

CENTRO UNIVERSITÁRIO FEEVALE

CARLOS MARCELO FABIAN

ESTIMANDO PRAZOS E CUSTOS DE APLICAÇÕES
UTILIZANDO ANÁLISE DE PONTOS POR FUNÇÃO

Novo Hamburgo, Novembro de 2007.

CARLOS MARCELO FABIAN

ESTIMANDO PRAZOS E CUSTOS DE APLICAÇÕES
UTILIZANDO ANÁLISE DE PONTOS POR FUNÇÃO

Centro Universitário Feevale
Instituto de Ciências Exatas e Tecnológicas
Curso de Ciência da Computação
Trabalho de Conclusão de Curso

Professor Orientador: Sandra Teresinha Miorelli

Novo Hamburgo, Novembro de 2007.

AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

A DEUS que sempre esteve comigo, aos meus pais que desde cedo vislumbraram que o meu futuro seria a área de ciência da computação, investiram em mim. A minha esposa que sempre me apoiou incondicionalmente para chegar até onde cheguei.

Aos amigos e às pessoas que convivem comigo diariamente, minha gratidão.

RESUMO

A necessidade constante de aumento da qualidade e melhoria dos processos de desenvolvimento de software, implicando assim na melhora da produtividade e por consequência na redução de custos têm se percebido uma tônica das empresas que fabricam programas de computador, conhecidas como “*software houses*”. A qualidade do software depende de um considerável nível de processo de desenvolvimento bem estruturado (Pressman, 2002). Até os dias atuais, percebemos que as empresas assumem riscos em projetos estipulando prazos e custos baseando-se no “*feeling*” e/ou experiências em projetos anteriores. Os resultados obtidos na maioria das vezes nem sempre são satisfatórios, prazos estourados, sem falar no alto custo de desenvolvimento (Pressman, 2002). Em virtude disto, podemos ver algumas empresas adotando modelos de qualidade de software, visando melhorar os seus processos. Este trabalho de conclusão tem como objetivo estudar os processos da empresa Rech Informática Ltda., empresa fabricante de software desde 1990. Estabelecida em Novo Hamburgo/RS, a Rech Informática tem como processo principal a prestação de serviços na área de tecnologia da informação, utilizando como ferramenta de apoio o SIGER®, ERP desenvolvido pela empresa. Com crescimento constante, a Rech Informática tem uma considerável demanda de implementações, e ainda tem necessidade de criar novos recursos para manter e conquistar novos clientes. Partindo deste pressuposto, a direção da empresa e colaboradores da área de planejamento de projetos aponta a complexidade de estimar prazos, custos e gerenciar a concorrência da demanda. A proposta deste trabalho é analisar software existente e verificar a possibilidade de desenvolver um protótipo de software que venha a auxiliar no gerenciamento, utilizando a técnica da Análise de Pontos por Função (APF), por indicação da direção da empresa a qual trabalho. Para construção do protótipo proposto, será utilizado como estudo de caso um dos projetos em desenvolvimento, que compreende novas funcionalidades e quebra de algumas restrições da base de dados atual do módulo de Contabilidade SIGER®. O protótipo proposto será desenvolvido pela mesma ferramenta de desenvolvimento que a empresa utiliza no seu ERP, baseado na metodologia estruturada de seus programas fonte.

Palavras-chave: Prazos, Custos, Produtividade, Pontos por função.

ABSTRACT

The constant efforts to increase quality and improve the software development process, in order to improve production and lower costs have been the companies which made programs focus, known as “software houses”. The software quality depends of a considerable structured development process level (Pressman, 2002). Nowadays, we can see the companies taking risks on projects getting dates and costs based on their feelings and/or experiences from the last project. The results don’t satisfy many times, late projects and a higher development costs (Pressman, 2002). That’s why we can see companies adopting software quality models, to avoid their processes. This graduate theory studies Rech Informática Ltda processes, who made software since 1990. Established in Novo Hamburgo/RS, the Rech Informática main process is information technology services, using a support tool known as “SIGER®”, the ERP developed by the company. The company is always growing, it has a considerable implementation demand, and need to create new features to keep and get new clients. From this subject, the company directors and the project planning department employees says that it’s hard to get dates, costs and manage a high level of software demand.

This study suggests an analysis of existing softwares and probably develops a software prototype which has one goal, to help on managing the development demand, based on function point analysis (FPA), asked by the company directors where I have been working. To develop this software prototype, one of the company’s projects that are being produced will be used to study, which has new functions and break some constraints on the database’s accounting software SIGER. The prototype will be developed using the same developing tool that company uses on their ERP, based on their structured software methodology.

Key words: Dates, costs, production, function points.

LISTA DE FIGURAS

Figura 1 – Fluxo do processo de contagem de pontos por função. _____	23
Figura 2 – Projetos pesquisados _____	24
Figura 3 – Fatores de sucesso nos projetos _____	24
Figura 4 – Fatores que ocasionam atrasos e custos além do previsto nos projetos _____	25
Figura 5 – Fatores que ocasionam cancelamentos nos projetos _____	25
Figura 6 – Fronteira de aplicação _____	30
Figura 7 – Processo de contagem de funções do tipo dados _____	33
Figura 8 – Processo de contagem de funções do tipo transação _____	42
Figura 9 – Processos da Rech Informática Ltda _____	66

LISTA DE TABELAS

Tabela 1 - Tabela de complexidade funcional de ALI e AIE _____	35
Tabela 2 - Tabela de contribuição do ponto por função das funções do tipo dado. _____	37
Tabela 3 – Tabela de complexidade para entradas externas _____	43
Tabela 4 – Tabela de complexidade para saídas externas e consultas externas _____	44
Tabela 5 – Tabela de contribuição dos pontos por função das funções de transação _____	44
Tabela 6 – Tabela de níveis de influência das características gerais de sistema _____	46
Tabela 7 – Tabela de níveis de influência para comunicação de dados _____	48
Tabela 8 – Tabela de níveis de influência para processamento distribuído _____	48
Tabela 9 – Tabela de níveis de influência para performance. _____	49
Tabela 10 – Tabela de níveis de influência para configuração altamente utilizada _____	50
Tabela 11 – Tabela de níveis de influência para volume de transações _____	50
Tabela 12 – Tabela de níveis de influência para entrada de dados <i>on-line</i> _____	51
Tabela 13 – Tabela de níveis de influência para eficiência do usuário final _____	53
Tabela 14 – Tabela de níveis de influência para atualização <i>on-line</i> _____	53
Tabela 15 – Tabela de níveis de influência para complexidade de processamento _____	55
Tabela 16 – Tabela de níveis de influência para reutilização _____	55
Tabela 17 – Tabela de níveis de influência para facilidade de instalação _____	56
Tabela 18 – Tabela de níveis de influência para facilidade de operação _____	57
Tabela 19 – Tabela de níveis de influência para múltiplos locais _____	57
Tabela 20 – Tabela de níveis de influência para facilidade de mudanças _____	59

LISTA DE ABREVIATURAS E SIGLAS

APF	Análise de Pontos por Função
AIE	Arquivo de Interface Externa
ALI	Arquivo Lógico Interno
BFPUG	<i>Brazilian Function Point Users Group</i>
CGS	Características Gerais de Sistema
COCOMO	<i>Constructive cost model</i>
COCOMO	<i>Constructive cost model II</i>
II	
ERP	<i>Enterprise Resource Planning</i>
FPA	<i>Function Point Analysis</i>
FSM	<i>Functional Size Measurement</i>
IEC	<i>International Engineering Consortium</i>
IFPUG	<i>International Function Point Users Group</i>
ISBSG	<i>International Software Benchmarking Standards Group</i>
JTC1	<i>Joint Technical Committee One</i>
ISO	<i>International Organization for Standardization</i>
LOC	<i>Lines of code</i>
PMI	<i>Project Management Institute</i>
SC7	<i>Sub-Committee Seven</i>
TI	Tecnologia da Informação
VAF	Valor do Fator de Ajuste
WG12	<i>Working Group 12</i>

SUMÁRIO

INTRODUÇÃO	11
1 A NECESSIDADE DE MEDIR SOFTWARE	14
1.1 Por que medir software?	14
1.2 Quais medidas utilizar?	15
1.3 O que é a metodologia de análise de pontos por função?	17
2 INSTITUIÇÕES VOLTADAS A ANÁLISE DE PONTOS POR FUNÇÃO	19
2.1 Breve histórico	19
2.2 IFPUG	20
2.3 BFPUG	20
2.4 ISBSG	20
2.5 Padronização ISO para medição	21
3 PROCESSO DA ANÁLISE DE PONTOS POR FUNÇÃO	23
3.1 Levantamento de requisitos	23
3.1.1 Conceito de usuário	26
3.1.2 Conceito de requisito	26
3.2 Análise de pontos por função	26
3.3 Determinar o que deve ser contado	28
3.4 Fronteira da aplicação	29
3.5 Escopo da contagem	31
3.6 Funções do tipo dado e transação	31
3.7 Pontos de função não-ajustados	32
3.8 Fator de ajuste e pontos de função ajustados	32
4 FUNÇÃO DO TIPO DADO	33
4.1 Arquivo lógico interno	33
4.2 Arquivo de interface externa	34
4.3 Contagem de ALI e AIE	35
4.3.1 Tipo de dado	35
4.3.2 Tipo de registro	36
4.3.3 Contribuição do ponto por função não-ajustados das funções do tipo dado	37
4.3.4 Considerações finais sobre contagem de ALI e AIE	37
5 FUNÇÃO DO TIPO TRANSAÇÃO	39
5.1 Entrada externa	39
5.2 Saída externa	39
5.3 Consulta externa	40
5.4 Contribuição do ponto por função não-ajustados das funções do tipo transação	41

6 FATOR DE AJUSTE	45
6.1 Características gerais de sistema (CGS)	45
6.1.1 Comunicação de dados	48
6.1.2 Processamento distribuído	48
6.1.3 Performance	49
6.1.4 Configuração altamente utilizada	50
6.1.5 Volume de transações	50
6.1.6 Entrada de dados <i>on-line</i>	51
6.1.7 Eficiência do usuário final	52
6.1.8 Atualização <i>on-line</i>	53
6.1.9 Complexidade de processamento	54
6.1.10 Reutilização	55
6.1.11 Facilidade de instalação	56
6.1.12 Facilidade de operação	57
6.1.13 Múltiplos locais	57
6.1.14 Facilidade de mudanças	58
6.2 Considerações	59
7 CÁLCULO DOS PONTOS DE FUNÇÃO AJUSTADOS	61
7.1 Projeto de desenvolvimento	61
7.2 Projeto de melhoria	62
7.3 Aplicação	63
8 ESTUDO DE CASO	65
8.1 Apresentando a empresa estudada	65
8.2 Processos da Rech Informática	66
8.2.1 Processo principal – Relacionamento com mercado	66
8.2.2 Processo de apoio – Pesquisa e desenvolvimento	66
8.2.3 Processo de apoio – Controladoria e finanças	67
8.3 Análise de programas sobre APF	67
8.4 Projeto utilizado como <i>case</i>	69
CONCLUSÃO	71
REFERÊNCIAS BIBLIOGRÁFICAS	73

INTRODUÇÃO

Com o crescimento econômico e aumento da concorrência, seja no segmento de prestação de serviços, indústria ou comércio, podemos observar que as organizações buscam diferenciais nos seus processos que possam fazer com que o seu negócio esteja sempre à frente no mercado. Entre estes diferenciais podemos destacar os custos do processo produtivo, a incansável busca por custos mais baixos, sem deixar de lado a qualidade do produto produzido, em meio a um mercado cada vez mais exigente.

No ramo de prestação de serviços, mais especificamente na área de tecnologia da informação, a necessidade constante de aumento da qualidade e melhoria dos processos de desenvolvimento de software tem sido crucial para sobrevivência no mercado das empresas desenvolvedoras de programas, conhecidas como “*software houses*”. Na empresa onde trabalho, por exemplo, estando os processos e papéis muito bem definidos podemos conseguir melhorar a produtividade e por conseqüência uma redução nos custos de produção.

A qualidade do software (Pressman, 2002) depende de um considerável nível de processo de desenvolvimento bem estruturado. Atualmente podemos observar empresas que não possuem processos mapeados e bem definidos, não adotam metodologias que auxiliem os gerentes de projeto e diretores no processo de planejamento dos seus projetos, tanto na parte de investimento quanto prazos. Em virtude disto, acabam assumindo riscos em projetos estipulando prazos e custos baseando-se no “*feeling*”, no sentimento dos seus gerentes, ou em experiências de anteriores. Os resultados nem sempre são satisfatórios, projetos com prazos estourados e com no alto custo de desenvolvimento (Pressman, 2002). Esta situação pode complicar se observarmos o dia-a-dia das empresas de desenvolvimento de programas, onde percebemos uma concorrência desenfreada de projetos a serem analisados e desenvolvidos, às vezes em um curtíssimo espaço de tempo.

Em virtude disto, podemos ver algumas empresas adotando modelos de qualidade de software (Koscianski, 2006), visando melhorar os seus processos. Visando a otimização de processos e por consequência a produtividade, podemos destacar a necessidade de metodologias para mensuração de prazos e custos de novos projetos, bem como a manutenção dos programas já existentes. A APF estudada neste trabalho de conclusão tem sido utilizada por organizações para atender esta necessidade, considerada como uma prática chave para empresas que pretendem atingir o nível 2 de maturidade do CMM (Côrtes, 1998). O CMM é um modelo de avaliação da maturidade dos processos, produtos e serviços das empresas. Foi desenvolvido pela Universidade de Carnegie Mellon a pedido do departamento de defesa dos Estados Unidos, sendo difundido em organizações ao redor do mundo. Conhecida internacionalmente como FPA, “*Function Point Analysis*”, esta metodologia foi criada por Allan Albrecht na empresa IBM no final da década de 70. A FPA tem como um dos objetivos medir a taxa de produtividade e o esforço no desenvolvimento de software com base na perspectiva do cliente, ou seja, o usuário final. A partir dos requisitos (Pressman, 2002) é feito o levantamento do que deve ser feito, para então chegar ao número de pontos por função a serem implementados. Em posse desta informação, são aplicadas fórmulas matemáticas para chegar ao resultado próximo do esforço que será necessário para produção do projeto, podendo ainda estimar o prazo e o custo com base nos recursos disponíveis da organização. Uma característica interessante desta metodologia é que ela independe de qualquer ferramenta ou técnica de desenvolvimento, por que a contagem é feita pela perspectiva do usuário. Podemos utilizar esta metodologia como:

- Um método para dimensionar as aplicações;
- Um método para estimar esforços, prazos e custos;
- Um método para quantificar produtividade e qualidade;
- Um fator de normalização para comparar software.

