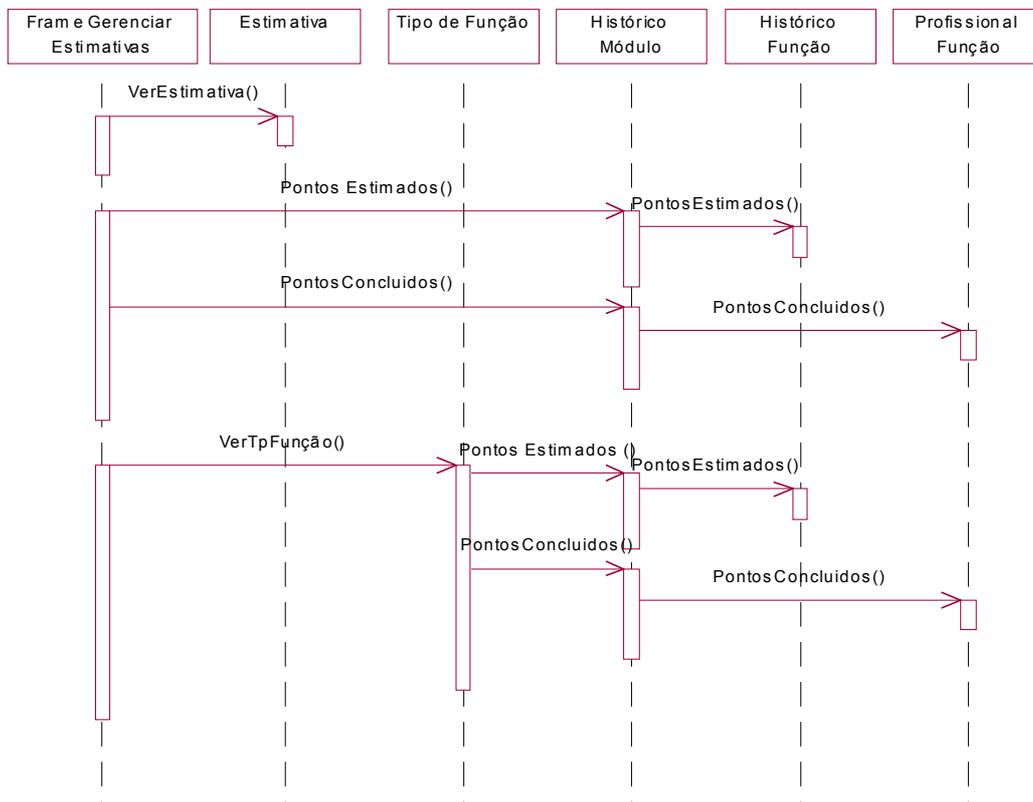
A.6.4 Acompanhamento de Profissionais

Acompanhamento do profissional é feito através de um confronto das informações estimadas e concluídas. O diagrama a baixo descreve o método de acompanhamento do profissional por tipo de função desenvolvida.



A.6.5 Acompanhamento das Estimativas

O Acompanhamento da estimativa confronta as informações estimadas e reais sobre o projeto, este acompanhamento pode ser efetuado por módulo ou por tipo de função.



Anexo 7 Diagrama de Estados

A Classe Função possui vários estados, e cada estado com características próprias:

Criada: é uma função nova na ferramenta.

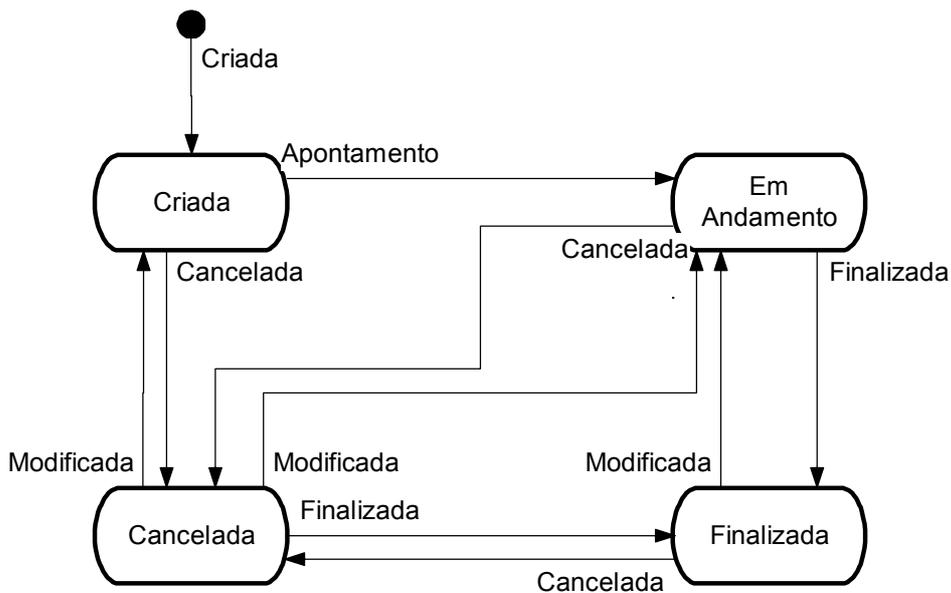
Em Andamento: a partir do momento que foi efetuado um apontamento de horas sobre a função, significa que já foi iniciada.

