

Um Método de Extração de Valores Referência para Métricas de Softwares Orientados por Objetos

Tarcísio G. S. Filó¹, Mariza A. S. Bigonha¹, Kécia Aline Marques Ferreira²

¹Programa de Pós-Graduação em Ciências da Computação (PPGCC)
Departamento de Ciência da Computação - Universidade Federal de Minas
Gerais (UFMG) - Belo Horizonte - MG - Brasil

²Departamento de Computação - Centro Federal de Educação Tecnológica de Minas
Gerais (CEFET-MG) - Belo Horizonte - MG - Brasil

{tfilo,mariza}@dcc.ufmg.br, kecia@decom.cefetmg.br

Nível: Mestrado

Ano de ingresso no programa: 2012

Época prevista de conclusão: março de 2014

Data da aprovação da proposta de dissertação: março de 2013

Eventos Relacionados: SBES

Resumo. *Valores referência para métricas de softwares orientados por objetos são úteis em diversos campos da Engenharia de Software, tais como processos de filtragem em estratégias de detecção de bad smells e identificação de medições anômalas em um processo de medição de produto, parte integrante de um programa de qualidade de software. Nesse trabalho apresentamos um método de extração de valores referência para métricas de softwares orientados por objetos, bem como mostramos a sua aplicação na identificação de valores referência para 18 métricas em um dataset de 111 softwares open-source. Os próximos passos do trabalho consistem em avaliar empiricamente a eficácia dos valores referência na avaliação da qualidade do software.*

Palavras-chave: *métricas, valores referência, qualide de software, medição de software.*

Abstract. *Thresholds for object-oriented software metrics are helpful in severous fields of Software Engineering, such as filtering mechanisms in bad smells detection strategies and identification of anomalous measurements in a software quality program. In this work, we present a method to extract thresholds of object-oriented software metrics and show its application in the thresholds identifying thresholds for 18 metrics of an 111 open-source software systems dataset. The next steps are related to the empirical evaluation of the thresholds effectiveness in software quality evaluation.*

Keywords: *metrics, thresholds, software quality, software measurement.*

1. Introdução

Pesquisas relacionadas com valores referência para métricas de software orientado por objetos vem sendo realizadas a partir de diferentes metodologias e apresentam valores referência de formatos e significados distintos. Além disso, valores referência para diversas métricas propostas na literatura ainda não são conhecidos. Uma consequência direta desse problema é a inexistência de condições adequadas para avaliar quantitativamente e de forma eficaz a qualidade dos softwares. Valores referência permitem responder questões simples como “Quais métodos do sistema possuem uma quantidade excessiva de parâmetros?” ou “Quais classes do sistema possuem muitos métodos?”.

Nesse contexto, o trabalho de pesquisa proposto tem por objetivo contribuir para a solução desse problema, catalogando e avaliando valores referência para um conjunto de 18 métricas de softwares orientados por objetos, a partir de um método sólido e embasado em dados estatísticos para derivação desses valores. Com valores referência catalogados, vislumbra-se que as métricas possam ser efetivamente aplicadas na indústria de software, e, conseqüentemente, a qualidade dos softwares possa ser avaliada de forma automática.

2. Solução Proposta

Nesta seção é descrita a metodologia a ser aplicada na derivação de valores referência para as métricas de softwares orientados por objetos coletadas no *Qualitas.class Corpus* [Terra et al. 2013], que é uma versão compilada do *Qualitas Corpus* proposto por Tempero *et al.* (2010), no qual são disponibilizados arquivos XML referentes aos 111 sistemas contendo medições de 18 métricas de softwares:

- **Básicas:** número de classes por pacote, número de métodos, número de atributos, número de métodos sobrescritos, número de parâmetros, número de métodos estáticos e número de atributos estáticos.
- **Complexidade:** linhas de código por método, índice de especialização, complexidade de *McCabe*, profundidade de blocos aninhados e distância normalizada.
- **Conjunto CK:** métodos ponderados por classe, profundidade de árvore de herança, número de filhas e ausência de coesão em métodos.
- **Acoplamento:** acoplamento aferente e acoplamento eferente.

Para a preparação dos dados, neste trabalho, foi desenvolvida uma ferramenta que lê os arquivos XML disponibilizados no *Qualitas.class Corpus* e alimenta um banco de dados *Mysql* com o objetivo de normalizar as medidas. Realizada a conversão por meio dessa ferramenta, têm-se uma forma rápida e ágil de agregar e sumarizar os dados.

