Experimental Assessment of Software Metrics Using Automated Refactoring

Mel Ó Connéide
University College Dublin

Laurence Tratt
King's College London

Mark Harman
University College London

Steve Counsell
Brunel University

Iman Hemati Moghadam
University College Dublin

Empirical Software Engineering and Management (ESEM)
August 2012

Gustavo Jansen de Souza Santos
Universidade Federal de Minas Gerais

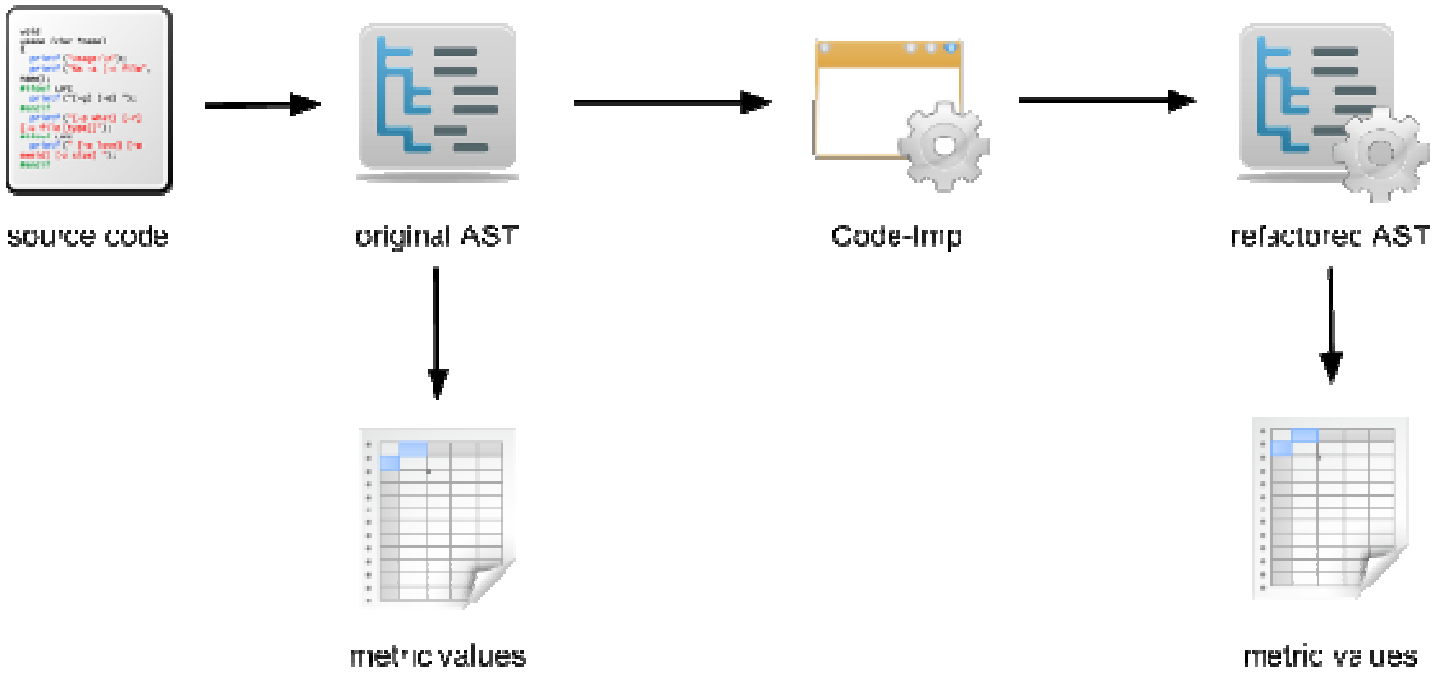
Motivação

- Há uma diversidade de métricas na literatura
  - Mais de quatro décadas
  - Difícil avaliar as próprias métricas
  - Como se relacionam entre si

- Métricas cobrem diversos objetivos
  - Cobrem diferentes aspectos
  - Mesma propriedade, mesmo comportamento?

