

Métricas e Heurísticas para Detecção de Problemas em Aplicações Web

Maurício Aniche
aniche@ime.usp.br

Quem sou eu?

- Graduado em CC pelo Mackenzie (2007)
- Mestre em CC pelo IME/USP (2012)
Orientador: Marco Aurélio Gerosa
- Aluno de doutorado pelo IME/USP (2012-?)
Orientador: Marco Aurélio Gerosa
- Líder técnico na Caelum

Motivação

- Nos últimos anos, tenho trabalhado com desenvolvimento de aplicações Web de todos os tamanhos, em diferentes linguagens.
 - De 80 a 1000 classes
 - Java, C#, Ruby
- Mas os problemas são os mesmos.

Problemas que vejo no dia-a-dia

- Controllers acoplados.
- Controllers com regras de negócio.
- Modelos confusos/complexos.
- JSPs e Taglibs com scriptlet.
- Testes de aceitação instáveis.
- DAOs com métodos inúteis ou em lugares errados.
- E por aí vai...

Como avaliar qualidade?

- Métricas de código
 - Úteis.
 - São baratas e executam rápido.
 - Eu, em particular, acho a ideia sensacional.
- Meu problema com elas?
 - Pouco contextuais.
 - Estudos na área só querem saber de benchmarking.

Falta contexto, como assim?

- Estudos que buscam por números mágicos.
 - Mas quem disse que o número mágico (digamos, 20 linhas por método) é bom para TODAS as classes do meu sistema?
 - E as minhas decisões arquiteturais?
- Não quero o melhor código do mundo, quero o código que facilite manutenção.

Você já olhou a fundo?

- Todas as vezes que usei a mesma massa gigantesca de dados que outro artigo e fui analisar, encontrei problemas.
 - Exemplo atual: artigo do ICSME sobre testes funcionais em aplicações Web de código aberto. Quando olhei a massa, vi que colocaram dois projetos nossos, que NÃO ERAM web.
 - Outro exemplo: Meu artigo no CSMR sobre *asserts*

Minha proposta

- Proponho a criação de métricas e heurísticas para identificação de problemas em aplicações Web.
 - Todas elas serão extraídas de questionários e entrevistas com desenvolvedores do mundo real.
 - Serão analisadas em projetos de código aberto, para fins exploratórios, e depois estudos de caso em projetos da indústria.
 - Projetos de código aberto escolhidos manualmente.

Já temos algo?

- Primeiro estudo sobre métodos em DAOs ambíguos foi publicado no MTD2014, colocado com o ICSME2014.
 - Bolamos uma heurística para identificar esses métodos e fizemos um pequeno estudo em 3 projetos da Caelum.

| Project | # of DAO methods | # of right methods | # of wrong methods | % of wrong methods |
|-------------|------------------|--------------------|--------------------|--------------------|
| Caelumweb | 590 | 511 | 79 | 13.38% |
| Codesheriff | 70 | 57 | 13 | 18.57% |
| Gnarus | 233 | 200 | 33 | 14.16% |

| Project | # of inspected methods | # of agreement | % of agreement |
|-------------|------------------------|----------------|----------------|
| Caelumweb | 79 | 59 | 74.68% |
| Codesheriff | 13 | 8 | 61.53% |
| Gnarus | 33 | 16 | 48.48% |

Dúvidas

- Como validar essas heurísticas?
 - Opinião dos desenvolvedores + estudos em projetos de código aberto são suficientes.
- Vale a pena criar métricas genéricas?
 - Ao invés de definir um número mágico, usar um número calculado para o projeto em particular?

Obrigado!

aniche@ime.usp.br

www.aniche.com.br

[@mauricioaniche](#)