

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciências da Computação
Curso de Bacharelado em Ciência da Computação

**Melhorias e Implementação de Novas Características para
BugMaps: Uma Ferramenta para Visualização e Análise de Bugs**

por

Pedro Araujo Pires

Proposta de Projeto Orientado em Computação I

Apresentado como proposta de trabalho na disciplina de Projeto Orientado em Computação I do Curso de Bacharelado em Ciência da Computação da UFMG

Prof. Dr. Marco Tulio de Oliveira Valente
Orientador(a)

Assinatura do aluno:

Assinatura do orientador:

Belo Horizonte
2012 / 2º semestre

INTRODUÇÃO

Atualmente existe um bom número de ferramentas de análise de software. Uma fonte de informação que tem sido pouco explorada por estas ferramentas são os *bugs* de um sistema. Algumas ferramentas já analisam esta informação, mas ainda pouco se sabe sobre o comportamento evolucionário, tempo de vida, distribuição e estabilidade dos *bugs*.

REFERENCIAL TEÓRICO

Atualmente existe um grande número de ferramentas de análise de software [1, 2, 3]. Tais ferramentas usam diferentes tipos de informação sobre a estrutura e o histórico de um sistema. Basicamente, estas ferramentas são usadas para analisar a evolução de um software, gerenciar a qualidade do código fonte, calcular métricas, analisar regras de codificação, etc. Em geral, estas ferramentas ajudam engenheiros de software a entender grandes quantidade de dados vindas de repositórios de software.

Por outro lado, uma fonte de informação que tem sido pouco explorada pelas ferramentas de análise existentes são os *bugs* de um sistema. Algumas ferramentas já analisam esta informação, mas ainda pouco se sabe sobre o comportamento evolucionário, tempo de vida, distribuição e estabilidade dos *bugs* [4, 5]. Além disso, raciocinar sobre *bugs* é uma tarefa inerentemente complexa, porque *bugs* são intangíveis, invisíveis e difíceis de serem rastreados. Particularmente, tal análise é complexa pois envolve: (i) recuperação de dados dos sistemas de gerenciamento de *bugs* e sistemas de controle de versão; (ii) mapeamento de *bugs* para defeitos em módulos de software; (iii) processamento de dados para extrair e raciocinar sobre informações relevantes.

MÉTODO DE PESQUISA

Esta pesquisa possui as seguintes características:

- aplicada (tecnológica)
- exploratória

- experimental
- em laboratório

As seguintes atividades serão desenvolvidas:

1. Revisão da Literatura: revisão da literatura sobre modelagem e predição de *bugs* em sistemas de software.
2. Estudo da linguagem Smalltalk: o conhecimento da linguagem de programação Smalltalk é necessário, pois o módulo de visualização de *bugs* da ferramenta BugMaps é escrito em Smalltalk.
3. Automatização da extração de dados vindos de sistemas de gerenciamento de *bugs* e sistemas de controle de versão. Atualmente este processo é manual.
4. Melhoramento do processo de integração de modelos extraídos do código fonte com a série temporal de número de defeitos. Atualmente este processo está sendo feito por meio de três arquivos CSV contendo informações sobre os defeitos, a ideia é fazer a integração por meio de um único arquivo XML.
5. Melhoramento das visualizações existentes acrescentando informações sobre o teste de Granger. O teste de Granger foi objeto de estudo do trabalho Couto *et al.* [6]. Basicamente este trabalho usa o teste de Granger para encontrar relações causais entre métricas de código fonte e *bugs*. A ideia é fornecer informações sobre quais propriedades do código fonte, de acordo com o Granger, foram as causas dos *bugs*.

RESULTADOS ESPERADOS

Ao final deste trabalho espera-se que a ferramenta BugMaps possua um conjunto de análises e visualizações que ajude engenheiros de software a entender melhor os sistemas sendo analisados. A ideia é que, após as melhorias propostas, ela se torne uma ferramenta profissional e usual.

BIBLIOGRAFIA CITADA

- [1] O. Nierstrasz, S. Ducasse, and T. Girba. The story of moose: an agile reengineering environment. *SIGSOFT Softw. Eng. Notes*, 30:1–10, 2005.
- [2] SonarSource. Sonar platform. <http://sonarsource.org>.
- [3] R. Wettel. Visual exploration of large-scale evolving software. In *International Conference on Software Engineering*, pages 391–394, 2009.
- [4] M. D’Ambros and M. Lanza. Bugcrawler: Visualizing evolving software systems. In *European Conference on Software Maintenance and Reengineering (CSMR)*, pages 333–334, 2007.
- [5] M. D’Ambros and M. Lanza. Distributed and collaborative software evolution analysis with churrasco. *Science of Computer Programming*, 75(4):276–287, 2010.
- [6] C. Couto, C. Silva, M. T. Valente, R. Bigonha, and N. Anquetil. Uncovering causal relationships between software metrics and bugs. In *European Conference on Software Maintenance and Reengineering (CSMR)*, pages 223–232, 2012.