DANILO BATISTA DOS SANTOS

Análise da Instabilidade de Software Baseada nas Medidas de Acoplamento Aferente e Eferente

Lavras – MG

2014

DANILO BATISTA DOS SANTOS

ANÁLISE DA INSTABILIDADE DE SOFTWARE BASEADA NAS MEDIDAS DE ACOPLAMENTO AFERENTE E EFERENTE

Artigo apresentado ao Departamento de Ciência da Computação para obtenção do título de Bacharel em Sistemas de Informação.

Orientador
Prof. Dr. Antônio Maria Pereira de Resende

LAVRAS – MG
2014

DANILO BATISTA DOS SANTOS

ANÁLISE DA INSTABILIDADE DE SOFTWARE BASEADA NAS MEDIDAS DE ACOPLAMENTO AFERENTE E EFERENTE

Trabalho de Conclusão de Curso de Graduação apresentado ao Colegiado do Curso de Bacharelado em Sistemas de Informação, para obtenção do título de Bacharel.

APROVADA em 25 de novembro de 2014.

Dr. Heitor Augustus Xavier Costa

Dr. André Pimenta Freire

MSc. Eudes de Castro Lima

Dr. Antônio Maria Perreira de Resende (Orientador)

LAVRAS-MG
Novembro/2014

Análise da Instabilidade de Software Baseada nas Medidas de Acoplamento Aferente e Eferente

Danilo B. Santos, Antônio M. P. de Resende

Grupo de Pesquisa em Engenharia de Software - PqES
Departamento de Ciência da Computação – Universidade Federal de Lavras (UFLA)
Caixa Postal 3037 – 37200-000 – Lavras – MG – Brasil

danilo_batista13bs@sistemas.ufla.br, tonio@dcc.ufla.br

Abstract. A medida denominada instabilidade de software indica o grau de suscetibilidade de uma entidade quanto às alterações ocorridas em outras entidades do software. A interpretação desta medida implica que quanto mais alto o valor da instabilidade, mais sujeita a alterações inesperadas de comportamento ela estará, mesmo não tendo sofrido alterações diretas em seu código. Caso haja baixa instabilidade então tem-se indícios de que a entidade analisada é pouco dependente das demais. Caso contrário, tem-se indícios de que uma reestruturação poderia ser necessária, pois a entidade analisada é extremamente sensível as alterações ocorridas nas entidades das quais ela depende. Neste artigo, apresenta-se uma análise da medida de instabilidade envolvendo uma comparação entre os valores de referência propostos pela academia e os praticados no mercado de software livre. Adotou-se a definição de instabilidade de Martin [2], calculada por meio dos valores de acoplamento aferente e eferente. Observou-se uma escassez de valores de referências nos artigos científicos e que o valor 1, valor máximo possível para a instabilidade, é o mais comum após a medição de 107 softwares livres em 3 versões cada. Concluiu-se que os softwares livres possuem alta instabilidade e que as medidas Ca e Ce tem um impacto direto sobre a instabilidade do software e que quanto maior Ce em detrimento a Ca, maior será a instabilidade da entidade analisada.

1 Introdução

A aplicação de métodos, técnicas e ferramentas auxiliam na gestão de projetos, detecção e prevenção de erros, contribuindo na produção de um software com qualidade, independente do tamanho da empresa.

Além disto, recomenda-se o uso de medidas de software, a fim de monitorar o projeto para descobrir inconformidades e riscos desde as primeiras etapas do desenvolvimento do projeto, os quais afetariam a qualidade final do produto.

Dentre os diversos tipos de medidas aplicáveis a softwares, tem-se a medida de instabilidade. Tal medida indica o grau de suscetibilidade de uma entidade quanto às alterações ocorridas em outras entidades do software. Assim, se uma entidade possui um alto valor de instabilidade, isto implica em afirmar que existe alto risco de se alterar o

comportamento da entidade analisada devido a alterações em outra entidade do sistema. O inverso também é verdadeiro, isto é, um baixo valor para instabilidade implica que afirmar que existe um baixo risco de se alterar o comportamento da entidade analisada devido a alterações em outras entidades do sistema.

Portanto, a instabilidade é altamente dependente do nível de acoplamento entre as entidades. Chindamber e Kemerer [1] afirmam que "...qualquer evidência de um objeto usando métodos ou variáveis de outro objeto constitui acoplamento". As medidas de software voltadas ao acoplamento visam analisar a relação entre duas entidades do software, o que pode influenciar diretamente na instabilidade do mesmo, considerando a definição de Martin [2].

Entretanto, a tarefa de avaliar e diagnosticar um software tem sido limitada pela falta de valores de referência para as medidas de software [3]. Este fato é consequência da ampla proposição de medidas de software sem validação e com ausência de valores de referência, informações importantes no auxílio à tomada de decisão [4].

Caso não haja valores de referência, as medidas ainda podem ser usadas com a devida cautela. Deve-se realizar medições, coletar os valores das medidas e manter um histórico do software. Assim, torna-se possível analisar a dinâmica das medidas e dos atributos de software em relação ao tempo e ao longo do desenvolvimento do produto, a fim de identificar melhoria ou deterioração nas medidas e planejar ações de correção, se necessário.

É essencial estabelecer tais valores de referência, a fim de propiciar mecanismos mais robustos para avaliar e diagnosticar se um software está melhorando, piorando ou estável.

Este trabalho analisa a instabilidade de Software pela observação dos valores de referência propostos pela academia e os valores da instabilidade praticados no mercado de software livre. A instabilidade é calculada com base nas medidas de acoplamento aferente e eferente propostas por [2]. Este artigo apresenta na Seção 2 uma definição das medidas de acoplamento aferente, eferente e instabilidade. Em seguida, na Seção 3, apresenta-se a execução da Revisão Sistemática de Literatura (RSL), cujo objetivo foi coletar informações de acoplamento aferente e eferente da literatura acadêmica. Na Seção 4, apresenta-se a análise comportamental da função de instabilidade e uma discussão sobre seus valores máximos e mínimos. Na Seção 5, realiza-se uma análise estatística de um conjunto de 107 softwares de código aberto em 3 versões cada, a fim de identificar as práticas de mercado. As conclusões e trabalhos futuros são apresentados nas Seções 6.

2 Acoplamento aferente, eferente e instabilidade

A medida de instabilidade, proposta por Martin [2], apresenta conceito oposto ao da estabilidade apresentada pela norma ISSO/IEC 25000 [5]. Consequentemente, os valores da estabilidade e instabilidade são inversamente proporcionais. Contudo, ambas possuem o mesmo objetivo, indicar os efeitos potenciais que uma entidade analisada pode sofrer em decorrência as modificações realizadas em outras entidades.

Caso uma entidade tenha baixa instabilidade então tem-se indícios de que a entidade analisada é pouco dependente das demais. Caso contrário, tem-se indícios de que uma reestruturação da entidade analisada poderia ser necessária, pois ela é sensível as alterações ocorridas nas entidades das quais ela depende.

A instabilidade (I) é uma medida indireta dependente das medidas de acoplamento aferente (Ca) e eferente (Ce) [2], determinada pela fórmula:

$$I = \frac{Ce}{Ce + Ca}$$

Segundo Martin [2], a Ca contabiliza o número de classes fora da entidade que dependem da entidade analisada, e Ce contabiliza o número de classes dentro da entidade que dependem de classes fora da entidade analisada.

Estas dependências são definidas como qualquer tipo de relação entre entidades tal que, qualquer alteração na entidade independente implica em alteração na entidade dependente. Havendo dependência, diz-se que há conexão entre as entidades. Na orientação a objetos, a chamada ou definição de métodos, atributos, composição, herança implicam em estabelecer uma dependência entre as classes envolvidas.

