

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

Marco Túlio Valente - mtov@dcc.ufmg.br

Caracterização do Problema: Atualmente existe um grande número de ferramentas de análise de software [5, 7, 6]. 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 quantidades de dados vindas de repositórios de software.

Por outro lado, um 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* [3, 2]. 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.

Pesquisa Desenvolvida: A primeira versão da ferramenta chamada de BugMaps já se encontra implementada e publicada [4]. A primeira versão foi desenvolvida em parceria com o INRIA França, projeto "Pequi", Cooperação FAPEMIG – INRIA, Edital 22/2010, proc APQ-00214-11. Até o presente momento BugMaps possui as seguintes características:

- A ferramenta automaticamente extrai uma série temporal com o número de defeitos a partir de informações disponíveis em sistemas de gerenciamento de *bugs* (Jira e Bugzilla) e sistemas de controle de versão (CVS e SVN).
- A ferramenta integra modelos extraídos do código fonte com a série temporal número de defeitos.
- A partir da integração, a ferramenta fornece um conjunto de visualizações que ajudam desenvolvedores de software e gerentes na resposta das seguintes questões: (a) Quais são os módulos envolvidos na correção de *bugs*? (b) Qual é o tempo de vida de um *bug*. (c) Qual é o período que um módulo apresentou mais *bugs*. (d) Quais módulos são estáveis ou instáveis com relação aos *bugs*? (e) Qual é o número total de *bugs* de um módulo?

Próximos Passos: Com o objetivo de tornar a ferramenta mais profissional e consequentemente usual, pretende-se implementar as seguintes atividades:

- Automatizar a extração dos dados vindos de sistemas de gerenciamento de *bugs* e sistemas de controle de versão. Atualmente este processo é manual.
- Melhorar o processo de integração de modelos extraídos do código fonte com a série temporal número de defeitos. Atualmente este processo está sendo feito por meio de três arquivos .csvs contendo informações sobre os defeitos, a idéia é fazer a integração por meio de um único arquivo XML.

- Melhorar as visualizações existentes acrescentando informações sobre o teste de Granger. O teste de Granger foi objeto de estudo do trabalho Couto *et al.* [1]. Basicamente este trabalho usa Granger para encontrar relações causais entre métricas de código fonte e *bugs*. A idéia é instruir a ferramenta BugMaps com informações sobre quais propriedades do código fonte, de acordo com o Granger, foram as causas dos *bugs*.
- Criar uma documentação contendo os procedimentos para instalação, configuração e uso.
- Criar um site para disponibilização da ferramenta contendo a documentação do item anterior.

Equipe: A equipe deste projeto será constituída por: (a) Um aluno de doutorado (César Couto), que ficará responsável por todas as atividades descritas anteriormente; (b) Um aluno de IC, cuja bolsa está sendo solicitada no presente Edital PRPq. Especificamente, pretende-se que esse aluno ajude na implementação da ferramenta conforme detalhado em Plano de Trabalho em anexo.

Referências

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