O uso da APF na mensuração de projetos ajuda também no momento de negociar com o cliente o valor a ser cobrado e o prazo para o desenvolvimento. Como a APF é baseada na perspectiva do usuário, a negociação se torna menos complicada para o usuário entender a complexidade e o esforço envolvido. Outro ponto na negociação com o cliente a ser observado é o emprego da palavra “custo”. Esta palavra pode criar uma restrição durante a negociação, é recomendável empregar então o termo “investimento”, já que os projetos

desenvolvidos normalmente têm como objetivo aperfeiçoar os processos do cliente. Podemos ressaltar também que o esforço empregado no desenho do projeto poderá ser ineficaz, se os requisitos não refletirem exatamente os anseios do cliente. Quanto ao prazo, estando o projeto bem desenhado e os pontos por função devidamente desdobrados, somando-se os tempos de produção de cada função, obteremos o prazo total do projeto. Com isso podemos acordar com o cliente a data de entrega do projeto e cronograma de treinamento caso necessário.

Tem-se percebido nas organizações que a taxa de produtividade no desenvolvimento é uma necessidade pontual apontada por diretores e gerentes. Este trabalho de conclusão tem como objetivo estudar os processos da empresa Rech Informática Ltda, empresa fabricante de software desde 1990. Estabelecida em Novo Hamburgo/RS, a Rech Informática tem como processo principal a prestação de serviços na área de tecnologia da informação, utilizando como ferramenta de apoio o SIGER, ERP desenvolvido pela empresa. Com crescimento constante, a Rech Informática tem uma considerável demanda de implementações, e ainda tem necessidade de criar novos recursos para manter e conquistar novos clientes. Partindo deste pressuposto, a direção da empresa e colaboradores da área de planejamento de projetos aponta a complexidade de estimar prazos, custos e gerenciar a concorrência da demanda.

A proposta deste trabalho é analisar software existente e verificar a possibilidade de desenvolver um protótipo de software que venha a auxiliar no gerenciamento, utilizando a técnica da APF, por indicação da direção da empresa a qual trabalho. Para construção do protótipo proposto, será utilizado como estudo de caso um dos projetos em desenvolvimento, que compreende novas funcionalidades e quebra de algumas restrições da base de dados atual do módulo de Contabilidade SIGER. O protótipo proposto será desenvolvido pela mesma ferramenta de desenvolvimento que a empresa utiliza no seu ERP, baseado na metodologia estruturada de seus programas fonte.

1 A NECESSIDADE DE MEDIR SOFTWARE

As empresas fabricantes de software cada vez mais precisam aperfeiçoar seus processos para ganhar competitividade no mercado, sendo que o diferencial na hora de ganhar do concorrente um contrato de prestação de serviços pode estar em pequenos detalhes, como estimar prazos e custos de projetos, o qual pode tornar-se um problema se a empresa não tiver nenhuma técnica para estas questões (HAZAN, 2003).

1.1 Por que medir software?

Além de vencer a concorrência, resultado de conjunto de esforços da empresa, devemos observar que antes disso a empresa precisa que todos seus colaboradores tenham ciência dos processos da empresa e seus papéis dentro do processo. Segundo o *Project Management Institute* - PMI, um projeto é um empreendimento temporário posto em execução para criar um único produto ou serviço. Dentro dos projetos encontramos 3 passos comuns que são o planejamento, execução e controle. O planejamento visa traçar o os objetivos do projeto e o caminho a ser seguido, a execução trata do gerenciamento de pessoas e recursos para dar vida ao projeto. Por fim o controle, garantindo que o planejamento foi seguido, monitorando e medindo, identificando as variações durante a execução e quando necessário, tomar ações corretivas para que o planejamento seja seguido. No planejamento, segundo Vasquez, Simões e Albert (2005, p.19):

[...] em sua fase inicial que compreende o levantamento de requisitos, ainda não há o conhecimento completo das características do produto que permita a apuração de sua futura dimensão. Nesse caso é necessário estimar.

A análise de pontos por função além de permitir medir o tamanho da aplicação pelo ponto de vista do usuário, pode ser utilizada para estimar seu tamanho em qualquer fase do ciclo de vida da aplicação desde sua concepção até futuras reestruturações e melhorias.

No controle, segundo Vasquez, Simões e Albert (2005, p.21):

[...] é de suma importância que a definição de meios para a comparação do progresso real com o planejado seja parte do planejamento do projeto. O controle é uma das principais atividades envolvidas na gerência de projetos. Trazendo estes conceitos para o contexto de um projeto de desenvolvimento de sistemas, é possível ter algumas idéias interessantes na busca da resposta de por que medir software. Afinal, não se consegue controlar o que não se consegue medir.

Segundo HAZAN (2007) as razões para se medir software são:

- Indicar a qualidade do produto ao usuário;
- Avaliar a produtividade do processo;
- Melhorar a gerência de projetos e relacionamento com clientes;
- Formar uma *baseline* para estimativas;
- Gerenciar contratos de software.

A partir desta afirmativa dos autores, torna-se claro a necessidade de medir software, tanto para mapeamento do tamanho da aplicação como o controle de sua execução. A APF deve ser aplicada no início do projeto e aplicada no decorrer da sua execução até a contagem final, por que podem ocorrer problemas durante a execução como: troca de pessoal da equipe ou até alguns itens de garantia da qualidade propositalmente esquecidos para se atingir marcos de projetos. Nestes casos é que a equipe de controle de projetos entra em ação, fazendo com que tudo que foi previsto no projeto será cumprido, uma nova medição do projeto deve ser feita e com base nos resultados, tomar medidas, avaliar prazos e custos. Além de servir como metodologia para mensurar e controlar projetos, a APF pode ser utilizada para cálculo da remuneração e avaliação da produtividade da equipe de desenvolvimento. Encontramos a APF também em licitações públicas na área de TI, onde os projetos das empresas candidatas são avaliados através da metodologia (HAZAN, 2003).

1.2 Quais medidas utilizar?

A definição da palavra medida é a quantificação de uma determinada característica. Segundo um artigo intitulado “*Software modeling and measurement: the goal/question/metric paradigm*”, publicado pelo departamento de ciência da computação da Universidade de Maryland, Victor Basili escreveu:

“Para cada um dos objetivos que deseja acompanhar é possível estabelecer um conjunto de perguntas que verifique o seu cumprimento; para muitas dessas perguntas é possível identificar uma métrica que possa quantificar a resposta”.

No caso de aplicações, na APF não pode ser feita medição apenas das características do produto final, mas devemos identificar as características relevantes durante o processo todo, desde a concepção passando pela execução (Vasquez, Simões e Albert, 2005). Podemos considerar alguns aspectos comuns em projetos, como: recursos, custos, tamanho do produto, qualidade do produto, cronograma e progresso, porém estes aspectos são complexos de poder medir e acompanhar. Uma alternativa é a criação de categorias que agrupem conjuntos de métricas de mesma característica, assim diluindo a sua complexidade permitindo assim um foco em itens mais específicos do projeto (Vasquez, Simões e Albert, 2005). Por exemplo, a equipe de desenvolvimento necessária para execução do projeto pode ser uma categoria, onde o esforço necessário, quantidade de pessoas envolvidas e o nível de capacitação da equipe e um percentual de tolerância de rotatividade de pessoas da equipe podem ser os itens desta categoria. Neste último, a rotatividade da equipe não quer dizer necessariamente demissões, mas sim a concorrência de projetos, onde pode ocorrer de um ou mais membros da equipe trocar de projeto, sendo substituído ou não por outros colegas. Esta categoria pode ainda ser utilizada como um quesito para avaliação da produtividade da empresa. O tamanho do produto deve ser outro aspecto relevante ao medir o projeto, devendo ser monitorado durante a execução do projeto, isto por que podemos observar projetos em que o usuário inclui, altera ou retira requisitos, conseqüentemente mudando o esforço e o prazo para entrega do projeto. Ao considerar que o tamanho é uma das características a ser medida, devemos observar a unidade para medição. A unidade mais lógica seria o número de linhas de código, inclusive estudantes da área utilizam esta unidade ao comparar programas desenvolvidos. Segundo Vasquez, Simões e Albert (2005), o uso de *lines of code* - LOC para mensuração pode ser perigoso caso não for levado em conta os seguintes detalhes:

- Não podem ser incluídos na contagem da quantidade de linhas de código os comentários, linhas em branco ou comandos nulos;
- A inclusão na contagem de diretrizes de compilação;
- A contagem de múltiplos comandos ou declarações em uma única linha como várias linhas, uma para cada comando ou declaração;

- A contagem de uma única linha nos casos em que um único comando ou declaração é expresso em múltiplas linhas;
- A inclusão na contagem de delimitadores de blocos de comandos nos casos em que de fato haja mais de um comando;
- A desconsideração de delimitadores de blocos de comandos nos casos em que sua utilização seja opcional.

A partir destes itens salientados pelos autores, podemos concluir que não há um padrão para contagem de linhas de código, por que a codificação muda de linguagem para linguagem de programação, bem como o emprego de comandos e a lógica utilizada para solução do problema varia de programador para programador. Outra dificuldade encontrada nesta técnica está em argumentar com o cliente esta medida, afinal o cliente não entende o que quer dizer linhas de código.

Em 1981, na tentativa de criar um modelo para medir, Dr. Barry Boehm lançou o COCOMO, *Constructive cost model*, baseando-se na técnica LOC para estimar esforço, prazo e custo. Este modelo, porém pode ser complicado de utilizar por que antes de realmente codificar o programa fonte não se sabe ao certo quantas linhas de código são necessárias para atender determinada funcionalidade, por exemplo. Hoje em dia esta questão fica mais complexa se observarmos as ferramentas de programação atuais onde fica difícil a abstração. A partir desta dificuldade, Allan Albrecht em 1975 criou os conceitos que seriam os subsídios para a criação da análise de pontos por função.

1.3 O que é a metodologia de análise de pontos por função?

A APF surgiu para quebrar o paradigma de mensuração do tamanho do produto pela quantidade de linhas de código, bem como a dependência de tecnologia a ser empregada. O foco desta metodologia criada por Allan Albrecht em 1975, na empresa IBM, compreende na mensuração da aplicação pelo que ela faz, ou seja, pela perspectiva do usuário e não como a aplicação foi construída. Utilizando como unidade de medida o ponto por função, esta metodologia independe de tecnologia empregada. A contagem dos pontos por função é baseada na avaliação dos requisitos do usuário, sendo que ela representa exclusivamente o tamanho funcional da aplicação, servindo de base para que com outras variáveis possa ser calculado o esforço, prazo e o custo.

Segundo HAZAN (2007), APF é:

[...] A Análise de Pontos por Função (APF) é um método padronizado para a medição de projetos de desenvolvimento de software, visando estabelecer uma medida de tamanho, em Pontos de Função (PF), considerando a funcionalidade implementada, sob o ponto de vista do usuário. [...]

Podemos concluir que a necessidade de medir software é de vital importância para medir projetos. Fazendo uma analogia, em um plano de vôo são necessários instrumentos e técnicas para se chegar a algum lugar, os pilotos simplesmente não pilotam uma aeronave sem destino algum, assim podemos considerar em outras áreas também, como TI, não podemos sair escrevendo linhas de código sem ter um norte. A APF é uma boa alternativa para se ter uma técnica para mensuração, a seguir serão apresentadas entidades responsáveis pela difusão da metodologia no mundo e no Brasil.

2 INSTITUIÇÕES VOLTADAS A ANÁLISE DE PONTOS POR FUNÇÃO

Neste capítulo, será abordado um breve histórico da APF e algumas entidades que são responsáveis para difusão da metodologia.

2.1 Breve histórico

A APF surgiu em 1975, por Allan Albrecht funcionário da empresa IBM. Albrecht na época tinha a incumbência de medir os projetos da empresa, e baseando-se na contagem de linhas de código, percebeu que era complexo medir a partir desta técnica, por motivos já explanados anteriormente. Além de a técnica ser deficiente, a empresa possuía vários projetos desenvolvidos em diferentes linguagens de programação, dificultando a análise da produtividade da empresa. A solução encontrada foi quebrar este paradigma, passando a medir pela perspectiva do que a aplicação deveria fazer, e não como seria construída, tornando assim independente de linguagem de programação. Após a publicação desta técnica e dos sucessivos trabalhos feitos Capers Jones, houve uma crescente legião de adeptos da técnica, culminando em 1986 a fundação do *International Function Point Users Group* - IFPUG. Com o passar dos anos começaram a surgir variações da teoria de Albrecht, fazendo com que em 1990 o IFPUG lançasse a primeira versão do manual de práticas de contagem de pontos por função objetivando a padronização da técnica. Esta metodologia do IFPUG é considerada como o padrão mais difundido no mercado de pontos por função, estando atualmente na versão 4.2.1, embora ainda tenham outras técnicas como o Mark II e COSMIC-FFP (Vasquez, Simões e Albert, 2005).

2.2 IFPUG

O IFPUG fundado em 1986 após uma crescente legião de adeptos da técnica de Albrecht. O IFPUG é uma entidade sem fins lucrativos, sendo composta por usuários e empresas de diversos países. A entidade visa o correto uso da metodologia de análise de pontos por função, sendo promovido pelo trabalho voluntário dos seus membros. O IFPUG promove várias iniciativas como: conferência anual, seminários e *workshops* educacionais, certificação profissional, comitês e grupos de trabalho visando à manutenção do manual de práticas entre outras atividades a fim da entidade. O manual de práticas do IFPUG é gratuito para os seus membros, porém para os não membros somente é possível adquirindo o manual (Vasquez, Simões e Albert, 2005).

