6
References

ARCOVERDE, R.; GARCIA, A. and FIGUEIREDO, E. Understanding the Longevity of Code Smells - Preliminary Results of an Explanatory Survey. 4th International Workshop on Refactoring Tools (WRT), in conjunction with the 33rd International Conference on Software Engineering (ICSE). May 2011.

ARCOVERDE, R.; MACIA, I.; GARCIA, A. and STAA, A. Automatically Detecting Architecturally-Relevant Code Anomalies. 3rd International Workshop on Recommendation Systems for Software Engineering (RSSE), in conjunction with the 34th International Conference on Software Engineering (ICSE). June 2012.

AYEWAH, N., HOVEMEYER, J. D. MORGEHTHALER, J. and PUGH, W. Using Static Analysis to Find Bugs. IEEE Software. vol. 25, 5, p. 22-29. September, 2008.

BAR-ILAN, J., MAT-HASSAN, M. and LEVENE, M. Methods for comparing rankings of search engine results – In Computer Networks: The International Journal of Computer and Telecommunications Networking - Web dynamics archive. Vol. 50, issue 10, p. 1448 – 1463. July, 2006.

BUSCHMANN, F., HENNEY, K., and SCHMIDT, D. Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing. In Wiley Software Patterns Series, John Wiley & Sons. 2007.

COUTO, C.; SILVA, C.; VALENTE, M.; BIGONHA, R. and ANQUETIL, N. Uncovering Causal Relationships between Software Metrics and Bugs. 16th European Conference on Software Maintenance and Reengineering (CSMR), March, 2012.

DIACONIS, P. and GRAHAM, R. Spearman’s footrule as a measure of disarray. In Journal of the Royal Statistical Society, Series B (Methodological), 39, p. 262-268. 1977.

EICHBERG, M., KLOPPENBURG, S., KLOSE, K. and MEZINI, M. Defining and continuous checking of structural program dependencies. In Proceedings of the 30th International Conference on Software engineering - ICSE '08. New York, NY, United States. 2008.

FAGIN, R., KUMAR, R. and SIVAKUMAR, D. Comparing top k lists. In Proceedings of the 14th annual ACM-SIAM symposium on Discrete algorithms (SODA '03). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, p. 28-36, 2003.

FIGUEIREDO, E., CACHO, N., SANT'ANNA, C., MONTEIRO, M., KULESZA, U., GARCIA, A., SOARES, S., FERRARI, F., KHAN, S., CASTOR, F. and DANTAS, F. Evolving software product lines with aspects: An empirical study on design stability. In Proceedings of the 30th International Conference on Software Engineering - ICSE '08. New York, NY, United States. 2008.

FIGUEIREDO, E., GARCIA, A. and LUCENA, C. AJATO: an AspectJ Assessment Tool. In proceedings of the 20th European Conference on Object-Oriented Programming - ECOOP'06, demo session D9. Nantes, France. 2006.

FISCHER, M., PINZGER, M. and GALL, H. Populating a release history database from version control and bug tracking systems. In Proceedings of the International Conference on Software Maintenance (ICSM), p. 23, Washington, DC, USA. 2003.

FOWLER, M. Refactoring: Improving the Design of Existing Code. Addison-Wesley. 1999.

GAMMA, E., HELM, R., JOHNSON, R. and VLISSIDES, J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. 1995.

GARCIA, J., POPESCU, D., EDWARDS, G. and NENAD MEDVIDOVIC. Identifying architectural bad smells. In Proceedings of the European Conference on Software Maintenance and Reengineering - CSMR '09. Washington, DC, United States. 2009.

GIT. Available at http://git-scm.com/. 2012.

GREENWOOD, P., BARTOLOMEI, T., FIGUEIREDO, E., DOSEA, M., GARCIA, A., CACHO, N., SANT’ANNA, C., SOARES, S., BORBA, P., KULESZA, U. and RASHID, A. On the impact of aspectual decompositions on design stability: An empirical study. In Proceedings of the 21st Conference of Object-Oriented Programming. - ECOOP'07. Springer-Verlag. pags. 176-200. 2007.

HEALTHWATCHER. Available at http://www.comp.lancs.ac.uk/~greenwop/tao/. 2012.

HOCHSTEIN, L. and LINDVALL, M. Combating architectural degeneration: A survey. Inf. Softw. Technol. 47. July, 2005.

INFUSION. Available at http://www.intooitus.com/infusion.html. 2012.

KELLY, D. A Study of Design Characteristics in Evolving Software Using Stability as a Criterion. IEEE Trans. Software Engineering, vol. 32, 5, p. 315-329. 2006.

