

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

**Claudia Hazan**<sup>1 2</sup>  
claudinhah@yahoo.com

**Arndt von Staa**<sup>2</sup>  
arndt@inf.puc-rio.br

<sup>1</sup> SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS (SERPRO)  
Superintendência de Produtos e Serviços - Soluções de Desenvolvimento (SUPSD)  
Rua Pacheco Leão, 1235, Prédio 1 – 2º andar, Jardim Botânico  
CEP: 22460-030 – Rio de Janeiro – RJ  
Brasil

<sup>2</sup> DEPARTAMENTO DE INFORMÁTICA PUC-Rio  
Rua Marquês de São Vicente 225, Gávea  
CEP: 22453-900 – Rio de Janeiro - RJ  
Brasil

**Abstract:** The documented estimations constitute the basis for the construction of software project plans. The size estimations of software projects must be used to derive the cost, effort and schedule estimations, according to CMMI practices. Several models have been used by the industry to estimate effort and cost, for instance COCOMO. However, the methods to estimate the project size have been a little bit talked about. This work analyzes, from the perspective of CMMI, methods used to generate software projects size estimations using Function Points. It then proposes a simplified process to generate and documenting the size, effort, budget, schedule and critical computer resources estimations.

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

**Resumo:** As estimativas documentadas constituem a base para a elaboração de um plano do projeto de software. As estimativas de tamanho de projetos de software devem ser utilizadas para a derivação das estimativas de custo, esforço e cronograma, conforme as diretrizes do CMMI. Muitos modelos têm sido utilizados pela indústria para estimar-se esforço e custo, como por exemplo o COCOMO. No entanto pouco tem se discutido sobre os métodos para estimar tamanho. 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 é proposto um método para geração de estimativas de tamanho, baseando-se em experimentos e é apresentado um processo simplificado para a geração e a documentação de estimativas de tamanho, esforço, custo, cronograma e recursos computacionais críticos.

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

## 1. Introdução

Os problemas relativos à previsibilidade de projetos constituem uma das maiores preocupações da indústria de *software*. Em 1994 o Standish Group [STA] relatou que apenas 16% dos projetos de *software* são bem sucedidos, ou seja atendem o seu objetivo dentro do cronograma e do orçamento previstos. Os dados do Standish Groups de 2001 [STA] mostram as seguintes estatísticas: 27% de projetos finalizados no tempo e custo previstos; 40% de projetos cancelados antes de finalizarem; 50% dos projetos custam em média 180% a mais da estimativa original. Note que ocorreu uma melhoria nas estatísticas de projetos bem sucedidos do Standish Group de 1994 (16%) para 2001 (27%). Esta melhoria ocorreu devido aos investimentos da indústria na implantação de processos e de qualificação em modelos da Qualidade de *Software*, tais como o CMM (*Capability Maturity Model*).

Embora tenham ocorrido melhorias, a indústria de *software* continua lidando com projetos mal sucedidos. O CHAOS Report de 2003 [STA] 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. Por outro lado, em [GLA03] Glass afirma que algumas das principais causas são estimativas baseadas em informação insuficiente (usualmente sem que a especificação de requisitos esteja disponível) e projetos com requisitos demasiadamente flutuantes.

Um dos principais riscos que atingem os processos de estimativas é a falta de credibilidade nas estimativas pelas equipes de desenvolvimento [BOE00]. 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 [SEI02].

Este trabalho tem como objetivo analisar um processo 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 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 adaptação destes métodos por meio experimentação. O propósito é obter-se melhor acurácia nas estimativas de tamanho. Além disso, 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 [SEI02].

