

8

Referências

ATKINSON, C.; KÜHNE, T. Model-Driven Development: A Metamodeling Foundation. **IEEE Software**. , 20 (5), 36-41, 2003

BAEZA, R.; RIBEIRO, B. **Modern Information Retrieval**. Addison Wesley Longman, 1999.

BANSIYA, J. **A Hierarchical Model For Quality Assessment Of Object-Oriented Designs** PhD thesis, University. of Alabama in Huntsville, 1997.

BASILI, V; ROMBACH, H. The TAME Project: Towards Improvement-Oriented Software Environments. **IEEE Transactions on Software Engineering** 14, pp. 758-773, 1988.

BASILI, V.; WEISS, D. A Methodology for Collecting Valid Software Engineering Data. **IEEE Transactions on Software Engineering**, 10, pp.728-738, 1984.

BELL, D.; MORREY, I.; PUGH, J. **Software Engineering - A Programming Approach**. Prentice-Hall, NJ, 1987.

BORLAND TOGETHERSOFT WEBSITE. URL: <http://www.togethersoft.com>. September 2008.

BRIAND, L.; BUNSE, C.; DALY, J. An experimental evaluation of quality guidelines on the maintainability of object-oriented design guidelines. **Technical Report** ISERN-97-02. Fraunhofer Institute (IESE), Kaiserslautern, Germany, 1997.

BRIAND, L.C.; DALY, J.W.; WUST, J. **A Unified Framework for Cohesion Measurement, Empirical Software Engineering**, v. 3, Dordrecht: Kluwer, no.1, pp. 65-115, 1998.

CHIDAMBER, S.; KEMERER, C. A Metrics Suite for Object Oriented Design. **IEEE Transactions on Software Engineering**. June 1994, 20 (6), pp. 476-493.

COAD, P.; YOURDON, E. **Object-Oriented Design**. Prentice Hall, London, 2 edition, 1991.

EMDEN, E.; MOONEN, L. Java Quality Assurance by Detecting Code Smells. **In WCRE 2002 proceedings**, 2002.

FENTON, N.; PFLEEGER, S. **Software Metrics: A Rigorous and Practical Approach**. 2.ed. London: PWS, 1997.

FOWLER, M.; BECK, K.; BRANT, J.; OPDYKE, W.; ROBERTS, D. **Refactoring: Improving the Design of Existing Code**. Addison-Wesley, 1999.

GAMMA E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns: Elements of Reusable Object-oriented Software**. Addison-Wesley, Reading, 1995.

GENERO, M.; PIATTINI, M. Empirical validation of measures for class diagram structural complexity through controlled experiments. **In Proceedings of 5th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2001)**, Budapest, Hungary, June 2001.

GENERO, M.; PIATTINI, M., CALERO, C. Empirical validation of class diagram metrics. **In Proceedings of 2002 International Symposium on Empirical Software Engineering**, Nara, Japan, pp.195–203, October 2002.

IBM Rational Rose website. URL: <http://www-01.ibm.com/software/rational>. February 2008.

INCODE WEBSITE. URL: <http://loose.upt.ro/incode>. August 2008.

JAVA. Java Reference Documentation. URL: <http://java.sun.com/reference/docs/index.html>. February 2008.

JHOTDRAW. JHotdraw website. URL: <http://www.jhotdraw.org>. May 2008.

KITCHENHAM, B.; PFLEEGER, S.; FENTON, N. Towards a Framework for Software Measurement Validation. **IEEE Transactions on Software Engineering**. 21 (12), December 1995.

LAKOS, J. **Large-Scale C++ Software Design**. Addison-Wesley, 1996.

LANZA, M.; MARINESCU, R. **Object-Oriented Metrics In Practice: Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems**. Springer-Verlag Berlin And Heidelberg GmbH & Co. Kg (Germany), 2006.

LORENZ, M.; KIDD, J. **Object-Oriented Software Metrics**. Prentice-Hall Object-Oriented Series, Englewood Cliffs, NY, 1994.

MACIA, I.; SANT'ANNA, C.; STAA, A. “Detectando Problemas de Design em Diagramas de Classes: Um Estudo Experimental.” **V Experimental Software Engineering Latin American Workshop (ESELAW'2008)**, Salvador, Brasil, November 2008.

MCCABE, T.J. A Complexity Measure. **IEEE Transactions on Software Engineering**, 2(4):308–320, December 1976.

MANSO, M.; GENERO, M.; PIATTINI, M. No-redundant metrics for UML class diagram structural complexity. **Lecture Notes on Computer Science**, 2681, pp.127-142. 2003

MARINESCU, R. Detecting Design Flaws via Metrics in Object-Oriented Systems. **Proceedings of TOOLS USA 2001**, pages 103–116. IEEE Computer Society, 2001.

MARINESCU, R. **Measurement and Quality in Object-Oriented Design**. PhD thesis, Politehnica University of Timisoara, 2002.

