

# Métricas de Software

Autor: Marco Aurélio Cordeiro - GPS

## INTRODUÇÃO

Software é atualmente um dos maiores componentes do orçamento de muitas organizações. A maioria delas reconhece a importância de controlar os gastos com software, de analisar a performance dos resultados obtidos com o seu desenvolvimento e manutenção, a fim de permitir uma padronização. Para fazer isto, necessitamos fazer uso de medidas e de modelos apropriados.

Medidas são necessárias para analisar qualidade e produtividade do processo de desenvolvimento e manutenção bem como do produto de software construído. Medidas técnicas são necessárias para qualificar a performance técnica dos produtos do ponto de vista do desenvolvedor. Por outro lado, medidas funcionais são necessárias para qualificar a performance dos produtos pela perspectiva do usuário. Medidas funcionais devem ser independentes das decisões do desenvolvimento técnico e implementação. Tais medidas podem ser utilizadas para comparar a produtividade de diferentes técnicas e tecnologias.[1]

E a partir de medições torna-se possível realizar uma das atividades mais fundamentais do processo de gerenciamento de projetos que é o planejamento. A partir deste planejamento, passamos a identificar a quantidade de esforço, o custo e as atividades que serão necessárias para a realização do projeto.[2]

Há bem pouco tempo, a única base para a realização de estimativas era a experiência da equipe técnica envolvida no projeto, ou seja, um processo inteiramente subjetivo e que fatalmente levava a atividades atropeladas ou não realizadas, produtos com deficiência funcional, custo de realização além do previsto e atraso na entrega do produto. Um dos grandes problemas da utilização da experiência passada de desenvolvimento de projetos de software em novos desenvolvimentos é a dificuldade de estabelecer semelhanças de funcionalidade e tamanho entre projetos de software.

## Medidas de Software

A medição é algo comum no mundo da engenharia. Infelizmente, a engenharia de software está longe de ter uma medição padrão amplamente aceita e com resultados sem nenhum fator subjetivo. Temos dificuldade em concordar sobre o que medir e como avaliar o resultado das medições obtidas.

As métricas de software, do ponto de vista de medição, podem ser divididas em duas categorias: medidas diretas e indiretas.

Podemos considerar como medidas diretas do processo de engenharia de software o custo e o esforço aplicados no desenvolvimento e manutenção do software e do produto, a quantidade de linhas de código produzidas e o total de defeitos registrados durante um determinado período de tempo. Porém, a qualidade e a funcionalidade do software ou a sua capacidade de manutenção são mais difíceis de ser avaliadas e só podem ser medidas de forma indireta.

Também podemos dividir as métricas de software, sob o ponto de vista de aplicação, em duas categorias: métricas de produtividade e 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.

## Métricas Orientadas ao Tamanho

A medida de software mais familiar é a contagem de linhas de código [3]. Embora esta métrica possa parecer simples, existe discordância sobre o que constitui uma linha de código. Para a maioria dos pesquisadores, a medida de linhas de código não deveria contar linhas de comentário e linhas em branco, uma vez que estas servem para a documentação interna do programa e não afeta a sua funcionalidade. Um outro problema é que este sistema de medidas está fortemente ligado à linguagem de programação utilizada, impossibilitando a utilização de dados históricos para projetos que não utilizam a mesma linguagem. Este tipo de medida é mais utilizado para a obtenção de informações de realização do projeto, sendo muito difícil o seu uso em estimativas. Um conjunto de métricas de qualidade e produtividade pode ser desenvolvido com esta técnica, conforme a tabela 1 abaixo:

---

<b>Produtividade</b>	<b>Qualidade</b>	<b>Custo</b>	<b>Documentação</b>
KLOC <sup>1</sup> /pessoa-mês	Defeitos/KLOC	\$/LOC	Páginas/KLOC

Tabela 1 – métricas orientadas ao tamanho.

### ***Métricas Orientadas à Função***

Em vez de contar as linhas de código, a métrica orientada à função concentra-se na funcionalidade do software. Proposta no início da década de 70 por pesquisadores da IBM, a pedido de um grupo de usuários, cujo trabalho era identificar as variáveis críticas que determinam a produtividade da programação. Descobriram que poderiam basear a avaliação de um software medindo o valor das funções executadas pelos programas, em vez de utilizar como base o volume ou a complexidade do código dos programas [4].

Em 1979, Allan Albrecht [5], prossequindo estas pesquisas, introduziu uma técnica de avaliação conhecida como Pontos por Função. Esta métrica está baseada na visão externa do usuário, sendo independente da linguagem utilizada, permitindo calcular o esforço de programação e auxiliando o usuário final a melhorar o exame e avaliação de projetos[6].

Seus objetivos são:

- Medir o que foi requisitado e recebido do usuário;
- Medir independente da tecnologia utilizada para a implementação;
- Prover uma métrica de medição para apoiar a análise de produtividade e qualidade;
- Prover uma forma de estimar o tamanho do software;
- Prover um fator de normalização para comparação de software.

