

# Do Crosscutting Concerns Cause Defects?

---

Marc Eaddy, Thomas Zimmermann, Student Members, IEEE  
Kaitlin D. Sherwood, Vibhav Garg, Gail C. Murphy, Nachiappan  
Nagappan, Members, IEEE Computer Society  
Alfred V. Aho, Fellow, IEEE

IEEE Transactions on Software Engineering  
July/August 2008

Gustavo Jansen de Souza Santos  
Universidade Federal de Minas Gerais

# Motivação

---

- *Crosscutting Concerns* são uma ameaça à qualidade do código
  - Responsabilidade distribuída em múltiplos módulos
- Qual a relação com o número de defeitos?

# Introdução

---

- Defeitos aparecem em diversos estágios do desenvolvimento
  - Esforço em evitar ou detectar
  - Inspeção de código, análise de programas, testes pré-release
- Indícios que *crosscutting concerns* degradam a qualidade do código
  - Estudos dirigidos a métricas de qualidade interna
  - Dificuldades durante manutenção

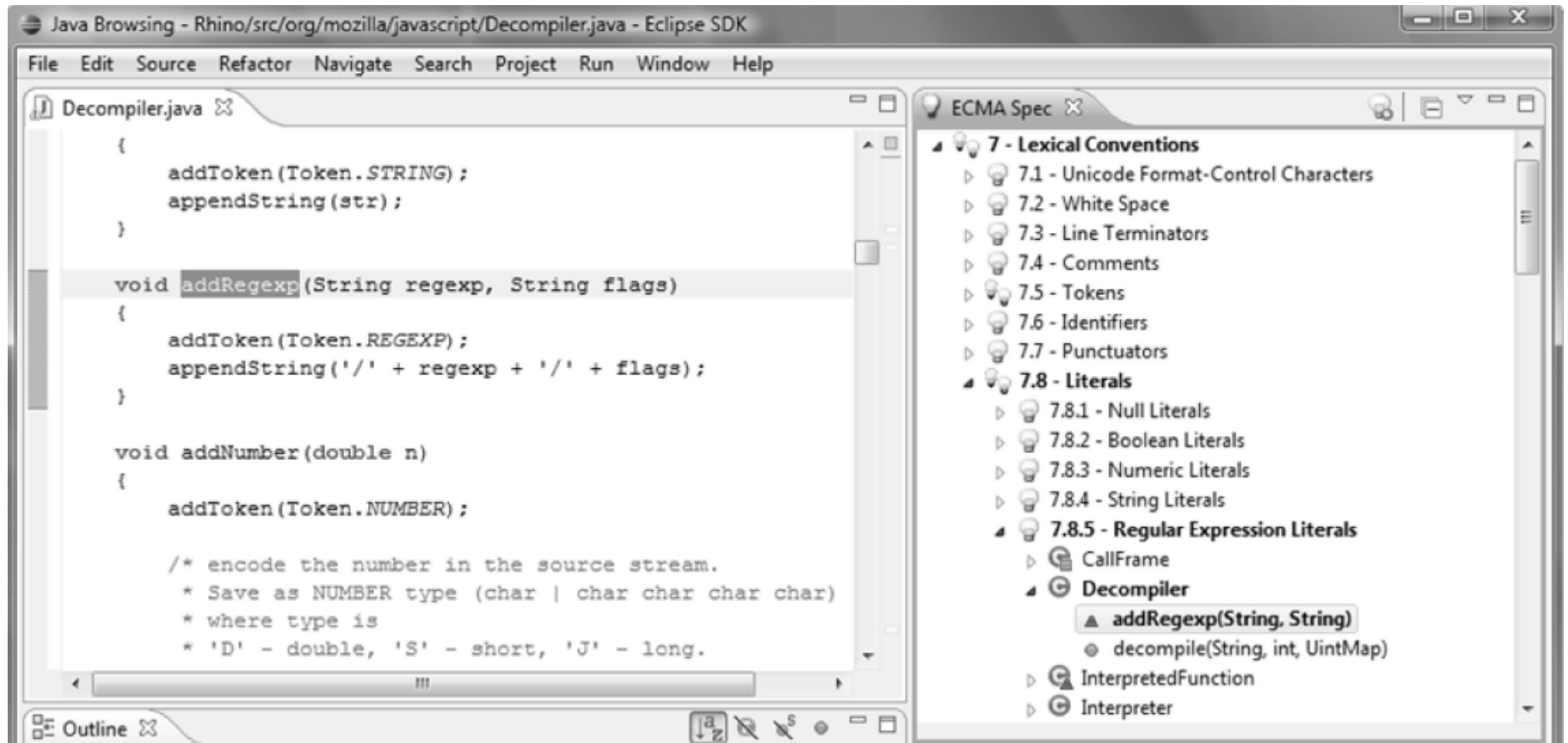
# Definição de *Concerns*

---

- Requisitos funcionais identificáveis no código
  - Extração semi-automática com ferramenta ConcernTagger
  - Mapeamento entre *concerns* e código é feito com engenharia reversa
- Relação de Remoção
  - “um elemento do código é relevante para um *concern* se ele é removido ou alterado quando o *concern* é removido”

# ConcernTagger

---



# Métricas

---

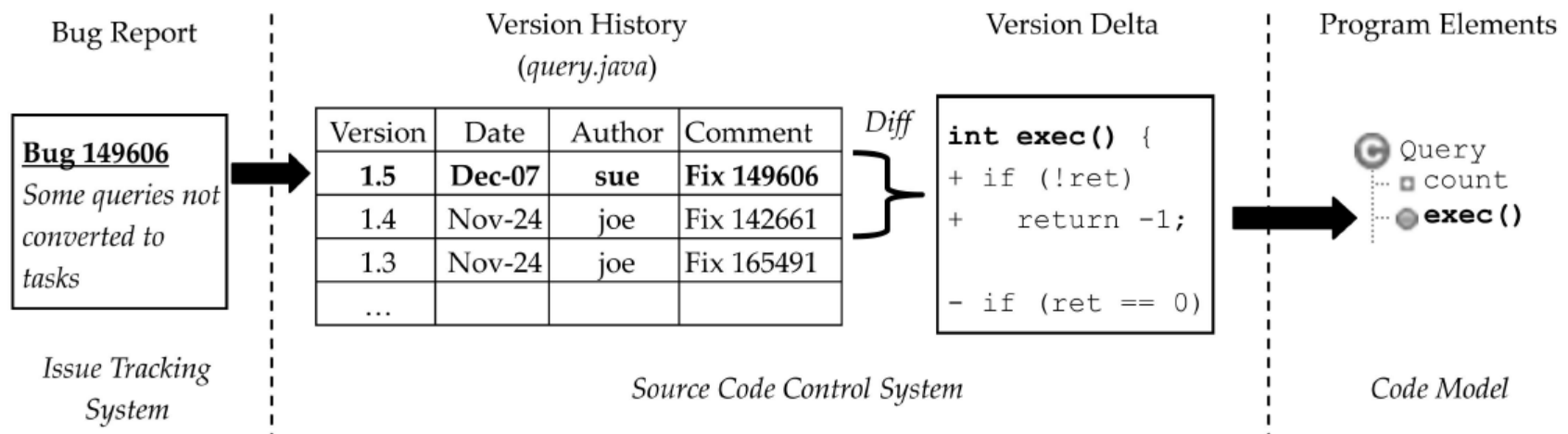
- Avaliam o quão espalhada a implementação do *concern* está
  - Avaliação feita com *concerns* não descendentes

TABLE 1  
Concern Metrics

Bug Count	Number of unique bugs associated with the concern.
Program Element Contribution (CONT)	Number of lines of code in the program element that are associated with the concern. In general, whitespace or comments are excluded; however, for one case study, they were included. Lines outside of class definitions (e.g., package declaration, imports) are not counted by our tool.
Lines of Concern Code (LOCC)	Total number of lines of code that contribute to the implementation of a concern.
Concern Diffusion over Components (CDC)	Number of classes that contribute to the implementation of a concern and other classes and aspects which access them. [28]
Concern Diffusion over Operations (CDO)	Number of methods which contribute to a concern's implementation plus the number of other methods and advice accessing them. [28]
Degree of Scattering across Classes (DOSC)	Degree to which the concern code is distributed across classes. Varies from 0 to 1. When DOSC is 0 all the code is in one class. When DOSC is 1 the code is equally divided among all the classes. [17]
Degree of Scattering across Methods (DOSM)	Degree to which the concern code is distributed across methods. Varies from 0 to 1 similar to DOSC. [17]