2.1. Data-Fitting

Um problema importante em Estatística é obter informação sobre a distribuição seguida pelo conjunto de dados analisados [Gibbons and Chakraborti 2003]. Para esse propósito, a ferramenta EasyFit (2013), por meio do teste de aderência de *Kolmogorov-Smirnov*, realiza a seleção da distribuição apropriada aos dados, o qual é o procedimento de escolher uma distribuição que tenha melhor ajuste para um conjunto de dados. O objetivo dessa etapa é estabelecer a distribuição apropriada a cada métrica de software estudada. Segundo Baxter *et al.* (2006), explorar as distribuições dos valores das métricas de software é fundamental para avançar no entendimento das estruturas internas de softwares. A partir da distribuição é possível identificar e entender suas características, como por exemplo, se o valor médio é ou não representativo para a análise.

2.2. Análise dos Dados

EasyFit gera um histograma de probabilidade dos dados ajustados a distribuição. A partir dessa visualização gráfica e do conhecimento das características da distribuição mais aderente aos dados, é possível derivar valores referência para as métricas. Na abordagem de Ferreira *et al.* (2012), quando a métrica apresenta uma distribuição que possui valor médio representativo, como por exemplo, a distribuição de *Poisson*, este valor representa o valor típico da métrica. Senão, são identificadas três faixas de valores: *Bom/Frequente*, *Regular/Ocasional* e *Ruim/Raro*, onde *Bom/Frequente* corresponde a valores com alta frequência, caracterizando os valores mais comuns da métrica, na prática. Para os autores, esses valores não expressam necessariamente as melhores práticas da Engenharia de Software, e sim um padrão seguido pela maioria dos softwares. *Ruim/Raro* corresponde a valores com baixa frequência, e a faixa *Regular/Ocasional* é intermediária, e são valores que não são nem muito frequentes nem raros. A frequência é estabelecida via uma análise gráfica do histograma de probabilidade dos dados ajustados a distribuição.

3. Avaliação dos Resultados

Ferreira *et al.* (2012) avaliaram os valores referência propostos via estudos de casos, considerando que a sua aplicação pode resultar em duas situações possíveis: o valor referência avalia corretamente ou não o método, a classe ou o pacote. A avaliação correta ocorre quando a faixa em que a métrica cai reflete a situação real do artefato avaliado. Quando não há avaliação correta do valor referência, os estudos de caso consideram duas situações: (1) *falso-positivo* - o valor referência gera uma classificação ruim e a inspeção qualitativa não identifica problemas estruturais; (2) *falso-negativo* - o valor não classifica a classe como ruim, mas a inspeção qualitativa identifica problemas em sua estrutura. A identificação de casos *falsos-negativos* demanda uma avaliação qualitativa de um número muito grande de classes. No entanto, considerando que a derivação é feita de forma quantitativa, via análise de distribuição das medidas coletadas de uma grande quantidade de sistemas, uma inspeção manual é inviável, dada a dimensão da amostra. Por essa razão, Ferreira *et al.* avaliaram os casos *falsos-positivos* a partir da inspeção manual apenas daquelas classes identificadas nas faixas *Ruim/Raro*, com o objetivo de avaliar a eficácia do valor referência para identificar problemas. Neste trabalho de dissertação é proposta a condução de 3 estudos de casos com os seguintes objetivos:

- **Estudo de Caso 1:** avaliar um software proprietário com a qualidade interna deteriorada, com o propósito de verificar a capacidade dos valores referência propostos em indicar essa má qualidade. Nesse sentido, esse estudo de caso investiga se a aplicação dos valores referência de fato identifica métodos e classes de qualidade, não produzindo assim resultados falsos-negativos. Em um software qualitativamente definido como ruim, espera-se que métodos e classes não sejam bem avaliados em consonância com a opinião dos desenvolvedores, refletindo o panorama de baixa qualidade. Esse estudo busca avaliar, respectivamente, as métricas de métodos, as métricas de classes e a correlação dessas métricas não avaliarem bem os métodos e classes devido a ocorrência de *bad smells* bem conhecidos.
- **Estudo de Caso 2:** avaliar uma amostra de métodos e classes classificadas como *Ruim/Raro* por meio dos valores referência estabelecidos, com o objetivo principal de verificar a presença de casos *falsos-positivos*.

- **Estudo de Caso 3:** avaliar o software JHotDraw, de reconhecida qualidade estrutural, com o objetivo de verificar a capacidade dos valores referência em indicar essa qualidade. Se isso se confirmar, a aplicação dos valores referência possui baixo risco de levarem a resultados *falsos-positivos*, afinal, as classes são de boa qualidade e, de fato, são bem avaliadas pelos valores referência. Outra confirmação dessa capacidade é que os valores referência são confiáveis quando sugerem um panorama de qualidade, minimizando o problema de avaliações com resultados *falsos-negativos*, que ocorre quando a classe não possui problemas estruturais, mas os valores referência os indicam.

