



CRICTE 2017

XXVIII Congresso Regional de Iniciação Científica e Tecnológica em Engenharia



ANÁLISE DA EVOLUÇÃO DA MANUTENIBILIDADE DA PLATAFORMA DE INTEGRAÇÃO MULESOFT

Rodolfo Berlezi

Aluno do curso de Ciência da Computação
Universidade Regional do Noroeste do Estado do RS (UNIJUI)
rodolfo_berlezi@hotmail.com

Rafael Z. Frantz

Professor do Programa de Pós-Graduação em Modelagem Matemática
Universidade Regional do Noroeste do Estado do RS (UNIJUI)
rzfrantz@unijui.edu.br

Resumo. *Uma empresa costuma ter diversas aplicações, localmente ou na nuvem. Estas aplicações formam o chamado ecossistema de software e são responsáveis por gerir as tarefas da empresa. A vasta diversidade de tecnologias, tipos de dados e linguagens de programação faz com que as aplicações tenham dificuldades em trocar informações, pois, geralmente, não foram pensadas para trabalhar em conjunto. A Integração de Aplicações Empresariais é um campo específico da Engenharia de Software que trata de estudar e providenciar soluções de integração para a colaboração entre aplicações através do uso de plataformas de integração. No mercado há diversas plataformas de código aberto, como o Mulesoft. Estas plataformas são de propósito geral e uma empresa pode precisar adaptá-la para um campo específico, como saúde, segurança, educação ou comércio digital. Neste artigo é feito uma revisão das versões do Mulesoft, que é uma plataforma de código aberto com suporte a padrões de integração documentados por Hohpe e Woolf, em vista a verificação do seu grau de manutenibilidade dada a evolução da plataforma nos aspectos do paradigma orientado a objeto.*

Palavras-chave: *Mulesoft. Integração de aplicações empresariais. Manutenibilidade de software.*

1. INTRODUÇÃO

Diversas aplicações são usadas em ambientes empresariais nos dias de hoje. Estas aplicações podem estar tanto localmente quanto em servidores salvos na nuvem, além de diferentes setores necessitem de diferentes aplicações para a realização e controle de suas tarefas. Esta diversidade de aplicações e softwares, tanto os feitos pela própria empresa quanto os terceirizados, criam o chamado ecossistema de software da empresa. Cada aplicação tem diferentes compatibilidades e não são pensadas para trabalhar em conjunto com outros programas do ecossistema, por isso necessitam de um meio para circular a informação incompatível entre eles. Existe uma área da engenharia de software conhecida por Integração de Aplicações Empresariais que é responsável pelo estudo e desenvolvimento de soluções de integração e existem, no mercado, diversas plataformas de integração, tal como o Mulesoft [1], desenvolvidas para este fim.

Mulesoft é uma das plataformas de integração de aplicações citadas no catálogo de integração de Hohpe e Woolf [2], com



características de código aberto, propósito geral e concebendo muitos dos padrões de integração considerados pelos autores. Foi utilizado distintas versões do Mule para uma análise comparativa da evolução do software, visto que uma empresa poderia optar por desenvolver sua própria solução de integração adaptando uma das versões existentes no mercado. Esta pesquisa tem por objetivo encontrar qual das versões do Mule tem o melhor grau de manutenibilidade em relação a orientação objeto.

Segundo a IEEE [3] a manutenção adaptativa é aquela executada em ordem para remodelar o sistema de software, adicionando novas possibilidades do mesmo ser executado em diferentes ambientes ou submeter processos que não foram inicialmente configurados para o sistema.

O paradigma da orientação a objeto, segundo Li e Henry [4], tem um ritmo de desenvolvimento de software mais acelerado e com uma melhor qualidade do que o paradigma sequencial, isso se dá pela diferença de filosofia entre os paradigmas. Através dos conceitos de herança, classes, objetos, atributos e métodos a orientação a objeto implica no comportamento dos programas.

Este artigo está organizado em mais três sessões, a metodologia da pesquisa, seus resultados, as conclusões finais e por fim as referências.

