

MANUTENÇÃO DE SOFTWARE: COMENTÁRIOS

ENGENHARIA DE SISTEMAS DE INFORMAÇÃO

Daniel Cordeiro

27 de outubro de 2017

Escola de Artes, Ciências e Humanidades | EACH | USP

*COMENTÁRIOS DEVEM DESCREVER COISAS QUE
NÃO SÃO ÓBVIAS NO CÓDIGO: “O PORQUÊ”,
NÃO “O QUÊ”*

— JOHN OUSTERHOUT

```
// Adiciona 1 a i  
i++;
```

```
// Lock para proteger contra acesso concorrente  
SpinLock mutex;
```

```
// Esta função troca os painéis  
void swap_panels(Panel* p1, Panel* p2) {...}
```

Comentários devem ser escritos em um nível de abstração maior do que o código

```
# Percorre o vetor para ver se o símbolo existe
```

E não...

```
# Em um laço para todo índice do array, pega  
# o terceiro valor da lista do conteúdo para  
# determinar se ela tem o símbolo que estamos  
# procurando. Define o resultado como sendo  
# o símbolo, se ele for encontrado.
```

```
def workaround_rails_bug_2298!  
  # Rails Bug 2298: when a db txn fails, the id's of the instantiated objects  
  # that were not saved are NOT reset to nil, which causes problems when  
  # successfully saved later on (eg when transaction is rerun). Also,  
  # new_record is not correctly reset to true.  
  # the fix is based on a patch shown here:  
  # http://s3.amazonaws.com/activereload-lighthouse/assets/fe67deaf98bb15d58218acdbbdf7d4f166  
  # If any of the saves was on a record that had already been saved previously,  
  # we can just reload it instead; but we have to force trying this since we can't  
  # trust @new_record to tell us this fact.  
  # If reload fails, and it really was a new record, we have to apply the fix.
```

IDENTIFICANDO O QUE ESTÁ ERRADO: MÉTRICAS, CHEIROS DE CÓDIGO E SOFA

Como dar feedback sobre a beleza do código?

- Existem diretrizes para o que é bonito?
- Avaliações qualitativas?
- Avaliações quantitativas?
- Se existem, funcionam?
- O Rails tem ferramentas para apoiar isso?

A sigla **SOFA** captura sintomas que normalmente indicam esses cheiros de código:

- O código é curto (**S**hort)?
- Faz uma única tarefa (**O**ne thing)
- Tem poucos argumentos (**F**ew arguments)
- Mantém um nível consistente de **A**bstração?

A ferramenta **reek** do Rails ajuda a encontrar cheiros de código.

ÚNICO NÍVEL DE ABSTRAÇÃO

- Tarefas complexas precisam de uma estratégia dividir para conquistar
- Alerta amarelo para “encapsule essa tarefa em um método”
- Como em uma notícia de jornal, classes & métodos devem poder ser lidos de cima para baixo
 - Bom: comece com um resumo de alto nível sobre os pontos chave, depois discuta cada ponto em detalhe
 - Bom: cada parágrafo descreve um tópico
 - Ruim: vagueie sobre o código, mude os “níveis de abstração” toda hora ao invés de refiná-los progressivamente

POR QUE TER MUITOS ARGUMENTOS É RUIM?

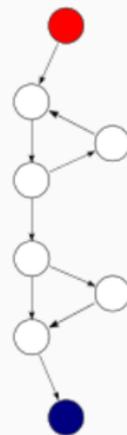
- Difícil de conseguir uma boa cobertura de testes
- Difícil de criar mocks/stubs enquanto testa
- Argumentos booleanos devem acender a luz amarela:
 - se uma função se comporta de forma diferente baseado no valor de um argumento booleano, talvez você devesse ter duas funções
- Se argumentos “andam sempre em bando”, talvez você devesse extraí-los para uma nova classe
 - mesmo conjunto de argumentos para um monte de métodos

- Conta o número de **A**tribuições, ramos (**B**anches) e **C**ondições
- Pontuação = $\sqrt{A^2 + B^2 + C^2}$
- NIST (Natl. Inst. Stds. & Tech.) recomenda ≤ 20 / método
- A ferramenta **flog**, pro Rails, verifica a complexidade ABC

QUANTITATIVO: COMPLEXIDADE CICLOMÁTICA

- Número de caminhos linearmente independentes do código = $E - N + 2P$, onde E são as arestas, N são os nós e P são os componentes conexos do digrafo

```
def meu_metodo
  while(...)
    ....
  end
  if (...)
    faca_algo
  end
end
```



- No exemplo, $E=9$, $N=8$, $P=1 \Rightarrow CC = 3$
- NIST recomenda que seja ≤ 10 / módulo
- Calculado pela ferramenta **saikuro**, para Rails

Métrica	Ferramenta	Meta de Pontuação
Razão código/testes	<code>rake stats</code>	$\leq 1 : 2$
C0 (expressão) cobertura	<code>SimpleCov</code>	$\geq 90\%$
Pontuação ABC	<code>flog</code>	< 20 por método
Complexidade ciclomática	<code>saikuro</code>	< 10 por método

- “Hotspots”: lugares onde múltiplas métricas fizeram a luz vermelha acender
 - adicione `require 'metric_fu'` ao Rakefile
 - `rake metrics:all`
- Não leve as métricas ao pé da letra:
 - assim como com cobertura, elas são melhores para identificar onde melhorias são necessárias do que para garantir algo

Pontuação do flog: 18,8

```
def combine_anagrams(words)
  output_array = Array.new(0)
  words.each do |w1|
    temp_array = []
    words.each do |w2|
      if (w2.downcase.split(//).sort
          == w1.downcase.split(//).sort)
        temp_array.push(w2)
      end
    end
    output_array.push(temp_array)
  end
  return output_array.uniq
end
```

Pontuação do flog: 5.2

```
def combine_anagrams(words)
  words.group_by { |word|
    word.chars.sort }.values
end
```

Nota: flog significa punir severamente com uma vara ou chicote.
Descrição do seu site: *"Flog shows you the most torturous code you wrote. The more painful the code, the higher the score"*.