

Engenharia Gramatical

Métricas e avaliação de qualidade em linguagens de Modelação / Especificação *

José Azevedo - pg10290@alunos.uminho.pt

José Pereira - pg17312@alunos.uminho.pt

Tiago Sá - pg18591@alunos.uminho.pt

4 de Abril de 2011

Resumo

Pequeno “Paper” de estudo e investigação sobre a temática “métricas e avaliação de qualidade em linguagens de Modelação / Especificação” no âmbito do módulo Engenharia Gramatical da UCE Engenharia de Linguagens. Aborda-se a noção de métrica aplicada à engenharia de Software bem como algumas métricas utilizadas na indústria. Procura-se fazer um estudo sobre o desenvolvimento realizado na área das linguagens de especificação, mais concretamente na especificação de modelos de Software. Quais os paradigmas utilizados e inovações desenvolvidas pelos investigadores da área.



Mestrado em Engenharia Informática
Engenharia de Linguagens

*Professores Orientadores: Pedro R. Henriques - pedrorangelhenriques@gmail.com, Daniela da Cruz - danieladacruz@gmail.com

Conteúdo

1	Métricas de Software	3
1.1	Medidas Directas	4
1.2	Medidas Indirectas	4
1.3	Métricas Orientadas ao Tamanho	5
1.4	Métricas Orientadas à Funcionalidade	5
1.5	Exemplos de métricas	6
1.6	Objetivos da Medição de Software e utilidade das métricas	6
2	Validação das Métricas de Software	6

1 Métricas de Software

“As a recognized discipline, software engineering traces its roots back to the 1968 NATO conference where the term was first used extensively to highlight the need for an engineering approach to the development of software. In the 30 years since that first “software engineering” conference, significant attempts have been made to improve the overall effectiveness of the software development process, and thus reduce the frequency and severity of software project failures. A major part of this improvement effort has been the attempt to develop quantitative measures which can be used to more accurately describe and better understand and manage the software development life cycle. Thus, many software metrics and models have been introduced during this period.”[3]

A **medição** é algo transversal em todo o mundo da engenharia. Contudo em engenharia de software este conceito apresenta um nível elevado de complexidade na formalização do mesmo. Esta área carece da implementação de uma normalização amplamente aceite, com resultados sem factores subjectivos, validados e amplamente aplicáveis.

“Effective management of any process requires quantification, measurement, and modeling. Software metrics provide a quantitative basis for the development and validation of models of the software development process. Metrics can be used to improve software productivity and quality. This module introduces the most commonly used software metrics and reviews their use in constructing models of the development process. Although current metrics and models are certainly inadequate, a number of organizations are achieving promising results through their use. Results should improve further as we gain additional experience with various metrics and models.”[2]

As **métricas de software** permitem realizar a gestão e planeamento de projectos. Dos resultados destas quantificações permitem avaliar a quantidade de esforço, o custo e as dependências no desenvolvimento de um projecto de software, permitindo assim tomar decisões cuidadas no seu planeamento.

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

Podemos considerar como **medidas directas** 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 registados durante um determinado período de tempo.

Porém, a qualidade e a funcionalidade do software ou a sua capacidade de manutenção são mais difíceis de serem avaliadas e só podem ser medidas de forma indirecta (**medidas indirectas**).

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** concentram-se na saída do processo de engenharia de software e **métricas de qualidade** indicam o quanto o software atende aos requisitos definidos pelo utilizador. [4]

1.1 Medidas Directas

Custo
Esforço
Linhas de Código
Velocidade de Execução
Memória
Número de Erros
Complexidade ciclomática

1.2 Medidas Indirectas

Funcionalidade
Qualidade
Complexidade
Eficiência
Confiabilidade
Manutensibilidade

Para melhor compreender o tema abordado é necessário então formalizar a noção dos seguintes conceitos:

Medida: *“uma indicação quantitativa da extensão, quantidade, dimensão, capacidade ou tamanho do produto ou do processo.”*

Medição: *“acto de determinação de uma medida.”*

Indicador: *Consiste n “uma métrica ou a combinação delas, que fornece compreensão do processo de software, de um projecto ou do produto.”*

As medições de software podem ser organizadas em outras classes, as quais serão definidas a seguir:

Métricas da produtividade, baseadas na saída do processo de desenvolvimento do software com o objectivo de avaliar o próprio processo;

Métricas da qualidade, que permitem indicar o nível de resposta do software às exigências explícitas e implícitas do cliente;

Métricas técnicas, nas quais encaixam-se aspectos como funcionalidade, modularidade, manutensibilidade, etc...

Sob uma outra perspectiva, é possível definir uma nova classificação das medições:

Métricas orientadas ao tamanho, baseadas nas medições directas da Engenharia de Software;

Métricas orientadas à função, que oferecem medidas indirectas;

Métricas orientadas às pessoas, as quais dão indicações sobre a forma como as pessoas desenvolvem os programas de computador.