2. METODOLOGIA

Uma metodologia proposta por Frantz *et al* [5] propõem um conjunto de métricas para medição do grau de manutenibilidade de um sistema de software. Para averiguar a evolução dos graus de manutenibilidade no Mulesoft foi extraído medidas de distintas versões lançadas em ordem evolutiva da plataforma, foram elas as versões 3.1, 3.2, 3.4, 3.5, 3.7 e a última lançada até o período

da pesquisa, a 3.8. Importante mencionar que apenas o núcleo do código fonte foi considerado, visto que apenas o núcleo garante uma comparação justa entre todas as versões da plataforma.

As métricas foram classificadas em quatro grandes grupos: Medidas de tamanho, de acoplamento, de complexidade e de herança. As mesmas foram obtidas através do uso do software Metrics [6], na sua versão 1.3.6, o qual auxiliava na coleta das informações, dados e medidas do núcleo do código fonte de cada versão do Mule. Neste artigo o foco será para as medidas de tamanho e de herança que serão explicadas a seguir.

Medidas de tamanho são as que indicam quantidades para o quão grande é um sistema de software nos requisitos de pacotes, classes, interfaces, métodos, número de linhas e outros. Destacamos para esta pesquisa as seguintes métricas:

Number of Classes (NOC), ou o número de classes. A fonte essencial para o princípio das linguagens orientadas a objetos são as classes. Logo, quanto mais classes houverem, mais difícil será entender todas as suas funcionalidades.

Number of Methods (NOM), para o número de métodos nas classes e interfaces. Uma grande quantidade de métodos em uma classe indica o quão específica uma classe é.

Classes indicam o quão complicado pode ser entender como o sistema de pacotes é organizado e promovem uma visão geral do design do mesmo. Já a quantidade de métodos tanto em classes como em interfaces indica o potencial reúso de uma classe. Contudo, como fora dito por Frantz *et al* [5], uma vasta gama de métodos demonstram o quanto uma classe está para uma aplicação específica, o que limita seu potencial de reúso adaptativo.

Herança é uma característica funcional da orientação a objeto e as medidas deste



grupo representam o quão bem implementadas foram suas funcionalidades. Para a pesquisa apresentada neste artigo foram destacadas as seguintes métricas:

Depth of Inheritance Tree (DIT), ou a profundidade da árvore de herança, demonstra a quantidade de níveis de parentescos que podem ter em um sistema. Logo é um utensílio que vem a melhorar as capacidades de reúso. Contudo esta medida pode ser um indicador de o quão complexo pode vir a ser manter uma classe pelo quão maior for o valor do grau de dependência.

Number of Immediate Children Classes of a Class (NOH), está medida demonstra o potencial impacto que uma classe pode ter em um sistema se for modificada devido a suas classes filhas. O quão maior ela for, maiores são as chances de a abstração das classes país terem sido fracamente pensadas.

Number of Overriden Methods (NRM), está medida demonstra o quão uma classe está adaptável respeitando as classes país. O quão maior for este valor mais a funcionalidade de herança está sendo utilizada para adaptar os métodos da classe em vez de apenas adicionar serviços a classe pai.

Conhecer a estrutura das classes na orientação a objeto para fazer um bom reúso de suas heranças é fundamental a medida que o sistema de software evolui. Assim sendo é visível a relação entre as medidas de herança com o grau de manutenibilidade, já mencionado por Li e Henry [4]. Imaginando a possibilidade de habilitar os novos métodos nas classes mais flexíveis e adaptar os métodos nas classes mais específicas para não causar um grande aumento no grau de manutenibilidade do software.

3. RESULTADOS

Na Tabela 1, é possível observar os resultados coletados através do Metrics [6], o qual esta exibindo os valores de cada medida

coletada do Mulesoft em cada uma das suas respectivas versões.

