

# Centro de Informática - Sistemas de Informação

## Engenharia de Software :: Exercício Escolar Final

Prof. Vinicius Cardoso Garcia - 12 de Julho de 2012

**Observações:** Este Exercício Escolar tem 1 questão com 5 itens para um total de 10.0 (dez) pontos. A resolução do mesmo é individual. Tempo para resolução: 2hs.

O *Royal Bank of Scotland* (RBS) quase faliu em 2009, resultado da crise global. Só não foi para o grande cemitério dos CNPJ porque o governo inglês passou a ser dono de 84% das ações. Há três anos, pois, é propriedade dos contribuintes.

Ao todo, o RBS tem **17 milhões de contas** (<http://on.wsj.com/Nskj38>) em suas mais de 2300 agências. e um problema de todo tamanho, talvez o primeiro na história do software no sistema financeiro mundial: *uma atualização de um dos seus sistemas deixou 12 milhões de usuários sem poder retirar dinheiro, sem saber o saldo das suas contas, sem poder fazer transferências ou pagar contas e sem usar os caixas eletrônicos.*

E não foi por alguns minutos. Ou horas. Até aqui [porque o problema ainda não está de todo resolvido] foram **SETE DIAS** (*texto publicado no dia 25/06/2012*). A ponto de, pela primeira vez na história bancária da ilha, as agências do grupo abriram no domingo para tentar resolver problemas dos clientes, pelo menos os que podiam ser resolvidos sem o software funcionando.

Difícilmente saberemos ao certo o que causou o problema do RBS. Sabe-se apenas que foi a “evolução” de uma plataforma de software, que não deve ter sido testada de forma exaustiva o suficiente para indicar problemas que aconteceriam com sua entrada em funcionamento. Há gente apontando uma certa “*queda nos padrões*” (<http://bit.ly/Nsl7oO>) de escrita de software como causa, e vamos lembrar que a NASDAQ citou uma “*falha de projeto*” (<http://bit.ly/NslgiH>) de software como a principal causa do pálido desempenho das ações de Facebook logo depois do IPO em maio passado. Ou seja, problemas de [ou com] software passaram a ser determinantes da performance [de pelo menos parte] da economia, em qualquer país.

Não parece que a forma de tratar tais eventos – que não são privilégios do sistema financeiro - é aceitar que eles vão acontecer e que isso “*faz parte*” do processo. Em “*Large-scale complex IT systems*”, Somerville e outros propõem uma outra visão “*sistêmica*”, a de tratar os sistemas de informação como os dos bancos e outras grandes instituições, que podem ter de milhões a dezenas de milhões de linhas de código por trás, como “*coalisões*” (<http://bit.ly/NsnDLA>), fugindo do reducionismo clássico da engenharia de software.

Parece ser claro que nenhum dos problemas que envolve software de grande porte para sua solução pode ser atacado por uma visão reducionista do mundo (onde o ambiente é racional e todas as decisões são técnicas; o problema é bem definido e os limites do sistema podem ser estabelecidos; que define e “*faz*” o sistema detém o controle de todas as partes e dependências). Não é este o caso.

Neste contexto, apresente e discuta 5 (cinco) [2,0 pontos cada] pontos de problema (oportunidades para pesquisa, desenvolvimento e inovação) ao redor da Engenharia de Software nos dias atuais.

### Referências

1. Silvio Romero de Lemos Meira, e o dinheiro... sumiu | dia a dia, bit a bit. URL: <http://bit.ly/NsmpQH>
2. Silvio Romero de Lemos Meira, Sistemas de Informação: Coalisões e Máquinas Sociais, URL: <http://bit.ly/NsmWC6>.
3. Ian Sommerville, Dave Cliff, Radu Calinescu, Justin Keen, Tim Kelly, Marta Kwiatkowska, John Mcdermid, and Richard Paige. 2012. Large-scale complex IT systems. *Commun. ACM* 55, 7 (July 2012), 71-77.

Boa Sorte!