### 1.1 Organização do Trabalho

Este trabalho está organizado da seguinte forma: a **seção 2** fornece uma visão geral do **gerenciamento de projetos**, destacando as diretrizes para os **processos de geração de estimativas** do modelo CMMI. A **seção 3** apresenta uma **análise das falhas em um processo de estimativas**, usando o Diagrama de Causa e Efeito. A **seção 4** apresenta uma **proposta de implementação de melhorias** no processo analisado. Esta seção relata, ainda, alguns resultados iniciais relativos a definição de **novos métodos para geração de estimativas de tamanho** e derivação empírica das estimativas de esforço, prazo e custo. A **seção 5** mostra uma **conclusão** para o trabalho e **recomendações para trabalhos futuros**.

## 2. Gerenciamento de Projetos

**Humphrey** [HUM90] define projeto como 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 de um cliente. Por exemplo: a elaboração de um sistema novo; a evolução de uma ou mais funcionalidades de um sistema existente.

O gerenciamento de projetos é a aplicação de conhecimentos, habilidades e técnicas para projetar atividades que visem atingir ou exceder as necessidades e expectativas dos *stakeholders* do projeto (partes interessadas no projeto). Este ato envolve o equilíbrio entre demandas concorrentes de escopo, prazo, custo e qualidade [PMI00].

O gerenciamento de projetos tem como objetivo 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 [ABN98]. As principais atividades realizadas por gerentes de projetos de *software* consistem nas estimativas de tamanho, cronograma, recursos, custos e eficiência de remoção de defeitos [JON97].

### 2.1 O Processo CMMI para o Estabelecimento de Estimativas

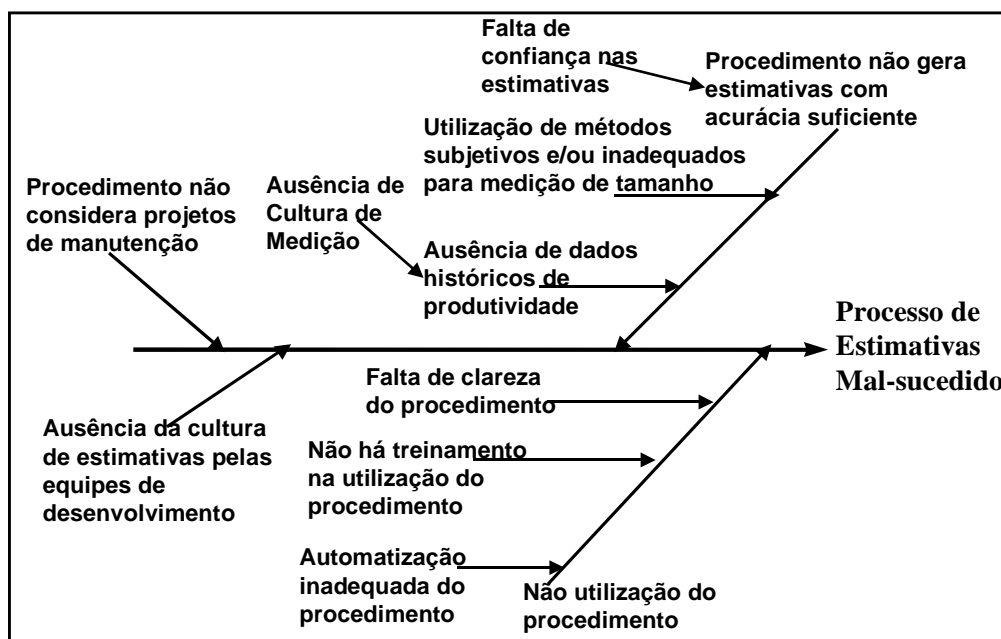
O estabelecimento de estimativas constitui uma das principais atividades do processo de planejamento do projeto de *software*. As estimativas devem ter uma base segura e fornecer confiança que os planos baseados nelas sejam capazes de suportar os objetivos do projeto, que englobam funcionalidades, custo, prazo e atributos da qualidade. 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[SEI02]:

- ◆ **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. Quando ocorrem mudanças nos requisitos, o projeto precisa ser reestimado. Sugere-se a utilização da métrica Pontos de Função (PF) [IFP04] para as estimativas de tamanho. Os PFs fornecem uma medida de tamanho do *software*, quantificando a dimensão das funcionalidades do *software*, sob o ponto de vista lógico ou seja, observando os requisitos do usuário [HAZ00]. Além da métrica apoiar a implantação dos modelos CMM [HAZ03] e CMMI [DEK02], esta foi reconhecida pela norma ISO/IEC 20926 [DEK03].
- ◆ **Determinar as estimativas de esforço e de custo para os artefatos e tarefas:** as estimativas de esforço e de custo devem ser geradas baseando-se em dados objetivos, ou seja em dados históricos de projetos concluídos. 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 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.

