



PUC

ISSN 0103-9741

Monografias em Ciência da Computação
nº 04/05

**Análise e Melhoria de um Processo de
Estimativas de Tamanho de
Projetos de Software**

**Claudia Hazan
Arndt von Staa**

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900

RIO DE JANEIRO - BRASIL

Análise e Melhoria de um Processo de Estimativas de Tamanho de Projetos de Software *

Claudia Hazan

claudinha@inf.puc-rio.br

Arndt von Staa

arndt@inf.puc-rio.br

Abstract. Documented estimation constitutes the basis for the construction of trustworthy software project plans. The size estimation of software projects is used to derive the cost, effort and schedule estimations. Several models have been used by industry to estimate effort and cost given system or artifact size estimates, for example COCOMO. However, the methods to estimate the size of a project have gotten little attention. This work analyzes, from the perspective of CMMI, methods used to generate software projects size estimations using Function Points. It then proposes a new method for estimating size based on experiments. Finally, we describe a simplified process for generating and documenting estimations regarding size, effort, budget, schedule and required critical computer resources.

Keywords: Estimations, Function Point Analysis, Software Project Management, CMMI.

Resumo. Estimativas documentadas constituem a base para a elaboração de planos fidedignos do projeto de software. A estimativa do tamanho de projetos de software é utilizada para a derivação das estimativas de custo, esforço e cronograma. Muitos modelos têm sido utilizados pela indústria para estimar-se esforço e custo baseados em estimativas de dimensão dos artefatos, como por exemplo o COCOMO. Neste trabalho analisamos, a partir de uma perspectiva baseada em CMMI, métodos utilizados para geração de estimativas de tamanho de projetos de software baseadas em Pontos de Função. A seguir propomos um novo método para geração de estimativas de tamanho, baseando-se em experimentação. Finalmente apresentamos um processo simplificado para a geração e a documentação de estimativas de tamanho, esforço, custo, cronograma e necessidade de recursos computacionais críticos.

Palavras-chave: Estimativas, Análise de Pontos por Função, Gerenciamento de Projetos de Software, CMMI.

* Trabalho patrocinado pelo Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil

Responsável por publicações:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3114-1516 Fax: +55 21 3114-1530
E-mail: bib-di@inf.puc-rio.br

Sumário

| | |
|---|-----------|
| 1. INTRODUÇÃO | 1 |
| 1.1. ORGANIZAÇÃO DO ARTIGO..... | 2 |
| 2. ESTADO DA ARTE | 3 |
| 2.1. PROPOSTA | 7 |
| 3. ELABORAÇÃO | 8 |
| 3.1. ANÁLISE DO PROCESSO DE ESTIMATIVAS DO SERPRO | 8 |
| 3.2. PROPOSTA E IMPLEMENTAÇÃO DE MELHORIAS NO PROCESSO DE ESTIMATIVAS | 9 |
| 3.2.1. MÉTODOS PARA GERAÇÃO DE ESTIMATIVAS DE TAMANHO | 10 |
| 3.2.1.1. MÉTODO CONTAGEM INDICATIVA | 10 |
| 3.2.1.2. MÉTODO CONTAGEM INDICATIVA INTELIGENTE | 11 |
| 3.2.1.3. MÉTODO DE CONTAGEM ESTIMATIVA | 12 |
| 3.2.1.4. MÉTODO DAS ESTIMATIVAS PERCENTUAIS | 15 |
| 3.2.2. MÉTODO PARA GERAÇÃO DE ESTIMATIVAS DE ESFORÇO..... | 17 |
| 3.2.3. MÉTODO PARA GERAÇÃO DE ESTIMATIVAS DE PRAZO | 20 |
| 3.2.4. MÉTODO PARA GERAÇÃO DE ESTIMATIVAS DE CUSTO | 22 |
| 3.2.5. ESTIMATIVAS DE RECURSOS COMPUTACIONAIS CRÍTICOS | 22 |
| 4. DISCUSSÃO E CONTRIBUIÇÕES | 23 |
| 4.1. INTEGRANDO O PROCESSO DE ESTIMATIVAS PROPOSTO COM O XP..... | 24 |
| 5. ESTUDO DE CASO | 26 |
| 6. CONCLUSÕES E RECOMENDAÇÕES PARA FUTUROS TRABALHOS | 28 |

1. Introdução

Os problemas relativos à previsibilidade de projetos constituem uma das preocupações da indústria de software. As estatísticas da *Scientific American* [Filho, 2000] mostram que o tempo realizado dos projetos de software excede em 50% o tempo planejado no cronograma do projeto. O *Standish Groups* relatou em 1994 [Standish, 1994] que apenas 16% dos projetos de software atingem o seu objetivo dentro do cronograma e do orçamento previstos. Por outro lado, Glass em [Glass, 2003] afirma que algumas das principais causas para isso são estimativas feitas baseadas em informação insuficiente (usualmente sem que a especificação de requisitos esteja disponível), estimativas feitas para satisfazer a direção, e projetos com requisitos demasiadamente flutuantes. Os dados de 2001 do *Standish Groups* [Standish, 2001] mostram as seguintes estatísticas: 27% dos projetos de software são finalizados no tempo e custo previstos; 40% dos projetos são cancelados antes de finalizarem; 42% dos projetos não apresentam as funcionalidades propostas originalmente; 50% dos projetos custam em média 180% a mais da estimativa original. Cabe salientar que ocorreu uma melhoria nas estatísticas de projetos que terminam dentro prazo (cronograma) e orçamento previstos do *Standish Group* de 1994 (16%) para 2001 (27%). Esta melhoria ocorreu devido a percepção da indústria sobre a importância estratégica do software [Pressman, 2000], devido ao surgimento de clientes mais exigentes e, especialmente, devido aos investimentos na implantação das melhores práticas da Qualidade de Software, preconizadas pelos modelos e normas da Qualidade. Neste contexto, é importante destacar o crescimento significativo da certificação ISO 9001 [ABNT, 1994] na indústria de software e qualificação em modelos da Qualidade de Software, como o CMM (*Capability Maturity Model*) [PAU93] no Brasil [MCT, 2004a] [MCT, 2004b]. Os dados da *Boeing Information Systems* apresentados em [Masters, 2000] demonstram que a previsibilidade de cronograma do projeto aumenta conforme a evolução nos níveis de maturidade do modelo CMM. A variação de cronograma da Boeing foi reduzida em 125% durante a transição do nível CMM 1 para o nível 2 e então em 24% durante a transição do nível 2 para o nível 3 e em 15% durante a transição do nível 3 para o nível 4. Os dados foram coletados entre os exercícios 1991 e 1999.

Embora tenham ocorrido melhorias, a indústria continua lidando com projetos mal sucedidos. O *CHAOS Report* de 2003 [Standish, 2003] apresentou os seguintes dados: apenas 34% dos projetos são bem sucedidos; 15% dos projetos foram cancelados; 43% é o erro médio em relação ao orçamento do projeto daqueles que foram completados; e apenas 52% das características (requisitos não funcionais) e funcionalidades são entregues no produto.

A maioria das tecnologias e metodologias de engenharia de software preocupam-se primordialmente com a qualidade da solução de software gerada. No entanto, existem também vários problemas associados à gestão do projeto, tais como: tempo estimado de duração do projeto, custo, alocação de recursos, esforço gasto e outros. Muitos desses aspectos podem ser estimados quantitativamente. A grande dificuldade reside na geração de estimativas fidedignas.

Note que se um projeto é subestimado, mesmo utilizando-se os melhores métodos para a elicitação de requisitos, desenho, implementação e testes, o projeto corre o risco de ser cancelado. O principal risco que afeta os projetos de software subestimados é o de pressão excessiva do cronograma, o qual pode forçar o término prematuro do projeto, colocando em risco a qualidade e a funcionalidade do projeto [Jones, 1994]. Entre as possíveis conseqüências desse risco temos: cancelamento do projeto, baixa moral da equipe, baixa qualidade, atrasos de cronograma, custos excessivos devido a um grande número de horas extras e atritos entre o gerente e a equipe [Hazan, 1999].

Assim, considera-se um dos principais problemas que afetam os projetos de software são os gerenciais [Forsberg, 1997]. Estes problemas, que possuem maior impacto em projetos complexos, freqüentemente estão associados à gestão de requisitos e à previsibilidade de prazo e custo. As principais decisões que devem ser tomadas por um gerente de projetos incluem, dentre outras, a negociação entre custo, cronograma, funcionalidade e fatores da qualidade. A definição do orçamento e do cronograma constitui uma base para o planejamento e controle do projeto. Por exemplo, quantas pessoas devem ser alocadas nas fases inicial, intermediárias e final do projeto. Quanto esforço, custo e cronograma deve ser esperado para atingir os principais marcos do projeto [Boehm, 2000].

Um dos principais riscos que atinge o processo de estimativas é a falta de credibilidade nas estimativas pelas equipes de desenvolvimento [Boehm, 2000]. Isto ocorre quando as estimativas são irreais, ou seja quando os projetos são subestimados ou superestimados. A acurácia das estimativas de tamanho torna-se fundamental para a elaboração de um cronograma e orçamento realistas, pois as estimativas de tamanho constituem a base para a derivação das estimativas de custo e de esforço [SEI, 2002].

Na área de geração de estimativas de esforço, prazo e custo, destacam-se os modelos paramétricos COCOMO II (*Constructive Cost Model*) [Boehm, 2000] e o SLiM (*Software Lyfe Cycle Model*) [Putnan, 1992]. Estes modelos foram publicados [Boehm, 2000] [Putnan, 1992] e são respeitados tanto no meio acadêmico quanto na indústria [Aguiar, 2004]. No entanto, os métodos para a geração das estimativas de tamanho de projetos de software, que constituem o principal insumo para a utilização dos modelos de estimativa de esforço, têm sido muito pouco discutido pela literatura [Meli, 1997].

Este trabalho tem como objetivo descrever e analisar o processo de geração de estimativas do Serviço Federal de Processamento de Dados (SERPRO), uma organização governamental de desenvolvimento de soluções de Tecnologia da Informação e de Comunicações, e implantar melhorias por meio do estabelecimento de um novo método de estimativas de tamanho de projetos de desenvolvimento e de manutenção evolutiva de software, integrando os conceitos da Análise de Pontos de Função ao conhecimento inicial do domínio do projeto. Este método foi definido por meio de pesquisa de métodos de estimativa de tamanho em Pontos de Função, e a adaptação destes métodos a partir da experiência adquirida. O propósito é melhorar a acurácia das estimativas de tamanho. Além disso, também é apresentado um processo simplificado para geração e documentação de estimativas. Este processo trata as estimativas de tamanho, esforço, prazo, custo e recursos computacionais críticos, seguindo as práticas de estimativas definidas na área de processo - Planejamento do Projeto de Software do modelo CMMI [SEI, 2002].

1.1. Organização do artigo

Este trabalho está organizado da seguinte forma: a seção 2 apresenta o estado da arte das atividades de planejamento de projetos de software, destacando as diretrizes para os processos de geração de estimativas do modelo CMMI. A seção 3 apresenta uma análise do processo de estimativas do Serviço Federal de Processamento de Dados (SERPRO). Esta análise utiliza a ferramenta da qualidade: Diagrama de Causa e Efeito. Também é apresentada uma proposta de implementação de melhorias no processo analisado por meio da definição de dois métodos de estimativa de tamanho de projetos de software em Pontos de Função, denominados Método de Contagem Indicativa Inteligente e Método de Contagem Estimativa. Estes métodos possibilitam a geração e documentação das estimativas no começo do ciclo de vida do desenvolvimento de software, ou seja durante a fase de planejamento. Além disso, nesta seção é descrito um procedimento simplificado para a derivação das estimativas de esforço, prazo e custo, utilizando a estimativa de tamanho.

A seção 4 apresenta uma proposta de integração do procedimento de estimativas proposto no processo de planejamento do Extreme Programming (XP). A seção 5 descreve um Estudo de Casos, ilustrando a aplicação do procedimento descrito na seção 3. Finalmente, a seção 6 conclui o trabalho e identifica possíveis trabalhos futuros.

2. Estado da arte

Como já foi visto, estimativas constituem o primeiro tipo de análise quantitativa executado no projeto de software. Estimativas de má qualidade (sem acurácia) constituem uma das principais causas dos projetos cancelados [Jones, 1994]. As estimativas inadequadas podem levar a prazos não cumpridos, custos excessivos com horas extras e má qualidade do software. Esta seção tem como propósito apresentar uma visão geral das práticas de planejamento de projetos de software descritas no CMMI [SEI, 2002], destacando as atividades relativas às estimativas de projetos. A área de processo de planejamento de projetos considera as seguintes estimativas: tamanho, esforço, custo, cronograma e necessidade de recursos computacionais críticos. Também são apresentadas as métricas mais utilizadas pela indústria de software para as estimativas de tamanho de projetos, a saber: Pontos de Função (PF) e Pontos por Casos de Uso (PCU).

Humphrey [Humphrey, 1990] define um projeto como sendo um esforço de trabalho realizado por um grupo de pessoas com um objetivo bem definido, dentro de prazo estabelecido e com recursos limitados. Dessa forma, pode-se caracterizar um projeto como uma ou mais demandas do cliente. Por exemplo: a elaboração de um sistema novo; a evolução de uma ou mais funcionalidades de um sistema existente (manutenção evolutiva); e a migração de um sistema existente para outra plataforma ou mudança de versão da linguagem de desenvolvimento do sistema (manutenção adaptativa).

