

# A Utilidade dos Valores Referência de Métricas na Avaliação da Qualidade de Softwares Orientados por Objeto

---

Priscila Pereira de Souza

Orientadora: Prof<sup>a</sup>. Mariza A. S. Bigonha (UFMG)

Coorientadora: Prof<sup>a</sup>. Kecia A. M. Ferreira (CEFET-MG)

31 de Outubro de 2016

# Roteiro da Apresentação

---

- Contextualização
- Objetivo
- Metodologia
- Mapeamento da Literatura
- Estratégias de Detecção
- Valores Referência X *Bad Smells*
- Valores Referência X Predição de Falhas
- Conclusão





# Contextualização

---

# Qualidade de Software

---

- Preocupação da Engenharia de Software
  - Eficiência, manutenibilidade e baixo custo
  
- Teste e Inspeção
  - Apoiam a qualidade de software
  - Demandam alto custo e esforços
  
- Métricas de software
  - Podem apoiar na garantia de qualidade

# Métricas de Software

---

- ❑ Valor numérico relacionado a alguma dimensão de software
  - Exemplo: tamanho do sistema
- ❑ Aplicação de métricas
  - Compreensão de código-fonte
  - Avaliar qualidade de um produto ou processo
- ❑ Valores referência são essenciais!

# Valores Referência (VR)

---

- ❑ Valores utilizados para caracterização de métricas de software
  - Exemplos: bom, ruim, alto, baixo, etc.
  
- ❑ Valores referência
  - Apoiam o uso de métricas no controle da qualidade interna de software
  - Para a maioria das métricas, são desconhecidos

# Valores Referência (VR)

---

- ❑ Alguns catálogos de valores referência disponíveis
  - Poucos deles validados
  - Filó (2014) apresenta a maior quantidade de VR de métricas propostos
- ❑ Podem ser aplicados na detecção de *bad smells* e predição de falhas

# *Bad Smells*

---

- Estruturas de código anômalas que sugerem a presença de problemas
  - Exemplo: métodos extensos com diversos desvios condicionais (*Long Method*)
  
- Detecção de *Bad Smells*
  - Inspeção manual
  - Automatizada: estratégias de detecção
    - Regras formais para categorização de elementos de código



# Predição de Falhas

---

- Objetivo
  - Auxilia na identificação dos componentes mais propensos a falhas
  
- Falha de software
  - Diferença indesejável entre o observado e o esperado, sob o ponto de vista do usuário final



# Objetivo

---

# Objetivo do Estudo

---

- Verificar a utilidade dos valores referência
  - Na avaliação da qualidade de software orientado por objetos
  
- Neste estudo, investiga-se
  - Detecção de *bad smells*
  - Predição de falhas

# Questões de Pesquisa ...

---

**QP1.** Valores referência de métricas auxiliam a identificar *bad smells*?

**QP1.1.** Qual é a eficácia da detecção de *bad smells* utilizando-se estratégias baseadas nos valores referência e tomando-se como base os resultados gerados por ferramentas de detecção de *bad smells*?

**QP1.2.** Os valores referência apoiam efetivamente a detecção de *bad smells* em relação a listas de referência geradas por um especialista com conhecimentos em orientação por objetos e *bad smells*?

# ... Questões de Pesquisa

---

**QP2.** Os valores referência de métricas de software orientados por objetos auxiliam a prever falhas em um software?

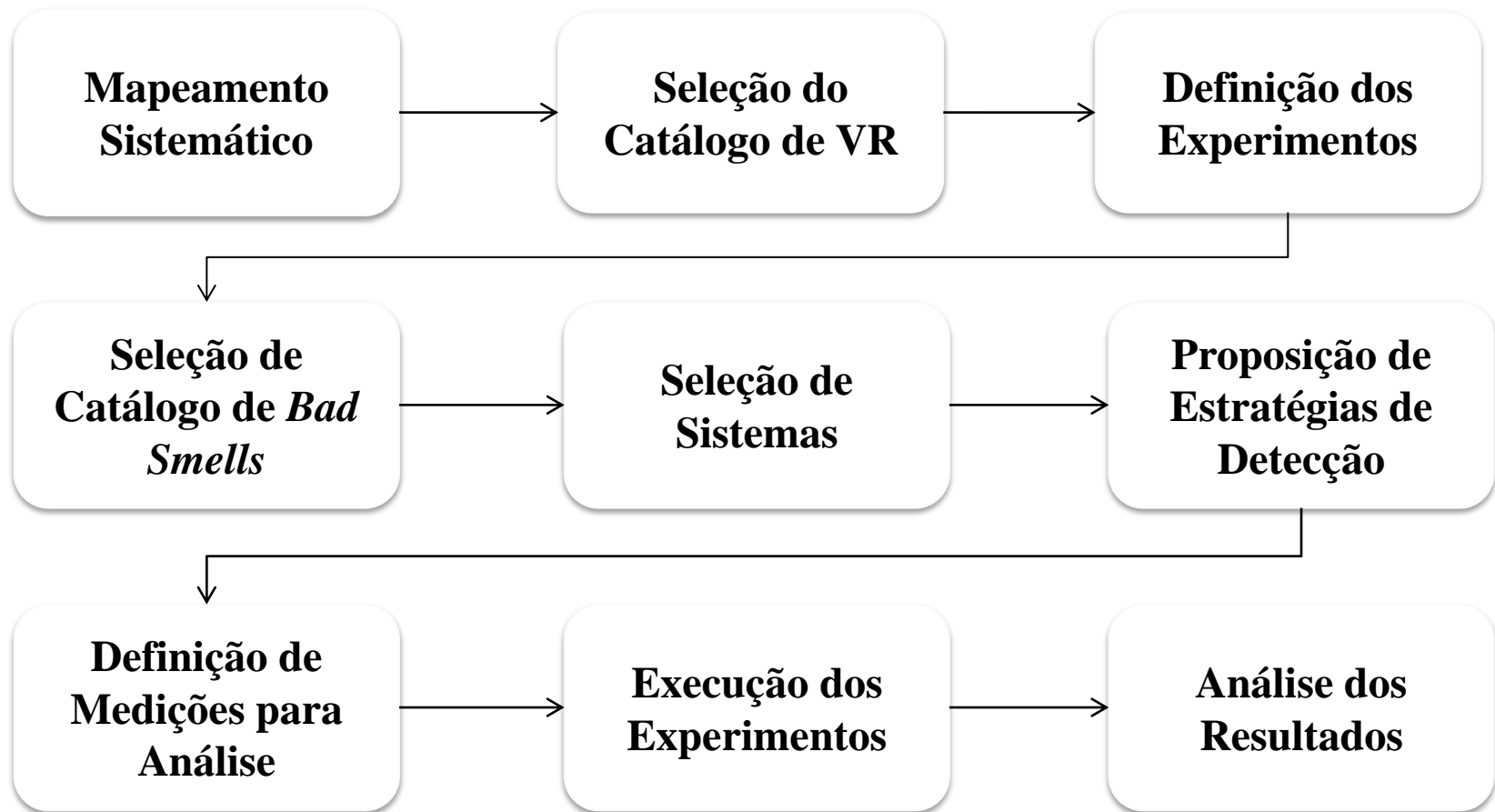


# Metodologia

---

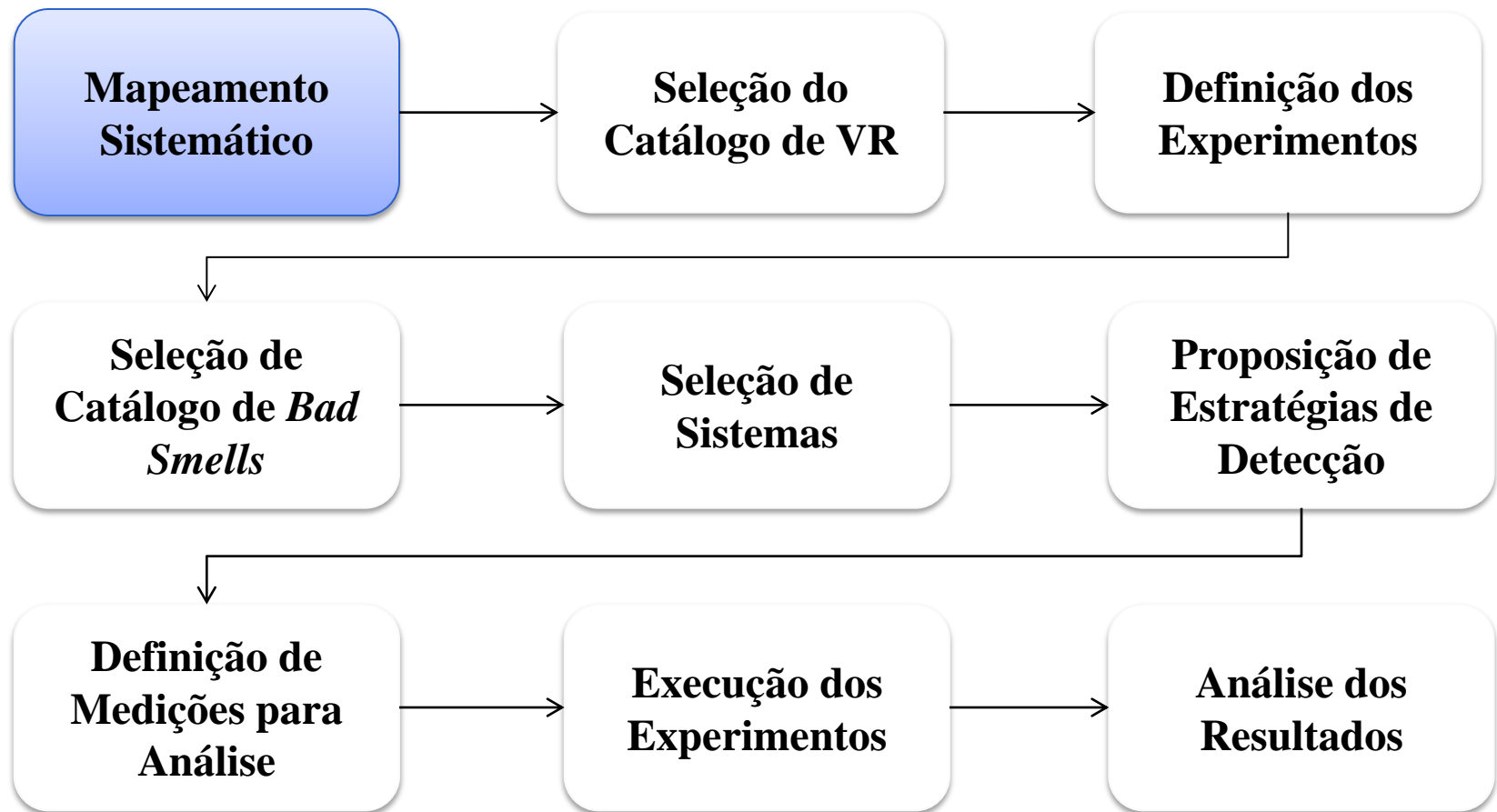
# Etapas do Estudo

---



# Etapas do Estudo

---







# Mapeamento Sistemático da Literatura

---

# Mapeamento Sistemático

---

- Por que mapeamento?
  - Mais abrangente que revisão de literatura
  
- Objetiva-se coletar evidências da relação entre valores referência e
  - Detecção de *bad smells*
  - Predição de falhas
  