KENDALL, M. and GIBBONS, J. Rank Correlation Methods. Edward Arnold, London. 1990.

KITCHENHAM, B., PFLEEGER, S., PICKARD, L., JONES, P., HOAGLIN, D., EMAM, K. and ROSENBERG, J. Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Trasactions on Software Engineering, vol. 28, p. 721-734. 2002.

KHOMH, F., PENTA, M. and GUEHENEUC, Y. An exploratory study of the impact of code smells on software change-proneness. In Proceedings of the 16th Working Conference on Reverse Engineering (WCRE), p. 75-84. 2009.

KIM, M., CAI, D. and KIM, S. An Empirical Investigation into the Role of API-Level Refactorings during Software Evolution. Proceedings of the 33rd International Conference on Software Engineering (ICSE), USA, 2011.

LANZA, M. and MARINESCU, R. Object-Oriented Metrics in Practice. In Springer-Verlag. New York, NY, United States. 2006.

LI, Z. Characterizing and Diagnosing Architectural Degeneration of Software Systems from Defect Perspective. In Electronic Thesis and Dissertation Repository. http://ir.lib.uwo.ca/etd/30. 2010.

MACCORMACK, A., RUSNAK, J. and BALDWIN, C. Exploring the structure of complex software designs: An empirical study of open source and proprietary code. In Management Science, 52(7), pags.1015-1030. 2006.

MACIA, I., ARCOVERDE, R., GARCIA, A., CHAVEZ, C. and STAA, A. On the Relevance of Code Anomalies for Identifying Architecture Degradation Symptoms. In Proceedings of the 16th European Conference on Software Maintenance and Reengineering - CSMR. Hungary. March, 2012b.

MACIA, I., GARCIA, J., POPESCU, D., GARCIA, A., MEDVIDOVIC, N. and STAA, A.. Are Automatically-Detected Code Anomalies Relevant to Architectural Modularity? An Exploratory Analysis of Evolving Systems. In Proceedings of the 11st International Conference on Aspect-oriented Software Development Companion - AOSD. Germany. March, 2012a.

MALEK, S., SEO, C., RAVULA, S., PETRUS, B. AND MEDVIDOVIC, N. Reconceptualizing a family of heterogeneous embedded systems via explicit architectural support. In Proceedings of the 29th international conference on Software Engineering (ICSE '07). IEEE Computer Society, Washington, DC, USA, 591-601.

MANTYLA, M and LASSENIUS, C. Subjective evaluation of software evolvability using code smells: An empirical study. Empirical Software Engineering, vol. 11, no. 3, pp. 395–431, 2006.

MARA, L., HONORATO, G., DANTAS, F., GARCIA, A. and LUCENA, C. Hist-Inspect: A Tool for History-Sensitive Detection of Code Smells. In Proceedings of the 10th International Conference on Aspect-oriented Software Development Companion - AOSD. 2011.

MARINESCU, R. Detection strategies: Metrics-based rules for detecting design flaws. In Proceedings of the 20th International Conference on Software Maintenance - ICSM. 2004.

MARINESCU, R. Detection strategies: Metrics-based rules for detecting design flaws. In Proceedings of the 20th International Conference on Software Maintenance (ICSM), pp 350-359, 2004.

MARINESCU, R., MARINESCU, C., RATIU, M. and WETTEL, R. iPlasma: An integrated platform for quality assessment of object-oriented design. In Proceedings of the 21th International Conference on Software Maintenance - ICSM'05. 2005.

MARTIN, R. Agile Software Development, Principles, Patterns, and Practices. Prentice Hall. 2002.

MARTIN, R. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall. 2008.

MARWAN A. and ALDRICH, J. Static Extraction and Conformance Analysis of Hierarchical Runtime Architectural Structure using Annotations. In Proceedings of the 24th international conference on Object-Oriented Programming, Systems, Languages and Applications - OOPSLA'09. 2009.

MARWAN, A. and ALDRICH, J. Static Extraction and Conformance Analysis of Hierarchical Runtime Architectural Structure using Annotations. Proceedings of the OOPSLA, 2009.

MEDVIDOVIC, N. and RICHARD, T. A classification and comparison framework for software architecture description languages. IEEE Transactions on Software Engineering. 2000.

MERKLE, B. Stop the software architecture erosion: building better software systems. In Proceedings of the ACM international conference companion on OOP systems languages and applications companion - SPLASH '10. New York, NY, United States, pags. 129-138. 2010.

MOHA, N., GUEHENEUC, Y., DUCHIEN, L. and LE MEUR, A. DECOR: A Method for the Specification and Detection of Code and Design Smells. In IEEE Trans. Soft. Eng. 36, pags. 20-36. January, 2010.

