



Extending a Taxonomy of Bad Code Smells with Metrics

R Marticorena, C Lopez, The University of Burgos
Y Crespo, University of Valladolid

Presented by D Rodríguez, The University of Reading

Outline

- Introduction
- Related Works
- Extended Taxonomy of Bad Smells
 - New Features to Apply
 - Metric Feature
- Usability Example of the Extended Taxonomy Mixing Metrics
- Conclusions and Future Works

Introduction

- Refactoring
- Bad Smells, i.e., collect design or code flaws
- but ...
 - ...their detection is based on certain programmer intuition
 - current taxonomies do not settle relations between bad smell groups and used metrics
 - Until now, grouping bad smells depending of certain similarity, more or less subjective

Related Works

- Fowler's Refactoring book.
 - "no set of metrics rivals human intuition"
 - but with large amounts of code, intuition can be difficult
- Taxonomies
 - "within classes" and "between classes".
 - <http://wiki.java.net/bin/view/People/SmellsToRefactorings>
 - Mantyla -> based on similarity criteria:
 - Bloaters, Object-Oriented Abusers, Change Preventers, Dispensables and Couplers.
 - Also, metrics for detecting bad smells
- Muñoz -> queries on logic predicates
- Marinescu -> queries on a database

Extended Taxonomy of Bad Smells

- Extending and Crossing Mantyla taxonomy with metrics:
 - Granularity - size of the component. In OO:
 - following levels: system class and method.
 - Intra vs. Inter-relations (Intra) the bad smell could be observed from the individual observation (intra) of the component
 - Inheritance (IH) information about inheritance hierarchy is needed to suggest the bad smells.
 - Access Modifiers (Acc) access level among the components.
- Crossing
 - With software metrics and specially OO metrics

Example

Bad Smell	Group	Gran.	Intra	IH	Acc
Data Clumps	Bloater	Class	N	N	N
Large Class	Bloater	Class	Y	N	N
Long Method	Bloaters	Method	Y	N	N
Long Parameter List	Bloaters	Method	Y	N	N
Primitive Obsession	Bloaters	Method	N	N	N
Alternative Classes with Diffe-	...	Method	N	N	N

Metric	Desc.	Gran.	Intra	Inh	Acc
CLOC	Comment Lines of Code	Class	Y	N	N
V(G)	McCabe's Cyclomatic Complexity	Method	Y	N	N
NP	Number of Parameters	Method	Y	N	N
DIT	Depth of Inheritance Tree	Class	N	Y	N
RFC	Response for Class	Class	N	N	N
...

Usability Example of the Extended Taxonomy

Mixing Metrics

- which tool can help us to suggest the presence of a greater number of bad code smells?
 - Metrics-1.3.6, RefactorIt 2.5, DMS

Group	BS Number	Not covered (Eclipse)	Not covered (RefactorIt)	Not covered (DMS)
Bloaters	5	2	0	2
O-O Abusers	4	1	1	2
Change Preventers	3	2	0	3
Dispensables	4	1	1	2
Couplers	4	4	2	4
Not Defined	2	1	1	1

- RefactorIt is the most adequate to detect bad code smells

Conclusions and Future Work

- Presented an extension to current taxonomies
- Metrics are useful for detecting bad code smells
- Decision system that can be improved with heuristics in future works.