

Análise de Pontos de Função: Um estudo de caso em uma empresa com MPS.BR nível F

Letícia de Cássia Santin¹, Jandira Guenka Palma¹

¹Departamento de Computação – Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – CEP 86057-970 – Londrina – PR – Brasil

leticia.santin@gmail.com, jandiragpalma@uel.br

Abstract. *The software development process involves high cost and also high technical efforts, which can be optimized by selecting techniques contained in Software Engineering. The results of these are highlighted and supported when there is a formalism Software Quality in the environment, in this context the metric of software becomes decisive as it helps performance monitoring as well as to assist in identifying areas for improvement. This work presents a case study of a small company that follows the formalism of the reference model MPS.BR, in order to improve the measurement process, the methodology to be applied is the Function Point Analysis.*

Resumo. *O Processo de desenvolvimento de software envolve alto custo e também elevados esforços técnicos, que podem ser otimizados com a seleção de técnicas contidas na Engenharia de Software. Os resultados destas, são evidenciadas e apoiadas quando há um formalismo de Qualidade de Software no ambiente, neste contexto a métrica de software torna-se fundamental visto que auxilia o monitoramento do desempenho assim como para apoio na identificação de pontos de melhoria. Este trabalho apresentará um estudo de caso de uma pequena empresa que segue o formalismo de qualidade de software do modelo de referência de MPS.BR, com a finalidade de melhorar o processo Medição, a metodologia a ser aplicada para esta melhoria será a Análise de Ponto de Função.*

1. Introdução

Diante da alta demanda de softwares para diversas áreas do conhecimento como: médica, imobiliária, telefônica, contábil, entre outras; da forte concorrência das empresas de software, ter qualidade no software produzido, entregar no prazo estipulado e a um custo competitivo são essências para uma empresa se manter no mercado. Existem normas e modelos internacionais para tratar questões de melhorias, mas a aplicação dessas envolvem um custo elevado para micro, pequenas e médias empresas. Como resolução desse problema, e para manter um padrão nos processos de software, o Ministério da Ciência e Tecnologia e a Associação para Promoção (SOFTEX), criaram o MPS.BR (Melhoria de Processo de Software Brasileiro), que é um Modelo de Processos de Softwares, dedicado ao mercado brasileiro. O modelo encontra-se dividido em níveis de maturidade, onde, a partir do segundo nível (F), um dos quesitos para a certificação da empresa, é possuir um procedimento para realização de medições nos processos [6]

Dessa maneira, as empresas podem utilizar metodologias de desenvolvimento de software que estejam amparadas por critérios de qualidade MPS.BR: onde são requeridas

a estimativa do custo e dos esforços nas atividades no processo desde a fase inicial até as fases de produção e testes; e o monitoramento dos mesmos para conferência da evolução do projeto com base nas estimativas realizadas, entre outros critérios. Para que isso ocorra, exige-se a instauração de um processo de medição e de estimativas de desenvolvimento de software.

As medições realizadas alimentam um histórico, onde a empresa terá uma base de dados para analisar os custos e os esforços da equipe de desenvolvimento. Há técnicas, para que estas medições sejam realizadas, como é o caso da Análise de Ponto de Função, Linhas de Código Fonte, Pontos de Caso de Uso, entre outras existentes no mercado. Este trabalho focará na Análise de Pontos de Função, pois é a técnica mais utilizadas em empresas nacionais, da área de desenvolvimento de software.

2. Fundamentação Teórica

Esta seção tratará de conceitos de métricas de software, sua inserção na Qualidade de Software – subárea da Engenharia de Software; a utilização dos modelos de processos de software e as técnicas para a realização da medição e da estimação de custos de software.

