



UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

**IMPLEMENTANDO UM MÉTODO DE ESTIMATIVA
DE PROJETOS DE SOFTWARE NO DOTPROJECT**

TRABALHO DE GRADUAÇÃO

Aluno: Eduardo Vinicius de Figueiredo Salvador (evfs@cin.ufpe.br)
Orientador: Hermano Perrelli de Moura (hermano@cin.ufpe.br)

agosto de 2005

Assinaturas

Este Trabalho de Graduação é resultado dos esforços do aluno Eduardo Vinicius de Figueiredo Salvador, sob a orientação do professor Hermano Perrelli de Moura, na participação projeto “Implementando um Método de Estimativa de Projetos de Software no DotProject”, conduzido pelo Centro de Informática da Universidade Federal de Pernambuco. Todos abaixo estão de acordo com o conteúdo deste documento e os resultados deste Trabalho de Graduação.

Eduardo Vinicius de Figueiredo Salvador

Hermano Perrelli de Moura

Agradecimentos

Primeiramente a Deus, depois meu pai, mãe, irmãos, primos, tios, avós, agregados, ou seja, a toda minha família que sempre fez parte da minha vida e que me acompanharam nesse percurso de 5 anos durante a graduação.

Ao professor Hermano Perrelli, meu orientador nesse trabalho, que me guiou durante essa pesquisa da forma mais atenciosa possível.

Agradeço a Ricardo, gerente do projeto Sig@, o qual foi o idealizador desse projeto. E junto com Ricardo agradeço a todos aqueles que formam a equipe Sig@, onde eu posso citar Novaes, .j, Luiz, Rita, Pimbolim, Carol, Renato, ao velho cave, Alessandro e Diego, Fábio, Alimafos, Adonay, Juuuuuuuuu e todos aqueles que passaram por essa equipe e hoje estão em outros projetos mundo a fora.

Bem, também não posso esquecer de agradecer a turma maravilhosa que eu fiz parte que foi a turma grad2000-2 + grad2001-1. Em especial lembro daqueles que passei mais tempo junto: asas, Andrino, Daniel Thiago, Mari, dedeco, magão, Brunão, Bourbon, Célio, Cristina, grande Gandhi, Mazza, Jarbinha, Jenny, presidente, marquinho, zumbi, ao grande e saudoso Mauro, Rodrigão, Rodrigo Mendes, Vituuuuuuu, Juuuuuuuuu de novo, Adonay de novo, Carol de novo, Pimbolim de novo, Jayro, Hermano Atrasado Brandão, Capiou, Diogão, Chico Bento, Dudu, Heitor, Marcílio, Pikachu, Chicão, Igor, Mateus.

Bem, e deixo por último o mais especial dos agradecimentos, para aquela pessoa que durante quase todo o curso me acompanhou de perto, me deu forças nas horas mais complicadas, me motivou, e que hoje faz parte da minha família, foi minha orientadora nesse trabalho a assuntos relacionados ao word, e que fez parte da minha turma, ela é minha amiga e minha namorada, o meu todo especial agradecimento a ela.....Bárbara.

Resumo

Gerência de projeto é aplicação de conhecimento, habilidades e técnicas do projeto, a fim de satisfazer as necessidades dos stakeholders (interessados e envolvidos). O gerenciamento de projetos deve ser auxiliado pela utilização de uma métrica, que permita a mensuração de um projeto e conseqüentemente a geração de sua estimativa de prazo, custo e recursos. Este trabalho tem como objetivo a implementação, no dotProject, de um método de estimativa de projeto utilizando Análise de Pontos de Função, tendo como foco a estimativa de prazo para realização de tarefas, e aplicar a ferramenta em um estudo de caso.

Abstract

Project Management means knowledge, abilities and projects techniques application, in order to satisfy the stakeholders needs (the involved and interested parties). The management of projects must be supported by the use of a metric, wich allows the measure of a project and consequent generation of this time, cost and resources estimatives. This work has the objective of the implementation, into the DotProject software, of a project estimative method using Function Point Analysis, and having as a focus, the estimative of the time to accomplish the project tasks, and apply this tool in a case study.

Sumário

1. Introdução	11
1.1. Contexto	12
1.2. Objetivo	14
1.3. Estrutura do Trabalho	14
2. Importância da Mensuração do Software	16
2.1. Gerência de Projetos	16
2.1.1. Planejamento	16
2.1.2. Controle	19
2.1.3. Adiamento do Projeto	19
2.1.4. Foco	20
2.2. Iniciativas de SPI (Software Process Improvement)	21
2.2.1. SEI CMMI – Capability Maturity Model Integration	22
2.2.2. SPICE – Software Process Improvement and Capability Determination (ISO/IEC 15504)	25
2.3. Ferramentas Livres de Auxílio ao Gerenciamento de Projetos	27
2.3.1. USC-COCOMO II	27
2.3.2. Rambutan	28
2.3.3. DotProject	29
3. Métricas Funcionais	30
3.1. Histórico	30
3.2. IFPUG	31
3.3. BFPUG	32
3.4. NESMA	32
3.5. MARK II	33
3.6. COSMIC – FFP	33
3.7. Padrão ISO de Medição Funcional	34
3.8. ISBSG	35
4. Análise de Pontos de Função	36
4.1. Os requisitos e a Contagem de Pontos de Função	36
4.2. Objetivos e Benefícios da Análise de Pontos de Função	37
4.3. Propósito da Contagem de pontos de Função	38
4.4. Processo de Contagem de Pontos de Função	38
4.4.1. Identificando Escopo da Contagem e Fronteira da Aplicação	39
4.4.2. Contagem de Funções do Tipo Dados	39
4.4.3. Contagem de Funções do Tipo Transação	42
4.4.4. Determinando a contagem de Pontos de Função Não-Ajustados	44
5. Implementação	45
5.1. Ferramenta e Tecnologia Escolhida	45
5.2. Modelagem do Negócio	45
5.2.1. Produtividade x Tecnologia	46
5.2.2. Produtividade x Pessoa	47
5.2.3. Calcular Pontos de Função	48
5.2.4. Estimativa de Prazo	48
5.3. Modelagem de Negócios Futuros	49
5.3.1. Inferência	49
5.3.2. Estimativa de Custo Financeiro	49
6. Estudo de Caso	50

6.1.	Descrição do Sistema SIG@	50
6.2.	Lista de Atores.....	51
6.3.	Especificação do Caso de Uso.....	51
6.4.	Identificação e Classificação das funcionalidades.....	52
6.5.	Cálculo de Pontos de Função	55
6.5.1.	Cálculo de Pontos de Função Não-Ajustados	55
6.6.	Aplicação na Estimativa de Prazo	56
7.	Conclusão e Trabalhos Futuros	57
7.1.	Conclusão	57
7.2.	Trabalhos Futuros	58
	Referências	60
	Apêndice A	62

Índice de Figuras

Figura 1: Pontos de contagem e estimativa ao longo do ciclo de vida do projeto.....	17
Figura 2: Variação do tamanho do projeto ao longo do seu ciclo de vida	18
Figura 3: Dinâmica dos 99% concluídos [9]	20
Figura 4: Melhoria contínua [9].....	23
Figura 5: Avaliação do processo de software [26]	25
Figura 6: Evolução dos métodos [25].....	35
Figura 7: processo de contagem de pontos de função	38
Figura 8:Funções do tipo dados.....	39
Figura 9: Funções do tipo transação	42
Figura 10: Contagem dos pontos de função	44
Figura 11: Tela principal da ferramenta	46
Figura 12: Questionário Perfil Social	62

Índice de Tabelas

Tabela 1: Complexidade funcional dos ALI e AIE.....	41
Tabela 2: Contribuição dos pontos de função não-ajustados do tipo dado	41
Tabela 3: Complexidade funcional para EE.....	43
Tabela 4: Complexidade funcional para SE e CE	43
Tabela 5: Contribuição dos pontos de função não-ajustados das função tipo transação	44
Tabela 6: Produtividade média em relação a tecnologia	47
Tabela 7: Formulário de contagem do tipo dado.....	53
Tabela 8: Formulário de contagem do tipo transação.....	55
Tabela 9: Contribuição das funções para o questionário de perfil social	56

“Não se pode gerenciar o que não se pode medir”.

Tom de Marco

1. Introdução

Existem várias razões para se medir o software. A frase “você não pode gerenciar o que não pode medir” tem-se tornado um forte slogan para programas de métricas. Existem tanto razões táticas, quanto estratégicas para medir. Métricas ajudam aos gerentes a controlarem o progresso dos projetos. Métricas têm sido utilizadas pelas organizações para transformar-se em vantagem estratégica através da melhoria do processo de software.

O principal objetivo para implementar um processo de coleta de métricas é fornecer aos gestores um conjunto de informações úteis e tangíveis para dimensionar, estimar, planejar e controlar os projetos com maior rigor e precisão. Dentre as principais razões para medir software temos:

- Formar uma base para estimativas;
- Determinar se as metas de produtividade do projeto estão sendo atingidas;
- Determinar se as metas de qualidade do processo e do produto estão sendo atendidas;
- Avaliar os benefícios de novos métodos, treinamentos e ferramentas de software;
- Melhorar o relacionamento com o cliente;
- Melhorar a gerência de contratos de software e relacionamento com terceiros;
- Reduzir os riscos de pressão excessiva nos cronogramas;
- Melhorar a gerência de projetos de desenvolvimento de software;
- Entender e aperfeiçoar o processo de software;

Este Trabalho de Graduação vai ter como objetivo final a implementação de um método de medidas de software em uma ferramenta de auxílio na gerência de projeto, e o foco será dado na estimativa de cronogramas para atividades de projetos.

1.1. Contexto

De acordo com o PMI (*Project Management Institute*) [1], um projeto é um empreendimento temporário posto em execução para criar um produto ou serviço único.

Ainda segundo o PMI, gerência de projeto é aplicação de conhecimento, habilidades, ferramentas e técnicas em atividades do projeto, a fim de satisfazer às necessidades dos *stakeholders* (interessados e envolvidos).

Na condução de um projeto existem vários processos para gerenciá-los. Esses processos são agrupados em passos básicos, dos quais se destacam o planejamento, a execução e o controle. No Planejamento não existe um completo conhecimento das características do produto que permita a apuração de seu dimensionamento, o que faz necessário, a realização de uma estimativa.

O gerenciamento de projetos deve ser auxiliado pela utilização de uma métrica, que permita à mensuração de um projeto e conseqüentemente a geração de sua estimativa de prazo, custo e recursos. Existem vários modelos de estimativas criados, com o intuito de fornecer métricas que permitam fazer uma estimativa do tamanho do projeto.

No ramo de desenvolvimento e manutenção de sistemas, é visível o crescente número de empresas interessadas na obtenção de certificados de qualidade ISO/CMMI [2] [3]. E nesse ramo de desenvolvimento e manutenção de sistemas, uma das métricas de mais destaque é a Análise de Pontos de Função (APF) [7, 9,10]. Em termos práticos, a aplicação da APF pode integrar-se de forma bastante útil às atividades compreendidas em várias áreas de processos do CMMI Nível 2.

A análise de pontos de função é uma técnica padrão de mercado, empregada para medir de uma forma consistente a funcionalidade fornecida pelo software (produto e processo) do ponto de vista do usuário, independente da tecnologia utilizada na sua implementação.

O propósito da contagem de pontos de função é fornecer uma resposta a um problema de negócio de uma instituição. Podemos citar como exemplo a contagem com

o propósito de fornecer elementos para uma estimativa mais eficiente de prazo de atividades de um projeto de software.

O CMMI é um modelo de qualidade que descreve o que deve ser feito para melhoria gradual de processos de uma organização. Para possibilitar a implantação das recomendações deste modelo, são necessárias ferramentas de software. Atualmente, existe um movimento crescente onde esforços individuais e coletivos, derivam à criação de softwares não proprietários com o objetivo de atender às necessidades comuns das organizações e de indivíduos.

