

Edeyson

An Integrated Approach for Studying Architectural Evolution

Resumo

O estudo de como um sistema de software evolui com o tempo é um processo difícil, custoso e que consome tempo. Diversas técnicas têm sido desenvolvidas para auxiliar tal processo, embora cada uma apresente um conjunto particular de restrições.

O artigo apresenta uma abordagem do estudo da evolução de software que integra métricas, metáforas visuais para a visualização de software e um processo que diferencia a criação de novos artefatos de código da sua mudança de nome e/ou localização. Este conjunto deve auxiliar na análise da mudança arquitetural e estrutural do software entre suas principais *releases*.

O padrão de evolução do software é assistido por ferramentas de navegação e análise que auxiliem na visualização de que mudanças ocorreram com o tempo e sua justificativa. Três aspectos são destacados para soluções nessa linha de pesquisa: organização do volume de dados a analisar, integração de múltiplas técnicas de análise de evolução e como aplicá-las às mudanças estruturais do software.

O artigo apresenta, como solução de integração das múltiplas técnicas supracitadas, o Beagle, solução web focada em avaliação de software de código aberto (OSS). Destaca sua visualização em diagrama estrutural (hierarquias do sistema) e de dependências.

O Beagle extrai os principais fatos arquiteturais do software através de engenharia reversa e usa uma coleção de métricas combinadas para a avaliação da sua evolução. Ainda, provê o suporte à diferenciação entre novos artefatos de uma release de software de sua mudança de nomes e/ou localização (como mudança de pacotes).

A análise de mudanças estruturais – análise de origem – capta resultados de rafatorações para diferenciar novos artefatos dos já existentes através das análises Bertillonage (análise de similaridade de código via aplicação de métricas) e de dependência (análise do padrão de chamadas entre módulos).

Crítica

O artigo apresenta uma interessante plataforma integrada de armazenamento, análise e visualização de software. Todavia, sua brevidade não subsidia a escolha das métricas usadas.

As análises Bertillonage e de Dependência deviam ser mais exploradas para permitir ao leitor melhor compreensão de sua aplicação e sua relação com as métricas. Mesmo à comparação da primeira com a detecção de clones caberia uma melhor descrição de diferenças.

O entendimento da análise e da visualização ficou restrito diante da descrição da arquitetura da solução. A visualização é pouco explorada como suporte à evolução arquitetural.

Como destaque do texto, aponto a análise de similaridades entre artefatos de código que suplanta a análise de clones. Ele realmente facilita rastrear mudanças que podem passar despercebidas.

Questões para discussão

1. Como as métricas contribuem para a análise de similaridades entre artefatos de código?
2. Que outras métricas podem ser usadas nessa análise?
3. Como mensurar a precisão e efetividade da análise de similaridade das análises Bertillonage e de Dependência?

Referência

Qiang Tu, Godfrey, M.W., An Integrated Approach for Studying Architectural Evolution. Program Comprehension, 2002. Proceedings.10th International Workshop on , pp. 127–136, 2002.