

Universidade do Minho
Mestrado em Engenharia Informática
2011



Análise e Conceção de Software
Análise e Transformação de Software
Benchmarking Tiger
Ano Lectivo de 2011/2012

PG20189 **André Pimenta Ribeiro**
PG19824 **Cedric Fernandes Pimenta**
PG20978 **Rafael Fernando Pereira Abreu**

7 de Abril de 2012

Resumo

Neste relatório, e no âmbito do módulo de análise e transformação de software, apresenta-se uma análise à linguagem *Tiger*. Esta análise tem como pressuposto a criação de uma aplicação Web que seja capaz de dar uma avaliação objectiva a programas escritos na linguagem *Tiger*. Para realizar a análise são necessários vários passos importantes, que passam desde a criação de um parser para interpretar a linguagem *Tiger*, definição de métricas a utilizar nas avaliações, assim como o modelo de avaliação baseado na norma **ISO 9126**.

Posto isto, ainda será apresentada a modelação do portal de certificação de qualidade que tem como objectivo realizar a avaliação de programas escritos na linguagem *Tiger*.

Conteúdo

Conteúdo	i
1 Introdução	1
1.1 Contextualização e Apresentação do Objecto de Estudo	1
1.2 Motivação e Objectivos	1
1.3 Estrutura	2
2 Domínio do problema	3
2.1 O que é a linguagem <i>Tiger</i>	3
2.2 Análise da linguagem <i>Tiger</i>	3
3 Qualidade de programas <i>Tiger</i>	4
3.1 Métricas de análise	4
3.2 Modelo de qualidade	5
4 Conclusão	6
5 Anexos	7

Capítulo 1

Introdução

1.1 Contextualização e Apresentação do Objecto de Estudo

Com o desenvolvimento da ciência e com a existencia de cada vez mais "programadores" começa-se a necessitar um maior controlo sobre o software desenvolvido, nomeadamente poder avaliar a qualidade do código desenvolvido.

Mas dar uma avaliação a um programa desenvolvido em qualquer linguagem, não é um processo simples e linear, antes pelo contrario, é um processo muitas vezes subjectivo e alvo de discussões, sobre quais as melhores medições, entre outros assuntos provenientes do processo de avaliação.

Neste trabalho prático é proposto a criação de um portal de certificação de qualidade de programas *Tiger*, que tem como principal objectivo, através de uma submissão de um programa poder dar uma avaliação da qualidade do mesmo programa submetido. Mas tal como já foi referido o processo de avaliação nem sempre é um processo pacifico. No entanto a avaliação automática em certos pontos é possível, desde que se faça uma análise correcta da linguagem e se destaquem características importantes destas, que deverão ser sempre cumpridas de forma a se poder atingir o grau de qualidade satisfatoria em qualquer programa.

Para atingir as características exatas, tal como foi referido, iremos criar um catalogo de métricas que se distinguem nos programas *Tiger* e posteriormente fazer uma análise baseada na norma **ISO 9126**, tendo como fundamentos as métricas definidas. Esta mesma análise, ou melhor, este modelo de qualidade irá ser posto em funcionamento numa aplicação Web, que terá como principal objectivo certificar a qualidade de programas *Tiger*. Toda a avaliação estará exposta à supervisão de um administrador que tendo em conta as diferentes estatísticas apresentadas irá definir periodicamente os limites de qualidade dos programas *Tiger*.

1.2 Motivação e Objectivos

"... Um dos objectivos da engenharia de software é transformação da criação de software de uma maneira artística, indisciplinada e pouco entendível para uma forma devidamente controlada , quantificavel e previsivel..."

Pretende-se com este trabalho prático a criação de um portal que seja capaz de avaliar programas escritos na linguagem *Tiger*, mas a cima de tudo o desenvolvimento de capacidade que garantam uma boa prática no desenvolvimento de software de forma genérica, e não apenas no caso de estudo.

Pesquisas realizadas em empresas de software indicam que mais da metade de grandes projetos

de software se deparam com algum tipo de atraso, excesso de custo ou prazo ou algum fracasso na execução quando implantado por falta de controlo. As **métricas de software** é um assunto já discutido à muitos anos, mas poucas vezes utilizado, o que muitas vezes leva a acontecer situações como a referida anteriormente. Um métrica é uma medição de um atributo (propriedade ou característica) de uma determinada entidade (produto, processo ou recurso), que irá ser utilizada neste caso de estudo, para efectuar medidas ao código da linguagem apresentada, que no caso de estudo é a linguagem *Tiger*. A utilização de métricas irá permitir aprofundar os processos de desenvolvimento, mas acima de tudo vai permitir uma avaliação real e objectiva. A avaliação já referida tem como objectivo permitir identificar as melhores práticas de desenvolvimento de software, assim como avaliar o impacto da variação de um ou mais atributos do produto ou processo na qualidade e/ou produtividade. Posto isto pretende-se como objectivo principal destacar o processo de avaliação de software, aplicando o mesmo conhecimento no portal de certificação de qualidade de programas, neste caso na linguagem *Tiger*.