2.3 BFPUG

Brazilian Function Point Users Group – BFPUG fundado em 1988 é o *chapter* do IFPUG no Brasil. O termo *chapter* é utilizado para as associações locais do IFPUG para difundir a metodologia. O BFPUG conta com centenas de associados, de estudantes passando por desenvolvedores chegando até os gerentes de sistemas. A entidade promove em seu site diversos eventos, inclusive um fórum de discussão gratuito. A APF começou a prosperar no Brasil somente depois do apoio da empresa Unisys, no início da década de 90. A partir disso, foram feitos vários encontros sobre o assunto, inclusive contando com a presença de palestrantes internacionais. O uso da metodologia se consolidou depois do interesse de entidades envolvidas no desenvolvimento de software por modelos de qualidade e maturidade (ISO e CMM), aumentando assim o interesse das pessoas no assunto (Vasquez, Simões e Albert, 2005).

2.4 ISBSG

O *International Software Benchmarking Standards Group* – ISBSG é uma entidade sem fins lucrativos, mantido por diversas organizações de métricas de software do mundo. A missão do ISBSG é manter um repositório público de métricas de projetos, auxiliando os usuários de pontos por função nas estimativas de projetos, produtividade, análise de riscos e *benchmarking*. Seu repositório de dados contém mais de 3.000 projetos de software de vários países, constituindo o seu *benchmarking*. Regularmente o ISBSG publica uma análise estatística detalhada do seu repositório, chamada de “*The Software Metrics Compendium*”. O

seu repositório é constantemente atualizado com novos estudos de caso, sendo que qualquer organização pode contribuir. O ISBSG coleta os dados e mantém a confidencialidade dos dados fornecidos e o anonimato da organização (Vasquez, Simões e Albert, 2005).

2.5 Padronização ISO para medição

No final de 1992 havia diversos métodos de mensuração do tamanho funcional *Functional Size Measurement* – FSM, surgidas de diferentes entendimentos da metodologia originalmente criada por Albrecht. Com o objetivo de manter uma padronização da metodologia, usuários da Holanda, Inglaterra, Austrália e Estados Unidos formaram o *Working Group 12* - WG12, que subordinado ao *Sub-Committee Seven* - SC7 do *Joint Technical Committee One* - JTC1 estabelecido pela *International Organization for Standardization* - ISO em conjunto com o *International Engineering Consortium* - IEC. O resultado do trabalho conjunto foi a criação da norma 14.143 o qual é composta pelas seguintes partes:

- 14.143-1: Definição de conceitos;
- 14.143-2: Avaliação da conformidade de métodos de medição de software com relação ao padrão ISSO/IEC 14.143-1;
- 14.143-3: Verificação de um método de medição de tamanho funcional;
- 14.143-4: Modelo de referência para medição de tamanho funcional;
- 14.143-5: Determinação de domínios funcionais para uso com medição de tamanho funcional.

A metodologia de análise de pontos por função se submeteu a certificação ISO norma 14.143 e no final de 2002, foi aprovada recebendo a denominação ISO/IEC 20296:2002. Entretanto o padrão da APF aprovado pela ISO vai somente até o estágio da determinação dos pontos de função não ajustados, sendo que o restante do processo conforme o manual do IFPUG contém itens que são considerados não aderentes ao padrão de medição funcional da ISO (Vasquez, Simões e Albert, 2005).

Podemos concluir que a padronização ISO foi de fundamental importância para que a APF fosse difundida no mundo inteiro, aliado ao trabalho dos membros do IFPUG e a

participação de empresas de prestígio como a Unisys aqui no Brasil. No próximo capítulo serão descritas as etapas do processo da APF.

3 PROCESSO DA ANÁLISE DE PONTOS POR FUNÇÃO

O processo da APF segue em linhas gerais o fluxo da figura abaixo.

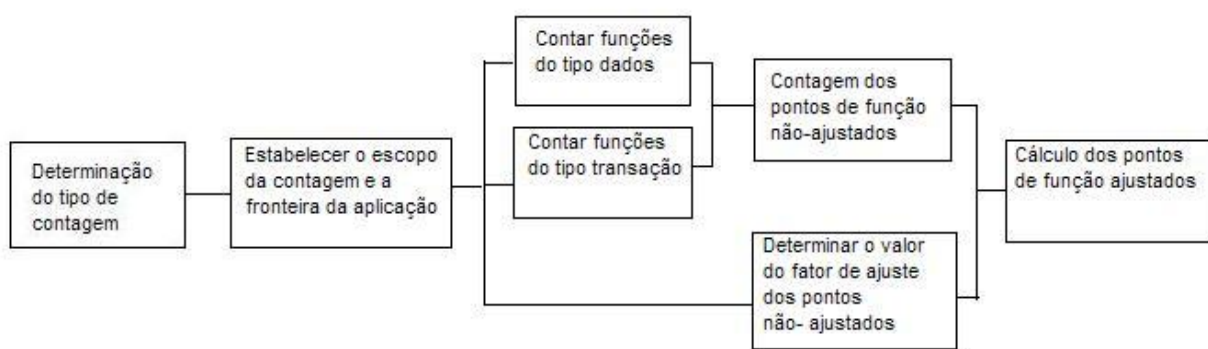


Figura 1 – Fluxo do processo de contagem de pontos por função.

Fonte: Vasquez, Simões e Albert (2005)

Este processo pode ser aplicado tanto em projetos já concluídos como nos projetos ainda em fase de levantamento de requisitos.

3.1 Levantamento de requisitos

O levantamento de requisitos tem papel fundamental para a APF, afinal o objetivo é medir a partir da perspectiva do usuário, logo é ele quem define os requisitos que o projeto deve ter, devendo ser medidas e contadas. Para fazer um levantamento de requisitos bem descrito é aconselhável fazer reuniões, perguntar, entender o processo do cliente (Vasquez, Simões e Albert, 2005). Os usuários nem sempre expressam diretamente o que realmente desejam, ficando nas “entrelinhas” detalhes que podem passar despercebidos durante a execução do projeto e vir à tona somente na entrega ao cliente, gerando retrabalhos.

Em 1994 foi feita uma pesquisa nos Estados Unidos publicada pelo *The Standish Group* buscando identificar onde ocorrem falhas em projetos, os fatores causadores e o que poderia reduzir as falhas. Os resultados da pesquisa foram interessantes, a figura abaixo demonstra que aconteceu com os projetos pesquisados:

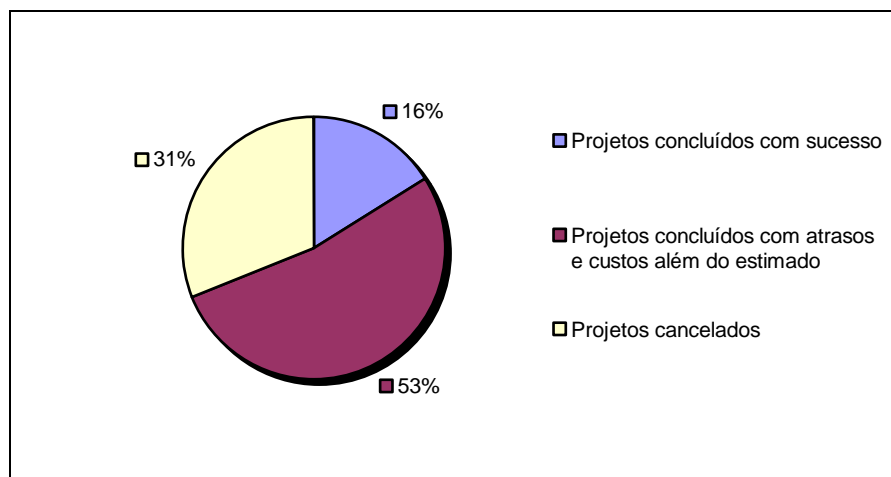


Figura 2 – Projetos pesquisados
Fonte: CHAOS Report – The Sandish Group

Dos fatores analisados como essenciais para o sucesso dos projetos, podemos observar a figura a seguir:

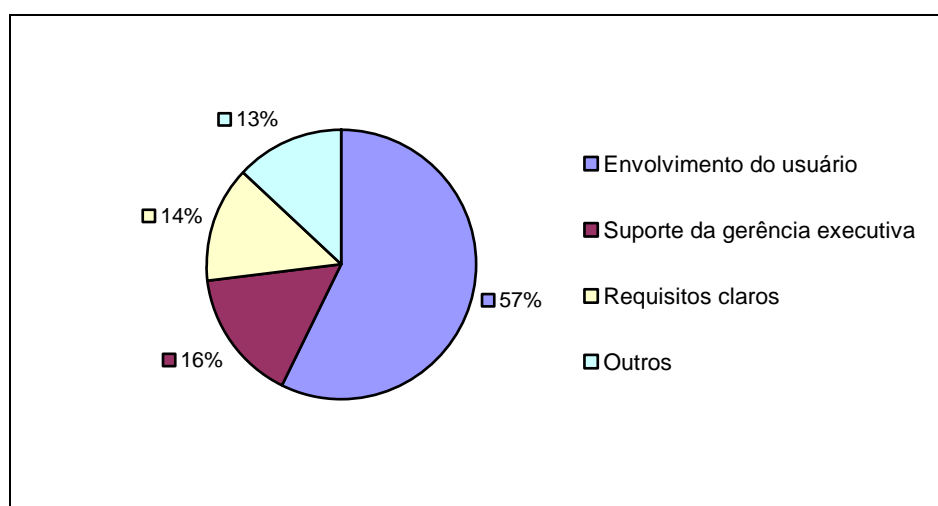


Figura 3 – Fatores de sucesso nos projetos
Fonte: CHAOS Report – The Sandish Group

Dos projetos pesquisados que tiveram atrasos e/ou custos além do previsto, os fatores apontados como determinantes para este fracasso são demonstrados na figura a seguir:

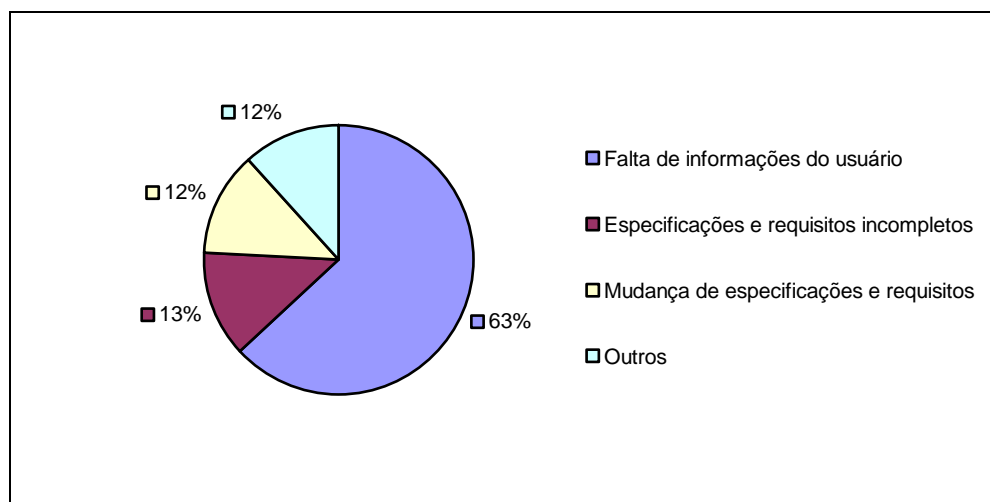


Figura 4 – Fatores que ocasionam atrasos e custos além do previsto nos projetos
Fonte: CHAOS Report – The Sandish Group

Dos projetos pesquisados que foram cancelados, os fatores apontados como determinantes para o cancelamento são demonstrados na figura a seguir:

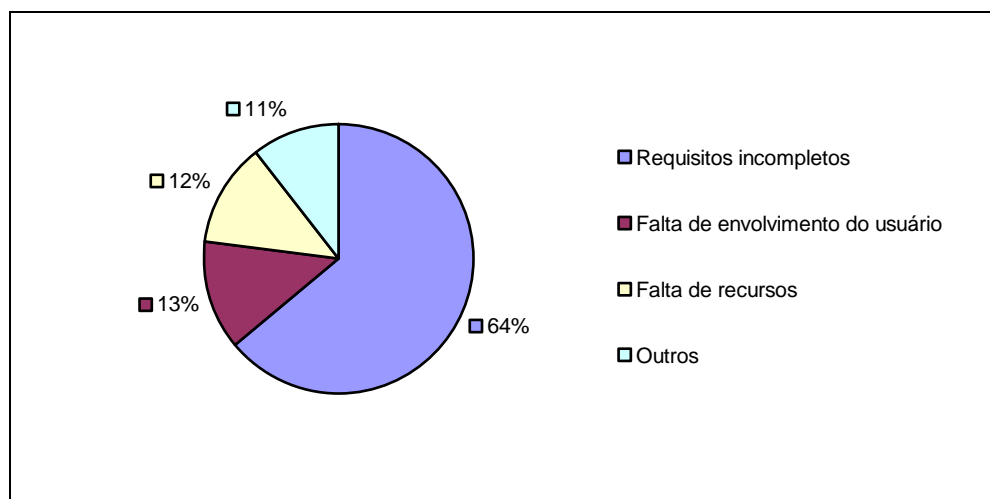


Figura 5 – Fatores que ocasionam cancelamentos nos projetos
Fonte: CHAOS Report – The Sandish Group

Analisando os gráficos apresentados, podemos concluir que a interação da equipe de desenvolvimento com o usuário é importante para o sucesso dos projetos, devendo sempre estreitar esta distância o máximo possível. Outra conclusão que podemos chegar é a

importância do levantamento de requisitos bem detalhado, preferencialmente feito junto ao cliente aumentando a chance de sucesso no projeto.

Antes de começar a explorar o processo da APF, é preciso ver os conceitos de usuário e requisitos, conforme segue abaixo.