Cancelada: quando uma função não tiver mais objetivo para o projeto.

Finalizada: quando à função já foi desenvolvida totalmente.

O diagrama de estados permite uma visualização da transferência de estados da Função do Projeto.

A.7.1 Função do Projeto



Anexo 8 Dicionário de Dados

Classe: Apontamento

Atributo	Descrição	Tipo
# Cd_profissional	Código do Profissional	Texto
# Dt_apontamento	Data do Apontamento	Data
# Nr_seqüência	Número de Seqüência	Número
Hr_inicio	Hora de Início	Número
Hr_fim	Hora Final	Número
Cd_cliente	Código do Cliente	Texto
Cd_projeto	Código do Projeto	Texto
Cd_módulo	Código do Módulo	Texto
Cd_função	Código da Função	Texto
Cd_serviço	Código do Serviço	Texto

Classe: Banco de Dados

Atributo	Descrição	Tipo
# Cd_banco	Código do Banco	Texto
Ds_banco	Descrição do Banco	Texto

Classe: Campos da Tabela

Atributo	Descrição	Tipo
# Cd_banco	Código do Banco	Texto
# Cd_tabela	Código da Tabela	Texto
# Cd_atributo	Código do Atributo	Texto
Nm_atributo	Nome do Atributo	Texto

Classe: Cliente

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
Nm_cliente	Nome do Cliente	Texto
Nr_cgc_cpf	Número do CGC/CPF do Cliente	Número
Nm_contato	Nome do Contato	Texto
Endereço ⁸	Endereço do Cliente	

Classe: Cliente Serviço

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_serviço	Código do Serviço	Texto
Vl_serviço	Valor do Serviço	Número

⁸ Endereço: Número do Telefone, Número do Fax, Rua, Nr, Bairro, Cidade, Estado e CEP.

Classe: Estimativa

Atributo	Descrição	Tipo
# Cd_estimativa	Código da Estimativa	Texto
Ds_estimativa	Descrição da Estimativa	Texto
Vl_complexidade_arquivo	Valor da Complexidade do Arquivo	Número
Vl_complexidade_entrada	Valor da Complexidade da Entrada	Número
Vl_complexidade_saída	Valor da Complexidade da Saída	Número
Vl_complexidade_consulta	Valor da Complexidade da Consulta	Número
Vl_complexidade_interface	Valor da Complexidade da Interface	Número
Vl_complexidade_algoritmo	Valor da Complexidade do Algoritmo	Número

Classe: Fator Ajuste

Atributo	Descrição	Tipo
# Cd_estimativa	Código da Estimativa	Texto
# Nr_seq_ajuste	Número da Seqüência de Ajuste	Numero
Ds_característica	Característica	Texto
Vl_mínimo_ajuste	Valor mínimo do Fator de Ajuste	Número
Vl_máximo_ajuste	Valor máximo do Fator de Ajuste	Número

Classe: Função

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Cd_modulo	Código do Módulo	Texto
# Cd_função	Código da Função	Texto
Ds_função	Descrição da Função	Texto
Cd_tipo_função	Tipo de Função	Texto
Cd_prof_resp	Código do Profissional Responsável	Texto
Dt_início	Data de Início da Função	Data/Hora
Dt_término	Data de Término da Função	Data/Hora
Dt_prevista_início	Data Prevista de Início	Data/Hora
Dt_prevista_término	Data Prevista de Término	Data/Hora
Nr_interfaces_acres	Número de interfaces a serem acrescidas	Número
Nr_entradas_acres	Número de entradas a serem acrescidas	Número
Nr_saídas_acres	Número de saídas a serem acrescidas	Número
Nr_consultas_acres	Número de Consultas a serem acrescidas	Número
Nr_algoritmos_acres	Número de algoritmos a serem acrescidas	Número
Id_situação	Situação da Função	Número
Dt_situação	Data da Situação	Data/Hora