### 3. Análise do Processo de Estimativas do SERPRO

Esta seção tem como propósito apresentar a aplicação da ferramenta **diagrama de causa e efeito** na identificação do relacionamento entre as principais causas e seus efeitos das falhas do processo de estimativas do SERPRO.

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 [KAN95] [OLI96]. A **Figura 1** ilustra a identificação das principais causas dos problemas que existiam no processo de estimativas do SERPRO.



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

A causa raiz dos principais problemas do processo de estimativas foi a utilização da métrica Pontos por Caso de Uso (PCU). O método de Pontos por Casos de Uso [PRO02] [SMI00] foi apresentado para as equipes de desenvolvimento como uma “Silver Bullet” para as estimativas de tamanho, um método simples e eficaz. O principal problema que os desenvolvedores encontraram na utilização da métrica PCU foram os seguintes:

- o Falta de acurácia das estimativas de esforço, devido à ausência de base histórica;
- o 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;
- o Dificuldade de estimar pequenos projetos de manutenção ou apurações especiais, nos quais não há modelagem de requisitos. Por exemplo: desenvolvimento de uma rotina para a atualização de um arquivo (*file*) e geração de um relatório;
- o Impossibilidade de geração de estimativas iniciais para o plano do projeto, a partir do artefato Documento de Visão [KRU00] 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, seja produzido no início do projeto, quando os requisitos estão sendo elicitados e ainda não foram modelados [FIO98].

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.

#### 4. Implementação de Melhorias no Processo de Estimativas

A análise do diagrama de causa e efeito (**figura 1**), apresentado na **seção 3**, permitiu a visualização, de forma organizada, de um grande problema. A decisão realizada, considerando as causas de problemas, foi a criação de um procedimento de estimativas e do curso “Processo de Estimativas” seguido de *mentoring*, visando a capacitação das equipes no procedimento.

O procedimento para geração de estimativas de tamanho utilizou os princípios do modelo CMM [FIO98] e do modelo XP – eXtreme Programming – [BEC01], com o intuito de tornar o processo de estimativas simples, devido aos custos para aquisição de uma ferramenta de estimativas. A abordagem de utilizar as diretrizes de simplicidade do XP para alcançar as metas do modelo CMM é defendida por Orr [ORR02] e por Paulk [PAU01].

O propósito da criação do procedimento é possibilitar que os líderes de projetos gerem e documentem as estimativas de tamanho, esforço, prazo, custo e recursos computacionais críticos do projeto de *software* de forma consistente e padronizada para o planejamento.

