References

1. Abran, A., Nguyen, K.: Measurement of the maintenance process from a demand-based perspective. Software Maintenance and Evolution: Research and Practice **5**(2) (1993) 63–90 [7]
2. Abran, A., Moore, J.W., Bourque, P., Dupuis, R., Tripp, L.L.: Guide to the Software Engineering Body of Knowledge (SWEBOK). IEEE (2004) [3, 93, 97, 308]
3. Advani, D., Hassoun, Y., Counsell, S.: Extracting refactoring trends from open-source software and a possible solution to the 'related refactoring' conundrum. In: Proc. Symposium on Applied computing, ACM Press (2006) 1713–1720 [96]
4. Aldrich, J., Chambers, C., Notkin, D.: ArchJava: Connecting software architecture to implementation. In: Proc. Int'l Conf. Software Engineering (ICSE), New York, ACM Press (May 19–25 2002) 187–197 [235]
5. Allen, R.: A Formal Approach to Software Architecture. PhD thesis, School of Computer Science, Carnegie Mellon University (January 1997) Issued as CMU Technical Report CMU-CS-97-144. [235]
6. Allen, R., Douence, R., Garlan, D.: Specifying and analyzing dynamic software architectures. In: Proc. Int'l Conf. Fundamental Aspects of Software Engineering (FASE). (1998) Lisbon, Portugal. [238]
7. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services. Springer-Verlag (2003) [139]
8. Altova: UModel UML software development tool. http://www.altova.com/products/umodel/uml_tool.html (2007) [167]
9. Ambler, S.W.: Agile modeling: Effective Practices for Extreme Programming and the Unified Process. John Wiley & Sons (2001) [176]
10. Ambler, S.W., Sadalage, P.J.: Refactoring Databases: Evolutionary Database Design. Addison-Wesley (2006) [292]
11. April A., Abran A.: Software Maintenance Management: Evaluation and Continuous Improvement, Wiley (2008) [292]
12. Andrade, L., Gouveia, J., Antunes, M., El-Ramly, M., Koutsoukos, G.: Forms2Net – Migrating Oracle Forms to Microsoft .NET. In Lämmel, R., Saraiva, J., Visser, J., eds.: Generative and Transformational Techniques in Software Engineering. Volume 4143 of Lecture Notes in Computer Science., Springer-Verlag (2006) [164]
13. ANSI/IEEE: Standard ANSI/IEEE 1042-1987 on Software Configuration Management. IEEE Press (1987) [293]

14. Antoniol, G., Casazza, G., Di Penta, M., Merlo, E.: Modeling clones evolution through time series. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2001) 273–280 [23]
15. Antoniol, G., Villano, U., Merlo, E., Di Penta, M.: Analyzing cloning evolution in the Linux kernel. Information and Software Technology **44**(13) (2002) [23]
16. Antoniol, G., Caprile, B., Potrich, A., Tonella, P.: Design-code traceability recovery: selecting the basic linkage properties. Science of Computer Programming **40**(2-3) (2001) 213–234 [9]
17. Aoyama, M.: Metrics and analysis of software architecture evolution with discontinuity. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), Orlando, Florida (2002) 103–107 [267]
18. Apache Ant: Apache ant. http://ant.apache.org (2007) [196]
19. ArgoUML: Cookbook for developers of ArgoUML. http://argouml.tigris.org/files/documents/4/0/argouml-0.14/cookbook-0.14.pdf (2006) [55, 237]
20. Arisholm, E., Sjoberg, D.: Evaluating the effect of a delegated versus centralized control style on the maintainability of object-oriented software. IEEE Trans. Software Engineering **30**(8) (2004) 521–534 [97]
21. Arnold, R.S.: An introduction to software restructuring. In Arnold, R.S., ed.: Tutorial on Software Restructuring. IEEE Press (1986) [6]
22. Arnold, R.S.: Software restructuring. In: Proc. IEEE. Volume 77. IEEE Computer Society Press (April 1989) 607–617 [5]
23. Arnold, R.S.: Software Reengineering. IEEE Computer Society Press (1993) [5, 291]
24. Arthur, L.J.: Software Evolution: The Software Maintenance Challenge. John Wiley & Sons (1988) [2, 291]
25. As-2 Embedded Computing Systems Committee SAE: Architecture Analysis & Design Language (AADL). SAE Standards n^o AS5506 (November 2004) [239]
26. Astels, D.: Refactoring with UML. In: Proc. Int'l Conf. eXtreme Programming and Flexible Processes in Software Engineering (XP). (2002) 67–70 Alghero, Sardinia, Italy. [96]
27. Bailey, J., Burd, E.: Evaluating clone detection tools for use during preventative maintenance. In: Proc. Workshop Source Code Analysis and Manipulation (SCAM), IEEE Computer Society Press (2002) 36–43 [29]
28. Baker, B.S.: A program for identifying duplicated code. In: Computer Science and Statistics: Proc. Symp. on the Interface. (March 1992) 49–57 [18]
29. Baker, B.S.: On finding duplication and near-duplication in large software systems. In Wills, L., Newcomb, P., Chikofsky, E., eds.: Proc. Working Conf. Reverse Engineering (WCRE), Los Alamitos, California, IEEE Computer Society Press (July 1995) 86–95 [15, 19, 27, 29, 30]
30. Baker, B.S.: Parameterized pattern matching: Algorithms and applications. Computer System Science **52**(1) (February 1996) 28–42 [27]
31. Balazinska, M., Merlo, E., Dagenais, M., Laguë, B., Kontogiannis, K.: Partial redesign of Java software systems based on clone analysis. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (1999) 326–336 [19, 25]
32. Balazinska, M., Merlo, E., Dagenais, M., Laguë, B., Kontogiannis, K.: Advanced clone analysis to support object-oriented system refactoring. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (October 2000) 98–107 [19, 25]
33. Balazinska, M., Merlo, E., Dagenais, M., Laguë, B., Kontogiannis, K.: Measuring clone based reengineering opportunities. In: Proc. IEEE Symp. Software Metrics, IEEE Computer Society Press (November 1999) 292–303 [19]

34. Baniassad, E., Clarke, S.: Theme: An approach for aspect-oriented analysis and design. In: Proc. Int'l Conf. Software Engineering (ICSE), Washington, DC, USA, IEEE Computer Society Press (2004) 158–167 [213]
35. Baniassad, E., Clements, P.C., Araujo, J., Moreira, A., Rashid, A., Tekinerdogan, B.: Discovering early aspects. IEEE Software **23**(1) (January-February 2006) 61–70 [213]
36. Barais, O.: Construire et Maîtriser l'Évolution d'une Architecture Logicielle à base de Composants. PhD thesis, LIFL, Université des Sciences et Technologies de Lille (Novembre 2005) [234, 258, 259]
37. Barais, O., Duchien, L.: SafArchie studio: An ArgoUML extension to build safe architectures. In: Architecture Description Languages. Springer-Verlag (2005) 85–100 [235, 236, 237, 256]
38. Barais, O., Le Meur, A.F., Duchien, L., Lawall, J.: Safe integration of new concerns in a software architecture. In: Proc. IEEE Int'l Symp. and Workshop on Engineering of Computer Based Systems (ECBS), IEEE Computer Society Press (2006) 52–64 [234, 256]
39. Basili, V.R., Briand, L.C., Melo, W.L.: A validation of object-oriented design metrics as quality indicators. IEEE Trans. Software Engineering **22**(10) (October 1996) 751–761 [7, 73]
40. Basili, V.R., Shull, F., Lanubile, F.: Building knowledge through families of experiments. IEEE Trans. Software Engineering **25**(4) (1999) 456–473 [86, 87]
41. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley (1998) [99]
42. Bass, L., Klein, M., Northrop, L.: Identifying aspects using architectural reasoning. Position paper presented at Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design, Workshop of the 3rd Int'l Conf. Aspect-Oriented Software Development (AOSD) (2004) [213]
43. Batini, C., Ceri, S., Navathe, S.B.: Conceptual Database Design : An Entity-Relationship Approach. Benjamin/Cummings (1992) [115]
44. Batista, T., Chavez, C., Garcia, A., Rashid, A., Sant'Anna, C., Kulesza, U., Filho, F.C.: Reflections on architectural connection: seven issues on aspects and ADLs. In: Proc. Int'l workshop on Early Aspects (EA), Int'l Conf. Software Engineering, New York, NY, USA, ACM Press (2006) 3–10 [245]
45. Baudry, B., Fleurey, F., France, R., Reddy, R.: Exploring the relationship between model composition and model transformation. In: Proc. Int'l Workshop on Aspect-Oriented Modeling (AOM), MoDELS 2005, Montego Bay, Jamaica (October 2005) [257]
46. Baxter, I., Pidgeon, C., Mehlich, M.: DMS®: Program transformations for practical scalable software evolution. In: Proc. Int'l Conf. Software Engineering (ICSE), Washington, DC, USA, IEEE Computer Society Press (2004) 625–634 [164]
47. Baxter, I.D., Yahin, A., Moura, L., Sant'Anna, M., Bier, L.: Clone detection using abstract syntax trees. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (1998) 368–377 [27, 29, 30]
48. Beck, K., Fowler, M.: Planning Extreme Programming. Addison-Wesley (2001) [177, 198]
49. Beck, K., Gamma, E.: Test infected: Programmers love writing tests. Java Report **3**(7) (1998) 51–56 [175, 176, 178, 180]
50. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley (1999) [3, 93, 176, 177, 178, 179, 185, 198]
51. Beck, K.: Aim, fire: Kent beck on test-first design. IEEE Software **18**(5) (September/October 2001) 87–89 [192]