Software livre (*free software*) é o software disponível com a permissão para qualquer um usá-lo, copiá-lo e distribuí-lo, seja na forma original ou com modificações, seja gratuito ou com custo. A possibilidade de modificações implica em que o código fonte esteja disponível (*open source*). Depois de uma pesquisa pela internet, verificou-se a inexistência de uma ferramenta de gerenciamento de projeto voltado para ambientes corporativos, e que seguem a ideologia de software livre, e ainda possuam a técnica de análise de pontos de função pra se fazer estimativas de projeto.

O que existe hoje são sistemas proprietários, cujos valores de licenças são altos, ou ferramentas exclusivas para se fazer uma análise de pontos de função, ou seja, não existe uma aplicação para gerenciamento de projeto voltado para ambientes corporativos onde existam vários projetos sendo executados simultaneamente, que seja software livre, e que possua integrado a técnica de análise de pontos de função.

Dentre as ferramentas pesquisadas uma das mais completas foi o *DotProject* [14], que é um sistema de gerenciamento de projetos desenvolvidos em ambiente web com funcionalidades bastante úteis para o gerenciamento da equipe de projeto. O *DotProject* possui módulos para companhias, projetos, tarefas, (com gráficos de *Gantt*), fóruns, repositório de arquivos, calendário, contatos, *bug report*, suporte multi-linguagem e gerenciamento de permissões de usuários. Mas, como já foi dito, ela não possui nenhuma técnica para estimativa de tamanho de software.

1.2. Objetivo

Esse trabalho tem como objetivo a implementação, no *DotProject*, de um método de estimativa de projetos de software utilizando Análise de Pontos de Função, tendo como foco a estimativa de prazo para realização de tarefas, e a aplicação da ferramenta em um estudo de caso.

O *DotProject* [14] auxilia no gerenciamento das atividades da equipe de trabalho do projeto, mas o mesmo não possui nenhuma técnica integrada a ele para estimativa de tempo, sendo o cronograma de atividades dos integrantes da equipe criado totalmente de modo *ad hoc*. Depois da implementação será realizado um estudo de caso que será descrito nesse trabalho, sendo o objetivo principal desse Trabalho de Graduação.

O estudo de caso será realizado no projeto Sig@ (Sistema de Informações e Gestão Acadêmica) [15]. Atualmente a equipe do projeto utiliza o *DotProject*. O coordenador e os analistas utilizam a ferramenta para controlar e cadastrar os cronogramas com as atividades, estimativa de tempo e o responsável por cada uma delas. Atualmente a estimativa de tempo das atividades é feita de uma forma totalmente *ad hoc*, utilizando-se apenas da experiência de atividades passadas para fazer uma estimativa de duração de uma atividade. Com a utilização da Análise de Pontos de Função é esperada uma aproximação entre a previsão e a realidade, e com isso fazer um melhor aproveitamento dos recursos.

1.3. Estrutura do Trabalho

Além desse capítulo introdutório, este trabalho é composto também pelos seguintes capítulos:

- Capítulo 2: é descrita nesse capítulo a importância da mensuração do software na gerência de projeto, assim como serão mostrados alguns modelos de qualidade e também será visto alguns softwares livres para auxílio na gerência de projeto;
- Capítulo 3: esse capítulo possui um breve histórico sobre métricas funcionais;

- Capítulo 4: mostra o processo utilizado para realizar a contagem de pontos de função;
- Capítulo 5: é descrito como foi feita a implementação de uma técnica de medida de software no DotProject;
- Capítulo 6: um estudo de caso é realizado com o auxílio da ferramenta desenvolvida no Capítulo 5;
- Capítulo 7: neste último capítulo estarão sendo feitos os fechamentos do trabalho.

2. Importância da Mensuração do Software

2.1. Gerência de Projetos

De acordo com o PMI – *Project Management Institute* – [1] um projeto é um empreendimento temporário, com o objetivo de criar um produto ou serviço único. Temporário significa que cada projeto tem um começo e fim bem definido. Único significa que o produto ou serviço produzido é de alguma forma diferente de todos os outros produtos ou serviços semelhantes.

A gerência de projeto é a aplicação de conhecimentos, habilidades e técnicas para projetar atividades que visam atingir requisitos definidos e existem vários processos para ajudar no gerenciamento, que são agrupados em alguns passos básicos, tais como:

- Iniciação
- Planejamento
- Execução
- Controle
- Finalização

2.1.1. Planejamento

No planejamento os objetivos são definidos e refinados e o melhor curso de ação é selecionado. Na fase inicial de um projeto, como por exemplo, no levantamento de requisitos, ainda não existe um total conhecimento das características do produto que permita avaliar sua futura dimensão. Nesse caso é necessário fazer uma estimativa. Vários modelos de estimativa foram criados para fornecer métricas que permitam atender com menor margem de erro às necessidades de comunicação e informação do projeto.

Um dos modelos mais conhecidos é a análise de pontos de função, que permite medir o tamanho do sistema em termos da funcionalidade que é oferecida ao usuário, e também permite estimar o tamanho em qualquer fase do seu ciclo de vida, mesmo que os requisitos ainda não tenham sido detalhados.

A manutenção de registros de outros projetos semelhantes, com a evolução das estimativas iniciais até a medição final, permite um acompanhamento da relação entre a quantidade de pontos de função estimados nos vários estágios de conhecimento do produto. Na Figura 1, a área em destaque indica que durante as fases de um projeto, toda medição funcional realizada ainda é uma projeção do que de fato será entregue.

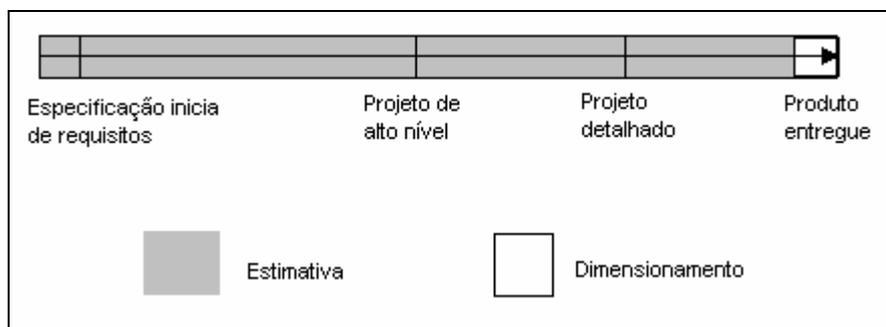


Figura 1: Pontos de contagem e estimativa ao longo do ciclo de vida do projeto

Ao final do projeto detalhado já temos uma quantidade de pontos de função mais próxima do produto final, mas ainda assim, até o produto final a quantidade de pontos pode sofrer alterações, pois no projeto detalhado o que vai existir é uma estimativa, e no produto final é obtido a quantidade real de pontos de função, ou seja, o tamanho real do projeto.

As métricas coletadas nas diversas fases do projeto não têm apenas valor quando analisadas isoladamente, mas também quando relacionadas entre si. Fazendo a comparação da evolução do tamanho funcional com a evolução de projetos passados, pode ser obtido um fator que indique o crescimento funcional entre as etapas do projeto. Ao identificar semelhanças com projetos passados, é possível ajustar as estimativas realizadas em fases iniciais, como descrito na Figura 2.

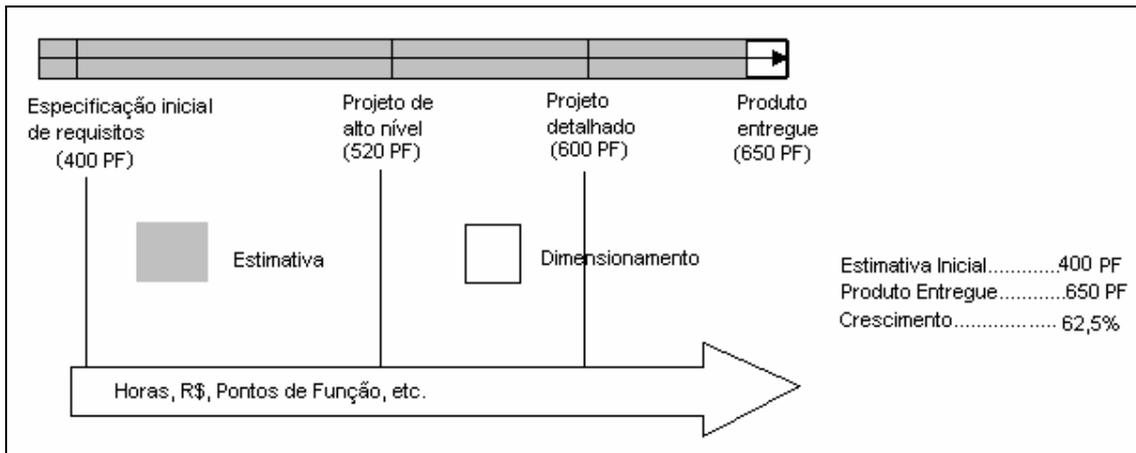


Figura 2: Variação do tamanho do projeto ao longo do seu ciclo de vida

O planejamento do projeto envolve equacionar um conjunto de aspectos que devem ser levados em conta, dentre eles:

- Quais artefatos devem ser produzidos (especificações, manuais, programas, módulos, base de dados, etc.);
- Utilização de quais atividades (construção de modelo funcional, identificação de classes de objetos, validação de documentação técnica, etc.);
- Quanto esforço deve ser empregado (horas/homem, equipe/mês);
- Quais são os profissionais (analistas, engenheiros de software, desenvolvedores, usuários, estagiários, etc.);
- Qual o prazo limite (semanas, meses, anos, etc.);
- Quais os riscos que devem ser identificados e gerenciados;

Sabendo disso, a experiência de profissionais contribuiu para a elaboração de planos mais reais. Porém, na área de tecnologia de informação, essa experiência é de certa forma perdida com o rápido avanço tecnológico e a alta rotatividade do pessoal envolvida no projeto. A falta de uma mensuração desses processos dificulta a experiência dos indivíduos em se transformarem em experiência coletiva.

Esforço, prazo e custo são algumas das informações selecionadas como objeto de medição e estimativa. Essas métricas são coletadas e distribuídas na forma de relatórios com a situação do projeto, seu progresso atual e as expectativas futuras.

2.1.2. Controle

Para se ter o controle é necessário o acompanhamento sistemático ou periódico, destinado a garantir o gerenciamento de ações, atividades, processos, resultados e recursos a fim de que os objetivos sejam alcançados, através do monitoramento e medidas regulares do progresso, identificando as variações no plano e tomando as ações corretivas conforme a necessidade.

A definição de meios de comparação do progresso real com o planejado é muito importante como parte do planejamento do projeto. O controle é uma das atividades mais importantes no gerenciamento de projeto. Colocando esses conceitos no contexto de um projeto de desenvolvimento de sistemas é possível ter algumas idéias na busca de respostas de por que medir, afinal, não se consegue controlar o que não se pode medir.

2.1.3. Adiamento do Projeto

A Figura 3 representa dinâmica de um desenvolvimento de projeto que não possui uma medida que tem como resultado o papel de permitir a comunicação efetiva entre os *stakeholders* do projeto.

Entre os resultados dessa dinâmica está o adiamento do encerramento do projeto, e até chegar a essa decisão de adiamento a equipe do projeto teve seu ânimo abatido, suas expectativas frustradas e o sentimento de impotência diante do inevitável, prejudicando a qualidade do produto ou serviço.

E a pergunta que deve ser feita é se será o último adiamento, visto que se as mesmas medidas forem adotadas com um novo prazo, não existirão garantias de finalização do projeto, a não ser quando for tarde demais para adotar novas medidas preventivas.

