

**Uma Proposta de Aplicação da Métrica de  
Pontos de Função em Aplicações de  
Dispositivos Portáteis**

*Marcus Vinicius La Rocca Macedo*

Trabalho Final de Mestrado Profissional

# **Uma Proposta de Aplicação da Métrica de Pontos de Função em Aplicações de Dispositivos Portáteis**

**Marcus Vinícius La Rocca Macedo**

Dezembro de 2003

## **Banca Examinadora:**

- **Prof. Dr. Paulo Lício de Geus (Orientador)**  
Instituto de Computação, UNICAMP
- **Profa. Thelma Chiossi (Co-Orientadora)**  
Instituto de Computação, UNICAMP
- **Profa. Dra. Ana Cervigni Guerra**  
Centro de Pesquisas Renato Archer
- **Profa. Dra. Eliane Martins**  
Instituto de Computação, UNICAMP
- **Profa. Dra. Ariadne Maria Brito Rizzoni Carvalho (Suplente)**  
Instituto de Computação, UNICAMP

**FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DO IMECC DA UNICAMP**

Macedo, Marcus Vinícius La Rocca

M151p Uma proposta de aplicação da métrica de pontos de função em aplicações de dispositivos portáteis / Marcus Vinícius La Rocca Macedo -- Campinas, [S.P. :s.n.], 2003.

Orientador : Paulo Lício de Geus.

Co-Orientadora: Thelma Chiossi.

Trabalho final (mestrado profissional) - Universidade Estadual de Campinas, Instituto de Computação.

1. Engenharia de software. 2. Software – Estimativas. 3. Telefone.  
I. Geus, Paulo Lício de. II. Chiossi, Thelma. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

# **Uma Proposta de Aplicação da Métrica de Pontos de Função em Aplicações de Dispositivos Portáteis**

Este exemplar corresponde à redação  
do Trabalho Final submetida à Banca  
Examinadora.

Campinas, dezembro de 2003.

Prof. Dr. Paulo Lício de Geus  
(Orientador)

Profa. Thelma Chiossi  
(Co-Orientadora)

Trabalho Final apresentado ao Instituto de  
Computação, UNICAMP, como requisito  
parcial para a obtenção do título de  
Mestre em Computação na área de  
Engenharia de Computação



**© Marcus Vinícius La Rocca Macedo, 2003**  
**Todos os direitos reservados**

# Resumo

Este trabalho apresenta uma proposta de contagem de pontos de função para aplicações de dispositivos portáteis (telefones celulares) tomando como base técnicas de contagem para aplicações de interface gráfica (GUI). A obtenção da contagem de pontos de função a partir de especificações funcionais e de layout fornecidas no início do projeto de desenvolvimento do produto de software abre a possibilidade de se estabelecer estimativas de custo (esforço) e duração mais precisas, beneficiando significativamente o planejamento do projeto.

Nesse sentido são apresentadas algumas técnicas de estimativas de esforço. Também são apresentadas e comparadas algumas métricas de software com ênfase na métrica de pontos de função. Finalmente são apresentados os conceitos básicos de aplicações gráficas de telefones celulares e é estabelecido um paralelo na contagem de pontos de função para então se mostrar um exemplo prático de contagem utilizando este paralelo.

# **Abstract**

This dissertation presents a proposal of function point account for portable device (wireless phone) applications based on normal graphical user interface (GUI) application accounting techniques. The function point accounting done with functional and layout specification provided in the beginning of the software development project allows the establishment of cost (effort) estimation more precise, significantly improving the project planning.

This way some effort estimation techniques are presented. Software metrics are also presented and compared with emphasis on function point metric. Finally, the basic concepts of graphical user interface wireless phone applications are presented and it is established a parallel in the function point accounting that is applied in a practical example.

# Agradecimentos

À Carmen pela paciência e apoio.

À Prof. Thelma Chiossi pela orientação, compreensão e apoio na realização deste trabalho.

Ao Prof. Paulo Lício de Geus pela orientação recebida.

À Área de Pesquisa e Desenvolvimento do setor PCS da Motorola Industrial Ltda pelo patrocínio e apoio na realização deste trabalho.

Aos colegas Fernando Nascimento e Rose Hecksher pela inspiração na definição do tema deste trabalho.

# Conteúdo

1	Introdução.....	1
1.1	Objetivo .....	1
1.2	Apresentação .....	2
2	Métricas no Planejamento do Desenvolvimento de Software.....	3
2.1	Entradas para o Planejamento .....	4
2.1.1	Necessidades do Cliente .....	4
2.1.2	Estudo de Viabilidade.....	5
2.1.3	Lista de Alternativas.....	5
2.2	Estimativas .....	6
2.2.1	Fatores que afetam o processo de estimativa .....	6
2.2.2	Técnicas de Estimativa .....	7
2.2.3	Modelos Algorítmicos de Estimativa .....	9
2.3	Análise de Risco .....	9
2.4	Definição das Atividades.....	10
2.5	Definição do Cronograma e Orçamento.....	11
3	Métricas de Software.....	12
3.1	A Métrica de Pontos de Função.....	12
3.1.1	Contagem dos Pontos de Função Não Ajustados .....	13
3.1.2	Ajustando a Contagem.....	16
3.1.3	Guia de Contagem .....	19
3.1.4	Pontos de Função e Casos de Uso .....	20
3.2	Outras Métricas .....	22
3.3	Discussão sobre Métricas de Tamanho Utilizadas para Estimativas de Esforço .....	23
3.3.1	LOC .....	23
3.3.2	Quantidade de Memória .....	24
3.3.3	Ponto de Função .....	24
4	Pontos de Função em Aplicações para Dispositivos Portáteis .....	27
4.1	Conceitos Básicos para Aplicações Gráficas de Dispositivos Portáteis.....	27
4.2	Elementos Básicos de Interface Gráficas de Dispositivos Portáteis.....	29
4.2.1	Elementos de Entrada de Dados .....	29
4.2.2	Elementos de Saída de Dados.....	31
4.3	Contagem de Pontos de Função para Aplicações GUI.....	32
4.3.1	Funcionalidades Incluídas na Contagem .....	32
4.3.2	Funcionalidades Excluídas da Contagem.....	33
4.4	Paralelo na Contagem de Pontos de Função.....	34
4.4.1	Funcionalidades Incluídas na Contagem .....	34
4.4.2	Funcionalidades Excluídas da Contagem.....	38
5	Exemplificação da Proposta de Contagem .....	39
5.1	Aplicação Agenda de Compromissos.....	39
5.2	Início da Contagem.....	40
5.3	Contagem dos Arquivos .....	40
5.4	Contagem das Transações .....	42
5.5	Totalização dos Pontos de Função Não Ajustados.....	51
5.6	Determinação do Fator de Ajuste de Valor .....	51

5.7 Valor Final dos Pontos de Função Ajustados.....	52
6 Considerações Finais.....	54
6.1 Trabalho Realizado.....	55
6.2 Contribuição.....	55
6.3 Extensões.....	56
Referências Bibliográficas.....	57

## Lista de Tabelas

Tabela 1 -	Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma EI ..	14
Tabela 2 -	Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma EO.	14
Tabela 3 -	Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma EQ.	15
Tabela 4 -	Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma ILF	16
Tabela 5 -	Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma EIF	16
Tabela 6 -	Totalização dos Pontos de Função Não Ajustados.....	51
Tabela 7 -	Características Gerais de Sistema.....	52

# Lista de Figuras

Figura 1 -	Estimativas e Projeto de Desenvolvimento do Produto de Software .....	4
Figura 2 -	Erro de Estimativa Versus Fase ([BOE 81]) .....	7
Figura 3 -	Exemplo de WBS .....	10
Figura 4 -	Diagrama de Casos de Uso.....	21
Figura 5 -	Telefone Celular (Dispositivo Portátil) .....	28
Figura 6 -	Formulário .....	29
Figura 7 -	Editor .....	29
Figura 8 -	Lista Marcada .....	30
Figura 9 -	Lista de Seleção Múltipla .....	30
Figura 10 -	Medidor Ajustável.....	30
Figura 11 -	Picker .....	31
Figura 12 -	Notice de Questionamento .....	31

# Capítulo 1

## Introdução

Segundo Watts S. Humphrey “até hoje, os produtos de software têm os seus cronogramas atrasados, custos maiores do que os esperados e apresentam defeitos. Isto tem como resultado uma série de inconvenientes para os usuários e traz consigo uma enorme perda de tempo e de recursos” [WEB 99].

A origem de cronogramas atrasados e custos maiores do que os esperados está relacionada à ausência de estimativas de esforço ou a estimativas de esforço incorretas nos projetos de desenvolvimento do produto de software. Segundo Maria Isabel Haufe “a estimativa é parte integrante do planejamento e controle do processo de gerência de projetos para que um gerente de projetos possa definir o prazo, o custo e o esforço do desenvolvimento” [HAU 01]. Na verdade, conforme explica Ariadne Carvalho, “na engenharia de software o custo principal é o esforço, ou seja, o custo de mão de obra; assim, para se calcular o custo do software, é necessário dimensionar o trabalho para desenvolvê-lo” [CAR 01].

Barry Boehm mostra através de seu estudo apresentado em [BOE 81] que a precisão da estimativa de esforço tende a ser pior na fase inicial do projeto pelo desconhecimento do software a ser desenvolvido. Nesse sentido, a métrica de ponto de função pode ser utilizada para a obtenção de estimativas melhores na fase inicial do desenvolvimento conforme afirma S. Furey [FUR 97].

A métrica de pontos de função representa uma medida de tamanho funcional do software. Existem diversas técnicas de contagem de pontos de função para os diferentes tipos de aplicação como aplicações cliente-servidor, aplicações baseadas na web (Internet), aplicações de prospecção de dados e aplicações de interface gráfica (GUI) [GAR 01], mas nenhuma destas técnicas trata especificamente de aplicações de dispositivos portáteis (telefones celulares) como é feito neste trabalho.

A seguir apresenta-se o objetivo deste trabalho para então se apresentar o seu conteúdo.

### 1.1 Objetivo

Este trabalho visa identificar semelhanças entre as aplicações de interface gráfica (GUI) para computadores em geral e as aplicações de interface gráfica para dispositivos portáteis (telefones celulares) a fim de utilizar as técnicas de contagem de pontos de função

para aplicações GUI para propor uma técnica de contagem de pontos de função para aplicações de dispositivos portáteis (telefones celulares). Visto que técnicas de contagem de ponto de função podem ser utilizadas no início dos projetos de desenvolvimento destas aplicações, será considerado como ponto de partida para a contagem documentos de especificação funcional e de layout da interface gráfica.

## **1.2 Apresentação**

Inicialmente o capítulo 2 apresentará a fase inicial do projeto de desenvolvimento do produto de software que é o planejamento do projeto de desenvolvimento do produto de software, enfatizando-se a utilização de métricas de tamanho para a obtenção de estimativas de esforço que viabilizam tal planejamento.

Em seguida, no capítulo 3, será apresentada a métrica de pontos de função e outras métricas, além de se discutir as vantagens e desvantagens de utilização da métrica de pontos de função em lugar de outras para a realização de estimativas iniciais.

Apresenta-se no capítulo 4 uma descrição dos elementos básicos da interface gráfica de telefone celular utilizada como exemplo, e apresentam-se também as diretrizes de obtenção da contagem de pontos de função para aplicações GUI em geral para então, estabelecer um paralelo comparativo na obtenção dos pontos de função em uma aplicação de dispositivo portátil (telefone celular).

No capítulo 5 exemplifica-se a contagem de pontos de função pela aplicação do paralelo proposto em uma aplicação chamada Agenda de Compromissos.

Considerações finais sobre os resultados atingidos e as possíveis extensões são feitas no capítulo 6.

## Capítulo 2

# Métricas no Planejamento do Desenvolvimento de Software

A complexidade das empresas modernas, fruto do elevado nível de competitividade e de avanços tecnológicos, provocou um aumento na exigência da qualidade e complexidade das decisões administrativas [CAN 99]. Estas decisões administrativas se concentram no gerenciamento do projeto de software no contexto do desenvolvimento do produto de software.

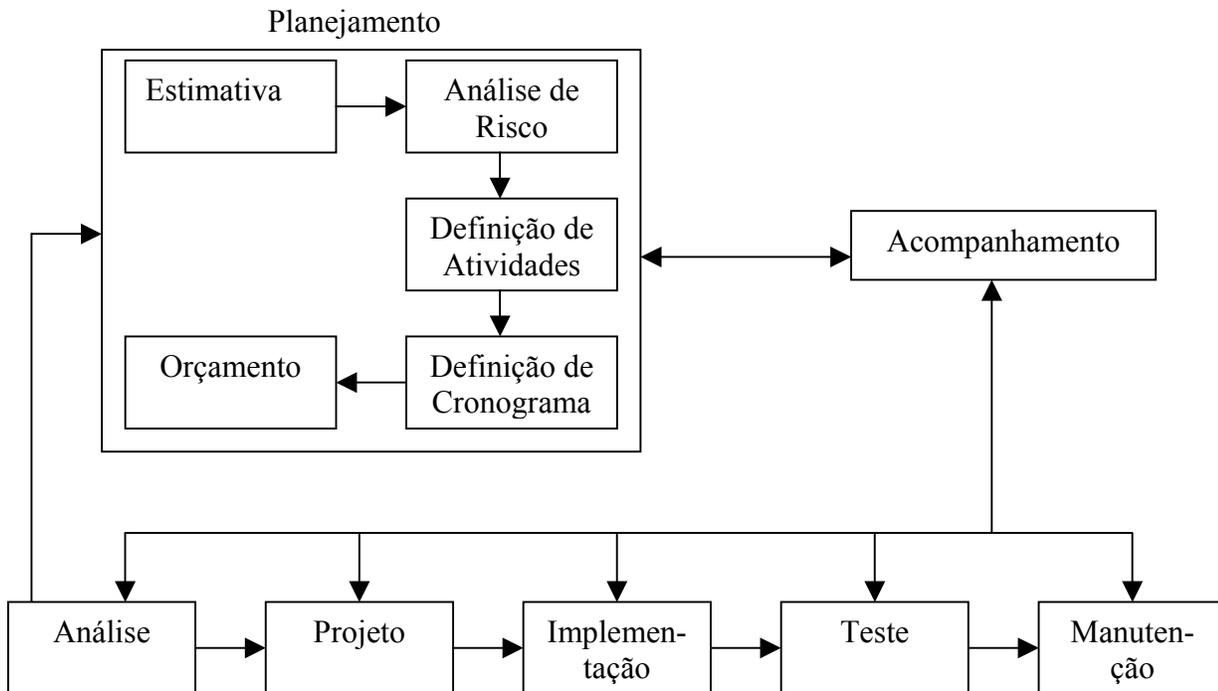
Praticamente todos os modelos de processo de desenvolvimento do produto de software possuem as atividades de análise, projeto, implementação, teste e manutenção (figura 1). Já o gerenciamento do projeto de desenvolvimento do produto de software divide-se basicamente em planejamento e acompanhamento.

O planejamento é a fase inicial do gerenciamento do projeto de desenvolvimento do produto de software no qual são realizadas várias atividades que podem ser visualizadas na figura 1. Nesta figura o planejamento aparece como atividade que, utilizando-se das informações iniciais da atividade de análise planeja a execução do projeto e serve de base para o seu acompanhamento.

É importante enfatizar a utilização de métricas de software como fundamental tanto no planejamento quanto no acompanhamento do projeto de desenvolvimento do produto de software, pois as métricas de software permitem quantificar de maneira objetiva grandezas relevantes para o planejamento e o acompanhamento do projeto sem dar margem a interpretações subjetivas. Afirmar que um projeto é longo ou de que ele é grande são subjetivas em comparação a dizer-se que o mesmo projeto durará dois anos e que envolverá a implementação de vinte mil linhas de código.

Uma das atividades da fase de planejamento extremamente importante é o processo de estimativa que consiste basicamente em estimar algumas métricas de software. A estimativa da métrica de custo (esforço) é um dado básico para o planejamento do projeto de software na medida em que permitem a tomada de decisões sobre viabilidade, elaboração de um cronograma e alocação de recursos [STU 96].

