

Um Modelo de Predição de Amplitude da Propagação de Modificações Contratuais em Software Orientado por Objetos

Kecia A. M. Ferreira¹, Mariza A. S. Bigonha¹, Roberto S. Bigonha¹

¹DCC – ICEx – Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 - Pampulha - CEP: 31270-010 - Belo Horizonte

{kecia,mariza,bigonha}@dcc.ufmg.br

1. Introdução

A medição, a avaliação e o controle da manutenibilidade do software são importantes durante as fases de desenvolvimento e operação para a predição de custos e riscos de software [SEI 2009]. Muitos trabalhos têm essa questão como objeto de estudo [Deissenboek et al. 2007, Heitlager et al. 2007]. Um dos trabalhos mais importantes na medição de manutenibilidade é a métrica *Índice de Manutenibilidade* (MI) [Ash et al. 1994, Pearse and Oman 1995]. Esta métrica provê um indicador do grau de manutenibilidade a partir de coleta de medidas de complexidade do código fonte do software. Foi proposta e validada com softwares estruturados, adotada pelo SEI e utilizada em organizações como Hewlett-Packard e Departamento de Defesa Americano (DoD) [SEI 2009]. Porém, o uso dessa métrica tem sido criticado [Heitlager et al. 2007, Sarwar et al. 2008]. Uma observação que se pode fazer sobre MI é que o seu cálculo baseia-se em métricas como número de linhas código, número de linhas de comentários em código e complexidade. Essas métricas não capturam características particulares de softwares orientados por objeto, para o qual há métricas específicas [Abreu and Carapuça 1994, Chidamber and Kemerer 1994]. Heitlager et al. (2007) afirmam que, com base em suas experiências com o uso de MI em um grande número de softwares de diversos propósitos e tecnologias, observaram fortes limitações do uso desta métrica, dentre as quais destaca-se a dificuldade de se identificar o fator que causa determinado valor de MI, o que torna o seu uso difícil tanto em nível gerencial como técnico.

Apesar do grande investimento em estudos para identificar uma maneira de medir a manutenibilidade de software, ainda não há uma solução satisfatória para isso. Diante deste cenário, este trabalho de doutorado tem por objetivo a definição de um modelo de predição de amplitude da propagação de modificações contratuais em software orientado por objetos. Define-se aqui modificação contratual como aquela ocorrida dentro de uma classe e que altere o seu contrato. O contrato de uma classe corresponde aos seus serviços públicos, as pré e pós condições desses serviços. As pré-condições de um serviço são as condições que devem ser satisfeitas em relação aos valores dos parâmetros do método chamado e do estado do objeto. As pós-condições de um serviço correspondem ao estado do objeto após a execução do método e o valor a ser retornado por ele [Meyer 1997]. Se uma modificação não altera uma pré ou uma pós-condição, as classes usuárias do serviço modificado não sofrerão impacto. Contudo, se as pré ou as pós-condições sofrem alterações em decorrência da modificação realizada, as classes usuárias do serviço são afetadas, podendo também necessitar ser modificadas para adaptarem-se, o que também pode gerar necessidades de modificações em suas respectivas classes usuárias.

A ideia desse modelo é que se possa obter, a partir da estrutura do software, por exemplo por meio de seu código, um número que indique a sua manutenibilidade e, como esse número, tanto gerentes quanto técnicos possam atuar no software de maneira a tornar a sua manutenção mais fácil. O modelo proposto nesta tese, denominado K3B, visa solucionar a seguinte questão: qual é número médio de passos $E(n,i)$ de modificações para que um software com n módulos alcance estabilidade quando for necessário modificar i de seus módulos. Na orientação por objetos, classes correspondem a módulos. Define-se *passo de modificação* como o conjunto de atividades de manutenção que ocorrem desde o início da modificação em um módulo até o momento em que ele é declarado atualizado. A *estabilidade* ocorre quando, a partir de um conjunto de modificações iniciadas em i módulos, todos os módulos do sistema estão no estado atualizado. Neste modelo, o grau de interconexão entre os módulos do software, a sua *conectividade*, é considerada como o fator de principal impacto, pois é determinante na propagação das modificações. Além da conectividade, o modelo utiliza duas outras métricas de características classicamente reconhecidas como essenciais para a obtenção de boas estruturas de software: coesão interna de módulos e grau de acoplamento entre módulos.

1.1. Andamento do Trabalho

Este trabalho de doutorado teve início em Março de 2007. As seguintes atividades já foram realizadas:

1. O modelo K3B já foi elaborado.
2. A ferramenta *Connecta* tem sido atualizada para atender os requisitos necessários à utilização do modelo proposto neste trabalho e viabilizar os experimentos que deverão ser realizados para sua avaliação.
3. Foi proposta uma nova métrica para avaliação de coesão, utilizada em K3B, e sua implementação já foi realizada na ferramenta *Connecta*.
4. Foi realizado um estudo com 25 softwares abertos, totalizando mais de 25000 classes, no qual foram coletados e analisados os valores de um conjunto de métricas de software OO. Como resultado do estudo foram identificados valores referência para as métricas estudadas. O artigo referente a este estudo foi publicado e premiado como o quarto lugar na categoria Melhor Artigo no SBES 2009. Em razão disso, uma extensão desse artigo será publicada no Journal of Software and Systems, em uma edição especial trazendo os cinco melhores artigos apresentados no SBES 2009.
5. A proposta do projeto de tese foi apresentada e aprovada em Setembro de 2009.

Para concluir o trabalho, a avaliação do modelo K3B está em andamento. Os resultados deverão ser publicados em um artigo. A defesa da tese está prevista para Março de 2011.

Referências

- Abreu, F. B. and Carapuça, R. (1994). Object-oriented software engineering: Measuring and controlling the development process. In *Proceedings of 4th Int. Conf. of Software Quality*, McLean, VA, USA.
- Ash, D., Alderete, J., Oman, P. W., and Lowther, B. (1994). Using software maintainability models to track code health. 15:123 – 149.

- Chidamber, S. R. and Kemerer, C. F. (1994). A metrics suite for object oriented design. pages 476–493.
- Deissenboek, F., Wagner, S., and Pizka, M. (2007). An activity-based quality model for maintainability. In *ICSM 2007*, pages 184–193.
- Heitlager, I., Kuipers, T., and Visser, J. (2007). A practical model for measuring maintainability - a preliminary report. *IEEE*.
- Meyer, B. (1997). *Object-oriented software construction*. Prentice Hall International Series, Estados Unidos, 2 edition.
- Pearse, T. and Oman, P. (1994). Maintainability measurements on industrial source code maintenance activities. 29:295–303.
- Sarwar, M. I., WasifTanveer, Sarwar, I., and Mahmood, W. (2008). A comparative study of mi tools: Defining the roadmap to mi tools standardization. *IEEE*.
- SEI, S. E. I. (2009). *Software Tecnology Roadmap*. Disponível em <http://www.sei.cmu.edu/str/7/28/2008>. Acesso em Abril de 2009.