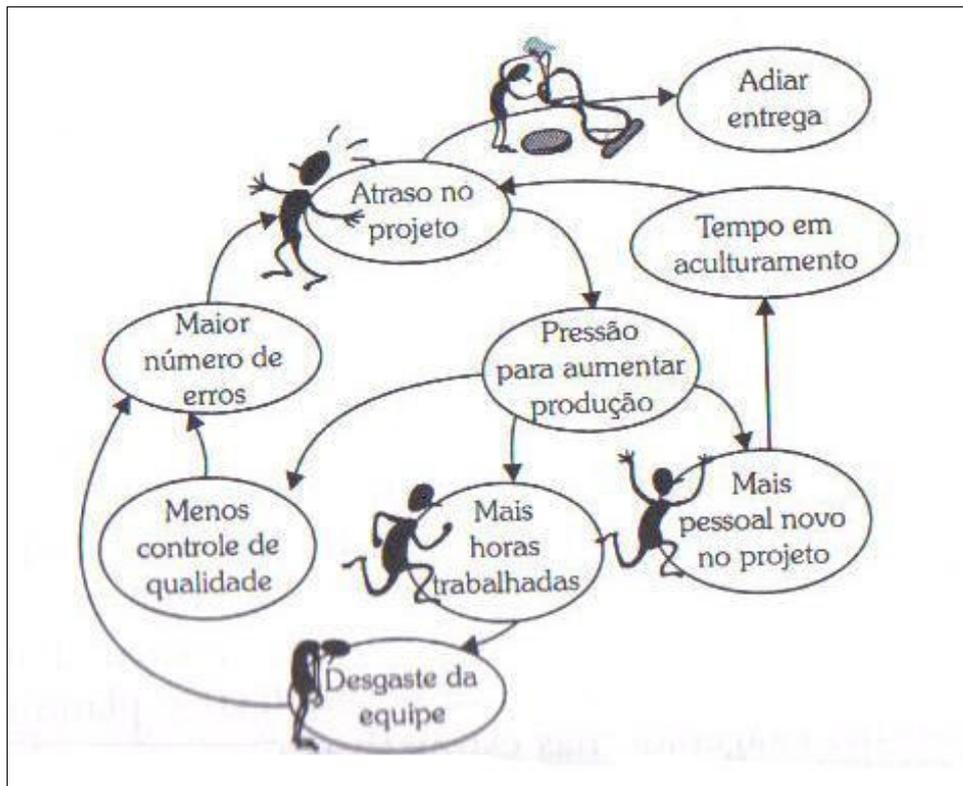


Figura 3: Dinâmica dos 99% concluídos [9]

A falta de métricas de projeto prejudica de forma geral o seu acompanhamento, uma vez que, apesar de o problema estar lá, ele não é percebido por aqueles que podem direcionar esforços na sua solução. O papel das métricas é permitir uma rápida identificação e correção dos problemas.

No capítulo 6 deste trabalho estaremos fazendo um estudo de caso para dimensionar uma atividade de um projeto e assim chegar a um cronograma estimado.

2.1.4. Foco

Um dos problemas do gerenciamento de projeto é a diferença da situação real em relação a que foi projetada. Para que esse desvio seja percebido é necessário analisar as métricas definidas e coletadas, assim como os indicadores gerados pelo relacionamento delas. Essa análise permite que vários aspectos dos projetos sejam monitorados, como os cronogramas, custos, qualidade, riscos e escopo.

É importante saber que o problema não é só o erro, é a demora em prevenir, identificar ou corrigir os defeitos. Aumentar os recursos nem sempre é a melhor

solução, pois não existe um relacionamento direto entre o prazo, o esforço e a quantidade de recursos disponíveis.

Com a utilização de métricas, justificar e defender as decisões tomadas ficam mais simples, pois o desvio entre o real e o planejado tende a ser menor, portanto mais fácil de identificar e gerenciar os erros com antecedência, tornando as conseqüências menos graves.

É importante destacar que falar em métrica de projeto não está se referindo unicamente à utilização de técnicas de dimensionamento, mas também de técnicas para ponderar o efeito de outros fatores como a análise de riscos e a determinação de *milestones*.

Esse trabalho procurou até aqui mostrar o papel que a utilização de métricas exerce no suporte ao empreendimento de atividades de desenvolvimento ou manutenção de sistemas com um enfoque de projeto.

2.2. Iniciativas de SPI (Software Process Improvement)

Pode-se definir SPI como um conjunto de metodologias e ferramentas que reduzem os riscos e aumentam a qualidade do *software* através de um conjunto de processos existentes, pelas modificações nesses processos ou atualizações de novos processos.

Existem alguns modelos que visam orientar a condução dessas ações, como por exemplo:

- SEI CMMI
- SPICE

Independentes do modelo escolhido, todos procuram tornar a atividade de desenvolvimento e manutenção de sistemas mais previsíveis, ou seja, que as estimativas sobre o comportamento das principais variáveis envolvidas, custo, tempo e qualidade, tenham mais chances de se confirmar nas suas respectivas medições.

2.2.1. SEI CMMI – Capability Maturity Model Integration

O modelo SEI [3] é o mais difundido no Brasil, sendo o que atualmente representa a maior tendência em termos de iniciativas de SPI. Esse modelo procura a melhoria contínua do processo de desenvolvimento de software (Figura 4).

Para classificar o grau de maturidade de cada empresa, o CMMI prevê a evolução do processo através de cinco níveis de maturidade, e o tempo necessário para passar de um nível para outro dependerá de cada empresa, do seu foco nas práticas de melhoria e do seu apoio gerencial. Os cinco níveis de maturidade são:

1. **Inicial:** o processo de software é caracterizado como *ad-hoc* e até mesmo ocasionalmente caótico. Poucos processos são definidos e o sucesso depende do esforço individual.
2. **Repetível:** os processos básicos de gestão de projetos são estabelecidos para acompanhar custo, cronograma e funcionalidade. A necessária disciplina do processo existe para repetir sucessos anteriores em projetos com aplicações similares.
3. **Definido:** o processo de software para as atividades de gestão e engenharia é documentado, padronizado e integrado em processo de software padrão para a organização. Todos os projetos utilizam uma versão aprovada do processo de software padrão para desenvolver e manter software.
4. **Gerenciado:** medidas detalhadas do processo de software e da qualidade do produto são realizadas. O processo e os produtos de software são quantitativamente compreendidos e controlados.
5. **Otimização:** a melhoria contínua do processo é propiciada pelo feedback quantitativo do processo e pelas idéias e tecnologias inovadoras.

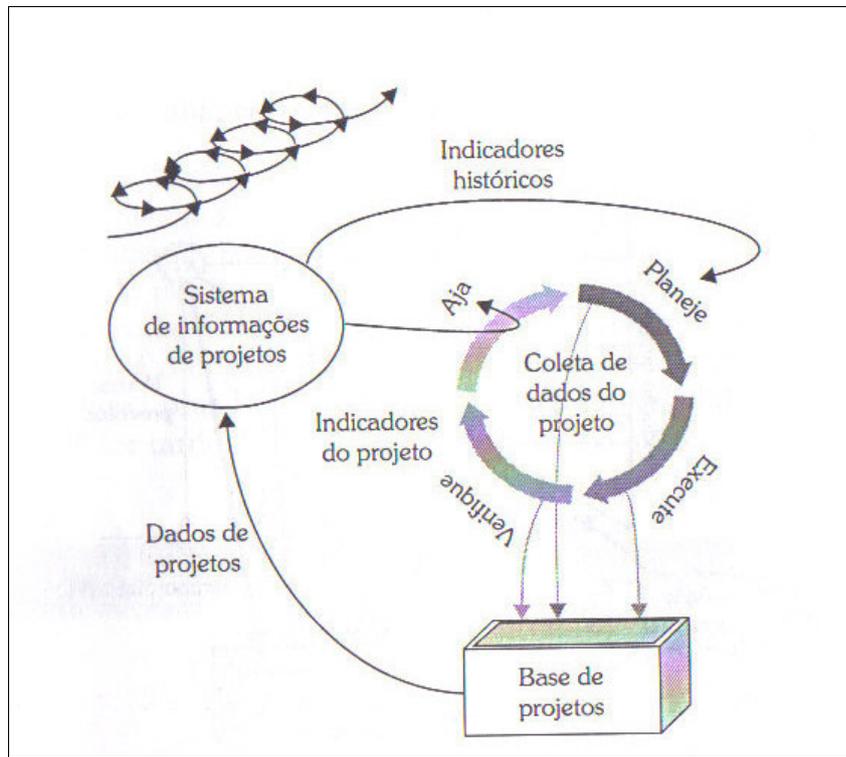


Figura 4: Melhoria contínua [9]

Esses níveis são estruturados em um conjunto de atividades relacionadas, que quando são executadas em conjunto, atingem um conjunto de objetivos considerados relevantes para a melhoria da capacidade de processos. Esses conjuntos de atividades são chamados de *Process Área (PA)*.

Os PA do segundo nível de maturidade são mostrados abaixo. O foco das PA desse nível é a gerência de projeto.

- Gerência de Requisitos;
- Planejamento de projeto;
- Monitoramento de controle de Projetos;
- Gerenciamento de acordos com o fornecedor;
- Medição e análise;
- Garantia de qualidade de produto e processo;
- Gerência de configuração.

A área de processo Medição e Análise têm como objetivo desenvolver e sustentar uma capacidade de medição usada para apoiar gerencialmente as necessidades de informação. Esta área envolve:

- Especificação dos objetivos de medição e análise de forma que estes sejam alinhados com as necessidades de informação identificados e objetivos;
- Especificação das medidas, mecanismos de coleta de dados e de armazenamento, técnicas de análise, e mecanismos de comunicação e de feedback;
- Fornecimento de resultados objetivos que podem ser usados na tomada de decisão e implementação de ações corretivas apropriadas.

A Medição e Análise relacionam-se diretamente com as seguintes áreas de processo do CMMI:

- Planejamento de Projeto (nível 2) para fornecer informações sobre as estimativas dos atributos de projeto e outras necessidades de informação para planejamento.
- Monitoração e Controle de Projeto (nível 2) para atender as necessidades de informação do monitoramento da performance do projeto.
- Gerência de Configuração (nível 2) para fornecer informação para o gerenciamento da medição dos artefatos, tais como: código fonte de programas, documento de requisitos, componentes, casos de testes etc.
- Desenvolvimento de Requisitos (nível 3) para apoiar a apresentação dos requisitos do cliente e as necessidades de informação relacionadas.
- Gerência de Requisitos (nível 2) para subsidiar com informações sobre manter a rastreabilidade dos requisitos e necessidades de informação relacionadas.
- Definição do Processo Organizacional (nível 3) para o estabelecimento do repositório de medições da organização.
- Gerência do Projeto Quantitativa (nível 4) para apoiar o entendimento da variação e o uso apropriado das técnicas de análise estatística. Esta nova área de processo é fortemente baseada no modelo Practical Software Measurement (PSM). O PSM fornece um guia baseado em experiências de como definir e implementar um processo de medição orientado à informação para projetos de

software. De um ponto de vista prático, o modelo procura resolver dois problemas: como especificar formalmente os indicadores a serem usados? ; como conduzir o processo de medição?

2.2.2. SPICE – Software Process Improvement and Capability Determination (ISO/IEC 15504)

O SPICE [5] é uma norma elaborada pelo ISO/IEC [2]. Ela constitui um padrão para avaliação de processo de software, visando determinar a capacitação dos processos de uma empresa e orientar a empresa para uma melhoria contínua de seus processos.

Um grupo de estudos da ABNT [6] participa do processo de desenvolvimento do SPICE, além de trabalhar na tradução da norma para o português.

O SPICE pode ser utilizado por organizações para planejar, gerenciar, monitorar, controlar, melhorar a aquisição, fornecimento, desenvolvimento, operação, evolução e suporte de software.