Nos itens seguintes serão descritas as principais atividades relacionadas ao planejamento do desenvolvimento do produto de software a utilização de métricas em algumas destas atividades.



**Figura 1 - Estimativas e Projeto de Desenvolvimento do Produto de Software**

## 2.1 Entradas para o Planejamento

A análise do problema envolve normalmente de 10 a 20% do esforço de projeto e objetiva dar subsídios para o planejamento do projeto ou até mesmo o seu cancelamento [PRE 92]. Os próximos tópicos descreverão as atividades desta fase que são: necessidades do cliente, estudo de viabilidade e lista de alternativas.

### 2.1.1 Necessidades do Cliente

Esta é a primeira tarefa da análise do problema e objetiva identificar as necessidades que originaram o desenvolvimento do produto de software.

Através de entrevistas com o cliente e/ou do fornecimento pelo cliente de um documento de descrição do produto de software a ser desenvolvido, as seguintes perguntas devem ser respondidas:

- Quais informações vão ser produzidas pelo sistema?
- Quais informações vão ser fornecidas para o sistema?
- Quais funções e/ou processamentos devem ser executados pelo sistema?
- Qual o desempenho esperado do sistema?

Nesse sentido, segundo Ariadne Carvalho, algumas dificuldades podem surgir ocasionadas pela falta de conhecimento do usuário das suas reais necessidades, falta de conhecimento do desenvolvedor sobre o domínio do problema, usuários afastados da

extração de requisitos, comunicação inadequada, problemas de comportamento e questões técnicas [CAR 01].

É interessante priorizar os requisitos desde o início do desenvolvimento [WIE 99], pois é difícil realizar esta priorização quando, apesar de boa parte do sistema já estar desenvolvida, o atraso no cronograma estabelecido tornou-se irrecuperável, e se necessita garantir que o seu lançamento não atrase. Pode-se priorizar os requisitos em três categorias:

- **Requisitos essenciais:** a não implementação destes requisitos impede o lançamento do sistema.
- **Requisitos desejáveis:** não impedem o lançamento do sistema, mas a sua implementação é interessante ou necessária.
- **Requisitos opcionais:** a sua necessidade é questionável.

Através da categorização de requisitos apresentada acima, torna-se mais fácil escolher requisitos que poderão não ser implementados. Por exemplo, descartar um ou mais requisitos opcionais em caso de problemas como falta de recursos ou prazos (cronogramas) excessivamente curtos.

### 2.1.2 Estudo de Viabilidade

Uma vez que os objetivos/funcionalidades principais do sistema são identificados, algumas informações adicionais precisam ser pesquisadas:

- Tecnologia para implementar o sistema.
- Recursos especiais de desenvolvimento e/ou manufatura que vão ser necessários.
- Restrições de custo e/ou cronograma.
- Mercado potencial para este sistema.
- Comparação com os sistemas similares da concorrência.

Estas informações dizem respeito à viabilidade do desenvolvimento do sistema.

### 2.1.3 Lista de Alternativas

Diversas técnicas podem ser utilizadas para criar uma lista de alternativas. Entre elas pode-se citar a técnica *brainstorming* [CAR 01]. Cada alternativa pode representar uma opção diferente em termos de tecnologia, arquitetura ou escopo para o desenvolvimento do produto de software.

Avalia-se o escopo das diferentes alternativas levantadas tendo-se como um dos critérios fundamentais que o escopo deve pelo menos atender aos requisitos essenciais do produto de software. Já em relação à tecnologia utilizada em cada alternativa, um dos critérios básicos é que a tecnologia deve estar disponível.

Uma vez determinado as alternativas viáveis em termos de escopo e tecnologia, deve-se avaliar a viabilidade econômica destas alternativas. A avaliação econômica pode começar estimando-se o valor da métrica de esforço (custo) associado.

O valor estimado da métrica de esforço (custo) serve para determinar quais alternativas viáveis em termos tecnológicos e de escopo apresentam boas relações custo-benefício. Também serve para determinar quais destas alternativas possibilitam o desenvolvimento do produto de software dentro do prazo especificado com os recursos disponíveis.

## 2.2 Estimativas

Apesar de antiga, a conhecida lei de Parkinson [PAR 57] (“Trabalho se expande ocupando o volume disponível”) exemplifica a importância de se obter estimativas precisas e confiáveis das métricas de software. Tal lei significa no contexto de estimativa da métrica de custo (esforço) de desenvolvimento do produto de software que, se um tempo excessivo for fornecido para um determinado projeto, os membros da equipe de desenvolvimento tenderão a preencher o seu tempo com atividades paralelas como treinamento, leitura de e-mail e atividades pessoais. Por outro lado, quantidade de esforço subestimada leva a cronogramas impossíveis de serem cumpridos ou demandam sacrifícios dos membros da equipe de desenvolvimento que poderão resultar em sérios problemas a longo prazo tais como: diminuição do moral, diminuição das habilidades técnicas, e aumento da rotatividade na equipe [BOE 81]. Finalmente, uma estimativa da métrica de custo (esforço) correta permite a gerência do projeto conduzi-lo no prazo e dentro do orçamento, além de garantir um ambiente de trabalho saudável para toda a equipe.

### 2.2.1 Fatores que afetam o processo de estimativa

Os fatores que afetam o processo de estimativa das métricas de software segundo Richard D. Stutzke [STU 96] e outros são relacionados a seguir.

- **Objetivos de projeto conflitantes:** surgem devidos aos diferentes interesses ou pontos de vista dos envolvidos no projeto como desenvolvedores, gerentes de projetos, patrocinadores e clientes.
- **Falta de uma descrição detalhada do sistema:** as estimativas são necessárias antes do produto de software estar bem definido [STU 96], e elas têm sua precisão intimamente relacionada com a quantidade de informação disponível no momento da obtenção ou elaboração das estimativas sobre as métricas do software a ser desenvolvido. A figura 2 [BOE 81] ilustra a melhoria da precisão das estimativas com o avanço de projetos de componentes de

software relacionados à interface humano computador. As duas linhas traçadas no gráfico mostram que o erro esperado da estimativa do valor da métrica de esforço pode ser de 300% para mais ou para menos no início do projeto e vai diminuindo conforme as estimativas são feitas em fases posteriores. A primeira boa estimativa do valor da métrica de esforço (custo), pode ser obtida quando já se definiu a arquitetura do software. Neste momento já se passou 40% da duração total do projeto e já se gastou 20% do esforço total do projeto [STU 96] (a precisão média da estimativa, ou seja, a diferença entre o valor estimado para a métrica e o valor real, neste ponto é de 25%).

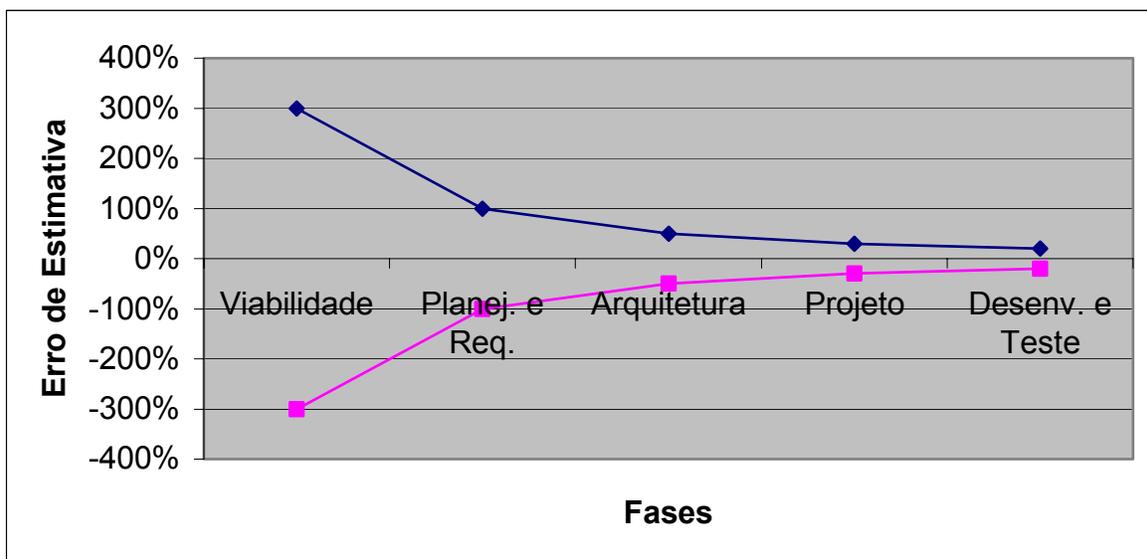


Figura 2 - Erro de Estimativa Versus Fase ([BOE 81])

Outros fatores que influenciam no processo de estimativa são o surgimento de novos processos, métodos e ferramentas de desenvolvimento [STU 96], o nível de particionamento do desenvolvimento [BRO 75] e as diferenças de produtividade individuais [SAC 68]. Tem-se ainda o trabalho de Capers Jones enumerando quarenta e cinco fatores que influenciam na produtividade de programação [JON86].

### 2.2.2 Técnicas de Estimativa

As diversas técnicas de estimativa de esforço de desenvolvimento do produto de software descritas na literatura especializada obtêm em geral uma estimativa do valor da métrica de software relacionada à quantidade de trabalho ou esforço a ser realizado. Nesse sentido, algumas destas técnicas utilizam como entrada o valor de uma métrica de software relacionada ao tamanho.

É recomendável a utilização de mais de uma técnica de estimativa a fim de comparar-

se o resultado das diferentes técnicas e possibilitando chegar a uma estimativa melhor.

- **Avaliação de Especialista**

A avaliação de especialista é a técnica mais primitiva de obter uma estimativa em um projeto de desenvolvimento do produto de software. Basicamente, solicita-se a um ou mais especialistas no tipo de software a ser desenvolvido os seus julgamentos pessoais e subjetivos de qual o tamanho e o esforço do software a ser desenvolvido.

A avaliação de especialista pode ter a sua qualidade melhorada através de técnicas de consenso de grupo como as descritas nos tópicos seguintes [BOE 81].

- **Delphi**

Requisitos detalhados são passados para especialistas para que cada um faça uma avaliação privada e produza uma estimativa. Os resultados são coletados e um resumo anônimo é produzido. Este resumo é então repassado para os especialistas que refazem suas estimativas. Um coordenador decide quando um nível suficiente de consenso foi atingido e a interação deve parar. Uma explicação mais detalhada pode ser encontrada em [PMI 96].

- **Wide Band Delphi**

Esta técnica é derivada da técnica Delphi, mas ela permite que os especialistas colaborem em questionamentos que eles possam ter em relação aos requisitos apresentados. Em seguida, os especialistas elaboram suas estimativas privadamente. Os resultados são então coletados e um sumário é produzido. O processo é repetido até o coordenador decidir que um nível suficiente consenso foi atingido. Maiores informações podem ser encontradas em [BOE 81].

- **Projeção a partir dos Dados Históricos**

Este método consiste basicamente num processo de interpolação. A partir dos dados históricos (é necessário que a organização possua tais dados) pode-se obter a relação entre diferentes métricas. Como exemplo, pode-se considerar a relação entre esforço (homens-mês) e linhas de código (LOC), obtendo-se a produtividade, ou seja, quantidade de linhas de código a ser escrita por pessoa em um mês. Tal produtividade de projetos passados pode ser utilizada para se estimar esforço para novos projetos.

- **Estimativa por Decomposição (Estimativa Bottom-Up)**

A estimativa por decomposição é normalmente utilizada em conjunto com outra técnica de estimativa como, por exemplo, avaliação de especialista ou projeção a partir de

dados históricos: o custo/esforço de cada componente de software é estimado individualmente (os componentes ou tarefas podem ser determinados, por exemplo, aproveitando-se o WBS do projeto). Em geral, a pessoa que estima é a mesma que vai desenvolver o componente. Os custos/esforços individuais são somados para chegar-se ao custo/esforço total do sistema [BOE 81].

Algumas vantagens da estimativa por decomposição segundo Barry Boehm [BOE 81] são:

- Cada estimativa vai ser baseada num entendimento mais detalhado do trabalho a ser feito.
- Cada estimativa vai ter o comprometimento pessoal de quem irá realizar o trabalho.

Esta técnica de estimativa tende a relevar custos de sistema como integração, gerência de configuração, garantia de qualidade e gerência de projeto. A fim de levar em consideração estes fatores pode-se empregar uma estimativa de sistema (Top-Down) complementando-se a estimativa por decomposição.

### **2.2.3 Modelos Algorítmicos de Estimativa**

Modelos algorítmicos são baseados em um ou mais algoritmos matemáticos que podem produzir uma estimativa de esforço como função de variáveis representando grandezas relevantes (métricas).

Um modelo algorítmico importante é o CONstructive COst MOdel (COCOMO) que foi desenvolvido por Barry W. Boehm no final dos anos 70 e descrita no seu livro clássico [BOE 81]. Diversas implementações do COCOMO continuam a ser amplamente utilizadas no mercado [STU 96]. Outro modelo algorítmico importante é o modelo SLIM desenvolvido por Putnam [PUT 78]. Trata-se de um modelo dinâmico de múltiplas variáveis que pressupõe uma distribuição de esforço específica ao longo da existência de um projeto de desenvolvimento do produto de software. O modelo foi construído a partir de uma distribuição de mão-de-obra encontrada em grandes projetos (esforço total de 30 pessoas-ano ou mais – curva Rayleigh-Norden), porém a extrapolação para projetos menores é possível [HAU 01].

## **2.3 Análise de Risco**

A análise de risco envolve basicamente três atividades distintas segundo Barry Boehm: identificação do risco, análise do risco e priorização do risco [BOE 91].

Alguns riscos são identificados e avaliados a partir da incerteza na estimativa da métrica de esforço (custo) [CAR 01]. Portanto, uma boa estimativa para o valor da métrica

de esforço diminui a probabilidade de ocorrência do risco de atraso no término do projeto de desenvolvimento, e aumenta as chances de sucesso do projeto (cumprimento dos objetivos do projeto como, por exemplo, o cronograma estabelecido).

A avaliação de risco do desenvolvimento do produto de software através da utilização de métricas pode orientar de forma quantitativa a priorização de risco.

## 2.4 Definição das Atividades

A decomposição das atividades a serem realizadas no desenvolvimento de um produto de software depende do ciclo de vida ou paradigma de desenvolvimento do produto de software utilizado (os paradigmas especificam algumas atividades que devem ser executadas, assim como a ordem em que devem ser executadas [CAR 01]). A organização destas atividades pode ser representada utilizando um WBS (Work Breakdown Structure). No exemplo da figura 3, tem-se um WBS simplificado de um projeto genérico de desenvolvimento do produto de software. O título do projeto Desenvolvimento de Software é colocado no retângulo superior. No próximo nível os retângulos representam as fases do projeto que neste caso são análise, projeto, implementação e testes, e finalmente para cada fase tem-se uma lista vertical que representam as atividades a serem executadas em cada fase embora tenham sido listados apenas alguns exemplos de atividades para a fase de análise.