- Fornecer um *background* e apoiar o trabalho

# Questões do Mapeamento

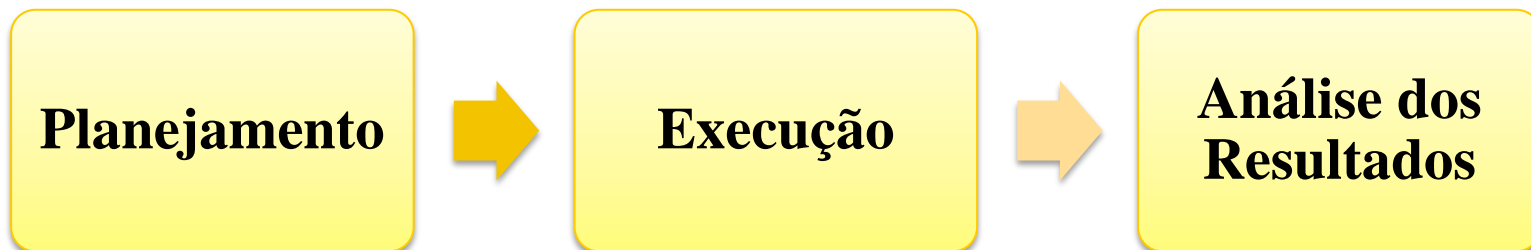
---

**QP1.** Como a literatura tem aplicado valores referência de métricas de softwares orientados por objeto para a detecção de *bad smells*?

**QP2.** Como a literatura tem aplicado valores referência de métricas de softwares orientados por objeto para a predição de falhas em software?

# Processo do Mapeamento

---



- Questões de investigação
  - *Strings* de busca
  - Bases de dados
  - Critérios de inclusão e exclusão
- Execução das *strings*
  - Seleção dos estudos
  - Coleta dos dados
- Análise dos dados obtidos
  - Publicar os resultados

## Resultados (2) - VR x *Bad Smell*

---

- Sahin *et al.* (2014)
  - Propõe detecção de *bad smells* baseada em métricas e otimização
  - Discute a dificuldade de definir valores referência
  
- Singh and Kahlon (2014)
  - Mostra a aplicação de VR na detecção de *bad smells* definindo VR com base em análise de risco

# Resultados (8) - VR x Predição de Falhas

---

- Algumas métricas avaliadas
  - Tamanho, complexidade, quantidade de mudanças
  
- Técnicas aplicadas
  - Análises estatísticas, aprendizado de máquina, análise histórica de falhas, etc.
  
- Tipos de falhas detectados
  - *Bugs*, vulnerabilidades, etc.

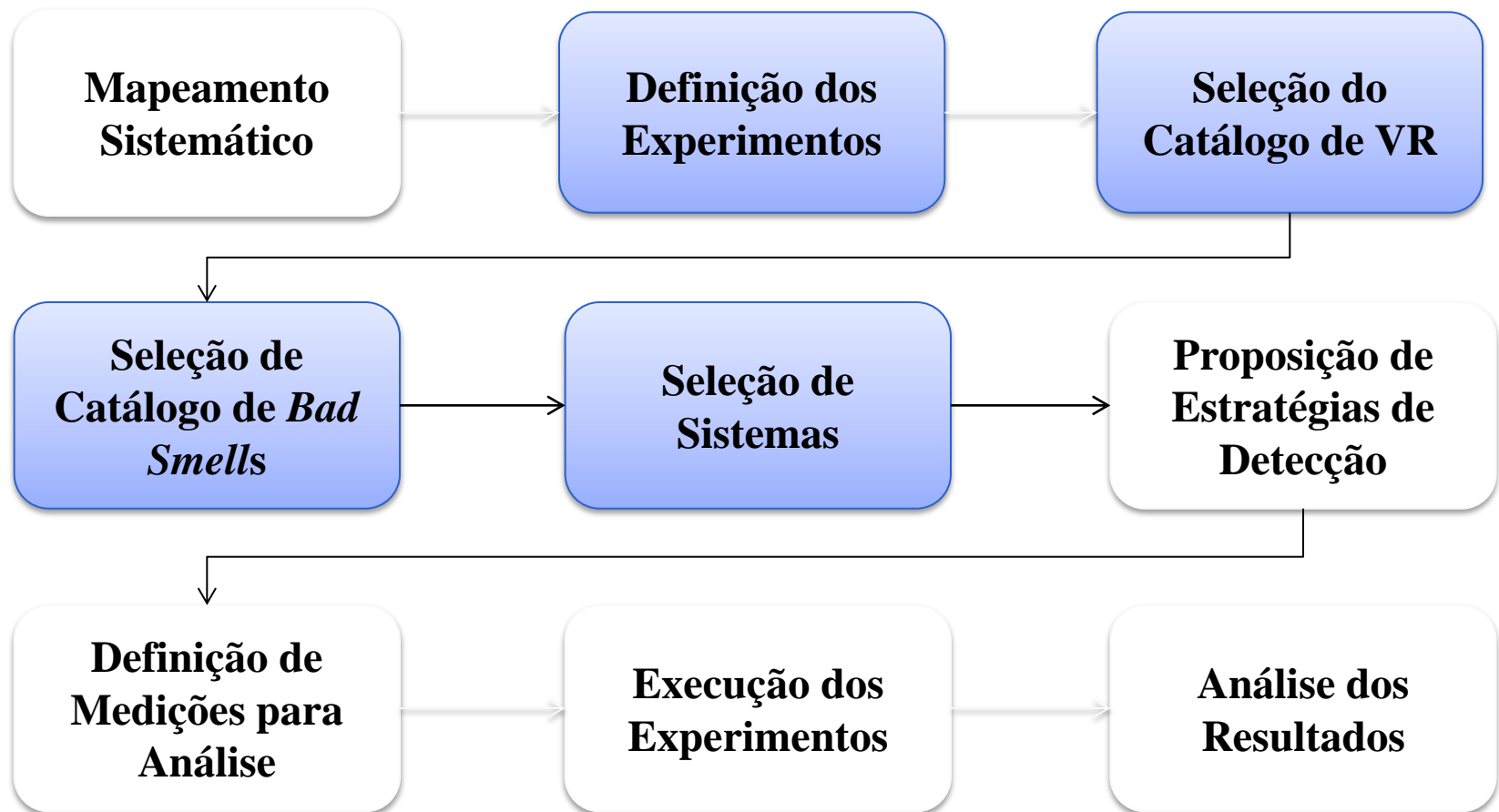
# Considerações

---

- Em geral, os trabalhos recuperados
  - são recentes, a partir 2010
  - poucas métricas e sistemas avaliados
  - conjunto de métricas CK é o mais abordado
  
- Portanto,
  - é necessário investigar mais métricas
    - com amostras de sistemas maiores
  - validar valores referência para mostrar sua importância e aplicabilidade

# Etapas do Estudo

---







# Definição dos Experimentos

---

# Estudos Realizados

---

- QP1. VR x *Bad Smells*:
  - Detecção automatizada
    - estratégias de detecção propostas
    - ferramentas *JSpIRIT* e *JDeodorant*
  - Inspeção manual de um especialista
  
- QP2. VR x Predição de Falhas
  - Apoiado por ferramenta de coleta de falhas
    - *BugMaps* [Couto et. al 2013]



# Seleção do Catálogo de Valores Referência

---

# Catálogo de Filó (2014) ...

---

- Justificativa para seleção do catálogo
  - Valores referência propostos para 18 métricas
- Valores referência derivados com base nas frequências de valores por métrica
  - *Bom/Frequente, Regular/Moderado, Ruim/Raro*

0%

70%

90%

100%

*Bom*

*Regular*

*Ruim*

# ... Catálogo de Filó (2014)

---

- Validação empírica na detecção de *bad smells*
  - *God Class e Long Method*
  - Utilizou-se 1 sistema de software proprietário



# Seleção de Catálogo de *Bad Smells*

---

# Catálogo de Fowler (1999)

---

- 22 *bad smells* definidos
  - Com foco em orientação por objetos (OO)
- Justificativa para seleção do catálogo
  - Catálogo pioneiro e amplamente difundido
- Exemplos: *Shotgun Surgery*, *Lazy Class*, etc.