2.1. Qualidade de Software

Ao definir Qualidade de Software, [8] descreve a dificuldade de se obter tão descrição. Para ele, qualidade de software encontra-se em constante modificação e consegue ser “interminavelmente” debatida; sugere, assim, que a qualidade deve enfatizar-se em três pontos principais: a gestão da qualidade efetiva, a criação de produto que seja útil e a agregação de valor, tanto ao fabricante quanto ao usuário de um produto de software. Em [3] podemos observar que, há mais de duas décadas, já existiam controvérsias sobre como referenciar a qualidade de software, onde mostra que deve-se incluir, para obtenção na qualidade de software: adequação à proposta, conformidade com a especificação, grau de excelência e linha do tempo. Atualmente, existem alguns modelos para gerenciamento dos processos de software, que oferecem suporte a qualidade – nos pontos citados por [3] – no cotidiano das empresas, como é o caso do CMMI, SPICE, MPS, onde, este é um dos focos deste trabalho pelo motivo de a empresa utilizada no estudo de caso já utilizar esse modelo de maturidade.

2.2. MPS.BR

O MPS.BR é desenvolvido pela SOFTEX desde 1996, com apoio do Ministério da Ciência. É um modelo de qualidade de processo, focado na realidade do mercado de pequenas e médias empresas de desenvolvimento de software do Brasil. É baseado nas normas ISO/IEC 12207 (trata do processo de ciclo de vida¹ do software) e ISO/IEC 15504 e compatível com o CMMI – que é um modelo de referência internacional [6].

Uma das metas do programa MPS.BR é definir e aprimorar um modelo de melhoria e avaliação de processos de software e serviços, visando, preferencialmente, às micros, pequenas e médias empresas. O modelo é constituído de três componentes: O Modelo de referência para melhoria do processo de software (MR-MPS), que define os processos

¹Ciclo de Vida de Software: Basicamente, se refere a ordem global das atividades envolvidas, inseridas em um contexto de projeto de software.

e atributos a serem atendidos pelas organizações; O Método de avaliação para melhoria do processo de software (MA-MPS), que descreve os métodos para o avaliador definir em qual nível de maturidade a empresa se encontra; e o Modelo de negócio para melhoria do processo de software (MN-MPS), onde encontra-se as regras de negócio utilizadas para implementação do MR-MPS pelas instituições implementadoras e do MA-MPS pelas instituições avaliadoras.

Os níveis de maturidade do MPS.BR, inicia no nível G, conhecido por Parcialmente Gerenciado, e segue até o mais alto nível de maturidade: o nível A (Em Otimização), no decorrer dos processos de certificação dos níveis, a organização passa a adaptar-se com novos quesitos de qualidade ou a evolução de quesitos já existentes (isso ocorre a partir do nível E). No nível F de maturação existe o quesito para implantação de Medições (MED) nos processos de software, em que é proposto para a empresa possuir um processo de coleta, armazenagem, análise e relatório dos dados relativos aos produtos em desenvolvimento e aos processos que são implementados na organização e em seus projetos.

2.3. Métricas de Software

Uma das abordagens utilizadas para garantir a qualidade de software é o uso de métricas, que são indicadores de qualidade de um projeto, processo ou do produto em si [8]. As medições são divididas em três fases, com dependência entre elas [7]:

- **Fase cognitiva:** Envolve julgamento e experiência, nessa fase é utilizada um processo mental de medição, como, por exemplo, um gerente de software experiente determina quanto tempo levará para a realização de uma tarefa;
- **Fase semântica:** Envolve a rotulação ou referência as observações, ou seja, a formalização da fase cognitiva. Em um sistema, a criação de padrões de nomes, para melhor entendimento dos componentes da equipe, é caracterizado como uma fase semântica.
- **Fase quantitativa:** Processo mais refinado e preciso de medição, são utilizados números para representar dados e níveis de dificuldade, permite a construção de equações numéricas, o que tornam a medição menos subjetivas – o que é o caso da APF

O processo de medição auxilia para a criação de um histórico, onde é possível armazenar e consultar dados que possibilitam estimativas, por exemplo de produtividade, complexidade, esforço e velocidade de execução. As métricas de software podem ser divididas em duas categorias: pelo ponto de vista da aplicação, onde subdividi-se em métricas de produtividade e métricas de qualidade.