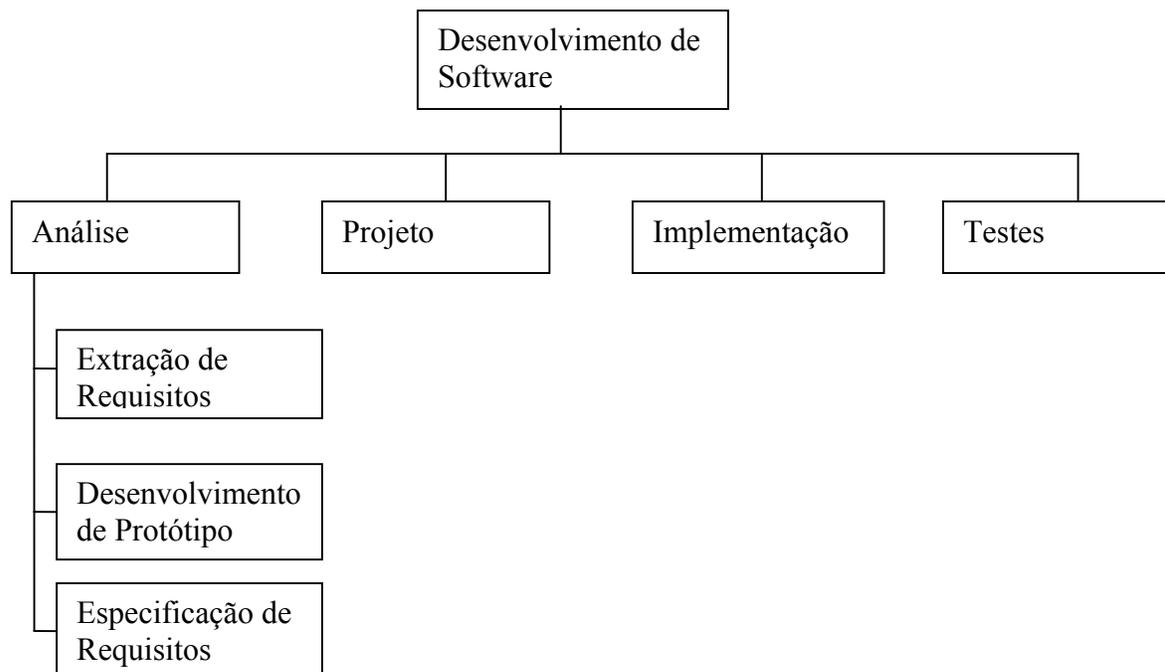


Figura 3 - Exemplo de WBS

Através da avaliação do valor da métrica de tamanho e complexidade obtida para o produto de software a ser desenvolvido, pode-se rever quais atividades realmente devem ser realizadas apesar do paradigma de desenvolvimento escolhido: sistemas pequenos (valores pequenos da métrica de tamanho e complexidade) podem permitir a eliminação de algumas atividades a fim de proporcionar uma simplificação do projeto de desenvolvimento com conseqüente diminuição de custo sem perda de qualidade. A definição sobre qual é um valor pequeno para a métrica de tamanho e complexidade depende do estabelecimento de uma política de avaliação destas métricas na organização responsável pelo desenvolvimento do produto de software. Considerando-se o exemplo da figura 3, a atividade de Desenvolvimento de Protótipo poderia não ser realizada em projetos de sistemas pequenos.

Por outro lado, a decomposição do desenvolvimento do produto de software em atividades facilita a obtenção da estimativa para métrica de esforço, pois tal estimativa é mais simples de ser realizada para atividades específicas do que para todo desenvolvimento.

## **2.5 Definição do Cronograma e Orçamento**

Uma vez definidas as atividades e estimado o valor da métrica de esforço (custo) para cada atividade, a definição do cronograma depende da relação de dependência entre as atividades e a alocação das pessoas disponíveis. No caso do orçamento, além do custo das pessoas alocadas para executar as diferentes atividades, pode-se utilizar o valor obtido para a métrica de tamanho do sistema a fim de avaliar o custo dos equipamentos, materiais e instalações necessárias.

# Capítulo 3

## Métricas de Software

As métricas de software são definidas como uma grandeza que se refere ao produto ou processo e que pode ser medida direta ou indiretamente dentro do contexto do desenvolvimento do produto de software. Uma das razões principais de coletar dados sobre uma determinada métrica é definir uma base de dados a ser utilizada no processo de estimativa. Além disso, a coleta de métricas durante o andamento do projeto de desenvolvimento do produto de software permite o acompanhamento deste projeto. A seguir serão discutidas algumas métricas de interesse para este trabalho.

### 3.1 A Métrica de Pontos de Função

A contagem de pontos de função como métrica para definir o tamanho funcional de um sistema (aplicação) foi inicialmente formulado por Albrecht [ALB 83] como resultado de um projeto na empresa IBM no início dos anos da década de 1980 [GAR 01].

Em 1986 foi criado o IFPUG (International Function Point User Group). Esta instituição não tem fins lucrativos e é gerida pelos seus membros com a missão de promover e encorajar o gerenciamento efetivo do desenvolvimento e manutenção do software de aplicação através da utilização da análise de pontos de função e outras técnicas de medida de software [GAR 01]. Além de formular um manual para a contagem de pontos de função (padronizando o método de contagem), o IFPUG certifica profissionais na realização de contagem de pontos de função.

A idéia fundamental da análise por ponto de função consiste em medir o tamanho de qualquer produto de software baseado em termos lógicos, orientados ao usuário. A análise por ponto de função não se preocupa diretamente com a plataforma tecnológica, ferramentas de desenvolvimento e linhas de código. Ela simplesmente mede a funcionalidade entregue ao usuário final. Nesse sentido, a métrica de pontos de função de acordo com o IFPUG avalia o produto de software e mede o seu tamanho baseando-se em características funcionais bem definidas deste sistema.

A fronteira da aplicação ou do sistema em questão é estabelecida a partir do ponto de vista do usuário final e é fundamental para a métrica dos pontos de função.

A métrica dos pontos de função contabiliza:

- Dados que estão entrando no sistema (EI);
- Dados que estão saindo do sistema (EO);

- Consulta externa (EQ - relatórios, apresentação de dados);
- Dados que são produzidos e armazenados dentro do sistema (ILF);
- Dados que são mantidos fora do sistema, mas que são necessários para satisfazer um requisito de processo (EIF).

O processo de contagem primeiramente deriva a contagem de pontos de função não ajustados (UFP) e depois avalia o fator de ajuste de contagem (VAF) para finalmente ser possível calcular a contagem de pontos de função ajustada (FP).

Os seguintes conceitos básicos são utilizados na métrica de pontos de função para definir a complexidade (quantidade de pontos de função não ajustados) de uma transação (EI, EO ou EQ) ou de um arquivo (ILF ou EIF) da aplicação [LON 02a]:

- **Tipo do Elemento de Registro (RET)** consiste num subgrupo de elementos de dados identificáveis pelo usuário do sistema.
- **Tipo de Arquivo Referenciado (FTR)** consiste em um tipo de arquivo referenciado por uma transação.
- **Tipo do Elemento de Dados (DET)** consiste num campo único não recursivo reconhecido pelo usuário. A informação do DET é dinâmica ou não estática. Se um DET é recursivo, apenas a primeira ocorrência do DET é considerada e não toda a ocorrência.

### 3.1.1 Contagem dos Pontos de Função Não Ajustados

A utilização dos conceitos apresentados é mostrada em acordo com a versão 4.1 do manual de contagem do IFPUG [GAR 01]. A contagem de pontos de função não ajustados (UFP), associada a cada EI, EO, EQ, ILF e EIF, representa a complexidade de cada um destes elementos. A fim de determinar-se tal complexidade, avalia-se o número de RET e DET para ILF e EIF, e de FTR e DET para EI, EO e EQ.

- **Entrada Externa (EI)**

Serão contabilizadas como entradas externas as transações nas quais dados cruzam a fronteira definida de fora para dentro. Estes dados vêm de uma tela de entrada de dados ou de uma outra aplicação. Os dados podem ser tanto informação de controle ou informação do negócio. Se o dado for informação de negócio, ele será usado para manter um ou mais arquivos lógicos internos (ILF). Se o dado for informação de controle, ele não necessariamente vai atualizar um ILF.

A avaliação da complexidade resulta no número de pontos de função não ajustados representado por uma determinada EI. Esta avaliação considera a quantidade de arquivos referenciados (alterados) por esta EI e da quantidade de elementos de dados que compõe a

EI. Tal relação é mostrada na tabela 1.

Arquivos Referenciados (FTR)	Elementos de Dados (DET)		
	1 a 4	5 a 15	Maior que 15
Menor que 2	Baixo (3)	Baixo (3)	Médio (4)
2	Baixo (3)	Médio (4)	Alto (6)
Maior que 2	Médio (4)	Alto (6)	Alto (6)

**Tabela 1 - Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma EI [LON 02 a]**

Nesse sentido, por exemplo, se uma determinada EI é composto de 6 elementos de dados (DET) e referência 2 arquivos (FTR), terá complexidade média e deverão ser contabilizados 4 pontos de função (número entre parêntese na tabela) não ajustados relativos a esta EI.

- **Saída Externa (EO)**

Serão contabilizadas como saídas externas as transações elementares nas quais dados derivados cruzam a fronteira de dentro para a fora. Estes dados representam relatórios ou arquivos de saída enviados para o usuário ou outras aplicações. Estes relatórios ou arquivos são criados a partir de um ou mais arquivos lógicos internos (ILF) e/ou arquivos de interface externa (EIF).

Dados Derivados são dados processados além da leitura direta e edição da informação contida em arquivos lógicos internos (ILF) ou arquivos de interface externa (EIF). Dados derivados são o resultado de algoritmos e/ou cálculos. Dados derivados ocorrem quando um ou mais elementos de dados são combinados através de uma fórmula para gerar ou derivar elementos de dados adicionais.

Arquivos Referenciados (FTR)	Elementos de Dados (DET)		
	1 a 5	6 a 19	Maior que 19
Menor que 2	Baixo (4)	Baixo (4)	Médio (5)
2 ou 3	Baixo (4)	Médio (5)	Alto (7)
Maior que 3	Médio (5)	Alto (7)	Alto (7)

**Tabela 2 - Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma EO [LON 02 a]**

A avaliação da complexidade resulta no número de pontos de função não ajustados representado por uma determinada EO. Esta avaliação considera a quantidade de arquivos

referenciados (lidos) por esta EO e da quantidade de elementos de dados que compõe a EO. Tal relação é mostrada na tabela 2.

Nesse sentido, por exemplo, se uma determinada EO é composto de 4 elementos de dados (DET) e referência 1 arquivo (FTR), terá complexidade baixa e deverão ser contabilizados 4 pontos de função não ajustados relativos a esta EO.

- **Consulta Externa (EQ)**

Serão contabilizadas como consultas externas as transações elementares compostas simultaneamente por entrada e saída que resulta na retirada de dados de uma ou mais arquivos lógicos internos e arquivos de interface externos. Esta informação é enviada para fora da fronteira da aplicação em questão. Os componentes de entrada não atualizam qualquer arquivo lógico interno e os componentes de saída não contém dados derivados.

É importante ressaltar que a diferença básica entre um EO e um EQ está no fato do EQ não possuir dados derivados ao contrário do EO.

A avaliação da complexidade resulta no número de pontos de função não ajustados representado por uma determinada EQ. Esta avaliação considera a quantidade de arquivos referenciados (lidos) por esta EQ e da quantidade de elementos de dados que compõe a EQ. Tal relação é mostrada na tabela 3.

Arquivos Referenciados (FTR)	Elementos de Dados (DET)		
	1 a 5	6 a 19	Maior que 19
Menor que 2	Baixo (3)	Baixo (3)	Médio (4)
2 ou 3	Baixo (3)	Médio (4)	Alto (6)
Maior que 3	Médio (4)	Alto (6)	Alto (6)

Tabela 3 - Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma EQ [LON 02 a]

Nesse sentido, por exemplo, se uma determinada EQ é composto de 20 elementos de dados (DET) e referência 3 arquivos (FTR), terá complexidade alta e deverão ser contabilizados 6 pontos de função não ajustados relativos a esta EO.

- **Arquivos Lógicos Internos (ILF)**

Serão contabilizados como arquivos lógicos internos os grupos de dados relacionados logicamente (identificáveis pelo usuário) que residem completamente dentro da fronteira do sistema e são mantidos através de entradas externas.

A avaliação da complexidade resulta no número de pontos de função não ajustados

representado por um determinado ILF. Esta avaliação considera a quantidade de RET's e DET's contidos neste ILF. Tal relação é mostrada na tabela 4.

Tipo de Registro (RET)	Elementos de Dados (DET)		
	1 a 19	20 a 50	Maior que 50
1	Baixo (7)	Baixo (7)	Médio (10)
2 a 5	Baixo (7)	Médio (10)	Alto (15)
Maior que 6	Médio (10)	Alto (15)	Alto (15)

**Tabela 4 - Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma ILF [LON 02 a]**

Nesse sentido, por exemplo, se um determinado ILF é composto de 4 elementos de dados (DET) subdivididos em 1 único tipo de registro (RET), terá complexidade baixa e deverão ser contabilizados 7 pontos de função não ajustados relativos a este ILF.

- **Arquivo de Interface Externa (EIF)**

Serão contabilizados como arquivos de interface externa os grupo de dados relacionados logicamente (identificáveis pelo usuário) que são usados somente para referência e residem completamente fora da fronteira da aplicação além de serem mantidos por outra aplicação. O arquivo de interface externa é arquivo lógico interno de outra aplicação.

A avaliação da complexidade resulta no número de pontos de função não ajustados representado por um determinado EIF. Esta avaliação considera a quantidade de RET's e DET's contidos neste EIF. Tal relação é mostrada na tabela 5.

Tipo de Registro (RET)	Elementos de Dados (DET)		
	1 a 19	20 a 50	Maior que 50
1	Baixo (5)	Baixo (5)	Médio (7)
2 a 5	Baixo (5)	Médio (7)	Alto (10)
Maior que 6	Médio (7)	Alto (10)	Alto (10)

**Tabela 5 - Complexidade (Número de Pontos de Função Não-ajustados) Associada a uma EIF [LON 02 a]**

Nesse sentido, por exemplo, se um determinado EIF é composto de 4 elementos de dados (DET) subdivididos em 1 único tipo de registro (RET), terá complexidade baixa e deverão ser contabilizados 5 pontos de função não ajustados relativos a este EIF.

### 3.1.2 Ajustando a Contagem

A somatória dos pontos de função não ajustados obtidos avaliando-se a complexidade

dos EI, EO, EQ, ILF e EIF da aplicação são ajustados através do **fator de ajuste de valor (VAF)** que é baseado em 14 características gerais de sistema (GSC) e trata-se de um fator multiplicador. Cada vez mais organizações não estão usando o VAF, procurando então usar os pontos de função não ajustados diretamente [LON 02a]. A razão de tal atitude, além da maior simplicidade na obtenção da contagem dos pontos de função, consiste no fato da homogeneidade no ambiente de desenvolvimento transformar o valor do VAF numa constante [LON 02a]. Também se pode argumentar contra a utilização do VAF que as características consideradas são voltadas ao ambiente de mainframe e não ao ambiente de PC, cliente-servidor ou Web encontrado hoje [GAR 01].

Para cada característica geral do sistema é atribuído um valor de zero a cinco ( $C_i$ ), representando o grau de influência desta característica no projeto a ser executado:

0 Não presente, ou sem influência.

1 Influência eventual

2 Influência moderada

3 Influência mediana

4 Influência significativa

5 Influência grande

A seguir descreve-se brevemente cada característica de acordo com [GAR 01]:

- **Comunicação de Dados** representa quantas funcionalidades de comunicação são implementadas pela aplicação. O valor zero poderia ser atribuído a uma aplicação batch pura ou isolada num PC, enquanto o valor cinco seria uma aplicação com mais de uma interface com o usuário e mais de um protocolo para teleprocessamento.
- **Processamento Distribuído de Dados** representa o quão distribuído é o processamento da aplicação em questão. A maioria das aplicações recebe zero neste item, mas caso aplicação execute em múltiplos servidores ou processadores com disponibilidade dinâmica, pode-se atribuir o valor cinco.
- **Desempenho** representa a quantidade de requisitos associados a tempo de resposta e capacidade de processamento (throughput) na aplicação em questão. O valor zero é atribuído a ausência de requisitos de desempenho enquanto o valor cinco implica na utilização de ferramenta de medição de desempenho durante o projeto e implementação.
- **Utilização Intensiva da Configuração** representa o quão intensamente a configuração de hardware existente vai ser utilizada pela aplicação em questão. O valor zero é atribuído a ausência de restrições de hardware enquanto o valor cinco significa restrições de execução em um ambiente de

processamento distribuído.