# Seleção de Sistemas

---



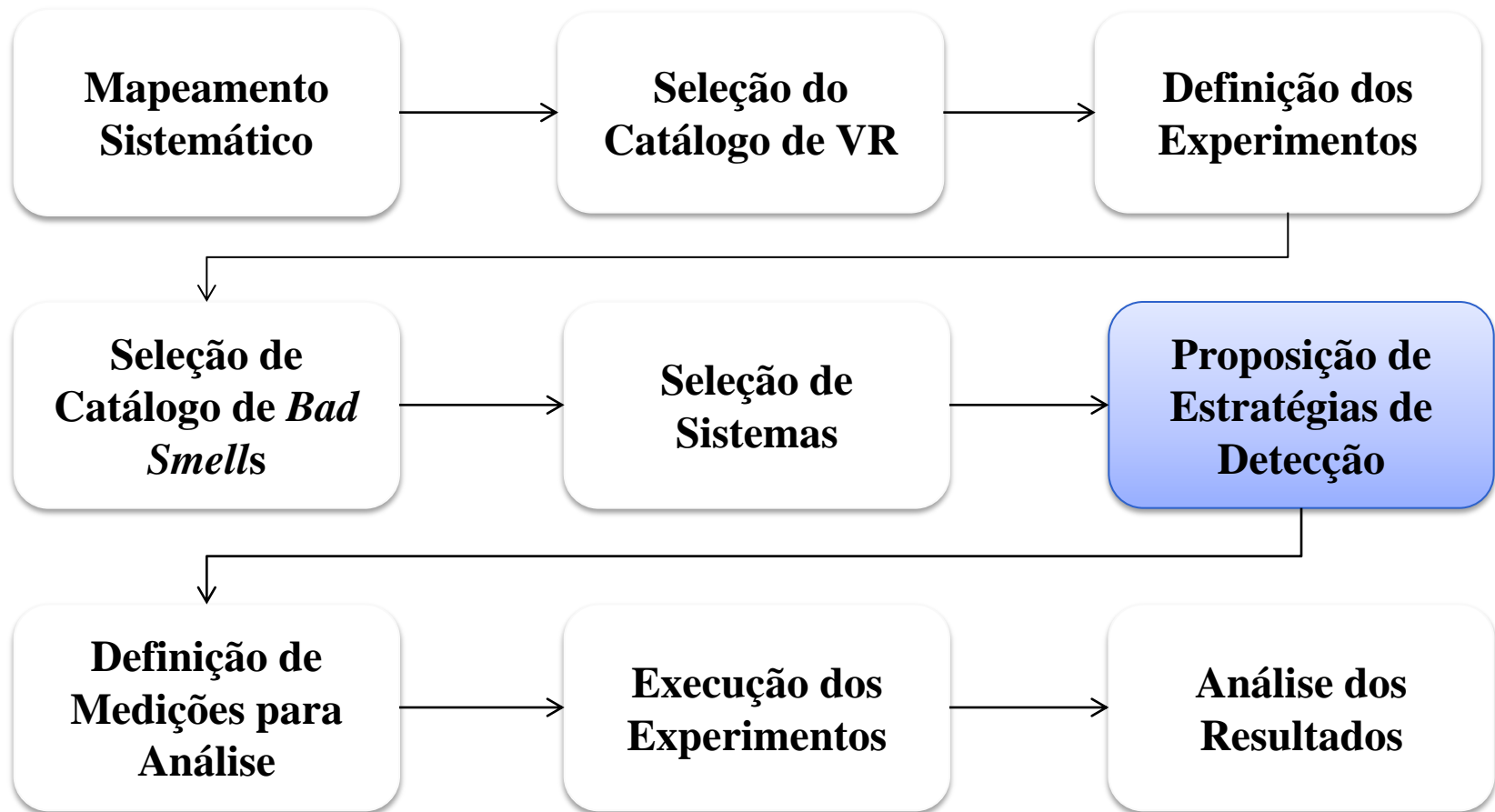
# *Qualitas.class Corpus (2013)*

---

- 111 sistemas Java de código-fonte aberto
  - 23 métricas de software calculadas
  
- Justificativa para seleção do *corpus*
  - Proposto para fins de avaliação de qualidade
  - Sistemas conhecidos: *Ant*, *Cobertura* e *Jedit*, etc.
  
- 22 sistemas avaliados, considerando-se os 2 estudos executados

# Etapas do Estudo

---



# Estratégias Propostas para Detecção de *Bad Smells*

---

# Estratégias de Detecção

---

- Compostas a partir de
  - Definições de cada *bad smell* (Fowler, 1999)
  - Valores referência de Filó (2014)
  - Métricas do *Qualitas.class Corpus* (2013)
  
- Propostas para 5 *bad smells*
  - *Large Class, Long Method, Data Class, Feature Envy e Refused Bequest*

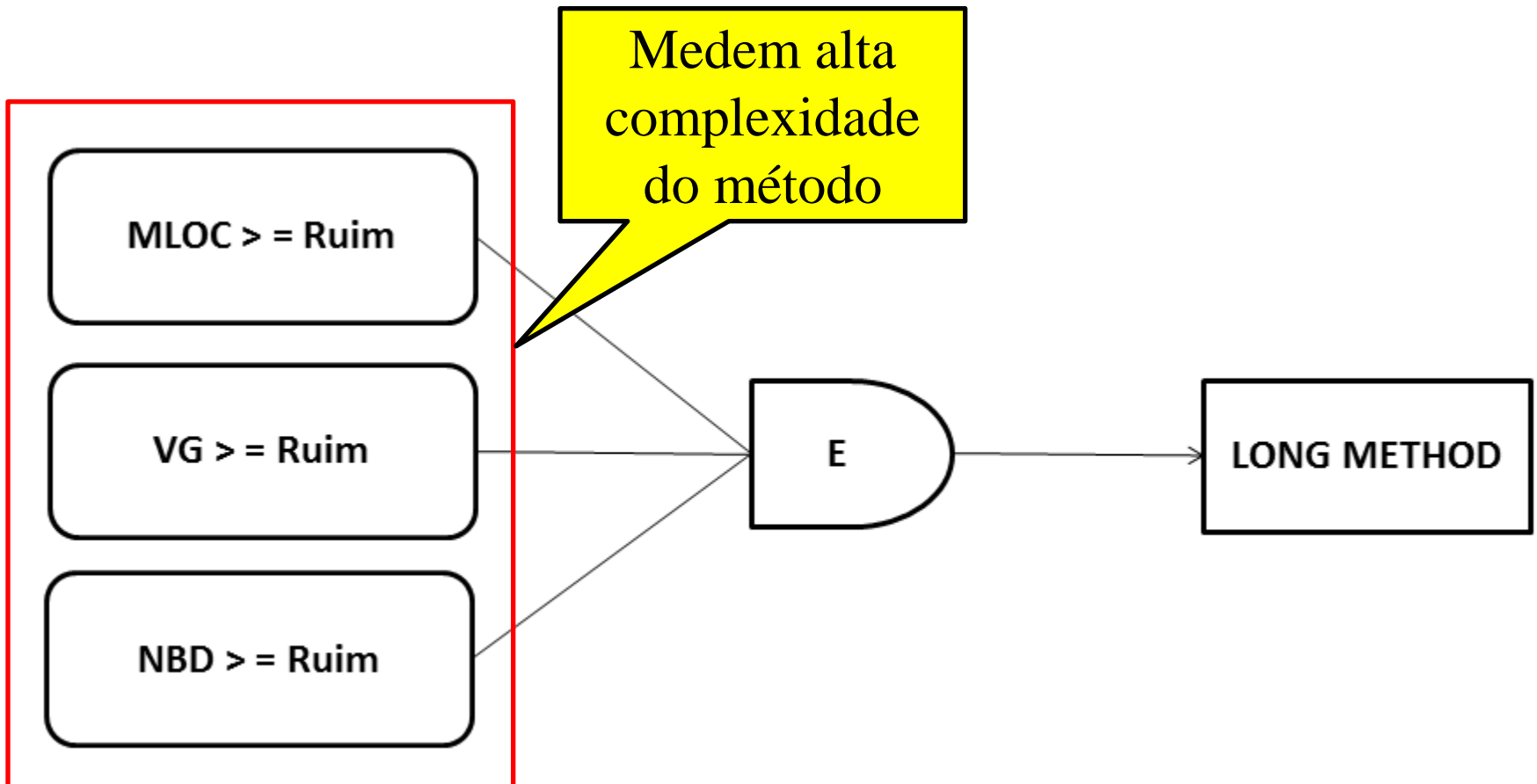
# Critérios para Seleção dos *Bad Smells*

---

- Falta de métricas adequadas para composição de estratégias
  - No *Qualitas.class* e em Filó (2014)
    - Exemplo: Filó (2014) não possui *Coupling between Objects (CBO)*, importante na detecção de *Shotgun Surgery*
- Falta de ferramentas disponíveis para detecção de alguns *bad smells*
- Limitações das ferramentas