MARINESCU, R. Detection Strategies: Metric-Based Rules for Detecting Design Flaws. **Proceedings of the 20nd IEEE International Conference on Software Maintenance (ICSM'04)**, pp. 350-359, 2004.

MARTIN, R.C. Stability. **C++ Report**. February 1997.

MARTIN, R.C.; NEWKIRK, J. W.; KOSS, R.S. **Agile Software Development, Principles, Patterns, and Practices**. Prentice Hall, 2002.

MetricView website. URL: <http://www.win.tue.nl/empanada/metricview>. April 2008.

MEYER, B. **Object-Oriented Software Construction**. 2.ed. Prentice Hall, New York, NY, 1997.

MOF, **MOF MetaObject Facility**, Version 1.4, January 2007, <http://www.omg.org/mof/>, December 2007.

MUNRO, M.J. Product Metrics for Automatic Identification of Bad Smell Design Problems in Java Source-Code. Software Metrics, 2005. **Proceedings of the 11th IEEE International Symposium**, pp. 19-22 September 2005.

NETBEANS, **Netbeans IDE**, Version 6.5, website URL: <http://www.netbeans.org>. December 2007.

OMG. **OMG Unified Modeling Language Specification**, Version 1.5, **OMG Document formal/03-03-01**, March 2003, <http://www.omg.org/technology/documents/formal/uml.htm>. December 2003.

PERRY, D.; PORTER, A.; VOTTA, L. **Empirical Studies of Software Engineering: A Roadmap**. Future of Software Engineering. Anthony Finkelstein Ed., ACM, pp. 345-355, 2000.

SAHRAOUI, H.; BOUKADOUM, M.; LOUNIS, H. **Building Quality Estimation models with Fuzzy Threshold Values**. L'Objet, 17(4), pp. 535-554, 2001.

SAX, **SAX Simple API for XML**, Version 2.0.2, April 2004, website URL: <http://www.saxproject.org/>

SCHMIDT, D.C. Model Driven Engineering. **IEEE Software** 39 (2). February, 2006.

SDMetrics website. URL: <http://www.sdmetrics.com>. May 2008.

SOMMERVILLE, I. **Software Engineering**. 6.ed. Addison-Wesley, Harlow, England, 2001.

STAA, A.v. **Programação Modular**. Rio de Janeiro, RJ: Elsevier/Campus. 2000

TONG, Y.; FANGJUN, W. Empirical analysis of entropy distance metric for UML class diagrams. **ACM SIGSOFT Software Engineering Notes** 29(5): 1-6 (2004)

Object Management Group. UML Revision Task Force, **OMG Unified Modeling Language Specification**, Version. 1.3, document ad/99-06-08. 1999, <http://www.uml.org>. November 2007

VAN SOLINGER, R; BERGHOUT, E. **The Goal/Question/Metric Method: A practical guide for quality improvement of software development**. McGraw-Hill. 1999

WIRTH, N. A Plea for Lean Software. **IEEE Computer**, 28(2), February 1995.

WOHLIN, C.; RUNESON, P.; HOST, M.; OHLSON, M.; REGNELL, B.; WESSLEN, A. **Experimentation in Software Engineering: An Introduction**, Kluwer Academic Publishers, 2000.

XMI XML Metadata Interchange, <http://xml.coverpages.org/xmiml>. January 2008

YI, T.; WU, F.; GAN, C. A comparison of Metrics for UML Class Diagrams. ACM SIGSOFT **Software Engineering Notes**. September 2004.

YOURDON, E.; CONSTANTINE, L. L. **Structured Design**. Prentice Hall, Englewood Cliffs, NJ, 1979.

ZHOU, Y., et al. (2003): Measuring structure complexity of UML class diagrams. **Journal of Electronics (China)**, 20(3), 2003, pp.227-231.