- **Taxa de Transação** representa a influência da frequência de transações no projeto da aplicação. O valor é atribuído a ausência de pico nas transações enquanto que cinco significa a utilização de ferramentas de avaliação de desempenho no projeto e implementação da aplicação para garantir que os requisitos de frequência e pico de transação sejam atendidos.
- **Entrada de Dados On-line** representa o percentual de transações interativas para a aplicação em questão. O valor zero somente é atribuído às aplicações totalmente batch enquanto o valor cinco indica que mais de 30% de todas as transações são interativas (a maioria das aplicações se encaixa nesta categoria).
- **Eficiência para o Usuário Final** representa a preocupação com a usabilidade da aplicação em questão pelo usuário final. Aplicações puramente batch receberiam valor zero enquanto o valor cinco está associado à utilização de testes e ferramentas para garantir um nível de qualidade desejado na usabilidade da aplicação.
- **Atualização On-line** representa quantos arquivos lógicos internos são atualizados on-line. As aplicações on-line normalmente recebem nota 3, mas se existe um sistema de recuperação automático para perda de dados através de Back up's preventivos, a aplicação deve receber 4 ou 5.
- **Processamento Complexo** representa o quão complexo é o processamento da aplicação em questão. Fatores como, por exemplo, operações matemáticas complexas, tratamento de entradas multimídia devem ser considerados nesta avaliação.
- **Reusabilidade** representa o quão reutilizável será o código sendo desenvolvido, ou seja, quanto maior o número de outras aplicações que estarão usando o código desenvolvido, maior a pontuação.
- **Facilidade de Instalação** o quão complexa é a instalação ou conversão da aplicação desenvolvida para outros ambientes. Quanto mais complexo maior a pontuação.
- **Facilidade de Operação** representa o quão autônoma a aplicação em questão é autônoma para a realização de tarefas complexas como inicialização, backup e recuperação de falhas. Quanto mais autônoma maior a pontuação.
- **Múltiplas Localidades** representa a aplicação ter sido desenvolvido para múltiplas localidades ou organizações. Zero representa a ausência desta característica enquanto a pontuação vai aumentando conforme a maior

diferença entre as localidades e organizações envolvidas em termos de ambiente de software e hardware.

- **Facilidade de Mudança** representa o grau de flexibilidade implementado para alteração do comportamento da aplicação em questão.

Considerando-se estas 14 características, utiliza-se a seguinte fórmula para obtenção do fator de ajuste de valor (VAF):

$$\text{VAF} = 0,65 + \sum C_i (i \text{ variando de } 1 \text{ a } 14) / 100$$

A constante 0,65 é uma constante empírica, portanto não é derivada de nenhuma outra fórmula.

Nesse sentido, o VAF pode representar uma variação de mais ou menos 35% na conversão da contagem de pontos de função não ajustados (UFP) para a contagem ajustada (FP) já que a fórmula de obtenção dos pontos de função ajustados (FP) é a seguinte:

$$\text{FP} = \text{UFP} * \text{VAF}$$

### 3.1.3 Guia de Contagem

Resumidamente, pode-se listar os seguintes passos para a contagem de pontos de função:

1) Identificação das fronteiras do sistema, ou seja, é necessário definir o escopo do sistema em questão.

2) Fazer os levantamentos dos itens correspondentes aos pontos de função de dados (ILF e EIF), identificar a complexidade de cada item levantado no item anterior de acordo com tabelas de complexidade que podem ser encontradas em [GAR 01]. Tal complexidade determina o número de pontos de função correspondentes ao item.

3) Fazer os levantamentos dos itens correspondentes aos pontos de função transacionais (EI, EO e EQ), identificando a complexidade de cada item levantado de acordo com tabelas de complexidade que podem ser encontradas em [GAR 01]. Tal complexidade determina o número de pontos de função correspondentes ao item.

4) Somar o número de pontos de função correspondente a cada item encontrado nos passos 2 e 3, obtendo o total de pontos de função não ajustados (UFP).

5) Determinar o fator de ajuste de valor (VAF).

6) O cálculo dos pontos de função ajustados (FP) dá-se através da fórmula:

$$\text{FP} = \text{UFP} * \text{VAF}$$

Existem aproximadamente 40 métodos diferentes de Análise de Pontos de Função hoje em dia [MAX 00]. Este texto aborda a técnica mais tradicional baseada no manual de contagem do IFPUG.

Uma variação interessante e muito usada da contagem de pontos de função

tradicional (IFPUG) é mostrada em [LON 02c]. Esta variação tem um caráter simplificador, pois, quando é difícil avaliar a complexidade dos EI's, EO's, EQ's, ILF's e EIF's, pode-se assumir um valor médio para a complexidade contido nas tabelas 1, 2, 3, 4 e 5. Dessa forma a contagem de pontos de função não ajustados (UFP) seria dada por:

$$UFP = (\Sigma EI's) \times 4 + (\Sigma EO's) \times 5 + (\Sigma EQ's) \times 4 + (\Sigma ILF's) \times 10 + (\Sigma EIF's) \times 7$$

### 3.1.4 Pontos de Função e Casos de Uso

Casos de Uso representam diálogos entre os usuários e o produto de software, sob o ponto de vista do usuário, ou seja, casos de uso descrevem como o usuário interage com um produto de software [LON 02b]. Neste contexto o usuário é chamado de Ator.

Aplicar a contagem de pontos de função a casos de uso é relativamente simples, pois casos de uso representam requisitos enquanto ponto de função é uma métrica que considera o software sobre a perspectiva de requisitos [LON 02c].

Se casos de uso são definidos no início do desenvolvimento e a contagem de pontos de função é realizada, pode-se obter estimativas de tamanho e esforço mais precisas logo no início do desenvolvimento da aplicação [LON 02c].

Como os casos de uso representam as transações de uma determinada aplicação, eles poderiam ser utilizados no lugar do layout de interface para a contagem de EI, EQ e EO [LON 02b].

A definição dos limites da aplicação é o primeiro passo na contagem de pontos de função e pode ser derivada dos casos de uso considerando a fronteira entre o sistema e seus atores [LON 02c].

Normalmente o tamanho máximo de um caso de uso não deve ser maior que 50 pontos de função segundo David Longstreet [LON 02c]. A aplicação da contagem de pontos de função melhora a qualidade dos casos de uso porque ajuda a determinar se o caso de uso está no nível correto de detalhe. Não é possível realizar a contagem de pontos de função se o caso de uso não está no nível correto de detalhe, ou seja, o caso de uso precisa descrever como os dados passam a fronteira do sistema em questão no sentido do ator ou vindo dele [LON 02c].

As seguintes definições foram adaptadas especificamente para a obtenção da contagem de pontos de função através casos de uso [LON 02c], mas elas estão de acordo com o IFPUG.

#### **Transações**

- Um ou mais passos no fluxo de eventos definem a transação.
- Um único caso de uso pode ter uma ou mais transações.

- Cenários secundários ajudam a clarificar as transações.
- As transações são entradas externas (EI), saídas externas (EO) ou consultas externas (EQ).

*Entradas Externas (EI)* – os dados vêm de um ator externo ao sistema, e caso seja informação de controle não necessariamente será usada para manter um ILF.

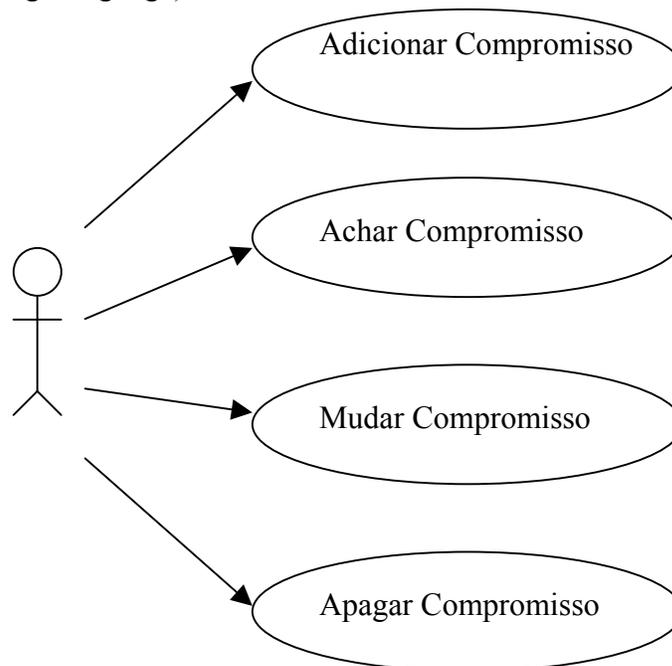
*Saídas Externas (EO)* – **dados derivados** são enviados para um ator externo ao sistema, e este ator pode manter um ILF com os dados recebidos.

*Consultas Externas (EQ)* – um ator solicita dados e recebe os dados contidos em um ou mais ILF e/ou EIF.

### Arquivos

Através do modelo lógico de dados (projeto da base de dados), os *Arquivos Lógicos Internos (ILF)*, que são internos ao sistema e mantidos pelos dados dos EI's, e *Arquivos de Interface Externos (EIF)*, que são externos ao sistema e mantidos por outros sistemas sendo usados apenas para referência pelo sistema em questão, podem ser facilmente identificados. Frequentemente, casos de uso incluem uma seção Casos de Usos Subordinados que inclui uma listagem de todos os casos de uso da base de dados.

Na figura 4 tem se um exemplo de diagrama de casos de uso seguindo o padrão UML (Universal Modeling Language).



**Figura 4 - Diagrama de Casos de Uso**

O diagrama de figura 4 mostra como um Ator pode manter seus compromissos numa

aplicação de Agenda de Compromissos. Os casos de uso Adicionar Compromissos; Mudar Compromisso; e Apagar Compromisso representam três Entradas Externas. Achar Compromisso é uma Consulta Externa. Neste nível de detalhe é impossível designar o número correto de Elementos de Dados e Arquivos Referenciados. Sendo assim, pode-se simplificar a contagem classificando-se todas as transações como tendo complexidade média. Portanto uma estimativa do número de pontos de função não ajustados seria:

$$(3 \text{ EI's} \times 4) + (1 \text{ EQ} \times 4) = 16 \text{ pontos de função não ajustados}$$

É importante ressaltar o cuidado que deve ser tomado com a falta de detalhe num diagrama de casos de uso fazendo com que mais de uma transação (EI, EO ou EQ) seja incluída no mesmo caso de uso [LON 02c].

## 3.2 Outras Métricas

A seguir são descritas algumas métricas relevantes para o software de dispositivos portáteis.

- **Homem-Mês**

Geralmente os projetos de desenvolvimento têm no pagamento de profissionais que participam destes projetos a sua principal fonte de despesa, e este pagamento é proporcional ao tempo (esforço) dedicado aos projetos. Nesse sentido, esta métrica pode ser utilizada para medir a quantidade de trabalho que foi despendida pelos profissionais envolvidos num determinado projeto ou atividade deste projeto por unidade de tempo, ou seja, trata-se de uma métrica de esforço. Concluindo, a definição da métrica consiste em um mês de trabalho de um profissional (quantidade de pessoas por unidade de tempo). É importante observar que o custo financeiro do tempo de trabalho de cada profissional pode variar de acordo com fatores como nível de especialização e tempo de carreira.

- **Linhas de Código – LOC**

Esta métrica talvez seja a métrica mais básica dentro do universo do desenvolvimento do produto de software, pois se trata da contagem do número de linhas do código-fonte compiladas ou interpretadas para produzir o software executável (principal produto resultante de um projeto de desenvolvimento do produto de software).

- **Quantidade de Memória**

Esta métrica consiste na quantidade de bytes utilizados pelo software, seja como RAM (random access memory) ou como ROM (read only memory). Tal métrica é muito

importante para se avaliar a viabilidade de projetos de software para dispositivos portáteis como telefones celulares devido à limitação na quantidade de memória ROM e RAM disponíveis.

### **3.3 Discussão sobre Métricas de Tamanho Utilizadas para Estimativas de Esforço**

Considerando-se a utilização de diferentes métricas relacionadas a tamanho de software na obtenção de estimativas iniciais de esforço, pode-se levantar vantagens e desvantagens na utilização de cada uma delas, o que será discutido a seguir.

#### **3.3.1 LOC**

A dificuldade de aplicação da métrica LOC consiste justamente em definir o método de contagem das linhas de código a fim de possibilitar a formação de uma base de dados consistente. Caper Jones [JON 86] e Graham Low [LOW 90] relacionam 11 maneiras de se definir a contagem de linhas de código, além do caso especial de como considerar linhas apagadas. As variações de contagem são subdivididas em dois grupos.

O primeiro grupo de variações na contagem de linhas de código refere-se ao programa:

- 1) Conta-se somente linhas executáveis.
- 2) Conta-se linhas executáveis e definição de dados.
- 3) Conta-se linhas executáveis, definição de dados e comentários.
- 4) Conta-se linhas executáveis, definição de dados, comentários e Job Control

Language.

- 5) Conta-se linhas como linhas físicas na tela de entrada.
- 6) Conta-se linhas como terminadas por delimitadores lógicos.

Jones afirma ser possível uma diferença de até 5 vezes entre a contagem mais compacta e a menos compacta.

O segundo grupo de variações na contagem de linhas de código refere-se ao tipo de projeto:

- 1) Conta-se somente novas linhas.
- 2) Conta-se novas linhas e linhas alteradas.
- 3) Conta-se novas linhas, linhas alteradas e linhas reusadas.
- 4) Conta-se todas as linhas entregues e linhas temporárias (um exemplo de linhas temporárias seria as linhas usadas para instrumentar o código original facilitando a sua depuração ou medição de desempenho).
- 5) Conta-se todas as linhas entregues, linhas temporárias e linhas de suporte.

Segundo Graham Low, variações de contagem ao nível de projeto são maiores quando o programa sendo desenvolvido não é completamente novo, ou seja, está se adicionando ou modificando funcionalidades a um software já existente [LOW 90]. Também se pode citar o caso do desenvolvimento do produto de software que se utiliza componentes genéricos já prontos.

Pode –se considerar como vantagem na utilização da métrica de linhas de código a existência de diversos modelos de estimativa baseados em linhas de código (COCOMO e SLIM já citados na seção 2.2.3) e de uma grande base de dados histórica baseada em linhas de código [PRE 92]. Por outro lado, pode-se listar várias desvantagens, além da grande variação nos métodos de contagem já citada. Por exemplo, a quantidade de linhas de código é dependente da linguagem de programação; esta métrica penaliza programas bem projetados, portanto menores; o processo de medição não acomoda facilmente linguagens não procedimentais; é necessário um elevado nível de detalhe para utilizar-se a métrica como entrada para a atividade de estimativa [PRE 92].

Recentemente tem se ainda a utilização de **linhas de código equivalentes** (SLOC) como uma tentativa de normalizar a contagem de linhas de código a partir da quantidade de instruções executáveis efetivamente geradas para um determinado software. Obviamente a obtenção desta métrica depende da existência de uma ferramenta para cada linguagem/ambiente.

### 3.3.2 Quantidade de Memória

A métrica de memória não é normalmente utilizada como entrada na obtenção de estimativas de esforço iniciais embora seja possível a sua utilização a partir de dados históricos através dos quais seja possível relacionar quantidade de memória e esforço. Uma limitação em relação à coleta de dados históricos sobre quantidade de memória consiste no fato desta métrica depender não somente da linguagem utilizada, mas também do ambiente alvo (sistema operacional, processador), ou seja, os dados históricos se tornam específicos de uma determinada configuração computacional.

### 3.3.3 Ponto de Função

A métrica de pontos de função não é uma métrica trivial para a obtenção do seu valor: a contagem dos pontos de função exige treinamento a fim de se aprender todas as regras de um manual de contagem (como o manual do IFPUG [GAR 01]) para que a contagem possa ser um processo reproduzível.

Entretanto, as vantagens da métrica de pontos de função foram consolidadas por [FUR 97] de acordo com os itens abaixo.

- **Independência da Tecnologia:** pontos de função medem o tamanho funcional

sem dependência em relação à ferramenta de desenvolvimento ou tecnologia utilizada. Eles focam na funcionalidade requisitada pelo usuário e implementada pelo software, portanto servem como uma medida indireta dos requisitos de software.