Na Fig. 1, apresenta-se um exemplo de diversos acoplamentos aferentes e eferentes entre três pacotes. O valor de Ca e Ce de cada pacote está localizado em um retângulo de borda vermelha junto ao pacote a que ele se refere.

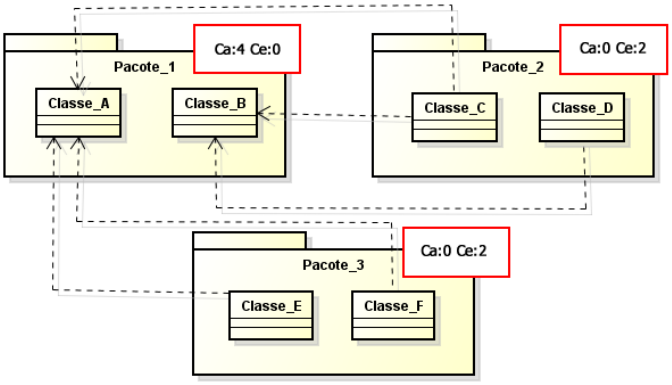

Fig. 1. Exemplo de Acoplamentos Aferentes e Eferentes

No exemplo da Fig. 1 cada pacote é uma entidade a ser analisada quanto a Ca e Ce. Para calcular o Ca do Pacote_1 (Fig. 1), deve-se contabilizar a quantidade de classes externas a esse pacote que possuem dependências incidentes sobre ele. Para isso, deve-se identificar o conjunto de classes que dependem das classes que integram o pacote analisado. Por exemplo, no Pacote_1 tem-se as classes A e B. Os conjuntos que definem as classes dependente de A e B são {C, E, F} e {C, D}, respectivamente. A união destes dois conjuntos representa o conjunto de classes que dependem do Pacote_1, nesse caso, o conjunto união {C, D, E, F}.

Destaca-se que a Classe C (Fig. 1) possui mais de um relacionamento de dependencia incidindo sobre o pacote 1, o que a faz estar presente nos dois conjuntos de dependencia. Entretanto, ainda que uma classe possua mais de um relacionamento de dependência incidente sobre um mesmo pacote, como é o caso da Classe_C, ela será contabilizada uma única vez.

Como Ca refere-se ao total de classes externas ao pacote que referenciam classes internas a ele, então o valor de Ca é o total de elementos presente no conjunto união {C, D, E, F}. Portanto, o Ca do Pacote_1 é igual a 4. Analogamente, faz-se o cálculo de CA para os demais pacotes, obtendo-se o valor zero para o Pacote_2 e o Pacote_3.

Para calcular o valor de Ce, deve-se contabilizar a quantidade de classes internas ao pacote analisado que dependem de classes pertencentes a outros pacotes. Por exemplo, para calcular o valor de Ce do Pacote_2 (Fig. 1), primeiramente, determina-se o conjunto de dependências de cada uma das classes que compõe o Pacote_2. Neste caso, tem-se que as classes C e D dependem respectivamente dos conjuntos de classes {A, B} e {B}. A partir destes conjuntos contabilizam-se quantas classes internas ao Pacote_2 possuem conjuntos de dependência diferentes de vazio, resultando em um Ce igual a 2. Isto é, duas classes pertencentes ao Pacote_2, C e D, apresentam dependências externas.

Além de medir Ca e Ce para entidades consideradas pacotes, é possível calcula-las considerando entidade como sendo o sistema. Para isto, basta somar os valores de Ca e Ce de cada pacote do sistema. No exemplo da Fig. 1, tem-se que o Ca e o Ce do sistema são, respectivamente, 4 e 4.

3 Revisão Sistemática da Literatura (RSL)

A RSL é um método baseado em evidências amplamente utilizado na área da medicina, que recentemente passou a ser largamente utilizado entre pesquisadores da área de Engenharia de Software. Mesmo sendo um estudo dependente de outros, a RSL é extremamente útil, pois reúne diversos estudos com o mesmo tema em um só estudo, auxiliando na execução de trabalhos futuros relacionados com o tema abordado [6]. Além disso, ela auxilia na delimitação da fundamentação teórica, evitando abordagens infrutíferas em uma pesquisa [7]26.

Seguindo o plano de execução de uma RSL [8], nesta Seção apresenta-se o planejamento, a execução e a análise dos resultados desta RSL.

3.1 Planejamento

As questões a serem respondidas ao final dessa RSL são: (i) Quais são os valores de referência sugeridos na literatura para as medidas de software Afferent coupling e Efferent coupling? (ii) Quais são os métodos de cálculo de tais valores para estas medidas?

A *string* de busca gerada é:

```
(("afferent coupling") OR ("efferent coupling")) AND (metric OR metrics) AND ((range OR ranges) OR (interval OR intervals) OR (measure OR measures OR measuring OR measurement) OR (threshold OR thresholds) OR (("reference value") OR ("reference values")) OR (limit OR limits)).
```

O conjunto de bases selecionado está listado na Table 1, juntamente com links destas bases na coluna endereço.

Table 1. Linsta de Bases Selecionadas

Bases	Endereço
IEEE Xplore	http://ieeexplore.ieee.org
Scopus	http://www.scopus.com
Springer	http://www.springer.com/
EI COMPENDEX	http://www.engineeringvillage2.org
Science Direct	http://www.sciencedirect.com/
ACM	http://dl.acm.org/

O idioma adotado para a seleção dos artigos foi o Inglês. Também definiu-se critérios para a seleção e exclusão dos estudos, são eles: (i) Pertencer a um evento e/ou periódico; (ii) Propor ou validar pelo menos uma das medidas trabalhadas; (iii) Propor valores de referência ou fornecer método de medição das medidas Ca e Ce; e (iv) Possuir acesso irrestrito ao seu conteúdo por parte dos autores deste trabalho.

Com o objetivo de obter maior eficiência na busca e reduzir o escopo dos artigos listados e que não eram de interesse, empregou-se diferentes filtros em cada base de busca conforme listado abaixo:

- IEEXplore: i) filtro FullText&Medata para que a ferramenta buscasse em todo o texto em vez de somente no título, palavras chaves e resumo;
- Scopus: selecionado filtros para se limitar: i) à subárea da computação; e ii) à Journals e Conference Proceedings;
- Springer: selecionado filtros para se limitar: i) à subárea da Computação; e ii) a trabalhos no idioma inglês;
- EI Compendex: selecionado filtros para se limitar: i) o parâmetro de busca a Sourcetitle; ii) ao idioma inglês; e iii) os artigos posteriores ao ano de 1994, ano em que as medidas trabalhadas foram propostas;
- Science Direct: selecionado filtros para se limitar: i) à subárea de Ciências da Computação; e ii) somente journals;
- ACM: Nenhum filtro aplicado.

3.2 Execução

Esta RSL foi executada em agosto de 2014. A busca inicial, mostrada na segunda coluna da Table 2, consistiu-se na aplicação da string de busca nas bases de dados.

Durante a execução da seleção primária sobre os estudos retornados pela busca inicial, usou-se a ferramenta JabRef [269] para auxiliar na ordenação e na seleção dos artigos de real importância à revisão. Tal ferramenta foi utilizada para evitar a realização do download das dezenas de trabalhos levantados na busca inicial.

Nessa seleção primária, os artigos foram selecionados mediante a realização da leitura do título, palavras chave e resumo, seguindo os critérios de exclusão explicitados na Seção 3.1. Os resultados podem ser vistos na terceira coluna da Table 2.

Na seleção secundária, cujos resultados estão explicitados nas colunas 4, 5 e 6 da Table 2, realizou-se uma triagem minuciosa pela leitura do título, palavras chaves e resumo, introdução, resultados e conclusão. Também, foram eliminados nessa fase, os trabalhos irrelevantes, repetidos e incompletos.