3.1.1 Conceito de usuário

Inicialmente devemos ressaltar que a palavra usuário na APF tem um conceito mais abrangente, não ficando apenas associado à pessoa física que utiliza software. Usuário é qualquer pessoa ou coisa que interaja (envia ou recebe dados) com a aplicação ou especifique seus requisitos. Exemplificando, um usuário pode tanto ser um operador (pessoa física), como uma aplicação ou um hardware que interaja com a aplicação a ser medida. Desta forma a APF pode ser aplicada também a programas em que não tenham interface com o usuário final (pessoa física). (Vasquez, Simões e Albert, 2005). Segundo HAZAN (2003): “Usuário pode ser qualquer pessoa, dispositivo ou sistema que se comunica ou interage com a aplicação”.

3.1.2 Conceito de requisito

A palavra requisito na área de desenvolvimento de sistemas está associada a uma característica, comportamento, capacidade ou condição que conforme as necessidades do usuário devem ser atendidas na aplicação a ser construída. Os requisitos podem estar declarados em aspectos funcionais e não-funcionais. Na APF os requisitos funcionais são considerados como a “matéria-prima” para o cálculo dos pontos por função não-ajustados. Já os requisitos não-funcionais podem servir como base para determinar o fator de ajuste, sendo utilizado posteriormente no cálculo de pontos por função ajustados, descritos logo adiante. (Vasquez, Simões e Albert, 2005)

3.2 Análise de pontos por função

A APF é uma metodologia para medir o tamanho funcional de um projeto de desenvolvimento, projeto de melhoria ou uma aplicação. No decorrer deste capítulo serão abordados os objetivos da APF conforme o IFPUG bem como os benefícios obtidos, e em seguida o processo de contagem de pontos por função. O processo de contagem se inicia ao estabelecer a fronteira da aplicação, por consequência obtemos o escopo da contagem. Ao

delimitarmos o escopo de contagem, torna-se necessário identificar as funções de tipo dado e transação. Identificadas funções, partimos para a contagem dos pontos por função, chegando aos pontos de função não-ajustados. Até este ponto estamos dentro do padrão ISO, o próximo passo é ajustar o cálculo utilizando o fator de ajuste, chegando finalmente ao total de pontos por função.

Ainda podemos destacar que o processo de contagem deve contar com o trabalho conjunto entre usuário e desenvolvedor para que o levantamento de requisitos seja o mais completo possível, sem ambigüidades, possíveis de programar, garantindo uma maior precisão para a contagem de pontos por função. Não seguindo desta forma, o próprio processo de contagem dos pontos por função mostrará esta imaturidade dos requisitos.

Conforme o IFPUG, os objetivos básicos da APF são:

- Medir a funcionalidade que o usuário solicita (previsto) e recebe (realizado);
- Medir o desenvolvimento e manutenção de software de forma independente da tecnologia utilizada para sua concepção.

Devemos ainda salientar que o processo de contagem deve ser simples suficiente para aplicá-lo sem muito esforço ao medir a aplicação, apresentando no final uma medida consistente.

A organização que adota a APF como metodologia, além de passar a ter um padrão para medir software, produtividade, traz vários benefícios ao seu processo de desenvolvimento. Um dos benefícios, por exemplo, estão na possibilidade de fazer uma avaliação de um produto quanto suas funcionalidades, quais se encaixam ao seu processo, e tendo informações como índice de produtividade da equipe e recursos, pode ser feita uma análise chamada “*make or buy*”, ou seja, decidir entre fazer seu próprio produto ou adquiri-lo. Na parte de projetos podemos dizer que a APF pode servir de suporte na análise de produtividade e qualidade, em conjunto com métricas de esforço, defeitos e o custo. Podemos concluir ainda outros benefícios como:

- Apoio ao gerenciamento de escopo de projetos, um dos desafios do gerente de projeto;

- Complementa o gerenciamento de requisitos, ao analisar a qualidade do levantamento de requisitos;
- Um meio de estimar custos e recursos nos processos e projetos;
- Uma forma para fundamentar a negociação de contratos, estabelecendo uma unidade tangível para o cliente, o ponto por função, estabelecendo um preço por esta unidade;
- Uma forma de normalização para comparação de software, ou ainda comparação de produtividade na utilização de tecnologias diferentes.

Lembramos que a APF não é considerada um fim e sim um propósito, um meio de auxílio, um suporte a diferentes áreas do processo de desenvolvimento de software.

3.3 Determinar o que deve ser contado

Conforme o IFPUG, uma aplicação é: “um conjunto coeso de dados e procedimentos automatizados que suportam um objetivo de negócio, podendo consistir de um ou mais componentes, módulos ou sistemas. Frequentemente, o termo aplicação é utilizado como sinônimo para sistema, sistema aplicativo ou sistema de aplicação.”. Reforçamos que a aplicação deve ser vista pela perspectiva do usuário e não pela visão técnica como arquitetura ou plataforma.

Iniciando o processo de contagem de pontos por função, a primeira questão que devemos responder é: o que deve ser contado? Determinar o tipo de contagem é o primeiro passo a ser dado neste processo, considerando que a contagem pode ser aplicada tanto a projetos quanto aplicações. Basicamente o tipo de contagem divide-se em três tipos: projeto de desenvolvimento, projeto de melhoria e aplicação.

No projeto de desenvolvimento o número de pontos por função mede a funcionalidade do projeto a ser entregue. Isto quer dizer que a contagem não compreende apenas o software em si, mas outras ações a serem feitas durante a sua execução também devem ser contadas, como uma migração de base de dados, por exemplo. Durante a concepção e execução de um projeto, toda medida realizada é apenas uma estimativa do que está sendo feito. É bastante comum durante o desenvolvimento de projetos a equipe encontrar

funcionalidades não medidas no planejamento do projeto. Conforme o projeto vai sendo produzido e feitas novas medidas, chegamos mais perto da sua medida final, permitindo assim uma avaliação final pela equipe do projeto.

O projeto de melhoria, diferente do projeto de desenvolvimento, parte de um software já produzido. A contagem de pontos por função em um projeto de melhoria compreende medir as funções novas, alteradas e eliminadas. Caso houver migração de base de dados, as funções para a migração também devem ser medidas.

Em uma aplicação o número de pontos por função mede a funcionalidade por uma aplicação instalada. Conhecida por pontos de função instalados ou *baseline*, esse número fornece uma medida atual da funcionalidade pelo usuário de uma aplicação. Ele começa ao final da contagem do projeto de desenvolvimento, sendo atualizado no término de todo projeto de melhoria o qual altera a funcionalidade da aplicação.

3.4 Fronteira da aplicação

Estabelecer a fronteira da aplicação é delimitar onde começa e onde termina a medição dos pontos por função. Esta etapa é considerada uma das mais importantes do processo de contagem, por que se a fronteira de aplicação não estiver bem definida há risco do restante da contagem não refletir o real tamanho da aplicação, comprometendo todo o trabalho. Podemos fazer uma analogia com a construção de uma casa, no qual durante a concepção do projeto de construção precisamos saber a medida do terreno, do contrário podemos ocupar o terreno vizinho, e depois de construído o prejuízo pode ser grande. Segundo o IFPUG, as regras para determinação da fronteira da aplicação são:

- Identificar a fronteira da aplicação pela perspectiva do usuário, no que ele pode entender e descrever;
- A fronteira entre as aplicações deve ser identificada na separação das funções conforme os processos do negócio e não pelo aspecto tecnológico;
- Em projetos de melhoria, a fronteira estabelecida no início do projeto deve estar em concordância com a fronteira já estabelecida para a aplicação.

Segundo Vasquez, Simões e Albert, 2005, as seguintes dicas podem ajudar a identificar a fronteira de aplicação:

- Obter a documentação do fluxo de dados e desenhar em volta uma fronteira para destacar quais são partes internas e externas a aplicação;
- Observar como os dados são armazenados;
- Identificar áreas funcionais como entidades e processos;
- Observar o gerenciamento da aplicação se é desenvolvida ou mantida na sua totalidade por uma equipe distinta;
- Observar se o software possui ordens de serviço específicas e independentes.

Abaixo podemos observar uma ilustração de fronteira de aplicação:



Figura 6 – Fronteira de aplicação

Fonte: Uma aplicação da APF nas estimativas de projetos web (Hazan, 2003)

Podemos concluir reforçando que a identificação correta da fronteira de aplicação é de vital importância para que a contagem de pontos por função seja bem sucedida, e que essa identificação deve ser baseada na lógica do negócio a atender e não por características técnicas.

3.5 Escopo da contagem

O objetivo da definição do escopo consiste em definir quais funções serão incluídas na contagem: se abrange uma ou mais aplicações, podendo compreender todas as funcionalidades; apenas as funcionalidades em uso pelo usuário ou funcionalidades específicas.

A definição do escopo em conjunto com a definição da fronteira de aplicação deve ser feita com atenção para não haver equívoco na contagem, como contar a uma mesma transação para mais de uma aplicação. Outro engano que pode acontecer é a contagem em duplicidade de tabelas, considerando uma tabela como arquivo lógico interno para uma aplicação e como arquivo de interface externa em outra aplicação. Segundo HAZAN (2003), “o escopo da contagem deve incluir todos os componentes necessários para apresentar as necessidades do negócio.”.

3.6 Funções do tipo dado e transação

Funções do tipo dado são aquelas projetadas para atender ao usuário no que se refere à base de dados da aplicação. As funções estão classificadas em arquivo lógico interno e arquivo de interface externa. Antes de detalhar estas classificações, precisamos entender o que significa o termo arquivo. Arquivo neste caso não se refere necessariamente a um arquivo do sistema operacional, tem um sentido mais abrangente, se referindo a um grupo de dados relacionados logicamente e reconhecido pelo usuário. Lembrando que o usuário como definido anteriormente não se refere especificamente a uma pessoa física. Pode ocorrer de um ou mais arquivos ou até tabelas de um banco ser tratados como um único arquivo pela APF, ou seja, a forma como a aplicação é implementada para o armazenamento dos dados não relevante para determinar as funções do tipo dado.

As funções do tipo transação como o próprio nome sugere, refere-se aos processos da aplicação, como a informação é processada e através de quais funcionalidades. As funções do tipo transação estão classificadas em: entrada externa, saída externa e consulta externa. Estas classificações serão abordadas a seguir.

3.7 Pontos de função não-ajustados

Uma vez apuradas as funções do tipo dado e transação, o próximo passo é chegar ao número de pontos de função não-ajustados. As funções de dado e transação devem ser analisadas e classificadas quanto a sua complexidade em três níveis: alta, média e baixa. A complexidade de uma função do tipo dado compreende em quantos campos e registros são necessários para armazenar a informação. A função de transação por sua vez compreende no número de tipos de dados e arquivos envolvidos. A escala de complexidade é atribuída conforme uma tabela que trata dos pontos de função não-ajustados. O resultado final deste passo é a soma da contagem da complexidade das funções, chamada de pontos de função não-ajustados.

3.8 Fator de ajuste e pontos de função ajustados

Esta fase tem como objetivo estabelecer um fator de ajuste para a soma dos pontos de função não-ajustados, baseando-se em 14 critérios de influência. O ajuste fica em média 35% de acordo com as 14 características de influência, definidas pelo IFPUG. Como mencionado anteriormente, esta fase não está compreendida no padrão ISO/IEC de medição funcional, o IFPUG para poder se adequar à norma tornou este passo opcional na APF.

A última etapa da APF é calcular os pontos de função ajustados, o qual em posse dos pontos por função não-ajustados e o fator de ajuste definido, é aplicada uma forma matemática, específica para cada tipo de contagem (projeto de desenvolvimento, melhoria e aplicação).

Neste capítulo foram descritas as fases da APF, a seguir estudaremos como determinar os tipos de dados a serem contados.

4 FUNÇÃO DO TIPO DADO

Neste capítulo será abordada uma das etapas do processo de contagem de pontos por função, o processo de identificar e contar as funções do tipo dado. Conforme descrito anteriormente, funções do tipo dado referem-se à base de dados da aplicação, estando dividida em lógico interno (ALI) e arquivo de interface externa (AIE). A seguir será detalhada cada uma destas divisões, e o processo conforme figura abaixo:



Figura 7 – Processo de contagem de funções do tipo dados

Fonte: Vasquez, Simões e Albert (2005)

4.1 Arquivo lógico interno

ALI por definição é considerado um grupo de dados, identificável pelo usuário, estando logicamente relacionado e mantido dentro da fronteira da aplicação. A principal função de um ALI é obviamente armazenar dados adicionados, modificados ou excluídos pela aplicação através de funções do tipo transação (Vasquez, Simões e Albert, 2005).

Segundo HAZAN (2003): ALI são grupos de dados ou informações de controle especificado pelo usuário logicamente relacionado, cuja manutenção é efetuada dentro da fronteira da aplicação. O objetivo do ALI é armazenar dados mantidos através de um ou mais processos da aplicação sendo contada.

Para exemplificar o que são considerados arquivos lógicos internos, podemos considerar uma tabela de usuários da aplicação, com campos conhecidos como usuário, *login*

e senha de acesso. Quanto ao projeto que será estudado neste trabalho, na função de inclusão de lançamentos contábeis temos o arquivo de lançamentos, que será considerado como um ALI.

Segundo (Vasquez, Simões e Albert, 2005), não são considerados ALI:

- Arquivos temporários de classificação;
- Arquivos que existem apenas durante a execução da aplicação, sem armazenar informações;
- Arquivos de *backup*;
- Arquivos gerados para processamento de outras aplicações;
- Arquivos de índices;
- Visões (*views*).

4.2 Arquivo de interface externa

AIE por definição é considerado um grupo de dados, identificável pelo usuário, estando logicamente relacionado e mantido fora da fronteira da aplicação. A finalidade de um AIE é utilizar dados para referência no processamento dentro da fronteira de aplicação medida, porém o AIE deve obrigatoriamente ser mantido por outra aplicação. (Vasquez, Simões e Albert, 2005).

Segundo HAZAN (2003): AIE são grupos de dados ou informações de controle especificado pelo usuário logicamente relacionado, cuja manutenção é efetuada dentro da fronteira de outra aplicação. O objetivo do AIE é armazenar dados referenciados através de um ou mais processos da aplicação sendo contada.