52. Beck, K.: Test-Driven Development: By Example. Addison-Wesley (2003) [178, 179, 192]
53. Behm, A., Geppert, A., Dittrich, K.: On the migration of relational schemas and data to object-oriented database systems. In: Proc. Re-Technologies in Information Systems, Klagenfurt, Austria (December 1997) [110]
54. Bellon, S.: Vergleich von Techniken zur Erkennung duplizierten Quellcodes. Diploma thesis, no. 1998, University of Stuttgart (Germany), Institute for Software Technology (September 2002) [29, 30]
55. Bellon, S., Koschke, R., Antoniol, G., Krinke, J., Merlo, E.: Comparison and evaluation of clone detection tools. IEEE Trans. Software Engineering **33**(9) (September 2007) 577–591 [29, 30]
56. Bennett, K.: Legacy systems: Coping with success. IEEE Software **12**(1) (1995) 19–23 [173]
57. Bennett, K.H., Rajlich, V.T.: Software Maintenance and Evolution: A Roadmap. In: The Future of Software Engineering. ACM Press (2000) 75–87 [2, 3, 4, 267, 268, 282]
58. Bevan, J., Whitehead, E.J., Kim, S., Godfrey, M.: Facilitating software evolution research with Kenyon. In: Proc. European Software Engineering Conf. and Foundations of Software Engineering (ESEC/FSE), New York, NY, USA, ACM Press (2005) 177–186 [44, 45]
59. Beyer, D., Hassan, A.E.: Animated visualization of software history using evolution storyboards. In: Proc. Working Conf. Reverse Engineering (WCRE), Washington, DC, USA, IEEE Computer Society (2006) 199–210 [60]
60. Bezem, M., Klop, J.W., de Vrijer, R., eds.: Term Rewriting Systems. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press (March 2003) [150]
61. Bhat, T., Nagappan, N.: Building scalable failure-proneness models using complexity metrics for large scale software systems. In: Proc. Asia Pacific Software Engineering Conf. (APSEC), Washington, DC, USA, IEEE Computer Society Press (2006) 361–366 [87]
62. Bianchi, A., Caivano, D., Visaggio, G.: Method and process for iterative reengineering of data in a legacy system. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2000) 86– [110]
63. Biggerstaff, T.J., Mitbander, B.G., Webster, D.E.: Program understanding and the concept assignment problem. Comm. ACM **37**(5) (May 1994) 72–82 [99]
64. Biggerstaff, T.J., Mittbander, B.G., Webster, D.: The concept assignment problem in program understanding. In: Proc. Int'l Conf. Software Engineering (ICSE), IEEE Computer Society Press (1993) 482–498 [99]
65. Binder, R.: Design for testability in object-oriented systems. Comm. ACM **37**(9) (1994) 87–101 [193, 194, 195]
66. Binder, R.V.: Testing Object-Oriented Systems: Models, Patterns, and Tools. Object Technology Series. Addison-Wesley (1999) [6, 98, 174, 304]
67. Binkley, D., Ceccato, M., Harman, M., Ricca, F., Tonella, P.: Automated refactoring of object oriented code into aspects. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2005) 27–36 [218, 219, 220, 222, 223]
68. Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A.: Mining email social networks. In: Proc. Int'l Workshop on Mining Software Repositories (MSR), New York, NY, USA, ACM Press (2006) 137–143 [51]
69. Bisbal, J., Lawless, D., Wu, B., Grimson, J.: Legacy information systems: Issues and directions. IEEE Software **16**(5) (September/October 1999) 103–111 [107]
70. Blostein, D., Schürr, A.: Computing with Graphs and Graph Rewriting. Software – Practice and Experience, John Wiley & Sons **29**(3) (1999) 1–21 [167]

71. Boehm, B.W.: A spiral model of software development and enhancement. IEEE Computer **21**(5) (1988) 61–72 [2, 299]
72. Boehm, B.: Software Engineering Economics. Prentice Hall (1981) [7, 195]
73. Boger, M., Sturm, T., Fragemann, P.: Refactoring browser for UML. In: Proc. Int'l Conf. eXtreme Programming and Flexible Processes in Software Engineering (XP). (2002) 77–81 Alghero, Sardinia, Italy. [93, 95, 96]
74. Bohner, S.A., Arnold, R.S.: Software Change Impact Analysis. IEEE Computer Society Press (1996) [6, 291]
75. Bosch, J.: Design and Use of Software Architectures – Adopting and Evolving a Product Line Approach. Addison-Wesley (2000) [292]
76. Bottoni, P., Parisi Presicce, F., Taentzer, G.: Specifying integrated refactoring with distributed graph transformations. Lecture Notes in Computer Science **3062** (2003) 220–235 [94]
77. Bouktif, S., Gueheneuc, Y.G., Antoniol, G.: Extracting change-patterns from CVS repositories. In: Proc. Working Conf. Reverse Engineering (WCRE), Washington, DC, USA, IEEE Computer Society Press (2006) 221–230 [59]
78. Bradbury, J.S., Cordy, J.R., Dingel, J., Wermelinger, M.: A survey of self-management in dynamic software architecture specifications. In: Proc. ACM SIGSOFT workshop on Self-managed systems (WOSS), New York, NY, USA, ACM Press (2004) 28–33 [238]
79. Braem, M., Gybels, K., Kellens, A., Vanderperren, W.: Automated pattern-based pointcut generation. In: Proc. Int'l Symp. Software Composition (SC), Springer-Verlag (2006) 66–81 [221, 223]
80. Breu, S., Zimmermann, T.: Mining aspects from version history. In: Proc. Int'l Conf. Automated Software Engineering (ASE), Washington, DC, USA, IEEE Computer Society Press (2006) 221–230 [59]
81. Briand, L.C., Labiche, Y., Yan, H.D., Pent, M.D.: A controlled experiment on the impact of the object constraint language in UML-based development. icsm (2004) 380–389 [241]
82. Briand, L.C., Wüst, J., Ikonomovski, S.V., Lounis, H.: Investigating quality factors in object-oriented designs: an industrial case study. In: Proc. Int'l Conf. Software Engineering (ICSE), Los Alamitos, CA, USA, IEEE Computer Society Press (1999) 345–354 [74]
83. Brito e Abreu, F., Melo, W.: Evaluating the impact of object-oriented design on software quality. In: Proc. IEEE Symp. Software Metrics. (March 1996) 90–99 [7]
84. Brodie, M.L., Stonebraker, M.: Migrating Legacy Systems. Gateways, Interfaces, and the Incremental Approach. Morgan Kaufmann (1995) [91, 106, 107, 111, 123, 137, 291, 301]
85. Brooks, F.P.: The Mythical Man-Month: Essays on Software Engineering. 20th anniversary edn. Addison-Wesley (1995) [7, 174]
86. Brown, W.J., Malveau, R.C., McCormick, H.W., Mowbray, T.J.: AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis. John Wiley & Sons (1998) [99, 102]
87. Bruneton, E., Coupaye, T., Leclercq, M., Quema, V., Stefani, J.B.: An open component model and its support in Java. In: Component-Based Software Engineering. (2004) 7–22 [235]
88. Bruntink, M., D'Hondt, M., van Deursen, A., Tourwé, T.: Simple crosscutting concerns do not exist. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM Press (2007) 199–211 [203, 211, 218, 223, 225]
89. Bruntink, M., van Deursen, A.: An empirical study into class testability. Systems and Software **79**(9) (2006) 1219–1232 [193, 194, 197]