Anexo A - Document Type Definition do arquivo XML utilizado por QCDTool

Este anexo apresenta a estrutura que descreve os arquivos XML criados como parte do mecanismo de análise dos diagramas UML apresentado na Seção 5.3.1.

```
<?xml version="1.0" encoding="ISO-8859-15"?>

<!ELEMENT metamodel ( package* | class* | generalization* | dependencyclass* | association* |
primitive*)* >

<!ENTITY % attrs "id NMTOKEN #REQUIRED name CDATA #IMPLIED" >

<!ELEMENT package ( package* | class* | generalization* | dependencyclass* | association* |
primitive*)* >
    <!ATTLIST package %attrs; >

<!ELEMENT class ( attribute* | operation* | generalization* )* >
    <!ATTLIST class %attrs; >
    <!ATTLIST class version NMTOKEN #IMPLIED >
    <!ATTLIST class isRoot (true | false) "false" >
    <!ATTLIST class isLeaf (true | false) "true" >
    <!ATTLIST class isAbstract (true | false) "false" >
    <!ATTLIST class isActive (true | false) "true" >
    <!ATTLIST class isClass (true | false) "true" >
    <!ATTLIST class visibility (public | private | package | protected) "public" >

<!ELEMENT attribute (#PCDATA)>
    <!ATTLIST attribute id NMTOKEN #IMPLIED >
    <!ATTLIST attribute name CDATA #IMPLIED >
    <!ATTLIST attribute type NMTOKEN #REQUIRED >
    <!ATTLIST attribute genericitytype CDATA #IMPLIED >
    <!ATTLIST attribute visibility (public | private | package | protected) "public" >
    <!ATTLIST attribute isStatic (true | false) "false" >

<!ELEMENT operation (parameter)* >
    <!ATTLIST operation id NMTOKEN #IMPLIED >
    <!ATTLIST operation name CDATA #IMPLIED >
    <!ATTLIST operation %attrs; >
    <!ATTLIST operation visibility (public | private | package | protected) "public" >

<!ELEMENT parameter (#PCDATA)>
    <!ATTLIST parameter id NMTOKEN #IMPLIED >
    <!ATTLIST parameter name CDATA #IMPLIED >
    <!ATTLIST parameter kind CDATA #IMPLIED >
    <!ATTLIST parameter type NMTOKEN #IMPLIED >
    <!ATTLIST parameter genericitytype NMTOKEN #IMPLIED >
    <!ATTLIST parameter visibility (public | private | package | protected) "public" >

<!ELEMENT generalization (#PCDATA)>
    <!ATTLIST generalization %attrs; >
```

```

<!--ATTLIST generalization child.id NMTOKEN #REQUIRED -->
<!--ATTLIST generalization parent.id NMTOKEN #REQUIRED -->
<!--ELEMENT dependencyclass (#PCDATA)-->
  <!--ATTLIST dependencyclass %attrs; -->
  <!--ATTLIST dependencyclass client_id NMTOKEN #REQUIRED -->
  <!--ATTLIST dependencyclass supplier_id NMTOKEN #REQUIRED -->
<!--ELEMENT association (associationend)+ -->
  <!--ATTLIST association %attrs; -->
  <!--ATTLIST association visibility (public | private | package | protected) "public" -->
<!--ELEMENT associationend (associationmultiplicity | associationparticipant)+ -->
  <!--ATTLIST associationend id NMTOKEN #IMPLIED -->
  <!--ATTLIST associationend name CDATA #IMPLIED -->
  <!--ATTLIST associationend type CDATA #REQUIRED -->
  <!--ATTLIST associationend isNavigable CDATA #IMPLIED -->
  <!--ATTLIST associationend visibility (public | private | package | protected) "public" -->
<!--ELEMENT associationmultiplicity (#PCDATA)-->
  <!--ATTLIST associationmultiplicity id NMTOKEN #IMPLIED -->
  <!--ATTLIST associationmultiplicity lower CDATA #IMPLIED -->
  <!--ATTLIST associationmultiplicity upper CDATA #IMPLIED-->
<!--ELEMENT associationparticipant (#PCDATA)-->
  <!--ATTLIST associationparticipant id NMTOKEN #REQUIRED -->
<!--ELEMENT primitive (#PCDATA)-->
  <!--ATTLIST primitive %attrs; -->

```

Anexo B - Documento gerado por QCDDTool para Armazenar os Resultados das Estratégias de Detecção

Este anexo apresenta um exemplo de arquivo gerado por QCDDTool para armazenar os resultados da aplicação das estratégias de detecção (Seção 5.3.6) no sistema JHotdraw versão 5.2.

JHotdraw 5.2

```
-- LongParameterList (NOP) --
CH.ifa.draw.contrib.PolygonFigure.distanceFromLine (6)
CH.ifa.draw.figures.ImageFigure.imageUpdate (6)
CH.ifa.draw.util.Geom.intersect (8)
CH.ifa.draw.util.Geom-lineContainsPoint (6)

-- DataClass (NOPA, NOAM, WOC, WMC) --
Clipboard (0,3,0.2,1)
Filler (0,3,0,2)

-- GodClass (ATFD, WMC, CAM) --
CompositeFigure (10,32,0.3)
DrawApplication (13,53,0.3)
StandardDrawingView (10,60,0.3)

-- ShotgunSurgery (CA, CM, CC) --
Command (1,12,12)
StorableInput (0,24,26)
StorableOutput (0,24,27)

-- GodPackage (PS, NOCC, NOCP) --
standard (55,20,4)
```

Anexo C - Resultados da Aplicação das Estratégias de Detecção no Primeiro Estudo Experimental

Este anexo apresenta todos os dados obtidos com a aplicação das estratégias de detecção *Long Parameter List*, *Data Class*, *God Class*, *Shotgun Surgery*, *Misplaced Class* e *God Package* tanto no código como no modelo. Estas estratégias foram aplicadas nos sistemas apresentados na Seção 6.1.1. Estes resultados foram obtidos utilizando as ferramentas *QCDTool*, *InCode* e *Together* 6.0. Note-se que cada métrica ou propriedade tem associado o valor correspondente a seu cálculo no modelo e no código fonte.