Classe: Função Profissional

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Cd_modulo	Código do Módulo	Texto
# Cd_função	Código da Função	Texto
# Cd_profissional	Código do Profissional	Texto
Vl_ponto_função	Valor do Ponto Função	Número
Nr_horas	Número de Horas Trabalhadas	Número

Classe: Função Tabela

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Cd_modulo	Código do Módulo	Texto
# Cd_função	Código da Função	Texto
# Cd_banco	Código do Banco	Texto
# Cd_tabela	Nome da Tabela	Texto
Id_entrada	Tabela de Entrada	Lógico
Id_saída	Tabela de Saída	Lógico
Id_consulta	Tabela de Consulta	Lógico
Id_interface	Tabela de interface	Lógico
Nr_atributos	Número de Atributos	Número

Classe: Histórico Fator de Ajuste Projeto

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Cd_estimativa	Código da estimativa	Texto
# Nr_seq_histórico	Seqüência do Histórico	Número
# Nr_seq_ajuste	Seqüência do Fator de Ajuste	Número
Vl_característica	Valor da Característica	Número

Classe: Histórico Função

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Cd_modulo	Código do Módulo	Texto
# Cd_função	Código da Função	Texto
# Cd_estimativa	Código da Estimativa	Texto
# Nr_seq_histórico	Seqüência do Histórico	Número
Nr_ponto_entrada	Número de Pontos de Entrada	Número
Nr_ponto_saída	Número de Pontos de Saída	Número
Nr_ponto_consulta	Número de Pontos de Consulta	Número
Nr_ponto_interface	Número de Pontos de Interface	Número
Nr_ponto_algoritmos	Número de Pontos de Algoritmos	Número
Nr_ponto_arquivos	Número de Pontos de Arquivos	Número
Vl_compl_entrada	Valor da Complexidade de Entrada	Número
Vl_compl_saída	Valor da Complexidade de Saída	Número
Vl_compl_consulta	Valor da Complexidade de Consulta	Número
Vl_compl_interface	Valor da Complexidade da Interface	Número
Vl_compl_algoritmos	Valor da Complexidade dos Algoritmos	Número
Vl_compl_arquivos	Valor da Complexidade dos Arquivos	Número
Vl_fator_ajuste	Valor do Fator de Ajuste	Número
Cd_profissional_alocado	Código do Profissional Alocado	Texto
Vl_ponto_função	Valor do Ponto Função	Número

Classe: Histórico Módulo

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Cd_modulo	Código do Módulo	Texto
# Cd_estimativa	Código da Estimativa	Texto
# Nr_seq_histórico	Seqüência do Histórico	Número
Dt_estimativa	Data da Estimativa	Data/Hora
Nr_pontos_estimados	Número de Pontos estimados neste Módulo	Número

Classe: Histórico Projeto

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Nr_seq_histórico	Número de Seqüência de Histórico	Número

Classe: Histórico Revisão

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Nr_seq_revisão	Número da Revisão do Projeto	Número
# Cd_estimativa	Código da Estimativa	Texto
# Nr_seq_histórico	Seqüência do Histórico	Número
Nr_pontos_previsos	Número de Pontos Previstos	Número
Nr_pontos_real	Número de Pontos Realmente Trabalhados	Número
Dt_revisão	Data da Revisão	Data/Hora
Cd_prof_revisão	Código do Profissional que fez a Revisão	Texto

Classe: Módulo

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Cd_modulo	Código do Módulo	Texto
Ds_modulo	Descrição do Módulo	Texto
Dt_início	Data de Início do Módulo	Data/Hora
Dt_término	Data de Término do Módulo	Data/Hora
Dt_prevista_início	Data Prevista de Início	Data/Hora
Dt_prevista_término	Data Prevista de Término	Data/Hora
Cd_profissional_resp	Código do Profissional Responsável	Texto
Nr_interfaces_acres	Número de Interfaces a serem acrescentadas	Número
Nr_entradas_acres	Número de Entradas a serem acrescentadas	Número
Nr_saídas_acres	Número de Saídas a serem acrescentadas	Número
Nr_consultas_acres	Número de Consultas a serem acrescentadas	Número
Nr_algoritmos_acres	Número de Algoritmos a serem acrescentados	Número

