

## 7

### Referências Bibliográficas

ACM RECOMMENDER SYSTEM. <http://recsys.acm.org/2012/>

ARNOLD, K., GOSLING, J. e HOLMES, D. The Java Programming Language. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000.

AZIVIENIS, A. et al. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, Los Alamitos, CA, USA, vol. 1, n. 1, p. 357-380. 2004.

BARBOSA, E. e GARCIA, A. Analyzing Crosscutting Exception Flows in an Evolving System. In Proceedings of the 4th Latin American Workshop on Aspect-Oriented Software Development – LA-WASP '10. Salvador, BA, Brasil. 2010.

BARBOSA, E. e GARCIA, A. Analyzing Exceptional Interfaces on Evolving Frameworks. In Proceedings of the 1st Workshop on Exception Handling in Contemporary Software Systems – EHCoS '11. São José dos Campos, SP, Brasil. 2011.

BRUCH, M., MEZINI, M. e MONPERRUS, M. Mining Subclassing Directives to Improve Framework Reuse. In Proceedings of the 7th IEEE Working Conference on Mining Software Repositories – MSR'10. 2010.

CABRAL, B. e MARQUES, P. Exception Handling: A field study in Java and .NET. In Proceedings of the 21st European Conference in Object-Oriented Programming – ECOOP'07. Berlin, Germany, 2007.

CHANG, B.-M. et al. Interprocedural Exception Analysis for Java. In Proceedings of the 2001 ACM Symposium on Applied Computing – SAC'01. 2001.

CHANG, B.-M. et al. Visualization of exception propagation for Java using static analysis.. In Proceedings of the Second International Workshop on Source Code Analysis. 2002.

CHEN, C.-T. et al. Exception handling refactorings: directed by goals and driven by bug fixing. The Journal of Systems and Software, vol 82, n 1, p 333-345. 2009.

COELHO, R. et al. Assessing the Impact of Aspects on Exception Flows: An Exploratory Study. In Proceedings of the 22nd European conference on Object-Oriented Programming – ECOOP '08. Berlin, Heidelberg: Springer-Verlag, 2008, p. 207-234.

DEVANBU, P., BRACHMAN, R. J. et al. LaSSIE: A Knowledge-Based Software Information System. Communications of the ACM, vol 34, n 5, p 34-39, 1991.

DREW, S e GOUGH, K. Exception handling: expecting the unexpected. Computer Languages, vol 8, p 69-86. 1994.

ECLIPSE-WIKI. <http://wiki.eclipse.org>.

ECKELS, B. Does Java need checked exceptions?, 2003. Disponível em <http://www.mindview.net/Etc/Discussions/CheckedExceptions>. Acessado em 20 de fevereiro de 2012.

FISCHER, G., HENNINGER, S. et al. Cognitive Tools for Locating and Comprehending Software Objects for Reuse. In Proceedings of the International Conference on Software Engineering - ICSE '91. Austin, TX, USA. 1991.

FOWLER, M e BECK, K. Refactoring: Improving the design of existing code. Addison-Wesley Longman Publishing Co., Inc. 1999.

FU, C. e RYDER, B. Exception Chain Analysis: Revealing Exception Handling Architecture in Java Server Applications. In Proceedings of the 29th International Conference on Software Engineering – ICSE'07. Minneapolis, MN, USA. 2007.

GAMMA, E. et al. Design Patterns: Elements of Reusable Object-Oriented Software. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. 1995.

GARCIA et al. A comparative study of exception handling mechanisms for building dependable object-oriented software. Journal of Systems and Software, vol 59, n 2, p 197-222. 2001.

GARCIA, I. e CACHO, N. eFlowMining: An Exception-Flow Analysis Tool for .NET Applications. In Proceedings of the 1st Workshop on Exception Handling in Contemporary Software Systems – EHCos '11. São José dos Campos, SP, Brasil. 2011.

GOLDBERG et al. Using collaborative filtering to weave an information tapestry. Communications of the ACM, vol 35, n 12, p 61-71. 1992.

GOODENOUGH, J. B. Exception handling: issues and a proposed notation. Communications of ACM, vol 18, n. 12, p. 683-696. 1975.

HILL, R. e RIDEOUT, J. Automatic Method Completion. In Proceedings of the IEEE International Conference of Automated Software Engineering. 2004.