A tabela abaixo mostra os métodos detectados pela estratégia de detecção *Long Parameter List* tanto no modelo como no código em cada sistema avaliado.

Tabela 16 – Resultados da estratégia de detecção *Long Parameter List*

Sistema S6
CH.ifa.draw.contrib.PolygonFigure.distanceFromLine
CH.ifa.draw.figures.ImageFigure.imageUpdate
CH.ifa.draw.util.Geom.intersect
CH.ifa.draw.util.Geom-lineContainsPoint
Sistema S7
org.jhotdraw.contrib.zoom.ScalingGraphics.copyArea
org.jhotdraw.contrib.zoom.ScalingGraphics.drawArc
org.jhotdraw.contrib.zoom.ScalingGraphics.drawImage
org.jhotdraw.contrib.zoom.ScalingGraphics.drawRoundRect
org.jhotdraw.contrib.zoom.ScalingGraphics.fillArc
org.jhotdraw.figures.ImageFigure.imageUpdate
org.jhotdraw.contrib.zoom.ScalingGraphics.fillRoundRect
org.jhotdraw.util.Geom.distanceFromLine
org.jhotdraw.util.Geom.distanceFromLine2D
org.jhotdraw.util.Geom.intersect
org.jhotdraw.util.Geom.lineContainsPoint
Sistema S8
net.n3.nanoxml.ValidatorPlugin.validationError
net.n3.nanoxml.XMLUtil.validationError
org.jhotdraw.draw.TextAreaFigure.drawParagraph
org.jhotdraw.draw.action.ToolBarButtonFactory.addColorButtonTo
org.jhotdraw.geom.BezierPath-curveTo
org.jhotdraw.geom.BezierPath-arcTo

org.jhotdraw.geom.DoubleStroke.computeThickLine
org.jhotdraw.geom.Bezier.computeMaxError
org.jhotdraw.geom.Bezier.fitCubic
org.jhotdraw.geom.Bezier.generateBezier
org.jhotdraw.geom.Geom.distanceFromLine
org.jhotdraw.geom.Geom.intersect
org.jhotdraw.geom.Geom.lineContainsPoint
org.jhotdraw.gui.JSheet.showConfirmSheet
org.jhotdraw.gui.JSheet.showInputSheet
org.jhotdraw.gui.JSheet.showOptionSheet
Sistema S9
net.n3.nanoxml.ValidatorPlugin.validationError
net.n3.nanoxml.XMLUtil.validationError
org.apache.batik.ext.awt.RadialGradientPaintContext.antiAliasFillRaster
org.apache.batik.ext.awt.RadialGradientPaintContext.cyclicCircularGradientFillRaster
org.apache.batik.ext.awt.RadialGradientPaintContext.fillRaster
org.apache.batik.ext.awt.RadialGradientPaintContext.fixedPointSimplestCaseNonCyclicFillRaster
org.apache.batik.ext.awt.MultipleGradientPaintContext.fillRaster
org.apache.batik.ext.awt.MultipleGradientPaintContext.getAntiAlias
org.apache.batik.ext.awt.MultipleGradientPaintContext.getRaster
org.apache.batik.ext.awt.LinearGradientPaintContext.fillHardNoCycle
org.apache.batik.ext.awt.LinearGradientPaintContext.fillRaster
org.apache.batik.ext.awt.LinearGradientPaintContext.fillSimpleNoCycle
org.apache.batik.ext.awt.LinearGradientPaintContext.fillSimpleReflect
org.apache.batik.ext.awt.LinearGradientPaintContext.fillSimpleRepeat
org.jhotdraw.draw.TextAreaFigure.drawParagraph
org.jhotdraw.draw.action.ImageBevelBorder.paintBorder
org.jhotdraw.draw.action.ButtonFactory.createEditorColorButton
org.jhotdraw.draw.action.ButtonFactory.createSelectionColorButton
org.jhotdraw.geom.Bezier.computeMaxError
org.jhotdraw.geom.Bezier.fitCubic
org.jhotdraw.geom.Bezier.generateBezier
org.jhotdraw.geom.Geom.distanceFromLine
org.jhotdraw.geom.Geom.intersect
org.jhotdraw.geom.Geom.lineContainsPoint
org.jhotdraw.geom.BezierPath.arcTo
org.jhotdraw.geom.BezierPath.curveTo
org.jhotdraw.geom.DoubleStroke.computeThickLine
org.jhotdraw.gui.JSheet.showConfirmSheet
org.jhotdraw.gui.JSheet.showInputSheet
org.jhotdraw.gui.JSheet.showOptionSheet

A tabela abaixo mostra as classes classificadas como *Data Class* tanto pela estratégia de detecção para modelo como pela correspondente para código em cada um dos sistemas. Além disso, para cada uma destas classes são mostrados os dados obtidos com a aplicação das métricas envolvidas nestas estratégias de detecção – NOPA, NOAM, WOC e WMC – tanto para modelo (MOD) como para código (COD).