Determinam-se os pontos por função de uma aplicação em três etapas de avaliação [7]. A primeira resulta na contagem de pontos por função não ajustados, que refletem as funções específicas e mensuráveis do negócio, provida ao usuário pela aplicação. A segunda etapa da avaliação gera o fator de ajuste, que representa a funcionalidade geral provida ao usuário pela aplicação. A terceira etapa resulta na contagem de pontos por função ajustados, que reflete o fator de ajuste aplicado ao resultado apurado na primeira etapa. O cálculo do fator de ajuste é baseado em 14 características gerais dos sistemas, que permitem uma avaliação geral da funcionalidade da aplicação. Estas características são:

1. comunicação de dados;
2. processamento distribuído;
3. performance;
4. utilização de equipamento;
5. volume de transações;
6. entrada de dados on-line;
7. eficiência do usuário final;
8. atualização on-line;
9. processamento complexo;

10. reutilização de código;
11. facilidade de implantação;
12. facilidade operacional;
13. múltiplos locais;
14. facilidade de mudanças.

A métrica de pontos por função (FP), como as linhas de código (LOC), é controversa. Este sistema é independente da linguagem de programação e se baseia em dados que são conhecidos logo no começo da evolução de um projeto, tornando-se mais atraente como abordagem de estimativa. Porém, a contagem se baseia parcialmente em dados subjetivos, implicando à organização estabelecer um plano de implantação da sistemática de medição, definindo padrões para a contagem, antes do início efetivo da utilização. Isto é fundamental para que os resultados das medições possam ser comparados entre os projetos, gerando uma linha de referência (baseline) das informações históricas coletadas e armazenadas. Mais adequada para a prática de estimativas de software, esta técnica chamou a atenção das organizações de desenvolvimento de software e da academia, sendo formado, em 1986, um grupo internacional de usuários da técnica de pontos por função, chamado IFPUG (International Function Point User Group) destinado a implementar melhorias e disseminar informações da técnica. Estas informações são disponibilizadas através do manual de práticas de contagem do IFPUG[8], cuja última versão 4.1 foi publicada em janeiro de 1999. Por este motivo, há que se cuidar quando da utilização das informações históricas das medições, já que esta técnica vem sendo melhorada seguindo uma evolução de versões, e neste caso, contagens realizadas com versões diferentes da técnica, não devem ser comparadas entre si.

## **CONCLUSÃO**

A grande quantidade que movimentou a indústria mundial de desenvolvimento de software, atualmente em valores acima dos 600 bilhões de dólares anuais, tem levado as organizações a buscar a melhoria do seu processo de desenvolvimento visando melhorar a qualidade e produtividade do software produzido, e por consequência, redução do custo e tempo de implementação e manutenção. Porém, existe um objetivo ainda maior das organizações, que é a previsibilidade desta qualidade e produtividade[9]. E se você não possui um processo para realizar estimativas apropriadas, você está com o seu processo de desenvolvimento de software comprometido, porque, como você vai confiar num planejamento para construção de um produto baseado num "achômetro"? Por mais capaz que seja a equipe técnica envolvida no projeto, a dificuldade de se estabelecer tamanho de um projeto de software com base na experiência passada é muito grande, pelo fato da dificuldade de se estabelecer as similaridades e as diferenças entre a funcionalidade dos softwares. Esta dificuldade é que tem levado as organizações a investimentos elevados nesta área, e muita evolução neste assunto é esperada para os próximos anos.

## **REFERÊNCIAS BIBLIOGRÁFICAS**

- [1] ABRAN, Alain. **Full function point measurement manual**: V2.0. Canadá: Software Engineering Laboratory in Applied Metrics - Universidade de Quebec, 1999.
- [2] PRESSMAN, Roger S. **Software engineering**: a practitioner's approach, 3. ed. New York: McGraw-Hill, 1995.
- [3] CONTE, S.D. **Software engineering metrics and models**. Califórnia: Benjamin/Cummings Publishing, 1985.
- [4] BRAGA, Antônio. **Análise de pontos de função**. Rio de Janeiro: Infobook, 1996.
- [5] IFPUG. **Function point counting practices manual**: V4.0. Atlanta, 1994.
- [6] AZEVEDO, Douglas J. P. de. **Análise de pontos por função para aplicações orientadas a documentos**. Porto Alegre: CPGCC da UFRGS, 1999.
- [7] IFPUG. **Function point counting practices manual**: V3.1. USA, 1991.

[8] IFPUG. **Function point counting practices manual**: V4.1. Ohio, 1999.

[9] JALOTE, Pankaj. **CMM in practice**: processes for executing software projects at Infosys. USA: Addison Wesley Longman, 1999.

---

<sup>1</sup> KLOC = Milhares de linhas de código.

---

PARANA

Bate Byte

Notícias

[cordeiro@celepar.gov.br](mailto:cordeiro@celepar.gov.br)

Copyright© 1990 / Companhia de Informática do Paraná - CELEPAR