Um exemplo de AIE, voltando ao projeto de estudo, está na função de inclusão de lançamentos. Nesta função temos o arquivo de plano de contas, cuja manutenção é feita em outra aplicação do sistema de contabilidade, mas é necessário utiliza-lo para associar um lançamento contábil a uma conta contábil existente.

Podemos concluir então que a diferença entre ALI e AIE é que o primeiro está dentro da fronteira da aplicação a ser medida já o outro não, logo um AIE é um ALI de outra aplicação.

4.3 Contagem de ALI e AIE

O primeiro passo para contagem de pontos por função de ALI e AIE é identificar os mesmos, para isso devem ser observadas regras para determinação de ALI e AIE. Cada ALI e AIE devem ser classificados conforme a sua complexidade funcional, estando dividida em três tipos: alta, média e baixa (IFPUG). Para determinar sua classificação são considerados o número de tipos de dados e o número de tipos de registro. Uma vez quantificados estes aspectos, a classificação da complexidade é determinada conforme abaixo:

Tabela 1 - Tabela de complexidade funcional de ALI e AIE

Fonte: IFPUG

	Tipos de dados			
		< 20	20 – 50	> 50
Tipos de registros	1	Baixa	Baixa	Média
	2 – 5	Baixa	Média	Alta
	> 5	Média	Alta	Alta

Exemplificando a tabela acima, um ALI com 60 tipos de dados em 3 tipos de registros é considerado como complexidade alta. Novamente voltando ao projeto de estudo, o arquivo de lançamentos (ALI) possui 45 tipos de dados em 1 tipo de registro, sendo a complexidade é considerada baixa. A seguir serão descritos os conceitos e regras de contagem para tipo de dado e tipo de registro.

4.3.1 Tipo de dado

Um tipo de dado é um campo único, não repetido e reconhecido pelo usuário. Para contagem de um tipo de dado devem ser observadas as seguintes regras:

- Deverá ser considerado um tipo de dado para cada campo único reconhecido pelo usuário e não repetido, utilizado por um ALI ou AIE por meio de um processo. Em se tratando de um campo de data, mesmo que existam campos separados para dia, mês e ano, estes devem ser contados como apenas um tipo de dado;
- Caso o ALI ou AIE possua duas aplicações ou mais que o mantenham ou referenciam, devem ser contados apenas os campos utilizados pela aplicação que está sendo contada;
- Deverá ser considerado um tipo de dado para cada campo solicitado pela aplicação para estabelecer um relacionamento com outro ALI ou AIE.

No projeto de estudo, na função de inclusão de lançamentos, existe uma verificação do usuário se este está autorizado a incluir lançamentos. Para esta verificação é acessado o arquivo de usuários (AIE) e os campos acessados são: código de usuário e o campo que indica se permite inclusão de lançamentos. Neste caso são contados 2 tipos de dados para o AIE, apesar de existirem outros campos, como nome e outros controles de acesso, mas para esta aplicação somente deve ser considerado 2 tipos de dados e 1 tipo de registro.

Concluindo, cada campo utilizado pela aplicação é considerado como tipo de dado e deve ser considerado na contagem para obter a complexidade do ALI ou AIE ao qual o campo pertence.

4.3.2 Tipo de registro

Tipo de registro é um subgrupo de dados de um ALI ou AIE, reconhecido pelo usuário. Para contagem de um tipo de registro devem ser observadas as seguintes regras:

- Deve ser considerado um tipo de registro para subgrupo de dados de um ALI ou AIE;
- Caso não identificado nenhum subgrupo de dados, deve ser considerado um tipo de registro para cada ALI ou AIE.

No projeto estudado, um lançamento contábil possui tipos de dados como: Conta, data, valor, documento, histórico e complemento do lançamento. Estes tipos de dados estão

armazenados logicamente em 2 arquivos distintos, porém são considerados como um grupo de dados do lançamento, logo, deve ser considerado 1 tipo de registro.

4.3.3 Contribuição do ponto por função não-ajustados das funções do tipo dado

Após a determinação dos tipos de dados, tipos de registros e determinadas a complexidade, o próximo passo é classificar a complexidade encontrada dentro da tabela de contribuição, conforme abaixo:

Tabela 2 - Tabela de contribuição do ponto por função das funções do tipo dado.

Fonte: IFPUG

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

4.3.4 Considerações finais sobre contagem de ALI e AIE

A identificação dos arquivos lógicos internos e arquivos de interface externa podem parecer bem próximos da interpretação de um modelo ER, mas devemos estar atentos, por que podem ocorrer interpretações equivocadas em determinadas situações. Na contagem também é necessário conseguir distinguir os requisitos de armazenamento funcionais e não funcionais da aplicação, que estão classificados como dados de código ou metadados, dados de referência e dados de código.

Dados de código ou metadados nunca devem ser considerados como arquivos lógicos, não devem ser considerados na contagem, por que são implementações de requisitos técnicos e por isso não devem influenciar no tamanho funcional da aplicação. Um exemplo de dados de código pode ser uma tabela de siglas de estado ou qualquer outra tabela que raramente tenha seu conteúdo modificado.

Dados de referência são informações no nível de regra de negócio da aplicação, armazenados em arquivos, como por exemplo, regras e cálculos de incidência de impostos em uma nota fiscal.

Existem também outras entidades que não são consideradas como arquivos lógicos:

- Arquivos de índices;
- Arquivos com dados consolidados ou visões;
- Arquivos temporários ou de classificação;
- Entidades de ligação.

As entidades acima não são consideradas como arquivos lógicos justamente por não terem sido apontadas como requisitos diretamente pelo usuário, são considerados meios técnicos de se atingir um determinado requisito.

Neste capítulo foi demonstrado como identificar tipos de dados e registros, as informações que a aplicação utilizará, a seguir estudaremos como determinar as transações a serem contadas na APF.

5 FUNÇÃO DO TIPO TRANSAÇÃO

Funções do tipo transação são processamentos realizados pela aplicação com objetivo de atender requisitos funcionais apontados pelo usuário. As funções do tipo transação estão divididas em três tipos: entrada externa, saída externa e consulta externa. A seguir será tratada a parte cada um destes tipos.

5.1 Entrada externa

Entradas externas são processamentos de dados recebidos de fora da fronteira de aplicação. Seu objetivo é a manutenção de um ou mais arquivos lógicos internos e/ou modificar o comportamento da aplicação (Vasquez, Simões e Albert, 2005).

Segundo HAZAN (2003): Entrada externa é um processo elementar que processam dados ou informações de controle que vem do lado de fora da fronteira de aplicação. Tem como objetivo manter um ou mais ALI e/ou alterar o comportamento do sistema.

Podemos exemplificar a entrada externa a partir de uma janela de manutenção de uma tabela qualquer onde tenha opção de incluir, alterar ou excluir registros da tabela. Neste exemplo devem ser contadas três entradas externas. Este exemplo também é encontrado no projeto estudado, onde existem funções de inclusão, alteração e exclusão de lançamentos. Outro exemplo é a atualização de um cadastro de cliente com o seu limite de crédito, após emissão de uma nota fiscal de venda. Por outro lado, telas de *login* de sistema e menus, por exemplo, não são consideradas entradas externas.

5.2 Saída externa

Saídas externas são processamentos de dados que são enviados para fora da fronteira de aplicação. Seu objetivo é demonstrar informação através de um processamento lógico, não

apenas a exibição de dados. Nesta lógica de processamento, por definição deve conter algum cálculo, ou criar dados derivados (Vasquez, Simões e Albert, 2005).

Segundo HAZAN (2003): Saída externa é um processo elementar que enviam dados ou informação de controle para fora da fronteira da aplicação. Tem como objetivo principal apresentar a informação para o usuário através de processamento lógico adicional a recuperação de dados ou informação de controle. O processamento deve conter no mínimo uma fórmula matemática ou criar dados derivados, ou alterar comportamento da aplicação ou manter ALI.

Como exemplo comum de uma saída externa, podemos considerar um relatório de estatística de vendas de um determinado período, com totalização e percentual de vendas de cada representante em relação ao total vendido. Podemos considerar também outros exemplos como:

- Relatórios que atualizam algum tipo de arquivo;
- Consultas com cálculos ou dados derivados;
- Arquivos de remessa, como cobrança bancária;
- Gráficos estatísticos;
- Telas de *login* (com criptografia).

Por outro lado, não são consideradas saídas externas telas de ajuda (*help*) da aplicação, ou qualquer outra saída de informação que seja apenas exibição dos dados armazenados.

5.3 Consulta externa

Consulta externa são processos de simples recuperação e exibição de dados armazenados para fora da fronteira da aplicação. Seu objetivo é a demonstração de dados através da simples recuperação das informações de ALIs e/ou AIEs, devendo obrigatoriamente não conter nenhum tipo de cálculo, do contrário trata-se de uma saída externa (Vasquez, Simões e Albert, 2005).

Segundo HAZAN (2003): Consulta externa é um processo elementar que envia dados ou informação de controle para fora da fronteira de aplicação. Tem como objetivo principal apresentar informação para o usuário através da recuperação de dados ou informação de controle de um ALI ou AIE. O processamento lógico não contém fórmulas matemáticas ou cálculos e não cria dados derivados. Além disso, não mantém ALI durante o processamento nem altera o comportamento do sistema.

Como exemplos de consulta externa têm:

- Telas de ajuda (*help*);
- Gráficos de informações simples;
- Telas simples de *login*;
- Menus gerados dinamicamente baseados na configuração da aplicação.

Não são consideradas consultas externas as demonstrações de dados que envolvam cálculos ou derivações de dados, como relatórios e gráficos estatísticos, pois estas são consideradas saídas externas.

5.4 Contribuição do ponto por função não-ajustados das funções do tipo transação

Para chegarmos à contribuição do ponto por função das funções do tipo transação, precisamos observar primeiro o processo de contagem conforme figura abaixo:

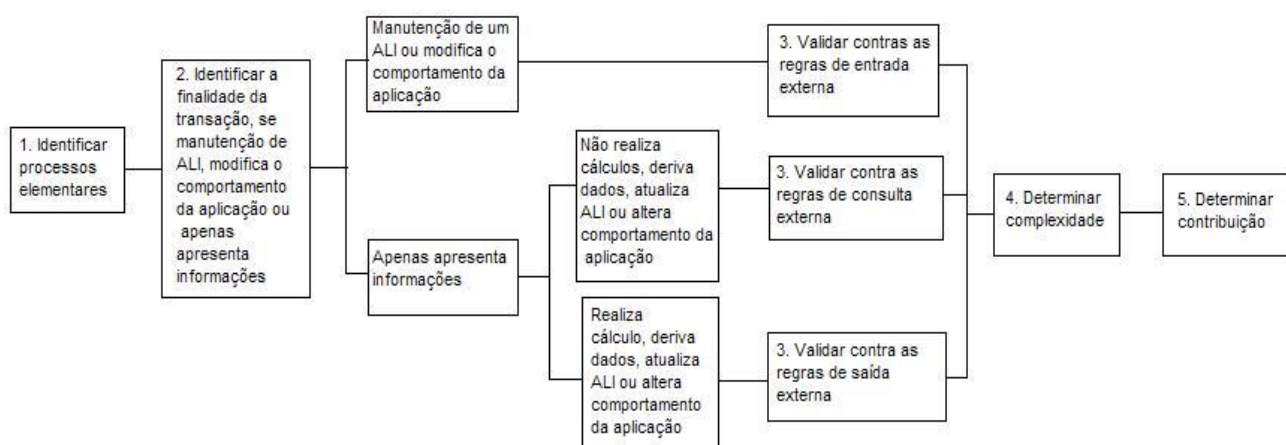


Figura 8 – Processo de contagem de funções do tipo transação

Fonte: Vasquez, Simões e Albert (2005)

No primeiro quadro, mais a esquerda da figura anterior, identificar os processos elementares, compreende-se por processo elementar a menor fração significativa para o usuário final. Esta definição é considerada uma questão chave na contagem de funções do tipo transação. Outra definição importante são as informações de controle, que são dados que influenciam um processo elementar. Este processo especifica o que, quando ou como os dados devem ser processados pela aplicação. Podemos dar o exemplo do *Internet Explorer*, o qual possui informações de controle para o funcionamento do *browser* durante a navegação entre as páginas da *internet* (Vasquez, Simões e Albert, 2005).

Assim como nas funções do tipo dado, devemos também classificar as funções do tipo transação quanto a sua complexidade funcional na aplicação. Cada entrada externa, saída externa e consulta externa deve ser classificada em alta, média ou baixa, de acordo com o número de arquivos referenciados e quanto ao número de tipo de dado (IFPUG).

Um arquivo referenciado é um ALI lido ou atualizado pela função do tipo transação ou ainda pode ser um arquivo de interface externa lido pela mesma função (Vasquez, Simões e Albert, 2005). Tipo de dado como vimos anteriormente, é um campo único e reconhecido pelo usuário. Para contar um arquivo referenciado devemos observar as seguintes regras:

- Contar um arquivo referenciado para cada ALI que é atualizado ou lido pela função;
- Contar um arquivo referenciado para cada AIE lido pela função;
- Embora o ALI e/ou AIE processar mais de um tipo de registro, deverá ser contado apenas uma vez;

Devemos considerar também que:

- Somente devemos contar como arquivo referenciado arquivos lógicos internos e arquivos de interface externa;
- Mesmo que o ALI e/ou AIE possa fazer várias leituras, deve ser contado apenas uma vez.

Para contagem dos tipos de dados, devemos observar as seguintes regras:

- Contar um tipo de dado para cada campo não repetido e reconhecido pelo usuário, que entra ou sai pela fronteira de aplicação e utilizado no processo. Não importa quantas vezes o campo entra ou sai da fronteira da aplicação, mesmo assim deve ser contado apenas uma vez;
- Não devem ser contados campos que durante o processo são derivados da aplicação ou recuperados de um ALI ou AIE, e que não atravessam a fronteira da aplicação;
- Troca de mensagens que sai da fronteira da aplicação, como tratamento de exceções, mensagem de término de processamento, etc.;

Não são considerados tipos de dados:

- Literais;
- Variáveis de paginação ou campos automáticos gerados pela aplicação.