O Gerenciamento de Projetos é a aplicação de conhecimentos, habilidades, e técnicas para projetar atividades que visem atingir ou exceder as necessidades e expectativas das partes envolvidas, com relação ao projeto. O ato de atingir ou exceder as necessidades e expectativas das partes envolvidas, envolve o equilíbrio entre demandas concorrentes de escopo, prazo, custo e qualidade [PMI, 2000].

Um dos objetivos da Gerência de Projetos é identificar, estabelecer, coordenar, e monitorar as atividades, tarefas e recursos necessários de um projeto para produzir um produto e/ou serviço, dentro do contexto dos requisitos e restrições do projeto [ABNT, 1998]. Um segundo objetivo é acompanhar, dirigir e registrar os resultados produzidos e os recursos consumidos no desenrolar da execução do projeto, confrontando-os com o que era esperado pelos planos. Como consequência desse confronto o gerente estará capacitado a tomar medidas corretivas racionais, sempre que a execução se desviar mais do que o tolerado do que era esperado. Os dois principais processos do Gerenciamento de Projetos de Software são: Planejamento do Projeto e Acompanhamento do Projeto. As principais atividades contidas nestes processos consistem no estabelecimento e acompanhamento das estimativas de tamanho, cronograma, recursos, custos e eficiência de remoção de defeitos e na obtenção e manutenção de compromissos [Jones, 1997] [SEI, 2002]. O planejamento do projeto de software inicia com uma visão geral dos requisitos que definem o produto, e tem como propósito estabelecer e manter planos que definam as atividades a serem realizadas durante o projeto. A área de processo *Planejamento de Projeto* do nível 2 do modelo CMMI envolve as seguintes atividades [SEI, 2002]:

- Desenvolver o plano do projeto
- Interagir apropriadamente com os interessados (*stakeholders*)

- Planejar o envolvimento dos interessados identificados
- Obter compromisso com o plano
- Manter o plano

O estabelecimento de estimativas constitui uma das principais atividades do processo de planejamento do projeto de software. As estimativas devem ter uma base segura e fornecer confiança que os planos nelas baseados são capazes de conduzir aos objetivos do projeto, e que englobam funcionalidades, custo, prazo e características da qualidade. Os fatores tipicamente considerados nas estimativas incluem, dentre outros, o seguinte:

- Requisitos do projeto
- Escopo do projeto
- Modelo do ciclo de vida do projeto selecionado (ex.: cascata, incremental, etc.)
- Abordagem técnica
- Tarefas e artefatos identificados e seus atributos (ex.: tamanho e complexidade)
- Modelos e dados históricos para conversão dos atributos dos artefatos e tarefas em horas de trabalho e custo

As estimativas e as premissas utilizadas para sua geração devem ser documentadas para possibilitar o acompanhamento do plano, conforme o progresso do projeto. O processo de **Estabelecer Estimativa** abrange, dentre outras, as seguintes atividades [SEI, 2002]:

- **Estabelecer e manter as estimativas dos atributos dos artefatos e das tarefas:** as estimativas de tamanho constituem o insumo principal para a maioria dos modelos usados para estimar custo, esforço e cronograma. As estimativas de tamanho devem ser consistentes com os requisitos do projeto para determinar-se esforço, custo e cronograma adequados. Assim, sempre que ocorrerem mudanças além do tolerado (relevantes) nos requisitos, as estimativas precisam ser revistas e o projeto replanejado. Sugere-se a utilização da métrica Pontos de Função (PF) [IFPUG, 2004] para as estimativas de tamanho. Os PF fornecem uma métrica de tamanho do projeto de software, quantificando as suas funcionalidades, sob o ponto de vista lógico, observando os requisitos do usuário [Garmus, 2001]. Além de apoiar a implantação dos modelos CMM [Hazan, 2003] e CMMI [Dekkers, 2002], a métrica foi reconhecida pela norma ISO/IEC 20926 [Dekkers, 2003].
- **Determinar as estimativas de esforço e de custo para os artefatos e tarefas:** as estimativas de esforço e custo devem ser geradas baseando-se em dados objetivos, ou seja, utilizando métodos e dados históricos. Os projetos com ausência de dados históricos de esforço de projetos similares disponíveis possuem um risco maior, necessitando de mais pesquisas para desenvolver bases racionais de estimativas.
- **Incluir a necessidade de uma infra-estrutura de suporte nas estimativas de esforço e custo:** considerar a necessidade de recursos computacionais críticos no ambiente de desenvolvimento, no ambiente de teste, no ambiente de produção, ou em qualquer combinação destes. Estimativas de recursos computacionais incluem o seguinte: identificar os recursos computacionais críticos; e basear estimativas de recursos computacionais críticos em requisitos alocados. Exemplos de recursos computacionais incluem: memória, processador, espaço em disco, periféricos, capacidade da rede e do banco de dados.

Uma vez que as estimativas foram estabelecidas, desenvolve-se o plano do projeto, que fornece uma base para execução e controle das atividades do projeto. O plano do projeto precisa ser revisado quando ocorrem mudanças de requisitos ou de compromissos durante o projeto. Este também precisa ser revisado quando forem observados erros estruturais nas estimativas utilizadas.

O propósito do processo do **Acompanhamento do Projeto** é fornecer um entendimento visível e confiável do progresso do projeto de modo que ações corretivas apropriadas possam ser tomadas quando o desempenho desviar significativamente do plano. Um desvio é significativo se, caso não seja resolvido, o projeto não alcançará seus objetivos (funcionalidades, atributos, custo e prazo). O progresso é determinado pela comparação dos artefatos, tamanho, esforço, custo e cronograma atuais e os planejados nos marcos estabelecidos no plano do projeto.

Note que o CMMI define as diretrizes para um processo de planejamento de projetos, estabelecendo as principais estimativas a serem consideradas: tamanho, esforço, prazo, custo e recursos computacionais críticos. No entanto, o modelo não descreve como estimar o tamanho de projetos de desenvolvimento e de manutenção de software, como derivar as estimativas de esforço, prazo e custo a partir das estimativas de tamanho.

As métricas de tamanho de projeto de software mais utilizadas pela indústria são as seguintes: Pontos de Função (PF) , Linhas de Código (LOC - *Line of Code*) e Pontos por Casos de Uso (PCU). O principal benefício das métricas PF sobre a LOC é que os Pontos de Função podem ser obtidos no início do ciclo de vida, diretamente das especificações de requisitos do projeto. Além disso, os PFs são úteis para a geração das estimativas de tamanho do projeto, independentemente da metodologia e linguagem de programação utilizada no desenvolvimento. Um método de estimativas de Pontos de Função bastante utilizado é o Contagem Indicativa proposto pela NESMA (*Netherlands Software Metrics Association*) [NESMA, 2005]. Este método foi considerado no estabelecimento de um novo método definido neste trabalho - o método Contagem Indicativa Inteligente.

A métrica Pontos por Casos de Uso (PCU) foi proposta por Gustav Karner [Karner, 1993] com o propósito de estimar recursos para projetos de software orientados a objeto, modelados por meio de especificação de Casos de Uso. Em linhas gerais, o método de contagem de Pontos por Caso de Uso consiste nos seguintes passos:

1. Contar os atores e identificar sua complexidade;
2. Contar os casos de uso e identificar sua complexidade;
3. Calcular os PCUs não ajustados;
4. Determinar o fator de complexidade técnica;
5. Determinar o fator de complexidade ambiental;
6. Calcular os PCUs ajustados.

Com o resultado desta medição e sabendo-se a produtividade média da organização para produzir um PCU, pode-se então estimar o esforço total para o projeto. A seguir são mostradas as principais vantagens da métrica Pontos de Função (PF) sobre a métrica PCU.

A métrica Pontos por Caso de Uso foi apresentada para o mercado como uma solução simples para as estimativas de tamanho, especialmente para os que não conheciam as práticas de Contagem de Pontos por Função ou as consideravam complexas para serem aplicadas, descritas em [IFPUG, 2004].

Além disso, existe uma falsa noção de que a métrica Pontos de Função não pode ser aplicada para dimensionar sistemas orientados a objetos. Como a contagem de Pontos de Função, considera os requisitos funcionais do sistema, de maneira independente da metodologia utilizada¹, os PFs também podem ser aplicados para sistemas cujos requisitos tenham sido modelados utilizando casos de uso. Enquanto que, o PCU somente pode ser aplicado em projetos de software cuja especificação tenha sido expressa em casos de uso. Note que a contagem de PF independe da forma como os requisitos do software foram expressos. Esta vantagem do PF foi citada pelo próprio Gustav Karner [Karner, 1993].

Não é possível na prática aplicar PCU na medição de aplicações existentes cuja documentação esteja desatualizada ou sequer exista. A alternativa seria escrever os casos de uso destas aplicações para só então medi-las. Porém isto tornaria a medição inviável devido ao alto custo. Com o PF é possível realizar a medição analisando-se a própria aplicação em uso (aplicação implantada ou pacote). A medição de aplicações instaladas pode ser útil na elaboração de contratos de *outsourcing* de sistemas e na aquisição de pacotes.

Não existe um padrão único para a escrita do caso de uso. Diferentes estilos na escrita dos casos de uso ou na sua granularidade podem levar a resultados diferentes na medição por PCU. A métrica Pontos de Função não sofre deste problema pois independe da forma como os requisitos são expressos ou documentados.

Devido ao processo de medição do PCU ser baseado em casos de uso, o método não pode ser empregado antes de concluída a análise de requisitos do projeto. Na maioria das vezes há necessidade de se obter uma estimativa antes da finalização desta etapa. O processo de contagem de PF também só pode ser empregado após o levantamento dos requisitos do projeto, na fase de projeto lógico. Porém existem técnicas estimativas de tamanho em pontos de função que podem ser aplicadas com sucesso antes da análise de requisitos ser concluída. Alguns destes métodos serão apresentados neste trabalho, a saber: Contagem indicativa, Contagem indicativa inteligente, Contagem estimativa e Estimativas percentuais.

A métrica PCU não contempla a medição de projetos de melhoria funcional do software (manutenção evolutiva), somente projetos de desenvolvimento. A métrica PF contempla a medição de projetos de desenvolvimento, projetos de manutenção evolutiva e aplicações implantadas. Note que a métrica Pontos de Função pode ser usada para dimensionar os projetos concluídos de uma organização, possibilitando a geração de uma *baseline* para as estimativas.

Não existe um grupo de usuários ou organização responsável pela padronização ou evolução do método PCU; e a bibliografia sobre o assunto é escassa. Para a métrica PF existe o IFPUG (*International Function Point Users Group*) que é o responsável por manter o Manual de Práticas de Contagem. Além disso, o método PCU não é aderente à norma ISO/IEC 14143 que define um modelo para a medição funcional de software. A APF, conforme o manual do IFPUG versão 4.1.1, está padronizada sob a norma ISO/IEC 20926.

Não existe um programa de certificação de profissionais que conheçam a técnica do PCU e saibam aplicá-la de forma adequada. A contagem de Pontos de Função possui o programa de certificação CFPS (*Certified Function Point Specialist*) promovido pelo IFPUG, garantindo assim a utilização correta do método Análise de Pontos de Função (APF) por um especialista certificado. É importante destacar que a certificação deve ser renovada a cada três anos, para garantir a atualização do especialista nas novas versões do manual ou até mesmo na versão em que ele foi certificado.

¹ Cabe salientar que Pontos de Função é um indicador de dimensão. O esforço é calculado a partir dos PF e da produtividade observada. Esta última evidentemente depende da tecnologia utilizada além da proficiência da equipe.

O fator ambiental inserido no PCU é subjetivo, dificultando a consistência da aplicação do método em programas de métricas de software e *benchmarking* entre organizações, pois torna o tamanho de um projeto variável; sem que sua funcionalidade sequer mude. Se um mesmo projeto for entregue a duas equipes distintas a contagem dos PCUs deste projeto será também diferente em cada situação. Ou seja, o mesmo projeto possui dois tamanhos distintos. É importante enfatizar que o fator de ajuste da contagem de PF, que possui finalidade semelhante ao fator ambiental da métrica Pontos por Caso de Uso, não é subjetivo. O fator de ajuste é calculado baseando-se na ponderação do Nível de Influência (NI) das 14 Características Gerais do Sistema (CGS) definidas. A ponderação do NI, que varia em uma escala de zero a cinco, é baseada em descrições contidas no manual de práticas de contagem - CPM 4.2 [IFPUG 2004]. Na versão do manual (CPM 4.2), publicada em 2004, foram incluídas *guidelines* para apoiar o entendimento das descrições associadas aos níveis de influência de zero a cinco das características gerais do sistema.

Embora muitas organizações e referências bibliográficas considerem os Pontos por Caso de Uso como o estado da arte em métricas de tamanho de software [Andrade, 2004] [Cunha, 2003] [Damodaram, 2004], analisando-se as desvantagens expostas acima pode-se concluir que o PCU não traz nenhum benefício adicional sobre o PF. Também é importante destacar a maturidade da métrica PF na indústria de software (25 anos de criação e evolução) em relação à métrica PCU (menos de 10 anos). E ainda, na aplicação de métricas para estimar-se tamanho, a métrica de Pontos de Função possui uma vantagem adicional que é a sua utilização como insumo para o COCOMO, modelo utilizado para a derivação das estimativas de esforço, prazo e custo a partir das estimativas de tamanho. Assim, os métodos de estimativas de tamanho propostos neste trabalho são baseados na métrica PF.