- **Consistência e Repetibilidade:** a introdução do manual de contagem do IFPUG [GAR 01] permitiu uma contagem consistente entre os contadores que utilizam este manual além de permitir a sua repetibilidade uma vez que se padronizou o processo e a documentação da contagem. Um estudo de 1994 do Quality Assurance Institute e do IFPUG apontou uma variação relativamente pequena (em torno de 11% segundo S. Furey [FUR 97]) no resultado de contagem entre contadores treinados com o manual do IFPUG.
- **Normalização dos Dados:** pontos de função permitem que as organizações normalizem dados relativos a defeitos e tamanho. Exemplificando a normalização dos dados de defeitos, um sistema A que apresentou 640 defeitos e que implementa 800 pontos de função tem uma densidade de defeitos de 0,8 defeito por ponto de função, enquanto um sistema B que apresentou 700 defeitos e que implementa 2.200 pontos de função tem uma densidade de defeitos de somente 0,32 defeito por ponto de função. Se fosse considerado somente o número de defeitos para analisar a qualidade, o sistema A teria uma qualidade melhor que a do B, mas se for considerado a densidade de defeitos, conclui-se que a qualidade do sistema B é melhor que a do A.
- **Estimativas Precoces:** como os pontos de função são baseados nos requisitos funcionais, eles podem ser estimados mais cedo e com mais precisão que linhas de código (LOC), ajudando conseqüentemente obtenção precoce de estimativas de esforço e o planejamento do projeto de desenvolvimento do produto de software.
- **Comparações:** o esforço normalizado de acordo com a quantidade de pontos de função pode ajudar a comparar diferentes tecnologias e/ou processos de desenvolvimento em termos de produtividade. Por exemplo, se uma tecnologia/processo A demanda em média 0,5 homem-mês para implementação de um ponto de função enquanto uma tecnologia/processo B demanda em média 1 homem-mês para a implementação de um ponto de função, pode-se concluir que a tecnologia/processo A favorece a produtividade em relação a tecnologia/processo B.
- **Escopo e Expectativas:** as requisições de clientes relativas a mudanças ou

melhorias que tenham um impacto significativo no projeto de desenvolvimento do produto de software podem ser mais bem avaliadas e planejadas na medida que se quantifica o impacto da mudança ou melhoria. Através da quantidade de pontos de função e dados de produtividade em termos de ponto de função, pode-se chegar ao impacto da mudança ou melhoria em termos de esforço (custo).

## Capítulo 4

# Pontos de Função em Aplicações para Dispositivos Portáteis

Inicialmente serão apresentados os conceitos básicos associados às aplicações de interface gráfica para dispositivos portáteis (telefones celulares) para então se fazer um paralelo entre as aplicações de interface gráfica para computadores em geral e as aplicações de interface gráfica para dispositivos pessoais no que se refere à contagem de pontos de função.

### 4.1 Conceitos Básicos para Aplicações Gráficas de Dispositivos Portáteis

A seguir são apresentados os termos básicos a serem usados para se descrever uma aplicação gráfica de dispositivo portátil (telefone celular) conforme a interface dos telefones celulares Motorola V120 e V60 [MOT 03].

- **Aplicação:** informação ou serviço que o usuário pode acessar, como, por exemplo, Agenda de Telefones, Mensagens ou Lista de Chamadas Recentes. O telefone celular tem como aplicação básica o estabelecimento de chamadas telefônicas, mas as aplicações citadas anteriormente diversificam a sua utilização e estão se tornando cada vez mais numerosas.
- **Menu Sensível ao Contexto:** lista de itens que se aplica a uma situação específica quando a tecla de Menu é pressionada.
- **Softkey:** tecla sem rótulo cuja função muda com o contexto e cujo título é mostrado em uma área específica do display.

Na figura 5 é apresentado um telefone celular genérico no qual alguns dos conceitos apresentados podem ser fisicamente localizados através da divisão do aparelho em zonas físicas/funcionais.

- **Zona 1:** constituída de um display e das teclas softkey e menu. Estas teclas devem se situar o mais próximo possível do display, pois o seu significado dependerá do que é mostrado na região do display logo acima a estas teclas – basicamente é mostrado um rótulo representando o significado da tecla.



Figura 5 - Telefone Celular (Dispositivo Portátil)

- **Zona 2:** possui três tipos de teclas que são explicadas abaixo.
  - Teclas de navegação (scroll UP/DOWN) que são utilizadas para fazer a rolagem do conteúdo (navegação vertical) do display no caso de estar sendo exibido uma lista ou um texto longo.
  - Tecla Send que é utilizada para iniciar chamadas quando uma entrada da agenda de telefones está selecionada ou um número de telefone foi digitado e está mostrado no display.
  - Tecla End que é utilizada para finalizar chamadas ou para transferi-las sem atendê-las.
- **Zona 3:** composta de várias teclas que podem ser classificadas em dois tipos explicados abaixo.
  - Teclas numéricas que são usadas principalmente para entradas de números, mas também podem ser usadas para entrada de caracteres alfanuméricos.
  - Teclas # e \* que são usadas para navegação horizontal ou para entrada dos caracteres # e \* que possuem significado especial em números de telefone.
- **Zona 4:** consiste basicamente na tecla de Power que é utilizada para ligar e

desligar o telefone.

- **Zona 5:** esta zona pode conter basicamente teclas relacionadas a aplicações específicas como, por exemplo, Ajuste de Volume, Agenda de Telefones ou Mensagens.

## 4.2 Elementos Básicos de Interface Gráficas de Dispositivos Portáteis

Será descrito brevemente, como exemplo, os principais elementos de interface gráfica dos telefones celulares Motorola V120 e V60 [MOT 03]. As figuras utilizadas neste tópico representam exemplos genéricos de cada elemento.

### 4.2.1 Elementos de Entrada de Dados

Os principais elementos de interface gráfica dos telefones celulares Motorola V120 e V60 [MOT 03] utilizados para a entrada de dados são brevemente descritos nos tópicos seguintes.

- **Formulário:** utilizado para a entrada de dados em múltiplos campos relacionados (figura 6). Um exemplo de utilização é o armazenamento dos dados relativos a um compromisso como título, data, horário e outros na aplicação de agenda de compromissos.

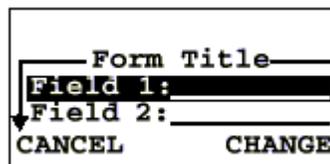


Figura 6 - Formulário

- **Editor:** permite a entrada de um dado numérico ou textual (figura 7). Um exemplo de utilização é o armazenamento do nome de uma pessoa na aplicação de agenda de telefones.

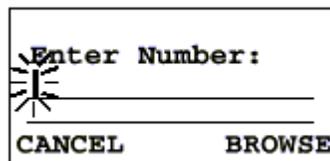


Figura 7 - Editor

- **Lista Marcada:** utilizada para a seleção de um único item entre vários (figura 8). Um exemplo de utilização é a seleção do período de antecedência do alarme de um compromisso na aplicação de agenda de compromissos.

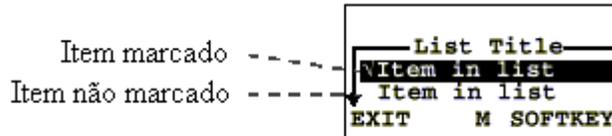


Figura 8 - Lista Marcada

- **Lista de Seleção Múltipla:** similar à lista marcada, mas diferentemente da lista marcada, permite a seleção simultânea de vários itens (figura 9). Um exemplo de utilização é a seleção de múltiplos destinatários contidos na agenda de telefones para receberem uma mensagem de texto – aplicação de envio de mensagens.

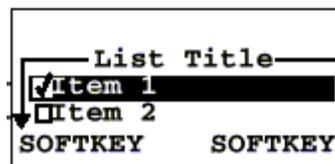


Figura 9 - Lista de Seleção Múltipla

- **Medidor Ajustável:** permite o ajuste do valor de uma determinada grandeza dentro de um intervalo definido (figura 10). Um exemplo de utilização é o ajuste do volume da campanha do aparelho - neste exemplo o título do medidor ajustável seria Volume.

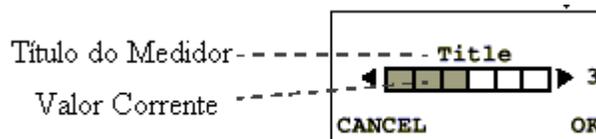


Figura 10 - Medidor Ajustável

- **Pickers:** permitem o ajuste (escolha) de uma ou mais grandezas dentro de um intervalo definido. Um exemplo de picker é mostrado na figura 11 e um exemplo de utilização poderia ser a escolha do período de tempo a partir do qual mensagens de texto com esta idade seriam automaticamente apagadas.
- **Notice de Questionamento:** é emitido basicamente para obter confirmação sobre uma ação irrecuperável (figura 12). Um exemplo de utilização é quando o usuário decide apagar uma mensagem.

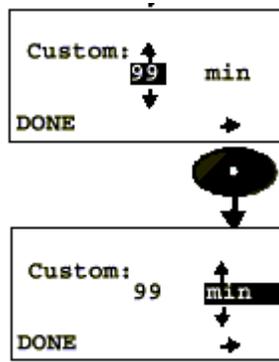


Figura 11 - Picker

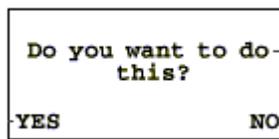


Figura 12 - Notice de Questionamento

#### 4.2.2 Elementos de Saída de Dados

Os principais elementos de interface gráfica dos telefones celulares Motorola V120 e V60 [MOT 03] utilizados para a saída de dados são brevemente descritos nos tópicos seguintes.

- **Lista:** apresentação vertical de um conjunto de itens na qual cada item pode ocupar apenas uma linha (visão resumida) ou várias linhas (visão detalhada). A figura 13 apresenta um exemplo de lista na visão resumida. Um exemplo de utilização é a lista de mensagens recebidas pelo telefone na aplicação de recebimento de mensagens, ou lista de contatos na aplicação de agenda de telefone.

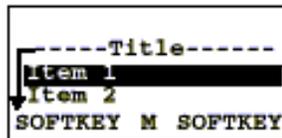


Figura 13 - Lista

- **Notice de Informação:** é usado basicamente para informar o usuário de um evento ou prover retorno sobre uma ação (figura 14). Um exemplo de utilização é a informação da chegada de uma nova mensagem (aplicação de recebimento de mensagens), ou alarme de ocorrência de um determinado compromisso (aplicação agenda de compromissos).

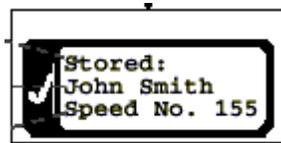


Figura 14 - Notice de Informação

- **Visualizador:** mostra informações de textual e de imagem para o usuário (figura 15). Um exemplo de utilização é a apresentação do conteúdo de uma mensagem recebida pelo telefone – aplicação de recebimento de mensagens.

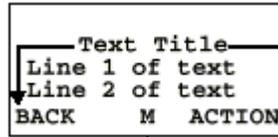


Figura 15 - Visualizador

- **Medidor:** mostra graficamente o valor de uma determinada grandeza dentro de um intervalo definido (figura 16). Um exemplo de utilização é a apresentação da carga corrente da bateria do telefone (o título seria Bateria).

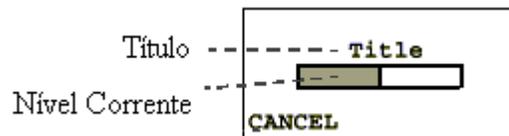


Figura 16 - Medidor

### 4.3 Contagem de Pontos de Função para Aplicações GUI

As regras e afirmações contidas nesta seção se baseiam nas definições do IFPUG para a contagem de pontos de função em aplicações GUI (aplicações de interface gráfica) conforme exposto por [GAR 01].

Alguns elementos encontrados em aplicações GUI normais podem ser associados aos elementos existentes na interface gráfica de dispositivos portáteis, tais como:

- Janelas
- Menus pull-down e pop-up
- Listas de seleção tipo check box (seleção de múltiplos itens) ou radio button (seleção de um único item)
- Barras de rolagem (scroll bars)
- Botões

#### 4.3.1 Funcionalidades Incluídas na Contagem

Descreve-se nos tópicos seguintes as funcionalidades que devem ser contabilizadas

na obtenção da contagem de pontos de função para aplicações GUI.

- **Funcionalidades de Consulta Externa (EQ)** representadas pelas operações que extraem dados de um ou mais arquivos e mostram estes dados para o usuário. Um exemplo é a operação de abrir um arquivo (seleção da opção Abrir do menu).
- **Funcionalidades de Saída Externa (EO)** representadas pelas operações que além de extrair dados de um ou mais arquivos, processam estes dados ou atualizam arquivos internos antes de mostrar os dados derivados para o usuário. Podem ser usados como exemplos:
  - Operação de abrir um arquivo (seleção da opção Abrir de um menu) quando os dados do arquivo são transformados antes de serem mostrados ao usuário.
  - Mensagens de status, de informação ou de aviso, que não são mensagens de confirmação ou erro de uma determinada operação.
- **Funcionalidades de Entrada Externa (EI)** representadas pelas operações que atualizam um ou mais arquivos a partir de dados de outra aplicação ou entrados pelo usuário. Alguns cuidados devem ser tomados quanto a avaliação da complexidade dos EI's:
  - A escolha de uma única opção entre diversas alternativas através de lista de seleção do tipo radio button deve ser contado como um único DET. Não se deve considerar um DET para cada escolha.
  - Se múltiplas seleções podem ser aceitas, como no caso de check boxes, então cada seleção possível que não é repetição de outra deve ser contada como DET.
  - Mensagens de confirmação ou erro emitidas a partir da realização de uma operação, não são processos elementares por si só, portanto devem ser consideradas como um DET adicional para a operação a qual estão relacionadas.

### 4.3.2 Funcionalidades Excluídas da Contagem

Descreve-se nos tópicos seguintes as funcionalidades que **não** devem ser contabilizadas na obtenção da contagem de pontos de função para aplicações GUI.

- **Funcionalidades de Navegação** tais como:
  - Barra de rolagem.
  - Aparecimento de menus pull-down e pop-up.
  - Operação Sair acionado através de um botão ou de uma opção de

menu quando ela simplesmente retorna o foco de execução para o sistema operacional ou para outra aplicação.

- **Funcionalidades do Ambiente**, ou seja, funcionalidades automaticamente disponibilizadas pelo sistema operacional ou interface gráfica e que não são implementadas pela aplicação em questão.
  - Maximizar uma janela.
  - Minimizar uma janela.
  - Restaurar uma janela.
- **Funcionalidades Repetidas**, ou seja, a existência de múltiplas teclas ou botões ou opções de menu constituindo diferentes maneiras de realizar a mesma operação.

## 4.4 Paralelo na Contagem de Pontos de Função

Nesta seção será estabelecido um paralelo entre a contagem de pontos de função para aplicações em sistemas GUI (Graphical User Interfaces) como Windows, Macintosh e Motif, de acordo com o IFPUG (International Function Point User Group) [GAR 01], e a contagem de pontos de função para aplicações gráficas em dispositivos portáteis como telefones celulares, tomando-se como exemplo a interface gráfica dos telefones celulares Motorola V120 e V60 [MOT 03].

### 4.4.1 Funcionalidades Incluídas na Contagem

Não há diferenças em relação à contagem de arquivos, pois a contagem é feita da mesma maneira, mas em relação à contagem de transações, é necessário estabelecer um paralelo, o que será descrito nos tópicos seguintes.

- **Funcionalidade de Consulta Externa (EQ)**

Como já foi explicado anteriormente, devem ser contabilizadas como EQ seleções do usuário que levam ao resgate de informações armazenadas num arquivo lógico interno (ILF) ou arquivo de interface externa (EIF) e exibição para o usuário no display. Como exemplo tem-se a ação VIEW contida em uma das softkeys da figura 17. É importante enfatizar que estes dados estavam armazenados e não foram calculados (não houve processamento para obtê-los, portanto não caracteriza um EO). Pode-se comparar esta operação a uma operação Abrir em uma aplicação GUI.

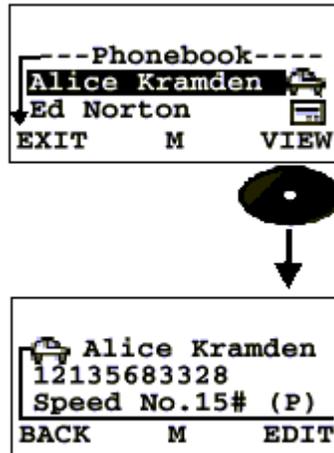


Figura 17 - Visualização Detalhada

- **Funcionalidade de Saída Externa (EO)**

As mensagens de notificação que eventualmente podem ser mostradas para o usuário contendo dados derivados de um processamento devem ser consideradas como EO's. Na figura 18 tem se um exemplo de notificação mostrada quando chegam novas mensagens de texto no telefone celular. Esta notificação é derivada do processamento de recebimento e contabilização das mensagens. Tal exemplo é análogo a exibição de uma janela com a mensagem de recebimento de novos e-mails por uma aplicação GUI de recebimento de e-mail.