4. Estado Atual do Trabalho

A preparação dos dados, *data-fitting* (Seção 2.1), análise dos dados (Seção 2.2) e a identificação dos valores referência para 18 métricas de softwares orientados por objetos foram concluídas. A Tabela 1 sumariza 5 dos 18 valores referência propostos. A avaliação dos resultados (Seção 3) está parcialmente concluída.

Table 1. Valores Referência Identificados.

Métrica	Bom/Frequente	Regular/Ocasional	Ruim/Raro
Acoplamento Aferente	$m \leq 7$	$7 < m \leq 39$	$m > 39$
Acoplamento Eferente	$m \leq 6$	$6 < m \leq 16$	$m > 16$
Número de Atributos	$m \leq 3$	$3 < m \leq 8$	$m > 8$
Número de Métodos	$m \leq 6$	$6 < m \leq 14$	$m > 14$
Complexidade de McCabe	$m \leq 2$	$2 < m \leq 4$	$m > 4$

O processo de derivação para a métrica *Número de Métodos* é o seguinte: o Gráfico de Frequência Relativa Acumulada, Figura 1(a), sugere uma distribuição de cauda-pesada. Isso significa que há um grande número de classes com poucos métodos e um pequeno número de classes com muitos métodos. Como mostrado na Figura 1(b), o conjunto de valores dessa métrica possui melhor ajuste à distribuição *Weibull*. A Figura 1(c) mostra esses dados em escala logarítmica, sugerindo uma lei de potência. Foram identificados os valores que representam o 70° e 90° percentil dos dados, que correspondem aos valores 6 e 14 (Tabela 1). Os valores 70% e 90% foram identificados via uma análise visual no Gráfico de Frequência Relativa Acumulada da Figura 1(a). Pontos marcados em formato de quadrado e losango representam as regiões identificadas nessa análise. Também determinam a escolha desses percentis os conceitos das faixas *Bom/Frequente*, *Regular/Ocasional* e *Ruim/Raro*, que devem caracterizar, respectivamente, valores muito frequentes, que ocorrem ocasionalmente e raros.

5. Contribuições Pretendidas

As principais contribuições pretendidas, oriundas da pesquisa proposta, incluem:

1. Um catálogo de valores referência para 18 métricas de softwares orientados por objetos, derivados a partir de um *dataset* de 111 softwares *open-source*.
2. Um catálogo de distribuições estatísticas que possuam melhor encaixe em relação as 18 métricas de softwares orientados por objetos, permitindo identificar características acerca de como os softwares estão sendo construídos.

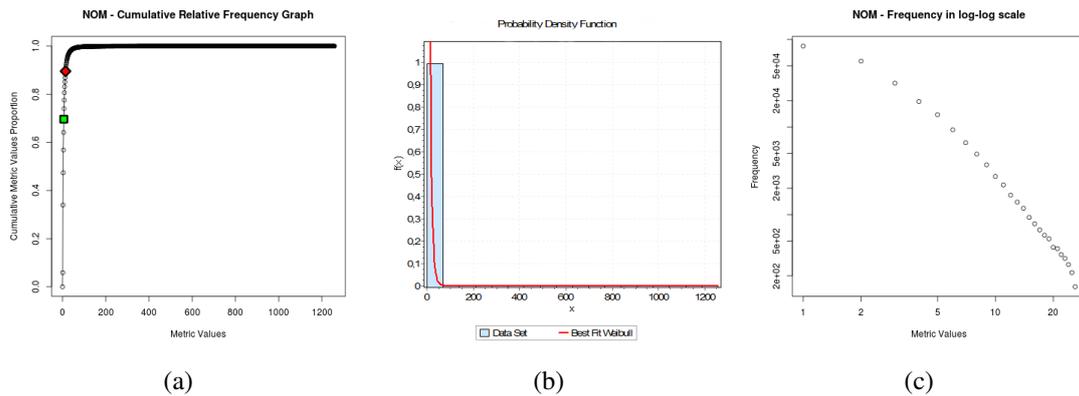


Figure 1. Número de Métodos: (a) Frequência Relativa Acumulada (b) Densidade de Probabilidade ajustada à Distribuição Weibull (c) Histograma em escala log-log.

3. Ampliação do universo de métricas com valores referência derivados a partir da metodologia de Ferreira *et al.* (2012), permitindo realizar uma análise comparativa em termos de resultados com outros trabalhos.
4. Avaliações dos valores referência em: (1) um software proprietário mal projetado, com o objetivo de estender a validação proposta a softwares fora do universo de código aberto e, (2) do *dataset* utilizado no processo de derivação.
5. Avaliação dos valores referência na identificação de métodos, classes e pacotes de baixa qualidade, bem como a avaliação em um software bem projetado.