2.1. Proposta

A literatura tem discutido bastante os modelos de estimativas de esforço, prazo e custo. O modelo COCOMO II [Boehm, 2000], desenvolvido na *University of Southern California* (USC) tem sido utilizado pela indústria [Aguiar, 2004]. As ferramentas SLiM, comercializada pela empresa QSM (*Quantitative Software Management*) e *Knowledge Plan* criada pela empresa SPR (*Software Productivity Research*), que utilizam modelos de estimativas proprietários, também tem sido utilizadas pela indústria. No entanto, a maioria das empresas de software continuam estimando projetos, baseando-se na opinião e sentimento do gerente ou líder do projeto. A dificuldade não está na utilização dos modelos de estimativas de esforço, prazo e custo. E sim no estabelecimento das estimativas de tamanho que constitui o principal insumo para a geração das estimativas de esforço, prazo e custo.

Na seção anterior, foram abordadas as práticas definidas pelo CMMI para o estabelecimento de um processo de estimativas, bem como as principais métricas de tamanho utilizadas pela indústria de software. Este trabalho tem como propósito definir e implantar um processo simplificado de estimativas de projetos de software, baseando-se na aplicação prática das diretrizes do CMMI, descritas nas práticas específicas de Planejamento de Projetos de Software. O foco deste trabalho encontra-se na análise e no estabelecimento de métodos para estimativas de tamanho de projetos de software. A métrica utilizada para as estimativas de tamanho é a de Pontos de Função (PF), devido aos seus benefícios, descritos na seção anterior. Assim, será apresentado como obter o tamanho aproximado de um projeto de software em Pontos de Função no início do ciclo de vida. E ainda, como derivar as estimativas de esforço, prazo e custo, de maneira empírica, da estimativa de tamanho gerada.

3. Elaboração

Esta seção tem como propósito apresentar um processo de estimativas simplificado, com ênfase nos métodos de estimativa de tamanho. Este processo foi definido para apoiar a organização Serviço Federal de Processamento de Dados (SERPRO) na certificação CMM - Nível 2 e encontra-se integrado sob a forma de um procedimento no processo de desenvolvimento de software da empresa - o PSDS (Processo Serpro de Desenvolvimento de Soluções). O primeiro passo executado para definição do procedimento foi a análise do processo estimativas do SERPRO, por meio da ferramenta **diagrama de causa e efeito**. Além do procedimento também foi definido um artefato, denominado Estimativas Documentadas (ED) que tem a finalidade de apoiar a documentação das estimativas e das premissas utilizadas para as estimativas. Este artefato é utilizado nas macroatividades de planejamento e acompanhamento dos projetos de software, definidas no processo da empresa.

3.1. Análise do Processo de Estimativas do SERPRO

Esta seção mostra uma aplicação da ferramenta da qualidade **diagrama de causa e efeito** na identificação do relacionamento entre as principais causas do efeito falhas do processo de estimativas do SERPRO.

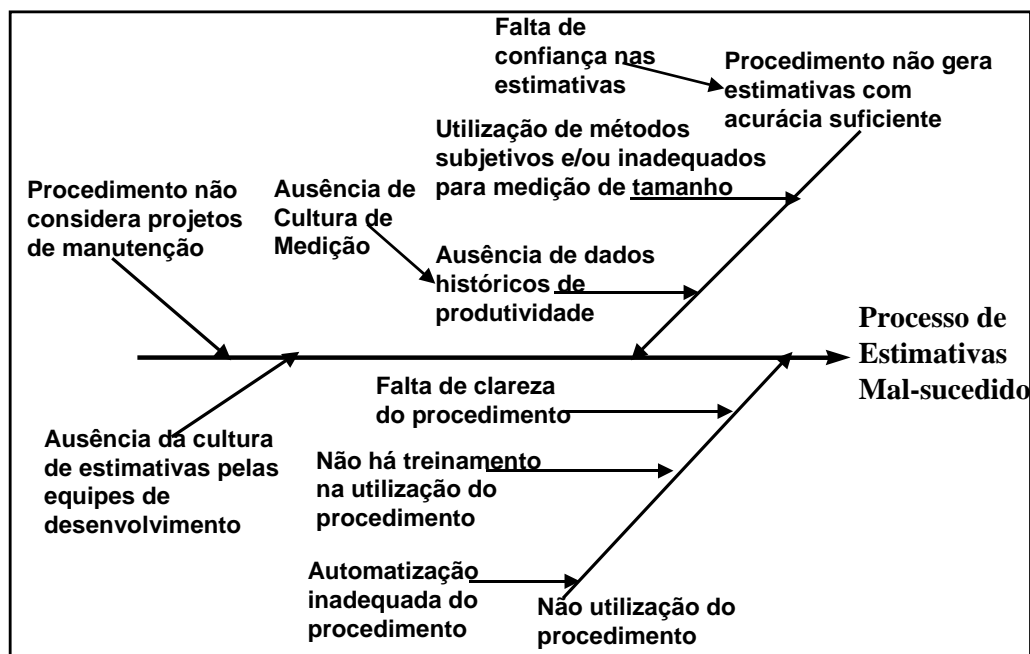


Figura 1: Análise de Falhas de um Processo de Estimativas de Software

O diagrama de causa e efeito, também conhecido como gráfico espinha-de-peixe, mostra o relacionamento entre a característica da qualidade e os fatores que afetam esta característica. Sua apresentação assemelha-se com uma espinha-de-peixe, com a característica da qualidade rotulada na posição da cabeça do peixe, e os fatores que afetam esta característica colocados onde os “ossos” estão localizados [Kan, 1995] [Oliveira, 1996]. A **Figura 1** ilustra a identificação das principais causas dos problemas do processo de estimativas do SERPRO.

A causa raiz dos principais problemas do processo de estimativas foi a utilização da métrica Pontos por Caso de Uso (PCU), discutida na seção 2. *Estado da Arte*. A métrica PCU [Probasco, 2002] [Smith, 2000] foi apresentada para as equipes de desenvolvimento como uma “*Silver Bullet*” para as estimativas de tamanho, ou seja, seria simples e eficaz.

Além disso, foi também erradamente colocado que a métrica Pontos de Função não seria adequada para estimar sistemas orientados a objetos ou sistemas modelados por meio de Casos de Uso. No entanto, a métrica Pontos de Função mede tamanho funcional, independentemente da tecnologia utilizada [IFPUG, 2004]. Dekkers [Dekkers, 2001], Aguiar [Aguiar, 1992] e Longstreet [Longstreet, 2002] afirmam que a Especificação de Casos de Uso é um excelente método para modelar requisitos e Pontos de Função é uma excelente métrica para aferição de tamanho funcional dos requisitos elicitados, considerando o ponto de vista do usuário. O principal problema que os desenvolvedores encontraram na utilização da métrica PCU foram os seguintes:

- Falta de acurácia das estimativas de esforço, devido a ausência de base histórica;
- Dificuldade de estimar pequenos projetos de manutenção ou apurações especiais, onde não há modelagem de requisitos, por exemplo: desenvolvimento de uma rotina para atualização de um arquivo e geração de um relatório;
- Dificuldade de geração de dados históricos de produtividade, devido à subjetividade do método. Cada pessoa tem a sua maneira de modelar os requisitos em Casos de Uso, alguns mais detalhistas outros menos prolixos. Por exemplo, alguns analistas modelam a funcionalidade cadastro de alunos com apenas um Caso de Uso - Cadastrar Alunos. Outros modelam com dois casos de uso: Cadastrar Alunos e Consultar Alunos. Já os mais detalhistas modelam com quatro Casos de Uso: Incluir Aluno, Alterar Aluno, Excluir Aluno e Consultar Aluno. Assim, o mesmo sistema pode ter uma medida de tamanho em Pontos por Caso de Uso distinta;
- Impossibilidade de geração de estimativas iniciais para o plano do projeto, a partir do artefato Documento de Visão [Kruchten, 2000] ou Proposta Comercial, onde não existem Casos de Uso modelados e sim necessidades e funcionalidades. O modelo CMM requer que o plano, contendo as estimativas, deve ser produzido no início do projeto, quando os requisitos estão sendo elicitados e ainda não foram modelados [Fiorini, 1998].

A ausência de cultura em métricas pelas equipes de desenvolvimento, associada à não adequação da métrica Pontos por Casos de Uso, à ausência de dados históricos de produtividade e de treinamentos geraram uma falta de credibilidade nas estimativas pelas equipes de desenvolvimento. Como consequência, as atividades associadas às estimativas caíram em desuso, especialmente pelos responsáveis por projetos de pequenas manutenções, onde os requisitos não são modelados por Casos de Uso.

3.2. Proposta e Implementação de Melhorias no Processo de Estimativas

A análise do diagrama de causa e efeito (**Figura 1**), apresentado na **seção 3.1**, permitiu a visualização de um grande problema de forma organizada. As limitações do diagrama é que este não serve para priorizar as causas a serem tratadas. A decisão realizada foi a criação de um novo procedimento de estimativas, considerando as causas de problemas apontadas e a criação do curso “Processo de Estimativas” com *mentoring*, visando a capacitação das equipes de desenvolvimento no procedimento.

O novo procedimento para geração de estimativas de tamanho utilizou os princípios do modelo CMM [Fiorini, 1998], devido à busca pela empresa da qualificação CMM - nível 2 e do modelo XP [Beck, 2000], com o intuito de tornar o processo de estimativas simples, devido a dificuldade de aquisição uma ferramenta para geração de estimativas pela empresa. A abordagem de utilizar as diretrizes de simplicidade do XP para alcançar as metas do modelo CMM é defendida por Orr [Orr, 2002] e por Paulk [Paulk, 2001].

O propósito da criação do procedimento é o de permitir que os líderes de projetos gerem as estimativas de tamanho, esforço, prazo, custo e recursos computacionais do projeto de software de forma consistente e padronizada para apoiar a elaboração do plano do projeto. O procedimento tem como objetivo integrar métodos para apoiar as seguintes atividades:

- Gerar estimativas de tamanho do projeto de software;
- Derivar estimativas de esforço, custo e prazo do projeto de software.

3.2.1. Métodos para Geração de Estimativas de Tamanho

Nesta seção propõe-se métodos para geração de estimativas de tamanho de projetos de software em Pontos de Função. A métrica Pontos de Função visa estabelecer uma estimativa confiável do tamanho do software, considerando a funcionalidade a ser entregue pela aplicação, independentemente da metodologia de desenvolvimento e da plataforma utilizadas no desenvolvimento da aplicação [Garmus, 2001].

Além disso, na prática a APF tem demonstrado grande utilidade na garantia da completeza da especificação de requisitos. Uma contagem de PF é realizada usando uma descrição formal das necessidades do negócio na linguagem do usuário. Assim, a contagem apoia uma revisão dos requisitos do sistema com o usuário [Hazan, 2003].

Na fase de Planejamento do Projeto de Software, os desenvolvedores possuem pouco conhecimento sobre o sistema a ser desenvolvido, tornando-se impraticável a aplicação do método de Contagem de Pontos de Função publicado no manual CPM 4.2 que preconiza a existência do projeto lógico da aplicação [IFPUG, 2004]. Note que nesta fase ainda não temos todos os requisitos elicitados, e portanto não existe um projeto lógico da aplicação completo e validado. Assim, foram identificados na literatura métodos para estimar o tamanho do projeto em Pontos de Função no início do projeto, utilizando o Documento de Visão.

Além dos métodos pesquisados, foi criado um novo método, denominado “NESMA Inteligente”, que consiste na calibragem dos índices utilizados no método Contagem Indicativa [NESMA, 2005], utilizando o conhecimento inicial dos requisitos do projeto, com a finalidade obter-se uma melhor acurácia nas estimativas de tamanho. O método criado tem sido aplicado com sucesso, gerando estimativas com menos de 10% de erro, especialmente em contagens rápidas de sistemas implementados. A grande vantagem da contagem de PF é que o método fornece facilidades para reestimativas. Então, a contagem estimada de PF é refinada, conforme o conhecimento dos requisitos do projeto evolui, em marcos definidos no plano do projeto. Note que não há retrabalho, as contagens anteriores não são descartadas e sim refinadas, conforme os requisitos do projeto evoluem.

Nas subseções seguintes são apresentados os métodos para estimativas de tamanho de software pesquisados na literatura e os métodos criados para integrar o procedimento de estimativas estabelecido para o SERPRO.

3.2.1.1. Método Contagem Indicativa

O Método denominado **Contagem Indicativa** é baseado nos estudos desenvolvidos pela NESMA (*Netherlands Software Metrics Users Association*) [NESMA, 2005]. A Estimativa de tamanho é obtida, por meio da seguinte fórmula:

$$PF = 35 * N^{\circ} ALI + 15 * N^{\circ} AIE$$

Na qual:

PF: Tamanho estimado do projeto de software em **Pontos de Função**

Nº ALI: Número estimado de **Arquivos Lógicos Internos**

Nº AIE: Número Estimado de **Arquivos de Interface Externa**

Um Arquivo Lógico Interno (ALI) é definido como um grupo de dados logicamente relacionados ou de informação de controle², identificado pelo usuário, mantidos dentro da fronteira da aplicação. Um Arquivo de Interface Externa (AIE) é definido como um grupo de dados logicamente relacionados ou informações de controle, identificado pelo usuário, referenciados pela aplicação, mantidos dentro da fronteira de outra aplicação [Hazan, 2000].