MORGAN, C. A static aspect language of checking design rules. In Proceedings of the 6th international conference on Aspect-oriented software development companion. 2007.

MORGAN, C. A static aspect language of checking design rules. Proceedings of the 6th AOSD, 63-72, 2007.

MURPHY-HILL, E., PARNIN, C. and BLACK, A. How We Refactor and How We Know It. Proceedings of the 31st International Conference on Software Engineering (ICSE). 2009.

NDEPEND. Available at http://www.ndepend.com/. 2012.

OLBRICH, S., CRUZES, D., BASILI, V. and ZAZWORKA, N.. The evolution and impact of code smells: A case study of two open source systems. In Proceedings of the 3rd International Symposium on Empirical Sofware Engineering and Measurement (ESEM), p. 390-400. 2009.

OLBRICH, S., CRUZES, D. and SJOBERG, D. Are all code smells harmful? A study of God Classes and Brain Classes in the evolution of three open source systems. In Proceedings of the 26th International Conference on Software Maintenance (ICSM), p. 1-10. 2010.

PERRY, D. and WOLF, A. Foundations for the study of software architecture. SIGSOFT Software. Eng. Notes 17.pags. 40-52. 1992.

PMD. Available at http://pmd.sourceforge.net/.2012.

RATZINGER, J. FISCHER, M. and GALL, H. Improving evolvability through refactoring. Proceedings 2nd International Workshop on Mining software repositories (MSR’05). ACM Press: New York NY, pags. 69–73, 2005.

SANGAL, N., JORDAN, E., SINHA, V. AND JACKSON, D. Using dependency models to manage complex software architecture. In Proceedings of the 20th on

Object-Oriented Programming, Systems, Languages and Applications - OOPSLA'05. 2005.

SCOOP. Available at http://www.inf.puc-rio.br/~ibertran/rsse12, May 2012.

SEMMLE CODE. Available at http://semmle.com/semmlecode/. 2012.

SOARES, G., GHEYI, R., SEREY, D. and MASSONI, T. Making Code Refactoring Safer. IEEE Software, vol. 27, p. 52-57. 2010.

SPRING. Available at http://www.springsource.org/. 2012.

SUBVERSION. Available at http://subversion.apache.org/. 2012.

SULLIVAN, K., GRISWOLD, W., CAI, Y. and HALLEN, B. The structure and value of modularity in software design. In 9th International Symposium on Foundations of Software Engineering - FSE. pags. 99-108. Vienna, Austria. 2001

TAYLOR, R., MEDVIDOVIC, N. and DASHOFY, E. Software Architecture: Foundations, Theory, and Practice. Wiley Publishing. 2009.

TERRA, R. and VALENTE, T. A Dependency Constraint Language to Manage Object-Oriented Software Architectures. In Software: Practice and Experience, v. 32, n. 12, John Wiley & Sons. 2009.

TERRA, R.; VALENTE, M.; CZARNECKI, K. and BIGONHA, R. Recommending Refactorings to Reverse Software Architecture Erosion. Proceedings of the 16th European Conference on Software Maintenance and Reengineering (CSMR), Early Research Achievements Track, p. 1-6, March 2012.

TOGETHER. Available at http://www.borland.com/us/products/together/. 2012.

VERBAERE, M., GODFREY, M. and GIRBA, T. Query Technologies and Applications for Program Comprehension. In Proceedings of the 16th IEEE Int’l Conference on Program Comprehension - ICPC'08. IEEE CS Press. pags. 185-288. 2008.

VESPUCCI– Modeling and Checking Software Architectures. Website. [Online]. http://www.opal-project.de/vespucci_project/index.html, May, 2012.

VIDAL, S. and MARCOS, C. Building an Expert System to Assist System Refactorization. In Expert Systems with Applications: An International Journal, Vol 39 Issue 3, February 2012.

WEIßGERBER, P. and DIEHL, S. Are refactorings less errorprone than other changes? In MSR ’06: Proceedings of the 2006 International Workshop on Mining Software Repositories, pages 112–118, New York, 2006. ACM.

WONG, S., CAI, Y. and DALTON, M. Detecting Design Defects Caused by Design Rule Violations. In Proceedings of the 18th ESEC/Foundations Software Engineering, 2010.

XI, G., DuBOSE, Q. and MURPHY-HILL, E. Reconciling Manual and Automatic Refactoring. In Proceedings of the 34th International Conference on Software Engineering (ICSE). 2012.

XING, Z. and STROULIA, E. Refactoring practice: How it is and how it should be supported — an Eclipse case study. In Proceedings of the 22nd IEEE International Conference on Software Maintenance (ICSM), p. 458–4. 2006.