

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

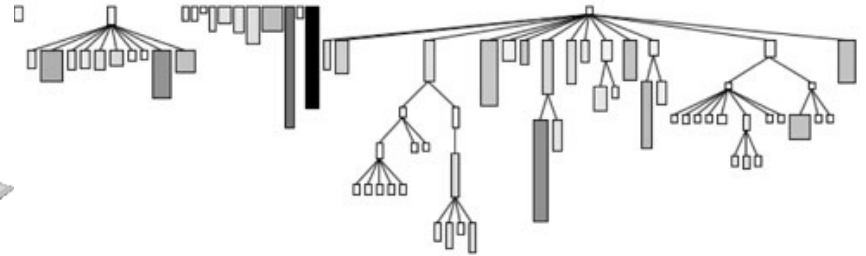
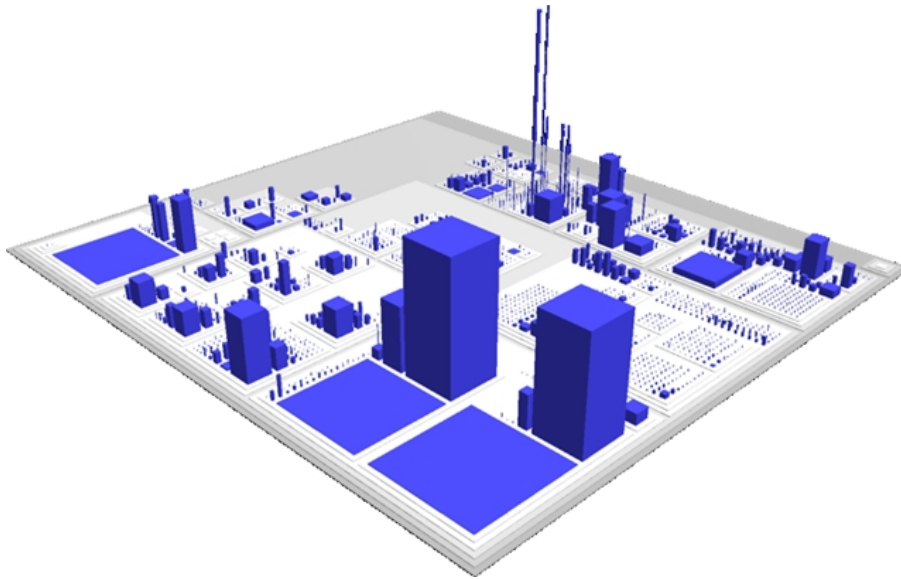
POC I - 2012/2

Pedro Araujo Pires

Prof. Marco Tulio de Oliveira Valente
Orientador

Introdução

- Atualmente existem várias ferramentas para visualização e análise de software:
 - Moose (www.moosetechnology.org)
 - Sonar (www.sonarsource.com)
 - CodeCity (codecity.inf.usi.ch)