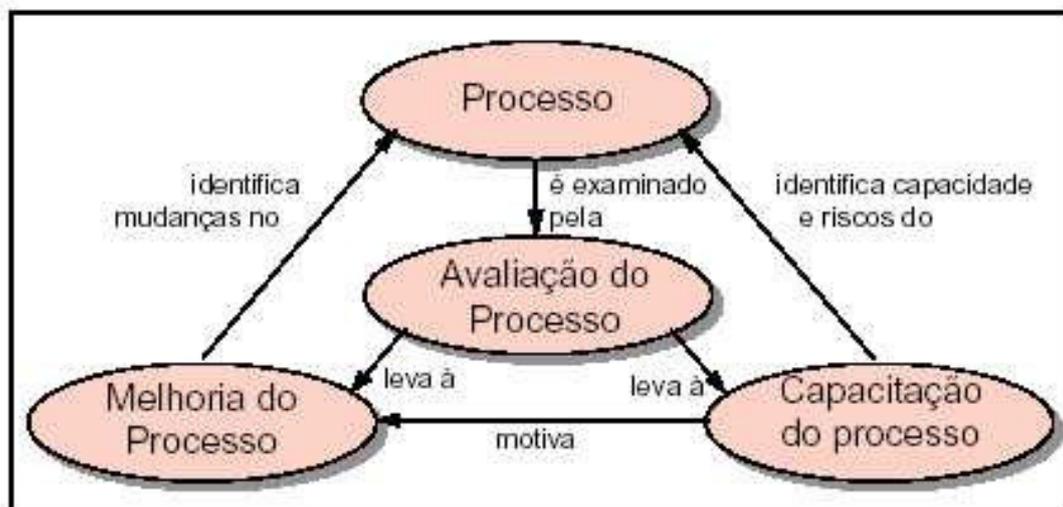


Figura 5: Avaliação do processo de software [26]

De acordo com a visão do SPICE (figura 5 – avaliação do processo de software - SPICE), a avaliação de processo de software tem como propósito:

- Entender os estados dos processos para sua melhoria;
- Determinar a adequação dos processos para um requisito particular ou uma classe de requisitos;

- Determinar a adequação dos processos para um determinado contrato ou para uma classe de contratos.

O SPICE possui um modelo de referência que serve de base para o processo de avaliação. Este modelo é um conjunto padronizado de processos fundamentais, visando uma boa prática da engenharia de software. O modelo é dividido em cinco categorias de processo: Cliente-Fornecedor, Engenharia, Suporte, Gerência e Organização. Cada uma dessas categorias é detalhada em processos mais específicos, que são detalhados pela norma.

Além dos processos, o SPICE possui seis níveis de capacitação de cada processo:

1. **Nível 0 (Incompleto)** – Não existem artefatos de trabalho e as saídas dos processos dificilmente são identificadas;
2. **Nível 1 (Realizado)** – O objetivo geralmente é atingido, mas não necessariamente de forma planejada e controlada;
3. **Nível 2 (Gerenciado)** – O processo produz os artefatos de trabalho com qualidade aceitável e no prazo, de forma planejada e controlada;
4. **Nível 3 (Estabelecido)** – O processo é realizado e gerenciado usando um processo bem definido, baseados em princípios de Engenharia de Software;
5. **Nível 4 (Previsível)** – O processo é realizado de forma consistente, dentro dos limites de controle para atingir os objetivos;
6. **Nível 5 (Otimizado)** – A realização do processo é otimizado para atender às necessidades do negócio.

Dentro do contexto de melhoria de processos (figura 5), a avaliação do processo significa a caracterização das práticas correntes de um projeto, em termos da capacidade dos processos selecionados. A análise dos resultados é feita em relação às necessidades de negócio do projeto, identificando os aspectos positivos e negativos, e os riscos associados aos processos.

Isto leva a determinar se os processos estão atingindo efetivamente seus resultados esperados e identificar possíveis causas para uma baixa qualidade, altos custos ou tempo excessivo, que indica a necessidade de uma melhoria do processo. A

utilização de métricas, em conjunto com indicadores do projeto, se tornam imprescindível para realizar essa avaliação do processo.

2.3. Ferramentas Livres de Auxílio ao Gerenciamento de Projetos

Um modelo de qualidade de software descreve o que deve ser feito para a melhoria dos processos, definindo níveis de maturidade que são organizados em áreas de processo que possuem objetivos nos quais são alcançados por práticas que produzem artefatos como resultados dos processos. Para se alcançar esses resultados, e conseqüentemente aplicarem o modelo de qualidade, se faz necessário a utilização de ferramentas de software.

Atualmente existe um movimento crescente onde grande esforço individual e coletivo, impulsionados pelo poder da internet, derivam a criação de softwares livres com objetivo de atender as necessidades comuns das organizações e de indivíduos. Estes softwares, devido à liberdade de qualquer um utilizá-lo e/ou modificá-los, estão cada vez mais presentes nas organizações, nos usos domésticos e até mesmo no governo, onde temos como exemplo o governo brasileiro que cria políticas de incentivo a utilização de software livre.

Nesse contexto, abaixo vai ser apresentada três ferramentas não proprietárias, que segundo o artigo *Ferramentas Livres para Suporte ao Atendimento de Áreas de Processo do CMMI Nível 2* [8] são as que melhor dão apoio ao gerenciamento de projeto.

2.3.1. USC-COCOMO II

COCOMO II [11] é um modelo que permite realizar uma estimativa de custo, esforço e cronograma.

USC-COCOMO II [12] é uma ferramenta CASE de análise que tem por objetivo gerar um resultado automatizado visando facilitar a administração de projetos de software.

USC-COCOMO II estima custo e tempo baseado em pessoa/mês e meses, respectivamente, para a determinação do *baseline* de exigências de um produto para a

conclusão de uma atividade. O software prevê uma adição de 20% do tempo computado que é sua margem de erro que podem ser provocados pelos riscos que um projeto possui.

Para obter o custo por fase, já que o COCOMO gera só o esforço/fase, é necessário gerar o custo médio de pessoa-mês: $\text{custo médio} = \frac{\text{custo total}}{\text{esforço total}}$. A este cálculo é assumido um adicional de 7% ao esforço total.

O custo do projeto inteiro é a soma dos custos dos módulos individuais. Considerando que estas são apenas estimativas, são usadas tabelas para exibir o custo atual, gerando uma estimativa otimista e outra pessimista para o custo do projeto inteiro.

Para mensurar o tamanho do sistema é necessário escolher qual medida será adotada (linhas de código, pontos de função).

O USC-COCOMO II apresenta um método de calibração, aconselhado para menos de 8 projetos. Esta calibração realiza a análise dos dados que foram fornecidos e gera como resposta uma estimativa.

Através desta ferramenta CASE também podem ser gerados pequenos relatórios com estimativas associando a cada etapa do processo.

2.3.2. Rambutan

Rambutan (Requirements Management Tool for Busy System Analysts) [13] é um conjunto de aplicações que ajuda um analista de sistemas na coleta e categorização de fatos para um documento de especificação de requisitos.

Em seu estado atual o produto consiste de dois programas com funções similares, sendo um rodado em aplicativos móveis (Palm), que é utilizado para fazer a coleta dos dados no local do cliente e outro sendo rodado em desktops que serve para editar e refinar mais as exigências do cliente, podendo ser utilizado no escritório do analista, por exemplo.

2.3.3. DotProject

O DotProject [14] é um framework de gerenciamento de projetos desenvolvido em ambiente Web. Incluem módulos para companhias, projetos, tarefas (com gráfico de Gantt), fóruns, repositório de arquivos, calendário, contatos, bug report, suporte multi-linguagem e gerenciamento de permissões de usuários.

O DotProject foi implementado utilizando a linguagem de programação PHP e JavaScript, e utiliza como sistema de gerenciamento de banco de dados o mysql. O DotProject se mostrou uma ferramenta muito valiosa no gerenciamento de projeto. A ferramenta pode ser utilizada para controlar e cadastrar os cronogramas das atividades do projeto, colocando uma estimativa de tempo e o responsável por executar a tarefa, assim como reportar devidos bugs do projeto.

Os desenvolvedores do projeto podem utilizar a ferramenta para visualizar suas atividades, para reportar (*time sheet*) as realizações diárias, assim como cadastrar lições aprendidas nos fóruns.

Dentre as vantagens do DotProject temos que é uma ferramenta pra ser utilizada na Web, podendo ser acessado de qualquer lugar por pessoas autorizadas. Outra vantagem é ter um repositório único, e a usabilidade, pois possui uma interface gráfica simples e intuitiva.

A limitação do DotProject está em não possuir uma comparação entre o que se estima de um projeto e o que realmente acontece. A ferramenta não possui nenhuma técnica de medida de software, como a Análise de Pontos de Função, por exemplo.

E uma das propostas desse trabalho de graduação é implementar uma técnica de medidas de software no DotProject. A técnica a ser utilizada na implementação vai ser a análise de pontos de função, focada na estimativa de cronograma pra atividades do projeto e como trabalho futuro iremos propor também a utilização de outras técnicas de medidas de software, como por exemplo, Pontos de Caso de Uso e Pontos de Função de Implementação. A proposta de implementação poderá ser visto com mais detalhes no capítulo 5, e um estudo de caso utilizando essa implementação poderá ser vista no capítulo 6.

3. Métricas Funcionais

Neste capítulo será apresentado um breve histórico da análise de pontos de função e também citar outros métodos de medições funcionais desenvolvidos e utilizados com certa relevância em vários países.

3.1. Histórico

A técnica de análise de pontos de função surgiu no início da década de 70, na IBM [16], como uma alternativa às métricas baseadas em linhas de código. Allan Albrecht foi o responsável por medir a produtividade de vários projetos de software desenvolvidos pela IBM. Esses projetos tinham sido desenvolvidos em uma grande variedade de linguagens, tornando inviável uma análise conjunta de produtividade. Com isso se motivou a busca por uma medida que fosse independente da linguagem de programação utilizada.

Após a apresentação da técnica à comunidade, no fim da década de 70, e dos sucessivos trabalhos de Capers Jones destacando a importância da técnica, houve um crescimento do número de usuários de pontos de função. Esse crescimento motivou, em 1986, a criação do *International Function Point Users Group* – IFPUG [17]. Juntamente com a aceitação pelo mercado, surgiram diversas variações do modelo apresentado por Allan Albrecht. No entanto, em 1990 surgiu a primeira versão do Manual de Práticas de Contagem ou CPM – *Counting Practices Manual* com objetivo de padronizar a técnica de análise de pontos de função. E desde então é o padrão reconhecido pela indústria.

Atualmente a versão mais recente do CPM é a Release 4.2. Embora tenham surgido outras técnicas de medição funcional, como Mark II [18] e COSMIC-FFP [19], nenhuma delas possui aceitação tão ampla pelo mercado quanto a análise de pontos de função do IFPUG.

3.2. IFPUG

O IFPUG é uma entidade sem fins lucrativos, compostos por pessoas e empresas de diversos países, com a finalidade de promover um melhor gerenciamento dos processos de desenvolvimento e manutenção de software com uso de análise de pontos de função e outras técnicas de medição.

O IFPUG promove uma série de ações, dentre elas:

- **Conferência Anual:** especialistas renomados apresentam o estado da arte na área de métricas de software e os usuários têm a oportunidade de trocar experiências.
- **Seminários e Workshops Educacionais:** cobrem uma série de tópicos como prática de contagem de pontos de função, técnicas de gerenciamento de projeto e melhoria de processos.
- **Certificação Profissional:** programa que envolve a certificação dos usuários da técnica como especialistas em pontos de função (CFPS), de materiais de treinamento de fornecedores e de ferramentas de software para apoio ao processo de contagem.
- **Comitês e Grupos de Trabalho:** é o centro das atividades do IFPUG. Dentre os comitês existentes temos: o comitê de certificação, responsável pelo programa de certificação; o comitê de comunicação e marketing, responsável pela comunicação entre os membros; o comitê de conferência, responsável pelas conferências anuais; o comitê de práticas de contagens, responsável pelo CPM; o comitê de educação, responsável pelos seminários e workshops; o comitê de performance de TI, responsável por planejar, gerenciar e melhorar as práticas e processos de engenharia de software; o comitê de novos ambientes, que desenvolve e provê diretrizes para a aplicação de pontos de função em novos ambientes; o grupo de trabalho de questões acadêmicas, que coordena os esforços entre o IFPUG e a comunidade acadêmica interessada no ensino e pesquisa de métricas de software; o grupo de trabalho do ISBSG representa o IFPUG no ISBSG; o grupo de trabalho do padrão ISO trabalha na aderência do CPM ao padrão de medição de tamanho funcional (ISO/IEC 14143).