Abordagem Experimental

- Aplicada uma sequência de refatorações
  - Medição do valor das métricas antes e depois
  - Identificar valores conflitantes



Search-Based Refactoring

- Algoritmo de otimização
  - *Hill Climbing*
  - Guiado por métricas de qualidade

- Função de *fitness*
  - Pequenas refatorações que melhorem a função

Code-Imp

A LIST OF IMPLEMENTED REFACTORINGS IN CODE-IMP

No.	Class-Level Refactorings	Description
1	Extract Hierarchy	Adds a new subclass to a non-leaf class C in an inheritance hierarchy.
2	Collapse Hierarchy	Removes a non-leaf class from an inheritance hierarchy.
3	Make Superclass Concrete	Removes the explicit *abstract* declaration of an abstract class without abstract methods.
4	Make Superclass Abstract	Declares a constructorless class explicitly abstract.
5	Replace Inheritance with Delegation	Replaces a direct inheritance relationship with a delegation relationship.
6	Replace Delegation with Inheritance	Replaces a delegation relationship with a direct inheritance relationship.
	Method-Level Refactorings	
7	Push Down Method	Moves a method from a class to those subclasses that require it.
8	Pull Up Method	Moves a method from some class(es) to the immediate superclass.
9	Decrease Method Accessibility	Decreases the accessibility of a method from protected to private or from public to protected.
10	Increase Method Accessibility	Increases the accessibility of a method from protected to public or from private to protected.
	Field-Level Refactorings	
11	Push Down Field	Moves a field from a class to those subclasses that require it.
12	Pull Up Field	Moves a field from some class(es) to the immediate superclass.
13	Decrease Field Accessibility	Decreases the accessibility of a field from protected to private or from public to protected.
14	Increase Field Accessibility	Increases the accessibility of a field from protected to public or from private to protected.

- Refatorações não refletem um cenário real
  - Observar comportamento das métricas
- Tornam o sistema melhor segundo a função *fitness*

Métricas

$$\mathrm{LSCC}(c) = \begin{cases} 0 & \text{if } l{=}0 \text{ and } k > 1, \\ 1 & \text{if } (l > 0 \text{ and } k{=}0) \text{ or } k{=}1, \\ \sum_{i=1}^{l} x_i(x_i-1)/lk(k-1) & \textit{otherwise.} \end{cases}$$

The similarity between two methods is the collection of their direct and indirect shared attributes.

$$\mathrm{TCC}(c) = \frac{|\{(m1,m2)|m1,m2 \in M_I(c) \wedge m1 \neq m2 \wedge \mathrm{cau}(m1,m2)\}|}{k(k-1)/2}$$

Two Methods interact with each other if they directly or indirectly use an attribute of class c in common.

$$\mathrm{CC}(c) = 2\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} \frac{|I_i \cap I_j|}{|I_i \cup I_j|}/k(k-1)$$

The similarity between two methods is the ratio of the collection of their shared attributes to the total number of their referenced attributes.

$$\mathrm{SCOM}(c) = 2\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} \frac{|I_i \cap I_j|}{\min(|I_i|,|I_j|)} * \frac{|I_i \cup I_j|}{l}/k(k-1)$$

The similarity between two methods is the ratio of the collection of their shared attributes to the minimum number of their referenced attributes. Connection intensity of a pair of methods is given more weight when such a pair involves more attributes.

$$\mathrm{LCOM5}(c) = \frac{k - \frac{1}{l}\sum_{a \in A_I(c)} |\{m|m \in M_I(c) \wedge a \in \mathrm{I_m}\}|}{k-1}$$

Measures the lack of cohesion of a class in terms of the proportion of attributes each method references. Unlike the other metrics, LCOM5 measures *lack* of cohesion, so a lower value indicates better cohesion.