90. Bruntink, M., van Deursen, A., Tourwé, T.: Isolating Idiomatic Crosscutting Concerns. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2005) 37– 46 [211, 212]
91. Bruntink, M., van Deursen, A., Tourwé, T.: Discovering faults in idiom-based exception handling. In: Proc. Int'l Conf. Software Engineering (ICSE), ACM Press (2006) 242–251 [203, 211, 212, 217]
92. Bruntink, M., van Deursen, A., van Engelen, R., Tourwé, T.: An evaluation of clone detection techniques for identifying crosscutting concerns. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society (2004) 200–209 [20, 34, 214]
93. Bruntink, M., van Engelen, R., Tourwé, T.: On the use of clone detection for identifying crosscutting concern code. IEEE Trans. Software Engineering **31**(10) (2005) 804–818 [30, 34, 109, 214]
94. Buckley, J., Mens, T., Zenger, M., Rashid, A., Kniesel, G.: Towards a taxonomy of software change. Software Maintenance and Evolution: Research and Practice **17**(5) (September/October 2005) 309–332 [5]
95. Bugzilla: Bugzilla.
http://www.bugzilla.org (2007) [41]
96. Bures, T., Hnetynka, P., Plasil, F.: Sofa 2.0: Balancing advanced features in a hierarchical component model. In: Proc. Int'l Conf. Software Engineering Research, Management and Applications (SERA), Washington, DC, USA, IEEE Computer Society Press (2006) 40–48 [235]
97. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-Oriented Software Architecture – A System of Patterns. John Wiley & Sons (1996) [98]
98. C2 Wiki: Two year itch.
http://c2.com/cgi/wiki?TwoYearItch (January 20 2007) [184]
99. Canfora, G., Cimitile, A., Munro, M.: An improved algorithm for identifying objects in code. Software – Practice and Experience, John Wiley & Sons **26**(1) (1996) 25–48 [109]
100. Canfora, G., Fasolino, A.R., Frattolillo, G., Tramontana, P.: Migrating interactive legacy systems to web services. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR), Washington, DC, USA, IEEE Computer Society Press (2006) 24–36 [168]
101. Canfora, G., Santo, G.D., Zimeo, E.: Developing and executing Java AWT applications on limited devices with TCPTE. In: Proc. Int'l Conf. Software Engineering (ICSE), New York, NY, USA, ACM Press (2006) 787–790 [109]
102. Capiluppi, A., Gonzales-Barahona, J., Herraiz, I., Robles, G.: Adapting the "staged model for software evolution" to free/libre/open source software. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), Dubrovnik, Croatia (3-4 Sept. 2007) [282]
103. Capiluppi, A., Morisio, M., Ramil, J.F.: The evolution of source folder structure in actively evolved open source systems. In: Proc. IEEE Symp. Software Metrics, IEEE Computer Society Press (2004) 2–13 [275]
104. Capiluppi, A., Morisio, M., Ramil, J.F.: Structural evolution of an open source system: A case study. In: Int'l Workshop on Program Comprehension (IWPC). (2004) 172–182 [275]
105. Capiluppi, A., Ramil, J.F.: Studying the evolution of open source systems at different levels of granularity: Two case studies. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), IEEE Computer Society Press (2004) 113–118 [275, 278]
106. Carrière, S.J., Woods, S.G., Kazman, R.: Software architectural transformation. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (1999) 13–23 [167]

107. Casais, E.: Automatic reorganization of object-oriented hierarchies: a case study. Object Oriented Systems **1** (1994) 95–115 [93]
108. Cenqua: Clover. http://www.cenqua.com/clover/ (January 20 2007) [192]
109. Chapin, N., Hale, J., Khan, K., Ramil, J., Than, W.G.: Types of software evolution and software maintenance. Software Maintenance and Evolution: Research and Practice **13** (2001) 3–30 [5]
110. Chen, K., Schach, S.R., Yu, L., Offutt, A.J., Heller, G.Z.: Open-source change logs. Empirical Software Engineering **9**(3) (2004) 197–210 [283]
111. Chidamber, S.R., Kemerer, C.F.: A metrics suite for object-oriented design. IEEE Trans. Software Engineering **20**(6) (June 1994) 476–493 [73, 195]
112. Chikofsky, E.J., Cross, J.H.: Reverse engineering and design recovery: A taxonomy. IEEE Software **7**(1) (1990) 13–17 [5, 106, 177, 303]
113. Chou, A., Yang, J., Chelf, B., Hallem, S., Engler, D.R.: An empirical study of operating system errors. In: Proc. Symp. Operating Systems Principles. (2001) 73–88 [22]
114. Church, K.W., Helfman, J.I.: Dotplot: A program for exploring self-similarity in millions of lines for text and code. Journal of American Statistical Association, Institute for Mathematical Statistics and Interface Foundations of North America **2**(2) (June 1993) 153–174 [31]
115. Clements, P.: A survey of architecture description languages. In: Proc. Int'l Workshop on Software Specification and Design, IEEE Computer Society Press (March 1996) 16–25 [235]
116. Cleve, A., Hainaut, J.L.: Co-transformations in database applications evolution. In: Generative and Transformational Techniques in Software Engineering. Volume 4143 of Lecture Notes in Computer Science. Springer-Verlag (2006) 409–421 [116]
117. Cleve, A., Henrard, J., Hainaut, J.L.: Data reverse engineering using system dependency graphs. In: Proc. Working Conf. Reverse Engineering (WCRE), Washington, DC, USA, IEEE Computer Society Press (2006) 157–166 [132]
118. Coady, Y., Kiczales, G., Feeley, M., Smolyn, G.: Using AspectC to improve the modularity of path-specific customization in operating system code. In: Proc. European Software Engineering Conf. (ESEC), ACM Press (2001) 88–98 [211]
119. Cockburn, A.: Agile Software Development. Addison-Wesley (2001) [3, 8, 174]
120. Code Generation Network: List of code generators. http://www.codegeneration.net/generators.php (2007) [168]
121. Collberg, C., Kobourov, S., Nagra, J., Pitts, J., Wampler, K.: A system for graph-based visualization of the evolution of software. In: Proc. ACM Symp. Software Visualization, New York, NY, USA, ACM Press (2003) 77–86 [65]
122. Colyer, A., Clement, A.: Large-scale AOSD for middleware. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM Press (2004) 56–65 [211]
123. Comino, S., Manenti, F.M., Parisi, M.L.: From planning to mature: on the determinants of open source take off. Technical report, Trento University, Dept. of Economics (January 2005) [282, 283]
124. Cooper, K., McIntosh, N.: Enhanced code compression for embedded RISC processors. In: Proc. ACM Conf. on Programming Language Design and Implementation, ACM Press (May 1999) 139–149 [34]
125. Corbi, T.A.: Program understanding: Challenge for the 1990s. IBM Systems Journal **28**(2) (1989) 294–306 [176, 177]
126. Cordy, J., Dean, T., Malton, A., Schneider, K.: Source transformation in software engineering using the TXL transformation system. Information and Software Technology **44**(13) (2002) 827–837 [164]