Como mostrado na Tabela 1, as versões do Mule cresceram constantemente em tamanho a cada novo lançamento. Na versão 3.1 o Mule possuía 766 classes, na 3.2 eram 822, um aumento de quase 10% e na última versão, a 3.8, o Mule conta com 1184 classes, o software está 50% maior que na sua primeira versão. Conseqüentemente, pela relação entre classes e métodos, o que ocorreu ao número de métodos não foi diferente, tínhamos 5390 métodos e já na versão 3.5 houve um aumento de quase 45%, chegando a 7142 métodos. O que ainda cresce mais de 15% até a versão 3.8 com 8309 métodos, 60% maior que a primeira versão. O quão maior for o sistema maior será o esforço para a sua manutenção.

Tabela 1. Comparação das medidas em relação a evolução do Mulesoft

Versão Mule	3.1	3.2	3.4	3.5	3.7	3.8
NOC	766	822	977	1015	1154	1184
NOM	5390	5915	6963	7142	8140	8309
DIT	7	7	7	7	7	7
NOH Total	399	423	484	525	568	584
NOH Max	29	28	30	31	27	26
NRM	364	387	446	457	494	500

Para a primeira medida de herança, DIT, foi constatado o mesmo valor para todas as versões, 7. Isto quer dizer que ao menos em termos de profundidade da árvore de herança o Mule conseguiu se manter estável porém a um valor que não é tão baixo para a medida de herança ao olhos da manutenibilidade.

Sobre o número de classes filhas de uma classe (NOH), temos uma linha para o total de classes filhas e outra para o máximo de classes filhas de uma única classe. É



perceptível que o número total apenas aumento a cada versão, o qual começa em 399 e chega até a 584 na versão 3.8, aproximadamente 50% maior. Contudo não é o mesmo que foi observado quando visto o valor máximo, o qual começa em uma onda de valores maiores 29, 28, 30 nas versões 3.1, 3.2 e 3.4 respectivamente, e chegando a 31 na versão 3.5, o que aumentaria o grau de manutenibilidade da plataforma. Contudo para as mais recentes versões, houve melhora, onde este valor caiu para 27 na 3.7 e 26 na última versão, a 3.8, um valor melhor do que o da primeira versão, mostrando que as classes país foram mais bem pensadas para a abstração.

Com relação ao número de métodos sobrescritos (NRM), que demonstra o grau de reuso e adaptatividade do software para a manutenibilidade, houve melhora, saindo de 364 métodos da versão 3.1 para 500 métodos na versão 3.8, porém, proporcionalmente foi menor que o crescimento do número de métodos (NOM).

4. CONSIDERAÇÕES FINAIS

A integração e suas ferramentas surgem com a necessidade de soluções de integrações de aplicações novas, o que exige que as ferramentas estejam em constante evolução. Entretanto a evolução de uma ferramenta pode acabar por aumentar o seu grau de esforço, dificuldade e complexidade para uma manutenção apropriada aos padrões de softwares recomendados. O uso das métricas de tamanho e herança possibilitaram averiguar o nível da estrutura do código fonte de um software. Isto permitiu descobrir o nível da plataforma de integração Mulesoft, que está em constante evolução em suas versões e com um número elevado de suas métricas, tendo em vista o número de classes e métodos, e as relações de orientação a objeto para a manutenibilidade adaptativa do software.

Agradecimentos

Agradecemos a UNIJUÍ pela concessão da bolsa de iniciação científica que permitiu o desenvolvimento desta pesquisa.

REFERÊNCIAS

- [1] D. Dossot, J. D’Emic, and V. Romero, *Mule in action*, Manning: 2014.
- [2] G. Hohpe and B. Woolf, “Enterprise integration patterns: Designing, building, and deploying messaging solutions,” Addison-Wesley Professional, 2004.
- [3] J. Radatz, A. Geraci, and F. Katki, “IEEE standard glossary of software engineering terminology,” IEEE Std 610121990, vol. 12199, 1990, pp. 3.
- [4] W. Li and S. Henry, “Object-oriented metrics that predict maintainability,” *Journal of systems and software*, vol 23, Feb. 1993, pp. 111–122.
- [5] R. Frantz, R. Corchuelo, and F. Roos-Frantz, “A methodology to evaluate the maintainability of enterprise application integration frameworks”, *International Journal of Web Engineering and Technology*, vol 10, Apr. 2015, pp. 334–354.
- [6] Metrics, Metrics 1.3.6: 2005.