



**MÉTRICAS DE SOFTWARE PARA
PROGRAMAÇÃO FUNCIONAL**

PROPOSTA DE TRABALHO DE GRADUAÇÃO

Aluno: Bruno Costa Bourbon (bcb@cin.ufpe.br)

Orientador: André L. M. Santos (alms@cin.ufpe.br)

Recife, 08 de novembro de 2004.

Contexto

De um modo em geral, um fabricante está interessado (na maioria da vez) na qualidade de seu produto, pois sabe que venderá mais, seja ele carro ou um pacote de software. A importância dos aspectos de qualidade depende do interesse das partes enquanto usuários se preocupam com o consumo do combustível de um carro, os engenheiros e os técnicos, que constroem e mantêm os veículos, a facilidade da montagem das partes (peças) se tornam uma preocupação maior. O fabricante está interessado no custo e duração da produção. A medição dos aspectos de qualidade se faz necessária para comparação com outros produtos ou processos.

No ambiente de desenvolvimento de software situações similares também ocorrem. Existem aspectos de qualidade relativa ao usuário, por exemplo, interface e desempenho, e outros aspectos relativos aos programadores que irão projetar e desenvolver o programa. A disciplina de engenharia de software oferece métodos para design e produção de software. O campo das medições de software provê maneiras para quantificação da qualidade dos aspectos dos softwares, relacionadas ao produto, o processo e os recursos. Uma métrica de software, bastante comum, é o tamanho do software, normalmente expressa no número de linhas de código. Mas existem várias outras medidas, e necessário ser capaz de decidir quando usar que métrica e como utilizar. Com essas métricas, será capaz de fazer uma avaliação objetiva dos méritos relativos aos produtos de software e métodos do desenvolvimento do software.

Nas linguagens de programação se destacam três paradigmas: o imperativo, o orientado a objeto e o funcional. Enquanto os dois primeiros vêm-se munidos de dezenas (até centenas) de ferramentas e metodologias que auxiliam o desenvolvimento de sistemas de software, desde dos menores até os mais complexos, as linguagens funcionais estão famintas por esse tipo de suporte. Muito foi feito no ambiente teórico, mas pouca coisa se tornou prática (seja comercial ou gratuita), excluídos os protótipos. Isso leva a custos e prazos pouco viáveis quanto se deseja o desenvolvimento de sistemas complexos utilizando linguagens funcionais.

Essa realidade está mudando aos poucos com o aparecimento de ambiente de desenvolvimento integrado (IDE sigla em inglês) extensíveis, como o Eclipse e Microsoft Visual Studio.

Datas e Assinaturas

Recife, 08 de novembro de 2004.

André L. M. Santos
(Orientador)

Bruno C. Bourbon
(Proponente)