127. Cordy, J.R., Dean, T.R., Synytskyy, N.: Practical language-independent detection of near-miss clones. In: Proc. Conf. Centre for Advanced Studies on Collaborative research (CASCON), IBM Press (2004) 1–12 [27]
128. Cordy, J.: Comprehending reality: Practical challenges to software maintenance automation. In: Int'l Workshop on Program Comprehension (IWPC), IEEE Computer Society Press (2003) 196–206 [21, 22]
129. Corradini, A., Montanari, U., Rossi, F.: Graph processes. Fundamenta Informaticae **26**(3 and 4) (1996) 241–265 [148]
130. Counsell, S., Hassoun, Y., Loizou, G., Najjar, R.: Common refactorings, a dependency graph and some code smells: an empirical study of Java OSS. In: Proc. Int'l Symp. Empirical Software Engineering, ACM Press (2006) 288–296 [96]
131. Crispin, L., House, T.: Testing Extreme Programming. Addison-Wesley (2002) [174, 179, 200]
132. Cristianini, N., Shawe-Taylor, J.: An introduction to Support Vector Machines. Cambridge University Press (2000) [78]
133. Cubranic, D., Murphy, G.C.: Hipikat: recommending pertinent software development artifacts. In: Proc. Int'l Conf. Software Engineering (ICSE), Portland, Oregon, IEEE Computer Society Press (2003) 408–418 [43]
134. Cunningham, W.: Episodes: A pattern language of competitive development. In Vlissides, J., ed.: Pattern Languages of Program Design 2. Addison-Wesley (1996) [179]
135. CVS: Concurrent versions systems. http://www.nongnu.org/cvs (2006) [40]
136. D'Ambros, M.: Software archaeology – reconstructing the evolution of software systems. Master thesis, Politecnico di Milano (April 2005) [40, 46]
137. D'Ambros, M., Lanza, M.: Reverse engineering with logical coupling. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (October 2006) 189–198 [52, 53, 56, 57, 58]
138. D'Ambros, M., Lanza, M.: Software bugs and evolution: A visual approach to uncover their relationship. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR), IEEE Computer Society Press (2006) 227–236 [48]
139. D'Ambros, M., Lanza, M., Gall, H.: Fractal figures: Visualizing development effort for CVS entities. In: Proc. Int'l Workshop on Visualizing Software for Understanding (Vissoft), IEEE Computer Society Press (2005) 46–51 [46, 47, 48]
140. D'Ambros, M., Lanza, M., Lungu, M.: The evolution radar: Integrating fine-grained and coarse-grained logical coupling information. In: Proc. Int'l Workshop on Mining Software Repositories (MSR). (2006) 26–32 [52]
141. Dashofy, E.M., van der Hoek, A., Taylor, R.N.: An infrastructure for the rapid development of XML-based architecture description languages. In: Proc. Int'l Conf. Software Engineering (ICSE), New York, NY, USA, ACM Press (2002) 266–276 [235, 236]
142. David, P.C., Ledoux, T.: Safe dynamic reconfigurations of Fractal architectures with FScript. In: Proc. Fractal CBSE Workshop, ECOOP'06, Nantes, France (2006) [238]
143. DB-MAIN: The DB-MAIN official website. http://www.db-main.be (2006) [129]
144. de Lucia, A., Lucca, G.A.D., Fasolino, A.R., Guerra, P., Petruzzelli, S.: Migrating legacy systems towards object-oriented platforms. In: Proc. Int'l Conf. Software Maintenance (ICSM), Los Alamitos, CA, USA, IEEE Computer Society Press (1997) 122 [109]
145. Del Grosso, C., Di Penta, M., Garcia-Rodriguez de Guzman, I.: An approach for mining services in database oriented applications. Proc. European Conf. Software Maintenance and Reengineering (CSMR) (2007) 287–296 [168]

146. Delcroix, C., Thiran, P., Hainaut, J.L.: Transformational approach to data reengineering. Ingénierie des Systèmes d'Information (December 2001) (in French). [130]
147. Demeyer, S., Bois, B.D., Rieger, M., Rompaey, B.V.: The LAN-simulation: A refactoring lab session. In: Proc. 1st Workshop on Refactoring Tools, University of Berlin (2007) [97]
148. Demeyer, S., Ducasse, S., Nierstrasz, O.: Finding refactorings via change metrics. In: Proc. ACM SIGPLAN Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA). Volume 35 of SIGPLAN Notices., ACM Press (October 2000) 166–177 [95]
149. Demeyer, S., Ducasse, S., Nierstrasz, O.: Object-Oriented Reengineering Patterns. Morgan Kaufmann, San Francisco, CA, USA (2002) [5, 92, 98, 99, 100, 102, 103, 139, 174, 200, 292, 300, 301]
150. Demeyer, S., Van Rysselberghe, F., Girba, T., Ratzinger, J., Marinescu, R., Mens, T., Du Bois, B., Janssens, D., Ducasse, S., Lanza, M., Rieger, M., Gall, H., El-Ramly, M.: The LAN-simulation: A refactoring teaching example. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), IEEE Computer Society Press (2005) 123–131 [97]
151. Di Lucca, G., Di Penta, M., Fasolino, A.: An approach to identify duplicated web pages. In: Proc. Int'l Computer Software and Applications Conf. (COMPSAC). (2002) 481–486 [27]
152. Dig, D., Johnson, R.: The role of refactorings in api evolution. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2005) 389–398 [96]
153. Dijkstra, E.W.: On the role of scientific thought (EWD447). In: Selected Writings on Computing: A Personal Perspective. Springer-Verlag (1982) 60–66 [234]
154. Du Bois, B.: A Study of Quality Improvements by refactoring. PhD thesis, University of Antwerp (September 2006) [97]
155. Duala-Ekoko, E., Robillard, M.: Tracking code clones in evolving software. In: Proc. Int'l Conf. Software Engineering (ICSE), ACM Press (2007) 158–167 [23]
156. Ducasse, S., Rieger, M., Demeyer, S.: A language independent approach for detecting duplicated code. In Yang, H., White, L., eds.: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (September 1999) 109–118 [15, 26, 29, 30, 31]
157. Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. Springer-Verlag (2006) [148]
158. El-Ramly, M., Eltayeb, R., Alla, H.: An experiment in automatic conversion of legacy Java programs to C#. In: Proc. IEEE Int'l Conf. Computer Systems and Applications. (2006) 1037–1045 [108]
159. Elbaum, S., Gable, D., Rothermel, G.: The impact of software evolution on code coverage information. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2001) 170–179 [201]
160. Endres, A., Rombach, D.: A Handbook of Software and Systems Engineering. Pearson Addison-Wesley (2003) [70, 72]
161. Erdogmus, H., Morisio, M., Torchiano, M.: On the effectiveness of the test-first approach to programming. IEEE Trans. Software Engineering **31**(3) (2005) 226–237 [201]
162. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, Upper Saddle River, NJ, USA (2005) [9, 140, 304]
163. Ettinger, R., Verbaere, M.: Untangling: a slice extraction refactoring. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM Press (March 2004) 93–101 [218]
164. Fabry, J.: Modularizing Advanced Transaction Management – Tackling Tangled Aspect Code. PhD thesis, Vrije Universiteit Brussel (September 2005) [203]

165. Fahmy, H., Holt, R.C., Cordy, J.R.: Wins and losses of algebraic transformations of software architectures. In: Proc. Int'l Conf. Automated Software Engineering (ASE), Washington, DC, USA, IEEE Computer Society Press (2001) 51– [167]
166. Fanta, R., Rajlich, V.: Removing clones from the code. Software Maintenance and Evolution: Research and Practice **11**(4) (July/Aug. 1999) 223–243 [24]
167. Favre, J.M.: Languages evolve too! Changing the software time scale. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), Los Alamitos, CA, USA, IEEE Computer Society (2005) 33–44 [9]
168. FEAST: Feedback, evolution and software technology projects website. http://www.doc.ic.ac.uk/ mml/feast/ (Sept 2001) [266]
169. Feathers, M.C.: Working Effectively with Legacy Code. Prentice Hall (2005) [100, 104, 292]
170. Feitelson, D.G., Heller, G.Z., Schach, S.R.: An empirically-based criterion for determining the success of an open-source project. In: Proc. Australian Software Engineering Conf. (ASWEC). (21 April 2006) 6 pp. [286]
171. Fenton, N., Pfleeger, S.L.: Software Metrics: A Rigorous and Practical Approach. 2nd edn. International Thomson Computer Press, London, UK (1997) [7, 296]
172. Fenton, N.E., Ohlsson, N.: Quantitative analysis of faults and failures in a complex software system. IEEE Trans. Software Engineering **26**(8) (2000) 797–814 [73, 87]
173. Ferenc, R., Beszédes, Á.: Data exchange with the Columbus schema for C++. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR), Washington, DC, USA, IEEE Computer Society Press (2002) 59–66 [163]
174. Fernandez-Ramil, J., Hall, P.: Maintaining and Evolving Software. M882 Course on Managing the Software Enterprise, Learning Space, The Open University, http://openlearn.open.ac.uk/course/view.php?id=1698 (2007) Work licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 2.0 Licence. [265, 268]
175. Ferrante, J., Ottenstein, K., Warren, J.: The program dependence graph and its use in optimization. ACM Trans. Programming Languages and Systems **9**(3) (July 1987) 319–349 [28]
176. Filman, R., Friedman, D.: Aspect-oriented programming is quantification and obliviousness (2000) October 2000, Minneapolis. http://ic-www.arc.nasa.gov/ic/darwin/oif/leo/filman/text/oif/aop-is.pdf. [209, 244]
177. Fioravanti, F., Nesi, P.: Estimation and prediction metrics for adaptive maintenance effort of object-oriented systems. IEEE Trans. Software Engineering **27**(12) (2001) 1062–1084 [7]
178. Fischer, M., Gall, H.: EvoGraph: A lightweight approach to evolutionary and structural analysis of large software systems. In: Proc. Working Conf. Reverse Engineering (WCRE), Washington, DC, USA, IEEE Computer Society (2006) 179–188 [60]
179. Fischer, M., Pinzger, M., Gall, H.: Populating a release history database from version control and bug tracking systems. In: Proc. Int'l Conf. Software Maintenance (ICSM), Los Alamitos CA, IEEE Computer Society Press (September 2003) 23–32 [40]
180. Flammia, G.: On the internet, software should be milked, not brewed. IEEE Expert **11**(6) (December 1996) 87–88 [34]
181. Foote, B., Yoder, J.W.: Big ball of mud. In Harrison, N., Foote, B., Rohnert, H., eds.: Pattern Languages of Program Design. Volume 4. Addison-Wesley (2000) 654–692 [98, 102]
182. Fowler, M.: Analysis Patterns: Reusable Object Models. Addison-Wesley (1997) [98, 228]