Além dessas ações, o IFPUG possui uma diretoria em que um dos objetivos é estimular a criação de associações locais (*chapters*) para trocar experiências e fortalecer uma rede de contatos entre pessoas com objetivos comuns: a análise de pontos de função e métricas funcionais.

3.3. BFPUG

A técnica de análise pontos de função começou a ser usada significativamente no Brasil no começo da década de 90. Entre 1991 e 1994 foram realizados seis ENUPF – Encontro Nacional de Usuários de Pontos de Função, com a participação de palestrantes internacionais. Porém o interesse do mercado começou a ganhar força quando grandes contratos públicos eram fechados baseados em pontos de função. E em paralelo, o interesse de empresas de desenvolvimento de software por modelos de qualidade (ISO e CMM) contribuiu para o aumento no interesse do assunto.

Em 1998, membros do IFPUG [20] fundaram o BFPUG - *Brazilian Function Point Users Group* – e não demorou muito para o BFPUG se tornar o *chapter* do IFPUG no Brasil. O BFPUG possui centenas de associados, dentre estudantes, desenvolvedores, consultores e gerentes de sistema. O BFPUG promove a troca de experiências entre os profissionais, palestras e também o exame de certificação do IFPUG no Brasil.

3.4. NESMA

NESMA – *Netherlands Software Metrics Users Association* – [21] é uma associação de métricas da Holanda, e utiliza um padrão diferente de medida funcional do que o apresentado pelo IFPUG. Porém suas ações e objetivos são bem próximos aos do IFPUG, inclusive com uma colaboração bem próxima entre ambas.

A NESMA possui seu próprio manual de contagem de pontos de função, atualmente na versão 2.0, cuja primeira versão foi baseada no manual do IFPUG. Utiliza a mesma filosofia, conceitos, termos e regras do IFPUG, com algumas diretrizes diferentes. Isso permite que uma contagem de pontos de função de um projeto de desenvolvimento ou de uma aplicação utilizando as duas abordagens cheguem a um resultado bem próximo.

Entretanto ambas organizações possuem abordagens distintas para a aplicação da análise de pontos de função em projetos de melhoria de software. Atualmente a NESMA é um dos maiores grupos de usuários de pontos de função da Europa.

3.5. MARK II

Também conhecido como MK II foi formulado por Charles Symons na década de 80, inspirado na proposta de Albrecht. Seu trabalho tornou amplamente conhecido em 1988, após uma publicação na revista *IEEE – Transactions on Software Engineering*.

Entretanto a técnica não obteve muito sucesso no começo, pelo fato de ser um método proprietário de uma empresa de consultoria. Mas atualmente o método é de domínio público e a associação responsável por manter o padrão da técnica é a associação de métricas do Reino Unido – UKSMA, que também possui ações e objetivos similares ao do IFPUG. No entanto, é um método de aplicação restrita no Reino Unido.

3.6. COSMIC – FFP

Em 1997 um grupo de pesquisadores da Universidade de Quebec desenvolveu um novo método de medição funcional para sistemas de tempo real, denominado Full Function Points (FFP). Posteriormente, experimento de campo mostrou que esse método pode ser aplicado também a sistemas de informações tradicionais, o que motivou os pesquisadores a trabalhar numa versão melhorada do método.

Em 1998, um grupo de especialistas em medição de software, criou o COSMIC – *Common Software Measurement Internacional Consortium* com o objetivo de desenvolver um novo método de medição de tamanho funcional baseada nas melhores características dos métodos existentes e que agregasse novas idéias. Esse método, proposto em 2000, batizado de COSMIC-FFP, na prática foi um refinamento do FFP.

Ainda não é uma técnica tão disseminada quanto à do IFPUG, mas observa-se que muitas pesquisas estão sendo feitas sobre esse método. Pode-se dizer que o mercado acompanha atentamente esse método.

3.7. Padrão ISO de Medição Funcional

Medição funcional é um termo geral para métodos de dimensionam o tamanho do software a partir das funções requisitadas pelo usuário. No começo da década de 90 podiam ser reconhecidos vários métodos de medição funcional, surgidos a partir da Análise de Pontos de Função proposto por Albrecht. Com o objetivo de acabar com a inconsistência existente entre esses métodos e criar um método mais rigoroso de medição funcional, grupos de trabalho de métricas de software da Austrália, Reino Unido, Holanda e Estados Unidos formaram um grupo de trabalho denominado WG12 (Working Group 12), subordinado ao SC7 – *Sub-Committee Seven do JTCl – Joint Technical Committee One* estabelecido pela ISO – *International Organization for Standardization* em conjunto com o IEC – *International Engineering Consortium*.

Como resultado dos trabalhos do WG12, foi estabelecido um conjunto de padrões internacional chamado de norma 14143 (ISO/IEC 14143). A norma ISO/IEC 14143 foi estabelecida para garantir que todos os métodos de medição funcional sejam baseados em conceitos semelhantes, e possam ser testados para assegurar o comportamento similar e da forma esperada pelo método, dependendo dos domínios funcionais a que se aplicam.

No fim de 2002 a técnica de análise de pontos de função mantida pelo IFPUG, foi aprovado como um método de medição funcional dentro dos padrões exigidos pela norma ISO/IEC 14143 e tem a denominação ISO/IEC 20926:2002. Porém, essa técnica é contemplada apenas até a determinação dos pontos de função não ajustados. As características gerais do manual de contagem do IFPUG, que são utilizadas para estabelecer o fator de ajuste, possuem requisitos tecnológicos e de qualidade o que tornam essa parte do processo excluída do padrão de medição funcional da ISO.

Atualmente, são quatro os métodos de medição funcional que são aderentes ao ISO/IEC 14143:

- IFPUG CPM 4.1 (ISO/IEC 20926)
- NESMA CPM 2.1 (ISO/IEC 24570)
- Mark II (ou MK II) CPM 1.3.1 (ISO/IEC 20968)
- COSMIC-FFP *Measurement Manual* 2.2 (ISO/IEC 19761)

A figura 6 mostra a evolução desses métodos ao longo dos anos.

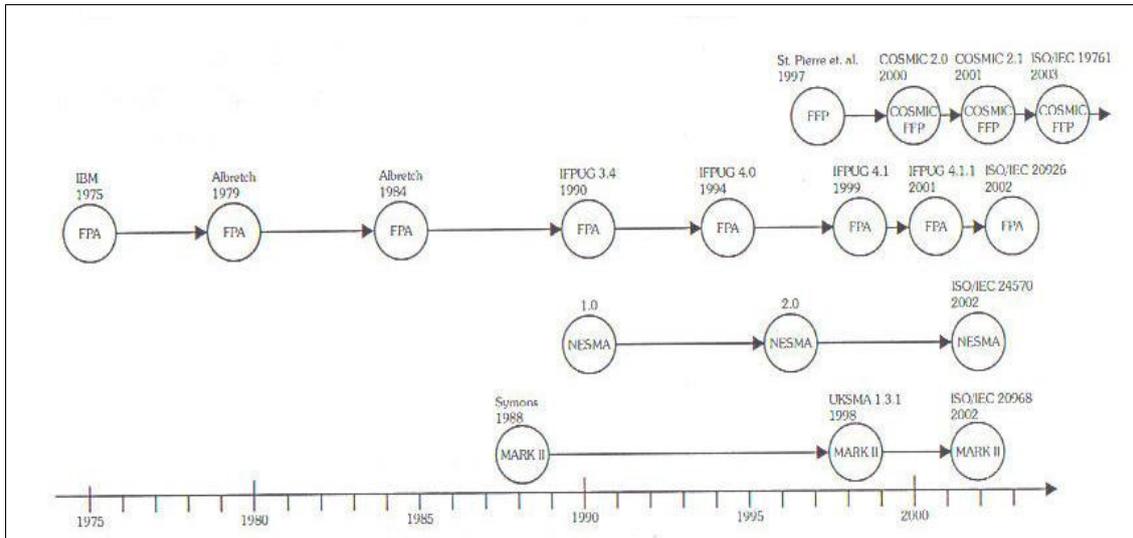


Figura 6: Evolução dos métodos [25]

3.8. ISBSG

O ISBSG – *International Software Benchmarking Standards Group* – [22] é uma organização sem fins lucrativos com sede na Austrália, resultado de uma iniciativa em conjunto com diversas organizações de métricas de software do mundo. A missão do ISBSG é “manter um repositório de métricas de projeto de software que possa ajudar na gestão dos recursos de TI pela melhoria das estimativas de projeto e produtividade, análise de riscos e *benchmarking*”.

Seu repositório de dados contém informações de mais de 2000 projetos de software de vários países, constituindo assim uma valiosa ferramenta de *benchmarking*. Além disso, permite que sejam feitas pesquisas e análise abordando temas relacionados a estimativa, produtividade, boas práticas, etc. Periodicamente o ISBSG publica um artigo chamado “*The Software Metrics Compendium*” contendo uma análise estatística do seu repositório.

Esse repositório pode ser alimentado por organizações do mundo todo e de diferentes áreas de negócio. Qualquer organização pode contribuir para o repositório com as informações do seu projeto, e os atributos a serem coletados incluem o tamanho funcional do projeto, esforço, custo, duração, qualidade do produto final contexto do projeto, ambiente de desenvolvimento, ferramentas e técnicas empregadas.

4. Análise de Pontos de Função

Nesse capítulo estaremos apresentando uma visão geral do processo de contagem de pontos de função ao qual serviu de base para a implementação mostrada no Capítulo 5.

Para adequar-se ao padrão ISO/IEC de medição funcional, o IFPUG tornou o fator de ajuste opcional na aplicação da técnica de pontos de função. E de acordo com pesquisas realizadas, pela NESMA [21], mostrou que o uso do fator de ajuste não traz nenhum benefício adicional aos pontos de função não-ajustados para a estimativa de esforço. Dessa maneira uma parte do processo de contagem foi excluída desse trabalho, que é justamente a que trata da determinação do valor do fator de ajuste e conseqüentemente dos pontos de função ajustado, pois o objetivo da implementação está diretamente relacionada com a estimativa de esforço.

Antes de explorar o processo, algumas considerações será feitas sobre o processo de definição de requisitos, que envolve desde a identificação e o entendimento da necessidade de negócio do usuário até a tradução dessa necessidade em funcionalidades que podem ser utilizadas na contagem de pontos de função.

4.1. Os requisitos e a Contagem de Pontos de Função

A análise de pontos de função é uma técnica para medir o desenvolvimento do software do ponto de vista do usuário, pela quantificação das funcionalidades a eles oferecidas. Nessa atividade, o processo de identificação de requisitos tem um papel fundamental. É através dele que as necessidades dos usuários são mapeadas para características e funcionalidades de um software, as quais poderão ser medidas ou contadas.

Os requisitos funcionais é a base para o cálculo dos pontos de função não-ajustados. As contagens com a finalidade de medição são viabilizadas à medida que o projeto evolui e novas características e detalhamento acerca dos requisitos são

descobertos. Ainda assim, a precisão alcançada com a contagem depende da qualidade e maturidade dos requisitos identificados.

A partir do momento que a visão do usuário é interpretada e completada pela visão do desenvolvedor, os requisitos começam a adquirir características mais fortes como:

- Mais específicos e sem ambigüidade;
- Fornecem descrição integrada de todas as necessidades do usuário, incluindo a normalização das necessidades de vários usuários;
- Suas terminologias podem ser entendidas por ambos;
- Tendência à estabilidade;

Com essas características os requisitos se tornam mais precisos para fazer o cálculo de pontos de função. Independentemente de a contagem ser uma estimativa ou um dimensionamento, todas as premissas assumidas e os resultados obtidos devem ser documentados como forma de manter um histórico de cada etapa da contagem, para que possam ser consultadas futuramente.