6. Trabalhos Relacionados

Alves *et al.* (2010) propuseram um método empírico para determinar valores referências a partir de um conjunto de softwares, onde as medições coletadas são agregadas gerando informações no formato: o valor 17 da métrica *McCabe* representa 0,658% de todos os sistemas. Esses valores são ordenados e os valores referência são identificados pela escolha do 70%, 80% e 90% percentil do conjunto, caracterizando os perfis de qualidade: *low risk* (entre 0 - 70%), *moderate risk* (70 - 80%), *high risk* (80 - 90%) and *very high risk* (> 90%). Alves *et al.* não consideram a distribuição apropriada aos dados na definição dos percentis que definem os perfis.

Ferreira *et al.* (2012) identificam valores referência para 6 métricas de softwares orientados por objetos via medição das estruturas de um conjunto de 40 softwares de código aberto desenvolvidos em Java, de diferentes tamanhos, domínios e tipos, e uma análise estatística dos dados baseada em uma distribuição estatística apropriada.

Oliveira *et al.* (2014) propuseram o conceito de valores referência relativos para avaliar métricas aderentes a uma distribuição de cauda-pesada. Os valores referência são relativos pois devem ser seguidos pela maioria das entidades, mas considera-se natural que um determinado número de entidades não siga os limites definidos. O formato dos valores referência é o seguinte: $p\%$ das entidades devem ter $M \leq k$, aonde M é uma métrica de código-fonte, k é o limite definido p é a porcentagem mínima.

Quando as contribuições desse trabalho são comparadas com os resultados apresentados por Ferreira *et al.* (2012) e estudos anteriores, identificam-se três diferenças prin-

cipais: (1) Foram introduzidas melhorias no método de derivação de valores referência de Ferreira *et al.*, bem como, houve um avanço nas discussões acerca de suas limitações e qualidades em relação ao panorama atual da área. (2) Foram identificados valores referência para um número maior de métricas. (3) Os valores referência propostos provêm um *benchmark* para a avaliação quantitativa da qualidade interna de softwares orientados por objetos, considerando não somente classes, mas também métodos e pacotes.

7. Conclusões

O objetivo dessa proposta de dissertação é definir um método para extração de valores referência para métricas de softwares orientados por objetos. O método proposto consiste na análise estatística da distribuição de um conjunto de medidas extraídas de um *dataset* de softwares, apresentando três faixas de valores: *Bom/Frequente*, *Regular/Ocasional* e *Ruim/Raro*. O método não usa técnicas estatísticas muito complexas, é reproduzível e foi aplicado a uma grande quantidade de softwares para extração dos valores referência.

Desde a submissão desse artigo até hoje, avançamos em nossa pesquisa e obtivemos alguns resultados preliminares bem animadores sobre os valores referência propostos via estudos de casos realizados. Na continuidade do trabalho, pretendemos aprimorar os estudos de casos realizados, afim de avaliar os valores referência de forma a verificar se eles de fato auxiliam na identificação de problemas estruturais de software.

References

- Alves, T. L., Ypma, C., and Visser, J. (2010). Deriving metric thresholds from benchmark data. In *ICSM*, pages 1–10. IEEE Computer Society.
- Baxter, G., Freen, M., Noble, J., Rickerby, M., Smith, H., Visser, M., Melton, H., and Tempero, E. (2006). Understanding the shape of java software. In *OOPSLA '06: Proceedings of the 21st annual ACM SIGPLAN*, pages 397–412, NY, USA. ACM.
- EasyFit (2013). Easyfit. Disponível em: <http://www.mathwave.com/products/easyfit.html>. Acesso em 09/2013.
- Ferreira, K. A., Bigonha, M. A., Bigonha, R. S., Mendes, L. F., and Almeida, H. C. (2012). Identifying thresholds for object-oriented software metrics. *Journal of Systems and Software*, 85(2):244–257.
- Fowler, M. (2003). Anemic domain model. Disponível em: <http://www.martinfowler.com/bliki/AnemicDomainModel.html>. Acesso em 09/2013.
- Gibbons, J. D. and Chakraborti, S. (2003). *Nonparametric Statistical Inference (Statistics: a Series of Textbooks and Monographs)*. CRC, 4 edition.
- Oliveira, P., Valente, M. T., and Lima, F. P. (2014). Extracting relative thresholds for source code metrics. In *CSMR-WCRE, 2014 Software Evolution Week - IEEE Conference on*, pages 254–263.
- Tempero, E., Anslow, C., Dietrich, J., Han, T., Li, J., Lumpe, M., Melton, H., and Noble, J. (2010). Qualitas corpus: A curated collection of java code for empirical studies. In *2010 Asia Pacific Software Engineering Conference (APSEC2010)*, pages 336–345.
- Terra, R., Miranda, L. F., Valente, M. T., and Bigonha, R. S. (2013). Qualitas.class Corpus: A compiled version of the Qualitas Corpus. *Software Engineering Notes*, 38(5):1–4.