1ª Investigação

```
Input: set of classes in program being refactored
Input: set of 14 refactoring types (e.g. PullUpMethod)
Input: set of metrics to be analysed
Output: metrics profile
refactoring_count = 0
repeat
    classes = set of classes in program
    while !empty(classes) do
        class = classes.pick()
        refactoring_types = set of refactoring types
        while !empty(refactoring_types) do
            refactoring_type =refactoring_types.pick()
            refactorings.populate(refactoring_type, class)
            if !empty(refactorings) then
                refactoring = refactorings.pick()
                refactoring.apply()
                if fitness_function_improves() then
                    refactoring_count++
                    update metrics profile
                else
                    refactoring.undo()
                end
            end
        end
    end
until refactoring_count == desired_refactoring_count;
```

Dataset

- Oito sistemas de código aberto
  - Total de 3453 refatorações
  - Natureza da aplicação influiu na coleta

- Propriedades medidas
  - Volatilidade
  - Probabilidade de mudança positiva

System	Description	LOC	#Classes	#refactorings
JHotDraw 5.3	Graphics	14,577	208	1007
XOM 1.1	XML API	28,723	212	193
ArtofIllusion 2.8.1	3D modeling	87,352	459	593
GanttProject 2.0.9	Scheduling	43,913	547	750
JabRef 2.4.2	Graphical	61,966	675	257
JRDF 0.4.1.1	RDF API	12,773	206	13
JTar 1.2	Compression	9,010	59	115
JGraphX 1.5.0.2	Java Graphing	48,810	229	525

Table 1: Software applications used in the first investigation

Volatilidade

- Métrica que muda de valor frequentemente
  - Determinar mudanças sutis

	JHotDraw (1007)	JTar (115)	XOM (193)	JRDF (13)	JabRef (257)
LSCC	96	99	100	92	99
TCC	86	53	97	46	61
SCOM	79	70	93	92	79
CC	100	98	100	92	99
LCOM5	100	100	100	100	100

	JGraph (525)	ArtOfIllusion (593)	Gantt (750)	All (3453)
LSCC	100	99	96	98
TCC	72	84	71	78
SCOM	89	77	80	81
CC	100	100	99	100
LCOM5	100	100	99	100

Probabilidade de Mudança Positiva

- Melhora de uma métrica influencia nas demais

	JHotDraw	JTar	XOM	JRDF	JabRef
LSCC	↑50 , 46↓	↑50 , 49↓	↑57 , 43↓	↑46 , 46↓	↑54 , 46↓
TCC	↑45 , 41↓	↑30 , 23↓	↑51 , 46↓	↑23 , 23↓	↑34 , 27↓
SCOM	↑38 , 40↓	↑34 , 36↓	↑50 , 44↓	↑46 , 46↓	↑37 , 42↓
CC	↑53 , 47↓	↑52 , 46↓	↑51 , 49↓	↑46 , 46↓	↑54 , 44↓
LCOM5	↑51 , 49↓	↑50 , 50↓	↑48 , 52↓	↑54 , 46↓	↑49 , 50↓

	JGraph	ArtOfIllusion	GanttProject	Average
LSCC	↑51 , 48↓	↑57 , 42↓	↑53 , 43↓	↑53 , 45↓
TCC	↑37 , 35↓	↑52 , 35↓	↑39 , 31↓	↑43 , 35↓
SCOM	↑36 , 53↓	↑44 , 33↓	↑40 , 40↓	↑40 , 41↓
CC	↑61 , 39↓	↑58 , 42↓	↑57 , 42↓	↑56 , 44↓
LCOM5	↑41 , 59↓	↑56 , 43↓	↑50 , 50↓	↑50 , 50↓

Discordâncias

- Comparação de duplas de métricas a cada refatoração
  - Concordância: quando ambos os valores aumentam, diminuem ou não se alteram (45%)
  - Dissonante: um valor aumenta ou diminui, enquanto o outro não se altera (17%)
  - Conflito: um valor aumenta enquanto o outro diminui (38%)

