

Utilizando métricas para dimensionar um software.

Entenda como funcionam as Métricas de Software, como e quando devem ser utilizadas, e qual a real necessidade do uso desta técnica da Engenharia de Software.

Conheça os métodos utilizados para realizar a medição de um software, como a Análise de Pontos por Função e Pontos por Caso de Uso.

ARTUR COSTA, DIEGO HENRIQUE, GLEIDSON SÉRGIO, LUCAS LISBOA, RICK ANDRADE

Segundo o IEEE, métrica é uma medida quantitativa do grau com que o sistema, componente ou processo possui um dado atributo. Em outras palavras, através de outras definições relacionadas, também formuladas pelo IEEE, podemos separar as métricas em alguns aspectos:

- Atributo - Propriedade física ou abstrata mensurável de uma entidade.
- Medição - O ato ou processo de atribuir um número ou uma categoria a uma entidade para descrever aquela entidade.
- Medida - Um número, extensão ou quantidade que resulta de uma medição.
- Medir - Aplicar uma métrica, ou atribuir valor por comparação com uma norma.

Medições requerem entidades (objetos de interesse), atributos (características das entidades) e regras ou escalas para atribuir valores (medidas) aos atributos. Na prática, métrica e medida tem o mesmo significado.

As métricas se dividem em dois principais tipos: Diretas e Indiretas. Uma métrica direta (básica) não depende da medição de nenhum outro atributo, como as métricas de esforço e tamanho funcional. Já a indireta (derivada) é calculada por meio de combinações de outras métricas, como a métrica de produtividade que depende das anteriores. Outras classificações trazem o conceito de métrica objetiva, independente do autor da medição. Enquanto a subjetiva depende do julgamento humano. Como exemplo tem-se: contar defeitos é objetivo, classificar o nível do defeito é subjetivo.

Ao se calcular métricas, podemos aperfeiçoar uma das tarefas mais importantes da Gerência de Projetos que é o planejamento. Com a utilização das métricas pode-se identificar o esforço, custo e atividades necessárias para a realização de um projeto.

Segundo Wilson de Pádua (PÁDUA, 2009), os principais objetivos das métricas de software e seu gerenciamento são:

- Aumentar a exatidão das previsões sobre custos e prazos dos projetos;
- Reduzir os defeitos e outros problemas do produto;
- Reduzir os custos e prazos do projeto;
- Aumentar o retorno do investimento da organização em processos, tecnologia e treinamento;
- Identificar necessidades de novos investimentos em processos, tecnologia e treinamento.

Ao analisar os tópicos acima, pode-se entender melhor alguns aspectos que estão envolvidos à utilização de métricas, como o comportamento e o funcionamento de produtos de software, determinação de padrões, metas e critérios de aceitação, controle de processos, produtos e serviços de software e previsibilidade de valores dos atributos.

O objetivo deste artigo consiste em explicar os fundamentos e funcionamento das métricas de software, abordando como exemplo a Análise por ponto de Função e os Pontos por Caso de Uso. Visando o melhor entendimento do assunto tem-se ainda a exposição de uma situação real que envolve uma empresa bem sucedida que utiliza métricas de software.

Porque devemos medir um software?

Em diversas áreas da Engenharia a medição é algo extremamente importante e comum, seja para definir o tamanho de uma planta, a distância de um lugar para outro, entre outros exemplos. Para isto, são estabelecidos padrões que são, em sua maioria, amplamente aceitos pela comunidade mundial que desenvolve trabalhos ligados a estas áreas. Porém, na área computacional, mais especificamente na Engenharia de Software, essa definição de um padrão para estabelecer medidas de um software é bastante complicada.

Essa dificuldade vem do fato da difícil concordância sobre o que medir em um software, quais critérios analisar, como definir padrões de avaliação, entre outras diversas divergências de pensamentos. Visando diminuir essas diferenças foi criado o que é chamado de Métricas de Software.

Métricas de Software é uma área de estudo que visa medir atributos de uma determinada entidade. Ou seja, consiste numa técnica para medir propriedades ou características de produtos, processos ou recursos. O tema “Métricas de Software” é alvo de discussão no ramo da Engenharia de Software há mais de 20 anos, porém na prática a técnica é pouco utilizada nos projetos de desenvolvimento de um software.