Concluído o levantamento do número de arquivos referenciados e tipos de dados de cada entrada externa, saída externa e consulta externa, devemos classificar conforme as tabelas abaixo:

Tabela 3 – Tabela de complexidade para entradas externas

Fonte: IFPUG

Arquivos referenciados	Tipos de dados		
		< 5	5 – 15
< 2	Baixa	Baixa	Média
2	Baixa	Média	Alta
> 2	Média	Alta	Alta

Uma entrada externa que possui 1 arquivo referenciado, sendo 10 tipos de dados, é considerado como complexidade baixa.

Tabela 4 – Tabela de complexidade para saídas externas e consultas externas

Fonte: IFPUG

Arquivos referenciados	Tipos de dados		
		< 6	6 – 19
< 2	Baixa	Baixa	Média
2 – 3	Baixa	Média	Alta
> 3	Média	Alta	Alta

Uma saída externa de 3 arquivos referenciados e 12 tipos de dados, é considerado complexidade média.

Após o enquadramento de cada função do tipo transação conforme as tabelas anteriores, basta calcular a sua contribuição conforme a tabela abaixo:

Tabela 5 – Tabela de contribuição dos pontos por função das funções de transação

Fonte: IFPUG

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

Uma entrada externa de complexidade média deve ser considerado 4 pontos por função, conforme tabela do IFPUG.

Até aqui abordamos o processo da APF e como determinar os tipos de dados e transações a serem contados, a seguir será abordada a determinação do fator de ajuste a ser aplicado aos pontos de função não-ajustados.

6 FATOR DE AJUSTE

Até este capítulo abordamos os passos para contagem dos pontos por função não-ajustados. No que compreende a norma ISO para contagem funcional de uma aplicação, o valor para fator de ajuste não é considerado, ou seja, o total de pontos por função não-ajustados conforme a norma ISO é o resultado final da medida do tamanho funcional.

Conforme o IFPUG, para determinação do valor do fator de ajuste (VAF) deve ser considerado 14 características gerais de sistema (CGS), sendo que para cada característica deve ser determinado um nível de influência na aplicação.

6.1 Características gerais de sistema (CGS)

As características gerais de sistema para determinação do valor do fator de ajuste dos pontos por função não-ajustados são:

- Comunicação de dados;
- Processamento distribuído;
- Performance;
- Configuração altamente utilizada;
- Volume de transações;
- Entrada de dados *on-line*;
- Eficiência do usuário final;
- Atualização *on-line*;

- Complexidade de processamento;
- Reutilização;
- Facilidade de instalação;
- Facilidade de operação;
- Múltiplos locais;
- Facilidade de mudanças.

As características gerais de sistema conforme definição do IFPUG trata de funções que afetam a aplicação de uma maneira geral, diferente das funções do tipo dado e transação que refletem requisitos de armazenamento e processos respectivamente. Para cada característica deve ser atribuído um nível de influência, que varia de 0 a 5 conforme tabela abaixo:

Tabela 6 – Tabela de níveis de influência das características gerais de sistema

Fonte: IFPUG

Nível	Influência
0	Nenhuma
1	Mínima
2	Moderada
3	Média
4	Significativa
5	Grande

Assim que os níveis de influência das características gerais de sistema foram determinados, devemos fazer um somatório dos níveis de influência e aplicar na seguinte fórmula:

Fator de ajuste (VAF) = (total dos níveis de influência x 0,01) + 0,65.

Por exemplo, dado um determinado nível de influência para cada uma das 14 CGS:

CGS	Peso
Comunicação de dados	5
Processamento distribuído	2
Performance	2
Configuração altamente utilizada	2
Volume de transações	2
Entrada de dados <i>on-line</i>	5
Eficiência do usuário final	2
Atualização <i>on-line</i>	5
Complexidade de processamento	2
Reusabilidade	0
Facilidade de instalação	1
Facilidade de operação	2
Múltiplos locais	2
Facilidade de mudanças	2

Temos o total dos níveis de influência igual a 34, aplicando a fórmula:

$$\text{VAF} = (34 \times 0,01) + 0,65 = 0,99.$$

A seguir será explicado cada critério e como determinar seu nível de influência conforme o IFPUG.

6.1.1 Comunicação de dados

Comunicação de dados refere-se ao nível em que a aplicação comunica-se diretamente com o processador. Os dados são enviados e recebidos por meio de recursos de comunicação, como terminais conectados localmente à unidade de controle e protocolo de comunicação. Na tabela abaixo segue classificação do nível de influência:

Tabela 7 – Tabela de níveis de influência para comunicação de dados

Fonte: IFPUG

	Critério de influência.
0	Aplicação em <i>batch</i> .
1	Aplicação em <i>batch</i> , mas com entrada de dados ou impressão remota.
2	Aplicação em <i>batch</i> , mas com entrada de dados e impressão remota.
3	Aplicação possui entrada de dados <i>on-line</i> , <i>front-end</i> de teleprocessamento para um processamento <i>batch</i> ou sistema de consulta.
4	Aplicação mais que um <i>front-end</i> , mas suporta apenas um protocolo de comunicação.
5	Aplicação mais que um <i>front-end</i> e suporta mais de um protocolo de comunicação.

6.1.2 Processamento distribuído

Processamento distribuído refere-se ao nível de transferência de dados que a aplicação faz entre seus componentes dentro da fronteira de aplicação. Na tabela abaixo segue classificação do nível de influência:

Tabela 8 – Tabela de níveis de influência para processamento distribuído

Fonte: IFPUG

	Critério de influência.
0	Aplicação não participa da transferência de dados ou processamento de funções entre os componentes do sistema.
1	Aplicação prepara dados para processamento pelo usuário final em outro

	componente do sistema, como planilhas ou banco de dados.
2	Dados são preparados para transferência, então são processados em outro componente do sistema (não para processamento pelo usuário final).
3	Processamento distribuído e transferência de dados são feitos <i>on-line</i> em apenas uma direção.
4	Processamento distribuído e transferência de dados são feitos <i>on-line</i> em ambas direções.
5	O processamento de funções é feito dinamicamente no componente mais apropriado do sistema.

6.1.3 Performance

Performance refere-se ao nível de tempo de resposta e taxa de transações que influenciam o desenvolvimento da aplicação. Na tabela abaixo segue classificação do nível de influência:

Tabela 9 – Tabela de níveis de influência para performance.

Fonte: IFPUG

	Critério de influência.
0	O usuário não estabeleceu nenhum requisito especial sobre performance.
1	Requisitos de performance e projeto foram estabelecidos e revisados, mas nenhuma ação em especial foi feita.
2	Tempo de resposta ou taxa de transações são críticos durante as horas de pico. Não é necessário nenhum projeto especial para a utilização de CPU. O limite para o processamento é o dia seguinte.
3	Tempo de resposta ou taxa de transações são críticos durante todas as horas de trabalho. Não foi necessário nenhum projeto especial para a utilização de CPU. O limite de processamento é crítico.
4	Adicionalmente, requisitos especificados pelo usuário são exigentes o bastante para

	que tarefas de análise de performance sejam necessárias na fase de projeto.
5	Adicionalmente, ferramentas de análise de performance devem ser utilizadas nas fases de projeto, desenvolvimento e/ou implementação para que os requisitos de performance do usuário sejam atendidos.

6.1.4 Configuração altamente utilizada

Configuração altamente utilizada refere-se ao nível que restrições de recursos computacionais influenciam no desenvolvimento da aplicação. Na tabela abaixo segue classificação do nível de influência:

Tabela 10 – Tabela de níveis de influência para configuração altamente utilizada

Fonte: IFPUG

	Critério de influência.
0	Não existem restrições operacionais implícitas ou explícitas nos requisitos.
1	Existem restrições operacionais, mas são menos restritivas que uma aplicação típica. Não há esforço especial necessário ao atendimento dessas restrições.
2	Existem restrições operacionais, mas são típicas da aplicação. Há esforço especial necessário ao atendimento dessas restrições.
3	Existem requisitos específicos de processador para uma parte específica da aplicação.
4	Restrições operacionais explícitas necessitam de um processador dedicado ou utilização pesada do processador central.
5	Adicionalmente, existem limitações nos componentes distribuídos da aplicação.

6.1.5 Volume de transações

Volume de transações refere-se ao nível que o alto volume de transações influencia o projeto, desenvolvimento, instalação e suporte da aplicação. Na tabela abaixo segue classificação do nível de influência:

Tabela 11 – Tabela de níveis de influência para volume de transações

Fonte: IFPUG

	Critério de influência.
0	Não é previsto nenhum período de pico de transações.
1	São previstos períodos de pico de processamento, como mensal, por exemplo, mas o impacto no esforço do projeto é mínimo.
2	Volumes de transação regulares, semanais, por exemplo, são previstos. Há algum impacto no esforço do projeto.
3	Altos volumes de transação, diários, por exemplo, são previstos, conseqüentemente com impacto no esforço do projeto.
4	Altas taxas de transação definidas pelo usuário nos requisitos ou os níveis de serviço acordados são altos o bastante para requererem tarefas de análise de performance na fase de projeto.
5	Adicionalmente, existem requisitos de ferramentas de análise de performance nas fases de projeto, desenvolvimento e/ou instalação.

6.1.6 Entrada de dados *on-line*

Entrada de dados *on-line* refere-se ao nível de entradas de dados na aplicação através de transações interativas. Na tabela abaixo segue classificação do nível de influência:

Tabela 12 – Tabela de níveis de influência para entrada de dados *on-line*

Fonte: IFPUG

	Critério de influência.
0	Todas as transações são processadas em lote.
1	De 1% a 7% das transações são entradas de dados <i>on-line</i> .
2	De 8% a 15% das transações são entradas de dados <i>on-line</i> .
3	De 16% a 23% das transações são entradas de dados <i>on-line</i> .
4	De 24% a 30% das transações são entradas de dados <i>on-line</i> .

5	Mais de 30% das transações são entradas de dados <i>on-line</i> .
---	---

6.1.7 Eficiência do usuário final

Eficiência do usuário final refere-se ao nível de considerações sobre fatores humanos e facilidade de uso pelo usuário final influenciam o desenvolvimento da aplicação. Para determinar o nível de influência deste quesito, devemos observar antes o questionário abaixo e em seguida analisar a tabela dos níveis de influência.

O projeto inclui:

- Auxílio para navegação, como por exemplo, teclas de função, saltos, menus gerados dinamicamente;
- Menus;
- Ajuda *on-line* e documentação;
- Movimentação automática de cursor;
- Paginação;
- Impressão remota por meio de transações *on-line*;
- Teclas de função predefinidas;
- Tarefas em lote submetidas a transações *on-line*;
- Seleção feita por posicionamento de cursor em tela de dados;
- Uso intenso de vídeo reverso, brilho, cores e outros indicadores;
- Documentação impressa das transações;
- Interface de *mouse*;
- Janelas *pop-up*;
- Utilização de número mínimo de telas para executar uma função do negócio;

- Suporte a dois idiomas (conte como quatro itens);
- Suporte a mais de dois idiomas (conte como seis itens).

De acordo com os itens acima determine o nível de influência, conforme tabela abaixo:

Tabela 13 – Tabela de níveis de influência para eficiência do usuário final

Fonte: IFPUG

	Critério de influência.
0	Nenhum dos itens anteriores.
1	De um a três dos itens anteriores.
2	De quatro a cinco dos itens anteriores.
3	Seis ou mais dos itens anteriores, mas não existem requisitos específicos do usuário associados à eficiência.
4	Seis ou mais dos itens anteriores e requisitos explícitos sobre a eficiência para o usuário final são fortes o bastante para necessitarem de tarefas de projeto que incluam fatores humanos como, por exemplo, minimizar o número de toques no teclado, maximizar padrões de campo e uso de modelos.
5	Seis ou mais dos itens anteriores e requisitos explícitos sobre a eficiência para o usuário final são fortes o bastante para necessitarem do uso de ferramentas e processos especiais para demonstrar que os objetivos foram alcançados.

6.1.8 Atualização *on-line*

Atualização *on-line* refere-se ao nível em que os arquivos lógicos internos são atualizados *on-line*. Na tabela abaixo segue classificação do nível de influência:

Tabela 14 – Tabela de níveis de influência para atualização *on-line*

Fonte: IFPUG

	Critério de influência.
0	Não há nenhuma atualização <i>on-line</i> .

1	Existe atualização <i>on-line</i> de um a três arquivos. Volume de atualização é pequeno e a recuperação é fácil.
2	Existe atualização <i>on-line</i> de quatro ou mais arquivos. Volume de atualização é pequeno e a recuperação é fácil.
3	Atualização dos arquivos internos é na maioria <i>on-line</i> .
4	Adicionalmente, a proteção contra a perda de dados é essencial e foi especialmente projetada e programada no sistema.
5	Adicionalmente, o alto volume de processamento torna necessária a análise do custo do processo de recuperação. São incluídos procedimentos altamente automatizados com um mínimo de intervenção do operador.

6.1.9 Complexidade de processamento

Complexidade de processamento refere-se ao nível em que o processamento lógico e/ou matemático influencia o desenvolvimento da aplicação. Para poder determinar o nível de influência antes é preciso observar os seguintes itens:

- Controle sensível e/ou processamento específico de segurança da aplicação. Exemplo: processamento especial de auditoria;
- Processamento lógico extensivo. Exemplo: sistema de controle de crédito;
- Processamento matemático extensivo. Exemplo: sistema de otimização de corte de tecidos;
- Muito processamento de exceção resultando em transações incompletas que devem ser processadas novamente. Exemplo: transação incompleta em ATM em função de problemas de teleprocessamento, falta de dados ou de edição;
- Processamento complexo para manipular múltiplas possibilidades de entrada e saída, como, por exemplo, multimídia ou independência de dispositivo. Exemplo: sistema de extrato de conta corrente que emite via terminal de retaguarda, auto-atendimento, pela *internet*, *e-mail*, celular.