Tabela 17 – Resultados das estratégias de detecção *Data Class*

Sistema S1								
Classe	NOPA		NOAM		WOC		WMC	
	MOD	COD	MOD	COD	MOD	COD	MOD	COD
Alert	0	0	14	14	0	0	1,4	16
AlertDTO	0	0	13	13	0,0	0,05	0,3	17
AlertForm	0	0	12	12	0,07	0,07	1,4	17
BeanLocator	0	0	5	2	0	0,3	1,1	13
Incident	0	0	16	16	0	0	1,4	17
IncidentDataGenerator Command	0	0	6	1	0	0,2	2,2	20
PeriodicReport	0	0	12	12	0	0	1,4	14
PeriodicReportDTO	0	0	12	12	0	0,05	1,4	15
PeriodicReportForm	0	0	12	12	0,07	0	1,3	15
ReportFilter	0	0	4	4	0	0	1,2	5
ReportForm	0	0	7	6	0	0	1,2	10
ReportValues	0	0	16	16	0	0	2,1	9
Sistema S2								
Classe	NOPA		NOAM		WOC		WMC	
	MOD	COD	MOD	COD	MOD	COD	MOD	COD
DataModuleFactory	6	6	6	6	0,1	0,1	1	5
EstadisticReport	2	2	2	0	0	0	0,66	23
ForumMessage	0	0	16	16	0,05	0,05	1,5	11
Metadata	0	0	8	8	0	0,1	1,5	10
SystemConfiguration	0	0	7	7	0	0	0,8	12
Sistema S3								
Classe	NOPA		NOAM		WOC		WMC	
	MOD	COD	MOD	COD	MOD	COD	MOD	COD
CargoFuncionario	0	0	4	4	0,1	0,1	1,5	6
ClasseTitulo	0	0	8	8	0,1	0,3	1,6	10
ClienteDependente	0	0	4	4	0	0,06	1,7	30
Contatos	0	0	18	18	0,05	0,05	1,75	20
CopiaTitulo	1	1	8	8	0,1	0,1	1,5	13
Fornecedor	0	0	8	8	0,1	0,05	1,5	10
Funcionario	0	0	6	6	0,2	0,2	1,8	10
GeneroTitulo	0	0	4	4	0,2	0,2	1,5	6
GrauParentesco	0	0	3	2	0,1	0,5	0,5	2
Locacao	5	5	20	17	0,16	0,24	1,5	35
LocacaoState	1	1	5	4	0	0,1	1,8	7
Pessoa	1	1	3	1	0,25	0,3	1,1	6
Telefone	0	0	16	16	0	0	1,5	16
Titulo	0	0	24	24	0,1	0,1	1,6	30
TituloMidia	0	0	6	6	0,14	0,14	1,5	8
Usuario	0	0	16	16	0,1	0,1	1,4	23
Sistema S4								
Classe	NOPA		NOAM		WOC		WMC	
	MOD	COD	MOD	COD	MOD	COD	MOD	COD
Associacao	0	0	8	8	0	0	1,7	11
Atributo	0	0	8	8	0	0	1,4	10
CasoDeUso	0	0	4	4	0	0	1,33	6
Classe	0	0	10	10	0,1	0,1	1,5	22
DadosModelo	0	0	6	6	0	0	1,2	7
ElementoXMI	1	1	8	8	0	0	1,4	10
Formulario	0	0	5	3	0,23	0,6	1,7	27
Heranca	0	0	4	4	0	0	1,3	6