Classe: Nível Complexidade

Atributo	Descrição	Tipo
# Cd_estimativa	Código da Estimativa	Texto
# Id_identificador	Identificador	Número
# Nr_seq_compl	Seqüência da Complexidade	Texto
Nr_tabela_de	Número de Tabelas de	Número
Nr_tabela_ate	Número de Tabelas ate	Número
Nr_atributos_de	Número de Atributos de	Número
Nr_atributos_ate	Número de Atributos ate	Número
Vl_complexidade	Valor da Complexidade	Número

Classe: Ponto Revisão

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Nr_seq_revisão	Seqüência da Revisão	Número
Cd_profissional_responsável	Código do Profissional Responsável	Texto
Dt_prevista_revisão	Data Prevista de Revisão	Data/Hora
Pc_esperado	Perc. Esperado de Conclusão	Número

Classe: Profissional

Atributo	Descrição	Tipo
# Cd_profissional	Código do Profissional	Texto
Nm_profissional	Nome do Profissional	Texto
Dt_admissão	Data de Admissão	Data/Hora
Dt_demissão	Data de Demissão	Data/Hora
Endereço ⁹	Endereço do Cliente	
Nr_horas_trabalhadas	Número de Horas Trabalhadas por dia	Número
Vl_ponto_função	Valor do Ponto por Função	Número
Cd_serviço	Código do serviço executado atualmente	Texto

Classe: Profissional Alocado

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Cd_profissional	Código do Profissional	Texto
# Nr_seq_alocacao	Número de Seqüência de Alocação	Número
Dt_início_prevista	Data de Início dos Trabalhos	Data/Hora
Dt_término_prevista	Data de Término dos Trabalhos	Data/Hora

Classe: Profissional Férias

Atributo	Descrição	Tipo
# Cd_profissional	Código do Profissional	Texto
# Nr_seq_férias	Seqüência de Férias	Texto
Dt_início_férias	Data de Início das Férias	Data/Hora
Dt_fim_férias	Data de Término das Férias	Data/Hora

⁹ Endereço: Número do Telefone, Rua, Nr, Bairro, Cidade, Estado e CEP.

Classe: Profissional Histórico - RH

Atributo	Descrição	Tipo
# Cd_profissional	Código do Profissional	Texto
# Nr_seq_histórico	Seqüência do Histórico	Texto
Dt_histórico	Data do Histórico	Data/Hora
Vl_salário	Valor do Salário	Número
Cd_serviço	Código do Serviço	Texto

Classe: Projeto

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
Ds_projeto	Descrição do Projeto	Texto
Cd_responsável	Código do Responsável	Texto
Dt_início	Data de Início do Projeto	Data/Hora
Dt_fim	Data de Fim do Projeto	Data/Hora
Dt_prevista_início	Data Prevista de Início do Projeto	Data/Hora
Dt_prevista_fim	Data Prevista de Término do Projeto	Data/Hora

Classe: Projeto Profissional

Atributo	Descrição	Tipo
# Cd_cliente	Código do Cliente	Texto
# Cd_projeto	Código do Projeto	Texto
# Cd_profissional	Código do Profissional	Texto
Nr_pontos	Número de pontos produzidos	Número
Id_empresa	Tempo de Empresa	Número
Id_conhec_técnico	Conhecimento da Tecnologia	Texto
Id_conhec_negócio	Conhecimento do Negócio	Texto
Id_escolaridade	Nível de Escolaridade	Texto

Classe: Serviço

Atributo	Descrição	Tipo
# Cd_serviço	Código do Serviço	Texto
Ds_serviço	Descrição do Serviço	Texto
Vl_serviço	Valor do Serviço	Número
Vl_salário_padrão	Valor do Salário Padrão	Número
Vl_ponto_função	Valor do Ponto por Função	Número

Classe: Tabela

Atributo	Descrição	Tipo
# Cd_banco	Código do banco	Texto
# Cd_tabela	Código da Tabela	Texto
Nm_tabela	Nome da Tabela	Texto

