

Experimento

Estimado participante:

Este tipo de estudo pode lhe trazer ganhos significativos em suas atividades de desenvolvimento de software uma vez que a partir da avaliação de ferramentas podemos contribuir para o seu aperfeiçoamento e, com isso, disponibilizar para você ferramentas com mais qualidade.

Você está sendo convidado a participar deste experimento em virtude de sua experiência em atividades de desenvolvimento. Se você decidir em participar, você será solicitado a utilizar ferramentas que você pode ou não ter usado anteriormente. Durante as atividades, vou observar como você utiliza essa ferramenta e, durante seu uso, vou coletar informações à medida que avançamos na execução das tarefas para que eu compreenda o que (e o porquê) você está fazendo determinadas ações.

Deixo aqui explícito que qualquer informação obtida neste estudo que possa ter qualquer relação com sua pessoa (ou seja, que possa ajudar a identificar você) será mantida em sigilo. Dados pessoais (nome do participante, dados da empresa entre outros) serão mantidos em sigilo uma vez que não armazenaremos qualquer informação desta natureza.

A sua participação é totalmente voluntária. Sua decisão de participar ou não, não vai afetar o seu relacionamento com este pesquisador ou mesmo com a instituição. Se você decidir participar do estudo, você pode optar por retirar-se a qualquer momento, sem quaisquer penalidades. Por favor, mantenha uma cópia desta carta para fins de registro das informações aqui contidas.

Se você tiver dúvidas, problemas ou sugestões sobre a sua participação neste estudo ou os seus direitos como participante da pesquisa, por favor, entre em contato com este pesquisador.

Sinceros Agradecimentos

Configuração do experimento

Data: _____

Número do Participante: _____

Checklist do Material

- Prancheta
- Fichas de avaliação
- Questionários
- Smell cards* no envelope
- Canetas
- Mouse
- Laptop com carregador
- relógio

Configuração

- Eclipse versão 3.3
- Instalação do *Plugin do Stench Blossom*
- Importar o projeto
- Deixar classes demo previamente abertas
- Mostrar linhas do código no eclipse
- Deixar em tela cheia a aba de código
- Sentado ao lado do participante
- Cartões 3X8 em mãos
- Prancheta com questionários e fichas

Finalização

- Realizar a pseudo-entrevista
- Agradecer o participante e liberá-lo
- Verificar data e horário nas fichas
- Fechar todos os códigos (excetos os de demonstração)
- Anotar observações

Questionário - Pré-Experimento

O questionário a seguir tem por objetivo fornecer informações do seu *background* como desenvolvedor. Suas respostas não irão afetar as demais atividades deste experimento, elas simplesmente irão fornecer um contexto para interpretação dos resultados.

Sinta-se livre para escrever nas margens de forma a explicar suas respostas, se necessário.

- 1) Você trabalha como: _____
- 2) Há quanto tempo você atua como desenvolvedor? _____
- 3) Durante os últimos doze meses, cerca de quantas horas por semana você diria que você gasta em desenvolvimento (em média)? _____
- 4) Quão proficiente, em uma escala de 0 a 4 (onde 0 significa “não proficiente” e 4 “muito proficiente”) você se considera em relação às seguintes linguagens?
 - a) Java 0 1 2 3 4
 - b) C++ 0 1 2 3 4
 - c) C# 0 1 2 3 4
 - d) PHP 0 1 2 3 4
 - e) _____ 0 1 2 3 4
- 5) Quão frequente, em uma escala de 0 a 4 (onde 0 significa “nunca uso” e 4 “uso muito frequentemente”) você utiliza as seguintes linguagens?
 - a) Java 0 1 2 3 4
 - b) C++ 0 1 2 3 4
 - c) C# 0 1 2 3 4
 - d) PHP 0 1 2 3 4
 - e) _____ 0 1 2 3 4
- 6) Quando está programando, você costuma utilizar alguma IDE (S/N)? _____
Caso positivo, você utiliza qual? E por quanto tempo?

Caso negativo, porque você não faz uso?

7) Caso você utilize algum tipo de IDE, você costuma utilizar algumas ferramentas que considere útil para o desenvolvimento (mecanismos de busca, depuradores, visualização da estrutura do código, métricas, etc.) (S/N)? _____

Caso positivo, você utiliza quais? E por quanto tempo?

Caso negativo, porque você não faz uso?