Mensagem	0	0	6	6	0	0	1,5	6
agentes.Mensagem	0	0	6	6	0	0	1,37	8
Objeto	0	0	4	4	0	0	1,3	6
Operacao	0	0	4	4	0	0	1,3	8
PanelAutoria	0	0	3	1	0	0,5	1,6	14
Parametro	0	0	4	4	0	0	1,6	6
ResultadoAnalise	0	0	10	10	0,3	0,3	1,5	10
Sistema S5								
Classe	NOPA		NOAM		WOC		WMC	
	MOD	COD	MOD	COD	MOD	COD	MOD	COD
Attribute	0	0	4	4	0	0	1,8	9
BinaryOperator	3	3	1	0	0	0,1	2	4
JErrorDialog	4	4	3	1	0	0,36	1,9	7
JResultFrame	6	6	0	0	0,2	0,2	0,6	10
Metric	3	3	8	8	0,08	0,08	1,5	10
XMIAttribute	0	0	9	6	0,1	0,41	1,3	25
XMIClass	0	0	18	10	0,25	0,8	1,5	52
XMIDependency	0	0	4	4	0,2	0,2	1,4	5
XMIElement	0	0	8	8	0	0	1,5	8
XMIEndAssociation	0	0	10	10	0,2	0,2	1,4	14
XMIGeneralization	0	0	4	4	0	0,2	1,5	4
XMIMethod	0	0	7	4	0,2	0,4	1,3	12
XMIPackage	0	0	9	9	0,12	0,08	1,4	18
XMIParameter	0	0	6	6	0,1	0,09	1,4	7
Sistema S6								
Classe	NOPA		NOAM		WOC		WMC	
	MOD	COD	MOD	COD	MOD	COD	MOD	COD
Clipboard	0	0	3	3	0,2	0,2	1	4
Filler	0	0	3	0	0	1	2	6
Sistema S7								
Classe	NOPA		NOAM		WOC		WMC	
	MOD	COD	MOD	COD	MOD	COD	MOD	COD
Clipboard	0	0	3	3	0,2	0,2	1	4
CommandCheckBoxMenu	1	1	2	2	0,5	0,5	2,5	6
DoubleBufferImage	0	0	7	1	0,13	0,88	2	9
FigureChangeEvent	0	0	3	2	0	0,25	2	6
GraphNode	5	5	1	0	0,27	1	1,5	3
Sistema S8								
Classe	NOPA		NOAM		WOC		WMC	
	MOD	COD	MOD	COD	MOD	COD	MOD	COD
AttributeKeys	0	0	7	7	0	0	2	22
DefaultDrawingViewBeanInfo	0	0	7	2	0	0,15	1,1	21
FigureEvent	0	0	5	5	0,2	0,2	1,9	7
JPopupButtonBeanInfo	0	0	7	2	0	0,7	1,1	21
SheetEvent	0	0	6	6	0,31	0,31	2,1	9
StackedReader	4	4	1	0	0,26	1	0	6
XMLAttribute	0	0	6	6	0,31	0,31	1,8	6
XMLValidationException	0	0	3	3	0,3	0,3	2,4	8
Sistema S9								
Classe	NOPA		NOAM		WOC		WMC	
	MOD	COD	MOD	COD	MOD	COD	MOD	COD

AttributeKeys	0	0	8	8	0	0	2	2
ColorIcon	0	0	5	4	0,15	0,2	2,4	8
DefaultDrawingViewBeanInfo	0	0	7	2	0	0,4	1	21
Double	4	4	5	5	0	0	2	9
FigureEvent	0	0	5	4	0	0,2	1,8	7
FigureSelectionEvent	0	0	4	4	0	0	1,7	3
Float	4	4	6	6	0	0	2,3	7
MultipleGradientPaint	6	6	6	2	0	0,02	1,7	26
MultipleGradientPaintContext	22	22	3	0	0,1	1	3,6	44
SheetEvent	0	0	6	4	0	0,3	2,1	6
StackedReader	4	4	2	0	0,08	0,3	0	6
ToolEvent	0	0	4	4	0	0	1	6
XMLAttribute	0	0	6	6	0,06	0,06	1,8	6
XMLValidationException	0	0	4	4	0,3	0,3	2,4	8

A tabela abaixo mostra as classes classificadas como *God Class* tanto pela estratégia de detecção para modelo como pela correspondente para código em cada um dos sistemas. Além disso, para cada uma destas classes são mostrados os dados obtidos com a aplicação das métricas para modelo – WMC (MOD), ATFD (MOD) e CAM – e para código – WMC (COD), ATFD (COD) e TCC.

Tabela 18 – Resultados da estratégia de detecção *God Class*.

Sistema S2						
Classe	WMC		ATFD		CAM	TCC
	MOD	COD	MOD	COD		
Item	20	57	3	5	0,2	0,3
Sistema S3						
Classe	WMC		ATFD		CAM	TCC
	MOD	COD	MOD	COD		
LocadoraFachada	29	7	10	0	0,11	0
ClienteLocadoraFachada	30	8	9	0	0,1	0
ReservaFachada	26	6	12	0	0,1	0
Sistema S5						
Classe	WMC		ATFD		CAM	TCC
	MOD	COD	MOD	COD		
IntermediateParser	26	153	10	5	0,1	0
JInitFrame	16	54	3	9	0,05	0,04
Sistema S6						
Classe	WMC		ATFD		CAM	TCC
	MOD	COD	MOD	COD		
CompositeFigure	32	18	10	0	0,3	0,1
DrawApplication	53	55	13	7	0,3	0
StandardDrawingView	60	63	10	4	0,3	0,1
Sistema S7						
Classe	WMC		ATFD		CAM	TCC
	MOD	COD	MOD	COD		

AbstractTool	35	12	9	3	0,07	0,1
DrawApplication	88	133	24	11	0,03	0
StandardDrawingView	83	96	11	3	0,03	0,1
Sistema S8						
Classe	WMC		ATFD		CAM	TCC
	MOD	COD	MOD	COD		
AbstractAttributeCompositeFigure	23	55	5	12	0,11	0,2
AbstractFigure	69	115	11	5	0,2	0,1
AttributedFigure	20	51	6	9	0,1	0,2
BezierPath	37	90	5	7	0	0,2
DefaultDrawingView	65	128	13	7	0,05	0,07
GraphicalCompositeFigure	27	66	8	2	0,07	0,2
JSheet	38	88	1	8	0,2	0,2
TextAreaFigure	32	70	8	3	0,21	0,3
XMLElement	64	191	3	7	0,07	0,01
Sistema S9						
Classe	WMC		ATFD		CAM	TCC
	MOD	COD	MOD	COD		
AbstractFigure	59	102	13	2	0,03	0,1
AbstractTool	39	58	10	3	0,05	0,1
BezierPath	41	154	9	4	0,04	0,1
DefaultDrawingView	64	141	16	10	0,05	0,4
JSheet	34	80	12	7	0,1	0,2
XMLElement	55	114	4	6	0,1	0,1