De acordo com os itens acima determine o nível de influência, conforme tabela abaixo:

Tabela 15 – Tabela de níveis de influência para complexidade de processamento

Fonte: IFPUG

	Critério de influência.
0	Nenhum dos itens anteriores.
1	Qualquer um dos itens anteriores.
2	Quaisquer dois itens anteriores.
3	Quaisquer três itens anteriores.
4	Quaisquer quatro itens anteriores.
5	Todos os cinco itens anteriores.

6.1.10 Reutilização

Reutilização refere-se ao nível em que a aplicação e seu código foram projetados, desenvolvidos e suportados para serem utilizados em outras aplicações. Na tabela abaixo segue classificação do nível de influência:

Tabela 16 – Tabela de níveis de influência para reutilização

Fonte: IFPUG

	Critério de influência.
0	Não há código reutilizável.
1	Código reutilizável é utilizado na aplicação.
2	Menos de dez por cento do código fonte da aplicação foi construído levando em consideração o uso em mais de uma aplicação.
3	Dez por cento ou mais do código fonte da aplicação foi construído levando em consideração o uso em mais de uma aplicação.

4	A aplicação foi especificamente empacotada e/ou documentada para fácil reutilização. Ela é customizada pelo usuário no nível de código.
5	A aplicação foi especificamente empacotada e/ou documentada para fácil reutilização. Ela é customizada pelo usuário por meio de manutenção de parâmetros.

6.1.11 Facilidade de instalação

Facilidade de instalação refere-se ao nível em que a conversão de ambientes preexistentes influencia o desenvolvimento da aplicação. Um plano e/ou ferramentas de conversão e instalação foram fornecidos e testados durante a fase de teste do sistema. Na tabela abaixo segue classificação do nível de influência:

Tabela 17 – Tabela de níveis de influência para facilidade de instalação

Fonte: IFPUG

	Critério de influência.
0	O usuário não definiu considerações especiais, assim como não é requerido nenhum <i>setup</i> para a instalação.
1	O usuário não definiu considerações especiais, mas é necessário <i>setup</i> para instalação.
2	Requisitos de instalação e conversão foram definidos pelo usuário, e guias de conversão e instalação foram fornecidas e testadas. Não é considerado importante o impacto da conversão.
3	Requisitos de instalação e conversão foram definidos pelo usuário, e guias de conversão e instalação foram fornecidas e testadas. É considerado importante o impacto da conversão.
4	Além do item 2, ferramentas de instalação e conversão automáticas foram fornecidas e testadas.
5	Além do item 3, ferramentas de instalação e conversão automáticas foram fornecidas e testadas.

6.1.12 Facilidade de operação

Facilidade de operação refere-se ao nível em que a aplicação atende a alguns aspectos operacionais como inicialização, segurança e recuperação. A aplicação minimiza a necessidade de atividades manuais como montagem de fitas, manipulação de papel e intervenção manual do operador. Na tabela abaixo segue classificação do nível de influência:

Tabela 18 – Tabela de níveis de influência para facilidade de operação

Fonte: IFPUG

	Critério de influência.
0	Não foi estabelecida pelo usuário outra consideração que não os procedimentos de segurança normais.
1 – 4	Um, alguns ou todos os itens abaixo são válidos para a aplicação. Conte apenas aqueles que sejam aplicáveis, sendo que cada vale um ponto, exceto os itens que mencionam a quantidade de pontos. Procedimentos de inicialização, salvamento e recuperação foram fornecidos, mas é necessária a intervenção do operador; Procedimentos de inicialização, salvamento e recuperação foram fornecidos, e não é necessária a intervenção do operador (vale dois pontos); A aplicação minimiza a necessidade de montagem de fitas; A aplicação minimiza a necessidade de manipulação de papel.
5	Aplicação projetada para operação não assistida, ou seja, não é necessária nenhuma intervenção por parte do operador, exceto as etapas de inicialização e término da aplicação. A recuperação automática de erros é uma característica da aplicação.

6.1.13 Múltiplos locais

Múltiplos locais refere-se ao nível em que a aplicação foi projetada, desenvolvida e suportada para diferentes ambientes de hardware e software. Na tabela abaixo segue classificação do nível de influência:

Tabela 19 – Tabela de níveis de influência para múltiplos locais

Fonte: IFPUG

	Critério de influência.
0	Os requisitos do usuário não consideram a necessidade de mais de um usuário/local de instalação.
1	Necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi projetada para operar apenas nos mesmos ambientes de hardware e software.
2	Necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi projetada para operar em apenas ambientes de hardware e software similares.
3	Necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi projetada para operar em ambientes diferentes de hardware e software.
4	Adicionalmente aos itens 1 ou 2, plano de suporte e documentação são fornecidos e testados para suportar a aplicação em múltiplos locais.
5	Adicionalmente ao item 3, plano de suporte e documentação são fornecidos e testados para suportar a aplicação em múltiplos locais.

6.1.14 Facilidade de mudanças

Facilidade de mudanças refere-se ao nível em que a aplicação foi desenvolvida para facilitar a mudança de sua lógica e/ou estrutura de dados. Para poder determinar o nível de influência antes é preciso observar os seguintes itens:

- São fornecidos mecanismos de consulta flexível, que permitem a manipulação de pedidos simples; por exemplo, lógica de e/ou aplicada a apenas um arquivo lógico (considerar um item);
- São fornecidos mecanismos de consulta flexível, que permitem a manipulação de pedidos de média complexidade; por exemplo, lógica de e/ou aplicada a mais de um arquivo lógico (considerar dois itens);
- São fornecidos mecanismos de consulta flexível, que permitem a manipulação de pedidos complexos; por exemplo, lógica de e/ou combinadas em um ou mais arquivos lógicos (considerar três itens);

- Dados de controle do negócio são mantidos pelo usuário por meio de processos interativos, mas as alterações só têm efeito no próximo dia útil;
- Dados de controle do negócio são mantidos pelo usuário por meio de processos interativos, e as alterações têm efeito imediato (considerar como dois itens).

De acordo com os itens acima determine o nível de influência, conforme tabela abaixo.

Tabela 20 – Tabela de níveis de influência para facilidade de mudanças

Fonte: IFPUG

	Critério de influência.
0	Nenhum dos itens anteriores.
1	Qualquer um dos itens anteriores.
2	Quaisquer dois itens anteriores.
3	Quaisquer três itens anteriores.
4	Quaisquer quatro itens anteriores.
5	Todos os cinco itens anteriores.

6.2 Considerações

Inicialmente quando Allan Albrecht concebeu a metodologia de pontos por função não existiam quaisquer critérios para determinar o fator de ajuste, sendo determinado de forma muito subjetiva. O fator de ajuste variava em torno de 25% nos pontos de função não-ajustados, sendo que em 1984 houve uma revisão técnica, sendo criado então os 14 CGS descritos anteriormente. Com esta revisão, o fator de ajuste passou a variar em torno de 35% nos pontos de função não-ajustados. Novamente devemos destacar que o fator de ajuste é opcional devido ao enquadramento da metodologia a norma ISO 14143 em 2002. Mesmo antes do fator de ajuste ser considerado opcional pelo IFPUG, a entidade fez uma pesquisa e constatou que vários usuários da metodologia já não adotavam o fator de ajuste. O assunto é

tão polêmico na comunidade dos usuários da APF que o IFPUG criou um grupo de trabalho para se dedicar à questão do fator de ajuste. As críticas são das mais variadas:

- Percentual de variação de 35% insuficiente;
- Falta de requisitos considerados importantes;
- Uso do fator de ajuste não trazer nenhum benefício na estimativa de esforço;
- Outros defendem que os 14 CGS são muitos subjetivos e que dá margem para diferentes interpretações, comprometendo a contagem final.

Por este item ser tão polêmico e subjetivo e o próprio IFPUG inseguro quanto ao assunto, neste trabalho de conclusão não será implementado o fator de ajuste ao aplicar a APF no projeto que será estudado. Outro motivo para não aplicar o fator de ajuste é que a empresa Rech Informática Ltda, apresentada a seguir neste trabalho, ao procurar padronizar seus processos busca sempre se enquadrar a normas técnicas como a ISO, que não considera o fator de ajuste. A empresa é participante do PGQP e visa futuramente implementar CMM.

7 CÁLCULO DOS PONTOS DE FUNÇÃO AJUSTADOS

Este é o último passo da APF, onde depois de atingido o número de pontos de função são aplicadas fórmulas matemáticas para chegar ao resultado final. A seguir serão descritas as fórmulas para cada tipo de contagem: projeto de desenvolvimento, projeto de melhoria e aplicação.

7.1 Projeto de desenvolvimento

Para cálculo dos pontos de função para projetos de desenvolvimento deverá ser aplicada a seguinte fórmula:

$$DFP = (UFP + CFP) \times VAF$$

Os termos da fórmula são:

- DFP: Número de pontos por função do projeto de desenvolvimento;
- UFP: Número de pontos de função não-ajustados das funções disponíveis após a instalação da aplicação, exceto as funções de conversão (contagem final do projeto, o realizado);
- CFP: Número de pontos de função das funções de conversão;
- VAF: Valor do fator de ajuste.

Como podemos observar a fórmula aplicada para estimativa em projetos não é complexa, o que determinará o sucesso da estimativa está realmente nas fases iniciais do processo de contagem.

7.2 Projeto de melhoria

Para cálculo dos pontos de função para projetos de melhoria deverá ser aplicada a seguinte fórmula:

$$EFP = [(ADD + CHGA + CFP) \times VAFA] + (DEL \times VAFB)$$

Os termos da fórmula são:

- EFP: Número de pontos de função do projeto de melhoria;
- ADD: Número de pontos de função não-ajustados das novas funções;
- CHGA: Número de pontos de função não-ajustados das funções modificadas, levando em consideração o funcionamento após a alteração;
- CFP: Número de pontos de função não-ajustados de funções de conversão;
- VAFA: Valor do fator de ajuste da aplicação após o projeto de melhoria;
- DEL: Número de pontos de função não-ajustados das funções excluídas da aplicação;
- VAFB: Valor do fator de ajuste da aplicação antes do projeto de melhoria.

A fórmula empregada em projetos de melhoria é mais complexa que a de projetos de desenvolvimento justamente por que a primeira envolve uma aplicação já em uso, devendo ser calculada a aplicação antes e depois do projeto. Neste tipo de contagem devemos considerar apenas as funções adicionadas, alteradas e excluídas da aplicação. Funções que não serão afetadas no projeto de melhoria não deverão ser consideradas na contagem, novamente podemos observar a importância na fase inicial do processo, que envolve a determinação da fronteira. Outra questão na contagem que deve ser destacada são as funções modificadas que são utilizadas por mais de uma transação, neste caso deve-se considerar que todas as transações foram alteradas, devendo então ser incluídas na contagem dos pontos de função não ajustados.

7.3 Aplicação

Para cálculo dos pontos de função para aplicações deve aplicar duas fórmulas matemáticas. A primeira calcula o tamanho da aplicação a partir apenas das funções solicitadas pelo usuário, sem considerar funções de conversão. A segunda recalcula o seu tamanho após um projeto de melhoria ter alterado suas funcionalidades. A primeira fórmula a ser aplicada lembra a fórmula do projeto de desenvolvimento, como podemos ver abaixo:

$$AFP = ADD + VAF$$

Os termos da fórmula são:

- AFP: Número de pontos de função ajustados da aplicação;
- ADD: Número de pontos de função não-ajustados das funções;
- VAF: Valor do fator de ajuste da aplicação.

Após o projeto de melhoria, devemos aplicar a fórmula abaixo:

$$AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)] \times VAFA$$

Os termos da fórmula são:

- AFP: Número de pontos de função ajustados da aplicação;
- UFPB: Número de pontos de função não-ajustados antes do projeto de melhoria;
- ADD: Número de pontos de função não-ajustados das funções incluídas no projeto;
- CHGA: Número de pontos de função não-ajustados das funções alteradas depois do término do projeto;
- CHGB: Número de pontos de função não-ajustados das funções alteradas antes do início do projeto;
- DEL: Número de pontos de função não-ajustados das funções excluídas pelo projeto;

- VAFA: Valor do fator de ajuste depois do projeto.

Neste capítulo encerramos a explanação do processo de contagem de pontos por função, no capítulo seguinte será apresentada empresa objeto de estudo e o projeto de melhoria a ser aplicada a APF.

8 ESTUDO DE CASO

8.1 Apresentando a empresa estudada

A Rech Informática Ltda, empresa estudada neste trabalho de conclusão, sediada em Novo Hamburgo/RS, vem atuando desde 1990 no mercado de software para gestão empresarial. A Rech possui seus clientes predominantemente no estado do Rio Grande do Sul, atuando mais especificamente no Vale dos Sinos e região metropolitana de Porto Alegre, mas têm alguns clientes também nos estados de Santa Catarina, Paraná e São Paulo. A empresa foi constituída a partir de um alicerce sólido de valores humanos, conjugando experiência, seriedade e competência nas áreas de informática para gestão empresarial.

A Rech Informática foi fundada em 15 de maio de 1990 pelos irmãos Carlos Vanderlei Rech e Rovani Marcelo Rech, que desde o início estiveram comprometidos em oferecer ao mercado produtos e serviços de qualidade. Foi idealizado o desenvolvimento de um software completo, que eliminasse completamente os retrabalhos e desperdícios nas empresas, através do conhecimento de gestão empresarial e do uso de tecnologias viáveis e produtivas. O objetivo principal da Rech Informática é a conquista diária da satisfação de seus clientes. Por isto, a empresa constantemente investe em novas tecnologias e no treinamento de pessoal, aprimorando o SIGER® e a qualidade dos serviços prestados. Com o total comprometimento em atingir este objetivo, a Rech Informática está constantemente investindo na empresa. Estes investimentos são tanto em estrutura física, quanto em recursos de pessoal, sempre buscando melhorar cada vez mais o relacionamento da empresa com seus clientes, fornecedores e colaboradores.