As métricas de produtividade se concentram na saída do processo de engenharia de software e métricas de qualidade indicam o quanto o software atende aos requisitos definidos pelo usuário. E sob o ponto de vista de medição, em que possui duas subdivisões: medidas diretas (quantitativas) e indiretas (qualitativas) [8] e [4], onde a diferença entre elas é, sobretudo, que a medida direta envolve apenas uma variável, enquanto medida indireta envolve uma n-tupla em seu domínio [4].

A norma IEEE 1061, define medida direta como métrica que não precisa de outra mensuração ou de qualquer outro atributo; como, por exemplo, altura, peso, horas trabalhadas, LOC etc. enquanto que medida indireta são as medidas extraídas com a utilização

de uma ou mais medidas diretas que podem estar combinadas com constantes [1], um exemplo de medida indireta é o índice de massa corporal: $mc = peso/altura^2$, se tratando de software, são consideradas medidas indiretas a eficiência, complexidade e produtividade do software.

Diversas são as métricas existentes e as suas aplicações no ciclo de vida de um software, cabe ao coordenador do projeto gerenciar as ações para chegar ao padrão esperado de qualidade e determinar quais pontos devem ser medidos e monitorados no decorrer do ciclo de vida do projeto. A coleta desses dados permite uma análise qualitativa desse software como um produto. A base histórica das métricas permite que futuras propostas de mudança ou criação sejam mais precisas, visto que projetos similares tendem a passar pelos mesmos problemas e soluções [4]. Segundo [9], essas estimativas são essenciais para a avaliação do projeto e para estimar se as melhorias, tanto do projeto como da tecnologia, são produtivos. As estimativas são baseadas em medir alguns elementos do software, onde o resultado é dividido pelo esforço total, demandado pelo desenvolvimento. As medidas relacionadas a tamanho e as medidas relacionadas a funções do sistema, são os dois tipos que têm sido largamente utilizados.

2.4. Técnicas de Medição e estimativa

2.4.1. Análise Orientada a Tamanho

Métricas de software orientadas a tamanho são realizadas levando em consideração o número de linhas de códigos de uma aplicação, Para [2] “Linha de Código é o número de linhas do código fonte do software entregue, excluindo comentários e linhas em branco”, mas, em [9] encontramos que também pode ser utilizadas são o número de instruções de código-objeto entregues ou o número de páginas de documentação do sistema. Essas métricas não são aceitas como a melhor forma de metrificação de software, por possuir variadas definições para a medição. Há muitas controvérsias a respeito do uso de linhas de códigos como medida principal. Em [8] encontra-se que essa técnica de medição é dependente da linguagem a ser utilizada e que a produtividade não consegue ser muito bem estimada.

2.4.2. Análise Orientada a Função

A Análise Orientada a Função é técnica mais comumente utilizada na indústria, pois podem ser computadas sem o conhecimento do modelo ou tecnologia. Esse método foi proposto por Allan J. Albrecht em 1979 e otimizado por ele mesmo e John E. Gaffney Jr. em 1983 [9]. A unidade de medida utilizada é pontos de função (PF) e é realizada, dependendo da metodologia, em até três etapas. A primeira, é a contagem de pontos de função não ajustados, após é realizada o ponderamento do sistema com 14 questões a respeito do sistema e, por último, o ajustamento dos pontos de função [10].

Como a análise de função é o tópico principal neste trabalho, a seção seguinte é totalmente dedicada a técnica de contagem de pontos de função.