Figura 18 - Mensagem de Notificação com Dados Derivados

- **Funcionalidade de Entrada Externa (EI)**

Como já foi explicado anteriormente, devem ser contados como EI processos elementares que criam, atualizam ou apagam dados de arquivos internos ou externos. Um cuidado especial deve ser tomado em relação à identificação do EI e dos DET's associados a este EI. Por exemplo, um formulário (figura 6) no qual aparecem vários campos editáveis pode ser considerado um único EI e cada campo do formulário como um DET associado a este EI mesmo que o campo seja editado em uma tela em separado através de um editor próprio (figura 7), pois se considera que a passagem da tela de formulário para a tela de editor como mera navegação como é mostrado na figura 19. A seção 5.4 contém exemplos

detalhados. Nas aplicações GUI normais não é necessário a edição dos campos de um formulário em uma tela em separado já que há espaço físico na tela para que a edição seja realizada no próprio contexto do formulário.

A softkey DONE que aparece quando um dos campos do formulário foi alterado é considerada DET adicional no EI formulário, pois esta ação representa sair do formulário atualizando o ILF (ou EIF) associado (o DONE é análogo a opção Save em uma aplicação GUI normal). É claro que se supõe aqui que quando o editor é finalizado com a softkey OK não há atualização de IFL ou EIF para que esta ação OK seja considerada somente uma ação de navegação.

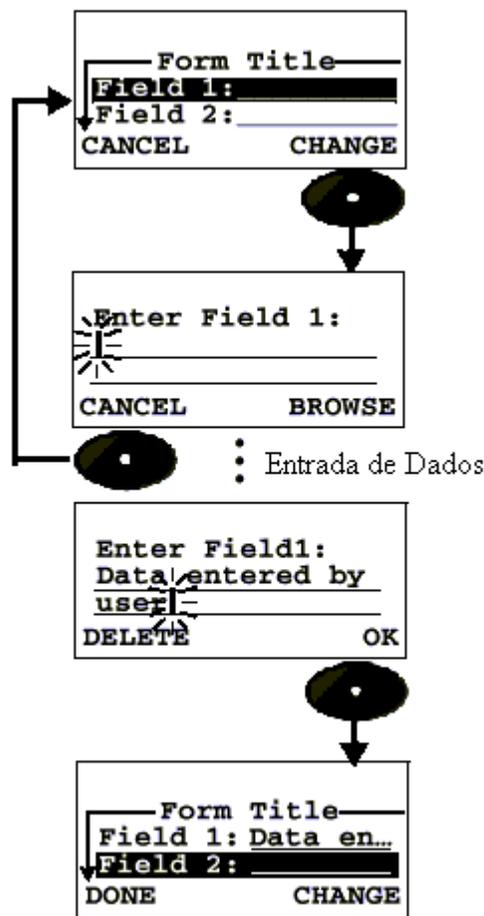


Figura 19 - Navegação entre Formulário e Editor

Na figura 20 exemplifica-se outra maneira de alterar um determinado campo de um formulário, neste caso utilizando uma lista de valores pré-definidos da qual seleciona-se um único valor (análogo a radio buttons em uma aplicação GUI). Cada valor pré-definido não deve ser considerado como um DET adicional, ou seja, tem-se um único DET associado ao campo em questão do formulário.

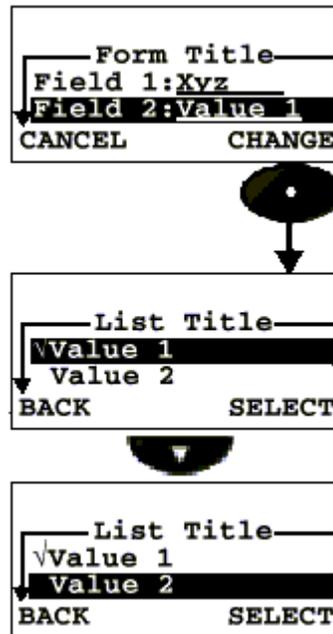


Figura 20 - Seleção de Valores Exclusivos

Diferentemente do exemplo da figura 20, tem-se o exemplo da figura 21 no qual múltiplos valores podem ser selecionados para o mesmo campo. Neste caso cada item passível de seleção deve ser contado como um DET adicional (análogo a check boxes em uma aplicação GUI).

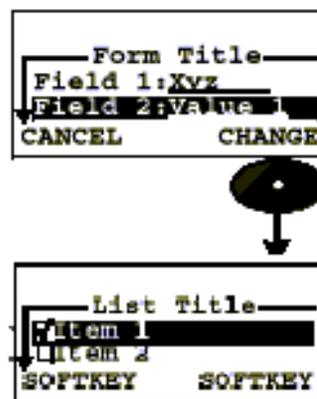


Figura 21 - Seleção de Múltiplos Valores

Durante a execução de um EI pode haver mensagens de questionamento do usuário, como é mostrado na figura 12, ou mensagens de erro devido a valores incorretos entrados pelo usuário ou ainda mensagens de confirmação sobre o resultado de um determinado processamento como apresentado na figura 14. Cada ocorrência não repetida destes três casos deve ser contada como um DET adicional no EI associado da mesma maneira como é

considerado em aplicações GUI normais.

Eventos externos à fronteira (aplicação) estabelecida para a contagem de pontos de função devem ser considerados como EI's, também da mesma maneira como é considerado em aplicações GUI normais.

#### 4.4.2 Funcionalidades Excluídas da Contagem

O paralelo em relação as funcionalidades que **não** devem ser contabilizadas na obtenção da contagem de pontos de função envolve as mesmas funcionalidades não contabilizadas para aplicações GUI.

- **Funcionalidades de Navegação:** na figura 19, a softkey CHANGE não representa um DET adicional para o EI formulário, pois ela é apenas uma ação de navegação entre telas (formulário e editor) e não a atualização de um ILF como já foi citado no item 4.4.1. Analogamente, as softkeys CANCEL (tanto no formulário como no editor) e softkey OK também representam ações de navegação, pois, CANCEL significa sair do formulário sem atualizar o ILF ou o retorno do editor para o formulário sem atualizar o conteúdo associado ao campo (DET) enquanto OK significa sair de editor atualizando o conteúdo do campo. A opção CANCEL de aplicações GUI, disponibilizada como botão ou como item de menu, tem o mesmo significado de cancelar a edição retornando a tela anterior e também não é contabilizado como um DET adicional já que é uma funcionalidade de navegação.
- **Funcionalidades do Ambiente:** funcionalidades automaticamente disponibilizadas pelo sistema ou interface gráfica e que não são implementadas pela aplicação em questão, como, por exemplo, as funcionalidades de ZOOM apresentada pela figura 22, análogas ao redimensionamento (maximizar, minimizar ou restaurar) de janelas da aplicação GUI, não devem ser consideradas na contagem de pontos de função para uma determinada aplicação.

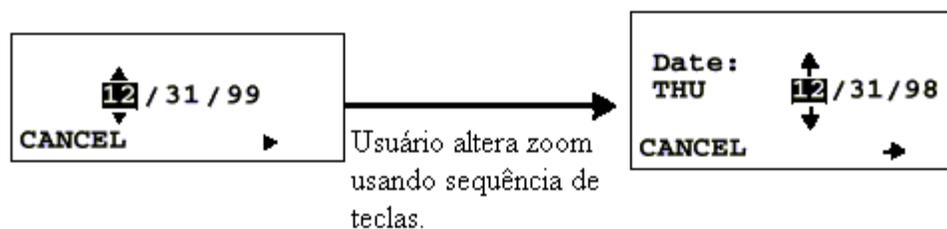


Figura 22 - Redimensionamento como Função Provida pelo Sistema

## Capítulo 5

### Exemplificação da Proposta de Contagem

Neste capítulo aplica-se na prática o paralelo de contagem de pontos de função estabelecido no capítulo 4 para a obtenção da contagem de uma aplicação já implementada utilizando-se a interface gráfica dos telefones celulares Motorola V120 e V60 [MOT 03].

Inicialmente a aplicação em questão é apresentada descrevendo-se as suas funcionalidades básicas. Em seguida inicia-se a contagem dos pontos de função baseada nas funcionalidades básicas e nas seqüências de telas (layouts) correspondentes. Aplica-se o paralelo de contagem na análise (avaliação de complexidade) das seqüências de telas. A contagem é finalizada com a totalização da quantidade de pontos de função não ajustados, cálculo do fator de ajuste e aplicação deste fator ajuste aos pontos de função não ajustados.

#### 5.1 Aplicação Agenda de Compromissos

A aplicação a ser usada como exemplo na contagem de pontos de função é a aplicação Agenda de Compromissos existente em telefones celular Motorola V120 e V60 [MOT 03]. Esta aplicação é um calendário que permite ao usuário agendar e organizar eventos tais como compromissos e encontros. Agendas podem ser criadas e visualizadas para dias específicos, e alarmes de lembrança de eventos podem ser criados se desejado. A Agenda de Compromissos também é capaz de mostrar os eventos existentes para uma semana toda. Devido ao espaço limitado da tela, a visualização dos eventos da semana objetiva apenas mostrar o quão cheio cada dia da semana está e facilitar a navegação de um dia para outro.

A estrutura da Agenda de Compromissos é hierárquica e o usuário pode navegar da visualização da semana para a visualização dos detalhes de um único dia. O usuário pode então selecionar um evento deste dia e navegar para a visualização de detalhes deste evento.

Os principais casos de uso que serão utilizados neste exemplo de contagem de pontos de função são mostrados na figura 23.

Os casos de uso Visualização da Semana, Visualização do Dia, Visualização do Compromisso e Edição do Compromisso representam quatro consultas externas (EQ's). Por outro lado, os casos de uso Armazenamento do Compromisso e Exclusão do Compromisso são duas entradas externas (EI's), e Lembrete do Compromisso é uma saída externa (EO's). Portanto uma estimativa inicial do número de pontos de função não ajustados considerando

apenas a classificação de transação (EQ, EI ou EO) dos casos de uso e sem identificar os arquivos referenciados seria (complexidade média é assumida para todos os casos de uso):

$$(2 \text{ EI's} \times 4) + (4 \text{ EQ's} \times 4) + (1 \text{ EO} \times 5) = 29 \text{ pontos de função não ajustados}$$

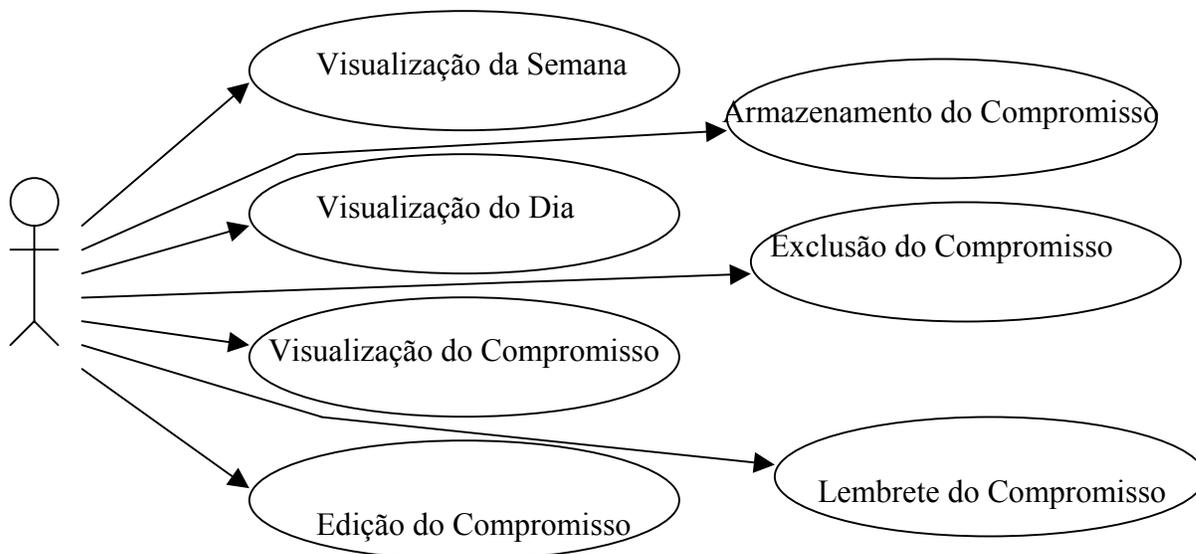


Figura 23 - Diagrama de Casos de Uso para a Agenda de Compromissos

Nos itens seguintes identificam-se os arquivos referenciados e se analisa cada caso de uso a fim de se obter a complexidade correta para cada caso de uso, chegando-se a contagem correta dos pontos de função.

## 5.2 Início da Contagem

Ao iniciar a contagem de pontos de função, os seguintes passos devem ser seguidos conforme foi explicado anteriormente:

- **Determinar o tipo de contagem:** trata-se de contagem para uma aplicação pré-existente.
- **Identificar o escopo e os limites da aplicação:** esta contagem se refere apenas a aplicação Agenda de Compromissos descrita acima.
- **Identificar todos os dados e transações com suas respectivas complexidades:** será feito nos itens seguintes.

## 5.3 Contagem dos Arquivos

Nos subitens seguintes serão apresentados os arquivos referenciados e suas

complexidades.

- **Arquivo de Calendário**

Trata-se de um arquivo de interface externa (EIF), pois só é utilizado para consulta e não é mantido por esta aplicação. Em relação ao número de RET's, tem-se um único.

Em relação aos DET's, considera-se também um único: data.

Finalmente, consultando a tabela de complexidade para EII

## 5.4 Contagem das Transações

Cada caso de uso é considerado uma transação a ser contabilizada.

- **Visualização da Semana**

O acesso a Agenda de Compromissos através do menu principal pode ser visualizado na figura 24.

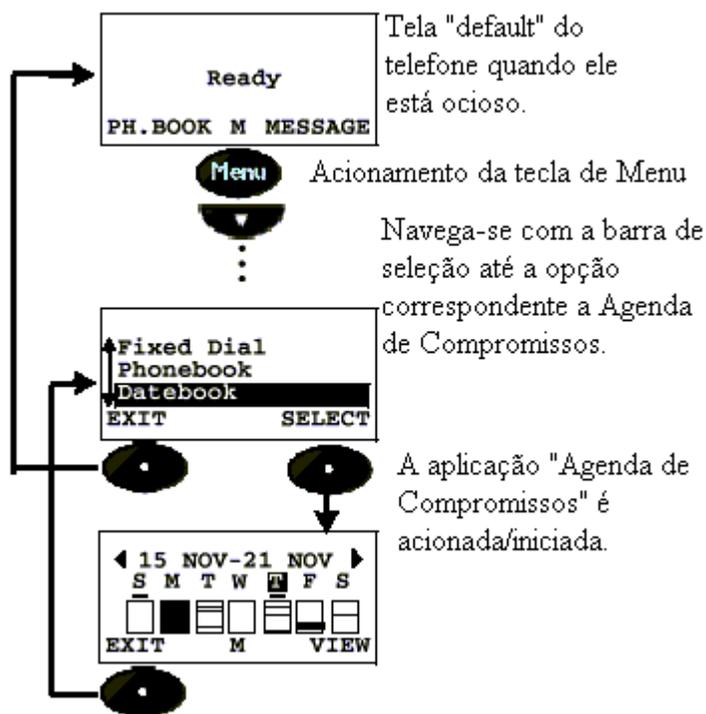


Figura 24 - Acesso a Agenda de Compromissos

Este acesso causa uma consulta externa (EQ), pois a partir da seleção da aplicação, todos os compromissos da semana corrente (domingo a sábado) são indicados (cada compromisso é um risco horizontal no retângulo correspondente ao dia). Dessa forma, a contagem deve ser avaliada considerando-se os arquivos referenciados (FTR) pela transação e pelos elementos de dados (DET) utilizados.

Quanto ao número de arquivos referenciados, esta transação acessa dois arquivos:

Arquivo de calendário.

Arquivos dos compromissos.