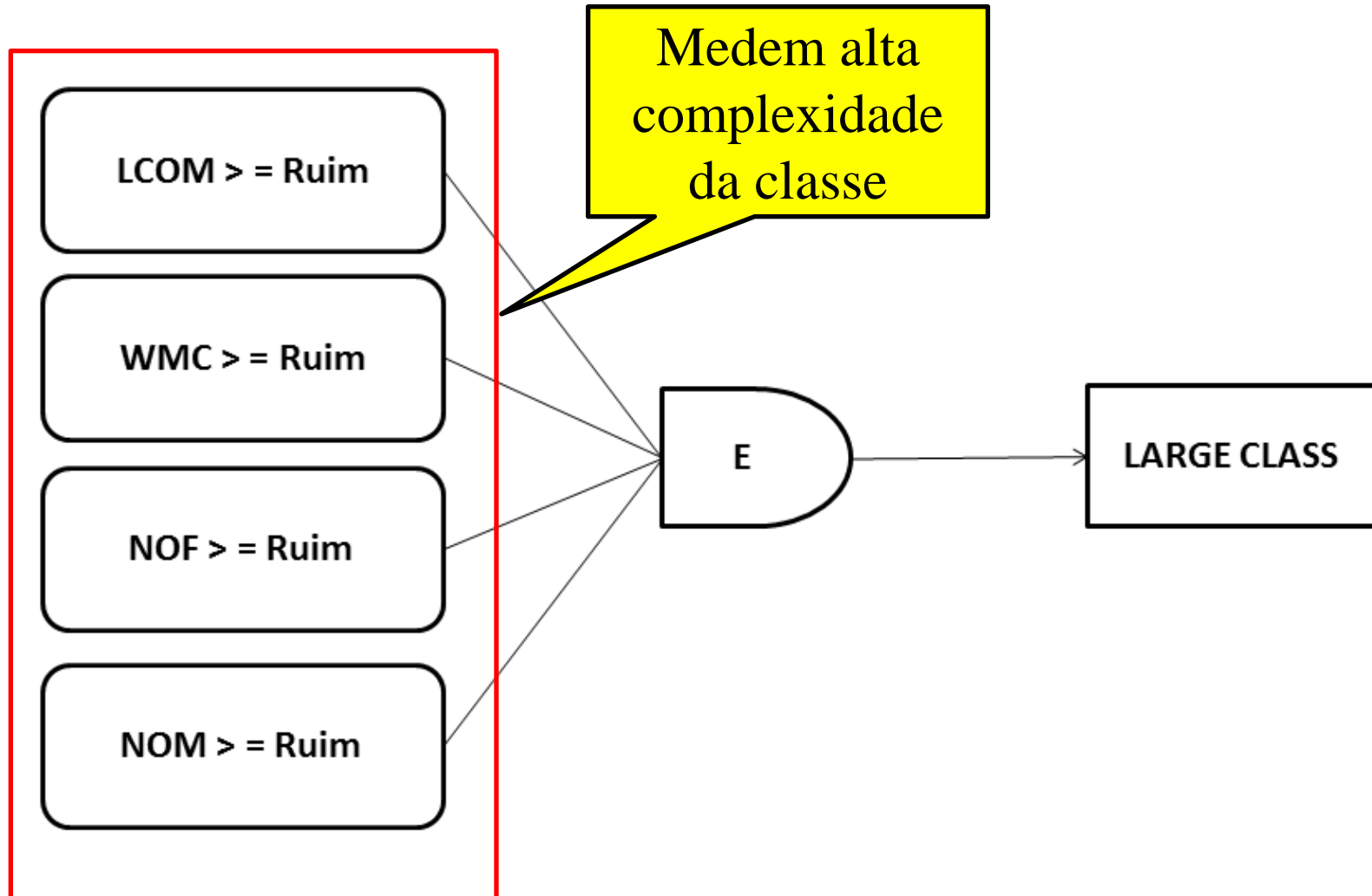
# Long Method

---



# Large Class

---



# Considerações

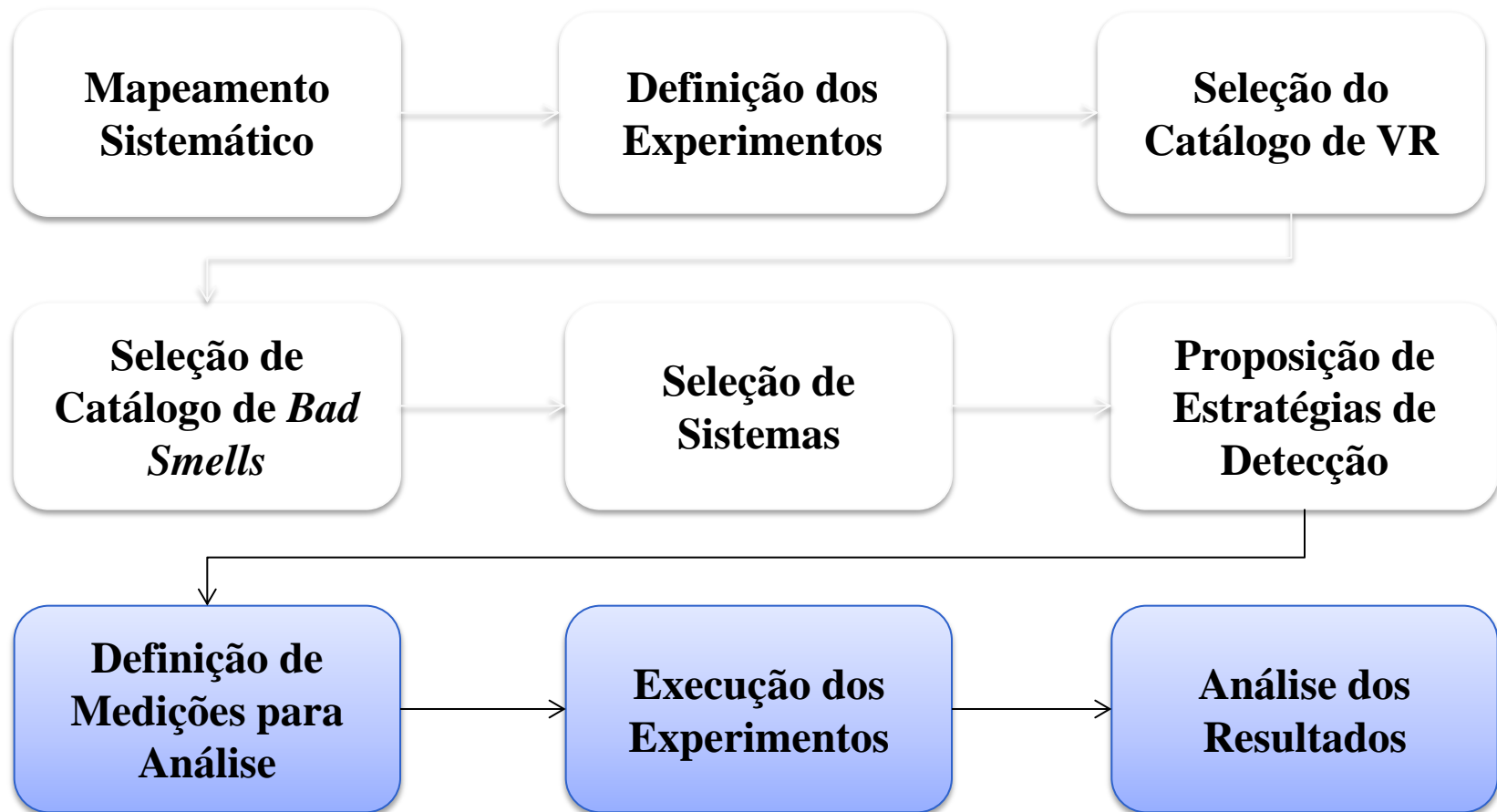
---

- Estratégias propostas
  - São heurísticas de apoio à detecção dos *bad smells*
  - Aplicam os valores referência de Filó (2014)
  
- *FindSmells*
  - Ferramenta que implementa as estratégias propostas



# Etapas do Estudo

---





# Definição de Medidas para Análise

---

# Medidas de Acurácia

---

- *Recall*
  - Completude, quantidade
  
- Precisão
  - Corretude, qualidade
  
- *F-Measure*
  - Balanceamento entre *Recall* e Precisão

# Execução dos Experimentos - Valores Referência e *Bad Smells*

---

# Etapas Específicas do Estudo

---

- Identificação de um conjunto de sistemas de software
  - *Qualitas.class Corpus 2013*
- Experimento com ferramentas de detecção
  - *JSpIRIT* (baseada em métricas): 5 bad smells
  - *JDeodorant* (híbrida): 3 bad smells
    - Exceto *Data Class e Refused Bequest*
- Experimento com detecção manual
  - Especialista
- Análise dos resultados dos experimentos

# Identificação de um Conjunto de Sistemas

---

- 12 sistemas Java do *Qualitas.class Corpus*
  - Pequeno a médio-porte
  - Importados com sucesso na IDE Eclipse
  - Analisados com sucesso pelas ferramentas
  
- Exemplos de sistemas
  - Apache *Ant*, *Cobertura*, *WebMail*, etc.

# VR x Detecção Automatizada

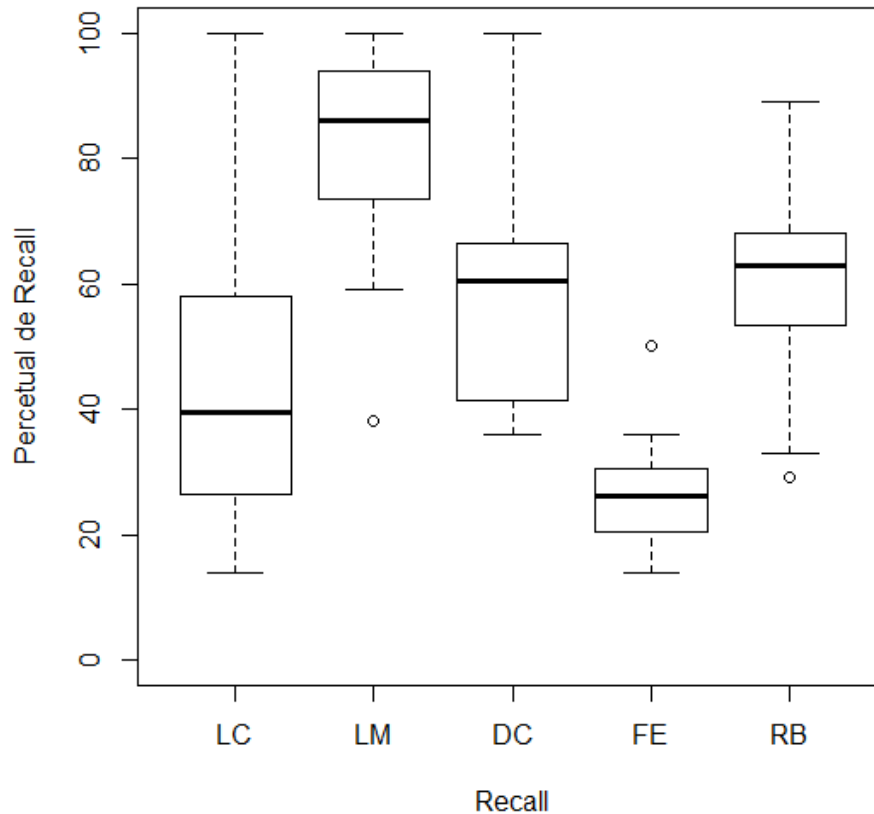
---

**QP1.1.** Qual é a eficácia da detecção de *bad smells* utilizando-se estratégias baseadas nos valores referência e tomando-se como base os resultados gerados por ferramentas de detecção de *bad smells*?

# *JSpIRIT* – Distribuição de *Recall* por *Bad Smell*

---

Distribuição de Recall por Bad Smell



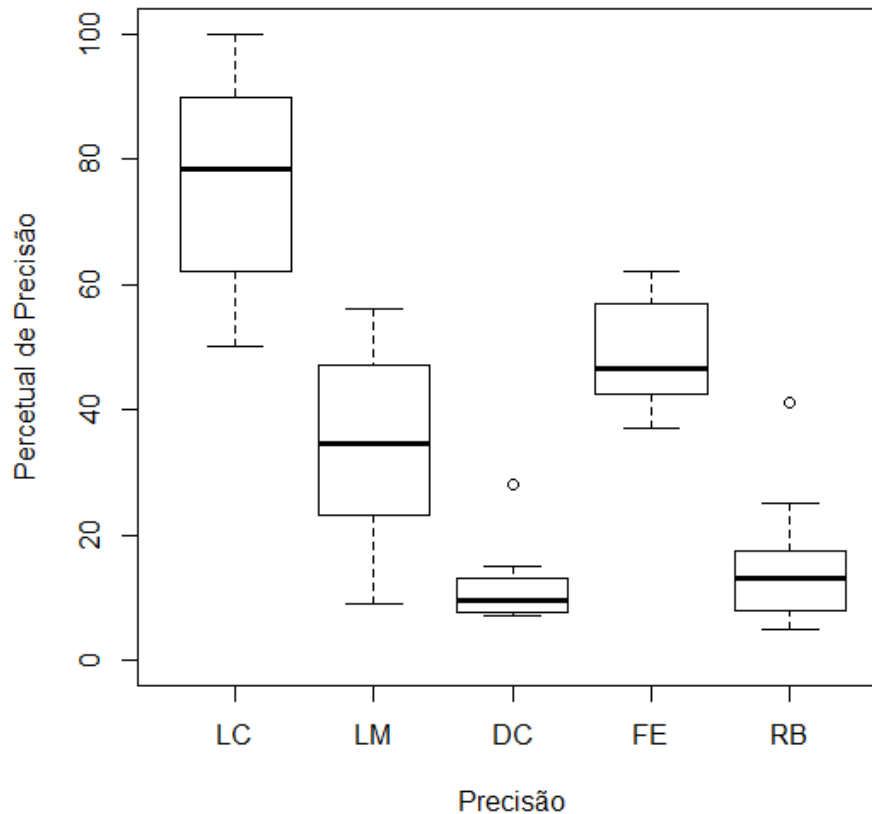
LC – Large Class  
LM – Long Method  
DC – Data Class  
FE – Feature Envy  
RB – Refused Bequest