2.4.3. Análise Orientada a Caso de Uso

A técnica de estimativa por Pontos de Caso de Uso foi proposta em 1993 por Gustav Karner, da *Objectory* (hoje, *Rational Software*). Ela baseia-se em dois métodos bastante utilizados: o mecanismo de Pontos de Função e uma metodologia conhecida como Mk II, uma adaptação da técnica de PFs, bastante utilizada na Inglaterra. A forma de lançar uma estimativa é o principal diferencial da métrica por Casos de Uso: o método trata de estimar o tamanho de um sistema de acordo com o modo como os usuários o utilizarão, a complexidade de ações requerida para cada tipo de usuário e uma análise em alto nível dos passos necessários para a realização de cada tarefa, em um nível muito mais abstrato que a técnica de Pontos de Função.

2.5. Contagem de Pontos de Função

Com base em [5], as fases para o cálculo de PF é realizado como mostra na figura 1:

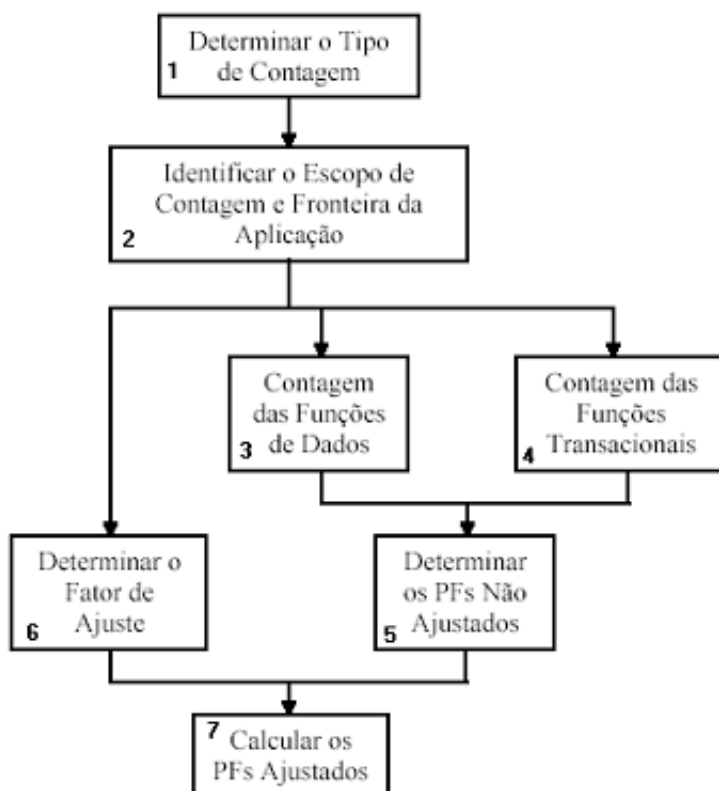


Figure 1. Fluxograma com as fases da APF

1. Determinar o Tipo de Contagem: 1) A contagem de PFs podem ser realizadas tanto em projeto em desenvolvimento quanto na aplicação. Existem três tipos de contagem de pontos de função:
 - Projeto de desenvolvimento: Realizada já na aplicação pronta, medindo as funções oferecidas ao usuário final.
 - Projeto de Melhoria: Realizada para medir as modificações na aplicação existente que incluem, alteram ou excluem funções.

- **Aplicação:** Está associada à aplicação instalada. Ela é também referenciada como uma contagem de pontos de função da baseline ou instalada. Esta contagem fornece uma medida das funções que a aplicação oferece atualmente ao usuário.
2. **Identificar o Escopo de Contagem e Fronteira da Aplicação:** Nessa fase, são realizadas as definições das funcionalidades a serem incluídas em uma contagem de PFs específica. A fronteira da aplicação é estabelecida como: um limite lógico entre a aplicação que está sendo medida, o usuário e outras aplicações. O escopo de contagem deve ser definido a parte do sistema (funcionalidades) a ser contada, como, por exemplo, na figura 2, onde a aplicação a ser contada é o sistema de RH e possui interfaces com outros sistemas e com o usuário.