Ao final da execução das seleções primárias e secundárias, 4 artigos foram selecionados e lidos completamente. Alguns trabalhos citados pelos 4 artigos lidos chamaram a atenção, passando por uma análise semelhante à empregada nas seleções primárias e secundárias. Analisando-se o motivo pelo qual o artigo não estava presente dentre os estudos selecionados na RSL, constatou-se que ele não está indexado em nenhuma base utilizada na RSL. Comprovou-se tal fato buscando o artigo citado em cada base utilizada. Após este procedimento, um artigo foi adicionado e está relacionado na última coluna da Table 2.

Table 2. Resultado da Revisão Sistemática

Bases	Busca Inicial	Seleção Primária	Seleção Secundária			Resultados dos estudos primários	Adicionados
			Irrelevantes	Repetidos	Incompletos		
IEEE	77	7	6	0	0	1	1
Science	27	4	2	0	0	2	
EI Compendex	57	4	2	1	0	1	
Scopus	3	1	0	1	0	0	
Springer	26	2	2	0	0	0	
ACM	40	3	3	0	0	0	
Total	230	21	15	2	0	4	1

Um total de 230 artigos foi encontrado nesta RSL, sendo 33,47% na base da IEEE, 11,73% na base da Science Direct, 24,78% na base da EI Compendex, 1,30% na base da Scopus, 11,30% na base da Springer e 17,39% na base da ACM (Fig. 2).

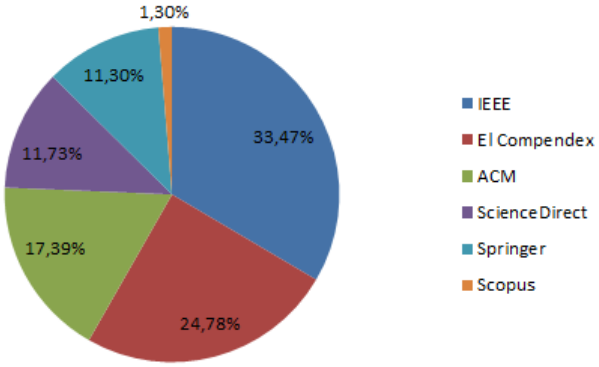

Fig. 2. Porcentagem de Artigos Por Base.

3.3 Análise de Resultados

Nesta etapa, leram-se os artigos selecionados, sumarizando seu conteúdo, e as questões de pesquisa respondidas. Na Table 3, observa-se o título, a referência, o ano e a base de cada artigo selecionado pela RSL. O X representa o artigo não indexado pelas bases utilizadas.

Table 3. Artigos Selecionados pela Revisão Sistemática

#	Título	Referência	Ano	Base
1	OO Design Quality Metrics An Analysis of Dependencies	[2]	1994	X
2	Exploring the Relationships between Design Metrics and Package Understandability A Case Study	[10]	2010	IEEE Xplore
3	Empirical comparison of three metrics suites for fault prediction in packages of object-oriented systems A case study of Eclipse	[11]	2011	Science
4	Investigation of Aspect-Oriented Metrics for Stability Assessment: A Case Study	[12]	2011	EI Compendex
5	Identifying thresholds for object-oriented software metrics	[13]	2012	Science

OO Design Quality Metrics An Analysis of Dependencies [2].

Esse artigo propôs e descreveu um conjunto de medidas de software para mensurar a qualidade de projetos OO em função das dependências entre os subsistemas do projeto. Essas dependências podem ser boas ou ruins, dependendo do nível de estabilidade da entidade.

Entre as medidas propostas e definidas estão o acoplamento aferente e eferente, a instabilidade, a abstração, e a distância da sequência principal. Apesar do autor propor tais medidas, ele não apresenta valores de referência para Ca e Ce.

O autor defende que as medidas de software não são um fator que se pode confiar cegamente. Elas podem ser boas ou ruins para medir a conformidade com o padrão de projeto desejado, devendo-se utilizar uma medida adequada para cada padrão de projeto.

Exploring the Relationships between Design Metrics and Package Understandability: A Case Study [10].

Esse artigo explora o relacionamento das medidas Number of classes (NC); Ca; Ce; instabilidade; Distance (D) aplicáveis sobre pacotes de software. Na condução da investigação o autor aplicou as medidas citadas sobre 80 pacotes de software, realizando diversas análises estatísticas.

O trabalho em questão não determinou um intervalo ou um valor típico para as medidas, tampouco determinou o método de cálculo destas medidas. Entretanto, o autor fez uma análise relacionando as medidas trabalhadas e o esforço para compreensão de um pacote.

Dentre os resultados obtidos, pode-se citar a falta de correlação entre Ca e o esforço requerido para a compreensibilidade do pacote. Além disso, pode-se afirmar que Ce está correlacionado com a alta reusabilidade dos serviços providos por outros pacotes.

Os resultados finais obtidos pelo autor, a partir desse estudo, mostram correlação significativa entre a maioria das medidas por ele analisadas e a compreensibilidade de um pacote, excetuando-se Ca e instabilidade.

Empirical comparison of three metrics suites for fault prediction in packages of object-oriented systems: A case study of Eclipse [11].

Este artigo avalia e compara empiricamente três suítes de medidas de software (Martin, MOOD e CK), e faz uma análise de predição relacionando Ca e Ce com a quantidade de falhas de pré-lançamento e pós-lançamento de um software. O resultado indica que quanto maior os valores apresentados por Ca e Ce, maiores as quantidades de falhas pré e pós lançamento.

O autor conduz um estudo de caso abrangendo três diferentes versões do software Eclipse (2.0, 2.1 e 3.0) para validar seu trabalho.

No artigo os autores não determinaram valores de referência ou o método de cálculo para as medidas Ca e Ce. Os autores concluem que, dentre os suítes de medidas analisadas, o suíte de Martin fornece a maior precisão para prever as falhas pré-lançamento e pós-lançamento, recomendando a utilização deste suíte.

Investigation of Aspect-Oriented Metrics for Stability Assessment: A Case Study [12].

Esse artigo teve como objetivo avaliar o relacionamento de 13 medidas Orientadas a Aspectos (OA) e a estabilidade de software. Um estudo de caso foi realizado, aplicando as 13 medidas sobre o software Glassbox, que possui 76 aspectos.

O artigo não apresenta valores ou intervalos de valores absolutos para as medidas trabalhadas. Entretanto, realizou uma análise estatística para determinar se as medidas Ca e Ce interferem na estabilidade de um software OA. Após aplicar uma análise de

correlação, afirmou-se que Ca e Ce não correlacionam-se com a quantidade de revisões dos aspectos definidos dentro do Glassbox.

Os autores concluíram que a maioria das 13 medidas analisadas possui correlação significativa com a estabilidade de software OA.

Identifying thresholds for object-oriented software metrics [13].

Nesse artigo, os autores objetivaram definir intervalos de referência para um conjunto de 6 medidas de software orientados a objetos (OO). Para a condução deste estudo, aplicou-se as medidas selecionadas em um conjunto de 40 softwares *open* source. Posteriormente, realizou uma análise probabilística e definiu três intervalos de valores possíveis definidos como: (i) Bom: valores mais comuns para a medida; (ii) Regular: valores com baixa frequência, mas não irrelevantes; (iii) Ruim: valores com rara probabilidade de ocorrer.

Apesar desse artigo sugerir intervalos de referência para a medida Ca, os autores empregaram uma análise para a medida de Ca a nível de classe, não adotando a definição de Martin. Portanto, os autores mediram, na realidade, o Fan-in sob a nomenclatura de Ca, impossibilitando a comparação entre elas, visto que a medida Fan-in refere-se à análise dos acoplamentos aferentes em nível de classe e a medida Ca refere-se à análise dos acoplamentos aferentes em nível de pacote [2].

Os autores concluíram que os valores propostos para as medidas avaliadas são úteis, podendo ajudar engenheiros e gerentes de software a avaliar seus projetos e verificar se o software viola alguma diretriz de projeto.