Sua utilização tem por objetivo aumentar o grau de gerenciamento de um projeto de software, uma vez que, traz com maior exatidão previsões sobre custo e prazos do projeto. Além disso,

visa aumentar e identificar necessidades de investimentos dentro da organização e a possibilidade de redução de custos e prazos do projeto.

Mas, porque devemos realmente medir um software durante o seu desenvolvimento? Essa é uma das grandes dúvidas que envolvem o tema dentro de organizações. E, muitas vezes, por não entenderem a real necessidade, acabam deixando de lado essa importante técnica de gerenciamento de custos, esforços, recursos, entre outros. Deve-se medir um software, principalmente, porque através da medição pode-se entender melhor e aperfeiçoar o processo de desenvolvimento e gestão, inclusive melhorando a relação com os clientes. Outros pontos importantes ligados à medição de um software estão relacionados com a melhoria na gerência dos contratos da empresa, nas melhores avaliações ligadas a produtividade do processo como um todo, aos benefícios da utilização de métodos de trabalho e novas ferramentas e ao retorno de investimento estimado para um determinado projeto.

No processo de utilização de Métricas de Software deve-se garantir algumas propriedades para que as mesmas sejam utilizada da maneira mais eficiente possível. As métricas devem ser facilmente entendidas, calculadas e testadas, devem gerar uma sugestão de melhoria nas estratégias, bem como é de extrema importância que a métrica seja obtida o mais cedo possível no processo de desenvolvimento de um software, visando obter o máximo de vantagens que ela proporcione.

As Métricas de Software devem ser a rigor práticas, com baixo custo e confiáveis. Elas se dividem basicamente em diversos tipos, dentre eles temos: Métricas Diretas, Métricas Indiretas, Métricas Orientadas a Tamanho, Métricas Orientadas por Função, Métricas de Produtividade, Métricas de Qualidade e Métricas Técnicas.

As Métricas Diretas, ou ainda, fundamentais ou básicas, são métricas realizadas em termos de atributos observados, ou seja, é determinada pela contagem. Como exemplo, tem-se: esforço, custo, linhas de código, número de páginas e diagramas. Já as Métricas Indiretas, ou derivadas, relacionam-se com medidas geradas através de outras métricas. Por exemplo: eficiência, confiabilidade, facilidade de manutenção de um código. Por sua vez, as Métricas Orientadas a Tamanho trazem consigo medidas diretas do tamanho dos softwares.

As Métricas Orientadas por Função consistem em medir o software através da visão do usuário final. Já as Métricas de produtividade relacionam-se com o processo da Engenharia de Software, como por exemplo, com a utilização dos números de casos de uso. As Métricas de Qualidade trazem uma melhor indicação de como o software se adequa às necessidades do cliente. As Métricas Técnicas visam a obtenção de características específicas do software tirando o foco do processo pelo qual o mesmo foi projetado e desenvolvido.

Existem dois métodos importantes no processo de medição de um software. São eles: Análise de Pontos por Função e Pontos por Caso de Uso. A seguir, aprofundaremos os estudos destes dois métodos, visando entender melhor como funciona o processo de medição de um software.

Métricas de Software

As Métricas de Software correspondem a técnicas para medição de um software. A seguir serão abordados dois dos principais métodos utilizados para se dimensionar um software. São eles: Análise por Ponto de Função (APF) e Pontos por caso de Uso (UCP).

Análise de Pontos por Função

A Análise de Pontos por Função (APF) consiste em uma importante técnica para a medição de projetos de software. Seu papel é estabelecer uma medida através dos Pontos de Função, observando as funcionalidades que o software possui através do ponto de vista do usuário. É importante destacar que a utilização do APF é totalmente independente de linguagens de programação e de tecnologias utilizadas em projetos. O objetivo dessa análise relaciona-se justamente com a medida de projetos desvinculando-os das tecnologias utilizadas. Outro ponto a ser destacado é a medição da funcionalidade requisitada pelo cliente, o que auxilia na tomada de decisões, relacionadas com custo, prazo, quantidade de recursos alocados, melhoria no cronograma e etc.

Os cálculos dos pontos por função devem seguir o Manual de Práticas de Contagem do IFPUG (*International Function Point Users Group*), que nada mais é que um grupo internacional que estabelece padrões e regras para a utilização da análise dos pontos por função. O manual traz operações básicas como funções dos tipos dados, transação, itens de influência, ambientes, recursos e equipes.