HOLMES, R., WALKER, R. J. e MURPHY, G. C. Approximate Structural Context Matching: An Approach to Recommend Relevant Examples. *IEEE Transactions on Software Engineering*, vol 32, n. 12, p. 952-970. 2006.

KITCHENHAM, B. A. et al. Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, vol 28, n 8, 2002.

LANG, J. e STEWART, D. A study of the applicability of existing exception handling techniques to component-based real-time software. *ACM Computing Surveys*, vol 20, p 274-301. 1998.

LEE, P. e ANDERSON, T. Fault Tolerance: Principles and Practice. Berlin: Springer. 1990.

MAAREK, Y. S., BERRY, D. M. et al. An Information Retrieval Approach for Automatically Constructing Software Libraries. *IEEE Transactions on Software Engineering*, vol 17, n 8, p 800-813. 1991.

MACIA, I. et al. On the Relevance of Code Anomalies for Identifying Architecture Degradation Symptoms. In Proceedings of the 16th European Conference on Software Maintenance and Reengineering (CSMR'12), 2012.

MACLAREN, M. Exception handling in PL/I. ACM SIGPLAN Notices, vol 12, n 3. 1977.

MANDELIN, D. et al. Jungloid Mining: Helping to Navigate the API Jungle. In Proceedings of the ACM Conference of Programming Language Design and Implementation. 2005.

MANNING, C., D, RAGHAVAN, P. e SCHÜTZE, H. Introduction to information retrieval. Cambridge University Press. 2008.

MCCUNE, T. Exception Handling Antipatterns. Disponível em: <http://today.java.net/pub/a/today/2006/04/06/exception-handling-antipatterns.html> Acesso em: 15 de nov. 2011.

MICHAIL, A. Data Mining Library Reuse Patterns Using Generalized Association Rules. In Proceedings of the International Conference on Software Engineering – ICSE'00. 2000.

MILI et al. A Survey of Software Reuse Libraries. Systematic Software Reuse. W. Frakes, (Ed.) Bussum, The Netherlands: Baltzer Science, pp. 317-347. 1998.

PARNAS, D. L. The influence of software structure on reliability. In Proceedings of the International Conference on Reliable software. New York, NY, USA, 1976, p. 358-362.

PORTLAND PATTERN REPOSITORY.  
<http://c2.com/cgi/wiki?ExceptionPatterns>. Acessado em 01 de março de 2012.

RICH, E. User modeling via stereotypes. Cognitive Science, vol 3, n 4, p. 329-354. 1979.

RITTRI, M. Using Types as Search Keys in Function Libraries. Journal of Functional Programming, vol 1, n 1, p 71-89. 1989.

ROBILLARD, M. e MURPHY, G. C. Static Analysis to Support the Evolution of Exception Structure in Object-Oriented Systems. ACM Transactions on Software Engineering and Methodology, vol 12, n. 2, p. 191-221. 2003.

ROBILLARD, M., WALKER, R. J. e ZIMMERMANN, T. Recommendations Systems for Software Engineering. IEEE Software, vol. 27, n. 4, p. 80-86. 2010.

SALTON, G. Automatic Text Processing. Addison-Weasley, 1989.

SHAH, H. et al. Why do Developers Neglect Exception Handling?. In Proceedings of the 4th International Workshop on Exception Handling – WEH '08. Atlanta/GA, USA. 2008a.

SHAH, H., et al. Visualization of Exception Handling Constructs to Support Program Understanding. In Proceedings of the ACM Symposium on Software Visualization – SOFTVIS'08. Herrsching am Ammersee, Germany. 2008b.

SHAH, H., et al. Understanding Exception Handling: Viewpoints of Novices and Experts. IEEE Transactions on Software Engineering, vol. 36, n. 2, p. 151-161. 2010.

WIRFS-BROCK, R. J. Toward Exception-Handling Best Practices and Patterns. IEEE Software, vol 23, n. 5, p. 11-13. 2006.

WEIMER, W. e NECULA, G. C. Finding and Preventing Run-Time Error Handling Mistakes. In Proceedings of the 19th ACM Symposium on Object-

Oriented Programming System, Languages and Applications - OOPSLA '04. Vancouver: ACM, 2004.

YE, Y. e FISCHER, G. Reuse-Conducive Development Environments. Automated Software Engineering, vol. 12, n. 2, 2005, pp. 199–235. 2005.