183. Fowler, M.: Refactoring: Improving the Design of Existing Code. Addison-Wesley, Boston, MA, USA (1999) [6, 21, 35, 92, 93, 96, 101, 104, 165, 167, 175, 180, 182, 184, 185, 186, 187, 189, 191, 198, 216, 218, 278, 291, 296, 303]
184. G. Sunyé, Pollet, D., LeTraon, Y., Jézéquel, J.M.: Refactoring UML models. In: Proc. UML. Volume 2185 of Lecture Notes in Computer Science., Springer-Verlag (2001) 134–138 [93, 95, 96]
185. Gall, H., Hajek, K., Jazayeri, M.: Detection of logical coupling based on product release history. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (November 1998) [59]
186. Gall, H., Jazayeri, M., Klösch, R., Trausmuth, G.: Software evolution observations based on product release history. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (September 1997) 160–166 [270]
187. Gall, H., Jazayeri, M., Krajewski, J.: CVS release history data for detecting logical couplings. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), Los Alamitos CA, IEEE Computer Society Press (2003) 13–23 [59]
188. Gall, H., Jazayeri, M., Riva, C.: Visualizing software release histories: The use of color and third dimension. In Yang, H., White, L., eds.: Proc. Int'l Conf. Software Maintenance (ICSM), Oxford, UK, IEEE Computer Society (1999) 99–108 [65]
189. Galli, M., Lanza, M., Nierstrasz, O., Wuyts, R.: Ordering broken unit tests for focused debugging. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2004) 114–123 [200]
190. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Languages and Systems. Addison-Wesley (1994) [98, 163]
191. Garcez, A., Russo, A., Nuseibeh, B., Kramer, J.: Abductive reasoning and inductive learning to evolve requirements specifications. IEE Proceedings – Software **150**(1) (February 2003) 25–38 [8]
192. Garcia, A., Chavez, C., Batista, T., Sant'Anna, C., Kulesza, U., Rashid, A., de Lucena, C.J.P.: On the modular representation of architectural aspects. In: Proc. European Workshop on Software Architecture (EWSA). (2006) 82–97 [244]
193. Garcia, A., Sant'Anna, C., Figueiredo, E., Kulesza, U., Lucena, C., von Staa, A.: Modularizing design patterns with aspects: a quantitative study. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM Press (2005) 3–14 [212]
194. Gartner Group: Gartner. http://www.gartner.com (2007) [139]
195. German, D.: Mining CVS repositories, the softChange experience. In: Proc. Int'l Workshop on Mining Software Repositories (MSR). (2004) 17–21 [43, 45]
196. German, D., Hindle, A., Jordan, N.: Visualizing the evolution of software using softChange. In: Proc. Int'l Conf. on Software Engineering & Knowledge Engineering (SEKE), New York NY, ACM Press (2004) 336–341 [43]
197. German, D.M.: A study of the contributors of PostgreSQL. In: Proc. Int'l Workshop on Mining Software Repositories (MSR), New York, NY, USA, ACM Press (2006) 163–164 [43]
198. Gibbs, C., Liu, C.R., Coady, Y.: Sustainable system infrastructure and big bang evolution: Can aspects keep pace? In: Proc. European Conf. Object-Oriented Programming (ECOOP), Springer-Verlag (2005) 241–261 [212]
199. Giesecke, S.: Clone-based Reengineering für Java auf der Eclipse-Plattform. Diplomarbeit, Carl von Ossietzky Universität Oldenburg, Department für Informatik, Abteilung Software Engineering, Germany (2003) [24]
200. Gilb, T.: Evolutionary development. ACM Software Engineering Notes **6**(2) (April 1981) 17– [2]

201. Girard, J.F., Koschke, R., Schied, G.: A metric-based approach to detect abstract data types and state encapsulations. Automated Software Engineering **6**(4) (1999) 357–386 [109]
202. Gîrba, T., Ducasse, S., Lanza, M.: Yesterday's weather: Guiding early reverse engineering efforts by summarizing the evolution of changes. In: Proc. Int'l Conf. Software Maintenance (ICSM), Chicago, Illinois, IEEE Computer Society Press (2004) 40–49 [53]
203. Gîrba, T., Kuhn, A., Seeberger, M., Ducasse, S.: How developers drive software evolution. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), IEEE Computer Society Press (2005) 113–122 [50, 51]
204. Gitchell, D., Tran, N.: Sim: a utility for detecting similarity in computer programs. In: Proc. Technical Symp. Computer Science Education (SIGCSE), ACM Press (1999) 266–270 [27]
205. Godfrey, M., Tu, Q.: Growth, evolution and structural change in open source software. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE). (September 2001) [23, 35]
206. Godfrey, M.W., Tu, Q.: Evolution in open source software: A case study. In: Proc. Int'l Conf. Software Maintenance (ICSM), Los Alamitos, California, IEEE Computer Society Press (2000) 131–142 [8, 35, 268, 270, 271, 274, 286]
207. Godfrey, M.W., Zou, L.: Using origin analysis to detect merging and splitting of source code entities. IEEE Trans. Software Engineering **31**(2) (2005) 166–181 [95]
208. Grady, R.B.: Successful Software Process Improvement. 1st edn. Prentice Hall (1997) [8]
209. Graves, T.L., Karr, A.F., Marron, J., Siy, H.: Predicting fault incidence using software change history. IEEE Trans. Software Engineering **26**(7) (April 2000) 653–661 [82, 87]
210. Grier, S.: A tool that detects plagiarism in Pascal programs. SIGSCE Bulletin **13**(1) (1981) 15–20 [34]
211. Griswold, W., Kato, Y., Yuan, J.: Aspect browser: Tool support for managing dispersed aspects. In: Workshop on Multi-Dimensional Separation of Concerns in Object-oriented Systems. (1999) [213]
212. Gross, H.G.: Component-Based Software Testing with UML. Springer-Verlag (2005) [200]
213. Grubb, P., Takang, A.A.: Software Maintenance: Concepts and Practice. 2nd edn. World Scientific (2003) [292]
214. Hainaut, J.L.: Introduction to Database Reverse Engineering. 3rd edn. LIBD Publish., Namur (2002) [120]
215. Hainaut, J.L., Englebert, V., Henrard, J., Hick, J.M., Roland, D.: Database reverse engineering: From requirements to CARE tools. Automated Software Engineering **3** (1996) 9–45 [112, 120, 129]
216. Hainaut, J.L.: Specification preservation in schema transformations – application to semantics and statistics. Data Knowledge Engineering **19**(2) (1996) 99–134 [115]
217. Hainaut, J.L.: The transformational approach to database engineering. In Lämmel, R., Saraiva, J., Visser, J., eds.: Generative and Transformational Techniques in Software Engineering. Volume 4143 of Lecture Notes in Computer Science., Springer-Verlag (2006) 95–143 [113, 114]
218. Hainaut, J.L., Henrard, J., Hick, J.M., Roland, D., Englebert, V.: Database design recovery. In: Proc. Int'l Conf. Advances Information System Engineering (CAiSE). Volume 1080 of Lecture Notes in Computer Science., Springer-Verlag (1996) 272–300 [116]
219. Hainaut, J.L., Hick, J.M., Henrard, J., Roland, D., Englebert, V.: Knowledge transfer in database reverse engineering: A supporting case study. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (1997) [120]