4.2. Objetivos e Benefícios da Análise de Pontos de Função

A análise de pontos de função é uma técnica para medir o tamanho funcional do software do ponto de vista do usuário. O IFPUG define como objetivos primários da técnica:

- Medir a funcionalidade que o usuário solicita e recebe;
- Medir o desenvolvimento e manutenção de software de forma independente da tecnologia utilizada na implementação.

Além desses objetivos, o processo de contagem de pontos de função deve ser:

- Simples o suficiente para minimizar o trabalho adicional envolvido no processo de medição;
- Uma medida consistente entre vários projetos e organizações.

Dentre os benefícios da aplicação da análise de pontos de função em uma organização, podem-se destacar:

- Suporte a análise de produtividade e qualidade;
- Apóia o gerenciamento de escopo de projetos;
- Complementa o gerenciamento de requisitos;
- Um meio de estimar custo e recursos;
- Uma ferramenta para fundamentar a negociação de contratos;

4.3. Propósito da Contagem de pontos de Função

Sempre existe um propósito para a contagem de pontos de função. Por exemplo: contagem com o propósito de medir o serviço entregue por um fornecedor para sua posterior remuneração. De acordo com a motivação podem-se assumir algumas premissas que podem agilizar o processo. Logo, é possível efetuar contagens com níveis diferenciados de detalhe e também de precisão. Sendo mais objetivo, o propósito da contagem de pontos de função é fornecer uma resposta a um problema de negócio.

Como esse trabalho tem o propósito de estimar o esforço aplicado a uma atividade e depois estimar o prazo de conclusão, a etapa do processo de contagem referente ao fator de ajuste não vai ser mostrada nesse trabalho.

4.4. Processo de Contagem de Pontos de Função

A Figura 7 apresenta as etapas do processo de contagem apresentados nesse trabalho.

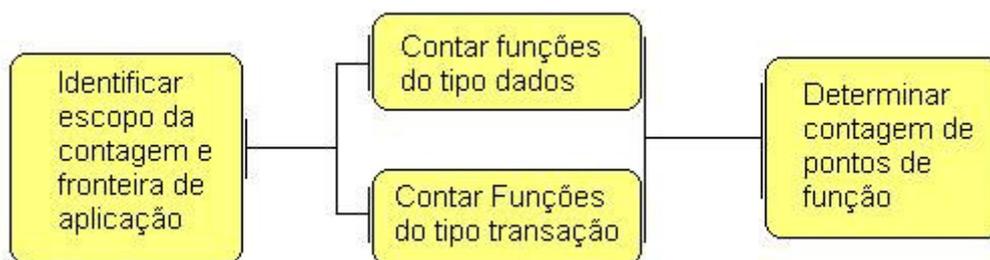


Figura 7: processo de contagem de pontos de função

4.4.1. Identificando Escopo da Contagem e Fronteira da Aplicação

Escopo da Contagem

O escopo define se a contagem abrangerá um ou mais sistemas ou apenas parte de um sistema. Por exemplo, o escopo da contagem de uma aplicação pode abranger:

- Todas as funcionalidades disponíveis;
- Apenas as funcionalidades efetivamente utilizadas pelo usuário;
- Apenas algumas funcionalidades específicas (relatórios, transações cadastrais, etc.).

Fronteira da Aplicação

A fronteira de aplicação é a interface conceitual que delimita o software que será medido e o mundo exterior (seus usuários). Quando há mais de uma aplicação incluída no escopo de uma contagem, várias fronteiras devem ser identificadas.

O IFPUG especifica as seguintes regras para a determinação da fronteira da aplicação:

- Sua determinação deve ser feita com base no Ponto de Vista do Usuário. O foco deve estar no que ele pode entender e descrever;
- A fronteira entre aplicações deve ser baseada na separação das funções conforme estabelecido pelos processos do negócio, não em considerações tecnológicas;

4.4.2. Contagem de Funções do Tipo Dados



Figura 8: Funções do tipo dados

As funções do tipo dados representam as funcionalidades fornecidas pelo sistema ao usuário, para atender as suas necessidades de dados. São classificadas em **Arquivo Lógico Interno (ALI)** e **Arquivo de Interface Externa (AIE)**.

Arquivo Lógico Interno é um grupo logicamente relacionado de dados ou informações de controle, identificável pelo usuário, mantido dentro da fronteira de aplicação sendo contada. Sua principal intenção é armazenar dados mantidos através de um ou mais processos da aplicação sendo contada. Exemplo: tabelas de banco de dados atualizadas pela aplicação.0

Arquivo de Interface Externa é um grupo logicamente relacionado de dado ou informações de controle, identificável pelo usuário, mantidos fora da fronteira de aplicação sendo contada. Sua principal intenção é armazenar dados referenciados de um ou mais processos elementares dentro da fronteira de aplicação sendo contada. Exemplo: tabelas de banco de dados lidas pela aplicação, mas atualizadas por outra aplicação.

Cada ALI e AIE devem ser classificados com relação à sua complexidade funcional (baixa, média ou alta) com base em:

- Número de Tipos de Dados (TD)
- Número de Tipo de Registros (TR)

Determinadas as quantidades de tipos de dados e de tipos de registros, a classificação com relação à complexidade é fornecida pelos valores mostrados na tabela abaixo:

		Tipos de Dados		
		<20	20-50	>50
Tipos de Registros	1	Baixa	Baixa	Média
	2-5	Baixa	Média	Alta
	>5	Média	Alta	Alta

Tabela 1: Complexidade funcional dos ALI e AIE.

Assim um ALI contendo 33 tipos de dados e 2 tipos de registros é de complexidade média e outro AIE contendo 44 tipos de dados e 1 tipo de registro é de complexidade baixa.

Após a determinação da complexidade dos arquivos, deve-se calcular a contribuição deles utilizando os valores mostrados na tabela abaixo:

Tipo de Função	Baixa	Média	Alta
Arquivo Lógico Interno	7 PF	10 PF	15 PF
Arquivo de Interface Externa	5 Pf	7 PF	10 PF

Tabela 2: Contribuição dos pontos de função não-ajustados do tipo dado

Por exemplo, um ALI de complexidade média contribui com 10 pontos de função e um AIE também com complexidade média contribui com 7 pontos de função.

4.4.3. Contagem de Funções do Tipo Transação

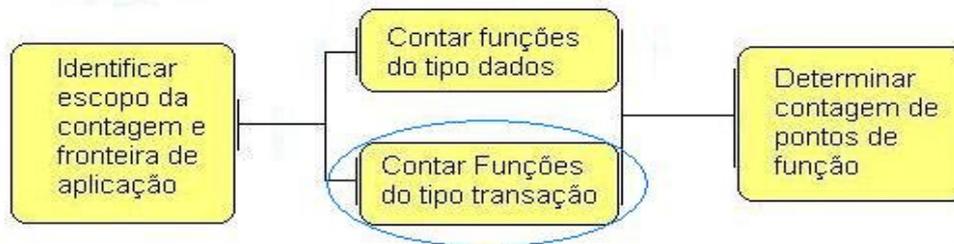


Figura 9: Funções do tipo transação

As funções do tipo transação representam as funcionalidades de processamento de dados fornecidas pelo sistema ao usuário. São classificadas em **Entrada Externa (EE)**, **Saída Externa (SE)** e **Consulta Externa (CE)**.

Entrada Externa é um processo elementar que processam dados ou informações de controles originados de fora da fronteira da aplicação. Sua principal intenção é manter um ou mais arquivos lógicos internos e/ou alterar o comportamento do sistema. Exemplo: incluir cliente, alterar cliente, excluir cliente.

Saída Externa é um processo elementar que envia dados ou informações de controle para fora da fronteira de aplicação. Sua principal intenção é apresentar informação ao usuário através de lógica de processamento que não seja apenas uma simples recuperação de dados ou informações de controle. Seu processamento deve conter cálculo, ou criar dados derivados, ou manter um arquivo lógico interno, ou alterar o comportamento do sistema. Exemplo: relatório de totais de faturamento por cliente.

Consulta Externa é um processo elementar que enviam dados ou informações de controle para fora da fronteira de aplicação. Sua principal intenção é apresentar informações ao usuário através da simples recuperação de dados ou informações de controle de um LAI ou AIE. Exemplo: consulta cadastro clientes.

Cada Entrada Externa, Saída Externa e Consulta Externa deve ser classificada em relação à sua complexidade funcional (baixa, média ou alta) baseada em:

- Número de Arquivos Referenciados (AR)
- Número de Tipos de Dados (TD)

Quando se determinar as quantidades de arquivos referenciados e de tipos de dados, a classificação com relação à complexidade é fornecida pelos valores dados nas tabelas 3 e 4:

		Tipos de Dados		
		<5	5-15	>15
Arquivos Referenciados	<2	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	>2	Média	Alta	Alta

Tabela 3: Complexidade funcional para EE

		<6	6-19	>19
		Arquivos Referenciados	<2	Baixa
2-3	Baixa		Média	Alta
>3	Média		Alta	Alta

Tabela 4: Complexidade funcional para SE e CE

Assim, podemos ter como exemplo uma EE com 16 tipos de dados e 2 arquivos referenciados sendo de complexidade alta. Uma SE com 16 tipos de dados e 2 arquivos referenciados é de complexidade média. E uma CE com 19 tipos de dados e um arquivo referenciado é de complexidade baixa.

Após determinar a complexidade das funções do tipo transação, deve-se calcular sua contribuição utilizando a tabela abaixo:

Tipo de Função	Baixa	Média	Alta
Entrada Externa	3 PF	4 PF	6 Pf
Saída Externa	4 PF	5 PF	7 PF
Consulta Externa	3 PF	4 PF	6 Pf

Tabela 5: Contribuição dos pontos de função não-ajustados das função tipo transação

4.4.4. Determinando a contagem de Pontos de Função Não-Ajustados



Figura 10: Contagem dos pontos de função

Após a identificação e classificação de todas as funções de contagem, vista nos tópicos anteriores, o número de pontos de função não-ajustados será simplesmente a soma do valor de cada uma dessas funções.

5. Implementação

Uma vez que, o contexto e a importância de uma técnica de medição de software para gerência de projeto foi discutida nesse trabalho, este capítulo vai mostrar como foi feita a integração da técnica de análise de pontos de função ao DotProject.

A técnica será utilizada com a finalidade de estimar um cronograma de uma atividade do projeto.

5.1. Ferramenta e Tecnologia Escolhida

Não fazia sentido desenvolver uma ferramenta apenas para medir software, porque esse tipo de ferramenta já existe. Mas tomando como base o artigo *Ferramentas Livres para Suporte ao Atendimento de Áreas de Processo do CMMI Nível 2* [3], onde foi feita uma pesquisa de programas para apoiar a gerência de projeto, verificou que o programa mais completo nesse sentido era o DotProject (ver Seção 2.3), porém a mesma não possui nenhuma ferramenta integrada que utilize um método de estimativa de prazo. Com isso, motivou a implementação de uma técnica de medida de software, para estimar cronograma de atividades, totalmente integrada ao DotProject.

Como o DotProject foi implementado utilizando a tecnologia PHP com o sistema de gerenciamento de banco de dados mysql, a implementação mostrada nesse capítulo também foi feita utilizando PHP [ref] e MySQL [ref].

5.2. Modelagem do Negócio

Alguns módulos foram identificados para a implementação da técnica de análise de pontos de função no DotProject, assim como outros módulos foram criados para fazer uma estimativa de prazo para as atividades do projeto. Os módulos implementados foram:

- Produtividade Tecnologia
- Produtividade Pessoa (RH)

- Calcular Pontos de Função
- Estimativa de Prazo

A figura 11 representa a tela principal da ferramenta desenvolvida. Por essa figura percebe-se a total integração da ferramenta com o DotProject.