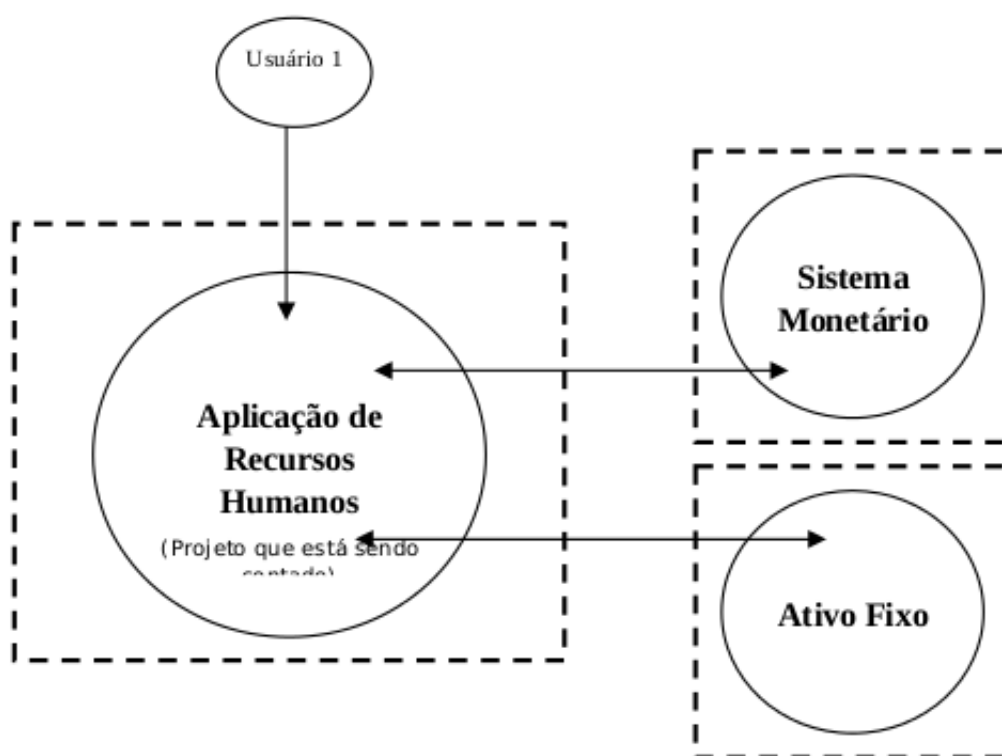


Figure 2. Exemplo de Definição de Fronteira de Aplicação

3. **Contagem das Funções de Dados:** As funções de dados representam as funcionalidades relativas aos requisitos de dados internos e externos à aplicação. São elas os arquivos lógicos internos (ALI) e os arquivos de interface externa (AIE). A única distinção entre os dois arquivos é que o ALI são grupos de arquivos que são mantidos dentro da fronteira de aplicação, enquanto o AIE são mantidos por outra aplicação, ou seja, um AIE de uma aplicação deve, necessariamente, ser um ALI em outra.
4. **4) Contagem das Funções Transacionais:** Funcionalidades de processamento de dados do sistema fornecidas para o usuário. São elas: as entradas externas, as saídas externas e as consultas externas. As Entradas Externas (EEs) são processos

elementares² que processam dados (ou informações de controle) que entram pela fronteira da aplicação. As Saídas Externas (SEs) são processos elementares que enviam dados (ou informações de controle) para fora da fronteira da aplicação. Por fim, uma Consulta Externa (CE), assim como uma SE, é um processo elementar que envia dados (ou informações de controle) para fora da fronteira da aplicação. Uma SE e uma CE possuem o mesmo objetivo: apresentar informação para o usuário, diferenciando por meio apenas da forma de recuperação, o CE visa a recuperação das informações de forma simples, enquanto uma SE utiliza-se de cálculos ou cruzamento de informações pra obtê-las.