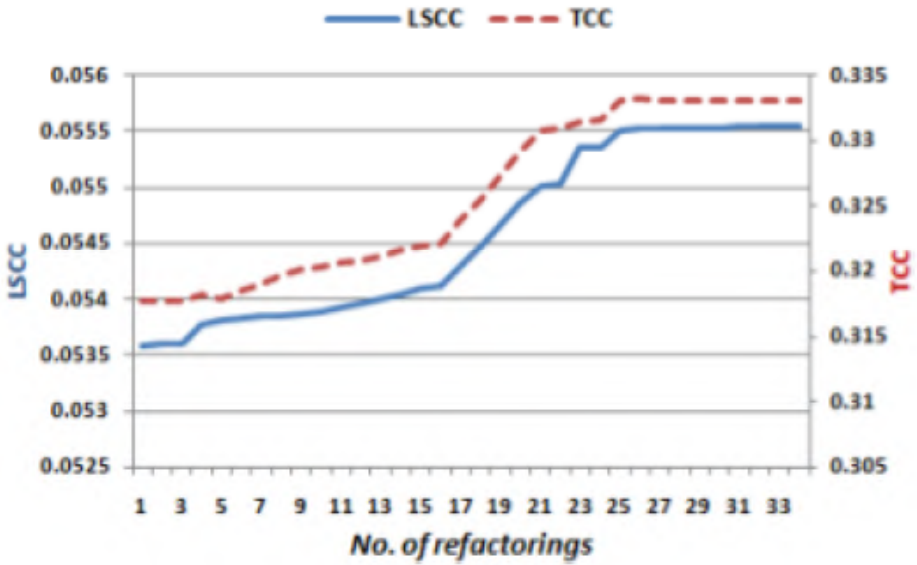
- Refletem diferentes aspectos da coesão
  - Melhorar um sistema com métricas conflitantes não é possível

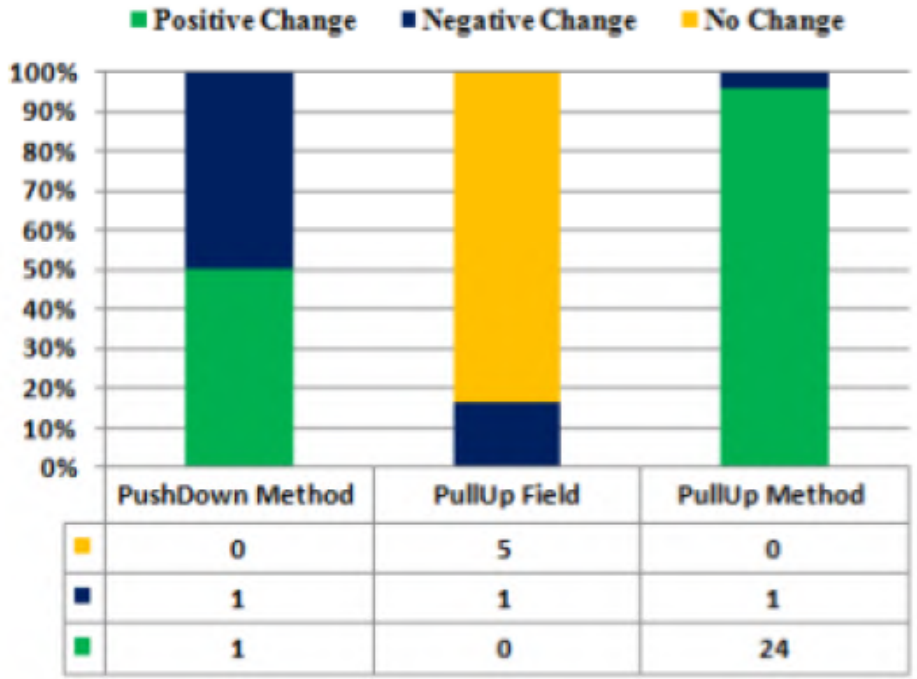
2ª Investigação

- Analisar a relação entre as métricas LSCC e TCC
  - Nova função de *fitness*, utilizando a otimalidade de Pareto
  - Melhorar valor da métrica sem deterioração
  - Apenas JHotDraw