3.4 Sumarização dos Resultados

A Table 4 apresenta um resumo das respostas referentes às questões da RSL. A coluna intitulada "Valores de Referência" responde a primeira pergunta da RSL: Quais são os valores de referência sugeridos na literatura para as medidas de software Ca e Ce?

A coluna intitulada "Método de Cálculo" responde a segunda questão da RSL: Quais são os métodos de cálculo de tais valores para estas medidas?

Em cada linha da tabela apresenta-se a resposta retirada de cada artigo selecionado por esta RSL.

Table 4. Sumarização dos Resultados da RSL

Artigos	Valores de Referência		Método de Cálculo	
	Ca	Ce	Ca	Ce
[2]	Não Apresenta	Não Apresenta	Ca=∑dependências A, onde dependências A, são as dependências de classes externas a categoria que dependem da categoria;	Ce=∑dependências E, onde dependências E são as dependências da categoria sobre classes externas à categoria;
[10]	Não Apresenta	Não Apresenta	Não Apresenta	Não Apresenta
[11]	Não Apresenta	Não Apresenta	Não Apresenta	Não Apresenta
[12]	Não Apresenta	Não Apresenta	Não Apresenta	Não Apresenta
[13][1]	Tamanho (#classes) / Intervalos bom/regular/ ruim) ≤100 0–1/2–20/>20 101–1000 0–1/2–20/>20 >1000 0–1/2–15/>15	Não Apresenta	Não Apresenta	Não Apresenta

Alguns fatores podem ameaçar a validade dessa RSL. Dentre elas, artigos não terem sido encontrados durante as buscas. As causas para eles não terem sido encontrados são: (i) utilização de outra nomenclatura para Ca e Ce; (ii) artigo não indexado em nenhuma das bases pesquisadas; e (iii) o título, resumo e palavras chaves serem descritos de maneira inadequada, induzindo a não seleção dele.

4 Análise da Instabilidade

Nesta seção apresenta-se uma análise da função de Instabilidade e dos valores mínimos e máximos de Ca e Ce.

4.1 Análise comportamental da Função de Instabilidade

Segundo Martin [2], para que uma entidade possua estabilidade a mesma deve ser: (i) Independente: uma entidade que não possui dependências em relação a outras partes do sistema; (ii) Responsável: entidade que possui diversas outras entidades dependendo dela.

A partir desta definição e da fórmula da Instabilidade explicada na Seção 2, uma análise comportamental da função foi realizada. Para isso aplicou-se os valores de Ca e Ce variando no intervalo real de 0 a 10 para se obter o valor da instabilidade. Nos eixos que variam de 0 a 10, pode-se imaginar variando de 0 a N, o importante é entender como a Instabilidade varia em função da variação de Ca e Ce.

[1] O valor de referência de Ca para o artigo [13] não foram calculados segundo os critérios de Martin, adotados neste trabalho. Por isto, eles são apresentados mas não são utilizados para comparação neste artigo.

O resultado pode ser visto na Fig. 3. No eixo horizontal tem-se os valores da medida Ca, no eixo de profundidade tem-se os valores da medida Ce, e no eixo vertical tem-se valores da medida instabilidade.

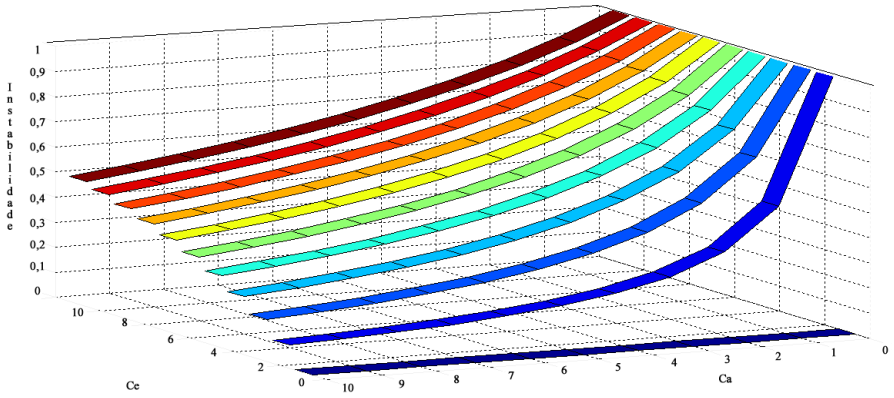

Fig. 3. Análise comportamental da Instabilidade em relação a Ca e Ce.

Analisando o gráfico, observa-se que:

- Para um valor de Ce igual a 0, independente da variação apresentada por Ca, o valor de I será 0;
- Fixando o valor de Ce e aumentando o valor de Ca, tem-se uma diminuição do valor de I;
- Fixando o valor de Ce e diminuindo o valor de Ca, ocasiona o aumento do valor de I;
- Fixando o valor de Ca e aumentando o valor de Ce, tem-se o aumento do valor de I;
- Mantendo Ca constante e diminuindo o valor de Ce, obtém-se a diminuição do valor apresentado pela medida I.

Com base na análise do Fig. 3, pode-se afirmar que quanto maior Ce em detrimento de Ca, maior será a instabilidade da entidade analisada. Quanto maior Ca em detrimento de Ce, menor a instabilidade do software.

4.2 Valores mínimos e máximos para Ca e Ce

Para melhor compreender a função de instabilidade, realizou-se a análise dos valores mínimos e máximos possíveis para as medidas Ca e Ce, em relação a um pacote perante os demais pacotes que formam um sistema. Tais valores foram determinados de acordo com os conceitos definidos por Martin [262] e citados na Seção 2.

A Table 5 apresenta os valores mínimos e máximos para as medidas Ca e Ce. O acrônimo NC representa a quantidade total de classes existentes no sistema menos a quantidade de classes presentes na entidade analisada.

Table 5. Valores Mínimos e Máximos de Ca e Ce

	Valor Mínimo	Valor Máximo
Ca	0	NC
Ce	0	NC

A medida Ca tem como valor mínimo 0, pois um pacote pode não possuir classes dependente dele. O valor máximo é igual a NC, pois todas as classes presentes no sistema podem ser dependentes do pacote analisado.

Assim como Ca, a medida Ce apresenta o valor mínimo igual a 0, isso se dá quando o pacote analisado não depende de outras classes externas presentes em outros pacotes. O valor máximo de Ce é dado por NC, pois o pacote pode possuir dependências a todas as classes no sistema, excetuando-se as presentes no próprio pacote.

5 Práticas de Mercado de Softwares *Open Source*

Nesta seção, apresentam-se os resultados obtidos da análise dos valores aplicados pelo mercado de softwares *open source* para as medidas Ca, Ce e instabilidade. Essa análise inclui uma análise estatística contendo estatísticas descritivas, o teste de normalidade, o teste de médias e por último apresenta-se uma análise exploratória sobre as medidas trabalhadas.

O conjunto de dados selecionados para analisar as práticas de mercado foi composto de 107 softwares open source em três versões cada, desenvolvidos em Java e obtidos no repositório SourceForge [14]. Estes softwares foram selecionados de maneira aleatória no repositório, entretanto seguindo alguns critérios de seleção: (i) ser desenvolvido em Java; (ii) disponibilidade do código fonte Java; (iii) possuir três versões disponíveis; e (iv) versão mais recente ter sido lançada após 2010.

Para coletar as medidas dos 107 softwares, foram relacionadas 10 ferramentas livres. Os critérios para seleção das ferramentas foram: apresentar o resultado por pacote, executar análise sobre o código fonte Java e ser uma ferramenta atualizada, tendo sua última versão posterior a 2011. Medir adequadamente as medidas, de acordo com o proposto por Martin [2]. O resultado desta aferição pode ser visto na Table 6.