8.2 Processos da Rech Informática

Os processos da Rech podem ser vistos na figura abaixo:

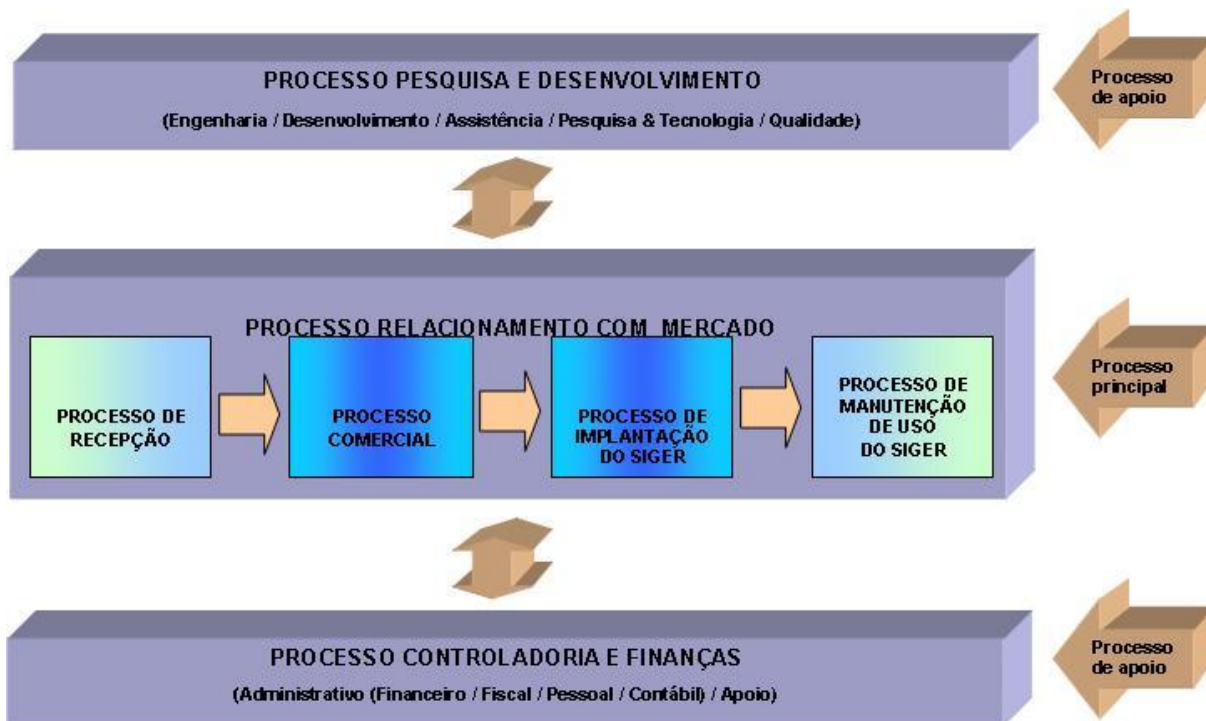


Figura 9 – Processos da Rech Informática Ltda
Fonte: MANUAL RECH

8.2.1 Processo principal – Relacionamento com mercado

O processo principal da Rech é composto dos subprocessos: recepção, comercial, implantação e manutenção do SIGER®. Os processos de implantação e manutenção que são responsáveis pelo contato direto com o cliente, é por este processo que a demanda de clientes atuais e novos chegam a empresa. Este processo tem como uma das tarefas fazer o levantamento de requisitos, a partir da necessidade dos clientes. Os colaboradores envolvidos neste processo possuem apenas capacitação do negócio e não de programação ou engenharia de software. Na empresa chamamos de levantamento do “o quê?”, o qual é devidamente registrado no sistema interno de controle o SICLA.

8.2.2 Processo de apoio – Pesquisa e desenvolvimento

O processo de apoio de pesquisa e desenvolvimento é composto dos subprocessos: engenharia, desenvolvimento, assistência, pesquisa e tecnologia, e qualidade. Este trabalho

está direcionado ao subprocesso de engenharia, é nele que a demanda recepcionada do processo principal da empresa chega. Os colaboradores que participam deste processo são programadores mais experientes da empresa, que possuem considerável noção de negócio para então traduzir o “o quê?” em “como?”. É neste processo que é feita a análise de requisitos, interpretando a necessidade do cliente e traduzindo em instruções aos programadores do processo de desenvolvimento. Todo este processo é registrado no sistema de controle interno da empresa, o SICLA.

8.2.3 Processo de apoio – Controladoria e finanças

O processo de apoio de controladoria e finanças é composto dos subprocessos: administrativo e apoio (limpeza e manutenção). Este processo, de forma sucinta, é responsável pelo funcionamento da empresa, tanto na parte administrativa como na manutenção da infra-estrutura da empresa. Este processo não tem contato direto com o assunto deste trabalho, está sendo apenas apresentado para o completo entendimento dos processos da empresa.

Todos os processos da empresa são registrados no sistema de controle interno da empresa o SICLA, um software construído pela própria Rech Informática para uso próprio. É a principal ferramenta utilizada para registrar, organizar e sincronizar os movimentos e operações de seus colaboradores, para planejar e acompanhar os processos da empresa e para realizar a análise dos resultados. Este software é utilizado por todos os setores da Rech e está em constante evolução para cada vez mais atender todas as necessidades de registros e controles da nossa gestão, tanto nos processos principais quanto de apoio. O SIGER® também é utilizado no processo de apoio Controladoria, para gestão Contábil, Fiscal, Financeiro e Pessoal. A seguir será estudado um aplicativo que trata da APF.

8.3 Análise de programas sobre APF

Foi feita pesquisa na internet por aplicativos que tratam sobre a APF e foi encontrado apenas o aplicativo “APF”, desenvolvido por Ivan Mescenas. Este aplicativo possui muitas versões, mas a versão que foi estudada neste trabalho foi a versão 2.2.0.0. O aplicativo foi desenvolvido em Delphi 5 e banco de dados Paradox. Segundo Mescenas: “A ferramenta APF - Análise de Pontos de Função foi construída para dar suporte à contagem de pontos de função de aplicativos (sistemas). A ferramenta está em conformidade com a metodologia de

contagem prevista no manual de práticas de contagens, versão 4.1, do IFPUG. Nesta versão foram acrescentadas características também do NESMA (*Netherlands Software Metrics Users Association*)”.

Como características gerais, o aplicativo possui:

- Manual do aplicativo;
- Barras de ferramentas para auxílio;
- Configuração de segurança;
- Configuração de preferências de usuário.

Apesar de o aplicativo ter muitos recursos, atender padrões de mensuração de aplicações e também ser gratuito, ele é muito complexo para o propósito dos colaboradores da Rech. A proposta deste trabalho é aprender esta metodologia e ensinar aos colaboradores do processo de engenharia da Rech, aliado a um protótipo baseado no básico da APF, dentro da norma ISO. A partir do protótipo concluído e exercitado no processo da empresa, a meta será incorporá-lo ao SICLA. A Rech, através dos seus diretores, sempre teve o costume de desenvolver as suas próprias ferramentas por vários motivos: pelo fato de poder melhor adaptá-la ao seu processo; não depender de ajuste de terceiros; não depender de suporte; poder modificar a aplicação quando necessário e da maneira desejada.

O protótipo proposto será desenvolvido com o mesmo compilador que é utilizado para manutenção do SIGER, NetExpress versão 3.0 e o sistema de arquivos que será utilizado é o sistema nativo do NetExpress. As funções que o protótipo conterà são: inclusão, alteração e exclusão de contagem de projetos de desenvolvimento, projetos de melhoria e aplicação. Seguindo o processo de contagem da APF serão solicitados os pontos por função, tipos de dados e registros. O enquadramento da complexidade será feito pelo próprio protótipo, e ainda permitirá incluir várias contagens para o mesmo projeto mensurado. O protótipo ainda terá como recursos:

- Relatórios das contagens;
- Gráficos comparativos das contagens de projetos.

Concluindo, o protótipo proposto é bem simples se comparado com o aplicativo estudado, porém o objetivo maior deste trabalho de conclusão é a capacitação dos colaboradores da Rech em contagem de pontos por função e a criação de sua própria ferramenta de mensuração de projetos, com maior aderência ao processo da empresa e com grande potencial de crescimento de recursos no futuro.

8.4 Projeto utilizado como *case*

O projeto da Rech que será utilizado como case e utilizado como teste do protótipo a ser criado será o projeto de melhoria do módulo de contabilidade. O projeto de melhoria é bastante extenso, já está sendo produzido pela empresa, porém para estudo deste trabalho de conclusão serão consideradas apenas duas funcionalidades previstas no projeto, que são as mais importantes do projeto de melhoria. Posso afirmar que são as mais importantes por que este projeto foi escrito e está sendo produzido por mim, com ajuda de outros dois colaboradores da Rech. As funcionalidades são:

1. Configuração da estrutura sintética do plano de contas contábil;
2. Amarração das partidas dos lançamentos contábeis.

A configuração da estrutura sintética do plano de contas contábil era uma necessidade há muito tempo requisitado pelos clientes. Até este projeto, o módulo de contabilidade do SIGER® não permitia que o usuário pudesse configurar a estrutura sintética, o sistema fornecia uma lista de opções de estrutura bem limitada. Constantemente surgiam novos clientes que não utilizava nenhuma das estruturas fornecidas, necessitando urgentemente de adaptação do sistema.

Estrutura sintética de um plano de contas é a organização das contas como no exemplo abaixo:

Estrutura sintética	Descrição da conta	Grau sintético da conta
1	Ativo	1º grau
1.1	Circulante	2º grau
1.1.1	Disponível	3º grau

1.1.1.01	Caixa	4º grau
----------	-------	---------

No exemplo acima, a conta 1.1.1.01 Caixa possui estrutura de 4 graus sintéticos, por que se contam os grupos de dígitos separados pelo ponto. A estrutura do plano de contas do exemplo acima é 9.9.9.99.

O objetivo de configurar a estrutura é para permitir que o usuário possa definir uma estrutura 9.99.999.9999, por exemplo. Para permitir esta funcionalidade, foi necessário fazer uma mudança estrutural da base de dados, passando por uma conversão de arquivos e então criar a opção de configuração de plano, no programa de manutenção do cadastro de empresas do módulo de contabilidade.

A outra funcionalidade, chamada pelos colaboradores da Rech como “amarração das partidas dos lançamentos contábeis”, também se fez necessária conversão da base de dados para permitir a sua implementação. No princípio básico de contabilidade, toda movimentação de uma empresa, com base em um documento idôneo como uma nota fiscal, deve-se fazer pelo menos dois lançamentos na contabilidade, um a débito e outro a crédito. A amarração está na facilidade do sistema conseguir identificar a partir de um lançamento, qual é o outro lançamento, chamado de “contra-partida” pelos contadores. No exemplo de partidas simples, uma partida a débito e outra a crédito do mesmo valor, não é complexo de identificar, mas existem casos em que possam ter lançamentos de “n” débitos para “n” créditos, tornando difícil para o usuário identificar no sistema as partidas neste caso, sem um campo que faça a amarração destas partidas. Este é o objetivo, em linhas gerais, desta segunda funcionalidade implementada no projeto.

CONCLUSÃO

A APF estudada neste trabalho de conclusão possui uma característica marcante que é a sua simplicidade, principalmente com relação ao cálculo dos pontos por função. Essa metodologia está entre as mais utilizadas para se estimar tamanho de projetos ou sistemas. A contribuição desta metodologia pode ser grandiosa e de valor inestimável para os processos da empresa se aplicada em todas as áreas possíveis, como gerência de projetos, estimativas de custo e produtividade e demais contribuições que a APF proporciona.

Estando o protótipo de APF proposto neste trabalho de conclusão produzido, este será imediatamente utilizado pela empresa Rech Informática, em caráter de testes. Estando o protótipo aprovado pelos colaboradores e pela direção da Rech, ele será agregado ao sistema de controle interno da empresa, o SICLA, o qual já possui uma parte destinada a controle de projetos da empresa.

A contribuição final deste trabalho de conclusão:

- Aprendizado dos colaboradores sobre a APF;
- Melhor gerenciamento dos projetos a partir de estimativas de tempo mais próximas do real;
- Melhor negociação com clientes ao estipular prazos das implementações.

Sugestões de melhoria no protótipo:

- Implementação de um plano mestre de produção de projetos. A partir de um calendário e horário de produção, montar um plano mestre de produção de projetos, baseando-se nas estimativas da APF, por colaborador ou equipe de colaboradores disponíveis;

- Implementar relatórios de análise de horas investidas em projetos de um período qualquer, permitindo fazer um controle de horas previstas e realizadas em projetos.

REFERÊNCIAS BIBLIOGRÁFICAS

INTERNATIONAL FUNCTION POINT USER GROUP. **Análise de pontos por função.** IFPUG, 1991. (Baseado na Release 3.4 do Manual de Práticas de Contagem do IFPUG).

VASQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado. **Análise de Pontos por Função. Medição, Estimativas e Gerenciamento de Projetos de Software.** São Paulo, 2005.

FIORINI, Soeli T.; STAA, Arndt von; BAPTISTA, Renan Martins. **Engenharia de software com CMM.** Rio de Janeiro, 1998.

FILHO, Wilson de Pádua Paula. **Engenharia de Software: Fundamentos, Métodos e Padrões.** Rio de Janeiro, 2003.

JOÃO, Belmiro do Nascimento. **Metodologias de Desenvolvimento de Sistemas.** São Paulo, 1999.

DEMARCO, Tom. **Controle de projetos de Software.** São Paulo, 1991.

JONES, Capers - **Estimating Software Costs** - McGraw-Hill, 1998.

HAZAN, Cláudia. **Análise de Pontos por Função: Uma Ferramenta na Implantação do Modelo CMM.** Brasília, 2003. Disponível em: <<http://www.serpro.gov.br/publicacao/tematec/2003/> Portal do Serpro - Análise de Pontos por Função Uma Ferramenta na Implantação do Modelo CMM.htm.

HAZAN, Cláudia. **Uma aplicação da APF nas estimativas de projetos web.** Disponível em: <http://www.serpro.gov.br>

AZEVEDO, Douglas José Peixoto de. **FPA – Function Point Analysis.** Disponível em: <http://www.pr.gov.br/batebyte/edicoes/1997/bb68/fpa.htm>, acesso em Agosto/2007.

MANUAL RECH. **Manual de integração Rech.** Novo Hamburgo, Março/2007.