# *JSpIRIT* – Distribuição de Precisão por *Bad Smell*

---

Distribuição de Precisão por Bad Smell



LC – Large Class  
LM – Long Method  
DC – Data Class  
FE – Feature Envy  
RB – Refused Bequest

# Resultados para *JSpIRIT*

---

- *Recall* alto (mediana  $\geq 60\%$ ) para *Long Method*, *Data Class* e *Refused Bequest*
- Precisão alta (mediana  $\geq 45\%$ ) somente para *Large Class* e *Feature Envy*
  - Estratégias retornaram muitos falsos positivos
- *F-Measure* moderado (mediana  $\geq 47\%$ ) para *Large Class* e *Long Method*

# Inspeção dos Resultados da *JSpIRIT*

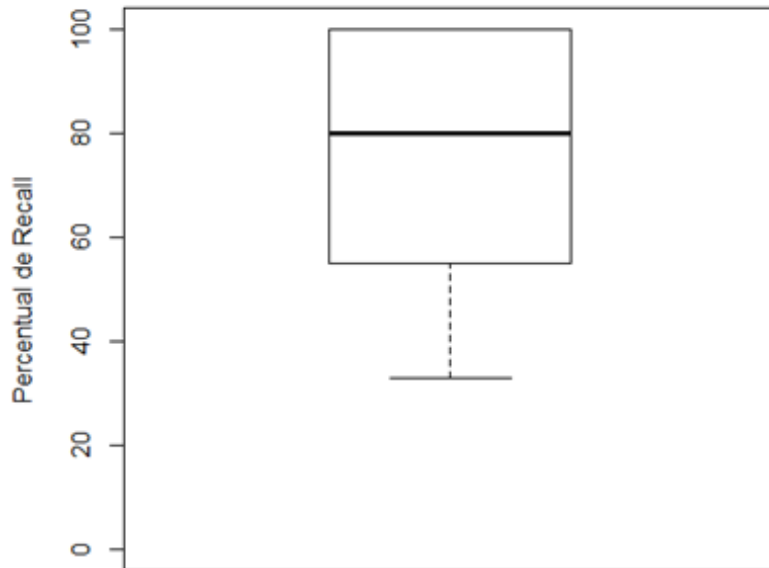
---

- ❑ Conduzida com apoio de um especialista
- ❑ Somente 2 sistemas foram selecionados, por limitações diversas
  - *Squirrel SQL e WebMail*
- ❑ O especialista identificou falsos positivos para *Large Class*, *Feature Envy* e *Refused Bequest*

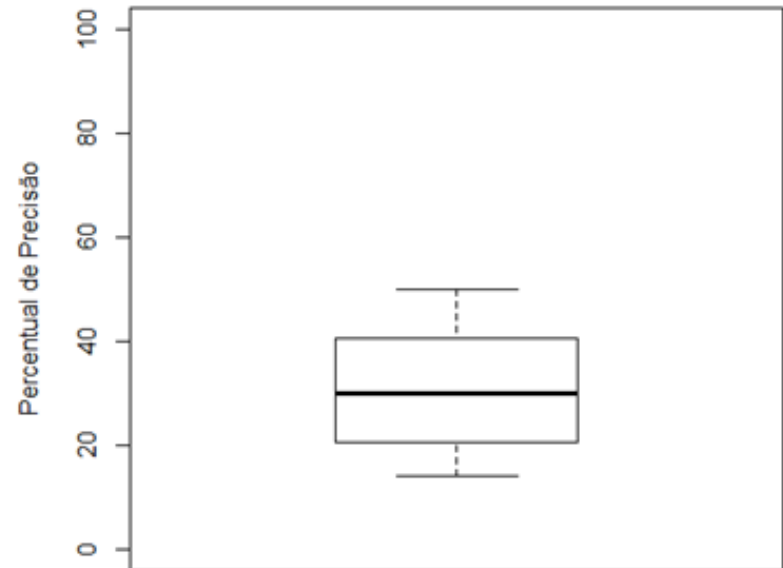
# Inspeção dos Resultados da *JSpIRIT*

---

Distribuição de Recall para Instâncias Validadas



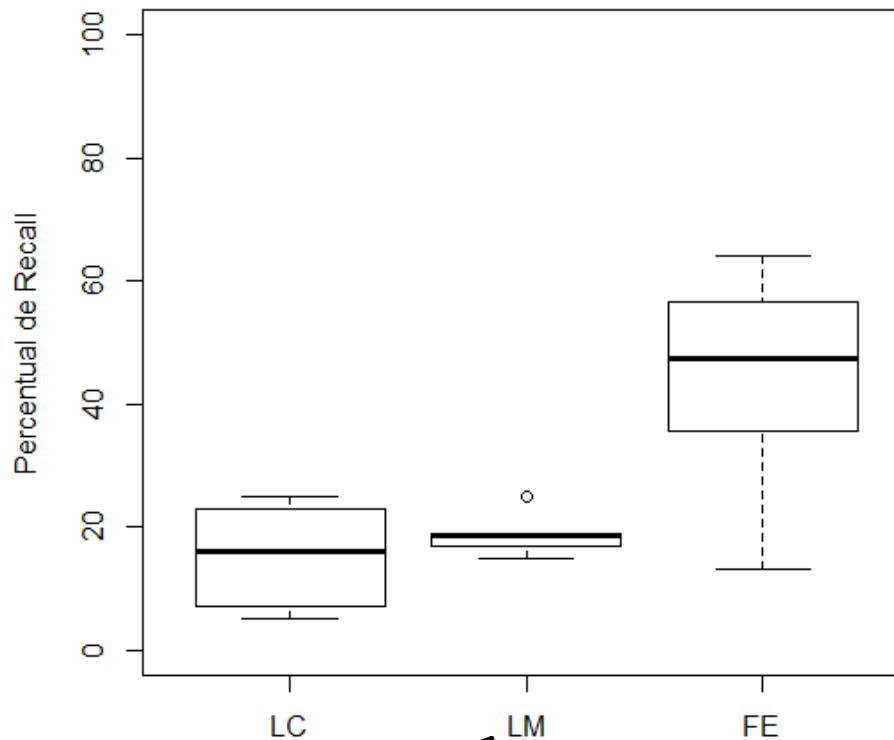
Distribuição de Precisão para Instâncias Validadas



# *JDeodorant* - Distribuição de *Recall* por *Bad Smell*

---

Distribuição de Recall por Bad Smell



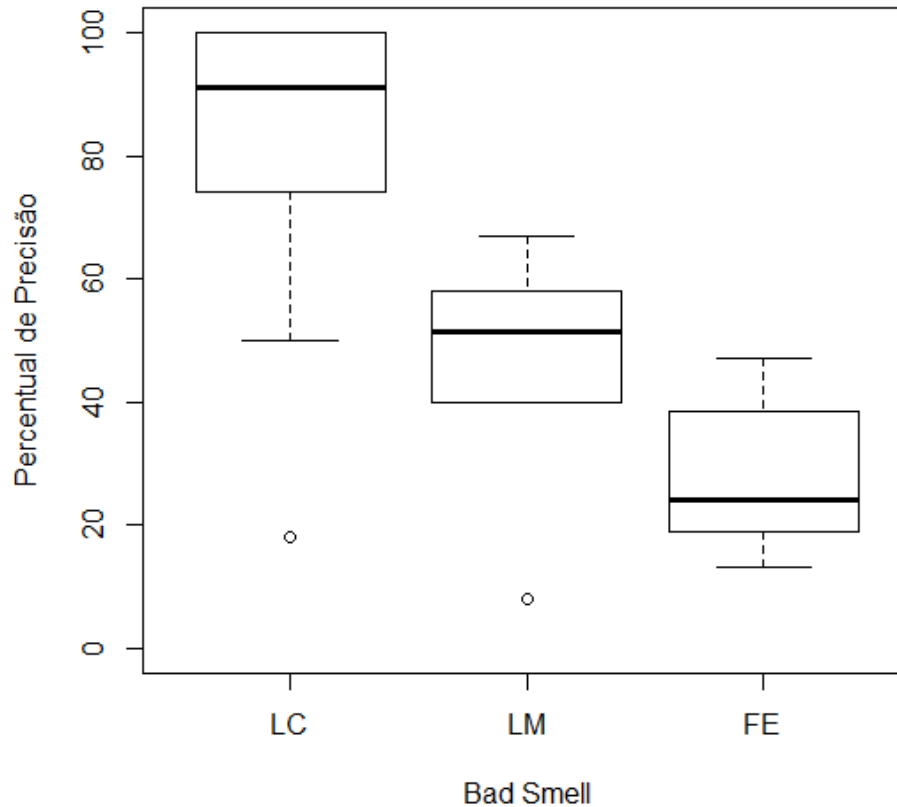
LC – Large Class  
LM – Long Method  
FE – Feature Envy

Sem instâncias para  
alguns sistemas

# *JDeodorant* - Distribuição de Precisão por *Bad Smell*

---

Distribuição de Precisão por Bad Smell



LC – Large Class  
LM – Long Method  
FE – Feature Envy

# Resultados para *JDeodorant*

---

- *Recall* baixo (no máximo 47%) para *Large Class*, *Long Method* e *Feature Envy*
- Precisão moderada a alta (mediana  $\geq 51\%$ ) para *Large Class* e *Long Method*
- *F-Measure* com mediana de até 30% para *Large Class*, *Long Method* e *Feature Envy*
  - Causada pela baixa concentração de *recall*

# VR x Detecção Manual

---

**QP1.2.** Os valores referência apoiam efetivamente a detecção de *bad smells* em relação a listas de referência geradas por um especialista com conhecimentos em orientação por objetos e *bad smells*?