Para obtenção das constantes “35” e “15” utilizados na fórmula acima, o método leva em consideração as seguintes premissas [NESMA, 2005]:

- A contagem é baseada no número de Arquivos Lógicos Internos e de Arquivos de Interface Externas.
- A contagem considera a complexidade **média** para os tipos funcionais da APF [IFPUG, 2004];
- Cada Arquivo Lógico Interno - 10 PF possui 3 Entradas Externas (inclusão, alteração, exclusão) - 12 PF, 2 Consultas Externas (consultas aos dados da tabela) - 8 PF e 1 Saída Externa (1 relatório contendo totalizações) - 5 PF; totalizando **35 PFs**;
- Cada Arquivo de Interface Externa - 7 PF possui 2 Consultas Externas (consulta e relatório com os dados da tabela) - 8 PF; totalizando **15 PFs**.

É importante ressaltar que o método Contagem Indicativa foi reconhecido pelo IFPUG em Junho de 2003, como um método potencialmente valioso para a estimativa do tamanho funcional. No entanto, isto não significa que o IFPUG endosse o método como válido, acurado ou preferível em relação a quaisquer outros métodos. A prática tem mostrado que o método tem superestimado o tamanho da maioria dos projetos, em relação a contagem de PF final do projeto implementado.

3.2.1.2. Método Contagem Indicativa Inteligente

A diretriz utilizada na criação do método é a seguinte: *Quanto maior o conhecimento dos requisitos do projeto, maior deve ser a acurácia das estimativas*. A idéia é integrar ao método **Contagem Indicativa** o conhecimento dos requisitos do projeto, gerando premissas que influenciarão nas constantes “35” e “15” do método, apresentado na seção anterior, visando a obtenção de uma estimativa de tamanho com maior acurácia. As contribuições funcionais (número de PFs associados) utilizadas neste método estão baseadas no Manual de Práticas de Contagem (CPM 4.2) [IFPUG, 2004]. Note que muitos sistemas, especialmente aqueles em versões iniciais não apresentarão as funcionalidades de relatório e exclusão para todos ALIs. E ainda, a prática tem demonstrado que a maioria dos ALIs e AIEs são de complexidade Simples, e não Média, conforme preconizado pelo método anterior. Os experimentos realizados pela autora mostraram que na maioria das vezes, o tamanho fica superestimado com a aplicação do método Contagem Indicativa.

² Informação de controle é definida como um dado que influencia um processo elementar (funcionalidade) da aplicação sendo contada. Ou seja, a informação de controle especifica qual, quando, ou como o dado será processado. Por exemplo, um sistema de pagamentos de pessoal deve armazenar informação de tempo que determina quando o processo elementar (funcionalidade) de pagamento de empregados ocorre.

Assim, é fundamental que as fórmulas de estimativas considerem as características de complexidade funcional específicas de cada projeto em questão. As premissas ou suposições utilizadas na geração das estimativas devem ser documentadas, para atender a prática relativa à estimativa de tamanho, incluída na área de processo planejamento do projeto do nível 2 segundo o CMMI.

Por exemplo, suponha que um sistema hipotético **XPTO** a ser desenvolvido, tenha **quatro** tabelas pequenas (menos de 20 campos) mantidas por meio das funções de inclusão, alteração e consulta (usada para a edição dos dados na alteração). Assim, considera-se que o sistema possui 4 ALIs Simples. O sistema também utiliza **uma** tabela de usuário do Sistema de Controle de Acesso, apenas para validação dos dados de acesso. Assim, considera-se que o sistema possui um AIE Simples. Note que o usuário não especificou relatórios nem funções de exclusão para esta *release*. Se utilizarmos o método contagem indicativa, a contagem seria de **155** Pontos de Função, obtidos segundo a fórmula abaixo:

$$PF = 4 \text{ ALIs } \times 35 + 1 \text{ AIE } \times 15$$

O tamanho está superestimado, o sistema **XPTO** é muito simples e não atende as premissas utilizadas na concepção do método Contagem Indicativa, descritas na seção anterior. Então para maior acurácia da estimativa, utiliza-se o conhecimento inicial do projeto para aplicar o método “Contagem Indicativa Inteligente”. Tem-se o seguinte:

Cada ALI Simples (Tabelas pequenas) - 7 PFs possui duas Entradas Externas (inclusão e alteração) de complexidade desconhecida, considera-se médias - 8 PFs (4 x 2) e uma Consulta Externa (recuperação de dados para alteração) de complexidade desconhecida, considera-se média 4 PFs. Note que o índice multiplicador dos ALIs não é mais 35 PF e sim **19** PF (8 + 7 + 4).

O AIE Simples (dados de acesso - *logon* e senha) - 5 PFs com uma Consulta Externa Simples (controle de acesso) com 3 PFs. Note que o índice multiplicador dos AIEs não é mais 15 PF e sim **8** PF (5 + 3).

Então, aplicando o método com novos índices, tem-se a estimativa de **84** Pontos de Função:

$$PF = 4 \text{ ALIs } \times 19 + 1 \text{ AIE } \times 8$$

3.2.1.3. Método de Contagem Estimativa

Este método foi definido inicialmente para atender uma área do SERPRO com muitas demandas de pequenas manutenções evolutivas. Estas demandas, muitas vezes, não possuem Arquivos Lógicos Internos (ALIs) e Arquivos de Interface Externa (AIEs) incluídos, alterados ou excluídos. Suponha um projeto de manutenção para gerar relatórios e gráficos. O tamanho estimado pelo método contagem indicativa de um projeto com zero ALIs e zero AIEs seria **zero Pontos Função**. O impacto seria “zerar” todas as estimativas (custo, esforço e prazo) e inviabilizar o plano do projeto.

O método de contagem estimativa visa aferir o tamanho em Pontos de Função de maneira simplificada, quando o responsável pelo projeto possui algum conhecimento dos requisitos do projeto. A idéia é mapear os requisitos descritos nas propostas comerciais ou Documentos de Visão nos tipos funcionais da Análise de Pontos por Função e estimar a complexidade, baseando-se nas tabelas de complexidade e contribuição funcional [IFPUG, 2004]. Os tipos funcionais são apresentados na **Figura 2**. As necessidades e funcionalidades especificadas para o projeto devem ser enquadradas em uma das seguintes tabelas abaixo:

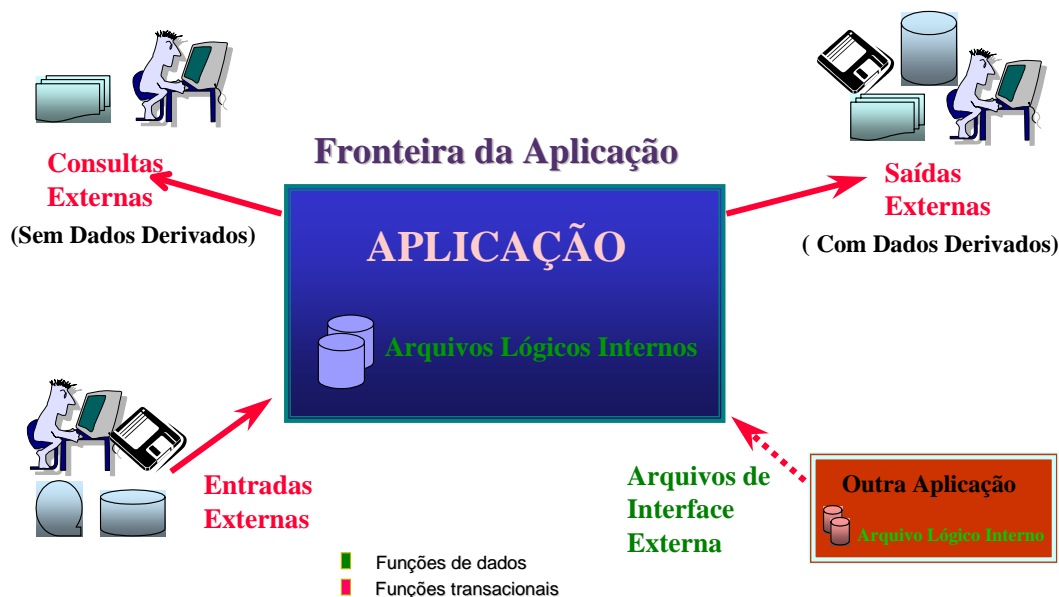


Figura 2: Visão Geral dos Tipos Funcionais da Análise de Pontos por Função [Hazan, 2000]

Tabela A - Contagem dos Arquivos Lógicos Internos (ALIs): Banco de Dados da Aplicação (tabelas e arquivos mantidos pela aplicação)

Arquivos Lógicos Internos (ALI): São grupos lógicos de dados, mantidos por meio de um processo elementar da aplicação. Note que deve-se observar a visão do usuário, não considere arquivos físicos, arquivos de índices, arquivos de trabalho e tabelas de relacionamento sem atributos próprios (tabelas que existem para quebrar o relacionamento nxn e apenas transportam as chaves estrangeiras). As entidades fracas também não são consideradas um ALI [IFPUG, 2004].

Questão 1: Você está alterando a estrutura de tabelas ou arquivos (incluindo, alterando ou excluindo campos)? Você está criando uma nova tabela ou um novo arquivo? Quantos campos possui esta tabela ou arquivo?

| | |
|------------------------------|----------------|
| Nº ALIs Simples: | X 7 PF |
| Nº ALIs Médio: | X 10 PF |
| Nº ALIs Complexo: | X 15 PF |
| Total PF da Tabela A: | |

A experiência tem mostrado que a maioria dos ALIs dos sistemas do SERPRO são **Simple**s.

Tabela B - Contagem de Arquivos de Interface Externa (AIEs): Banco de Dados de outras Aplicações APENAS referenciados pela aplicação que está sendo mantida (tabelas e arquivos mantidos por outra aplicação)

Arquivos de Interface Externa (AIEs): são grupos lógicos de dados, mantidos por outra aplicação, apenas referenciados pelo projeto sendo desenvolvido. Note que deve-se observar a visão do usuário, não são considerados arquivos físicos, arquivos de índice, arquivos de trabalho, tabelas de relacionamento sem atributos próprios e entidades fracas [IFPUG, 2004].

Questão 2: Você está alterando a estrutura de tabelas ou arquivos mantidos por outras aplicações (incluindo, alterando ou excluindo campos)? Você está utilizando uma nova tabela ou um novo arquivo, mantido por outra aplicação? Quantos campos possui esta tabela ou arquivo?

| | |
|------------------------------|----------------|
| Nº AIEs Simples: | X 5 PF |
| Nº AIEs Médio: | X 7PF |
| Nº AIEs Complexo: | X 10 PF |
| Total PF da Tabela B: | |

A experiência tem mostrado que praticamente 100% dos AIEs dos sistemas são **Simples**.

Tabela C - Contagem de Entradas Externas (EEs): Funcionalidades que mantêm arquivos ou tabelas ou alteram o comportamento da aplicação. São considerados os processos elementares de inclusões, alterações, exclusões de dados.

Entradas Externas (EEs) - é um processo elementar que processa dados ou informação de controle que entram pela fronteira da aplicação. O objetivo principal de uma Entrada Externa é manter um ou mais Arquivos Lógicos Internos e/ou alterar o comportamento do sistema. Um processo elementar é definido como a menor unidade de atividade que é significativa para o usuário. O processo elementar deve ser auto-contido e deixar o negócio da aplicação em um estado consistente [IFPUG, 2004].

Questão 3: Você está incluindo, alterando ou excluindo registros de tabelas ou arquivos (*files*)? Quantas funções deste tipo possui o seu projeto (conte separadamente as inclusões, alterações e exclusões de dados) ? Você está entrando com dados que alteram o comportamento da aplicação (exemplo: processamentos *batch*, ou processamento de informações de controle)?

| | |
|------------------------------|---------------|
| Nº EEs Simples: | X 3 PF |
| Nº EEs Média: | X 4 PF |
| Nº EEs Complexa: | X 6 PF |
| Total PF da Tabela C: | |

Caso não haja conhecimento da aplicação de APF ou sobre o processo elementar (funcionalidade analisada), considere as Entradas Externas como **Médias**.

Tabela D - Contagem de Consultas Externas (CEs): Funcionalidades que apresentam informações para o usuário SEM utilização de cálculos ou algoritmos. São os processos elementares do tipo "lê - imprime", "lê - apresenta dados", incluindo consultas, relatórios, geração de disquetes ou CDs, *downloads* ...

Consultas Externas (CEs): é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. O objetivo principal de uma CE é apresentar informação para o usuário através da recuperação de dados ou informação de controle de ALIs ou AIEs. O processamento lógico não contém fórmulas matemáticas ou cálculos, e não cria dados derivados. Nenhum Arquivo Lógico Interno é mantido durante o processamento, nem o comportamento do sistema é alterado [IFPUG, 2004].