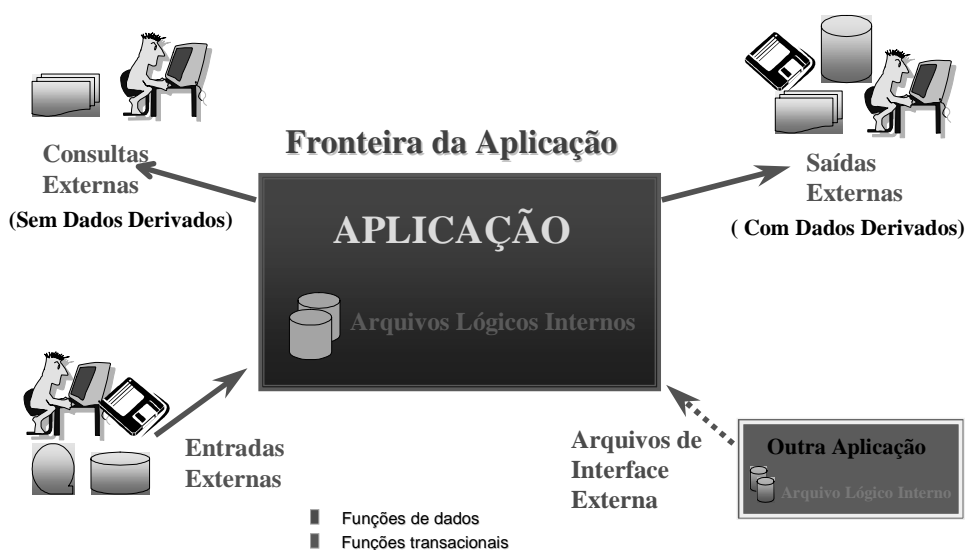
##### 4.1 Métodos para Geração de Estimativas de Tamanho

Nesta seção propõe-se um método para geração de estimativas de tamanho em Pontos de Função (PF). De acordo com Dekkers [DEK01], Aguiar [AGU02] e Longstreet [LON02] 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.

A métrica Pontos de Função visa estabelecer uma medida de tamanho do *software*, considerando a funcionalidade implementada, sob o ponto de vista do usuário, independentemente da metodologia e da plataforma utilizadas no desenvolvimento do sistema [HAZ00]. Além disso, na prática a técnica Análise de Pontos por Função tem demonstrado grande utilidade na garantia da completeza da especificação de requisitos. Uma contagem de Pontos de Função é realizada usando uma descrição suficientemente formal das necessidades do negócio na linguagem do usuário. Assim, a contagem pode servir também como uma revisão dos requisitos do sistema com o usuário [HAZ03].

A aplicação do método de Contagem de Pontos de Função publicado pelo IFPUG (*International Function Points Users Group*) no CPM (*Counting Practices Manual*) versão 4.2 preconiza a existência do projeto lógico da aplicação [IFP04]. Note que na fase de planejamento os requisitos estão sendo elicitados, e portanto não existe um projeto lógico completo e validado. Assim, foi definido um método de contagem estimativa, visando a aferição do tamanho do projeto em Pontos de Função no início do projeto, utilizando-se o Documento de Visão.

O método consiste no mapeamento dos requisitos descritos no Documentos de Visão nos cinco tipos funcionais da APF, ilustrados na **Figura 2**, e na identificação do tamanho do requisito, baseando-se nas tabelas de complexidade e de contribuição funcional, descritas no CPM 4.2 [IFP04]. As necessidades e funcionalidades especificadas para o projeto devem ser enquadradas em uma das tabelas abaixo:



**Figura 2: Visão Geral dos Tipos Funcionais da Análise de Pontos por Função [HAZ00]**

**Tabela A - Contagem dos Arquivos Lógicos Internos (ALIs): Banco de Dados da Aplicação (tabelas e files 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 trabalho, tabelas de relacionamento sem atributos próprios (que transporte apenas as chaves estrangeiras) e as entidades fracas como um ALI.

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

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 **Simples**.

**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 files 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 considere arquivos físicos, arquivos de trabalho e tabelas de relacionamento sem atributos próprios e entidades fracas.

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

Nº AIEs Simples:	X 5 PF
Nº AIEs Médio:	X 7 PF
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 *files* 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.

**Questão 3:** Você está incluindo, alterando ou excluindo registros de tabelas ou 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)?

<b>Nº EEs Simples:</b>	<b>X 3 PF</b>
<b>Nº EEs Média:</b>	<b>X 4 PF</b>
<b>Nº EEs Complexa:</b>	<b>X 6 PF</b>
<b>Total PF da Tabela C:</b>	

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.

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

<b>Nº CEs Simples:</b>	<b>X 3 PF</b>
<b>Nº CEs Média:</b>	<b>X 4PF</b>
<b>Nº CEs Complexa:</b>	<b>X 6 PF</b>
<b>Total PF da Tabela IV:</b>	

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.