220. Hanenberg, S., Oberschulte, C., Unland, R.: Refactoring of aspect-oriented software. In: Proc. Int'l Conf. Object-Oriented and Internet-based Technologies, Concepts, and Applications for a Networked World (Net.ObjectDays), Springer-Verlag (2003) 19–35 [218, 220, 222, 227]
221. Hannemann, J., Kiczales, G.: Overcoming the prevalent decomposition in legacy code. In: Workshop on Advanced Separation of Concerns, Int'l Conf. Software Engineering (ICSE). (2001) [213]
222. Hannemann, J., Kiczales, G.: Design pattern implementation in Java and AspectJ. In: Proc. ACM SIGPLAN Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), ACM Press (2002) 161–173 [212]
223. Hannemann, J., Murphy, G.C., Kiczales, G.: Role-based refactoring of crosscutting concerns. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM Press (2005) 135–146 [218]
224. Harrold, M.J.: Testing: a roadmap. In: The Future of Software Engineering (ICSE 2000), ACM Press (2000) 61–72 [199]
225. Haug, M., Olsen, E.W., Cuevas, G., Rementeria, S.: Managing the Change: Software Configuration and Change Management: v. 2. Springer-Verlag (2001) [292]
226. Havinga, W., Nagy, I., Bergmans, L.: An analysis of aspect composition problems. In: Proc. European Workshop on Aspects in Software. (August 2006) 1–8 [228]
227. Heckel, R., Wagner, A.: Ensuring consistency of conditional graph grammars: A constructive approach. Electronic Notes in Theoretical Computer Science **1** (1995) [149]
228. Henrard, J., Hick, J.M., Thiran, P., Hainaut, J.L.: Strategies for data reengineering. In: Proc. Working Conf. Reverse Engineering (WCRE), Washington, DC, USA, IEEE Computer Society Press (2002) 211–220 [110]
229. Henrard, J.: Program Understanding in Database Reverse Engineering. PhD thesis, University of Namur (2003) [129, 138]
230. Henry, S.M., Kafura, D.G.: Software structure metrics based on information flow. IEEE Trans. Software Engineering **7**(5) (1981) 510–518 [74]
231. Herraiz, I., Robles, G., Gonzalez-Barahona, J.M., Capiluppi, A., Ramil, J.F.: Comparison between SLOCs and number of files as size metrics for software evolution analysis. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR), Bari, Italy (21–24 March 2006) [7, 270, 274, 275]
232. Hick, J.M., Hainaut, J.L.: Database application evolution: A transformational approach. Data & Knowledge Engineering **59** (December 2006) 534–558 [116]
233. Highsmith, J., Fowler, M.: The agile manifesto. Software Development Magazine **9**(8) (2001) 29–30 [175]
234. Higo, Y., Ueda, Y., Kamiya, T., Kusumoto, S., Inoue, K.: On software maintenance process improvement based on code clone analysis. In: Proc. Int'l Conf. Product Focused Software Process Improvement. Volume 2559 of Lecture Notes in Computer Science., Springer-Verlag (2002) 185–197 [27]
235. Horwitz, S., Reps, T., Binkley, D.: Interprocedural slicing using dependence graphs. ACM Trans. Programming Languages and Systems **12**(1) (January 1990) 26–60 [132]
236. Humphrey, W.: Managing the Software Process. Addison-Wesley (1989) [198]
237. Hunt, J.: Extensible, Language Aware Differencing and Merging. Dissertation, University of Kaiserslautern (2001) [35]
238. IEEE. In: Standard 610.12-1990: Glossary of Software Engineering Terminology. Volume 1. IEEE Press (1999) [73, 301, 304]
239. IEEE. In: Standard IEEE Std 1219-1999 on Software Maintenance. Volume 2. IEEE Press (1999) [1, 7, 293, 301]

240. International Standards Organisation (ISO): International standard ISO/IEC 9126. information technology: Software product evaluation: Quality characteristics and guidelines for their use (1991) [193, 293, 304]
241. International Standards Organisation (ISO): Software life cycle processes. In: ISO 12207 Information Technology. ISO (1995) [293, 301]
242. International Standards Organisation (ISO): Standard 14764 on Software Engineering – Software Maintenance. ISO/IEC (1999) [3, 4, 7, 293, 301]
243. ITU: Recommendation Z.120: Message Sequence Chart (MSC). Ø Haugen (ed.), Geneva (1999) [260]
244. Ivkovic, I., Kontogiannis, K.: A framework for software architecture refactoring using model transformations and semantic annotations. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR), Washington, DC, USA, IEEE Computer Society Press (2006) 135–144 [167]
245. Jackson, A., Klein, J., Baudry, B., Clarke, S.: Testing aspect models. In: Model Driven Development and Model Driven Testing workshop at ECMDA. (2006) [260]
246. Jackson, J.E.: A Users Guide to Principal Components. John Wiley & Sons (2003) [76]
247. Jacobson, I., Ng, P.W.: Aspect-Oriented Software Development with Use Cases. Addison-Wesley (2004) [8]
248. Jacoby, R., Masuzawa, K.: Test coverage dependent software reliability estimation by the HGDmodel. In: Proc. Int'l Symp. Software Reliability Engineering (ISSRE). (1992) 193–204 [73]
249. Jahnke, J.H., Wadsack, J.: Varlet: Human-centered tool support for database reengineering. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (May 1999) [109]
250. Jakobac, V., Egyed, A., Medvidovic, N.: Improving system understanding via interactive, tailorable, source code analysis. In Cerioli, M., ed.: Proc. Int'l Conf. Fundamental Aspects of Software Engineering (FASE). Volume 3442 of Lecture Notes in Computer Science., Springer-Verlag (2005) 253–268 [164]
251. Jankowitz, H.T.: Detecting plagiarism in student Pascal programs. Computer Journal **1**(31) (1988) 1–8 [34]
252. Jazayeri, M.: On architectural stability and evolution. In: Proc. Reliable Software Technologies-Ada-Europe. Volume 2361 of Lecture Notes in Computer Science., Vienna, Austria, Springer-Verlag (2002) 397–420 [9]
253. Jazayeri, M., Ran, A., Van Der Linden, F.: Software Architecture for Product Families: Principles and Practice. Addison-Wesley (2000) [9]
254. Jeffries, R., Anderson, A., Hendrickson, C.: Extreme Programming Installed. Addison-Wesley (2000) [177, 179, 198]
255. JetBrains: IntelliJ IDEA. http://www.jetbrains.com/idea/ (2007) [165]
256. Jeusfeld, M.A., Johnen, U.A.: An executable meta model for re-engineering of database schemas. In: Proc. Conf. on the Entity-Relationship Approach, Manchester (December 1994) [109]
257. Jiang, L., Misherghi, G., Su, Z., Glondu, S.: Deckard: Scalable and accurate tree-based detection of code clones. In: Proc. Int'l Conf. Software Engineering (ICSE), ACM Press (2007) 96–105 [28]
258. Johnson, J.H.: Identifying redundancy in source code using fingerprints. In: Proc. Conf. Centre for Advanced Studies on Collaborative research (CASCON), IBM Press (1993) 171–183 [26]

259. Johnson, J.H.: Substring matching for clone detection and change tracking. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (1994) 120–126 [26]
260. Johnson, J.H.: Visualizing textual redundancy in legacy source. In: Proc. Conf. Centre for Advanced Studies on Collaborative research (CASCON), IBM Press (1994) 32–41 [31, 32]
261. Jörgensen, M.: Experience with the accuracy of software maintenance task effort prediction models. IEEE Trans. Software Engineering **21**(8) (1995) 674–681 [7]
262. JUnit: JUnit.
http://www.junit.org (2007) [175, 178, 193]
263. Kamiya, T., Kusumoto, S., Inoue, K.: CCFinder: A Multi-Linguistic Token-based Code Clone Detection System for Large Scale Source Code. IEEE Trans. Software Engineering **28**(7) (2002) 654–670 [27, 29, 30]
264. Kapser, C., Godfrey, M.: A taxonomy of clones in source code: The re-engineers most wanted list. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2003) [19, 20]
265. Kapser, C., Godfrey, M.W.: Toward a taxonomy of clones in source code: A case study. In: Proc. Int'l Workshop on Evolution of Large Scale Industrial Software Architectures (ELISA). (2003) 67–78 [19, 20]
266. Kapser, C., Godfrey, M.W.: "Clones considered harmful" considered harmful. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2006) 19–28 [20, 21]
267. Kapser, C.J., Godfrey, M.W.: Supporting the analysis of clones in software systems: Research articles. Software Maintenance and Evolution: Research and Practice **18**(2) (2006) 61–82 [19]
268. Karp, R.M.: Combinatorics, complexity, and randomness. Comm. ACM **29**(2) (February 1986) 98–109 [26]
269. Karp, R.M., Rabin, M..: Efficient randomized pattern-matching algorithms. IBM Journal Research and Development **31**(2) (March 1987) 249–260 [26]
270. Kazman, R., Carrière, S.J.: Playing detective: Reconstructing software architecture from available evidence. Automated Software Engineering **6**(2) (1999) 107–138 [99]
271. Kazman, R., Woods, S., Carrière, J.: Requirements for integrating software architecture and reengineering models: CORUM II. In: Proc. Working Conf. Reverse Engineering (WCRE), Washington, DC, USA, IEEE Computer Society Press (1998) 154–163 [5, 140]
272. Kellens, A., Mens, K., Tonella, P.: A survey of automated code-level aspect mining techniques. Trans. Aspect-Oriented Software Development (2007) To be published. [204, 213]
273. Kerievsky, J.: Refactoring to patterns. Addison-Wesley (2004) [101, 180, 292]
274. Khoshgoftaar, T.M., Allen, E.B., Goel, N., Nandi, A., McMullan, J.: Detection of software modules with high debug code churn in a very large legacy system. In: Proc. Int'l Symp. Software Reliability Engineering (ISSRE), Washington, DC, USA, IEEE Computer Society Press (1996) 364–371 [87]
275. Khoshgoftaar, T.M., Szabo, R.M.: Improving code churn predictions during the system test and maintenance phases. In: Proc. Int'l Conf. Software Maintenance (ICSM), Washington, DC, USA, IEEE Computer Society Press (1994) 58–67 [87]
276. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J.M., Irwin, J.: Aspect-oriented programming. In Aksit, M., Matsuoka, S., eds.: Proc. European Conf. Object-Oriented Programming (ECOOP). Volume 1241 of Lecture Notes in Computer Science., Springer-Verlag (1997) 220–242 [200, 234, 244]