Nessa tabela, a coluna intitulada "Ferramentas" apresenta o nome das ferramentas aferidas e a coluna intitulada "Resultado da Aferição" exibe se a ferramenta foi aceita (apresentou todos os critérios de seleção) ou rejeitada (não atendeu pelo menos um dos critérios de seleção). Por fim, a coluna intitulada "Motivo" na qual se justifica o principal motivo de rejeição da ferramenta.

Table 6. Resultado da Aferição das Ferramentas de Medição

Ferramentas	Resultado Aferição	Motivo
AnalysT4j	Rejeitada	Ferramenta Desativada
CCCC	Rejeitada	Ferramenta Desativada
CodePro	Aceita	X
CKJM	Rejeitada	Não executa análise sobre os pacotes
DependecyFinder	Rejeitada	Não suporta arquivos em Java
Jdepend	Rejeitada	Medição inadequada das medidas
Metrics1.3.8	Rejeitada	Não suporta arquivos em Java
Ndepend	Rejeitada	Não suporta arquivos em Java
Refactor IT	Rejeitada	Somente projetos com menos que 50 classes

5.1 Análise Estatística

Após definir a ferramenta, executou-se o processo de medição dos 107 softwares. Em seguida, aplicou-se uma análise estatística, utilizando as ferramentas R-Software e sua interface RStudio [9].

Estatística Descritiva

O resultado da estatística descritiva para os 107 softwares em suas 3 versões, totalizando 321 tratamentos pode ser visto na Table 7.

Na segunda coluna da Table 7, intitulada "Ca", exibe-se os valores da estatística descritiva de Ca nas três versões dos softwares (V1, V2, V3), e em cada linha apresenta-se uma medida estatística diferente. O mesmo é apresentado para Ce na terceira coluna e instabilidade da quarta coluna.

Table 7. Estatística Descritiva das medidas Ca, Ce e Instabilidade

Estatísticas	Ca			Ce			Instabilidade		
	V1	V2	V3	V1	V2	V3	V1	V2	V3
Media	36,3	49,0	73,7	140,6	188,7	198,8	0,7	0,7	0,7
Erro padrão médio	11,2	12,4	16,5	18,7	22,2	22,7	0,0	0,0	0,0
Mediana	0,0	0,0	0,0	57,8	100,0	126,0	0,8	1,0	0,7
Moda	0,0	0,0	0,0	0,0	0,0	0,0	1,0	1,0	1,0
Desvio Padrão	115,8	128,3	171,2	193,4	229,5	235,3	0,4	0,4	0,4
Mínimo	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Máximo	804	843	916	881	950	963	1,0	1,0	1,0
25%	0,0	0,0	0,0	1,0	1,4	1,4	0,5	0,6	0,6
Quartis 50%	0,8	0,0	0,0	57,8	100,0	126,0	0,8	1,0	0,7
75%	15,0	31,0	62,0	192,3	316,0	302,0	1,0	1,0	1,0

Efetuando-se a análise dos dados apresentados na Table 7, nota-se que as medidas analisadas possuem uma distribuição assimétrica, pois a média, a mediana e a moda não são iguais, considerando uma mesma medida em uma mesma versão.

Observa-se um alto desvio padrão para as medidas, revelando alta dispersão dos valores em relação à média.

Teste de Normalidade.

Aplicou-se o teste de normalidade denominado Kolmogorov-Smirnov, com significância de 5%. Testando as seguintes hipóteses: (i) H0, os dados têm distribuição normal; (ii) H1, os dados não têm distribuição normal.

A Table 8 apresenta o resultado da execução desse teste. Nela, mostra-se que as medidas Ca, Ce e de instabilidade não tem distribuição normal, pois os valores apresentados na coluna "Sig." são menores que o p-valor de 0,05. Na primeira coluna tem-se as medidas trabalhadas, na segunda coluna apresenta-se o teste de Kolmorgorov-Smirnov com os valores da estatística, os Graus de Liberdade (g.l) e a significância do mesmo.

Table 8. Testes de Normalidade

	Kolmogorov-Smirnov		
	Estatística	g.l	Sig.
Ca	0,353	321	0,000
Instabilidade	0,280	321	0,000
Ce	0,213	321	0,000

Teste de Médias.

O teste de médias definido foi o teste não paramétrico denominado Kruskall-Wallis [15], considerando que os dados não possuem distribuição normal. Tal teste exige independência dos dados, podendo ser aplicado somente entre (Between) os tratamentos (softwares), pois estes são independentes. Não aplicou-se o teste dentro (Within) de cada tratamento (versões do software), pois tratam-se de amostras dependentes, isto é, a versão X de um software influência a versão X+1, por que este último é a evolução do anterior.

O objetivo foi verificar se os softwares apresentam diferenças significativas para as medidas I, Ca e Ce, considerando a mesma versão de cada software. A confiabilidade adotada foi de 95%.

As hipóteses para esse teste foram: (i) H0: a medida é estatisticamente igual para os 107 softwares; (ii) H1: existe pelo menos um software, que possui um valor diferente para a medida entre os 107 softwares.

O resultado obtido pode ser visto na Table 9, que exibe em suas colunas a versão dos softwares, valor do Kruskal-Wallis Qui-quadrado, os graus de liberdade do teste, e o p-valor, respectivamente.

Table 9. Teste de Médias Kruskall- Wallis

Versão	Kruskal-Wallis Qui-quadrado	g.l	p-valor		
			I	Ca	Ce
V1	106	106	0.4817	0.4817	0.4817
V2	106	106	0.4817	0.4817	0.4817
V3	106	106	0.4817	0.4817	0.4817

O resultado desse teste indica que não existem diferenças significativas entre os softwares nas três versões, pois os valores apresentados por p-valor são maiores que a significância determinada (0,05). Portanto aceita-se H0 e afirma-se que a variação da instabilidade entre os softwares não apresentam diferença significativa. Analogamente, o mesmo pode ser afirmado para Ca e Ce.

5.2 Análise Exploratória de I, Ca e Ce Entre Versões

Essa análise exploratória teve por objetivo responder a questão: Qual a variação de instabilidade, Ca e Ce entre as versões de um mesmo software?

Para executar essa análise, primeiramente calculou-se a variação da instabilidade entre as versões de um mesmo software, subtraindo-se o valor da instabilidade entre uma versão mais recente e outra mais antiga. Aplicou-se esse procedimento as combinações possíveis entre as versões como: versão 1 para a versão 2 (V1→V2), da versão 2 para a versão 3 (V2→V3) e da versão 1 para a versão 3 (V1→V3).

Em seguida, classificou-se o valor da variação da instabilidade como positiva, nula, ou negativa. Se a instabilidade aumentou de uma versão para outra, a variação foi positiva, se houve diminuição, a variação foi negativa, e no caso da variação valer zero, ela foi denominada de nula.

Posteriormente, calculou-se a distribuição de frequência dessas variações, classificando-as em intervalos de classe com 10% do total de possíveis valores para a medida analisada. Para determinar o tamanho do intervalo de classe, somaram-se o menor e o maior valor apresentado pela medida e, em seguida, dividiu-se a soma por 10, determinando-se assim a amplitude de cada intervalo.

A frequência desses intervalos de classe para a medida instabilidade pode ser vista na Table 10. Apresenta-se na primeira coluna os intervalos de classe. A segunda, terceira e quarta colunas exibem a frequência dos intervalos de classe referente a comparação entre versões.

Durante a análise, o valor zero foi colocado em um intervalo independente (50%-50%), veja linha destacada na tabela, pois sua alta incidência interferiria na análise de frequência dos intervalos (Table 10).