Em relação ao número de elementos de dados, a transação utiliza sete elementos listados a seguir:

Requisição de abertura da aplicação Agenda de Compromisso com a semana corrente (esta requisição é enviada a partir do pressionamento da softkey SELECT no diagrama da figura 24).

Dia da semana (o dia corrente da semana aparece selecionado, ou seja, fica em negrito, mas esta seleção pode ser alterada usando-se as teclas de navegação horizontal).

Dia do mês (para o domingo e sábado que delimitam a semana).

Mês (para o domingo e sábado que delimitam a semana).

Horário do compromisso (esta informação é exibida indiretamente através da posição da linha ou retângulo preenchido correspondente ao compromisso em questão).

Duração do compromisso (esta informação também é exibida indiretamente através da espessura da linha ou retângulo preenchido correspondente ao compromisso em questão).

Indicação de compromisso sem horário (como aniversários, por exemplo - pequeno traço horizontal sobre a letra do dia da semana).

Portanto, para a transação (caso de uso) Visualização da Semana foram contabilizados 2 FTR's e 7 DET's. Consultando-se a tabela de complexidade para EQ's (tabela 3), conclui-se que a complexidade é média e o número de pontos de função não ajustados para este EQ é 4.

A funcionalidade acionada através da softkey EXIT simplesmente sai da aplicação Agenda de Compromissos e retorna o usuário para o menu principal (figura 23), portanto consiste somente numa funcionalidade de navegação e não é considerada na contagem dos pontos de função.

#### • Visualização do Dia

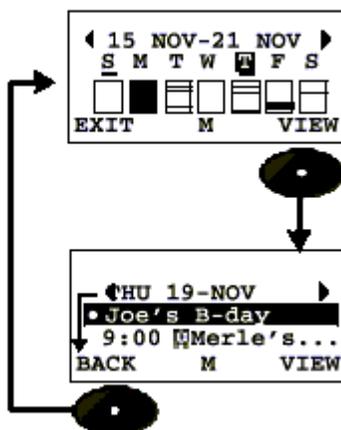


Figura 25 - Visualização do Dia

O acesso aos detalhes dos compromissos referentes a um determinado dia é mostrado na figura 25.

Esta transação deve ser contada como uma consulta externa (EQ). Dessa forma, a contagem deve ser avaliada considerando-se os arquivos referenciados (FTR) pela transação e pelos elementos de dados (DET) utilizados.

Quanto ao número de arquivos referenciados, a transação acessa:

Arquivo dos compromissos.

Em relação ao número de elementos de dados, a transação utiliza oito elementos listados a seguir:

Requisição da visualização dos dados referentes a um determinado dia. Esta requisição é enviada a partir do pressionamento da softkey VIEW (figura 24) para o dia selecionado (dia com a letra em negrito – figura 25).

Dia da semana (mostrado no topo da tela).

Dia do mês (mostrado no topo da tela).

Mês (mostrado no topo da tela).

Horário do compromisso.

Indicação de existência de alarme configurado para o compromisso.

Indicação de compromisso sem horário (como aniversários, por exemplo – ponto na frente do item do compromisso).

Início do título (descrição) do compromisso.

Portanto, para a transação (caso de uso) Visualização do Dia foram contabilizados 1 FTR's e 8 DET's. Consultando-se a tabela de complexidade para EQ's (tabela 3), conclui-se que a complexidade é baixa e o número de pontos de função não ajustados para este EQ é 3.

A funcionalidade acionada através da softkey BACK simplesmente retorna o usuário para a visualização da semana (figura 25), portanto consiste somente numa funcionalidade de navegação e não é considerada na contagem dos pontos de função.

#### • **Visualização do Compromisso**

O acesso aos detalhes de um determinado compromisso é mostrado na figura 26.

Esta transação deve ser contada como uma consulta externa (EQ). Dessa forma, a contagem deve ser avaliada considerando-se os arquivos referenciados (FTR) pela transação e pelos elementos de dados (DET) utilizados.

Quanto ao número de arquivos referenciados, a transação acessa:

Arquivo dos compromissos.

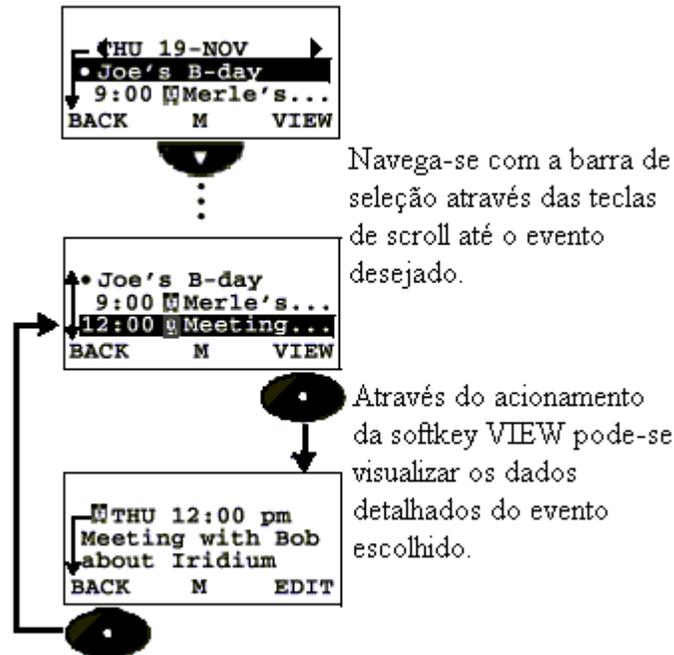


Figura 26 - Visualização do Compromisso

Em relação ao número de elementos de dados, a transação utiliza nove elementos listados a seguir:

Requisição da visualização dos dados referentes a um determinado compromisso. Esta requisição é enviada a partir do pressionamento da softkey VIEW (figura 26) para o compromisso selecionado (linha em negrito – figura 26).

Dia da semana (mostrado no topo da tela representando a data do compromisso).

Horário do compromisso.

Indicação de existência de alarme configurado para o compromisso.

Indicação de existência de nota de voz associada ao compromisso.

Título (descrição) do compromisso.

Duração do compromisso.

Período de antecedência do alarme.

Ciclo de repetição do compromisso.

Portanto, para a transação (caso de uso) Visualização do Compromisso foram contabilizados 1 FTR's e 9 DET's. Consultando-se a tabela de complexidade para EQ's (tabela 3), conclui-se que a complexidade é baixa e o número de pontos de função não ajustados para este EQ é 3.

A funcionalidade acionada através da softkey BACK simplesmente retorna o usuário para a visualização do dia (figura 26), portanto consiste somente numa funcionalidade de

navegação e não é considerada na contagem dos pontos de função.

### • Edição do Compromisso

Em vez de se utilizar os detalhes de um compromisso, pode se selecionar a edição do compromisso como é mostrado na figura 27.

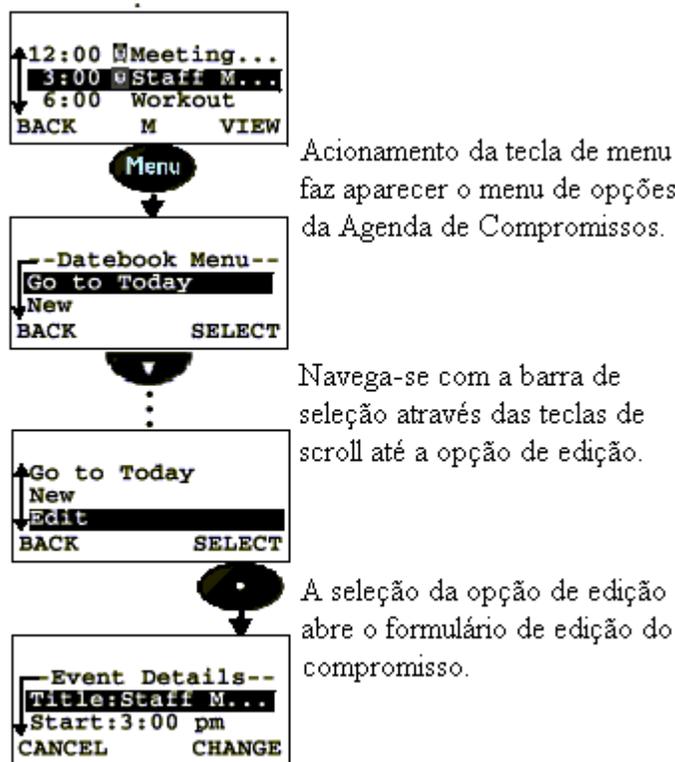


Figura 27 - Início da Edição do Compromisso

Esta transação deve ser contada como uma consulta externa (EQ). Dessa forma, a contagem deve ser avaliada considerando-se os arquivos referenciados (FTR) pela transação e pelos elementos de dados (DET) utilizados.

Quanto ao número de arquivos referenciados, a transação acessa:

Arquivo dos compromissos.

Em relação ao número de elementos de dados (DET), a transação utiliza oito elementos listados a seguir:

Requisição da edição dos dados referentes a um determinado compromisso. Esta requisição é enviada a partir da seleção da opção Edit do menu (figura 27).

Título (descrição) do compromisso.

Horário de início do compromisso.

Duração do compromisso.

Data do compromisso.

Ciclo de repetição do compromisso.

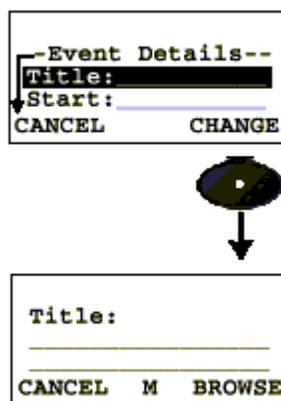
Período de antecedência do alarme (incluindo a desativação do alarme).

Nota de voz associada ao compromisso.

Portanto, para a transação (caso de uso) Edição do Compromisso foram contabilizados 1 FTR's e 8 DET's. Consultando-se a tabela de complexidade para EQ's (tabela 3), conclui-se que a complexidade é baixa e o número de pontos de função não ajustados para este EQ é 3.

A funcionalidade acionada através da softkey CANCEL simplesmente retorna o usuário para a visualização do dia (figura 27), portanto consiste somente numa funcionalidade de navegação e não é considerada na contagem dos pontos de função.

A alteração dos valores relativos ao compromisso pré-existente dá-se através da utilização de editores ou listas de seleção exclusiva apropriados, mas considera-se que as alterações provenientes das edições só alterarão um FTR após o conjunto de dados relativos ao compromisso for armazenado (esta operação de armazenamento será abordada em outro item), portanto as transições entre o formulário de edição do compromisso e os respectivos editores ou listas de seleção não representam transações adicionais e não é contabilizado nenhum ponto de função adicional. A figura 28 mostra um exemplo de transição entre o formulário de edição e o editor do título do compromisso (acionamento da softkey CHANGE).



**Figura 28 - Transição entre Formulário e Editor para Edição do Título**

- **Armazenamento de um Compromisso**

Após os dados relativos a um compromisso pré-existente serem editados, ou após os

dados relativos a um novo compromisso terem sido entrados, o compromisso em questão pode ser armazenado conforme é mostrado na figura 29.

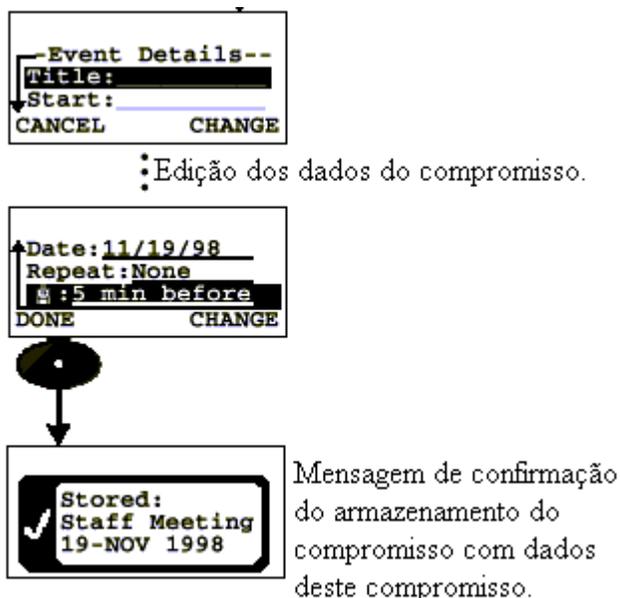


Figura 29 - Armazenamento de um Compromisso

Esta transação deve ser contada como uma entrada externa (EI). Dessa forma, a contagem deve ser avaliada considerando-se os arquivos referenciados (FTR) pela transação e pelos elementos de dados (DET) utilizados.

Quanto ao número de arquivos referenciados, a transação acessa:

Arquivo dos compromissos.

Em relação ao número de elementos de dados (DET), a transação utiliza nove elementos listados a seguir:

Requisição de armazenamento dos dados referentes a um determinado compromisso. Esta requisição é enviada a partir do acionamento da softkey DONE (figura 29).

Título (descrição) do compromisso.

Horário de início do compromisso.

Duração do compromisso.

Data do compromisso.

Ciclo de repetição do compromisso.

Período de antecedência do alarme (incluindo a desativação do alarme).

Nota de voz associada ao compromisso.

Mensagem de confirmação do armazenamento como mostrado na figura 29.

Portanto, para a transação (caso de uso) Armazenamento do Compromisso foram

contabilizados 1 FTR's e 9 DET's. Consultando-se a tabela de complexidade para EI's (tabela 1), conclui-se que a complexidade é baixa e o número de pontos de função não ajustados para este EI é 3.

- **Exclusão de um Compromisso**

Um compromisso que está selecionado pode ser apagado como mostra a figura 30.

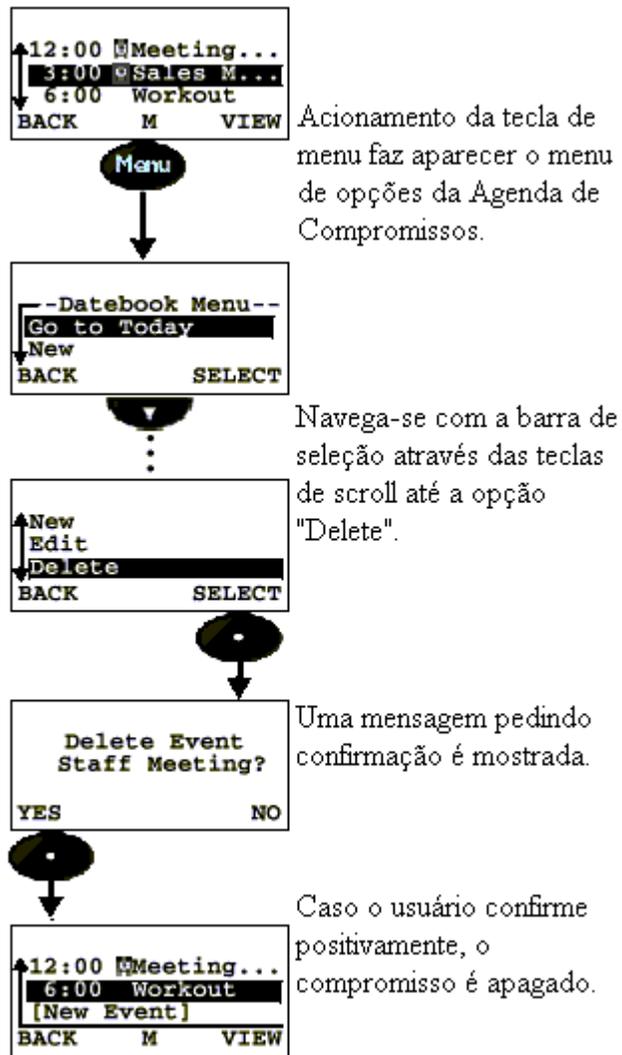


Figura 30 - Exclusão de um Compromisso

Esta transação deve ser contada como uma entrada externa (EI). Dessa forma, a contagem deve ser avaliada considerando-se os arquivos referenciados (FTR) pela transação e pelos elementos de dados (DET) utilizados.

Quanto ao número de arquivos referenciados, a transação acessa:

Arquivo de armazenamento dos compromissos.