277. Kim, M., Bergman, L., Lau, T., Notkin, D.: An ethnographic study of copy and paste programming practices in OOPL. In: Proc. Int'l Symp. Empirical Software Engineering, IEEE Computer Society Press (2004) 83–92 [19, 21]
278. Kim, M., Sazawal, V., Notkin, D., Murphy, G.C.: An empirical study of code clone genealogies. In: Proc. European Software Engineering Conf. and Foundations of Software Engineering (ESEC/FSE). (2005) 187–196 [23]
279. Kitchenham, B.A.: System evolution dynamics of VME/B. ICL Tech. J. (May 1982) 42–57 [265]
280. Kitchenham, B.A., Pfleeger, S.L., Hoaglin, D.C., Rosenberg, J.: Preliminary guidelines for empirical research in software engineering. IEEE Trans. Software Engineering **28**(8) (August 2002) 721– 734 [282]
281. Klein, J., Hélouêt, L., Jézéquel, J.M.: Semantic-based weaving of scenarios. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), New York, NY, USA, ACM Press (2006) 27–38 [260]
282. Kleinbaum, D.G., Kupper, L.L., Nizam, A., Muller, K.E.: Applied Regression Analysis and Multivariable Methods. 4th edn. Duxbury Press (2007) [75]
283. Kniesel, G.: Conditional transformation. http://roots.iai.uni-bonn.de/research/jtransformer/cts (2003) [258]
284. Kniesel, G.: Type-safe delegation for runtime component adaptation. In Guerraoui, R., ed.: Proc. European Conf. Object-Oriented Programming (ECOOP). Volume 1628 of Lecture Notes in Computer Science., Springer-Verlag (1999) 351–366 [9]
285. Kniesel, G., Koch, H.: Static composition of refactorings. Science of Computer Programming **52**(1-3) (2004) 9–51 [95]
286. Komondoor, R., Horwitz, S.: Using slicing to identify duplication in source code. In: Proc. Int'l Symp. Static Analysis. (July 2001) 40–56 [28, 30]
287. Komondoor, R., Horwitz, S.: Eliminating duplication in source code via procedure extraction. Technical report 1461, UW-Madison Dept. of Computer Sciences (December 2002) [24]
288. Kong, J., Zhang, K., Dong, J., Song, G.: A graph grammar approach to software architecture verification and transformation. Proc. Int'l Computer Software and Applications Conf. (COMPSAC) (2003) 492– [165]
289. Kontogiannis, K., DeMori, R., Bernstein, M., Galler, M., Merlo, E.: Pattern matching for design concept localization. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (July 1995) 96–103 [27]
290. Kontogiannis, K., Martin, J., Wong, K., Gregory, R., Müller, H., Mylopoulos, J.: Code migration through transformations: an experience report. In: Proc. Conf. Centre for Advanced Studies on Collaborative research (CASCON), IBM Press (1998) 13 [108]
291. Kontogiannis, K., Mori, R.D., Merlo, E., Galler, M., Bernstein, M.: Pattern matching for clone and concept detection. Automated Software Engineering **3**(1/2) (June 1996) 79–108 [15, 27]
292. Kontogiannis, K.: Program representation and behavioural matching for localizing similar code fragments. In: Proc. Conf. Centre for Advanced Studies on Collaborative research (CASCON), IBM Press (1993) 194–205 [302]
293. Koppen, C., Störzer, M.: PCDiff: Attacking the fragile pointcut problem. In Gybels, K., Hanenberg, S., Herrmann, S., Wloka, J., eds.: European Interactive Workshop on Aspects in Software (EIWAS). (September 2004) [228, 230]
294. Koschke, R., Falke, R., Frenzel, P.: Clone detection using abstract syntax suffix trees. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2006) 253–262 [28]

295. Koschke, R., Girard, J.F.: An intermediate representation for reverse engineering analyses. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (1998) 241–250 [295]
296. Koskinen, J.: Software maintenance costs. http://www.cs.jyu.fi/ koskinen/smcosts.htm (2003) [7]
297. Kramer, J., Magee, J.: The evolving philosophers problem: Dynamic change management. IEEE Trans. Software Engineering **16**(11) (November 1990) 1293–1306 [9]
298. Krinke, J.: Identifying Similar Code with Program Dependence Graphs. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2001) 301–309 [28, 29]
299. Kruchten, P.: The Rational Unified Process. An Introduction. Addison-Wesley (1998) [198]
300. Kulesza, U., Sant'Anna, C., Garcia, A., Coelho, R., von Staa, A., Lucena, C.: Quantifying the effects of aspect-oriented programming: A maintenance study. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2006) 223–233 [212]
301. Kung, D.C., Gao, J., Kung, C.H.: Testing Object-Oriented Software. IEEE Computer Society Press (1998) [174]
302. Laddad, R.: AOP@Work: AOP myths and realities – beyond hype and misunderstandings. Published as article in IBM's developerWorks series http://www.ibm.com/developerworks (February 2006) [204]
303. Laguë, B., Proulx, D., Merlo, E.M., Mayrand, J., Hudepohl, J.: Assessing the benefits of incorporating function clone detection in a development process. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (1997) 314–321 [15, 24, 25, 27]
304. Lämmel, R., Verhoef, C.: Semi-automatic Grammar Recovery. Software – Practice & Experience **31**(15) (December 2001) 1395–1438 [131]
305. Lämmel, R.: Towards generic refactoring. In: Proc. ACM SIGPLAN workshop on Rule-based programming (RULE), ACM Press: New York NY (2002) 15–28 [93]
306. Lämmel, R.: Coupled software transformations (ext. abstract). In: Proc. Int'l Workshop on Software Evolution Transformations (SETra). (Nov. 2004) [116]
307. Lanubile, F., Mallardo, T.: Finding function clones in web applications. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR). (2003) 379–386 [27]
308. Lanza, M.: The evolution matrix: Recovering software evolution using software visualization techniques. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), Vienna, Austria, ACM (September 2001) 37–42 [65, 66]
309. Lanza, M., Ducasse, S.: Polymetric views – a lightweight visual approach to reverse engineering. IEEE Trans. Software Engineering **29**(9) (September 2003) 782–795 [32, 47, 61]
310. Lanza, M., Ducasse, S., Gall, H., Pinzger, M.: Codecrawler – an information visualization tool for program comprehension. In: Proc. Int'l Conf. Software Engineering (ICSE), ACM Press (2005) 672–673 [48]
311. Lanza, M., Marinescu, R.: Object-Oriented Metrics in Practice. Springer-Verlag (2006) [103, 104]
312. Larman, C., Victor R. Basili: Iterative and incremental development: A brief history. IEEE Computer **36**(6) (June 2003) 47–56 [3]
313. Lauder, A., Kent, S.: More legacy system patterns. In Henderson, P., ed.: Systems engineering for business process change: new directions. Springer-Verlag, New York, NY, USA (2002) 225–240 [103]