# VR x Detecção Manual

---

- Ferramentas automatizadas
  - Acurácia dos seus resultados não é conhecida
  
- Especialista
  - Mestrando em Ciência da Computação na área de Engenharia de Software
  - Possui conhecimento
    - Orientação por objetos (OO)
    - *Bad Smells*

# VR x Detecção Manual

---

- 1 sistema avaliado
  - *Apache Maven*
  
- Duas etapas
  1. Identificação de instâncias de: *Large Class, Long Method, Data Class e Refused Bequest*
    - *Featery Envy* não foi considerada
  2. Comparação dos resultados com as estratégias de detecção propostas
    - *Recall, Precisão e F-measure*

# Resultados - VR x Detecção Manual

---

<i>Bad Smell</i>	<i>Recall</i>	<i>Precisão</i>	<i>F-measure</i>
	%	%	%
<i>Large Class</i>	70	37	48
<i>Long Method</i>	73	19	30
<i>Data Class</i>	47	32	38
<i>Refused Bequest</i>	100	2	4
Média	72,5	22,5	30
Desvio Padrão	21,7	15,6	18,8

RB valor de  
precisão  
muito baixo!

# Resultados - VR x Detecção Manual

---

- Análise do especialista mostrou-se restritiva e subjetiva
  - Precisão pequena: 23%
    - Principalmente para *Refused Bequest*
  - Porém, *Recall* foi alto, em torno de 70%
  - *F-measure* igual a 30%
  
- Em geral, não se observou discrepância entre os resultados das estratégias e do especialista



# Execução dos Experimento - Valores Referência e Predição de Falhas

---

# VR x Predição de Falhas

---

**QP2.** Os valores referência de métricas de software orientados por objetos auxiliam a predizer falhas em um software?

# Etapas Específicas do Estudo

---

- ❑ Seleção da ferramenta de coleta de falhas
- ❑ Seleção dos sistemas de software para análise
- ❑ Extração dos dados de falhas
- ❑ Classificação das medidas de acordo com as faixas de valores referência
- ❑ Análise dos resultados

# Seleção de Ferramenta para Coleta de Falhas

---

- *BugMaps* (Couto, 2013)
  - Coleta dados de falhas por classe do software
  - Exporta os resultados para CSV
  
- 10 métricas de classe foram avaliadas
  - DIT, LCOM, NOF, NOM, NORM, NSC, NSF, NSM, SIX e WMC



# Seleção de Sistemas

---

- 10 sistemas do *Qualitas.class Corpus*
  - Com dados de falhas disponíveis no *Jira* ou *Bugzilla*
  - Exemplos: *AspectJ*, *Cayenne* e *JMeter*

# Extração dos Dados de Falhas

---

- Dados de falhas coletados
  - Jira e Bugzilla
  
- Execução da *BugMaps*
  - Arquivo de falhas como entrada
  - Definido período para filtrar falhas
    - 2013 a 2016
  - Associação das falhas com as classes dos sistemas avaliados

# Classificação das Métricas

---

- A partir do arquivo de métricas do *Qualitas.class*
  - Verificou-se, para cada métrica, a faixa de valores correspondente
- Faixas para classificação das métricas
  - *Bom, Regular e Ruim*
- *FindSmells*
  - Automatizou o processo de classificação

# Resultados - VR x Predição de Falhas

---

- Para 7 das 10 métricas, a maior concentração de falhas está na faixa *Ruim*
  
- Para SIX, obteve-se
  - A maior concentração de classes com falha a partir da faixa *Regular* até *Ruim*
  
- Quanto a DIT e NSC
  - As faixas *Regular* e *Ruim* não se mostraram bons indicadores de falhas

# Resultados - VR x Predição de Falhas

Criticidade da faixa cresce,  
% de falhas cresce

Métrica	<i>Bom</i>	<i>Regular</i>	<i>Ruim</i>
	%	%	%
DIT	21,00	20,88	20,73
LCOM	16,39	23,61	35,08
NOF	17,09	29,59	41,53
NOM	15,22	24,82	39,77
NORM	20,10	28,79	31,21
NSC	21,17	20,69	18,17
NSF	17,52	30,34	39,31
NSM	19,26	33,11	39,18
SIX	19,45	25,10	21,43
WMC	11,57	24,73	44,37
Média	17,88	26,17	33,08
Desvio Padrão	2,98	4,13	9,65

Maior % de  
falhas na  
faixa *Ruim*

# Resultados - VR x Predição de Falhas

---

Métrica	<i>Bom</i>	<i>Regular</i>	<i>Ruim</i>
	%	%	%
DIT	21,00	20,88	20,73
LCOM	16,39	23,61	35,08
NOF	17,09	29,59	41,53
NOM	15,22	24,82	39,77
NORM	20,10	28,79	31,21
NSC	21,17	20,69	18,17
NSF	17,52	30,34	39,31
NSM	19,26	33,11	39,18
SIX	19,45	25,10	21,43
WMC	11,57	24,73	44,37
Média	17,88	26,17	33,08
Desvio Padrão	2,98	4,13	9,65

WMC métrica  
com maior %  
em Ruim



# Ameaças à Validade

---

# Principais Ameaças e Tratamentos

---

- Poucos artigos encontrados no mapeamento
  - Termos de busca variadas
  
- Qualidade dos resultados das ferramentas não é conhecida
  - Resultados usados somente como referencial
  - Especialista
  
- Repositórios públicos de dados de falhas
  - Amplamente utilizados





# Conclusão

---

# Conclusão

---

- Constatou-se que valores referência auxiliam
  - a identificação de *bad smells* (QP1)
    - Em geral, observou-se alto *Recall* e valores moderados de Precisão e *F-measure* por *bad smell*
  - a predição de falhas em software (QP2)
    - Faixas *Regular* e *Ruim* mostraram-se eficazes
    - Quanto maior a criticidade da faixa de VR maior o percentual de classes com falhas

# Conclusão

---

- Principais contribuições
  - Mapeamento sistemático sobre VR x *bad smells* e predição de falhas
  - Avaliação dos VR de Filó (2014) na detecção de *bad smells* (QP1)
  - Avaliação dos VR de Filó (2014) na predição de falhas (QP2)
  - Ferramenta *FindSmells* para detecção de *bad smells*
  
- Publicação de artigo
  - XIV Workshop de Teses e Dissertações em Qualidade de Software (WTDQS), do Simpósio Brasileiro de Qualidade de Software (SBQS 2016)

# Conclusão

---

## □ Trabalhos futuros

- Catálogo de Filó (2014) X padrões de projeto, projeto arquitetural, etc.
- Avaliar o catálogo de Filó (2014) com sistemas em outras linguagens de programação
- Replicação do estudo com outros catálogos de valores referência



# Muito Obrigada!

---

Priscila Pereira de Souza

Orientadora: Prof<sup>a</sup>. Mariza A. S. Bigonha (UFMG)

Coorientadora: Prof<sup>a</sup>. Kecia A. M. Ferreira (CEFET-MG)

Deus seja louvado!

# Anexos

---

# Questões do Mapeamento

---

**QP1.** Como a literatura tem aplicado valores referência de métricas de softwares orientados por objeto para a detecção de *bad smells*?

**QP2.** Como a literatura tem aplicado valores referência de métricas de softwares orientados por objeto para a predição de falhas em software?

# Seleção das Fontes de Pesquisa

---

- Portal da CAPES
  - Amplo acervo: mais de 37 mil títulos com texto completo, 126 bases referenciais, etc.
  - Possui acesso ao IEEE e ACM, dentre outros.



# Strings do Mapeamento

---

1. *(METRIC OR METRICS) AND (“THRESHOLD” OR “THRESHOLDS” OR “REFERENCE VALUE”) AND SOFTWARE AND (“BAD SMELL” OR “BAD SMELLS” OR “CODE SMELL” OR “CODE SMELLS” OR “DESIGN FLAW” OR “ANTI-PATTERNS” OR “DESIGN DEVIANCE” OR “MODULARITY ANOMALIES”)*.
2. *(METRIC OR METRICS) AND (“THRESHOLD” OR “THRESHOLDS” OR “REFERENCE VALUE”) AND SOFTWARE AND (“FAULT PREDICTION” OR “FAILURE PREDICTION” OR “BUG PREDICTION”)*.

# Critérios de Inclusão e Exclusão

---

## □ Critérios de Inclusão

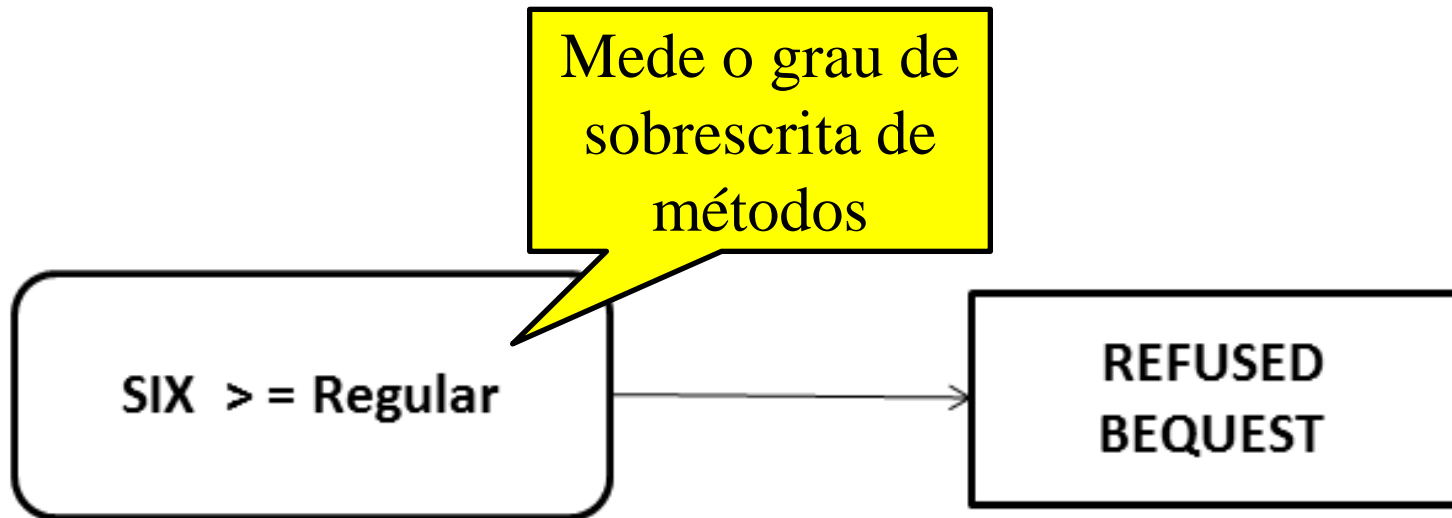
- Os documentos devem apresentar claramente uma relação entre os termos relacionados nas strings de busca
- Os documentos devem estar completos