Questão 4: Você está desenvolvendo ou alterando uma função para apresentar informações para o usuário: uma consulta, relatório, *browse*, *listbox*, *download*, geração de um arquivo, geração de CD ou de disquete? Esta função **NÃO** possui cálculos ou algoritmos para trabalhar os dados referenciados nem altera um Arquivo Lógico Interno?

| | |
|-------------------------------|---------------|
| Nº CEs Simples: | X 3 PF |
| Nº CEs Média: | X 4PF |
| Nº CEs Complexa: | X 6 PF |
| Total PF da Tabela IV: | |

Caso não haja conhecimento da aplicação de APF ou sobre o processo elementar (funcionalidade analisada), considere as Consultas Externas como **Médias**.

Tabela E - Contagem de Saídas Externas (SEs): Funcionalidades que apresentam informações para o usuário COM utilização de cálculos ou algoritmos. São as consultas ou relatórios com totalização de dados, relatórios estatísticos, gráficos, geração de disquetes com atualização *log* ou índice, *downloads* com cálculo de percentual ...

Saídas Externas (SEs): é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. O objetivo principal de uma Saída Externa é apresentar informação para um usuário através de um processamento lógico adicional a recuperação de dados ou informação de controle. O processamento lógico deve conter no mínimo uma fórmula matemática ou cálculo, ou criar dados derivados. Uma Saída Externa pode também manter um mais Arquivos Lógicos Internos e/ou alterar o comportamento do sistema[IFPUG, 2004].

Questão 5: Você está desenvolvendo uma função para apresentar informações para o usuário: uma consulta, relatório com totalização de dados, etiquetas de código de barras, gráficos, relatórios estatísticos, *download* com percentual calculado, geração de arquivo, geração de CD, geração de disquete com atualização de *log*? Esta função **DEVE** ter cálculos ou algoritmos para trabalhar os dados referenciados nos arquivos lógicos **OU** atualizar campos (normalmente indicadores) nos arquivos?

| | |
|------------------------------|---------------|
| Nº SEs Simples: | X 4 PF |
| Nº SEs Média: | X 5 PF |
| Nº SEs Complexa: | X 7 PF |
| Total PF da Tabela E: | |

Caso não haja conhecimento da aplicação de APF ou sobre o processo elementar (funcionalidade analisada), considere as Saídas Externas como **Médias**.

A Estimativa de tamanho do projeto de desenvolvimento ou manutenção em PFs pelo método empírico deve ser gerada totalizando-se os PFs obtidos nas tabelas **A, B, C e D, E**.

Embora, o método contagem estimativa, que consiste em uma simplificação do método de Contagem de Pontos de Função, tenha sido apresentado inicialmente para as estimativas de manutenção evolutiva, recomenda-se fortemente sua utilização em novos projetos de desenvolvimento visando obter maior acurácia das estimativas.

3.2.1.4. Método de Estimativas Percentuais

O Método foi criado por Capers Jones, baseando-se em uma pesquisa feita em um Banco de Dados Histórico com cerca de 120 projetos da empresa *Software Productivity Research* em 1990 [SPR, 2005]. Foi observada uma correlação entre os tipos de função da aplicação que pode ser expressa da seguinte forma:

- A quantidade de ALIs representa 25% do total de funções de uma aplicação
- A quantidade de AIEs representa 3% do total de funções de uma aplicação
- A quantidade de EEs representa 30% do total de funções de uma aplicação
- A quantidade de SEs representa 28% do total de funções de uma aplicação
- A quantidade de CEs representa 14% do total de funções de uma aplicação

Baseando-se nesta análise estatística, precisa-se identificar apenas a quantidade de um dos tipos de função da aplicação para derivar a quantidade dos outros. A quantidade de um tipo de função deve ser representada sempre em valores inteiros. Para calcular os Pontos de Função Não-Ajustados, utiliza-se complexidade funcional média para os tipos de função (Arquivo Lógico Interno, Arquivo de Interface Externa, Entrada Externa, Saída Externa e Consulta Externa).

Este método presume um relacionamento lógico e previsível entre variáveis independentes, quando na realidade para uma determinado projeto esse relacionamento pode não acontecer. Nesses casos, à medida que aumenta o grau de conhecimento da aplicação, pode-se identificar qual e em que quantidade um determinado tipo de função ficou fora da média [Hazan, 1996]. Na prática, o grau de conhecimento de um projeto na fase de planejamento não é suficiente para este tipo de análise.

Segue um exemplo da aplicação do método. Ao analisar uma determinada aplicação foram identificados 3 Arquivos Lógicos Internos. Utilizando o Método de Estimativa por Percentual pode-se afirmar que:

- Se 3 ALIs = 25 % do TOTAL DE FUNÇÕES, então TOTAL DE FUNÇÕES = **12**
- Se AIEs = 3% do TOTAL DE FUNÇÕES (12) , então AIEs = 0,36 (arredonda-se para 0)
- Se EEs = 30 % do TOTAL DE FUNÇÕES (12) , então EEs = 3,6 (arredonda-se para 4)
- Se SEs = 28 % do TOTAL DE FUNÇÕES (12) , então SEs = 3,36 (arredonda-se para 3)
- Se CEs = 14% do TOTAL DE FUNÇÕES (12), então CEs = 1,68 (arredonda-se para 2)

A partir desses valores calcula-se os PFs Não-Ajustados da aplicação, conforme o seguinte:

| TIPO DE FUNÇÃO | COMPLEXIDADE FUNCIONAL |
|---|------------------------|
| ALIs | 3 MÉDIA X 10 = 30 |
| AIEs | 0 MÉDIA X 7 = 0 |
| EEs | 4 MÉDIA X 4 = 16 |
| SEs | 3 MÉDIA X 5 = 15 |
| CEs | 2 MÉDIA X 4 = 8 |
| TOTAL DE PS DE FUNÇÃO NÃO AJUSTADOS = 69 | |

Em 2000, o ISBSG [ISBSG, 2003] publicou uma nova tabela de percentuais por tipo de projeto desenvolvimento, redesenvolvimento e manutenções evolutivas, baseado em seu Banco de Dados Histórico com milhares de projetos. Em Junho de 2003, o ISBSG publicou uma nova tabela substituindo a anterior apenas para projetos de novos desenvolvimento. Recomenda-se a utilização das tabelas do ISBSG por serem dados de projetos mais atuais (**Figura 3**).

Note que no sistema exemplo, tem-se **80 PFs** estimados pelos dados do ISBSG de 2000 e **76 PFs** estimados pelos dados do ISBSG de 2003 e **105 PFs** estimados pelo NESMA. Este sistema hipotético utilizado como exemplo nos cursos de Análise de Pontos de Função possui **84 Pontos de Função**.

| Estudo do ISBSG (Data 6 /2000) | | | |
|---------------------------------------|-----------------------------|---------------------------|------------------|
| Objeto | Novo Desenvolvimento | Re-desenvolvimento | Melhorias |
| ALI | 22,00% | 23,00% | 15,60% |
| AIE | 5,00% | 2,50% | 6,50% |
| EE | 33,50% | 52,90% | 32,70% |
| SE | 23,50% | 17,50% | 32,00% |
| CE | 16,00% | 4,10% | 13,20% |

| NOVO DESENVOLVIMENTO (ISBSG Junho/2003) | |
|--|------------------------------|
| FUNÇÕES DE DADOS | FUNÇÕES TRANSACIONAIS |
| ALI = 23.5% | EE = 29.3% |
| AIE = 8.1% | SE = 24.3% |
| | CE = 14.8% |

Figura 3: Tabelas de Estimativas Percentuais do ISBSG [ISBSG, 2003]

O Método Estimativas Percentuais deve ser aplicado somente quando se tem pouco conhecimento sobre o projeto, inviabilizando a aplicação dos métodos de Contagem Estimativa e de Contagem Indicativa. O método de Estimativas Percentuais não possui acurácia adequada, devido à premissa utilizada de que todas as aplicações possuem a mesma distribuição de funcionalidades associadas aos tipos funcionais (ALI, AIE, EE, CE, SE). Na prática, este método foi utilizado pela autora em apenas um projeto quando o conhecimento era a quantidade de Casos de Uso, por exemplo 10 Casos de Uso. Segundo os desenvolvedores, cada Caso de Uso teria 4 Entradas Externas (Inclusão, Alteração, Exclusão Lógica, Reinclusão). Assim, tem-se o total de 40 Entradas Externas e então o método pode ser aplicado. A complexidade de cada tipo funcional foi determinada baseando-se no conhecimento do projeto, por exemplo ALIs e AIEs Simples, EEs Complexas, CEs e SEs Médias. A acurácia da estimativa comparando-se com o realizado não foi adequada.

3.2.2. Método para Geração de Estimativas de Esforço

A estimativa de esforço deve ser derivada da estimativa de tamanho, segundo o CMMI. Nesta seção é descrito um método para derivar as estimativas de esforço a partir das estimativas de tamanho. As Estimativas de esforço são geradas em homem_horas (HH) por um modelo de estimativas simplificado, baseando-se apenas em dois parâmetros: estimativa de tamanho em PF e fator de conversão HH/PF.

É importante ressaltar que a produtividade considerada no fator de conversão (HH/PF), depende de diversos atributos do projeto, dentre outros: tamanho, plataforma de desenvolvimento, complexidade da aplicação e experiência da equipe. O fator de conversão precisa ser estabelecido de acordo com as características de cada projeto.

Assim, para obtenção de estimativa de esforço e de prazo mais precisa sugere-se a construção de um banco de dados histórico de projetos com dados estratificados, de acordo com os atributos do projeto que a organização considere relevante no seu contexto.

Para facilitar a geração de estimativas de esforço, nesta versão do procedimento de estimativas, o método utiliza como insumo seguintes parâmetros:

- **Estimativa de tamanho do projeto:** Tamanho estimado do projeto em Pontos de Função.
- **Fator de Conversão HH/PF:** Este parâmetro é a quantidade de horas para se implantar um Ponto de Função. Este fator deve ser recuperado de dados históricos de projetos similares ao projeto em questão.

Para os projetos nos quais não há dados históricos de projetos similares disponíveis sugere-se utilizar dados de produtividade pesquisados por meio de *benchmarking*. A **Tabela I** apresenta alguns dados pesquisados na tabela de nível de linguagem da empresa *Software Productivity Research* [SPR, 2005] e *benchmarking* com outras empresas do governo: Caixa Econômica Federal, DATAPREV e CELEPAR.

A **Tabela I** contém fatores de conversão HH classificados por linguagem de programação e três níveis de produtividade: Baixa, Média, Alta. O nível de produtividade deve ser definido baseando-se na complexidade da especificação, experiência da equipe de desenvolvimento e outros fatores que influenciem a produtividade.

TABELA I: Produtividade em Pontos de Função/hora por linguagem

| Ambiente/Linguagem | Índice Produtividade (horas/PF) | | |
|---------------------------------|---------------------------------|--------|--------|
| | Baixa | Média | Alta |
| Mainframe | | | |
| COBOL | 26,4 h | 17,6 h | 13,2 h |
| NATURAL | 13,2 h | 8,8 h | 6,6 h |
| Micro e Cliente/Servidor | | | |
| Visual Basic | 8,8 h | 6,8 h | 5,7 h |
| Delphi | 8,8 h | 6,8 h | 5,7 h |
| ORACLE | 13,2 h | 8,8 h | 6,6 h |
| WEB/Documentos | | | |
| ASP | 12h | 10 h | 8 h |
| Java | 17 h | 15 h | 12 h |
| Lotus Notes | 5,5 | 3,9 | 3,1 |

Note que o nível de **produtividade Baixa** está associado, dentre outros fatores, a: **aplicações complexas** ou a **equipes inexperientes na plataforma de desenvolvimento**. Já o nível de **produtividade Alta** está associado a **aplicações simples**, ou a **equipes experientes na plataforma**. O nível de produtividade da equipe influencia diretamente nas estimativas de prazo, a produtividade Alta implica em prazos de desenvolvimento menores. Como esta tabela não está baseada em projetos reais do SERPRO, os setores que já possuem um pequeno histórico de projetos concluídos têm utilizado os seus próprios índices de produtividade.

Caso, a linguagem de desenvolvimento utilizada não seja encontrada na tabela de linguagens, seleciona-se uma linguagem com produtividade similar. Caso o projeto seja desenvolvido em mais de uma linguagem de programação, define-se um **índice de produtividade - horas/PF**, contemplando o percentual de cada linguagem.

Por exemplo, suponha um projeto de baixa complexidade, desenvolvido por uma equipe experiente na plataforma e na aplicação, utilizando as linguagens NATURAL e COBOL, sendo 20% COBOL e 80% NATURAL. Consultando a tabela I tem-se as seguintes produtividades altas: COBOL -13,2 HH/PF e NATURAL 6,6 HH/PF. Então, para o projeto em questão, temos o seguinte fator de conversão (HH/PF) para a geração da estimativa de esforço: $0,2 \times 13,2 + 0,8 \times 6,6 = 7,92$. As linguagens utilizadas e os percentuais devem documentadas em **Premissas**.

É importante ressaltar que a **Tabela I** está sendo apresentada para ilustrar o método. A produtividade por linguagem pode ser bastante variável, por exemplo segundo o banco de dados do ISBSG [ISBSG, 2003], a produtividade do JAVA varia de 10 h/PF a 60 h/PF, devido às características específicas de cada projeto de software. O processo de desenvolvimento de software utilizado na organização também influencia a produtividade. Portanto, a acurácia das estimativas de esforço é fortemente dependente da existência de um banco de dados histórico, contendo dados de esforço, tamanho e atributos de produtividade de projetos concluídos da organização. O próximo passo é o estudo de fatores que influenciam na produtividade para a estratificação dos dados históricos de projetos a serem coletados. A construção do banco de dados histórico deve levar em consideração poucos atributos relevantes sobre o ponto de vista de influência na produtividade. Tem-se realizado pesquisas com as equipes de desenvolvimento nos cursos de estimativas, buscando identificar tais atributos.