# Recuperação de Defeitos

- Uso de plataformas de *issue tracking* e controle de versão
  - Análise do comentário do *commit*
  - Recuperação da alteração
  - BugTagger



# Estudo de Caso

---

- Requisitos
  - Código aberto em Java
  - Ao menos 30 *concerns* identificáveis
  - Dê suporte a alguma ferramenta de *issue tracking*
  - Bugs resolvidos marcados nos *commits*

TABLE 2  
Project Summaries

	<i>Mylyn–Bugzilla</i>	<i>Rhino</i>	<i>iBATIS</i>
<i>Application Domain</i>	Development Tools	Compilers	Databases
<i>Project Size (KLOCS)</i>	Small (~14)	Medium (~32)	Small (~13)
<i>Issue Tracking System</i>	Bugzilla	Bugzilla	Jira
<i>Source Code Control System</i>	CVS	CVS	Subversion
<i>Concern Domain</i>	Requirements	ECMAScript Specification	Requirements
<i>Bug Assignment Technique</i>	By-Hand	BugTagger	BugTagger



# Coleta

---

TABLE 3  
Size and Assignment Coverage Statistics

	<i>Mylyn–Bugzilla</i>			<i>Rhino</i>			<i>iBATIS</i>		
	All	Mapped <sup>a</sup>	%	All	Mapped	%	All	Mapped	%
<i>Classes</i>	56	44	79	138	80	57	212	207	97
<i>Methods</i>	427	253	59	1870	1415	75	1844	1807	97
<i>Fields</i>	457	230	50	1339	962	71	536	529	98
<i>Lines<sup>b</sup></i>	13649	5914	43	32134	28308	88	13314	13144	98
<i>Concerns<sup>c</sup></i>	28	28	100	480	417	86	183	173	94
<i>Bugs<sup>d</sup></i>	110	101	92	241	160	66	87	47	53

# Experimentos

TABLE 4  
Spearman Correlation Coefficients (a) Mylyn-Bugzilla, (b) Rhino, (c) iBATIS

	<i>DOSM</i>	<i>CDC</i>	<i>CDO</i>	<i>LOCC</i>	<b><i>Bugs</i></b>
<i>DOSC</i>	.64	.84	.57	.38	<b>.39</b>
<i>DOSM</i>	—	.77	.91	.63	<b>.50</b>
<i>CDC</i>	—	—	.78	.65	<b>.57</b>
<i>CDO</i>	—	—	—	.71	<b>.61</b>
<i>LOCC</i>	—	—	—	—	<b>.77</b>

(a)

	<i>DOSM</i>	<i>CDC</i>	<i>CDO</i>	<i>LOCC</i>	<b><i>Bugs</i></b>
<i>DOSC</i>	.62	.96	.74	.60	<b>.67</b>
<i>DOSM</i>	—	.63	.88	.68	<b>.66</b>
<i>CDC</i>	—	—	.80	.67	<b>.73</b>
<i>CDO</i>	—	—	—	.80	<b>.77</b>
<i>LOCC</i>	—	—	—	—	<b>.90</b>

(b)

	<i>DOSM</i>	<i>CDC</i>	<i>CDO</i>	<i>LOCC</i>	<b><i>Bugs</i></b>
<i>DOSC</i>	.67	.90	.73	.43	<b>.46</b>
<i>DOSM</i>	—	.67	.90	.64	<b>.29</b>
<i>CDC</i>	—	—	.78	.55	<b>.58</b>
<i>CDO</i>	—	—	—	.77	<b>.44</b>
<i>LOCC</i>	—	—	—	—	<b>.53</b>

# Ameaças à Validação

---

- Indicação de *concerns* manual
- Indicação de bugs com dados do repositório
- Espalhamento não é a única fonte de defeitos
- *Dataset* não representativo

# Conclusões

---

- Existe uma correlação entre métricas e número de bugs
  - Correlação não implica em causalidade
  - Através de ferramental estatístico, foram mostrados indícios de causalidade
- Causas do espalhamento
  - Redução implica na geração de outros tipos de defeitos?

# Obrigado

---