1.3 Métricas Orientadas ao Tamanho

Estão fortemente ligadas ao tamanho da linguagem de programação utilizada, impossibilitando a utilização de dados estatísticos para projectos que não utilizam a mesma linguagem. O número de linhas de código é o exemplo de uma **métrica orientada ao tamanho**, havendo contudo discordância na definição do conceito.

Linhas de código fonte (SLOC - Source lines of code) é uma medida de software usada para medir o tamanho de um programa de software, através da contagem do número de linhas no texto do código fonte do programa. A SLOC é normalmente usada para prever a quantidade de esforço que será necessário para desenvolver um programa, bem como a estimativa de produtividade de programação ou do esforço. Também são feitas estimativas sobre o custo de desenvolvimento de um programa em moeda corrente por linha de código.

1.4 Métricas Orientadas à Funcionalidade

Em vez de contar as linhas de código, a métrica orientada à função concentra-se na funcionalidade do software. Em 1979, Allan Albrecht, introduziu uma técnica de avaliação conhecida como **Pontos por Função**.

A avaliação é realizada sob o ponto de vista do utilizador, avaliando-se o tamanho e a complexidade de um produto de software. Efectua-se um cálculo sobre os **Arquivos Lógicos Internos, Arquivos de Interface Externa, Entradas Externas, Consultas Externas e Saídas Externas**. Cada item deste define um peso que no final determina a quantidade de pontos de função da aplicação. Para o desenvolvimento de um novo sistema ou os pontos necessários para se realizar uma manutenção em um sistema já existente. Os pontos calculados servem para se chegar as horas totais necessárias ao projecto.

As **métricas orientadas à funcionalidade** baseia-se essencialmente na visão externa do utilizador. São independentes da linguagem de programação utilizada, permitindo calcular o esforço de programação exigido pelos projectos. Auxiliam os utilizadores a melhorar o os testes e avaliações realizadas nos projectos.

Tem como principais objectivos:

- Medir o que foi requisitado e recebido do cliente/utilizador;
- Efectuar medições, independentemente da tecnologia utilizada na implementação;
- Providenciar uma métrica de medição para apoiar a análise de produtividade e qualidade;
- Providenciar uma forma de estimar o tamanho do software;
- Providenciar um factor de normalização para comparação de software.

A medição de propriedades do software deve ser realizada pois permite indicar a qualidade de um produto, permite avaliar a produtividade dos que desenvolvem o produto, permite determinar os benefícios derivados de novos métodos e ferramentas de engenharia de software, permite formar uma base para estimativas, permite ajudar a justificação de aquisição de novas ferramentas ou formações adicionais.

Um aumento de produtividade mais representativo será obtido quando se conseguir estabelecer uma sistemática de métricas significativa para os resultados do desenvolvimento de software e utiliza-la efectivamente.

As métricas de software, sob o ponto de vista de aplicação, também podem ser divididas em duas categorias: **métricas de produtividade e de qualidade**. As **métricas de produtividade** concentram-se nos resultados obtidos no final do processo de engenharia de software e

as **métricas de qualidade** indicam o quanto o software atende aos requisitos definidos pelos utilizadores.

1.5 Exemplos de métricas

Número de defeitos introduzidos por programador por hora.

Número de “patches” disponibilizados.

Número de mudanças nos requisitos

Número de linhas de código útil.

Análise de pontos de função (APF).

1.6 Objetivos da Medição de Software e utilidade das métricas

Entender: ajudam a entender o comportamento e o funcionamento de produtos de software.

Avaliar: utilizadas para determinar padrões, metas e critérios de aceitação.

Controlar: utilizadas para controlar processos, produtos e serviços de software.

Prever: utilizadas para prever valores de atributos.

2 Validação das Métricas de Software

Cem Kaner e Walter Bond afirmam que as métricas aplicadas em engenharia de Software devem acima de tudo ser validadas. Nos seus trabalhos indicam como e em que medida deve ser realizada esta validação.

“We assert that Software Engineering as a field presents an approach to measurement that underemphasizes measurement validity (the condition that the measurement actually measures the attribute in question). This has a likely consequence: if a project or company is managed according to the results of measurements, and those metrics are inadequately validated, insufficiently understood, and not tightly linked to the attributes they are intended to measure, measurement distortions and dysfunctional should be commonplace.”[1]

Referências

- [1] Cem Kaner and Walter P. Bond. Software engineering metrics: What do they measure and how do we know? (num.14-16), September 2004. Chicago, IL. 6
- [2] Everaldo E. Mills. Software metrics. December 1998. Computer Science/Software Engineering Department, Seattle University, 900 Broadway, Seattle, WA 98122-4340, USA E-mail: mills@seattleu.edu. 3
- [3] Everaldo E. Mills. Metrics in the software engineering curriculum. vol.6(num.1-4), April 1999. Computer Science/Software Engineering Department, Seattle University, 900 Broadway, Seattle, WA 98122-4340, USA E-mail: mills@seattleu.edu. 3
- [4] Wikipedia. Métrica de software. http://pt.wikipedia.org/wiki/M%C3%A9trica_de_software, 2011. 3