Table 1. Exemplo de Definição de Fronteira de Aplicação

Número de Arquivos Referenciados	Número de Itens de Dados Referenciados		
	De 1 a 5	De 6 a 19	20 ou mais
0 ou 1	Simple	Simple	Média
2 ou 3	Simple	Média	Complexa
4 ou mais	Média	Complexa	Complexa

Table 2. Exemplo de Definição de Fronteira de Aplicação

Função	Complexidade		
	Simple	Média	Complexa
ALI	7	10	15
AIE	5	7	10
EE	3	4	6
SE	4	5	7
CE	3	4	6

5. Determinar os PFs Não Ajustados: Após a contagem das funções de dados e de transição, o próximo passo é determina a contribuição de cada um para a aplicação. Nesse ponto é utilizado regras, feitas pela IFPUG, para definir a contribuição e complexidade de cada função, onde a complexidade é dependente da quantidade de Arquivos Referenciados com o a quantidade de itens de dados referenciados, como mostrado na tabela 1. Em seguida ao cálculo de complexidade, utiliza-se outra regra, onde é possível definir a contribuição, em PFs, das funções calculadas (tabela 2), como, por exemplo, uma EE que tenha complexidade média, resulta em 4 PFs não ajustados da aplicação.
6. Determinar o Fator de Ajuste: está baseado em 14 características gerais do sistema onde classifica as funcionalidades gerais da aplicação que está sendo contada. Cada característica tem descrições associadas que ajudam a determinar o nível de influência da característica, que varia em uma escala de 0 a 5 (de sem influência até forte influência). Quando aplicado, o fator de ajuste ajusta a contagem de pontos de função não- ajustados em +/- 35 para produzir a contagem de pontos de função ajustados. As características gerais do sistema são:
 - (a) Comunicação de Dados
 - (b) Processamento de Dados Distribuídos

²Processo Elementar: menor unidade de atividade significativa para o usuário. Deve ser completo e deixar o negócio da aplicação sendo contada em estado consistente [10]

- (c) Desempenho
- (d) Utilização do Equipamento (Restrições de recursos computacionais)
- (e) Volume de Transações
- (f) Entrada de Dados On-line
- (g) Eficiência do Usuário Final (Usabilidade)
- (h) Atualização On-line
- (i) Processamento Complexo
- (j) Reusabilidade
- (k) Facilidade de Implantação
- (l) Facilidade Operacional
- (m) Múltiplos Locais e Organizações do Usuário
- (n) Facilidade de Mudanças (Manutenibilidade)

Finalmente o valor de Ajuste (VFA) é definido pela função $VFA = (NIT * 0,01) + 0,65$, onde NIT = nível de influência total.

7. Calcular os PFs ajustados: No último passo da contagem de PF's, para cada tipo de contagem (projeto de desenvolvimento, melhoria e aplicação) é utilizado a equação específica para o tipo de contagem.

3. Objetivos

O objetivo deste trabalho consiste no estudo e aplicação de Análise de Pontos de Função em uma empresa com MPS.BR nível F, Resultando na otimização do processo de estimação de software.

4. Procedimentos metodológicos/Métodos e técnicas

Para a realização deste trabalho, inicialmente, executará um levantamento bibliográfico de artigos, livros, e conferências recentes publicados na área, que contenham estudos sobre métricas de softwares e uso de Análise de Pontos de Função. Após, o levantamento bibliográfico, realizará o estudo dos métodos utilizados para a contagem de PFs, para obtenção de conhecimento teórico sobre o assunto.

Finalizada a fase de capacitação no assunto proposto, será realizada uma entrevista na empresa escolhida para estudo de caso, e a aquisição de requisitos de software, de projetos finalizados, para computação dos Pontos de Função. Em seguida será realizado a procura por um padrão de conversão entre os PF's e as horas trabalhadas no projeto, gerando dados de conversão para futuras estimativas.