O processo de contagem dos pontos de função (PF) é composto por três fases:

- Componentes considerados, que envolvem a contagem dos seguintes componentes: Arquivos Lógicos Internos, Arquivos de Interface Externa, Entradas Externas, Consultas Externas e Saídas Externas.
- Etapas do processo de contagem, que trazem os seguintes aspectos: Identificação, Complexidade e Contribuição, Pontos de Função Não Ajustados e Pontos de Função Ajustados.
- Aplicação da contagem, que envolve os Pontos de Função de um Projeto de Sistema, de Alterações em Sistemas e de um Sistema. O primeiro significa a análise de um projeto como um todo, desde suas modificações à entrega final. O segundo relaciona-se com a inclusão na contagem dos componentes novos, alterados e, possivelmente, excluídos. Por fim, o terceiro relaciona-se exclusivamente com o sistema entregue ao usuário, medindo o tamanho do sistema.

Pontos por Caso de Uso

O método *Use Case Points* (UCP) foi elaborado por Gustav Karner no ano de 1993 tendo como base o APF (Análise por Ponto de Função). Esse tipo de métrica procura avaliar o sistema através da análise da forma como os usuários utilizarão o sistema, a complexidade das tarefas requisitadas, bem como um estudo dos passos para efetuar cada atividade, determinando a eficiência do projeto.

O método UCP tem como característica a possibilidade de mensurar o sistema na fase de levantamento de requisitos do projeto utilizando os documentos feitos na mesma como base para fazer o cálculo dimensional.

Os passos para fazer a medição nesse método são:

1. Calcular o **UUCP** (*Unadjusted Use Case Point*)

– Relacionar os atores de acordo com nível de complexidade representado na tabela 1 abaixo:

Complexidade	Peso	Definição
Simple	1	Quando o ator representa um ambiente externo acessado por uma API (<i>Application Programming Interface</i>).
Médio	2	Quando o ator representa um sistema externo, acessado através de um protocolo de comunicação (por exemplo: TCP/IP).
Complexo	3	Quando o ator interage com o sistema através de uma interface gráfica (GUI).

Tabela 1. Pesos de Atores

- Relacionar e classificar os casos de usos de acordo com a complexidade da tabela 2 abaixo:

Complexidade	Peso	Definição
Simple	5	Quando o Caso de Uso possui 3 (três) ou menos transações, incluindo cenários alternativos e sua realização deve acontecer com menos de 5 objetos (classes de análise).
Médio	10	Quando o Caso de Uso possui de 4 a 7 transações, incluindo cenários alternativos e sua realização deve acontecer com 5 a 10 objetos (classes de análise)
Complexo	15	Quando o Caso de Uso possui mais de 7 transações, incluindo cenários alternativos, e sua realização deve acontecer com mais de 10 objetos (classes de análise).

Tabela 2. Peso de Casos de Uso

- Para calcular o **UUCP** faz-se:

UUCP = Total de peso do atores relacionados + Total de peso dos casos de uso relacionados

2. Calcular o TCF (*Technical Complexity Factor*)

Para cada requisito listado na tabela 3 abaixo deve ser atribuído um valor que determina a influência do requisito no sistema, variando entre 0 e 5:

Fator	Requisito	Peso
T1	Sistema distribuído	2
T2	Tempo de resposta	2
T3	Eficiência	1
T4	Processamento complexo	1
T5	Código reusável	1
T6	Facilidade de instalação	0.5
T7	Facilidade de uso	0.5
T8	Portabilidade	2
T9	Facilidade de mudança	1
T10	Concorrência	1
T11	Recursos de segurança	1
T12	Acessível por terceiros	1
T13	Requer treinamento especial	1

Tabela 3. Peso de Fatores Técnicos

- O cálculo do **Tfactor** é realizado fazendo o somatório dos fatores identificados (T1 a T13) do sistema a ser feito multiplicados pelo respectivo peso;

- Calcular o fator de complexidade de acordo com a fórmula: $TCF = 0,6 + (0,01 * Tfactor)$

3. Calcular o **Efactor** (*Environmental Factor*)

- Para cada requisito da tabela 4 atribui-se um valor que determina a influência do requisito no sistema variando de 0 a 5:

Fator	Descrição	Peso
E1	Familiaridade com RUP ou outro processo formal	1.5
E2	Experiência com a aplicação em desenvolvimento	0.5

E3	Experiência em Orientação a Objetos	1
E4	Presença de analista experiente	0.5
E5	Motivação	1
E6	Requisitos estáveis	2
E7	Desenvolvedores em meio-expediente	-1
E8	Linguagem de programação difícil	2

Tabela 4. Pesos de Fatores Ambientais

- O calculo do **EFactor** é realizado fazendo o somatório dos fatores identificados (E1 a E8) do sistema a ser feito multiplicados pelo respectivo peso;

- Calcular o fator de complexidade de acordo com a fórmula: **ECF = 1.4 + (0.03 * Efactor)**

4. Calcular o **UCP (Use Case Points)**

Finalmente calcular o UCP utilizando os resultados dos fatores obtidos acima com a fórmula:

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{ECF}$$

5. Estimar projeto em horas

Karner sugere a aplicação de 20 horas/homem por ponto de UCP:

$$\text{Estimativa (horas)} = \text{UCP} * 20$$

A estimativa do custo de desenvolvimento é obtida pela multiplicação do número de casos de uso estimado, pelo valor médio da hora de desenvolvimento.

Para cada empresa que desenvolve software, estes valores são ajustados em função do que realmente ocorreu nos projetos já terminados e entregues ao usuário e com o tempo pode-se chegar a estimativas da proporcionalidade do envolvimento de programadores e analistas no projeto, fazendo-se cálculos mais realistas onde deve ser somado todos os aspectos inerentes ao projeto, que incluem custo de treinamento, custo homem/hora, custo de software de apoio, custo de desenvolvimento.

Depois do que foi explanado, pode-se observar que o método Pontos por Caso de Uso é útil para realizar estimativas em curto prazo, porém é mais apresentada para controle de qualidade em detrimento a cálculo de tamanho.

Experiência Prática

Visando obter uma experiência prática sobre Métricas de Software entrevistamos uma empresa de pequeno porte do ramo de desenvolvimento de sistemas web em Salvador, a Angulare

Negócios Web LTDA (<http://www.angular.com.br/>). A Angular tem 4 (quatro) anos de existência e possui uma equipe de 6 (seis) funcionários.

Durante a entrevista com o senhor Rodrigo Jerry (rodrigo@angular.com.br) percebemos que sua empresa, como a maioria em Salvador, não utiliza nenhum tipo de Métrica de Software. Existe apenas a tentativa de se estimar prazos, que é feita baseada totalmente no empirismo e na experiência de cada profissional.

O principal problema relatado decorrente da falta de utilização de métricas é o fato do profissional abandonar o projeto porque não soube avaliar direito o custo do software, causando grandes problemas para quem o contratou.

Foi feita a sugestão de se utilizar Pontos por Caso de Uso, pois essa métrica possui boa metodologia de estimativa de tempo. A sugestão foi recusada com a justificativa de que "é impossível implantar um sistema de métricas em uma equipe muito heterogênea e não contratada". A afirmação não tem fundamento, pois a medição do software utilizando UCP pode ser feita durante as reuniões de levantamento de requisitos.

Conclusão

Atualmente as métricas existentes são modificadas e novos modelos são criados especificamente para melhorar a precisão da medição de softwares. Por isso, e pela resistência cultural encontrada nas organizações, a utilização de métricas de software em um projeto enfrenta algumas dificuldades dentro das empresas.

As atividades de medição devem sempre ser guiadas por objetivos, pois os planos de métricas detalham como criar programas de medição visando atender objetivos técnicos específicos. Métricas mal-utilizadas e dados mal interpretados comprometem os resultados das medições. É difícil estabelecer medições apropriadas e úteis, e mais difícil ainda é manter.

É possível contornar estas dificuldades. Precisa-se manter o foco desde os estágios iniciais da melhoria de processo, começar pequeno, selecionar um conjunto coerente e descartar o que não for útil.

É importante definir cada detalhe da métrica. Não adianta criar uma métrica se a equipe de desenvolvimento não fizer uso dela. É preciso treinar e incentivar a equipe a usar a métrica. Todas as pessoas afetadas devem ser envolvidas. Deve-se compreender que a adoção da métrica leva tempo.