A tabela abaixo mostra as classes classificadas como *Shotgun Surgery* tanto pela estratégia de detecção para modelo como pela correspondente para código em cada um dos sistemas. Além disso, para cada uma destas classes são mostrados os dados obtidos com a aplicação das métricas para modelo - CA, CC (MOD), CM (MOD)- e para código – CM (COD) e CC(COD).

Tabela 19 – Resultados da estratégia de detecção *Shotgun Surgery*.

Sistema S1					
Classe	CA	CM		CC	
		MOD	COD	MOD	COD
BeanLocator	1	10	13	6	12
CommandEngine	0	1	11	2	9
LabelValue	0	11	14	2	6
Report	0	21	19	12	8
ReportBuilder	1	15	17	6	6
Sistema S2					
Classe	CA	CM		CC	
		MOD	COD	MOD	COD
Collection	2	16	7	10	2
Item	4	9	6	12	2
Sistema S3					
Classe	CA	CM		CC	
		MOD	COD	MOD	COD

ClasseTitulo	1	14	8	10	6
Cliente	6	36	6	19	2
CopiaTitulo	1	20	9	18	6
Fornecedor	1	13	2	10	1
Titulo	2	26	8	16	6
Sistema S4					
Classe	CA	CM		CC	
		MOD	COD	MOD	COD
ConjuntoCriterios	5	8	1	8	1
Criterio	2	11	1	6	1
Metrica	3	11	1	16	1
ValorMetrica	1	12	2	15	1
Sistema S5					
Classe	CA	CM		CC	
		MOD	COD	MOD	COD
DataSource	2	42	9	34	6
Filter	2	11	8	7	4
XMIClass	2	21	7	19	3
Sistema S6					
Classe	CA	CM		CC	
		MOD	COD	MOD	COD
Command	1	12	11	12	6
StorableInput	0	24	6	26	2
StorableOutput	0	24	6	27	2
Sistema S7					
Classe	CA	CM		CC	ChC
		MOD	COD		
AbstractCommand	0	22	70	15	23
AbstractFigure	0	42	197	4	26
AbstractHandle	0	23	58	7	22
AbstractTool	0	55	108	14	23
AttributeFigure	0	33	45	8	14
CollectionsFactory	0	7	68	3	43
Command	8	20	30	13	8
CompositeFigure	2	14	17	6	9
ConnectionFigure	7	34	51	9	11
Connector	9	37	25	20	11
CreationTool	0	15	21	7	6
Desktop	2	9	8	9	4
DesktopListener	1	13	11	7	3
DrawApplication	2	18	26	7	7
Drawing	5	49	85	25	37
DrawingEditor	6	65	49	38	23
DrawingView	11	182	232	46	54
Figure	27	206	259	80	68
FigureAttributeConstant	19	28	39	16	12
FigureChangeEvent	1	47	18	14	11
FigureChangeListener	1	11	50	15	9
FigureEnumeration	2	53	112	46	44
FigureEnumerator	0	8	39	3	22
Geom	1	12	27	4	20
Handle	2	20	42	12	6
HandleEnumerator	2	22	17	23	18
Locator	5	27	9	18	4

Painter	2	9	8	9	1
TextHolder	3	16	40	9	4
Sistema S8					
Classe	CA	CM		CC	
		MOD	COD	MOD	COD
AbstractApplicationAction	0	11	23	7	8
AbstractBean	0	2	25	2	8
AbstractCompositeFigure	0	25	61	8	15
AbstractDrawing	0	22	21	10	9
AbstractFigure	0	19	173	8	37
AbstractHandle	0	23	103	8	17
AbstractProjectAction	0	16	30	10	12
AbstractSelectedAction	0	9	34	6	14
AbstractTool	0	52	90	8	16
Application	3	35	83	29	23
AttributedFigure	0	39	73	25	20
AttributeKey	6	40	79	24	35
AttributeKeys	0	12	108	21	42
BezierFigure	3	16	72	8	21
BezierPath	2	10	37	9	15
CompositeEdit	0	11	19	8	18
CompositeFigure	2	12	51	8	12
ConnectionFigure	10	37	50	19	13
Connector	9	41	33	16	10
Constrainer	4	7	19	6	10
DOMFactory	7	8	17	8	6
DOMInput	0	26	82	25	27
DOMOutput	0	26	82	25	28
DOMStorable	0	4	46	14	6
Drawing	2	33	153	18	47
DrawingEditor	8	74	65	45	32
DrawingView	7	42	187	30	56
Figure	17	203	371	84	68
FigureEvent	0	67	21	13	9
Geom	0	6	52	6	29
IXMLReader	4	27	41	12	7
JSheet	0	1	23	1	10
Locator	2	26	13	13	4
Project	6	72	128	27	25
ResourceBundleUtil	11	3	73	12	68
Tool	3	14	18	12	4
Sistema S9					
Classe	CA	CM		CC	
		MOD	COD	MOD	COD
AbstractAttributedDecoratorFigure	0	7	12	3	7
AbstractAttributeFigure	0	15	43	4	12
AbstractCompositeFigure	0	20	64	14	10
AbstractConnector	0	6	14	2	10
AbstractFigure	2	9	151	12	31
AbstractHandle	0	47	153	14	23
AbstractTool	0	52	94	20	16
AttributeKey	0	13	33	6	21
AttributeKeys	0	20	58	13	26
BezierFigure	6	5	41	16	16
BezierPath	7	5	29	9	6
ChopRectangleConnector	0	6	11	7	9