A **Tabela II** apresenta os atributos mais relevantes identificados pelos participantes de cinco turmas do treinamento "Técnicas de Estimativas" ministradas nas Regionais do SERPRO. Cada turma teve uma média de 30 colaboradores. A pergunta colocada foi a seguinte: *Quais os principais atributos exercem impacto nas estimativas de esforço e prazo?* É importante destacar que os cargos e níveis de experiência dos participantes foi bastante diversificado, alguns participantes possuem mais de 30 anos de empresa, outros estão no SERPRO há somente 3 anos. Os participantes realizaram o exercício em grupos de 2 a 4 pessoas e debateram os resultados com os colegas de outros grupos e a instrutora.

Os parâmetros utilizados para derivação das estimativas de esforço descritos no Modelo CMMI [SEI, 2002] incluem o seguinte:

- Riscos
- Competências e papéis críticos utilizados para executar o trabalho
- Níveis de habilidade de gerentes e equipes necessárias para realizar o trabalho
- Necessidades de habilidades, conhecimento e treinamento
- Necessidades de facilidades (ex.: espaço do escritório e reuniões, estações de trabalho)
- Trabalho direto e *overhead*
- Requisitos para o produto e componentes do produto
- Estimativas de tamanho de artefatos e mudanças antecipadas
- Modelo do ciclo de vida do projeto e processos selecionados
- Abordagem técnica
- Capacidades e ferramentas fornecidas no ambiente de engenharia
- Necessidades de facilidades de Engenharia

- Nível de segurança requerida para tarefas, artefatos, *hardware*, *software*, pessoal e ambiente de trabalho

TABELA II: Atributos que Influenciam as Estimativas de Custo, Esforço e Cronograma

| Atributos | Turma 1 | Turma 2 | Turma 3 | Turma 4 | Turma 5 |
|---|----------------|----------------|----------------|----------------|----------------|
| Tamanho (PF) | X | X | X | X | X |
| Experiência da Equipe: plataforma, negócio, processo de desenvolvimento, ferramentas e projetos similares | X | X | X | X | X |
| Motivação da equipe | X | X | X | X | X |
| Rotatividade de Pessoas | | | | X | |
| Integração (relacionamento) da Equipe | X | X | | | |
| Habilidade dos gestores | | X | X | X | |
| Disponibilidade da equipe | | | X | X | |
| Participação de profissionais em mais de 2 projetos simultaneamente | | | X | | |
| Tamanho da equipe de desenvolvimento | X | X | X | | X |
| Disponibilidade de infra-estrutura (suporte, rede, ferramentas, hardware, software) | X | X | X | X | X |
| Utilização de ferramentas automáticas | | | X | | |
| Reutilização de Código e Componentes | X | | X | | X |
| Plataforma de Desenvolvimento | X | X | X | X | X |
| Desenvolvimento de componentes reusáveis | X | | | | |
| Ambiente de Trabalho | | X | | | |
| Processo de Desenvolvimento utilizado | X | | | X | X |
| Complexidade da demanda (projeto) | X | X | X | X | X |
| Clareza e Qualidade da especificação | X | X | X | X | X |
| Estabilidade de Requisitos | X | | | X | X |
| Utilização de novas tecnologias (ex: JAVA) | | | X | | |
| Imposição de prazo pelo cliente | | | X | | X |
| Relacionamento e Parceria com o Cliente | X | X | X | X | X |
| Conhecimento do negócio pelo cliente | X | X | X | X | |

Na área de Engenharia de Software têm sido desenvolvidos muitos modelos parametrizados para apoiar as estimativas de custo, esforço e cronograma. O uso destes modelos como fonte única de estimativa não é recomendado, porque estes modelos são baseados em dados de projetos históricos que podem não ser similares aos projetos da sua organização. Diversos modelos e/ou métodos combinados podem ser usados para assegurar uma melhor acurácia e conseqüentemente uma maior confiança na estimativa [Boehm, 2000].

3.2.3. Método para Estimativas de Prazo

Após estimar-se o esforço necessário para o desenvolvimento do projeto, o próximo passo é a derivação da estimativa de prazo. A estimativa de prazo é gerada inicialmente de forma empírica considerando os insumos: tamanho e esforço estimados do projeto e tamanho da equipe para o desenvolvimento do projeto.

Algumas considerações foram feitas para as estimativas de prazo em **horas** e em **meses**:

- 1 mês possui **22 dias úteis**;
- Em uma jornada de trabalho de 8 horas/dia, a produtividade média do profissional no Brasil é de **6 horas/dia** [Jones, 1997].

Então, utiliza-se a seguinte fórmula:

$$\text{Prazo (em dias)} = \text{Esforço (horas)} / (\text{Tam. equipe} * 6)$$

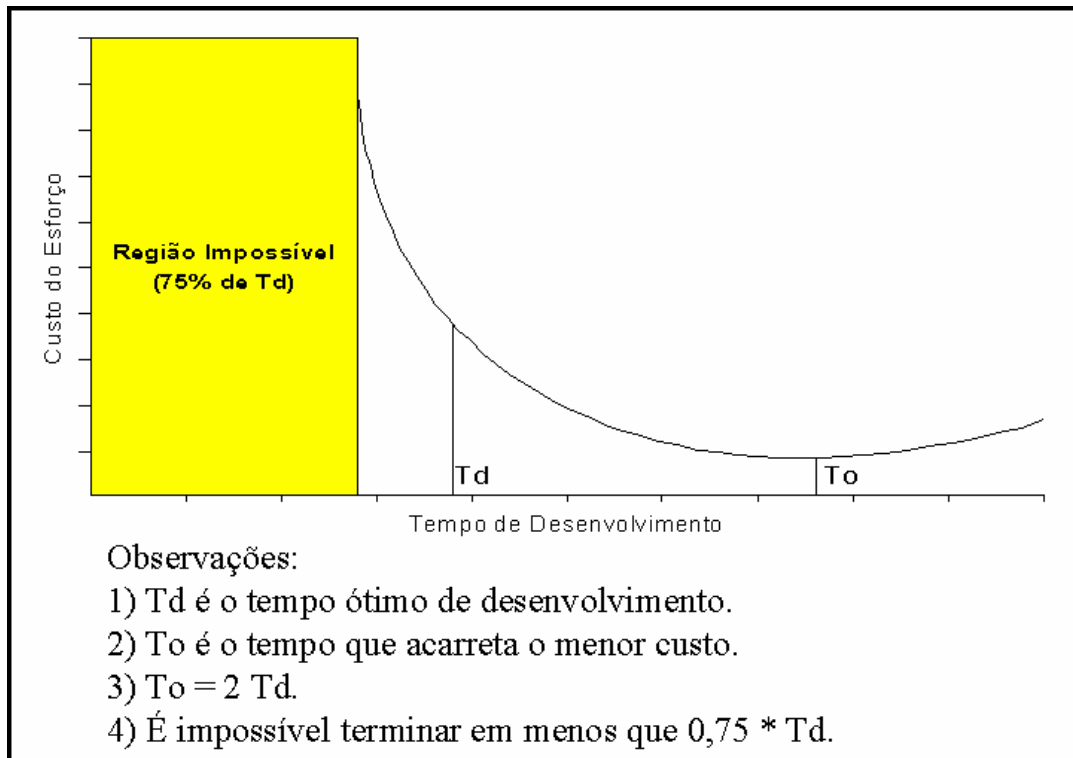


Figura 4: Relação entre o esforço e o prazo [PUT92]

As estimativas de prazo não são lineares com o tamanho, assim pretende-se pesquisar mais sobre o melhor tempo desenvolvimento (onde há uma melhor relação custo x benefício de alocação de recursos e menor prazo de desenvolvimento), dado o tamanho de um projeto específico. Jones propõe uma fórmula para o cálculo do melhor tempo de desenvolvimento, denominado Td e de Região Impossível (RI) de desenvolvimento (**Figura 4**), descrito em [Jones, 1998]. Na Região Impossível (RI), a adição de mais recursos ao projeto não implicará em redução no prazo. O cálculo da região impossível já está sendo utilizado para apoiar a confecção de cronogramas. Note que a curva (**Figura 4**) mostra que quanto menor o prazo almejado para a conclusão do projeto, maior será o esforço requerido [PUT92]. O aumento do esforço para reduzir o prazo acontece através da realização de horas-extras e da inclusão de pessoal adicional. A redução de prazo tem um limite, como demonstra a região impossível da **Figura 4**.

O método ainda não considera o cálculo da Região Impossível. O método é utilizado apenas para se ter uma idéia inicial da viabilidade do projeto. O cronograma do projeto deve ser construído baseando-se nas atividades do projeto e na disponibilidade dos papéis para execução delas. As ferramentas de gestão de projetos, tais como MS Project, têm sido utilizadas para apoiar na confecção do cronograma.

3.2.4. Método para Estimativas de Custo

Segundo o modelo CMM, as estimativas de custo devem ser derivadas da estimativa de tamanho. Assim, de posse do tamanho da aplicação em Pontos de Função, do esforço e do tamanho da equipe, o próximo passo é a geração da estimativa de custo. Neste procedimento, considera-se os seguintes atributos para a aferição da estimativa de custo: estimativa de mão de obra (considerando o custo por perfil do profissional), recursos computacionais, treinamento, consultoria, viagens e custos indiretos.

É importante destacar que as estimativas de custos de mão de obra dos sistemas terceirizados são aferidas com base no número de Pontos de Função estimados e o preço por Ponto de Função contido no contrato firmado entre a empresa contratante e a empresa contratada. Nos sistemas desenvolvidos por profissionais da empresa, o custo é estimado considerando-se o esforço e o custo dos profissionais alocados para o desenvolvimento do projeto. É importante destacar que o custo por Ponto de Função pode ser bastante variável, o esforço (HH/PF) e outros custos indiretos influenciam fortemente no custo/PF.

3.2.5. Estimativas de Recursos Computacionais Críticos

Além das estimativas de tamanho, esforço, prazo e custo, o modelo CMMI também pede que as estimativas de Recursos Computacionais Críticos sejam documentadas. Visando a padronização da documentação destas estimativas, sugere-se a utilização da planilha, ilustrada na **Figura 5**.

Deve-se considerar recursos computacionais críticos dos ambientes de: desenvolvimento, teste e produção. Exemplos de recursos computacionais críticos incluem o seguinte: memória, espaço em disco capacidade da rede, potência de processadores, potência das estações de trabalho e capacidade dos periféricos [SEI, 2002].

| Nome do Recurso Computacional Crítico | Descrição | Responsável pela Disponibilização | Data Limite | Parâmetros | (*) D/P/H | Custo (OPCIONAL) |
|---------------------------------------|-----------|-----------------------------------|-------------|------------|-----------|------------------|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| Total | | | | | | |

Nome do Recurso, hardware: micro, periférico, expansão de memória, área em disco, banda de rede, etc
Parâmetros: Características do recurso, como: quantidade, perfil, configuração, etc
(*) D:Recurso para ambiente de desenvolvimento; **P**: recurso para ambiente de produção; **H**: recurso para amb. Homologação
Custo: Preencher este campo quando for possível a definição dos custos envolvidos com o recurso computacional

Figura 5: Modelo para Estimativas de Recursos Computacionais Críticos

4. Discussão e contribuições

Alguns experimentos iniciais têm sido realizados com os métodos de estimativas de tamanho propostos. Os métodos têm demonstrado uma boa precisão com menos de 10% de erro, sem levar em consideração as mudanças de requisitos no decorrer do desenvolvimento. Inicialmente, recomendou-se a utilização do método Contagem Indicativa (NESMA) devido a sua simplicidade, no entanto na prática as estimativas de tamanho, na maioria das vezes ficaram muito acima do tamanho real. Isto ocorreu por causa das premissas utilizadas na concepção do método, considerando a complexidade média para as funções de dados, que costumam ser simples; e ainda duas consultas externas para os Arquivos de Interface Externa, que costumam ser usados apenas para validação de dados, sem nenhuma funcionalidade de consulta específica. Alguns sistemas de consultas gerenciais possuem funções de consulta aos AIEs, no entanto estas consultas gerenciais, normalmente são Saídas Externas e também a fórmula original da Contagem Indicativa não pode ser aplicada. Assim, o método “Contagem Indicativa Inteligente” criado tem se mostrado adequado para as contagens rápidas, exceto para os casos de manutenções evolutivas, onde não há arquivos lógicos incluídos ou com a estrutura alterada. Nestes casos, recomenda-se a utilização do Método de Contagem Estimativa. O método de estimativas percentuais pode distorcer bastante a contagem se o projeto a ser dimensionado não se encaixar no modelo (premissas) proposto pelo método. Embora, a literatura proponha a calibração dos percentuais para o projeto a ser estimado [UNISYS, 1995]. Na prática, isto torna-se extremamente complicado para ser realizado no início do projeto, quando há pouco conhecimento sobre o mesmo. Assim, devido a subjetividade do método, este não foi incorporado no procedimento de estimativas do SERPRO. O método de Contagem Estimativa tem sido bastante utilizado nas estimativas. Além deste, proporcionar uma melhor acurácia nas estimativas, este facilita as reestimativas de tamanho que devem ocorrer no acompanhamento do projeto.