Classe: Tipo de Função

Atributo	Descrição	Tipo
# Cd_tp_função	Código do Tipo de Função	Texto
Ds_tp_função	Descrição do Tipo de Função	Texto
Nr_algoritmos_acres	Número de Algoritmos a ser acrescidos	Número
Nr_interfaces_acres	Número de Interfaces a ser acrescidos	Número
Nr_entradas_acres	Número de Entradas a ser acrescidos	Número
Nr_saídas_acres	Número de Saídas a ser acrescidos	Número
Nr_consultas_acres	Número de Consultas a ser acrescidos	Número
Vl_ponto_função	Valor do Ponto por Função	Número

Bibliografia

- [ABE 96] ABEL, Mara. **Um estudo sobre Raciocínio Baseado em Casos**. Porto Alegre: CPGCC/UFRGS, 1996.
- [AGU 99] AGUIAR, Maurício. Estimativas Confiáveis de Prazos para Gerentes de Projetos. **Developers Magazine**, Rio de Janeiro, v.3, n.33, maio 1999.
- [BEC 99] BECKER, Shirley A. BOSTELMAN, Mitchell L. Aligning Strategic and Project Measurement Systems. **IEEE Software**, New York, v.16, n.3, May/June 1999.
- [BOE 81] BOEHM, Barry. **Software Engineering Economics**. New Jersey: Prentice-Hall, 1981.
- [CAN 99] CASAROTTO FILHO, Nelson; FAVERO Jose S; CASTRO, João E. E. **Gerência de Projetos / Engenharia Simultânea**. São Paulo: Atlas, 1999.
- [CAS 98] CASTRO, Jaelson F. B. Introdução a Medição de Software através de Pontos por Função. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, SBES, 12., 1998. **Anais...** Maringá: Departamento de Informática/Universidade Estadual de Maringá, 1998.
- [DEM 86] DEMING, Edward. **Out of the Crisis**. [S.l.]: MIT Press, 1986.
- [DRA 96] DRAKE, Thomas. Measuring Software Quality: A Case Study. **Computer**, Los Alamitos, v.29, n.11, Nov. 1996.
- [FER 95] FERNANDES, Aguinaldo A. **Gerencia de Software Através de Métricas**. São Paulo: Atlas, 1995.
- [FUR 97] FUREY, Sean Why We Should Use Function Points? **IEEE Software**, New York, v.14, n.2, March/April 1997.
- [GRA 99] GRABLE, Ross et al. Metrics for Small Projects: Experiences at the SED. **IEEE Software**, New York, v.16, n.2, March/April 1999.
- [HAL 97] HALL, Tracy; FENTON, Norman. Implementing Effective Software Metrics Programs. **IEEE Software**, New York, v.14, n.2, March/April 1997.
- [HON 97] HONG, Danfeng. **Software Cost Estimation**. Canadá: Department of Computer Science University of Calgary, 1997 Disponível em: <<http://www.cpsc.ucalgary.ca/~hongd/SENG/621/report2.html>> Acesso em: ago 2000.
- [HUM 95] HUMPHREY, Watts **A Discipline for Software Engineering**, [S.l.]: Addison-Wesley, 1995.
- [IFP 94] IFPUG. **Function Points Counting practice manual**: Relise 4. Westerville, Ohio, 1994.
- [JOH 98] JOHNSON, Kim. **Software Cost Estimation: Metrics and Models**. 1998. Disponível em: <<http://sern.ucalgary.ca/courses/seng/621/w98/johnsenk/cost.htm>>. Acesso em: set. 2000.
- [JON 86] JONES, Capers. **A Short History of Functions Points and Feature Points Software Productivity Research**. [S.l.]: Beurlington Inc, 1986.
- [JON 91] JONES, Capers. **Produtividade no Desenvolvimento de Software**. 2.ed. São Paulo: Makron Books, 1991.
- [KAU 99] KAUTZ, Karlheinz. Making Sense of Measurement for Small Organizations. **IEEE Software**, New York, v.16, n.2, March/April 1999.
- [KAV 2000] KAVACHAR. **Visual Enginnering**. Disponível em: <<http://www.ve.com/kavachart/index.html>>. Acesso em: março 2000.
- [KEM 87] KEMERER, Chris F. An Empirical Validation of Software Cost Estimation Models. **Communications of the ACM**, [S.l.], v.30, n.3, 1987.
- [KOL 93] KOLODNER, J. **Case Based Reasoning**. San Mateo, Califórnia: Morgan