The screenshot shows a software interface with several tabs: Details, Dates, Dependencies, Human Resources, and FPA (selected). The FPA tab contains the following elements:

- Task Owner:** A dropdown menu with "Ferreira, Joao" selected.
- Task Type:** A dropdown menu with "Unknown" selected.
- Access:** A dropdown menu with "Public" selected.
- Web Address:** An empty text input field.
- Task Parent:** A dropdown menu with "None" selected.
- Target Budget:** A text input field containing "R\$0.00".
- Move this task (and its children), to project:** A dropdown menu with "Modulo de Perfil" selected.
- Description:** A large text area for entering task details.
- Buttons:** "Select contacts..." and "Associar com a Tarefa".

Below the FPA tab is a table with the following columns: Descrição, Tipo, TD, AR/TR, Complexidade, and PF. The table is currently empty. Below the table, there is a note: "Clique aqui para visualizar todas as funcoes inseridas ate agora." and a list of abbreviations: "ALI, AIE, EE, CE, SE, TD, TR, e AR".

At the bottom of the interface, there are several calculation and association tools:

- Calculo de Ponto de Funcoes Nao Ajustados:** A text input field followed by a "Calcular" button.
- Estimativa de prazo:** A text input field followed by a "Calcular" button.
- Associar a tarefa:** A button.
- Produtividade:** A label.

Figura 11: Tela principal da ferramenta

5.2.1. Produtividade x Tecnologia

Esse módulo é responsável por estabelecer o valor da produtividade em horas/métrica utilizando como parâmetro alguma tecnologia, como por exemplo, java, c, cobol, etc.

A modelagem desse módulo foi feita de uma maneira que se aceitasse inserir outras métricas que não fosse pontos de função, como por exemplo, pontos de caso de uso, ou pontos de função de implementação. Foi feito dessa maneira porque em um trabalho futuro é pretendido colocar outras técnicas de medidas de software no DotProject.

Como a técnica que está sendo implementada nesse momento é a de pontos de função, alguns valores já foram atribuídos nas tabelas de produtividade tecnologia. O

relacionamento da tabela de produtividade com a de tecnologia foi preenchido com os seguintes valores:

Ambiente/Linguagem	Produtividade (HH/PF) Em Horas
COBOL	15,60
Assembler	17,60
Visual Basic	5,80
Delphi	5,80
C	15,60
C ++	7,80
Visual C++	6,80
VB–Script	12,00
HTML	5,10
Java	6,80

Tabela 6: Produtividade média em relação a tecnologia

Essa produtividade reflete a média das empresas de tecnologia de informação, de acordo com a lista de discussão do BFPUG. Obviamente, a produtividade do Sig@, na hora em que se fizer o estudo de caso, pode ser maior ou menor que essa apresentada, mas com o tempo e uma base que guarde o histórico de produção da equipe, esses valores vão se ajustando a realidade.

5.2.2. Produtividade x Pessoa

Sabe-se que duas pessoas podem ter produtividade diferente utilizando a mesma tecnologia, portanto a utilização da produtividade utilizando como apoio a tecnologia, muitas vezes pode refletir um erro na hora de se fazer uma estimativa.

Esse módulo é responsável por estabelecer o valor da produtividade em horas/métrica utilizando como parâmetro algum recurso humano, como por exemplo, a produtividade de um engenheiro de software, de um analista. Em um exemplo, é

possível cadastrar que o engenheiro de software Renato tem uma produtividade de 6 horas/ponto de função.

A modelagem desse módulo foi feita de uma maneira que se aceitasse inserir outras métricas que não fosse pontos de função, como por exemplo, pontos de caso de uso, ou pontos de função de implementação. Foi feito dessa maneira porque em trabalho futuro é pretendido colocar outras técnicas de medidas de software no DotProject.

Uma vez que obtenha o histórico do fator de produtividade da pessoa, é possível através dos PFs aplicar o fator e obter o esforço necessário para o desenvolvimento da atividade realizada por essa pessoa que foi cadastrada no DotProject como responsável pela atividade. Após o cálculo do esforço, é possível gerar um cronograma para a atividade a qual foi mensurada.

5.2.3. Calcular Pontos de Função

Esse módulo vai ser baseado no fluxo de contagem de pontos de função descrito no Capítulo 4 e vai ter uma ligação direta com o DotProject, pois as tarefas cadastradas nele, vão possuir o tamanho em PF. O algoritmo para fazer a contagem foi feito utilizando a linguagem JavaScript.

5.2.4. Estimativa de Prazo

Esse módulo é o responsável pela estimativa de prazo da atividade, utilizando como parâmetros a quantidade de pontos de função calculada e o fator de produtividade a ser utilizada. Quando estabelecer o prazo para uma atividade, esse prazo vai poder ser associado a tarefa, e automaticamente já é atualizada no banco de dados utilizado pelo dotProject.

5.3. Modelagem de Negócios Futuros

5.3.1. Inferência

Quando se possuir uma boa base de indicadores, vai ser possível modificar o fator de produtividade de algum recurso humano através do histórico de produtividade dessa pessoa, tornando assim o processo de estimativa de atividade mais automatizado.

5.3.2. Estimativa de Custo Financeiro

Assim como é feita a estimativa de prazo, poderá ser feita também a estimativa financeira, com base no esforço gasto. Por exemplo, se um projeto possui 1000 pontos de função, e é conhecido que o valor do ponto de função de uma empresa é de R\$ 30, eu posso estimar que o projeto tem um valor de R\$ 30.000,00.

6. Estudo de Caso

O estudo de caso tem como objetivo realizar a contagem de pontos de função para uma atividade do projeto Sig@, e com isso poder estimar um cronograma utilizando a quantidade de pontos de função e o fator de produtividade.

Esse estudo de caso utilizará a ferramenta implementada no DotProject descrita no Capítulo 5.

6.1. Descrição do Sistema SIG@

O projeto Sig@(Sistema de Informações e Gestão Acadêmica) teve início em 2001, vinculado ao NTI (Núcleo de Tecnologia e Informação), e é responsável pelo desenvolvimento do Sig@UFPE. No projeto são utilizados softwares livres para auxiliar no desenvolvimento, produção e gerenciamento do projeto.

O Sig@ é um sistema cooperativo para gestão universitária, está implantado na UFPE, na Escola Politécnica de Pernambuco e na UNIVASF (Universidade Federal do Vale do São Francisco), e possui vários subsistemas como Ensino, Controle de Acesso, Financeiro, Pessoal, Estrutura Física e Gerencial.

O sistema foi desenvolvido no padrão arquitetural em camadas, com JSP/Servlet na camada de apresentação e Java nas demais camadas, e utiliza como tecnologia proprietária apenas o banco de dados Oracle, mas em estudo à adoção da solução livre Hibernate, o que possibilitará a utilização de banco de dados livres. Para ter uma idéia do sistema, pode-se dizer que o sistema possui aproximadamente 2500 classes java e 290 tabelas no banco.

Para auxiliar no gerenciamento do projeto algumas ferramentas são utilizadas, e para fazer o estudo de caso, iremos utilizar a ferramenta de gerenciamento de projeto DotProject. O Dotproject é utilizado por toda a equipe do projeto. O coordenador e os analistas utilizam a ferramenta para controlar e cadastrar o cronograma com as atividades, estimativa de tempo e responsável; e para reportar os bugs do sistema para

devidas correções. Os desenvolvedores utilizam a ferramenta para visualizar suas atividades, para reportar (*time sheet*) as realizações diárias, e para cadastrar as lições aprendidas em um fórum.

6.2. Lista de Atores

- Discentes

6.3. Especificação do Caso de Uso

Caso de uso identificado:

- Preenchimento do Questionário Perfil Social

O estudo de caso será realizado no subsistema de recursos humanos, tendo como base o questionário de perfil social. Este questionário é destinado aos discentes e faz parte do projeto nacional chamado **Conexão de Saberes: diálogo entre a Universidade e espaços populares** que está sendo desenvolvido pelo MEC/SECAD em parceria com diferentes universidades federais do país.

Um dos seus principais objetivos consiste em buscar subsídios para a construção de novos parâmetros que possam servir de base para a concepção e execução de políticas públicas para o ensino universitário que tenham como preocupação assegurar **o acesso e a permanência com qualidade** de estudantes das classes populares, portadores de necessidades especiais e com dificuldades de concluir seus cursos dentro do prazo.

As informações buscadas sobre alguns aspectos da trajetória do estudante, acesso e desempenho na instituição de ensino superior servirão de base para as futuras ações na UFPE tanto do ponto de vista pedagógico como administrativo, capazes de contribuir para alcançar os objetivos traçados.

As respostas são importantes para compor um quadro avaliativo da situação e dos desafios enfrentados no cotidiano dos estudantes universitários. Os dados do questionário não serão divulgados individualmente, serão tratados apenas por processos estatísticos.

No Apêndice C é mostrado o questionário para preenchimento.

6.4. Identificação e Classificação das funcionalidades

Seguindo o processo de contagem descrito no Capítulo 4, iremos identificar as funções do Tipo Dados e em seguida as Funções do Tipo Transação.

6.4.1. Identificando funções do Tipo Dados

Nesse estudo de caso foram encontrado duas funções do tipo dado, um Arquivo Lógico Interno e outro Arquivo de Interface Externa.

- Discente (AIE)

- Tipo de dados: nome, matrícula, cpf, data de nascimento, sexo, curso.

- Tipo de registros: Pessoa, Vínculo.

- Perfil Social (ALI)

- Tipo de dados: matrícula, turno, ano de primeira admissão em universidade pública, dificuldade no aprendizado, descrição da dificuldade, deficiência física, descrição da deficiência, cor/etnia, tipo da instituição de ensino médio que estudou, local de moradia da família, renda do pai, renda da mãe, endereço dos pais, escolaridade do pai, escolaridade da mãe.

- Tipo de registros: Perfil Social, Vínculo.

Utilizando a ferramenta implementada no DotProject chegamos aos valores de complexidade descrito na tabela abaixo.

Descrição	Tipo	TD	TR	Complexidade
Discente	AIE	6	2	BAIXA
Perfil Social	ALI	15	2	BAIXA
Legenda: (Tipo) – Classificação da Funcionalidade (ALI, AIE). (TD) – Quantidade de tipo de dados (TR) – Quantidade de tipo de registros				

Tabela 7: Formulário de contagem do tipo dado

6.4.2. Identificando funções do Tipo Transação

Foram identificadas também funções do tipo transação, onde temos duas Entradas Externas e uma Consulta Externa. Assim detalhando os processos elementares identificados temos:

- Registrar Perfil Social (EE):
 - Principal Intenção: Inserir o ALI Perfil Social.
 - Tipo de dados: comando (botão enviar), mensagem para o usuário, turno, ano de primeira admissão em universidade pública, dificuldade no aprendizado, descrição da dificuldade, deficiência física, descrição da deficiência, cor/etnia, tipo da instituição de ensino médio que estudou, local de moradia da família, renda do pai, renda da mãe, endereço dos pais, escolaridade dos pais.
 - Arquivos referenciados: Perfil Social.
- Alterar Perfil Social (EE):

- Principal Intenção: Alterar o ALI Perfil Social.

- Tipo de dados: comando (botão enviar), mensagem para o usuário, turno, ano de primeira admissão em universidade pública, dificuldade no aprendizado, descrição da dificuldade, deficiência física, descrição da deficiência, cor/etnia, tipo da instituição de ensino médio que estudou, local de moradia da família, renda do pai, renda da mãe, endereço dos pais, escolaridade dos pais.

- Arquivos referenciados: Perfil Social.

- Consultar Perfil Social (CE):
 - Principal Intenção: Apresentar dados dos ALIs Perfil Social e Pessoa.

 - Tipo de dados: matrícula, turno, ano de primeira admissão em universidade pública, dificuldade no aprendizado, descrição da dificuldade, deficiência física, descrição da deficiência, cor/etnia, tipo da instituição de ensino médio que estudou, local de moradia da família, renda do pai, renda da mãe, endereço dos pais, escolaridade do pai, escolaridade da mãe, nome, cpf, data de nascimento, sexo.

 - Arquivos referenciados: Perfil Social, Pessoa.

