



## Utilizando Métricas para Observação do Decaimento Arquitetural na Evolução Contínua do Software

Guilherme Sbrolini Mazzariol, Breno Bernard Nicolau de França

### Resumo

A utilização de métricas de acoplamento e coesão para observação do decaimento arquitetural na evolução contínua de um software identifica momentos em que uma intervenção no código deve ser considerada, a fim de facilitar a sua manutenção e desenvolvimento futuros.

### Palavras-chave:

decaimento arquitetural, métricas, acoplamento, coesão, software.

### Introdução

Durante o desenvolvimento de um software, podem ocorrer alterações arquiteturais como, por exemplo, a substituição de uma tecnologia por outra que tenha se mostrado mais eficiente, mas pouco se discute sobre qual o impacto dessa mudança no decorrer do desenvolvimento do software. Neste estudo, procuramos avaliar quanto um software, durante o seu ciclo de vida, deixa de seguir os princípios e boas práticas definidas no projeto de sua arquitetura, observando características de má qualidade da arquitetura através de métricas de coesão e acoplamento extraídas do seu código fonte.

Utilizamos uma ferramenta existente para extrair as métricas e desenvolvemos um software em linguagem de programação Java para manipular e apresentar essas métricas e suas variações durante um período de tempo pré-determinado. Os softwares analisados neste estudo são *open-source* e também foram (e ainda estão sendo) desenvolvidos em linguagem de programação Java, utilizando práticas de entrega contínua.

### Resultados e Discussão

Após a análise de diferentes ferramentas para extração de métricas, optamos pela utilização do programa *JHawk* [1]. Nossa escolha se baseou nas métricas de coesão e acoplamento suportadas por esse programa e pela possibilidade de executá-lo em linha de comando, sem interação com a interface gráfica do usuário. O *JHawk* permite a extração de métricas em três níveis: pacote, classe e método. Neste estudo, utilizamos os três níveis para uma abordagem mais abrangente, extraindo os dados em formato CSV para análise. Ainda, desenvolvemos um programa em linguagem de programação Java para realizar as análises dos softwares em três etapas: extração, agregação e apresentação dos resultados em um determinado período de tempo. Cada software analisado neste estudo foi desenvolvido utilizando entrega contínua, ou seja, novas versões são disponibilizadas com alta frequência e seu código fonte pode ser acessado através da plataforma <https://github.com/>.

Na etapa de extração, utilizamos o *JHawk* para coleta das métricas em cada alteração do software disponível no período analisado. Em seguida, seguimos para a etapa de agregação, em que extraímos as métricas de coesão e acoplamento pré-definidas, escolhidas previamente após análise da documentação do *JHawk*, e montamos uma estrutura de dados com

essas informações. Na etapa de apresentação, mostramos as métricas analisadas em gráficos, exibindo-as por pacote, em seguida por métrica e por classe do pacote. Essa última parte pode ser exibida de duas formas distintas: todas as métricas analisadas ou apenas as métricas que variaram no período.

Os resultados mostram que é possível observar um decaimento arquitetural na evolução contínua de um software através da observação de métricas. Por exemplo, analisando a métrica *CBO* (*Coupling Between Objects*), que mede acoplamento entre objetos, encontramos casos em que ela se apresenta constante, aumenta ou diminui (Figura 1) e, quanto maior for o valor do *CBO*, mais dependente as classes são uma das outras, dificultando assim a manutenção do mesmo. Dependendo da métrica em questão, de coesão e acoplamento, seu aumento ou diminuição pode indicar a necessidade de manutenção do código fonte em determinado momento do ciclo de vida do software.



Figura 1. Constância da métrica CBO.

### Conclusão

O estudo de métricas no ciclo de desenvolvimento de um software para medir a sua qualidade depende de como as métricas são analisadas, como apresentado nos trabalhos de *Briand et al* [2-3]. Analisamos as métricas de acordo com as suas características e com o que representam. Verificamos que o decaimento arquitetural de um software pode ser observado através da análise de algumas métricas de coesão e acoplamento de software, indicando possíveis alterações a serem realizadas. O momento para realização dessas alterações é objeto de um próximo estudo a ser desenvolvido.

[1] Virtual Machinery, 2019. *Jhawk* Product Overview. Disponível em: <http://www.virtualmachinery.com/jhawkprod.htm>. Acesso: 03, julho de 2019.

[2] BRIAND, Lionel C.; DALY, John W.; WÜST, Jürgen. A unified framework for cohesion measurement in object-oriented systems. *Empirical Software Engineering*, v. 3, n. 1, p. 65-117, 1998.

[3] BRIAND, Lionel C., DALY, John W., e WUST, Jurgen K. "A unified framework for coupling measurement in object-oriented systems." *IEEE Transactions on software Engineering* 25, no. 1 (1999): 91-121..