Em relação ao número de elementos de dados (DET), a transação utiliza dois elementos listados a seguir:

Requisição de exclusão do compromisso. Esta requisição é enviada a partir da seleção da opção Delete conforme é mostrado na figura 30.

Mensagem solicitando a confirmação da exclusão como mostrado na figura 30.

Portanto, para a transação (caso de uso) Exclusão do Compromisso foram contabilizados 1 FTR's e 2 DET's. Consultando-se a tabela de complexidade para EI's (tabela 1), conclui-se que a complexidade é baixa e o número de pontos de função não ajustados para este EI é 3.

#### • Lembrete de um Compromisso

Quando há a expiração do alarme relativo a um compromisso, uma mensagem de notificação (lembrete) do compromisso é mostrada como mostra a figura 31.

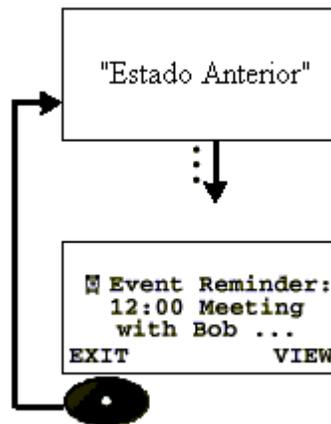


Figura 31 - Lembrete de um Compromisso

Esta transação deve ser contada como uma saída externa (EO). Dessa forma, a contagem deve ser avaliada considerando-se os arquivos referenciados (FTR) pela transação e pelos elementos de dados (DET) utilizados.

Quanto ao número de arquivos referenciados, a transação acessa três arquivos:

Arquivo de calendário.

Arquivo de horário.

Arquivo dos compromissos.

Em relação ao número de elementos de dados (DET), a transação utiliza dois elementos listados a seguir:

Horário de ocorrência do compromisso.

Título do compromisso.

Portanto, para a transação (caso de uso) Lembrete do Compromisso foram contabilizados 3 FTR's e 2 DET's. Consultando-se a tabela de complexidade para EO's (tabela 2), conclui-se que a complexidade é baixa e o número de pontos de função não ajustados para este EI é 4.

A funcionalidade acionada através da softkey EXIT simplesmente retorna o usuário para a tela anterior (figura 31), portanto consiste somente numa funcionalidade de navegação e não é considerada na contagem dos pontos de função. Já a softkey VIEW permite a visualização dos dados do evento recaindo no caso de uso Visualização do Compromisso já abordado.

#### • Menu da Agenda de Compromissos

O simples acionamento da tecla de menu e o aparecimento das opções pertinentes a cada situação (visualização da semana, dia ou evento) é considerada uma mera função de navegação e por isso não deve ser considerada na contagem de pontos de função conforme orientação contida em [GAR 01].

## 5.5 Totalização dos Pontos de Função Não Ajustados

O total de pontos de função não ajustados será obtido somando-se todos os pontos de função provenientes tanto dos arquivos como das transações identificadas, conforme tabela 6.

	COMPLEXIDADE			TOTAL
	Baixa	Média	Alta	
Entrada Externa	2 x 3	0 x 4	0 x 6	6
Saída Externa	1 x 4	0 x 5	0 x 7	4
Consulta Externa	3 x 3	1 x 4	0 x 6	13
Arquivo Lógico Interno	1 x 7	0 x 10	0 x 15	7
Arquivo de Interface Externa	2 x 5	0 x 7	0 x 10	10
Total de Pontos de Função Não Ajustados				<b>40</b>

Tabela 6 - Totalização dos Pontos de Função Não Ajustados

## 5.6 Determinação do Fator de Ajuste de Valor

O cálculo do fator de ajuste de valor (VAF), como já foi explicado na seção 3.1.2 depende das características gerais de sistema. A tabela 7 apresenta o grau de influência de

cada característica geral de sistema tomando como guia de referência a avaliação das características gerais de sistema da aplicação Windows Calendar contida em [GAR 01].

Característica	Grau de Influência
1. Comunicação de Dados	4
2. Processamento Distribuído de Dados	0
3. Desempenho	4
4. Utilização Intensiva de Configuração	4
5. Taxa de Transação	3
6. Entrada de Dados On-line	5
7. Eficiência para o Usuário Final	5
8. Atualização On-line	3
9. Processamento Complexo	1
10. Reusabilidade	2
11. Facilidade de Instalação	0
12. Facilidade de Operação	3
13. Múltiplas Localidades	2
14. Facilidade de Mudança	2
Grau Total de Influência	<b>38</b>

**Tabela 7 - Características Gerais de Sistema**

É interessante lembrar que se tratando de uma aplicação embarcada, características ligadas ao desempenho e tamanho do software são significativas. Também é bastante significativa a eficiência para o usuário final devido às limitações do dispositivo em questão (telefone celular). Algumas características como facilidade de instalação não se aplica ao tipo de aplicação em questão uma vez que o usuário não é responsável pela instalação.

Considerando-se a fórmula para o cálculo do valor de ajuste apresentada na seção 3.1.2 tem-se:

$$VAF = 0,65 + 38/100 = \mathbf{1,03}$$

O valor de ajuste igual a 1,03 indica que haverá um acréscimo de 3% no valor encontrado para a contagem de pontos de função não ajustados.

## **5.7 Valor Final dos Pontos de Função Ajustados**

Baseado na contagem total de pontos de função não ajustados e no valor obtido para o VAF, o total de pontos de função ajustados será dado por  $40 \times 1,03 = \mathbf{41,2}$  (acrécimo de 3%). Por outro lado, este valor é aproximadamente 42% maior que o valor obtido somente com a análise dos casos de uso apresentada na seção 5.1. Tal diferença mostra que a

avaliação da complexidade de cada caso de uso (EI, EO e EQ) e a identificação dos arquivos (ILF e EIF) podem ser significativas na obtenção da contagem de pontos de função.

# Capítulo 6

## Considerações Finais

As métricas de software são fundamentais para a realização do planejamento do projeto de desenvolvimento do produto de software. A primeira atividade do planejamento é a realização de estimativa para obtenção do valor para a métrica de custo/esforço (homem-mês). Existem diversas maneiras (técnicas ou algoritmos) de realizar tal estimativa da métrica de esforço, mas várias destas maneiras utilizam inicialmente uma estimativa do valor de uma métrica de tamanho (LOC ou pontos de função) para então através de uma produtividade histórica obter o valor da estimado da métrica de esforço (homem-mês). Com o valor estimado para a métrica de esforço, viabiliza-se a análise de risco (segunda atividade do planejamento) e, junto com a definição de atividades (terceira atividade do planejamento), permite a confecção de um cronograma (quarta atividade do planejamento) e orçamento (última atividade do planejamento) para o projeto de desenvolvimento do produto de software.

Independentemente da técnica ou algoritmo de estimativa da métrica de esforço/custo utilizada, a precisão do valor estimado está intimamente relacionada à quantidade de informação sobre o software a ser desenvolvido ou em desenvolvimento no momento da realização da estimativa. Além da quantidade de informação, diversos outros fatores como processo, tecnologia, ferramentas e reutilização de código influenciam na precisão do resultado da estimativa.

Uma técnica de estimativa colaborativa bastante utilizada quando não há dados históricos sobre as métricas de software é o Wide Band Delphi. Já entre os modelos algorítmicos, destaca-se o COCOMO.

Os pontos de função são uma métrica de tamanho funcional que é avaliada a partir da quantidade e da complexidade de transações realizadas (EI, EO ou EQ) e arquivos referenciados (ILF ou EIF) pelo sistema. É possível fazer uma avaliação inicial da quantidade de pontos de função a partir da análise dos casos de uso deste sistema. Outras métricas de tamanho relevantes são o LOC (expressa a quantidade de linhas do código fonte) e quantidade de memória (quantidade de ROM e RAM utilizada em bytes). Entretanto o valor da métrica de pontos de função pode ser obtido com mais precisão que o das outras métricas de tamanho na fase inicial do projeto de desenvolvimento do produto de software.

## 6.1 Trabalho Realizado

A fim de aplicar a métrica de pontos de função a aplicações gráficas de dispositivos portáteis foi feita uma analogia entre este tipo de interface gráfica e interfaces gráficas (GUI) utilizadas nos computadores em geral.

Entre os conceitos básicos para aplicações gráficas em dispositivos portáteis (telefones celulares), tem-se o menu sensível a contexto (lista de opções que se aplicam a uma situação específica) e as softkeys (teclas de rótulos variáveis de acordo com o contexto). Vale destacar também o teclado limitado como dispositivo de entrada de dados.

Pode-se identificar alguns elementos básicos da interface gráfica de dispositivos portáteis (telefone celular) como formulário, editor, lista marcada (análogo a radio button), lista de seleção múltipla (análogo a check box), medidor ajustável, picker, notice, lista, visualizador e medidor.

Estabelecendo-se analogias de alguns destes elementos com os elementos encontrados numa interface gráfica normal (janelas, menus, check box, radio button, barra de rolagem, botões), e aplicando-se os mesmos princípios básicos de contagem de ponto de função para aplicações GUI, foi possível estabelecer um paralelo na contagem de pontos de função para aplicações gráficas de dispositivos portátil (telefone celular). Finalmente, este paralelo foi utilizado na contagem de pontos de função da aplicação Agenda de Compromissos de telefones celulares Motorola.

## 6.2 Contribuição

Este trabalho mostrou ser realmente possível a obtenção do valor da métrica de pontos de função analisando-se especificações funcionais e de layout de aplicações gráficas para telefones celulares através de um paralelo com a contagem de pontos função para aplicações GUI em geral.

Apesar das possíveis vantagens da utilização da métrica de pontos de função para obtenção de estimativas no início do ciclo do desenvolvimento, foi também constatada neste trabalho a dificuldade em se obter a contagem de pontos de função devido ao fato dessa métrica não ser trivial: a sua correta obtenção exige treinamento e esforço.

Alguma simplificação pode ser obtida na obtenção da contagem de pontos de função se for desprezada a avaliação das características gerais de sistema, pois o fator de ajuste de valor obtido para o exemplo do capítulo 5 alterou em apenas 3% a contagem de pontos de função não ajustados, e as características gerais de sistema tendem a não mudar para aplicações desenvolvidas para o mesmo tipo de sistema (neste caso o telefone celular). Algumas publicações como [LON 02a], já comentam sobre uma tendência generalizada de

não avaliação das características gerais de sistema.

### **6.3 Extensões**

Uma das extensões para este trabalho deve utilizar a proposta de contagem de pontos de função para aplicações de dispositivos portáteis (telefones celulares) em outras aplicações além da Agenda de Compromisso. Baseando-se nos resultados destas contagens e em dados históricos da métrica de esforço (homem-mês), avaliar se a produtividade dada pela quantidade de esforço por ponto de função é aproximadamente a mesma para as diferentes aplicações considerando-se todos os fatores que influenciam na produtividade do desenvolvimento do produto de software como é enumerado por [JON86]. Tal análise serviria para verificar se aplicação da métrica de ponto de função como foi proposta aqui é reprodutível ou não.

A obtenção de uma produtividade consistente por ponto de função poderá ser utilizada na realização de estimativas de esforço a partir das especificações funcionais e de layout fornecidas para o projeto e, aplicando a produtividade por ponto de função, seriam obtidas estimativas de esforço confiáveis no início do ciclo de desenvolvimento do produto de software.

Pode-se tentar obter outras métricas, tais como as métricas de tamanho (LOC, quantidade de memória) a partir da produtividade por ponto de função de forma a estimar os valores de tais métricas também no início do ciclo de desenvolvimento, contribuindo para a definição do hardware final e permitindo que o custo deste hardware seja avaliado com antecedência.

Finalmente, as interfaces gráficas de software utilizadas nos telefones celulares e em outros dispositivos portáteis como PDA's (personal digital assistance) estão se tornando cada vez mais sofisticadas e parecidas com as interfaces gráficas de computadores pessoais normais devido à evolução e melhoria da capacidade de processamento e memória destes dispositivos. Tal fato tende a facilitar o estabelecimento de um paralelo para a aplicação da métrica de ponto de função nos dispositivos portáteis como foi feito neste trabalho.

## Referências Bibliográficas

- [ALB 83] ALBRECHT, A. J. Software Function, Lines of Code and Development Effort Prediction: a Software Science Validation. **IEEE Transactions on Software Engineering**, 1983. SE-9(6), 639-47(Ch 23)
- [BOE 81] BOEHM, Barry. **Software Engineering Economics**. New Jersey: Prentice-Hall, 1981.
- [BOE 91] BOEHM, Barry W. Software risk management: principles and practices. **IEEE Transactions on Software Engineering**, Vol. 18, No. 1, Jan. 1991, pp. 32-41.
- [BRO 75] BROOKS, F. **The Mythical Man-Month**. Addison-Wesley, 1975.
- [CAN 99] CASAROTTO FILHO, Nelson; FAVERO, Jose S.; CASTRO, João E. E. **Gerência de Projetos / Engenharia Simultânea**. São Paulo: Atlas, 1999.
- [CAR 01] CARVALHO, Ariadne M. B. R.; CHIOSSI, Thelma C. S. **Introdução à Engenharia de Software**. Campinas, SP: Editora da Unicamp, 2001.
- [FER 95] FERNANDES, Aguinaldo A. **Gerência de Software Através de Métricas**. São Paulo: Atlas, 1995.
- [FUR 97] FUREY, S. Why We Should Use Function Points. **IEEE Computer**, March/April 1997, pp. 28-30.
- [GAR 01] GARMUS, David; HERRON, D. **Function Point Analysis: Measurement Practices for Successful Software Projects**. Addison-Wesley, 2001
- [HAU 01] HAUFE, Maria Isabel **Estimativa da Produtividade no Desenvolvimento de Software**. Porto Alegre, RS: PPGC da UFRGS, 2001.
- [JON 86] JONES, Capers **Programming Productivity**. McGraw-Hill, 1986.
- [LON 02a] LONGSTREET, David **Function Point Analysis Training Course**. [www.SoftwareMetrics.Com](http://www.SoftwareMetrics.Com), 2002.
- [LON 02b] LONGSTREET, David **Pontos de Função Aplicados a Tecnologias Novas e Emergentes**. [www.SoftwareMetrics.Com](http://www.SoftwareMetrics.Com), 2002.
- [LON 02c] LONGSTREET, David **Use Cases and Function Points**. [www.SoftwareMetrics.Com](http://www.SoftwareMetrics.Com), 2002.
- [LOW 90] LOW, Graham C.; JEFFERY, D. Ross Function Points in the Estimation and Evaluation of the Software Process. **IEEE Transactions on Software Engineering**. VOL. 16 NO. 1, January 1990.
- [MAX 00] MAXWELL, Katrina D. FORSELIUN, Pekka. Benchmarking Software Development Productivity. **IEEE Software**, New York, Jan./Feb. 2000.
- [MOT 03] MOTOROLA e V. Séries são marcas registradas da Motorola, Inc. ®Reg. U.S. Pat. & Tm. Off.
- [PAR 57] PARKINSON, G. N. **Parkinson's Law and Other Studies in Administration**, Houghton-Mifflin, Boston, 1957.
- [PMI 96] **A Guide to the Project Management Body of Knowledge**. ISDN 1-880410-12-5, Project Management Institute Standards Committee, 1996.
- [PRE 92] PRESSMAN, Roger S. **Software Engineering** 3.ed. New York: McGraw-Hill, 1992.

- [PUT 78] PUTNAM, Lawrence H. A General Empirical Solution to the Macro Software Sizing and Estimating Problem. **IEEE Transactions on Software Engineering**, [S.1.] v.4, n.4, June 1978.
- [SAC 68] SACKMAN, H.; ERIKSON, W. J.; GRANT, E. E. Exploratory Experimental Studies Comparing Online and Offline Programming Performance. **CACM**, 11, 1 (1968), pp. 3-11
- [STU 96] STUTZKE, Richard D. **Software Estimation Technology: A Survey**. Science Applications International Corporation, Huntsville, AL, 1996.
- [WEB 99] WEBER, Kival C.; ROCHA, Ana R. C. **Qualidade e Produtividade em Software**, 3.ed. São Paulo: Makron Books, 1999.
- [WIE 99] WIEGERS, K. First Things First: Prioritizing Requirements. **Software Development Magazine**, September 1999.