- Kaufmann, 1993.
- [KRA 97] KRAMPE, Dirk. LUSTI, Markus. **Case-Based Reasoning for Information System Design**. [S.l.:s.n], 1997.
- [MAD 97] MADACHY, Raymond J. Heuristic Risk Assessment Using Cost Factors. **IEEE Software**, New York, v.13, n.4, May/June 1997.
- [MAF 92] MAFFEO, Bruno. **Engenharia de Software e Especificação de Sistemas**. Rio de Janeiro: Campus, 1992.
- [MAT 94] MATSON, Jack E; BARRETT, Bruce E; MELLICHAMP, Joseph M. Software Development Cost Estimation Using Function Points. **IEEE Transactions on Software Engineering**, [S.l.], v.20, n.4, April 1994.
- [MAX 2000] MAXWELL, Katrina D. FORSELIUN, Pekka. Benchmarking Software Development Productivity. **IEEE Software**, New York, Jan./Feb. 2000.
- [MAX 99] MAXWELL, Katrina; WASSENHOVE Luk Van, DUTTA Soumitra Performance Evaluation of General and Company Specific Models in Software Development Effort Estimation. **Management Science**, [S.l.], v.45, n.6, June 1999.
- [PIL 97] PILLAI, K; SUKUMARAN, Nair A Model for Software Development Effort and Cost Estimation. **IEEE Transactions on Software Engineering**, [S.l.], v.23, n.5, May 1997.
- [PRE 95] PRESSMAN, Roger S. **Engenharia de Software** 3.ed. São Paulo: Makron Books, 1995.
- [PRE 97] PRESSMAN, Roger S. **Software Engineering** 4.ed. New York: McGraw-Hill, 1997.
- [PUT 78] PUTNAM, Lawrence H. A General Empirical Solution to the Macro Software Sizing and Estimating Problem. **IEEE Transactions on Software Engineering**, [S.l.] v.4, n.4, June 1978.
- [ROT 99] ROTHENBERGER, Marcus A; HERSHAUER, James A software reuse measure: Monitoring an enterprise-level model driven development process. **Information & Management** [S.l.], v. 35, May 1999.
- [RUB 2000] RUBIN, Howard. **The mutiple Dimensions of Metrics: Metrics and the learning Organization**. 1998. Disponível em: <<http://www.cutter.com/itms/itms0002.html>> Acesso em: out. 2000.
- [SAK 98] SAKTHIVEL, Sachi. **A Comparative Study of Two non-sloc Methods to Estimate System Development Effort**. Disponível em: <<http://www.sbaer.uca.edu/docs/proceedingsII/98dsi0840.htm>> Acesso em: set. 2000.
- [SEL 96] SELLERS, Henderson. **Object Oriented Metrics: Measures of Complexity**. New Jersey: Prentice Hall, 1996.
- [SIL 97] SILVA, Djalma D. **Uma proposta de Versão e Ferramenta do PSP para o Domínio de Aplicações Comercias**. São Carlos: USP, 1998.
- [SPC 2000] SOFTWARE PRODUCTIVITY CENTRE. **Tools and Solutions for Software Development**. 2000. Disponível em <<http://www.spc.ca>>. Acesso em: maio 2000.
- [SPC 99] SOFTWARE PRODUCTIVITY CENTRE. **Tools and Solutions for Software Development**. 1999. Disponível em: <<http://ww.spc.ca/about/index.html>> Acesso em: fev. 1999.
- [TSO 99] TSOI, Ho L. The Usage of Estimating Tools for Software Project Development – A Survey Study. **International Journal of the Computer, the internet and management**, [S.l.] v.7, n.2, May/Aug. 1999. Disponível em: <<http://www.journal.au.edu/ijcim/may99/article2.html>>. Acesso em: set. 2000.

- [WEB 99] WEBER, Kival C.; ROCHA, Ana R. C **Qualidade e Produtividade em Software**. 3.ed. São Paulo: Makron Books, 1999.
- [WU 97] WU, Liming. **The Comparison of the Software Cost Estimating Methods**. 1997. Disponível em:
<http://sern.ucalgary.ca/courses/seng/621/w97/wul/seng621_11.html> Acesso em: fev. 1999.