Após esta explanação sobre as Métricas de Software podemos concluir que o artigo visa desmitificar, esclarecendo aos leitores, o que é, para que servem e quando deve-se utilizar Métricas de Software para medir um determinado software. Com sua utilização da maneira ideal, pode-se garantir que as mesmas tenham um elevado grau de eficiência dentro das organizações, auxiliando no planejamento de um projeto, bem como no melhor gerenciamento e entendimento do mesmo.



Augusto Artur Costa (arturcostta@gmail.com) é graduando em Sistemas de Informação pela Universidade Estadual da Bahia (UNEB). Tem interesse em desenvolvimento de software e atualmente faz estágio na TecnoTRENDS com programação web.



Diego Henrique (diegoh2222@gmail.com) é graduando em Sistemas de Informação pela Universidade Estadual da Bahia (UNEB) e estagiário do Laboratório de Automação e Controle Industrial da Universidade Estadual da Bahia (UNEB).

Gleudson Sérgio (gleidsonsergio@hotmail.com) é graduando em Sistemas de Informação pela Universidade Estadual da Bahia (UNEB).



Lucas Lisboa (lucas.lisboa.santos@gmail.com) é graduando em Sistemas de Informação pela Universidade Estadual da Bahia (UNEB) e estagiário das Comunidades Virtuais (UNEB).



Rick Andrade (rickandrade89@gmail.com) é graduando em Sistemas de Informação pela Universidade Estadual da Bahia (UNEB) e estagiário na PRODEB - Cia de Processamento de Dados do Estado da Bahia.

Referências Bibliográficas

PAULA FILHO, Wilson de Pádua. Engenharia de Software: Fundamentos, Métodos e Padrões. 3ª Edição, Rio de Janeiro-RJ. LTC, 2009.

Introdução a Métricas de Software, disponível em www.cin.ufpe.br/~if720/slides/introducao-a-metricas-de-software.ppt. Acesso em 26/07/2011.

Métricas de Software - Placido Antônio de Souza Neto, disponível em <http://www.cefetrn.br/~placido/disciplina/pgp/aulas/Metricas.pdf>. Acesso em 26/07/2011.

Artigo Engenharia de Software 21 - Métricas de Software disponível em <http://www.devmedia.com.br/articles/viewcomp.asp?comp=15776>. Acesso em 26/07/2011.

Internation Function Point Users Group (IFPUG), disponível em <http://www.ifpug.org/>. Acesso em 26/07/2011.

Artigo Pontos de Função e Medidas, DEKKERS, Carol A. Publicado no QAI Journal em Dezembro de 1998. Brazilian Function Point Users Group (BFPUG), disponível em <http://www.bfpug.com.br/Artigos/Dekkers-PontosDeFuncaoEMedidas.htm>. Acesso em 26/07/2011.

Como os Pontos de Função são contados. Brazilian Function Point Users Group (BFPUG), disponível em <http://www.bfpug.com.br/Como%20Contar%20PF.htm>. Acesso em 26/07/2011.

Artigo Pontos de Função ou Pontos por Caso de Uso? Como estimar projetos orientados a objetos. AGUIAR, Mauricio. Publicado na Developers' Magazine, disponível em [http://www.bfpug.com.br/Artigos/UCP/Aguiar-Pontos de Funcao ou Pontos por Caso de Uso.pdf](http://www.bfpug.com.br/Artigos/UCP/Aguiar-Pontos%20de%20Funcao%20ou%20Pontos%20por%20Caso%20de%20Uso.pdf). Acesso em 26/07/2011.

Métricas em Engenharia de Software - Pontos por Caso de Uso / Pontos por Função, disponível em http://www.inf.ufrgs.br/~zirbes/MaterialAula/Gerencia%20e%20Administracao%20de%20Projetos%202010_2/gap/Metricas%202008%20PCU_PPF.ppt, Acesso em 26/07/2011.

Plataforma de desenvolvimento Pinhão Paraná. Métrica para Estimativa de Projetos (UCP). CELEPAR - Informática do Paraná, disponível em <http://pt.scribd.com/doc/57886000/UCP-Use-Case-Points>, Acesso em 26/07/2011.