Serão realizadas reuniões com os coordenadores de área da empresa, para difusão da APF, e identificação das melhores práticas a serem utilizadas no ambiente da empresa, para que o impacto da APF seja o mínimo possível no cotidiano da empresa. Assim, o procedimento para realização deste trabalho segue abaixo:

Atividades:

1. Levantamento Bibliográfico: Levantamento de referências bibliográficas de livros e artigos científicos de trabalhos já desenvolvidos e surveys sobre métricas de software, estudos de casos já realizados na aplicação de APF em projetos de desenvolvimento. Estudo do MPS.BR, mais precisamente nos quesitos de métricas de software;

2. Estudo de métodos: Estudo dos métodos mais utilizados para a contagem de Pontos de Função. Para ampliação do conhecimento no assunto;
3. Entrevista na empresa: Entrevista na empresa Guenka. Primeiramente para conhecer o ambiente de trabalho, e, posteriormente, fazer validações e apurações no decorrer do processo de aplicação da APF na empresa
4. Análise: Utilização da APF na empresa para estimativas de software;
5. Redação do TCC;
6. Preparação para apresentação do TCC a banca examinadora.

5. Cronograma de Execução

Tabela 3. Cronograma de Execução

	fev	mar	abr	mai	jun	jul	ago	set	out	nov
Atividade 1	X	X								
Atividade 2		X								
Atividade 3			X	X	X	X	X			
Atividade 4								X	X	
Atividade 5		X	X	X	X	X	X	X	X	
Atividade 6										X

6. Contribuições e/ou Resultados esperados

Ao final deste trabalho é esperado o desenvolvimento de um método para a utilização da Análise de Pontos de Função na empresa escolhida pra estudo de caso, em que cause o menor impacto possível aos processos já instaurados na empresa e facilite o processo de estimativa e avaliação do tempo que será gasto para o desenvolvimento de softwares. Esta pesquisa contribui para o desenvolvimento da capacidade de análise, devido ao caráter prático que ela oferece, e inicialização profissional na área de qualidade de software. Espera-se que a empresa adote a metodologia proposta, possibilitando uma melhora, significativa, nos processos de estimativa de softwares, e, a longo prazo, a estimativas sejam mais precisas, para um melhor gerenciamento do tempo e de recursos.

7. Espaço para assinaturas

Londrina, 31 de março de 2014.

Aluna

Orientadora

References

- [1] *IEEE Standard for a Software Quality Metrics Methodology (IEEE Std 1061)*, 1998.
- [2] Syed Ali Abbas, Xiaofeng Liao, Aqeel Ur Rehman, Afshan Azam, and Abdullah M. I. Cost estimation: A survey of well-known historic cost estimation techniques. *Journal of Emerging Trends in Computing and Information Sciences*, 3(4):612–636, Abril 2012.
- [3] Norman E. Fenton and Shari L. Pfleeger. *Software Metrics: A Rigorous & Practical Approach*. PWS, 2 edition, 1997.
- [4] Cem Kaner and Walter P. Bond. Software engineering metrics: What do they measure and how do we know? In *International Software Metrics Symposium*, page 12, 2004.
- [5] MPS.BR - Melhoria de Processo do Software Brasileiro. *Manual de Práticas de Contagem de Pontos de Função*, 2004.
- [6] MPS.BR - Melhoria de Processo do Software Brasileiro. *Guia Geral MPS de Software*, Junho 2011.
- [7] C. Ravindranath Pandian. *Software Metrics: a guide to planning, analysis, and application*. Auerbach Publications, 2004.
- [8] Roger S. Pressman. *Engenharia de Software - Uma Abordagem Profissional*. McGrawHill, 7 edition, 2011.
- [9] Ian Sommerville. *Engenharia de Software*. PEARSON EDUCATION - BR, 9 edition, 2011.
- [10] Carlos E. Vazquez, Guilherme S. Simões, and Renato M. Albert. *Análise de Pontos de Função: Medição, estimativas e gerenciamento de projetos de software*. Érica, 2013.