## □ Critérios de Exclusão

- Documentos duplicados
- Tutoriais, pôsteres, painéis, palestras, mesas redondas, oficinas

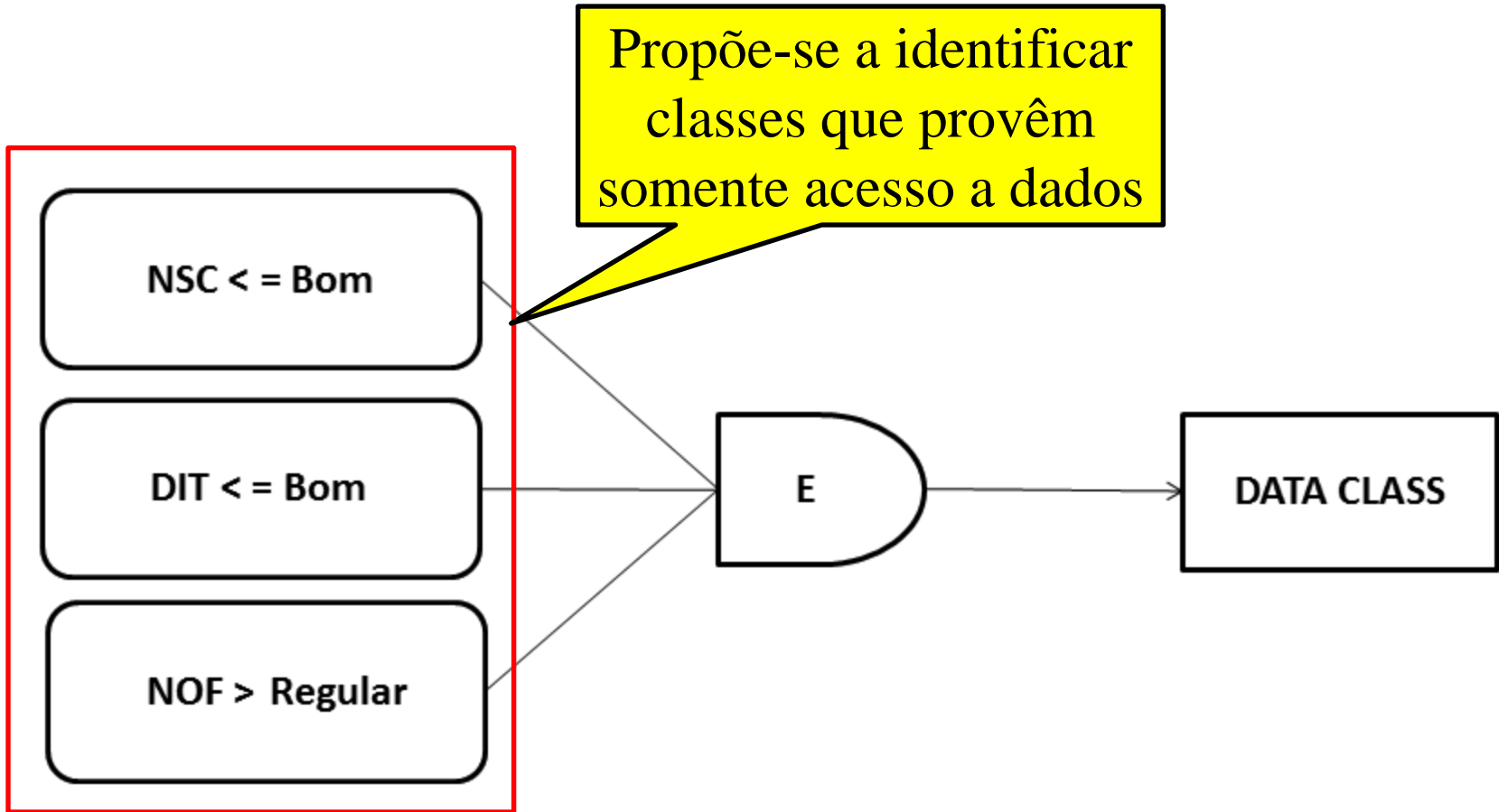
# Refused Bequest

---



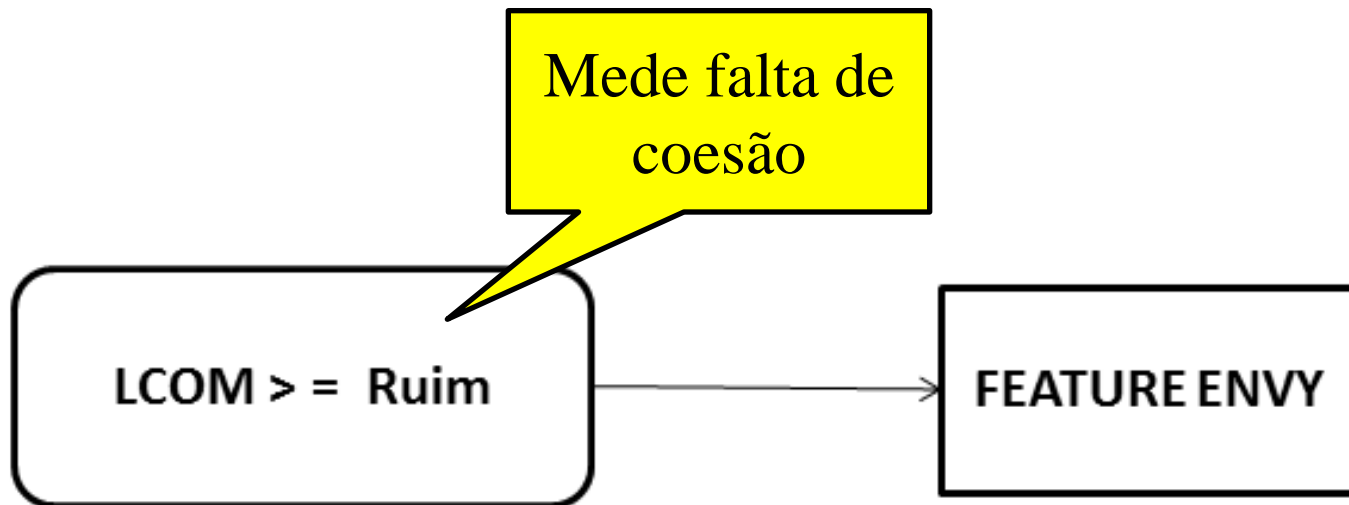
# Data Class

---



# Feature Envy

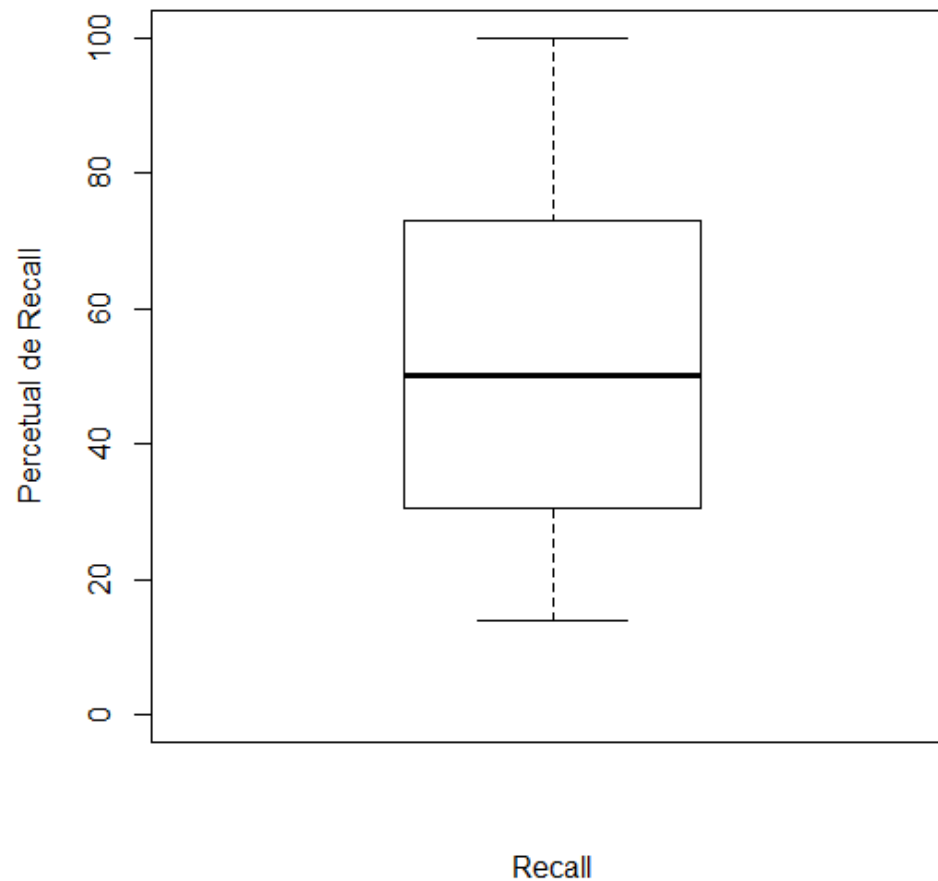
---



# *JSpIRIT* – Distribuição de *Recall* para Todos Bad Smells

---

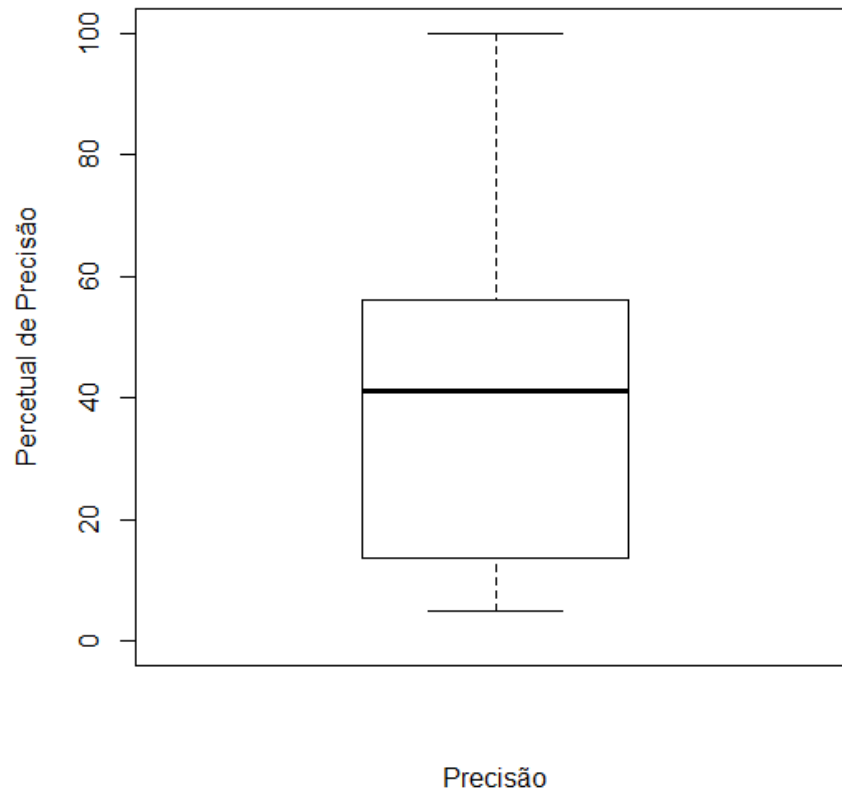
Distribuição de Recall para Todos os Bad Smells