8) Em uma escala de 0 to 4, quão familiarizado você está com a prática de refatoração? (0 = não familiarizado e 4 = muito familiarizado)

a) Refatoração 0 1 2 3 4

Caso positivo, realiza manualmente ou qual(is) ferramenta(s) você utiliza? _____

9) Em uma escala de 0 a 4, quão familiarizado você está com análise/detecção de *code smells*? (0 = não familiarizado, 4 = muito familiarizado)

a) *Code Smells* 0 1 2 3 4

Caso positivo, realiza manualmente ou qual(is) ferramenta(s) você utiliza? _____

Pegar antes do experimento

Procedimento Experimental

Introdução

O que vamos fazer neste experimento é investigar *code smells* que foram propostos no livro do Martin Fowler (*Refactoring: Improving the Design of Existing Code*). A ideia definida por este autor é que os *code smells* podem auxiliar na identificação de trechos de código candidatos a refatoração. Por exemplo, o *smell* "Long Method" (Método Longo) sugere que você talvez deva realizar a refatoração "Extract Method" (Extração de método).

Você não precisa estar muito familiarizado com estes conceitos. Vamos fazer alguns comentários à medida que avançamos neste experimento. Sinta-se a vontade para fazer perguntas.

Este experimento será composto por 4 partes:

- I) Na primeira parte (I), iremos realizar um pequeno treinamento sobre a conceituação de *code smells* e as formas de incidência código.
- II) Na segunda parte (IIA e IIB), iremos fornecer a você uma ferramenta para auxiliar a tarefa de análise / detecção de *code smells*.
- III) Na terceira parte (IIIA e IIIB), iremos utilizar as informações obtidas através da mesma ferramenta (do passo anterior) visando a realização de julgamentos para as ações de refatoração.
- IV) Na quarta parte (IVA e IVB), iremos responder um questionário pós-experimento e colher os comentários gerais deste experimento.

[I] Treinamento

Neste momento, você está vendo os cartões 3X8. Cada um contém o nome de um *code smell* e sua definição. Em seu verso, colocamos um pequeno trecho de código exemplificando a ocorrência deste *code smell*. Será dado o tempo de 5 minutos para você ler os cartões.

Caso você tenha o correto entendimento dos conceitos e exemplo e queira encerrar esta atividade antes do tempo, basta avisar!

Após esta atividade então, você vai devolvê-los e, quando você estiver pronto, favor avisar!

Passado o tempo inicialmente estimado, vamos começar a olhar para alguns trechos de código. Alguma dúvida?

Neste momento vou entregar alguns cartões para respostas e aguardar a sua conclusão. Se demorar mais do que o tempo estabelecido (irei informar antes da atividade) irei perguntar se os cartões foram devidamente preenchidos.

Caso não esteja devidamente concluído o preenchimento, avise (ponha uma anotação “Não Concluído”).

Agora vamos olhar para trechos de código. Enquanto você estiver verificando estes trechos de código, por favor, não os modifique, ou mesmo navegue fora do editor. Como regra geral tente não gastar muito tempo (mais do que 3-5 minutos) por arquivo.

[IIA] Busca Manual

Levar os cartões (Se for o caso). Agora vou pedir a você para olhar para alguns arquivos de código em Java e tentar detectar algum dos *code smells* listados nos cartões. Você vai percorrer este arquivo de código (através da barra de rolagem). Se você identificar algum *code smell*, é só preencher o formulário com a linha de código e o *code smell* identificado.

Exemplificar. Assim, por exemplo, você vai percorrer esse arquivo de cima para baixo observando a ocorrência de qualquer *code smell*.

Dúvidas?

| Rolagem 1 | | Rolagem 2 | |
|-------------------|-------|-------------------|-------|
| Code Smell | Linha | Code Smell | Linha |
| Large Method | | Large Method | |
| Large class | | Large class | |
| Data Clumps | | Data Clumps | |
| Message Chains | | Message Chains | |
| Switch Statements | | Switch Statements | |
| Typecast | | Typecast | |
| InstanceOf | | InstanceOf | |
| Feature Envy | | Feature Envy | |
| Rolagem 3 | | Rolagem 4 | |
| Code Smell | Linha | Code Smell | Linha |
| Large Method | | Large Method | |
| Large Class | | Large Class | |
| Data Clumps | | Data Clumps | |
| Message Chains | | Message Chains | |
| Switch Statements | | Switch Statements | |
| Typecast | | Typecast | |
| InstanceOf | | InstanceOf | |
| Feature Envy | | Feature Envy | |

[IIB] Busca com auxílio da ferramenta

Levar os cartões (Se for o caso). Vou pedir-lhe para identificar alguns dos *code smells* que você viu nas cartas. Você vai percorrer um arquivo de código Java (através da barra de rolagem) com o auxílio de uma ferramenta do tipo *Smell Detector*.

Treinamento da ferramenta. A ferramenta está representada por uma linha de visualização atrás seu código Java (Contexto atual e *ambient view*). Ele é semelhante a pétalas de uma flor (o próprio nome da ferramenta sugere – *Stench Blossom*).

Cada pétala representa um *code smell*, e podemos passar o mouse sobre para ver o nome do mesmo (*Active view*). O tamanho da pétala representa o “quão forte pode ser” a ocorrência desse *code smell* no código que você está olhando. Podemos clicar na pétala e obter mais informações a respeito do *code smell* (*Explanation view*).