Outro ponto a ser considerado nas Estimativas de Tamanho é o Fator de Ajuste da Contagem de Pontos de Função que influencia os Pontos por Função não Ajustados em +/- 35%. O valor do fator de ajuste, que varia de 0,65 a 1,35, indica a funcionalidade geral fornecida para o usuário da aplicação, considerando as 14 Características Gerais do Sistema (CGS) que estimam a magnitude das funcionalidades gerais da aplicação. Cada característica possui descrições associadas que ajudam a determinar o nível de influência da característica. Os níveis de influência estão definidos em uma escala de 0 (nenhuma influência) a 5 (forte influência) [Hazan, 2000]. Embora o fator de ajuste não tenha atualizações relevantes desde a década de 80, não seja reconhecido pela ISO [Dekkers, 2003] e algumas empresas tenham desconsiderado o fator de ajuste nas contagens. A opção do SERPRO foi seguir o manual CPM 4.2 [IFPUG, 2004], considerando o cálculo fator de ajuste também nas estimativas de tamanho. Assim, a empresa está utilizando os Pontos de Função Ajustados nas estimativas de tamanho desde a etapa de planejamento. A experiência tem demonstrado que sistemas *batch* ficam com um Fator de Ajuste Baixo (entre 0,8 e 0,9), em Sistemas *mainframe* on-line ou *micro stand alone* o fator de ajuste fica em torno 1 não exercendo impacto na contagem dos Pontos de Função não ajustados, os Sistemas Cliente Servidor e Web possuem fator de ajuste mais altos (entre 1,1 a 1,2). Alguns sistemas Cliente Servidor mais complexos tiveram fator de ajuste maior que 1,25. O AulaNet [EDUWEB, 2003], ambiente de software baseado na Web, desenvolvido Laboratório de Engenharia de Software - LES - do Departamento de Informática da PUC-Rio tem um fator de ajuste de 1,27. O Fator de Ajuste do AulaNet foi aferido pela autora, especialista em contagem de Pontos de Função, por meio de entrevistas com a equipe de desenvolvimento da aplicação. Também é importante destacar que a aferição do fator de ajuste tem apoiado a elicitação e a análise (revisão) de requisitos não-funcionais e até mesmo de requisitos funcionais.

As estimativas de esforço têm sua precisão dependente da existência de dados históricos de projetos similares. Neste momento, o SERPRO está em fase de construção de um Banco de Dados Histórico de produtividade. O Banco de Dados histórico será construído partir da contagem de PF de projetos concluídos e obtenção de dados de esforço dos projetos em questão. A estratégia da empresa é estratificar os projetos, inicialmente por linguagem de desenvolvimento, tipo do projeto e experiência da equipe. Também pretende-se realizar Benchmarking no ISBSG (*International Software Benchmarking Standards Group*) [ISBSG, 2003] para comparar a produtividade do SERPRO com o mundo. O modelo de estimativa de esforço apresentado, que multiplica o tamanho de projeto em Pontos de Função pela produtividade (número de horas para produzir 1 Ponto de Função) é denominado Modelo simplificado de estimativas [Vazquez, 2004]. Este modelo tem se apresentado adequado. No entanto, com a construção do Banco de Dados Histórico, pretende-se utilizá-lo na calibragem dos parâmetros do COCOMO II [Boehm, 2000]. A idéia é comparar as estimativas geradas pelo modelo simplificado e pelo COCOMO, visando apoiar os líderes de projetos na confecção de cronogramas mais realistas especialmente para os projetos dos novos clientes, onde a experiência não pode ajudar.

As estimativas de prazo não são lineares com o tamanho, assim pretende-se pesquisar mais sobre o melhor tempo desenvolvimento, dado o tamanho de um projeto específico. Jones propõe um método para cálculo de Tempo de Ótimo de Desenvolvimento e de Região Impossível de desenvolvimento, descrito em [Jones, 1998]. Este método foi implantado na planilha de documentação de estimativas do SERPRO apenas para efeito de análise. Este modelo pode não ser apropriado para a organização, devido a este método ser bastante empírico e não ter sido calibrado utilizando dados históricos de projetos da organização. É importante destacar que o COCOMO II [Boehm, 2000] também propõe fórmulas para o cálculo da Região Impossível (RI) e do Tempo Ótimo de Desenvolvimento (Td), baseando-se no tamanho do projeto e de outros parâmetros. Estes parâmetros referem-se à calibragem, que deve ser realizada com dados históricos de projetos da organização, referentes ao contexto para o qual as estimativas serão elaboradas e ao mapeamento do modelo, que leva em consideração a seqüência de atividades pertinentes ao ciclo de vida do projeto, as características do produto, o processo de desenvolvimento, características da equipe de desenvolvimento e outros fatores relativos ao projeto a ser estimado [Aguiar, 2004].

Na subseção seguinte é apresentada uma proposta de integração dos métodos de estimativas propostos de acordo com o CMMI no processo de planejamento de projetos (*Planning Game*) do Extreme Programming (XP).

4.1. Integrando o Processo de Estimativas Proposto com o XP

O planejamento em Extreme Programming (XP) focaliza em como uma equipe de desenvolvedores pode ser otimamente conduzida. Assim, o planejamento é realizado para assegurar que a equipe está sempre trabalhando na coisa mais importante que precisa ser feita; coordenar efetivamente com outras pessoas e responder rapidamente aos eventos inesperados. Os planos são baseados em estimativas feitas por indivíduos. Quando eventos inesperados ocorrem, os planos devem ser revisados. Qualquer técnica de planejamento deve fornecer visibilidade, tal que qualquer pessoa envolvida no projeto possa realmente ver quanto falta e quanto já foi feito do projeto. Isto significa que precisa-se criar marcos de acompanhamento que representem claramente o progresso do projeto [Beck, 2001].

O propósito do plano inicial é verificar se o projeto faz sentido como um todo. Normalmente, estes planos (estudo de viabilidade) são feitos antes de envolver o pessoal técnico. Para elaboração do plano são necessárias estimativas de tempo (cronograma) e de custos (orçamento). O XP sugere que não sejam gastas muitas horas no primeiro plano.

O processo de planejamento XP confia claramente na separação de papéis de pessoas do negócio e pessoas de software. Isto assegura que as pessoas do negócio (analistas de negócios) sejam responsáveis pelas decisões de negócio e as pessoas técnicas (programadores) tomem as decisões técnicas. A estimativa é uma decisão técnica que deve ser realizada pelos programadores. A escolha de prioridade entre as funcionalidades e características não funcionais é uma decisão de negócio realizada pelas pessoas de negócio que conhecem o mercado e podem utilizar a experiência em aplicações similares. Note que o planejamento constitui uma maneira de descobrir quais requisitos (ou *stories*) devem ser construídos em cada iteração. Uma *story* é a unidade (pedaço) de funcionalidade em um projeto XP. A definição de uma *story* requer uma comunicação eficaz entre clientes e desenvolvedores. Os clientes definem a *story* e os desenvolvedores estimam a *story*. As estimativas devem ser realizadas no início do projeto, isto é assim que se inicia a redação da *story*. Existem três abordagens-chaves para uma estimativa efetiva:

- Mantenha simples
- Utilize o que aconteceu no passado
- Aprenda com a experiência

O desenvolvedor deve ser capaz de estimar quanto tempo levará para construir a *story* em questão. Beck [Beck, 2001] sugere uma maneira simples para estimar-se o tamanho de uma *story*, a saber :

- Procurar uma *story* similar que tenha sido entregue;
- Assumir que a nova *story* consumirá a mesma quantidade de esforço;
- Se não achar um *story* similar, então procurar uma *story* duas vezes maior ou menor, e então dividir ou multiplicar o esforço.

Assim, o seguinte é preconizado: “Você apenas pode estimar da experiência. Você tem experiência? Convide um amigo que tenha, chame um programador para conversar”. Note que o processo apresentado é fortemente baseado na experiência em projetos anteriores. A equipe de desenvolvimento discute cada *story*, considera quanto tempo leva para implementar, e decide sobre a estimativa. Assim, caso a equipe não possua experiência em projetos similares, esta não possuirá suporte científico para negociar o tamanho máximo de uma *story* possível de ser construída em uma iteração.

Segundo Beck [Beck, 2001], as duas principais áreas de incerteza consideradas no desenvolvimento do plano do projeto são as seguintes: velocidade da equipe e tamanho das *stories*. Assim, sugere-se a utilização da métrica de Pontos de Função (PF) e do Método de Contagem Estimativa, descrito em 3.2.1.3 para identificação do tamanho da *story* na fase de planejamento do projeto. A velocidade da equipe de desenvolvimento apenas poderá ser identificada com acurácia, caso exista um banco de dados histórico com projetos similares. No entanto, as fórmulas para o cálculo de região de impossível, descritas na literatura [Jones, 1998] e [Putnam, 1992], podem ajudar na definição do tamanho ideal da *story* (em Pontos de Função) possível de ser construída em cada iteração.

O XP, assim como o CMM, pressupõe a utilização do plano do projeto para acompanhar e avaliar o progresso, que é demonstrado por meio da entrega de código integrado e testado que implementa uma *story*. O processo de planejamento do XP também considera as reestimativas periódicas das *stories* para incorporar informações adicionais como dependências entre as *stories* ou tecnologia [Beck, 2001]. Portanto, torna-se importante um processo simplificado de estimativas – “ágil” para apoiar o processo de planejamento do XP.

5. Estudo de Caso

Esta seção tem como propósito apresentar um estudo de caso utilizando os métodos de estimativas propostos na seção 3. Com a finalidade de apresentar o procedimento de estimativas de forma simplificada, esta seção descreve um sistema hipotético (parte de um sistema real).

Suponha que o setor de treinamento de uma empresa solicitou um sistema, denominado STREINA com as necessidades e funcionalidades, descritas na **Tabela III**. Note que é necessária uma análise dos requisitos do projeto documentados no Documento de Visão antes da aplicação do Método de Contagem Estimativa, visando uma melhor acurácia das estimativas. Na prática, o analista de negócios é entrevistado pelo profissional responsável pelas estimativas.

TABELA III: Funcionalidades do Sistema de Treinamentos (STREINA)

| Necessidade 1 | | Benefício |
|------------------------|---|------------------|
| Controlar capacitações | | Crítico |
| Id Func. | Descrição das Funcionalidades/atores envolvidos | |
| F 1.1 | Cadastrar evento de capacitação Técnico | |
| F1.2 | Planejar evento de capacitação Técnico | |
| F 1.3 | Planejar cronograma de capacitação Técnico | |
| F 1.4 | Consultar cronograma de capacitação Técnico | |
| F 1.5 | Consultar eventos de capacitação por data e local Técnico | |
| F 1.6 | Consultar detalhes de um evento de capacitação Técnico | |
| F 1.7 | Informar resultado da avaliação de um evento de capacitação Técnico | |
| F 1.8 | Informar acompanhamento de comunidade capacitada (avaliação pós-treinamento) Participantes do treinamento | |
| F 1.9 | Consultar acompanhamento de comunidade capacitada Técnico | |
| F 1.10 | Gerar Consultas Gerenciais (Gráficos e Relatórios) de Capacitação Gerente | |

Note que algumas funcionalidades que não aparecem explicitamente no Documento de Visão , cujas funcionalidades estão representadas na **Tabela III**, por exemplo como Controle de Acesso serão consideradas na estimativa.

Conforme descrito no Procedimento de Estimativas, o primeiro passo é estimar-se o tamanho do projeto. Vamos utilizar o método de contagem estimativa.

- ALI: Usuários - Simples - 7 PF

- 3 EEs: Cadastro Usuários (Inclusão, Alteração e Exclusão) – Simples – 9 PF
- 2 CEs: Lista de Usuários, Consultar Usuários – Simples – 6 PF
- EE: Alterar Senha – Simples – 3 PF
- ALI: Capacitação – Médio – 10 PF
- 2 EEs: Cadastrar evento de Capacitação (Inclusão e Alteração)– Média – 8 PF
- CE: Lista de Eventos Capacitação – Simples – 3 PF
- SE: Consultar evento de capacitação – Média – 5 PF
- 2 EEs Planejar Evento de Capacitação (Planejar e Alterar Plano) – Média – 8 PF
- CE: Consultar Plano Evento Capacitação – Média – 4 PF
- 2 EEs Planejar Cronograma de Capacitação (Criar e Alterar Cronograma)– Média – 8 PF
- SE: Consultar Cronograma Evento Capacitação – Média – 5 PF
- SE: Consultar eventos de capacitação por data e local – Média – 5 PF
- CE: Consultar Detalhes de Evento de Capacitação – Complexa – 6 PF
- 3 EEs: Cadastrar participantes no evento: (Inclusão, Alteração/Exclusão) - 2Médias/1 Simples – 11 PF
- CE: Consultar Participantes cadastrados no evento – Média – 4 PF
- SE: Enviar para e-mail para participação do evento – Média – 5 PF
- 2 EEs: Informar resultados da avaliação de um evento de capacitação – Simples – 6 PF
- 5 SEs: Consulta Avaliação de um evento de capacitação (1 relatório e 4 gráficos) – Simples – 20 PF
- SE: Enviar e-mail de notificação para avaliação do evento – Média – 5 PF
- 2 EE: Informar acompanhamento de comunidade capacitada (Informar e alterar dados de acompanhamento) – Média - 8 PF
- CE: Consultar dados de acompanhamento inseridos - Média – 4 PF
- SE: Consultar Consolidada acompanhamento comunidade capacitada – Simples – 4 PF
- SE: Consultas Gerenciais (2 gráficos e 2 relatórios) – Simples – 16 PF

Totalizando o número de PFs, tem-se 170 PFs não ajustados. Suponha que o fator de ajuste deste projeto seja de 1,10. Assim, temos 187 Pontos de Função Ajustados estimados.