CompositeFigure	2	14	62	9	6
ConnectionFigure	7	21	47	9	8
Connector	11	26	34	18	7
Constrainer	4	7	28	6	14
Drawing	4	27	122	28	31
DrawingEditor	14	44	30	30	10
DrawingView	7	26	165	37	42
Figure	18	135	388	40	66
FigureEvent	0	47	22	13	9
IXMLReader	4	27	41	12	7
Locator	2	18	13	13	4
RelativeLocator	0	10	11	1	8
ResourceBundleUtil	11	17	0	16	1
View	4	72	58	27	6

A tabela abaixo mostra as classes classificadas como *God Package* tanto pela estratégia de detecção para modelo como pela correspondente para código em cada um dos sistemas. Além disso, para cada uma destas classes são mostrados os dados obtidos com a aplicação das métricas NOCC e NOCP para modelo e para código. A métrica PS tem os mesmos valores para modelo e código.

Tabela 20 – Resultados da estratégia de detecção *God Package*.

Sistema S3							
Pacote	PC		PS	NOCC		NOCP	
	MOD	COD		MOD	COD	MOD	COD
entidades	0,14	0,21	55	9	8	9	4
Sistema S4							
Pacote	PC		PS	NOCC		NOCP	
	MOD	COD		MOD	COD	MOD	COD
uml	0,09	0,12	23	21	26	4	6
Sistema S5							
Pacote	PC		PS	NOCC		NOCP	
	MOD	COD		MOD	COD	MOD	COD
metrics	0,05	0,1	22	10	22	3	4
Sistema S6							
Pacote	PC		PS	NOCC		NOCP	
	MOD	COD		MOD	COD	MOD	COD
standard	0,2	0,32	55	20	34	4	4
Sistema S7							
Pacote	PC		PS	NOCC		NOCP	
	MOD	COD		MOD	COD	MOD	COD
standard	0,17	0,3	73	41	94	8	9
util	0,1	0,1	43	108	136	9	11
Sistema S8							
Pacote	PC		PS	NOCC		NOCP	
	MOD	COD		MOD	COD	MOD	COD
app.action	0,24	0,28	36	1	23	0	8

draw	0,3	0,2	120	35	93	4	9
draw.action	0,11	0,13	33	0	26	0	5

A tabela abaixo mostra as classes classificadas como *Misplaced Class* tanto pela estratégia de detecção para modelo como pela correspondente para código em cada um dos sistemas. Além disso, para cada uma destas classes são mostrados os dados obtidos com a aplicação das métricas CL, NOED e DD para modelo e para código.

Tabela 21 – Resultados da estratégia de detecção *Misplaced Class*.

Sistema S1						
Classe	CL		NOED		DD	
	MOD	COD	MOD	COD	MOD	COD
MessageSenderImpl	0	0,3	3	7	2	4
PeriodicReportBehavior	0	0,26	9	9	5	5
ReportTag	0	0,24	1	8	1	4
Sistema S7						
Classe	CL		NOED		DD	
	MOD	COD	MOD	COD	MOD	COD
AbstractCommand	0,12	0,27	7	8	2	4
AbstractLocator	0	0,3	4	7	2	5
DrawApplication	0	0,09	34	87	4	8
FastBufferedUpdateStrategy	0	0,09	5	10	2	4
NullDrawingView	0,13	0,16	13	21	1	4
SelectAllCommand	0,3	0,25	7	47	5	11
SelectAreaTracker	0,3	0,5	8	14	4	5
StandardDrawingView	0,29	0,46	9	12	4	5
TextAreaFigure	0	0,15	11	41	4	11
UndoableHandle	0,16	0,09	5	10	1	4
ZoomDrawingView	0	0,11	10	25	4	8
Sistema S8						
Classe	CL		NOED		DD	
	MOD	COD	MOD	COD	MOD	COD
LineDecorationIcon	0	0,08	2	12	5	8