**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 *file*, 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 5PF
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 em PFs pelo método de contagem estimativa deve ser gerada totalizando-se os PFs obtidos nas tabelas **A, B, C e D, E**.

#### 4.2 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. Para uma estimativa de esforço e prazo mais precisa recomenda-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 simplificar a geração de estimativas de esforço, 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 onde 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] e benchmarking com outras empresas.

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 influenciam na produtividade.

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

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

Note que o nível de **produtividade Baixa** está associado, dentre outros fatores, à: **aplicações complexas** ou **equipes inexperientes na plataforma de desenvolvimento**. Já o nível de **produtividade Alta** está associado à **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 do SERPRO, os setores que já possuem um pequeno histórico de projetos concluídos tem 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 **fator de conversão HH**, contemplando o percentual de cada linguagem. 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. 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. A construção do banco de dados histórico deve levar em consideração poucos atributos relevantes sobre o ponto de vista de influência na produtividade.

### 4.3. 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** [JON97].

O próximo passo é utilizar a fórmula abaixo:

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

As estimativas de prazo não são lineares com o tamanho, assim pretende-se pesquisar mais sobre o menor tempo desenvolvimento, dado o tamanho de um projeto específico. Jones propõe uma fórmula para o cálculo de menor tempo de desenvolvimento, denominado Td e de Região Impossível (RI) de desenvolvimento, descrito em [JON98]. 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. O método ainda não considera o cálculo da Região Impossível.

#### 4.4. Método para Estimativas de Custo

Segundo o CMMI, 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 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.

#### 4.5. 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 3**.

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

**Figura 3: Modelo para Estimativas de Recursos Computacionais Críticos**

**Nome do Recurso:** servidor, periférico, 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 ambiente de Homologação

**Custo:** Preencher este campo quando for possível a definição dos custos do recurso

## 5. Conclusão

As estimativas constituem a base para a elaboração do plano do projeto de *software*. Assim, 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 os treinamentos. Os experimentos realizados com os métodos de estimativas de tamanho têm demonstrado um desvio entre o tamanho previsto e o realizado inferior a 20%, sem levar em consideração as mudanças de requisitos no decorrer do desenvolvimento, as quais acarretam em uma reestimativa do projeto, conforme preconizado no CMMI.

A definição do procedimento de estimativas, descrito neste trabalho foi um passo importante na implantação da área chave de planejamento de projeto no SERPRO. Note que o modelo CMM não é descritivo, apresenta "O que fazer". O procedimento estabelece o "Como fazer" as práticas associadas às estimativas do CMM. A utilização dos métodos não é trivial. Além do curso básico de estimativas, as equipes solicitaram o treinamento de APF. Além dos cursos teóricos, é necessária a realização de *mentoring* com as equipes.

Um ponto importante a ser considerado nas estimativas de tamanho é o Fator de Ajuste da Contagem de PF, embora este não seja atualizado pelo IFPUG desde a década de 80, não seja reconhecido pela ISO [DEK03] e algumas empresas o tenham desconsiderado. A opção do SERPRO foi seguir o manual CPM 4.2 [IFP04], considerando o cálculo fator de ajuste nas estimativas de tamanho. 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 tem sua precisão dependente da existência de dados históricos de projetos similares. Neste momento, a empresa está iniciando a construção de um Banco de Dados Histórico de produtividade. Também pretende-se utilizar o COCOMO II [BOE00] nas estimativas de esforço, visando comparar seus resultados com os resultados gerados mediante a aplicação do modelo simplificado de estimativas de esforço.

Como recomendações para trabalhos futuros, sugere-se a análise de métodos estatísticos para o tratamento dos dados históricos de produtividade, visando a melhoria das estimativas de esforço. A validação das fórmulas para o cálculo da região impossível também deve ser realizada, para que estas venham incorporar o método de estimativas de prazo.