Essa parte da ferramenta destina-se a dar-lhe uma ideia de que *code smells* estão presentes. Há mais detalhes que podemos citar em relação a ferramenta, mas nós vamos comentar mais a seu respeito mais tarde.

Realização da tarefa. Assim, o objetivo da nossa tarefa é verificar se a ferramenta ajuda a identificar algum *code smell* (e que você acredita que, de fato, esta ocorrência exista de verdade – lembre-se a ferramenta pode ter problemas em sua acurácia (explicar o conceito da palavra), basta preencher o *form* com linha do código e o padrão encontrado.

Pronto?

| Rolagem 1 | | Rolagem 2 | |
|--------------------------|-------|--------------------------|-------|
| Code Smell | Linha | Code Smell | Linha |
| <i>Large Method</i> | | <i>Large Method</i> | |
| <i>Large class</i> | | <i>Large class</i> | |
| <i>Data Clumps</i> | | <i>Data Clumps</i> | |
| <i>Message Chains</i> | | <i>Message Chains</i> | |
| <i>Switch Statements</i> | | <i>Switch Statements</i> | |
| <i>Typecast</i> | | <i>Typecast</i> | |
| <i>Instanceof</i> | | <i>Instanceof</i> | |
| <i>Feature Envy</i> | | <i>Feature Envy</i> | |
| Rolagem 3 | | Rolagem 4 | |
| Code Smell | Linha | Code Smell | Linha |
| <i>Large Method</i> | | <i>Large Method</i> | |
| <i>Large Class</i> | | <i>Large Class</i> | |
| <i>Data Clumps</i> | | <i>Data Clumps</i> | |
| <i>Message Chains</i> | | <i>Message Chains</i> | |
| <i>Switch Statements</i> | | <i>Switch Statements</i> | |
| <i>Typecast</i> | | <i>Typecast</i> | |
| <i>Instanceof</i> | | <i>Instanceof</i> | |
| <i>Feature Envy</i> | | <i>Feature Envy</i> | |

[IIIA] Refatoração - Manual

Agora, o que nós vamos fazer é olhar para um *code smell* com características particulares (e em maior profundidade): *Feature Envy*. Suponha analisar a ocorrência desse *code smell* através da inspeção visual.

Analisando o contexto, eu poderia concluir, por exemplo, que o método (NOME DO METODO), ou algumas partes dele, deve ser mudado da classe (NOME DA CLASSE) para outra classe (NOME DA OUTRA CLASSE).

Assim, a tarefa que eu quero que você faça é realizar alguns julgamentos sobre a necessidade de realizar refatoração no código: Quão disperso está o *Feature Envy*, qual a probabilidade de removê-lo, e como você pode fazer esta remoção. Eu vou fazer essas perguntas enquanto você trabalha, e se você tem alguma pergunta, fique à vontade para fazer. Quando terminar, avise!

Alguma pergunta?

Executar estes passos em outro método.

| Feature Envy [1] | |
|------------------------------------|--|
| Quão disperso está no código? | |
| Qual a probabilidade de removê-lo? | |
| Qual a melhor forma de removê-lo? | |
| Feature Envy [2] | |
| Quão disperso está no código? | |
| Qual a probabilidade de removê-lo? | |
| Qual a melhor forma de removê-lo? | |

[IIIB] Refatoração – Uso da ferramenta

Agora, o que nós vamos fazer é olhar para um *code smell* com características particulares (e em maior profundidade): *Feature Envy*. (Abra a ferramenta) e verifique a *active view* e *explanation view* para obter maiores detalhes sobre a ocorrência deste *code smell*.

Aberta a *wizard* da *explanation view* será demonstrado a gradação em cores das classes afetadas pelo *Feature Envy*. Assim, podemos ver que muitos membros da classe (NOME DA CLASSE) são referenciados, mas apenas um membro desta classe (NOME DA CLASSE) é referenciado. Os membros associados são destacados no código fonte.

Olhando para estes detalhes, podemos concluir que o método (NOME DO METODO), ou algumas partes dele, deve ser mudado para a classe (NOME DA CLASSE).

Assim, a tarefa que eu quero que você faça é realizar alguns julgamentos sobre a necessidade de refatoração no código; Quão disperso está o *Feature Envy*, qual a probabilidade de removê-lo, e como você pode fazer esta remoção. Eu vou fazer essas perguntas enquanto você trabalha, e se você tem alguma pergunta, fique à vontade para fazer. Quando terminar, avise.

Alguma pergunta? Executar estes passos em outro método.

| Feature Envy [1] | |
|------------------------------------|--|
| Quão disperso está no código? | |
| Qual a probabilidade de removê-lo? | |
| Qual a melhor forma de removê-lo? | |
| Feature Envy [2] | |
| Quão disperso está no código? | |
| Qual a probabilidade de removê-lo? | |
| Qual a melhor forma de removê-lo? | |