Table 10. Frequência Relativa da Variação da Instabilidade Entre Versões

Porcentagem do Intervalo	Frequência Relativa		
	V1→V2	V2→V3	V1→V3
]0%-10%]	7%	6%	7%
]10%-20%]	0%	0%	0%
]20%-30%]	2%	3%	2%
]30%-40%]	3%	1%	4%
]40%-50%]	13%	16%	16%
]50%-50%[	52%	59%	50%
[50%-60%]	7%	9%	7%
]60%-70%]	4%	2%	5%
]70%80%]	1%	1%	2%
]80%-90%[	2%	0%	2%
]90%-100%[	9%	4%	7%

O Fig. 4 e o Fig. 5 apresentam o resultado dessa análise referente a medida de instabilidade.

No Fig. 4, observa-se a frequência das variações positivas, nulas e negativas entre cada versão. Por exemplo, no primeiro valor do eixo x, denominado "1-2", observa-se que 52% das variações de instabilidade entre as versões 1 e 2 dos 107 softwares foi nula, enquanto as variações positivas e negativas foram 23% e 24% respectivamente.

Nota-se que a variação nula apresentou frequência mínima de 50% nos três casos de variação das versões (1-2, 2-3, 1-3) apresentadas no gráfico. Conclui-se que, em mais da metade dos casos, a Instabilidade não variou quando o software mudou de versão. Pode-se observar também que há um equilíbrio entre as variações positivas e negativas, as quais tiveram valor menor ou igual a 25% na maior parte dos casos.

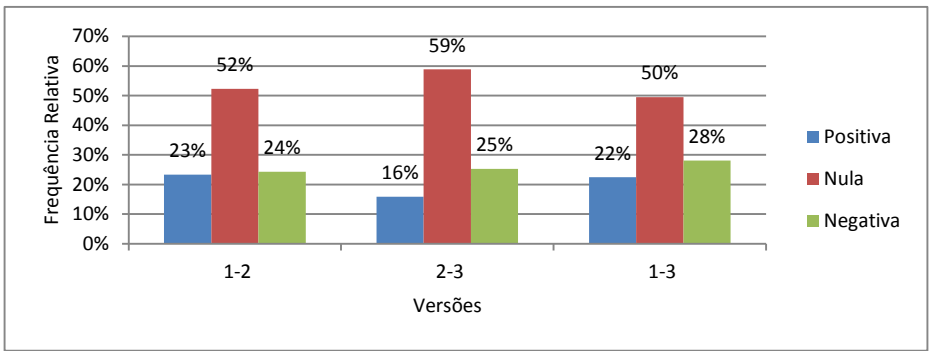

Fig. 4. Análise da Variação da Instabilidade Entre Versões.

No Fig. 5, apresenta-se a frequência relativa da variação da instabilidade como mostrada na Table 10. Pode-se observar que no intervalo de classe denominado "50%-50%", que equivale a variação zero do valor da instabilidade, obteve frequencia maior de 50% em todas as comparações de versões.

Os intervalos de classe "40%-50%" e "50%-60" representam as variações de instabilidade de até 10% positiva ou negativa, bastando somar os rótulos dessas classes para

se obter a frequência relativa total. Por exemplo, no Fig. 5, observa-se que aproximadamente 73% dos 107 softwares apresentaram uma variação de instabilidade menor ou igual 10% entre as versões v1→v2 e v1→v3.

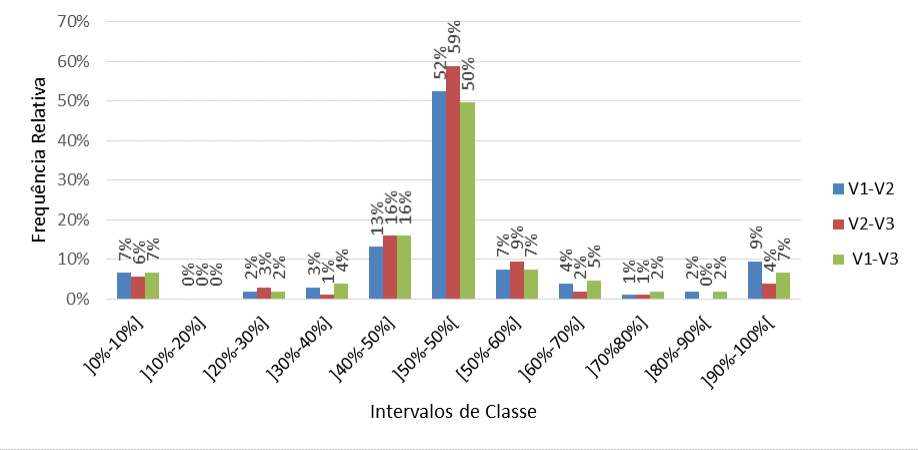

Fig. 5. Análise da Variação da Frequência da Instabilidade Entre Softwares de Uma Mesma Versão.

Essa mesma metodologia de análise foi aplicada à medida Ca, como pode ser observado na Table 11, na Fig. 6 e na Fig. 7.

Table 11. Frequência Relativa da Variação de Ca Entre Versões

Porcentagem do Intervalo	Frequência Relativa		
	V1→V2	V2→V3	V1→V3
]0%-10%]	1%	1%	1%
]10%-20%]	0%	0%	0%
]20%-30%]	0%	0%	0%
]30%-40%]	0%	0%	0%
]40%-50%]	1%	20%	13%
]50%-50%[	66%	65%	62%
[50%-60%]	17%	10%	17%
]60%-70%]	12%	1%	3%
]70%80%]	12%	1%	2%
]80%-90%[	1%	1%	0%
]90%-100%[	0%	1%	3%

Assim como para a Instabilidade, nota-se que a medida Ca possui grande porcentagem de casos sem variação entre as versões (Fig. 6) na maioria dos softwares analisados.

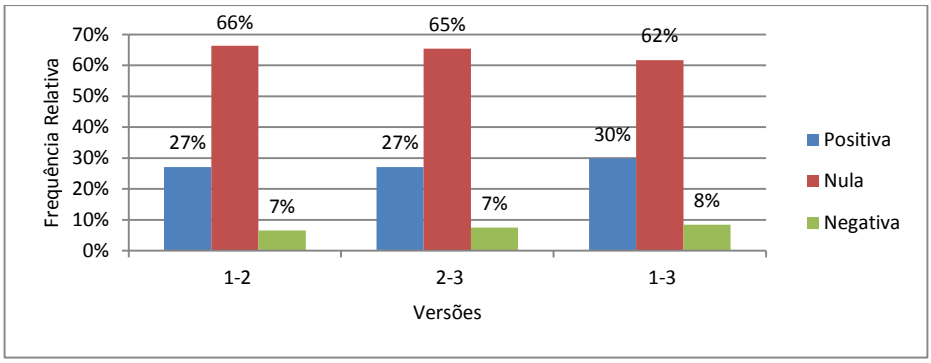

Fig. 6. Análise da Variação de Ca Entre Versões de um Mesmo Software.

Entretanto, a porcentagem de casos em que Ca apresentou variação positiva é superior à porcentagem de variação negativa (Fig. 6). Este comportamento revela uma tendência da elevação de Ca com a evolução do software. Essa afirmação é comprovada pelo Fig. 7, cuja frequência presente nos intervalos que expressam variação positiva é superior a aqueles que indicam variação negativa entre as versões, excetuando-se o intervalo onde os valores se mantiveram constantes.

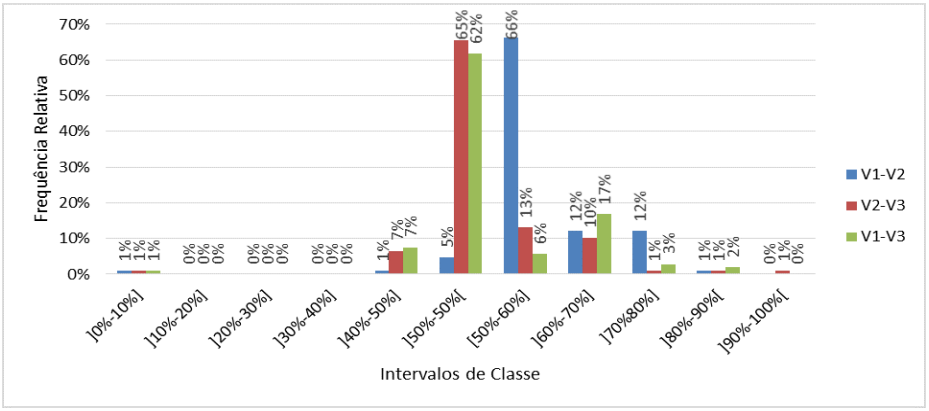

Fig. 7. Análise da Variação da Frequência de Ca Entre Softwares de Uma Mesma Versão.