# *JSpIRIT* – Distribuição de Precisão para Todos os *Bad Smells*

---

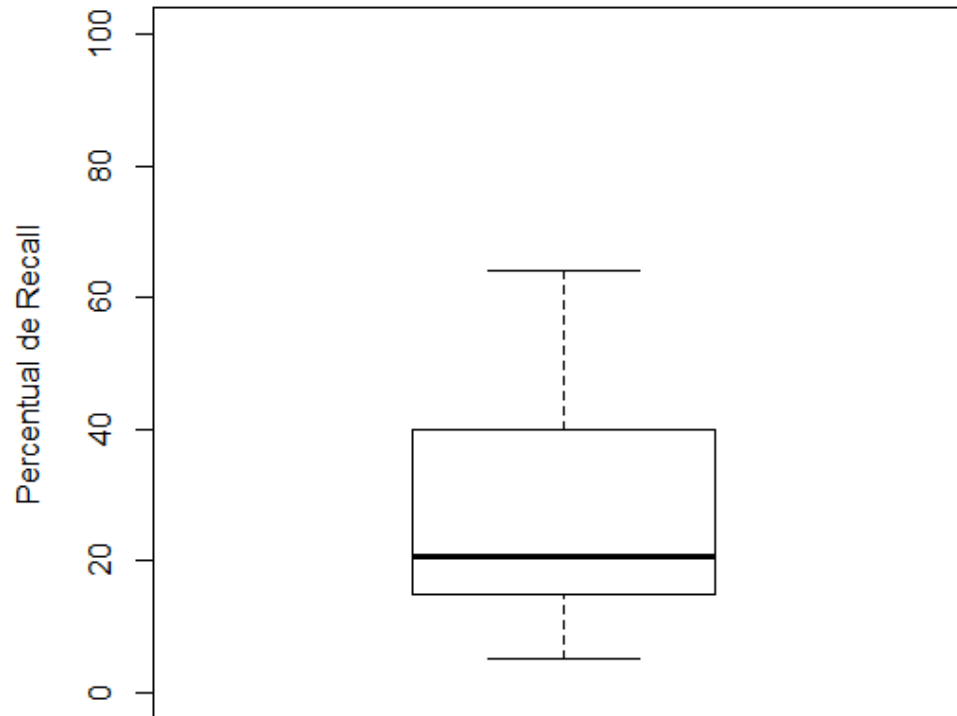
**Distribuição de Precisão para Todos os Bad Smells**



# *JDeodorant* - Distribuição de *Recall* para todos os *Bad Smells*

---

Distribuição de Recall para Todos os Bad Smells

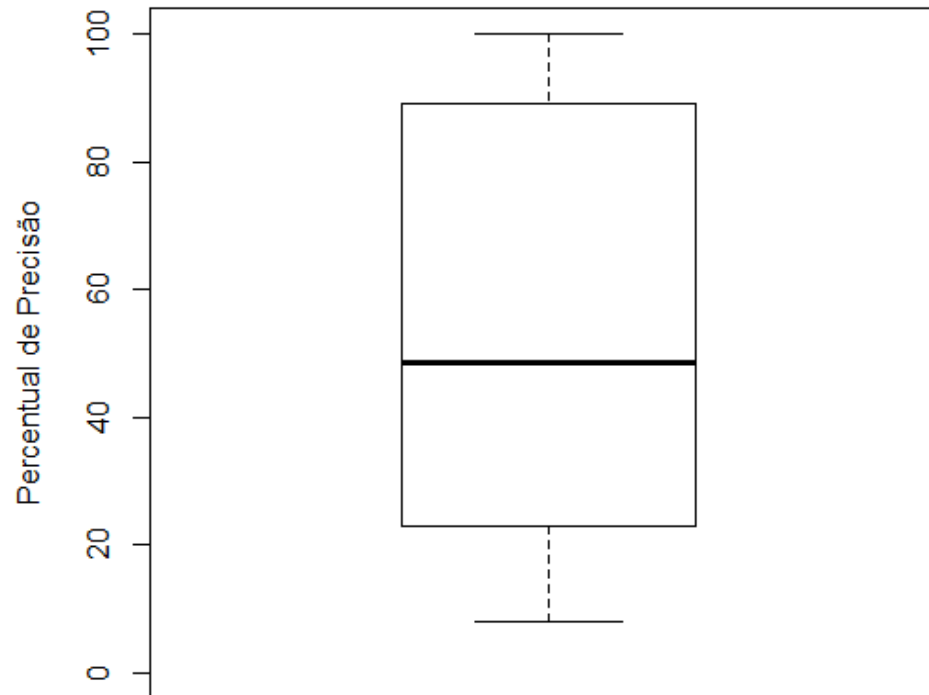




# JDeodorant - Distribuição de Precisão para todos os *Bad Smell*

---

Distribuição de Precisão para Todos os Bad Smells



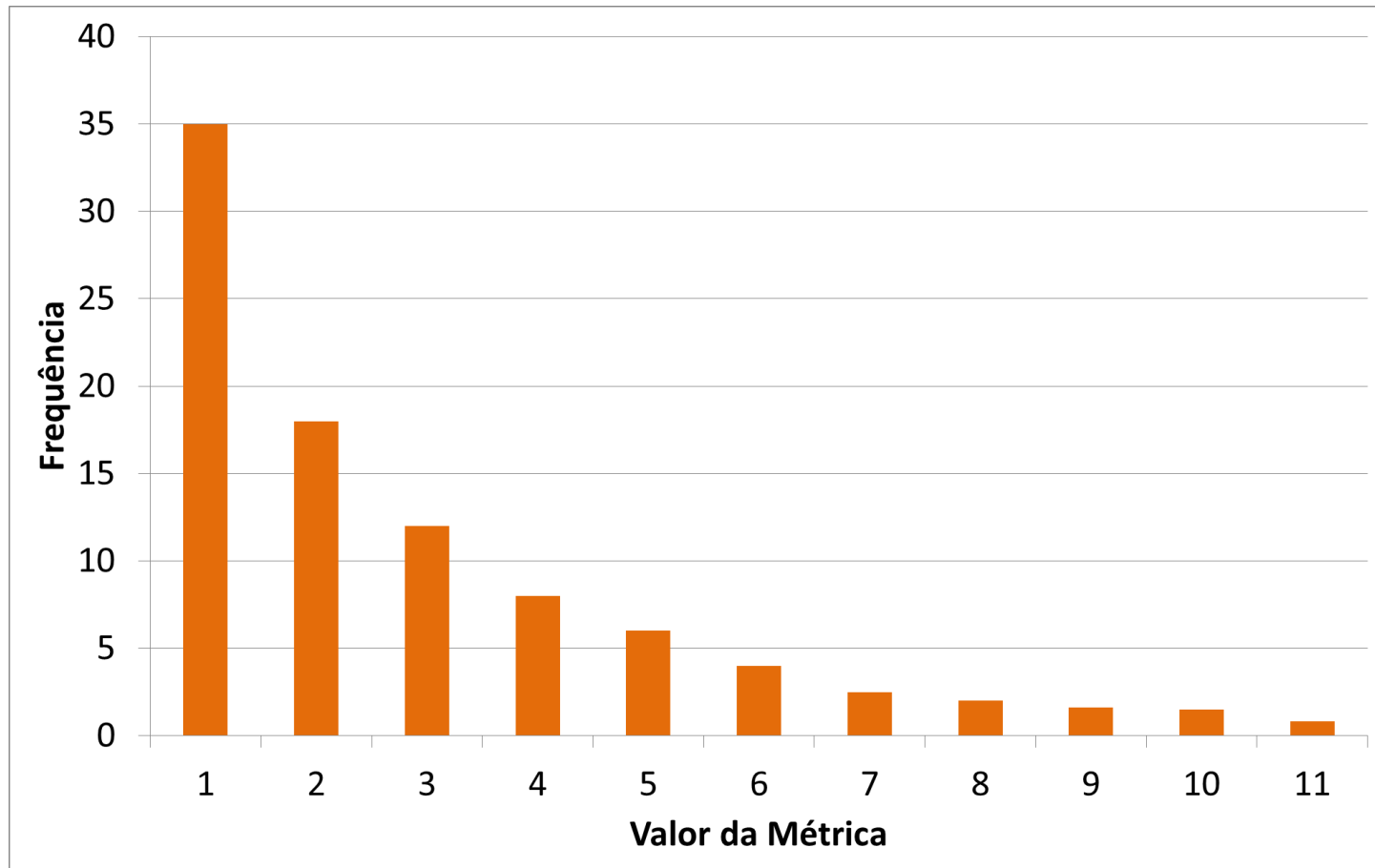
# Derivação de VR de Filó (2014)

---

- Considera o histograma de frequência para cada métrica
  - Valores mais frequentes são considerados bons (faixa *Bom*)
  - Valores com frequência intermediária são ditos moderados (faixa *Regular*)
  - Valores menos frequentes são ditos ruins (faixa *Ruim*)

# ... Derivação de VR de Filó (2014)

---



# ... Derivação de VR de Filó (2014)

---