Utilizando a ferramenta implementada no DotProject chegamos aos valores de complexidade descrito na tabela abaixo.

Descrição	Tipo	TD	AR	Complexidade
Registrar Perfil Social	EE	15	1	Baixa
Alterar Perfil Social	EE	15	1	Baixa
Consultar Perfil Social	CE	19	2	Média
Legenda: (Tipo) – Classificação da Funcionalidade (EE, SE, CE). (TD) – Quantidade de tipo de dados (AR) – Quantidade de arquivos referenciados				

Tabela 8: Formulário de contagem do tipo transação

6.5. Cálculo de Pontos de Função

6.5.1. Cálculo de Pontos de Função Não-Ajustados

Com o auxílio da ferramenta, calculamos o total de pontos de função, a tabela 9 descreve os totais por tipo de função.

Tipo de Função	Complexidade Funcional	Totais por tipo de complexidade	Totais por tipo de função
AIE	1 Baixa x 7=	7	7
	0 Média x 10=	0	
	0 Alta x 15=	0	
ALI	1 Baixa x 5=	5	5
	0 Média x 7=	0	
	0 Alta x 10=	0	
EE	2 Baixa x 3=	6	6
	0 Média x 4=	0	
	0 Alta x 6=	0	
CE	0 Baixa x 3=	0	4
	1 Média x 4=	4	
	0 Alta x 6=	0	

Tabela 9: Contribuição das funções para o questionário de perfil social

Sendo assim percebemos que temos um total de **22 pontos de função** não ajustados para essa atividade de preenchimento do questionário de perfil funcional.

6.6. Aplicação na Estimativa de Prazo

Utilizando uma produtividade de 6.5 h/PF, que foi a produtividade utilizando a linguagem Java que foi cadastrada no banco, encontramos uma estimativa para conclusão da atividade de 143 horas. Depois de achada a estimativa de prazo da atividade, a mesma foi associada ao cronograma de atividades no DotProject

7. Conclusão e Trabalhos Futuros

7.1. Conclusão

Esse trabalho mostrou que métricas têm sido utilizadas pelas organizações para serem usadas com o intuito estratégico através da melhoria do processo de software. O gerenciamento de projetos deve ser auxiliado, portanto pela utilização de uma métrica, que permita à mensuração de um projeto e conseqüentemente a geração da estimativa de prazo, custo e recursos.

No estudo de caso apresentado falamos da ferramenta DotProject, que auxilia o gerenciamento de projeto. Por ser uma ferramenta *open source*, isto nos permitiu implementar uma ferramenta que utiliza a técnica de análise de pontos de função para estimar cronograma para atividades do projeto. A escolha desse programa se deveu ao fato dela, além de ser *open source*, ser uma ferramenta *web* e é utilizado no Sig@ (Sistema de Informação para Gestão Acadêmica), onde foi aplicada a ferramenta desenvolvida no estudo de caso.

Verificamos mediante as pesquisas realizadas para elaboração esse trabalho, que profissionais e organizações, que têm intenção de iniciar a utilização da análise de pontos de função para estimativas, buscam pelo conhecimento da técnica através de auto-estudo ou participando de algum treinamento. Em seguida, procuram por indicadores de mercado para a produtividade. E por fim, chegam a conclusão de que os resultados obtidos são muito diferentes da realidade da empresa, e acabam portanto abandonando a iniciativa.

O principal argumento para não-utilização de indicadores gerados internamente é exatamente a falta de dados básicos para a sua geração, especialmente base histórica. A melhor forma de fazer uma estimativa de esforço a partir da estimativa de tamanho da atividade é utilizar dados internos da própria organização, como o número de horas apropriadas em atividades passadas, em conjunto com a quantidade de pontos de função.

O estudo de caso foi aplicado, como já dissemos, no projeto Sig@. Este não possui disponibilidade de indicadores de produtividade, e por isso a técnica deve ser aplicada de maneira conveniente. Para isso será necessário identificar as atividades já realizadas, e dimensioná-las, e em seguida levantar dados de esforço. Dessa maneira poderá gerar indicadores internos de produtividade. Outro meio seria começar a gerar os indicadores a partir das atividades que surgem ao longo do tempo, com isso a estimativa de prazo de uma atividade a princípio não seria muito precisa, mas com o tempo a estimativa iria ficar mais próxima da real, ou seja, em um primeiro momento é mais interessante utilizar a ferramenta para coletar dados do dimensionamento da atividade e gerar indicadores de produtividade, para posteriormente começar a realizar estimativas que irão ser mais precisas.

Esse trabalho serviu como um primeiro passo para uso de uma técnica de medida de software no projeto Sig@, pois anteriormente não haviam sido utilizadas técnicas de medidas de software. Com essa iniciativa esperasse que em médio prazo a utilização da ferramenta instalada no DotProject sirva para melhorar a alocação de recursos do projeto.

Este é um processo contínuo e a cada iteração o papel dos indicadores passa a ter maior relevância no processo de planejamento. Inicialmente sendo apenas um apoio para, com o tempo, passar a ser um instrumento fundamental na determinação das variáveis de projeto.

Vimos que o uso da técnica de análise de pontos de função pode trazer vários benefícios para o gerenciamento de um projeto, tais como o suporte a análise de produtividade e qualidade, apoio ao gerenciamento de escopo, ser um meio de estimar custo, prazo e recursos, e ainda fundamentar a negociação de contratos. Esperamos com esse trabalho contribuir para melhorias do gerenciamento de projeto.

7.2. Trabalhos Futuros

Melhorias na ferramenta têm que ser feita, e outras funcionalidades implementadas, mas o que existe já pode ser utilizado para criar um *baseline* do projeto e gerar uma estimativa de prazo sem muita precisão por causa da falta de indicadores. A

coleta de indicadores de produtividade e de dimensionamento é o próximo passo a ser adotado para que estimativas fiquem mais precisas.

A implementação foi feita de um jeito que pode ser utilizada em outra ferramenta sem levantar grande esforço para instalação, e em breve deve estar sendo implantada no software GMP (Gerenciador de Multi Projetos), que é desenvolvido no Centro de Informática da UFPE, e tem também como objetivo auxiliar no gerenciamento de projetos.

O desenvolvimento de outras técnicas de medição de software podem ser aplicadas no DotProject, aproveitando a modelagem que foi utilizada para implementar a técnica de análise de pontos de função, assim como adicionar novas funcionalidades, como estimativa de custo e dedução da produtividade através do histórico da base de dados de uma maneira automatizada.

Referências

1. PMI. **A Guide to The Project Management Body of Knowledge**. Project Management Institute, 2000.
2. ISO (International Organization for Standardization), IEC (International Electrotechnical Commission) “**ISO/IEC 15504 Software Process Improvement and Capability Determination Model (SPICE)**”, www.iso.org, 2004.
3. CMMI Product Team “**Capability Maturity Model Integration (CMMI) for Software Engineering Version 1.1, Staged Representation**”, Carnegie Mellon, USA, 2002
4. JONES, C. **Applied Software Measurement**. 2. ed. McGraw-Hill, 1996.
5. SPICE, “**ISO/IEC 15504 Software Process Improvement and Capability Determination Model (SPICE)**”, www.iso.org, 2004.
6. ABNT (Associação Brasileira de Normas Técnicas) “**NBR ISO 8402/1994 – Gestão de Qualidade e Garantia de Qualidade**”, Terminologia, Brasil, 1994.
7. DEKKERS, C.; EMMONS, B. **How Function Points Support the Capability Maturity Model Integration**. CrossTalk, fev. 2002.
8. SOUZA, R.A.C, OLIVEIRA, S.R.B., **Ferramentas Livres para Suporte ao Atendimento de Áreas de Processo do CMMI Nível 2**, 2005.
9. VAZQUEZ, C.E., SIMÕES, G.S., ALBERT, R.M. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software**, 3 ed. Érica, 2005.
10. BRAGA, A. **Análise de Pontos de Função**, IBPI PRESS, 1996.
11. BOEHM, B. et al. **Software Cost Estimation With COCOMO II**. Prentice Hall, 2000.
12. **USC-COCOMOII**, sunset.usc.edu/research/COCOMOII, 2005
13. Rambutan, **Requirements Management Tool for Busy System Analysts**, rambutan.sourceforge.net, 2005.
14. DotProject, **Open Source Software**, www.dotproject.net, 2005
15. **Sistema de Informação para Gestão Acadêmica (Sig@)**, www.siga.ufpe.br, 2005.
16. **IBM**, www.ibm.com, 2005

17. IFPUG. **Function Point Counting Practices Manual: Versão 4.2.** International Function Point Users Group, www.ifpug.org, jan., 2004.
18. UKSMA. **Mk II Function Point Analysis Counting Practises Manual: Versão 1.3.1.** United Kingdom Software Metrics Association, www.ukσμα.co.uk, set. 1998.
19. COSMIC. **COSMIC-FFP Measurement Manual: Versão 2.2.** Common Software Measurement International Consortium. Jan, 2003.
20. BFPUG. **Brazilian Function Point User Group**, www.bfpug.com.br, 2005
21. NESMA. **Netherlands Software Metrics Users Association**, www.nesma.nl, 2005
22. ISBSG. **International Software Benchmarking Standards Group**, www.isbsg.org, 2005.
23. IEEE . **Transactions on Software Engineering**, www.ieee.org, 2005
24. Hazan, C. **Uma ferramenta na busca da excelência Medições de Software**, Portal do Serpro, www.serpro.gov.br, 2004.
25. JTC1/SC7. **Software life cycle processes.** ISO. Geneva. Switzerland, 1994.
26. Tsukumo, A.N., Rêgo, C.M., Azevedo, G.F., **Qualidade de Software: Visões de Produto e Processo de Software**, II ERI da SBC/SP, jun, 1996

Apêndice A

Nome*: EDUARDO VINICIUS DE FIGUEIREDO SA

CPF*: 032.637.164-84

Data Nascimento*: 22/03/1979

Sexo*: MASCULINO

Curso*: CIÊNCIA DA COMPUTAÇÃO - CIN

Email principal*: EDUARDO.SA@UFPE.BR

Email secundário*: EVFS@CIN.UFPE.BR

Qual(is) turno(s) você cursa, predominantemente*?

M/T MANHA TARDE NOITE T/N

Em que ano você ingressou pela primeira vez nesta ou em outra universidade pública*?

AAAA

Você sente dificuldade de acompanhar o curso ao qual está vinculado*?

Sim Não

Em caso afirmativo, mencione as principais dificuldades.

Obs.: Não utilize caracteres especiais (' , * , * , * , ! , etc).

Portador de Necessidades Especiais:

Auditiva Visual Física

Cor/etnia (auto-identificação)*:

Amarelo(as iática) Branca Indígena Parda Preta

Em que tipo de estabelecimento de ensino você cursou o Ensino Médio*?

em estabelecimento público em estabelecimento NÃO público maior parte em estabelecimento público maior parte em estabelecimento NÃO público

Como você definiria o local de moradia de sua família*:

Comunidade / favela / morro Conjunto Habitacional Loteamento Popular Bairro de Classe Média

Bairro de Classe Média Bairro Popular

Alta

Assinale distintamente a renda do Pai e da Mãe ou das pessoas que exerceram esse papel.

Não teve pai/mãe ou pessoa que exerceu tal papel na criação.

Renda do Pai R\$ Renda da Mãe R\$

Localidade de moradia dos pais (bairro, município, cep, se souber) ou daqueles que exerceram tal papel em sua criação (se seus pais têm endereços diferentes, opte pelo endereço do responsável com quem você usualmente mora/moraria)

CEP: 52011000

Logradouro: RUA JOAQUIM NABUCO

Número: 483 Complemento: AP.203

Bairro: DERBY

Pais: BRASIL

UF: PERNAMBUCO

Cidade: RECIFE

Qual é a escolaridade dos seus pais (* ou daqueles que exerceram tal papel em sua criação)?

Pai* Superior COMPLETO

Mãe* Superior COMPLETO

Figura 12: Questionário Perfil Social