Melhorando LSCC

- Correlação de Spearman igual a 0.8

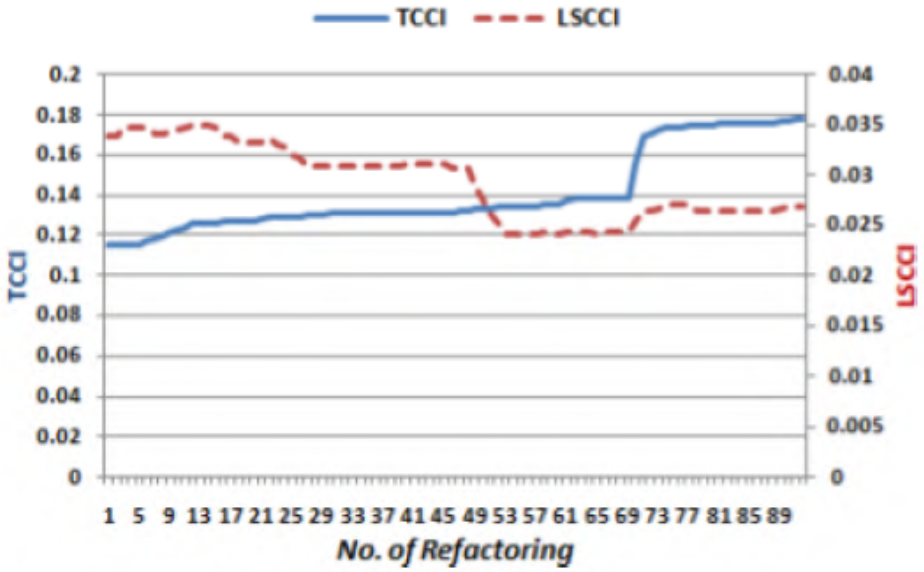


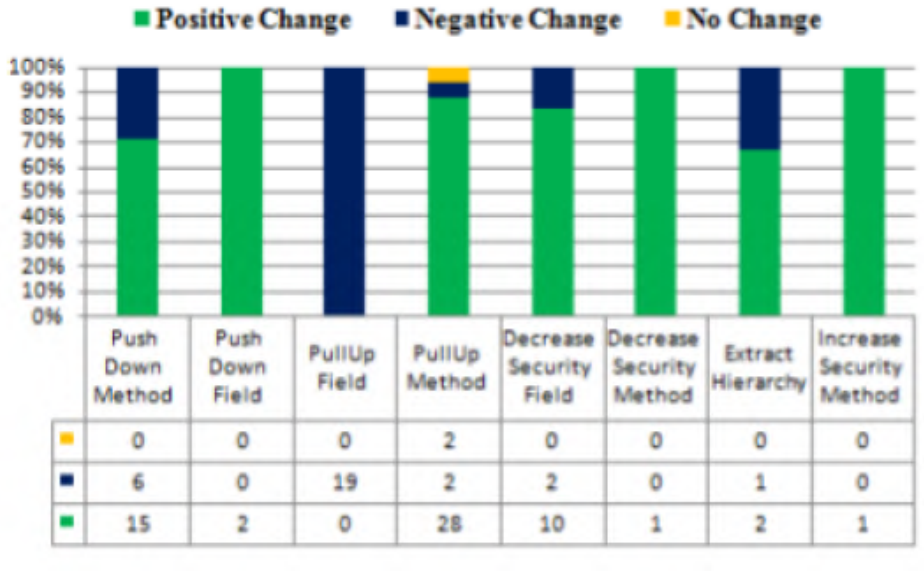


	PushDown Method	PullUp Field	PullUp Method
■	0	5	0
■	1	1	1
■	1	0	24

Melhorando TCC_i

- Correlação de Spearman igual a -0.8





	Push Down Method	Push Down Field	PullUp Field	PullUp Method	Decrease Security Field	Decrease Security Method	Extract Hierarchy	Increase Security Method
No Change	0	0	0	2	0	0	0	0
Negative Change	6	0	19	2	2	0	1	0
Positive Change	15	2	0	28	10	1	2	1

Conclusões

- Nova proposta para avaliação de métricas de software
  - Observar o comportamento
  - Refletem diferentes e conflitantes aspectos da coesão
  - Não há como criar uma métrica unificada

- Trabalhos Futuros
  - Realizar experimentos com métricas de acoplamento
  - Automatizar o processo

Obrigado