De posse da estimativa de tamanho, procede-se com estimativa de esforço. Este projeto será desenvolvido em JAVA por equipe experiente na plataforma. Como o projeto é de complexidade simples e pequeno (< 200 PFs), será utilizada a produtividade 12 horas/PF. Assim, a estimativa de esforço é de 2244 HH (homens_hora).

O próximo passo é a estimativa de prazo, considerando-se a produtividade de 6 h/dia, o mês com 22 dias úteis e uma equipe com 3 recursos (incluindo desenvolvimento e gestão) tem-se: 5,7 meses estimados para realização do projeto. Como 10% do projeto já foi realizado (modelagem do negócio), então o prazo restante é de 5,2 meses.

A estimativa de custo deve considerar o valor da hora dos profissionais alocados ao projeto, bem como outros custos de ambiente, ferramentas, deslocamentos, consultoria, etc.

A estimativa de recursos computacionais críticos em projeto WEB deve considerar a disponibilidade dos servidores utilizados para desenvolvimento, homologação e produção do sistema e outros recursos de Hardware relevantes.

6. Conclusões e Recomendações para Trabalhos Futuros

Com a adoção de modelos de Qualidade de Software, tais como CMM ou CMMI, torna-se fundamental a existência de um procedimento para a geração e documentação das estimativas a serem utilizadas no acompanhamento do projeto de software. Os métodos descritos neste trabalho foram definidos em 2002, no entanto sua utilização efetiva iniciou-se em junho de 2003 após a realização dos treinamentos e *mentoring*. Os experimentos realizados com os métodos de estimativas de tamanho têm demonstrado uma acurácia razoável, menos de 10 % de erro, sem levar em consideração as mudanças significativas de requisitos no decorrer do desenvolvimento, as quais acarretam em uma reestimativa do projeto, conforme preconizado no CMMI.

A definição do procedimento de estimativas, contendo os métodos descritos neste trabalho foi um passo importante na implantação da área chave de planejamento de projeto no SERPRO. Os treinamentos e *mentorings* de “Métodos para Estimativas de Projetos de Software” possibilitou a utilização do procedimento definido pelas equipes de desenvolvimento.

Note que o modelo CMM não é descritivo, apresenta “O que fazer”. Os métodos pesquisados e implementados mostram “Como fazer” as práticas associadas às estimativas do CMM visando uma certificação ou aderência ao modelo. A utilização dos métodos não é trivial. Além do curso básico de estimativas, as equipes solicitam o treinamento de APF. E ainda, após os cursos teóricos para a apresentação dos conceitos, é necessário *mentoring* com as equipes.

Os principais benefícios da utilização de um procedimento de estimativas são: maior acurácia das estimativas geradas e melhor relacionamento com o cliente. A utilização da métrica Pontos de Função para determinar o tamanho e das fórmulas de cálculo de Região Impossível e do tempo ótimo de desenvolvimento, propiciam a utilização efetiva de um processo de desenvolvimento incremental (baseado na RUP), onde o cliente estabelece as prioridades e a equipe de desenvolvimento negocia a melhor forma de atender as necessidades dos clientes. Também é importante destacar que vários clientes do SERPRO também tem participado dos cursos de Estimativas e Análise de Pontos de Função. Isto facilita bastante a negociação de funcionalidades/cronograma.

Como recomendações para trabalhos futuros, sugere-se a análise de métodos estatísticos para o tratamento dos dados históricos de produtividade para as estimativas de custo e de esforço. A validação das fórmulas para o cálculo do tempo ótimo de desenvolvimento (Td) região impossível (RI) também deve ser realizada, para que estas venham incorporar o método de estimativas de prazo. Outro ponto que deve ser considerado nas estimativas é como o percentual de evolução de requisitos impacta no tamanho do projeto. Também devem ser realizados experimentos, tratando os índices de produtividade (homem-hora/PF) em cada ambiente, e ainda como cada fator de impacto na produtividade levantado nos treinamentos do SERPRO, Modelo CMMI e COCOMO II influenciam nos índices de produtividade.

Referências bibliográficas

- [ABNT, 1994] ABNT - Associação Brasileira de Normas Técnicas. NBR ISO 9001/1994: Sistemas da qualidade - Modelo para garantia da qualidade em projeto, desenvolvimento, produção, instalação e serviços associados. Rio de Janeiro, ABNT, 1994.
- [ABNT, 1998] ABNT - Associação Brasileira de Normas Técnicas. NBR ISO/IEC 12207 - Tecnologia de informação - Processos de ciclo de vida de software. Rio de Janeiro: ABNT, 1998.
- [Aguiar, 1992] AGUIAR, M. Pontos de Função ou Pontos por Caso de Uso? Como Estimar Projetos Orientados a Objetos.
[http://www.bfpug.com.br/islig-rio/ Downloads/pf_ucp.pdf](http://www.bfpug.com.br/islig-rio/Downloads/pf_ucp.pdf) 2002
- [Aguiar, 2004] AGUIAR, M. Estimando os projetos com COCOMO II. 2004.
http://www.metricas.com.br/Downloads/Estimando_Projetos_COCOMO_II.pdf
- [Andrade, 2004] ANDRADE, E. *Pontos de Caso de Uso e Pontos de Função na Gestão de Estimativa de Projetos de Software Orientados a Objeto*. Tese de Mestrado, Universidade Católica de Brasília (UCB) , Fevereiro/2004.
- [Beck, 2000] BECK, K. *Extreme Programming Explained*. Addison-Wesley Professional, 2000.
- [Beck, 2001] BECK, K & FOWLER, M. *Planning Extreme Programming*. Addison-Wesley, 2001.
- [Boehm, 2000] BOEHM, B. *Software Cost Estimation With COCOMO II*. Prentice Hall, New Jersey, 2000.
- [Cunha, 2003] CUNHA, G.; ALMEIDA, E. *Estimativa de projetos com base em Casos de Uso*. Simpósio Internacional de Melhoria de Processos de Software (SIMPROS), Recife, 2003 pp.53-60.
- [Damodaram, 2004] DAMODARAN, M & WASHINGTON, A. Estimation using use case points. Computer Science Program. Texas -Victoria: University of Houston.
http://www.bfpug.com.br/Artigos/UCP/Damodaran-Estimation_Using_Use_Case_Points.pdf
- [Dekkers, 2001] DEKKERS, C. Function Points and Use Cases - Where's the Fit?.
www.qualityplustech.com/FPUseCases-files/frame.htm
- [Dekkers, 2002] DEKKERS, C. *How Function Points Support the Capability Maturity Model Integration*. Crosstalks, February 2002 pp 21-24.
- [Dekkers, 2003] DEKKERS, C. *Measuring the "logical" or "functional" Size of Software Projects and Software Application*. Spotlight Software, ISO Bulletin May 2003 pp10-13.
- [EDUWEB, 2003] EDUWEB:
http://www.eduweb.com.br/portugues/elearning_tecnologia.asp
- [Filho, 2000] FILHO, J. *Conceitos Preliminares em Homologação de Software*.
<http://www.informal.com.br/artigos/a16022000001.htm>
- [Fiorini, 1998] FIORINI, S.; STAA, A.v.; BAPTISTA, R.; *Engenharia de Software com CMM*. Brasport ;1998.

- [Forsberg, 1997] FORSBERG, K. & MOOZ, H. *System Engineering Overview. Software Requirements Engineering*, Second Edition, IEEE Computer Society Press, Washington, 1997 pp.44-72.
- [Garmus, 2001] GARMUS, D. & HERRON, D. *Function Point Analysis: Measurement Practices for Successful Software Projects*. Addison-Wesley, 2001.
- [Glass, 2003] GLASS, R.L.; *Facts and Fallacies of Software Engineering*; New York, NY: Addison-Wesley; 2003.
- [Hazan, 1996] HAZAN, C. *Análise de Pontos de Função*. Projeto Final - Universidade do Estado do Rio de Janeiro (UERJ), 1996.
- [Hazan, 1999] HAZAN, C. *Metodologia para o Uso de Indicadores na Gerência de Projetos de Desenvolvimento de Software*. Tese de Mestrado, IME, Maio 1999.
- [Hazan, 2000] HAZAN, C. *Análise de Pontos por Função: Uma Abordagem Gerencial*. Congresso Nacional da SBC, XIX Jornada de Atualização em Informática (JAI); Julho 2000 - Volume 2, pp 287-326.
- [Hazan, 2003] HAZAN, C. *Análise de Pontos por Função: Uma ferramenta na implantação do Modelo CMM*. Tema - a Revista do SERPRO, seção tematec, Jan/Fev 2003.
- [Humphrey, 1990] HUMPHREY, W. *Managing the Software Process*. Addison-Wesley, 1990.
- [IFPUG, 2004] IFPUG. *Counting Practices Manual*. Version 4.2, 2004. www.ifpug.org.
- [ISBSG, 2003] ISBSG. International Software Benchmarking Standards Group. www.isbsg.org.au
- [Jones, 1994] JONES C. *Assessment and Control of Software Risks*. Prentice Hall, 1994.
- [Jones, 1997] JONES, C. *Applied Software Measurement, Assuring Productivity and Quality*. Prentice Hall, Second Edition, 1997.
- [Jones, 1998] JONES, C. *Estimating Software Costs*. McGraw-Hill, 1998.
- [Longstreet, 2002] LONGSTREET, D. *Use Cases and Function Points*. www.softwaremetrics.com/Articles/usecases.htm
- [Kan, 1995] KAN, S. *Metrics and Models in Software Quality Engineering*. Addison Wesley ;1995 pp. 148,149.
- [Karner, 1993] KARNER, G. *Use Case Points - Resource Estimation for Objectory Projects, Objective Systems SF AB, University of Linköping, Suécia, 1993*.
- [Kruchten, 2000] KRUCHTEN, P. *The Rational Unified Process - An Introduction*. Addison-Wesley, 2000. pp167,168.
- [Masters, 2000] MASTERS, S. *Keys to Conducting Collaborative Assessments*. SIMPROS, September 2000. pp 17.
- [MCT, 2004a] MCT. Ministério da Ciência e Tecnologia. *Evolução de Certificados ISO 9000 no Brasil, 2004*.
<http://www.mct.gov.br/Temas/info/Dsi/qualidad/certiso.htm>

- [MCT, 2004b] MCT. Ministério da Ciência e Tecnologia. Qualificação CMM no Brasil, 2004.
<http://www.mct.gov.br/Temas/info/Dsi/qualidad/CMM.htm>
- [Meli, 1997] MELI, R. *Early and Extended Function Point: A new method for Function Points Estimation*. IFPUG - Fall Conference - September 1997 - Arizona, USA.
- [NESMA, 2005] NESMA - Netherlands Software Metrics Association. Indicative Function Point Count: www.nesma.org.nl
- [Oliveira, 1996] OLIVEIRA S. T. *Ferramentas para o Aprimoramento da Qualidade*. GRIFO, 2ª Edição; 1996 pp 29 a 34.
- [Orr, 2002] ORR, K. CMM Versus Agile Development: Religious Wars and Software Development. Cutter Consortium, www.cutter.com/freestuff/apmreport.htm, 2002.
- [Paulk, 2001] PAULK, M. Extreme Programming from a CMM perspective. *IEEE Software*, Novembro 2001.
- [PMI, 2000] PMI- Project Management Institute. A guide to the project management body of knowledge. Syba: PMI Publishing Division, www.pmi.org, 2000.
- [Pressman, 2000] PRESSMAN, R. S. *Software engineering: a practitioner's approach*. 5th Edition, New York: McGraw-Hill, 2000.
- [Probasco, 2002] PROBASCO, L. Dear Dr. Use Case: What About Function Points and Use Cases?. *Rational Software*, Rational Edge, 2002.
- [Putnan, 1992] PUTNAN, L. & MYERS, W. *Measures for Excellence - Reliable Software On Time, Within Budget*, Prentice-Hall, 1992.
- [Smith, 2000] SMITH, J. *The Estimation of Effort Based on Use Cases*. Rational Software White Paper, 2000.
- [SEI, 2002] SEI. Capability Maturity Model Integration (CMMISM). Version 1.1 - CMU/SEI-2002-TR-012; Março 2002.
- [SPR, 2005] SPR. Software Productivity Research. www.spr.com
- [Standish, 1994] Standish Group. The Chaos Report (1994). www.standishgroup.com
- [Standish, 2001] Standish Group. The Chaos Report (2001). www.standishgroup.com
- [Standish, 2003] Standish Group. The Chaos Report (2003). www.standishgroup.com
- [UNISYS,1995] UNISYS. *Análise de Pontos de Função*. Apostila - Curso de Análise de Pontos de Função da UNISYS, 1995.
- [Vazquez, 2004] VAZQUEZ et al. *Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software*. Editora Érica Ltda, São Paulo, pp17-35, segunda edição, 2004.