

Trabalho 2 - Engenharia de Software Experimental

A empresa UCSoft oferece ampla expertise em testes de software. Eles possuem profissionais qualificados e experientes disponíveis para atuar em diversos projetos de software. Além disso, eles dispõem de boas metodologias e práticas gerenciais que permite registrar inconsistências e mensurar a qualidade da entrega de um software. No mercado soteropolitano, a empresa SoftMaX atua auxiliando diversas outras empresas que desenvolvem aplicativos mobile na plataforma Android. A SoftMAX reutiliza diversos trechos de código, aproveitando partes importantes e comuns que diversos sistemas utilizam. Infelizmente a SoftMax está enfrentando problemas de qualidade em seus códigos impactando diretamente na qualidade de software dos seus clientes. Para melhorar esta situação, a SoftMax contratou a UCSoft para realizar consultorias, experimentos e testes. A UCSoft decidiu fazer um experimento: durante o período de 1 mês (Abril), a UCSoft iria analisar os códigos desenvolvidos pela SoftMAX sem interferir em nenhum processo de desenvolvimento, averiguando a quantidade de Bad Smells ali presente. Os Bad Smells que eles buscaram encontrar são:

Métrica	Descrição
Swiss Army Knife (SAK)	A <i>Swiss army knife</i> é uma classe com inúmeras assinaturas de interface, resultando em uma interface de classe muito complexa
Long Method (LM)	<i>Long methods</i> são implementados com muito mais linhas de código comparado a outros métodos. Eles são frequentemente muito complexos e, portanto, difíceis de entender e manter.
Member Ignoring Method (MIM)	No Android, quando um método não acessa um atributo de um objeto, é recomendável usar um método estático, pois ele é cerca de 15% a 20% mais rápido do que uma chamada dinâmica.
No Low Memory Resolver (NLMR)	Quando o sistema Android está com pouca memória, o sistema chama o método <code>onLowMemory()</code> das atividades em execução, que deve reduzir o uso de memória. Se esse método não for implementado, o sistema Android eliminará o processo para liberar memória e poderá causar uma finalização anormal dos programas.
Blob Class (BLOB)	Blob class, também conhecido como <i>God class</i> , é uma classe com uma grande quantidade de código, atributos e operações.
Internal Getter/Setter (IGS)	No Android, os campos devem ser acessados diretamente dentro de uma classe para aumentar o desempenho. O uso de um getter interno ou de um setter é convertido em uma invocação virtual, o que torna a operação três vezes mais lenta que um acesso direto.
Leaking Inner Class (LIC)	Em Java, as classes internas e anônimas não estáticas mantêm uma referência à classe externa, enquanto as classes internas estáticas não são. Isso poderia provocar um vazamento de memória nos sistemas Android

Complex Class (CC)	<i>Complex class</i> é uma classe que contém métodos complexos. Essas classes são difíceis de entender e manter e precisam ser refatoradas.
--------------------	---

Os resultados da coleta de dados neste período inicial (sem interferências) estão presentes no arquivo **bad_smells_abril.csv**. Após este período a UCSoft encontrou pontos fracos nos processos de desenvolvimento da SoftMAX. Desta forma, a segunda parte do experimento consistiu em treinar toda a equipe de desenvolvimento a seguir a metodologia **SCRUM**. A cada dia de uma Sprint, a equipe faz uma breve reunião (normalmente de manhã), chamada Daily Scrum, onde UCSoft aponta os principais problemas e Bad Smells que eles encontraram no primeiro mês do experimento. Com isto, uma nova etapa de coleta de dados se iniciou, onde novamente foram armazenados os Bad Smells encontrados durante o período de um mês (Maio). Os resultados e a nova contagem desses Bad Smells estão presentes no arquivo **bad_smells_maio.csv**. Com base nessas informações, e utilizando a linguagem de programação R para analisar os dados, resolva e implemente as seguintes questões:

1. Leia dentro da linguagem R o arquivo correspondente ao mês de **abril** faça a implementação dos seguintes itens:
 - a. Faça uma contagem geral de todos os Bad Smells e imprima na tela.
 - b. Faça a média total de todos os Bad Smells de todos os softwares.
 - c. Encontre o software que possui a maior quantidade de Bad Smells.
 - d. Faça um gráfico (pode ser em barras, pontos ou linhas) da contagem total de cada classe de Bad Smell. (exemplo: No total de todos os softwares houveram 100 Bad Smells do tipo SAK, então você pode fazer uma barra representando esse valor).
 - e. Calcule o valor de Variância e o Desvio Padrão de todos os 8 tipos de Bad Smells.
 - f. Faça um gráfico Boxplot para cada classe de Bad Smell presente no csv.
2. Leia dentro da linguagem R o arquivo correspondente ao mês de **maio** faça a implementação dos seguintes itens:
 - a. Faça uma contagem geral de todos os Bad Smells e imprima na tela.
 - b. Faça a média total de todos os Bad Smells de todos os softwares.
 - c. Encontre o software que possui a maior quantidade de Bad Smells.
 - d. Faça um gráfico (pode em barras, pontos ou linhas) da contagem total de cada classe de Bad Smell (exemplo: No total de todos os softwares houveram 100 Bad Smells do tipo SAK, então você pode fazer uma barra representando esse valor).
 - e. Calcule o valor de Variância e o Desvio Padrão de todos os 8 tipos de Bad Smells.
 - f. Faça um gráfico Boxplot para cada classe de Bad Smell presente no csv.
3. **Compare os dois resultados:**
 - a. Faça gráficos no R comparando os dois resultados dos dois meses.
 - b. Imprima a porcentagem de aumento ou diminuição de Bad Smells entre o mês de abril e maio.
 - c. A última análise você irá escolher qual será! Esta parte é livre e cabe a você pesquisar e criar algo. Tente tirar alguma boa informação entre esses dois meses! Tente fazer alguma relação entre os dados! Tente trazer alguma informação interessante ao experimento!