## Referências

- [ABN98] 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.
- [AGU02] AGUIAR, M. Pontos de Função ou Pontos por Caso de Uso? Como Estimar Projetos Orientados a Objetos. [www.metrics.com.br](http://www.metrics.com.br), 2002
- [BEC01] BECK, K & FOWLER, M. Planning Extreme Programming. Addison-Wesley, 2001.
- [BOE00] BOEHM, B. Software Cost Estimation With COCOMO II. Prentice Hall, New Jersey, 2000.

- [DEK01] Dekkers, C. Function Points and Use Cases – Where’s the Fit?.  
[www.qualityplustech.com/FPUseCases-files/frame.htm](http://www.qualityplustech.com/FPUseCases-files/frame.htm)
- [DEK02] DEKKERS, C. How Function Points Support the Capability Maturity Model Integration, CrossTalk, February 2002. pp 21 a 24.
- [DEK03] DEKKERS, C. Measuring the “logical” or “functional” Size of Software Projects and Software Application. Spotlight Software, ISO Bulletin may 2003. pp 10 a 13.
- [FIO98] FIORINI S., STAA A., BAPTISTA R. Engenharia de Software com CMM. Brasport, 1998.
- [GLA03] GLASS, R.L. Facts and Fallacies of Software Engineering; Addison-Wesley; 2003
- [HAZ99] HAZAN, C. Metodologia para o Uso de Indicadores na Gerência de Projetos de Desenvolvimento de Software. Tese de Mestrado, IME, Maio 1999.
- [HAZ00] HAZAN, C. Análise de Pontos por Função: Uma Abordagem Gerencial. Congresso Nacional da SBC, Jornada de Atualização em Informática (JAI); 2000.
- [HAZ03] HAZAN, C. Análise de Pontos por Função: Uma ferramenta na implantação do Modelo CMM. Tema - a Revista do SERPRO, seção Tematec, Jan/Fev 2003.
- [HUM90] HUMPHREY, W. Managing the Software Process. Addison-Wesley, 1990.
- [IFP04] IFPUG. Counting Practices Manual. Version 4.4, June2004. [www.ifpug.org](http://www.ifpug.org).
- [JON97] JONES, C. Applied Software Measurement, Assuring Productivity and Quality. Prentice Hall, Second Edition, 1997.
- [JON98] Jones, C. Estimating Software Costs. McGraw-Hill, 1998.
- [LON02] Longstreet, D. Use Cases and Function Points. [www.softwaremetrics.com](http://www.softwaremetrics.com)
- [KRU00] KRUCHTEN, P. The Rational Unified Process – An Introduction. Addison-Wesley, 2000. pp 167,168.
- [OLI96] OLIVEIRA S. T., Ferramentas para o Aprimoramento da Qualidade, GRIFO, 2ª Edição; 1996 pp 29 a 34.
- [ORR02] ORR, K. CMM Versus Agile Development: Religious Wars and Software Development. Cutter Consortium, [www.cutter.com/freestuff/apmreport.htm](http://www.cutter.com/freestuff/apmreport.htm), 2002.
- [PAU01] PAULK, M. Extreme Programming from a CMM perspective. IEEE Software, Novembro 2001 pp 1 a 8.
- [PMI00] PMI- Project Management Institute. A guide to the project management body of knowledge. Syba: PMI Publishing Division, [www.pmi.org](http://www.pmi.org), 2000.
- [PRO02] PROBASCIO, L. Dear Dr. Use Case: What About Function Points and Use Cases?. Rational Software, Rational Edge, 2002.
- [SMI00] Smith, J. The Estimation of Effort Based on Use Cases. Rational Software White Paper, 2000.
- [SEI02] SEI. Capability Maturity Model Integration (CMMI-SW). Version 1.1 - CMU/SEI-2002-TR-012; Março 2002.
- [SPR] Software Productivity Research. [www.spr.com](http://www.spr.com)
- [STA] Standish Group. The Chaos Report. <http://www.standishgroup.com>