1.3 Estrutura

O presente relatório é constituído por 4 capítulos: **Introdução** onde é feita a contextualização do problema exposto neste relatório; **Preliminares** onde serão apresentados os conceitos necessários para o problema; **Descrição da gramática e Análise dos Resultados** onde é apresentada a solução do problema (linguagem) e um exemplo desta. Por fim, o capítulo **Conclusão** que possui um pequeno balanço sobre a realização deste trabalho e particularidades que valham a pena ser mencionadas.

Capítulo 2

Domínio do problema

2.1 O que é a linguagem *Tiger*

Falar um pouco da sua historia, de ser imperativa e etc.

2.2 Analise da linguagem *Tiger*

Falar nas palavras reservadas e apresentar o parser.

Capítulo 3

Qualidade de programas *Tiger*

3.1 Métricas de análise

O processo de validação e certificação de software é cada vez mais importante, assim como necessário para a construção do mesmo.

Mas como podemos afirmar que um programa é bom ou mau? Só existe bons programas se existirem maus programas, e é o conhecimentos, a experincia da interpretação e leitura de varios programas, que atingimos a noção de um bom ou mau programa.

No entanto é importante saber quais as características que podem ser utilizadas, e que ao mesmo tempo têm uma posição relevante na decisão de um bom ou mau software. Estas características serão as métricas a usar para a analise de programas escritos na linguagem *Tiger*.

... "uma métrica de qualidade de software é uma função cujos inputs são os dados do software e cujo a saida é um único valor núnmerico que pode ser interpretado como o grau que determina a qualidade do software..."

Existem então diversas métricas, tanto directas, como indirectas capazes de contribuir para um feedback à cerca da qualidade de um programa, mas no entanto nem todas são objectivas ou podem ser usadas sem a experiencia e conhecimento humano. Por isto decidimos apresentar um catalogo de métricas, que contenha apenas métricas que reunam minimo de concenso e que sejam passíveis de utilização na linguagem *Tiger*. Estamos então a falar de métricas que devem ser calculaveis, facilmente entendidas e que possam ser testadas, que sejam passíveis de estudos estatisticos e que conseqüentemente se expresse em alguma unidade, por ultimo e para o caso de estudo em concreto, deverão ser repetíveis e independentes do observador.

- **Linhas de código** . O numero de linhas de código é capaz de ser a métrica mais vulgar e comum entre as diferentes linguagens, assim como a mais evidente. Pretende-se com o numero de linhas de código sobre tudo analisar a dimensão do código em análise, pois como é do senso comum da engenharia de software *"...Um bom código não é um código grande..."*
- **Métricas de Complexidade Ciclométrica de McCabe**. Métricas que se baseiam na medida do numero de caminhos linearmente indeendentes de um modulo ou função. completar ...
- **Métricas Halstead** . São métricas que se baseia na complexidade do problema e das funções, estas métricas têm em conta por exemplo o numero de operadores distintos, ou numero de operandos distinitos (etc) completar ...
- **Comentarios**. A existencia de comentarios ao longo do codigo é sempre positiva, pois

deve ter como objectivo auxiliar a interpretação deste. Pode-se até ser mais exigente, e afirmar que a existencia de um comentario antecedente a uma função/método deveria ser obrigatorio, contendo uma explicação da utilidade da mesma, assim como os seus principios de funcionamento.

- **Numero de erros.** A existencia de erros em qualquer programa é um indicador por si só de má qualidade, ou melhor, de que no exato momento a que se submeteu à avaliação é um software inválido. Pode-se então considerar o numero de erros um métrica com utilidade para o caso de estudo.
- **Tempo .** Esta é uma metrica que indirectamente se encontra relacionada com a complexidade do problema , no entanto iremos apresentar o tempo de analise como uma métrica a ter em consideração no processo de Benchmarking.

3.2 Modelo de qualidade

Modelo de qualidade baseado na norma ISO 9126		
Caracteristica	Sub-Caracteristica	Métricas a usar
Funcionalidade	Confiança	Lucus Radebe
	adequação	Michael Duberry
	acurácia	Dominic Matteo
	Segurança	Dominic Matteo
	interopeabilidade	Didier Domi
Segurança	Maturidade	David Batty
	Tolerancia a falhas	nº erros, nº de linhas
	Recuperabilidade	Jody Morris
Usabilidade	Inteligibilidade	Alan Smith
	Apreensibilidade	Mark Viduka
	Operabilidade	Mark Viduka
Eficiencia	Comportamento em relação ao tempo	Alan Smith
	Utilização de recursos	Mark Viduka
Manutenção	Analisabilidade	Alan Smith
	Modificabilidade	Mark Viduka
	Estabilidade	Mark Viduka
	testabilidade	Mark Viduka
Portabilidade	Adaptatibilidade	Alan Smith
	Capacidade para ser instalado	Mark Viduka
	Capacidade para substituir	Mark Viduka
	Coexistencia	Mark Viduka

Capítulo 4

Conclusão

Capítulo 5

Anexos