Assim como feito para as demais medidas, aplicou-se a mesma metodologia de análise para medida Ce. O resultado é apresentado na Table 12, na Fig. 8 e na Fig. 9.

Table 12. Frequência Relativa da Variação de Ce Entre Versões

Porcentagem do Intervalo	Frequência Relativa		
	V1→V2	V2→V3	V1→V3
]0%-10%]	3%	4%	2%
]10%-20%]	2%	2%	0%
]20%-30%]	3%	1%	5%
]30%-40%]	7%	1%	5%
]40%-50%]	52%	26%	36%
]50%-50%[	17%	18%	15%
[50%-60%]	7%	42%	25%
]60%-70%]	3%	1%	6%
]70%80%]	4%	3%	2%
]80%-90%[	1%	1%	3%
]90%-100%[	1%	2%	2%

Em relação à medida Ce nota-se que a porcentagem de variações positivas sobressai em relação as variações negativas e nulas (Fig. 8). Isto implica num crescimento da quantidade de dependências sobre serviços oferecidos por entidades externas à entidade analisada, podendo acarretar no aumento da instabilidade.

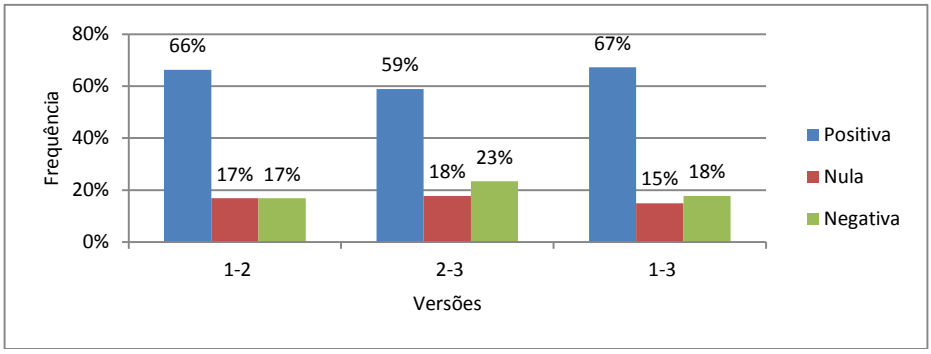

Fig. 8. Análise da Variação de Ce Entre Versões de um Mesmo Software.

Apesar da alta variação positiva de Ce entre as versões, essa variação apresentou baixa amplitude. Seu intervalo variou 10% para mais ou para menos em relação ao intervalo central representado pelo valor zero ou nulo (intervalo de classe 50%-50%) (Fig. 9).

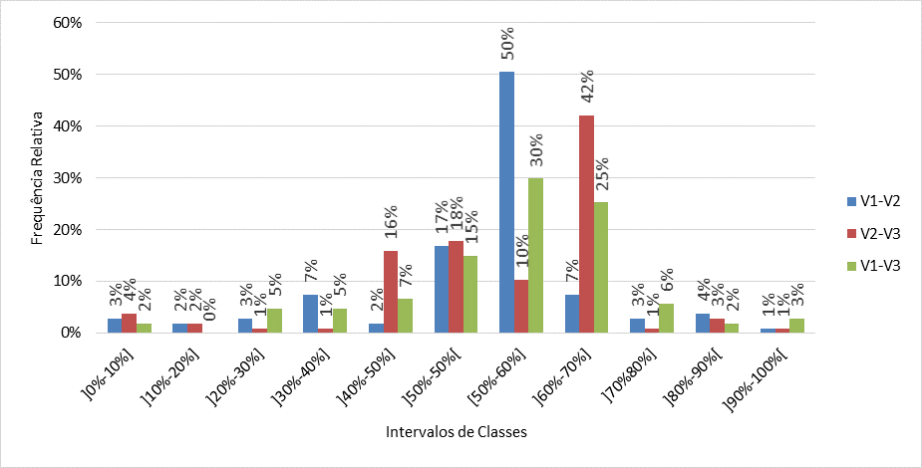

Fig. 9. Análise da Variação da Frequência de Ce Entre Softwares de Uma Mesma Versão.

5.3 Distribuição de Frequência das Práticas de Mercado

Em relação à medida instabilidade, nota-se que 48% dos softwares produzidos atualmente possuem valor igual a 1, valor máximo permitido para a instabilidade. Portanto, 48% dos softwares presentes no mercado podem ser considerados altamente instáveis segundo a definição de Martin [2].

Na Fig. 10, a barra referente a instabilidade de valor 1 foi retirada, a fim de facilitar a visualização, considerando que a instabilidade 1 obteve 48% de frequência.

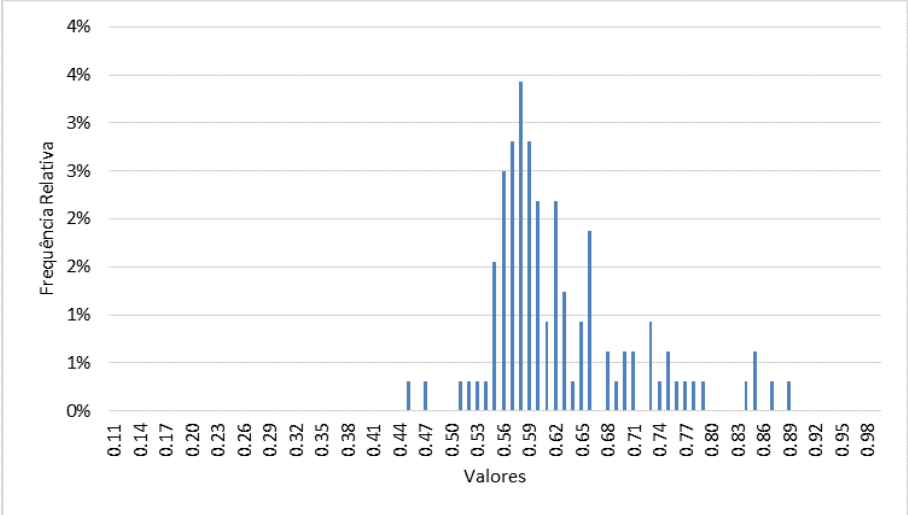

Fig. 10. Frequência da instabilidade nos 107 Softwares em suas 3 versões

Na Fig. 11 apresenta-se a frequência relativa dos valores de Ce dos 107 softwares analisados em suas três versões. Nota-se que o valor um (1) é o mais praticado pelo mercado atualmente, com uma frequência relativa de 21%. A barra referente ao valor de Ce igual a 1 foi omitida para facilitar a visualização das demais barras no gráfico.