Introdução

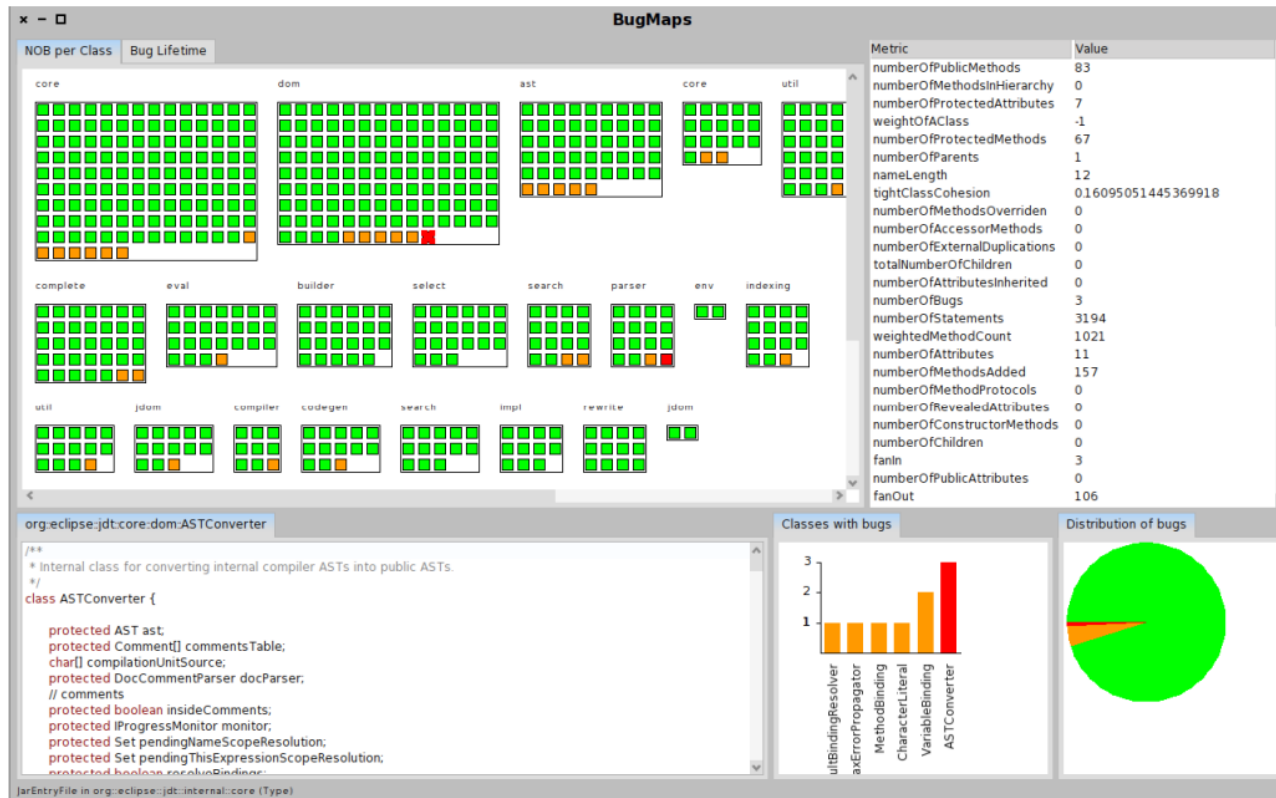
- Diversas fontes de informações sobre a estrutura e a história de sistemas são utilizadas por essas ferramentas.
- Tais ferramentas geralmente são usadas para:
 - Analisar a evolução de um software.
 - Gerenciar a qualidade de código fonte.
 - Computar métricas de software.
 - Analisar regras de codificação.
- Ferramentas de análise ajudam os engenheiros de software a compreender grandes quantidades de dados provenientes de repositórios de software.

Introdução

- Entretanto, uma fonte de informação pouco explorada por tais ferramentas são os bugs de um sistema.
- Pouco se sabe sobre o comportamento evolucionário, tempo de vida, distribuição e estabilidade de bugs.
- Bugs são intangíveis, invisíveis e difíceis de serem rastreados.

BugMaps

- Ferramenta de análise de software.
- Construída sobre a plataforma Moose.
- Linguagem: smalltalk.



BugMaps

- Extração de uma série temporal de defeitos em classes a partir de informações provenientes de:
 - Sistemas de gerenciamento de bugs (Jira, Bugzilla, etc.)
 - Sistemas de gerenciamento de código fonte (SVN, Git, etc.)
- Integração entre modelos extraídos do código fonte e a série temporal de defeitos.

BugMaps

- A partir da integração dos dados, são geradas uma série de visualizações que ajudam o desenvolvedor a responder questões como:
 - Quais módulos estiveram envolvidos na correção de um bug?
 - Qual o tempo de vida de um bug?
 - Em qual período um determinado módulo apresentou mais bugs?
 - Quais módulos são estáveis/instáveis em relação a bugs?
 - Em quais módulos o número de bugs aumentou/diminuiu ao longo do tempo?
 - Qual o número total de bugs de um módulo?

Objetivos

- Automatização da extração dos dados provenientes dos sistemas de gerenciamento de bugs e de código fonte.
- Melhora do processo de integração dos modelos extraídos do código fonte.
- Melhora das visualizações acrescentando informações sobre o teste de Granger.

Atividades Planejadas

- Revisão da Literatura sobre modelagem e previsão de bugs.
 - 100% concluído.
- Estudo da linguagem smalltalk.
 - 100% concluído.
- Adição de informações sobre o teste de Granger às visualizações.
 - 100% concluído.
- Automatização da extração dos dados.
 - 50% concluído.
- Melhora do processo de integração dos dados (3 CSVs → 1 XML).
 - 0% concluído.