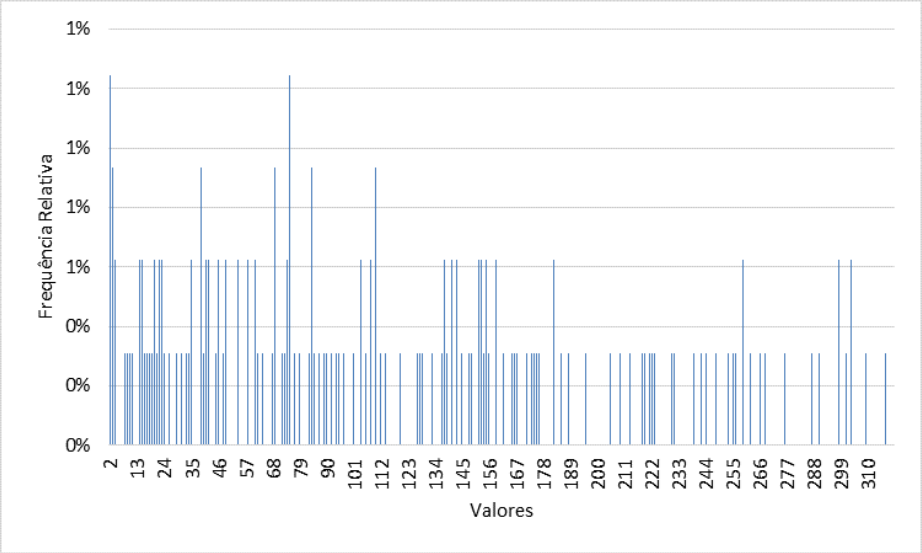

Fig. 11. *Frequência dos Valores de Ce nos 107 Softwares*

Na Fig. 12, apresenta-se a frequência relativa dos valores de Ca dos 107 softwares analisados em suas três versões. Nota-se que o valor zero (0) foi o mais praticado no mercado com 69% de frequência relativa. A barra referente ao valor de Ca igual a zero foi omitida para facilitar a visualização das demais barras no gráfico.

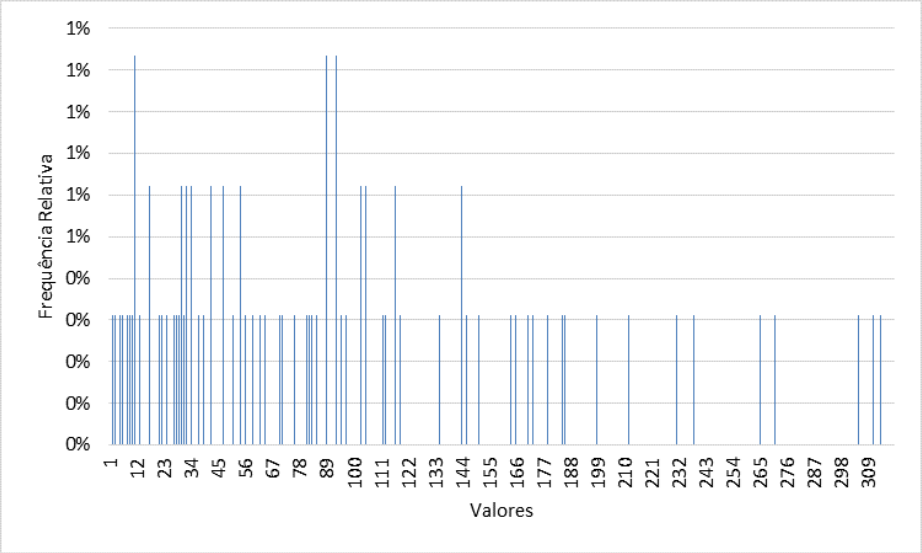

Fig. 12. *Frequência dos Valores de Ca nos 107 Softwares*

6 Conclusão

O objetivo desde trabalho foi realizar uma análise da instabilidade baseado nas medidas Ca e Ce. Para isso, primeiramente realizou-se uma RSL para identificar os valores de referência e os métodos de cálculo de Ca e Ce. Posteriormente, foram analisadas as práticas adotadas pelo mercado de software *open source*, em relação às medidas Ca, Ce e instabilidade.

Em relação ao estado da arte para as medidas Ca e Ce, nota-se uma escassez de trabalhos definindo o método de cálculo ou propondo intervalos de referência para tais medidas, assim como qualquer outra informação que dê suporte a tomada de decisão de Engenheiros e Gerentes de Software ao longo do desenvolvimento de um projeto.

Este fato é confirmado pela RSL executada onde não houve artigos propondo valores de referência para Ca e Ce. Somente um artigo abordou o método de cálculo de Ca e Ce. Isto significa que somente 0,31% dos artigos obtidos na busca inicial realmente continham o conteúdo desejado. Entretanto, foi possível observar que todos os artigos analisados utilizaram a definição original das medidas, conferindo as mesmas uma grande robustez em relação a sua definição, fato que é muito divergente nas demais medidas existentes na literatura.

A análise das práticas de mercado indicou que as medidas instabilidade e na tendem a não variar ou variar muito pouco entre as versões dos softwares. A medida Ce apresentou tendência de elevação de seus valores com a evolução das versões do software.

Embasado nas análises realizadas neste trabalho, conclui-se que as medidas Ca e Ce têm um impacto direto sobre a instabilidade do software. Sendo que Ca possui um peso

maior neste impacto, pois sua modificação leva a uma alteração de maior magnitude na instabilidade. Já para que Ce gere alterações significativas no valor da Instabilidade, a amplitude de sua variação deve ser mais significativa, comparando-se com Ca.
Como trabalhos futuros sugere-se fazer uma análise de correlação entre Ca, Ce e I em relação a frequência de erros encontrados em um sistema de software, bem como o custo de erros.

7 Referências

1. Chindamber S. and Kemerer C. F. “Towards a Metrics Suite for object-oriented Design. In Proc. OOPSLA '91, ACM,” Conference proceedings on Object-oriented programming systems, languages, and applications, pp. 197-211, 1991.
2. Martin R. "OO design quality metrics." An analysis of dependencies, 1994. [Online]. Available: http://www.objectmentor.com/resources/articles/oodmetrc.pdf. [Último acceso: 27 2014 2014].
3. Tempero H., Baxter G., Noble J. and Frean M. “Understanding the shape of java software,” OOPSLA’06, pp. 397-412, 22–26 October 2006.
4. Fenton N. and Neil M., “Software Metric: Roadmap,” Future Software Enginering, pp. 357-370, 2000.
5. ISO/IEC 25000, «Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) » 2014.
6. Linde K. and Willich S., «How objective are systematic reviews? Differences between reviews on complementary medicine» JRSM (Journal of the Royal Society of Medicine), vol. 96, nº 3, pp. 156-157, March 2003.
7. Justus R., «A guide to writing the Dissertation Literature Review» Pratical Assesement Rsearch & Evaluation, p. 13, Junho 2009.
8. Biolchhini J. E. A., «Systematic review in software engineering» UFRJ, Rio de Janeiro, 2005.
9. JabRef, 2014. [En línea]. Available: http://jabref.sourceforge.net/. [Último acceso: 2014 janeiro 18].
10. Elish M. O., “Exploring the relationships between design metrics and package understandability: A case study” 2010.
11. Muhammed A. M., Elish M. O. and Al-Yafei A. H., “Empirical comparison of three metrics suites for fault prediction in packages of object-oriented systems: A case study of Eclipse,,” Advances in Engineering Software, vol. 42, nº 10, pp. 852-859, 10 October 2011.
12. Elish M. O., Mohammad A.M. and Al-Khiaty M., “Investigation of Aspect-Oriented Metrics for Stability Assessment: A Case Study” Journal of Software, vol. 6, nº 12, pp. 2508-2514, 2011.
13. Ferreira K. A., Bigonha M. A., Bigonha R. S., Mendes L. F. and Almeida H. C., “Identifying thresholds for object-oriented software metrics.",” Journal of Systems and Software 85.2:, pp. 244-257., 2012.
14. Sourceforge, “SOURCEFORGE. Find, create, and publish open source software for free,” 2014. [Online]. Available: <http://sourceforge.net/. [Acesso em 03 Aug 2014].
15. Kruskall W. L. and Wallis D. H., "Use of ranks in one-criterion variance analysis," Journal of the American Statistical Associates v. 47, n. 260, pp. p. 583-621, 1952.