314. Lawrence, M.: An examination of evolution dynamics. In: Proc. Int'l Conf. Software Engineering (ICSE), IEEE Computer Society Press (13–16 Sep 1982) 188–196 [265, 279]
315. Leclercq, M., Ozcan, A.E., Quema, V., Stefani, J.B.: Supporting heterogeneous architecture descriptions in an extensible toolset. Proc. Int'l Conf. Software Engineering (ICSE) (2007) 209–219 [236]
316. Lehman, M.M., Fernandez-Ramil, J.: Software evolution and feedback: Theory and practice. John Wiley & Sons, Chichester, U.K. (2006) 7–40 [266]
317. Lehman, M.M.: On understanding laws, evolution and conservation in the large program life cycle. Systems and Software **1**(3) (1980) 213–221 [2, 173]
318. Lehman, M.M.: Programs, life cycles, and laws of software evolution. Proc. IEEE **68**(9) (September 1980) 1060–1076 [2, 8]
319. Lehman, M.M.: Software's future: Managing evolution. IEEE Software **15**(1) (January/February 1998) 40–44 [173]
320. Lehman, M.M., Belady, L.A.: Program Evolution: Processes of Software Change. Apic Studies In Data Processing. Academic Press (1985) [VII, 2, 82, 205, 264, 265, 270, 276, 287, 291, 295, 299]
321. Lehman, M.M., Ramil, J.F., Wernick, P., Perry, D.E., Turski, W.M.: Metrics and laws of software evolution – the nineties view. In: Proc. IEEE Symp. Software Metrics, IEEE Computer Society Press (1997) 20–32 [173]
322. Lehman, M., Ramil, J.F., Kahen, G.: Evolution as a noun and evolution as a verb. In: Proc. Workshop on Software and Organisation Co-evolution (SOCE). (July 2000) [4, 269]
323. Lehman, M., Ramil, J.: An overview of some lessons learnt in FEAST. In: Proc. Workshop on Empirical Studies of Software Maintenance (WESS). (October 2002) [266]
324. Leitao, A.M.: Detection of redundant code using R2D2. In: Proc. Workshop Source Code Analysis and Manipulation (SCAM), IEEE Computer Society Press (2003) 183–192 [28]
325. Lethbridge, T., Tichelaar, S., Plödereder, E.: The Dagstuhl middle metamodel: A schema for reverse engineering. Electronic Notes in Theoretical Computer Science **94** (2004) 7–18 [163]
326. Li, P.L., Herbsleb, J.D., Shaw, M.: Finding predictors of field defects for open source software systems in commonly available data sources: A case study of OpenBSD. In: Proc. IEEE Symp. Software Metrics, IEEE Computer Society Press (2005) 32 [87]
327. Li, Z., Lu, S., Myagmar, S., Zhou, Y.: CP-Miner: A tool for finding copy-paste and related bugs in operating system code. In: Operating System Design and Implementation. (2004) 289–302 [28]
328. Li, Z., Lu, S., Myagmar, S., Zhou, Y.: Copy-paste and related bugs in large-scale software code. IEEE Trans. Software Engineering **32**(3) (March 2006) 176–192 [19, 23]
329. Lientz, B.P., Swanson, E.B.: Software maintenance management: a study of the maintenance of computer application software in 487 data processing organizations. Addison-Wesley (1980) [4, 7, 173]
330. Liu, Y., Stroulia, E., Erdogmus, H.: Understanding the open-source software development process: a case study with CVSChecker. In: Proc. Intl'l Conf. on Open Source Systems, NRC 47453 (2005) 154–161 [49, 50]
331. Lopez-Herrejon, R.E., Batory, D.S., Lengauer, C.: A disciplined approach to aspect composition. In: Proc. ACM SIGPLAN Symposium on Partial Evaluation and Program Manipulation (PEPM), New York, NY, USA, ACM Press (2006) 68–77 [229, 260]

332. Löwe, M., Korff, M., Wagner, A.: An algebraic framework for the transformation of attributed graphs. In Sleep, R., Plasmeijer, M., van Eekelen, M., eds.: Term Graph Rewriting: Theory and Practice. John Wiley & Sons (1993) 185–199 [148]
333. Lucia, A.D., Francese, R., Scanniello, G., Tortora, G., Vitiello, N.: A strategy and an Eclipse based environment for the migration of legacy systems to multi-tier web-based architectures. In: Proc. Int'l Conf. Software Maintenance (ICSM), Washington, DC, USA, IEEE Computer Society Press (2006) 438–447 [108]
334. Luckham, D.C., Kenney, J.J., Augustin, L.M., Vera, J., Bryan, D., Mann, W.: Specification and analysis of system architecture using Rapide. IEEE Trans. Software Engineering **21**(4) (1995) 336–355 [235]
335. Lynch, N.A., Tuttle, M.R.: An introduction to input/output automata. CWI Quarterly **2**(3) (1989) 219–246 [246]
336. Mackinnon, T., Freeman, S., Craig, P.: Endotesting: Unit testing with mock objects. In: Proc. Int'l Conf. eXtreme Programming and Flexible Processes in Software Engineering (XP). (2000) [180]
337. Madhavji, N.H.: Compare: a collusion detector for Pascal. Techniques et Sciences Informatiques **4**(6) (1985) 489–497 [34]
338. Madhavji, N.H., Ramil, J.F., Perry, D.E.: Software Evolution and Feedback: Theory and Practice. John Wiley & Sons (2006) [4, 7, 292, 299]
339. Magee, J.: Behavioral analysis of software architectures using ltsa. In: Proc. Int'l Conf. Software Engineering (ICSE), IEEE Computer Society Press (1999) 634–637 [235]
340. Magee, J., Kramer, J., Giannakopoulou, D.: Behaviour analysis of software architectures. In: Proc. IEEE/IFIP Working Conf. Software Architecture (WICSA), Deventer, The Netherlands, The Netherlands, Kluwer Academic Publishers (1999) 35–50 [235, 237]
341. Maier, M.W., Emery, D., Hilliard, R.: ANSI-IEEE 1471 and systems engineering. Systens Engineering **7**(3) (2004) 257–270 [243]
342. Malton, A.J.: The software migration barbell. In: ASERC Workshop on Software Architecture. (August 2001) [108]
343. Manber, U.: Finding similar files in a large file system. In: Proc. Winter Usenix Technical Conf. (1994) 1–10 [34]
344. Marchesi, M., Succi, G., Wells, D., Williams, L.: Extreme Programming Perspectives. Addison-Wesley (2003) [179]
345. Marcus, A., Maletic, J.: Identification of high-level concept clones in source code. In: Proc. Int'l Conf. Automated Software Engineering (ASE). (2001) 107–114 [26]
346. Marick, B.: The Craft of Software Testing. Prentice Hall (1995) [174, 192, 201]
347. Marin, M., Moonen, L., van Deursen, A.: A common framework for aspect mining based on crosscutting concern sorts. In Sim, S.E., Di Penta, M., eds.: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2006) 29–38 [212, 213]
348. Marin, M., van Deursen, A., Moonen, L.: Identifying aspects using fan-in analysis. In: Proc. Working Conf. Reverse Engineering (WCRE), Washington, DC, USA, IEEE Computer Society Press (2004) 132–141 [109]
349. Marks, E., Bell, M.: Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology. John Wiley & Sons (2006) [140]
350. Martin, J., Müller, H.A.: Strategies for migration from C to Java. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR). (2001) 200–209 [108]
351. Martin, R.C.: Agile Software Development: Principles, Patterns, and Practices. Prentice Hall (2002) [3, 8]
352. Maruyama, K.: Automated method-extraction refactoring by using block-based slicing. In: Proc. Symp. Software Reusability (SSR), ACM Press (2001) 31–40 [219]

353. Mayrand, J., Leblanc, C., Merlo, E.: Experiment on the automatic detection of function clones in a software system using metrics. In: Proc. Int'l Conf. Software Maintenance (ICSM). (1996) 244–253 [27, 29, 30]
354. McCabe, T.J.: A complexity measure. IEEE Trans. Software Engineering **2**(4) (1976) 308–320 [276, 278, 297]
355. McCoy, D., Natis, Y.: Service-oriented architecture: Mainstream straight ahead. Technical Report LE-19-7652, Gartner Research (April 2003) [139]
356. McCreight, E.: A space-economical suffix tree construction algorithm. Journal of the ACM **32**(2) (1976) 262–272 [27]
357. McEachen, N., Alexander, R.T.: Distributing classes with woven concerns: an exploration of potential fault scenarios. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM Press (2005) 192–200 [200]
358. Medvidovic, N., Taylor, R.N.: A classification and comparison framework for software architecture description languages. IEEE Trans. Software Engineering **26**(1) (2000) 70–93 [233, 235]
359. Meier, A.: Providing database migration tools – a practitioner's approach. In: Proc. Int'l Conf. Very Large Data Bases (VLDB), San Francisco, CA, USA, Morgan Kaufmann (1995) 635–641 [110]
360. Meier, A., Dippold, R., Mercerat, J., Muriset, A., Untersinger, J.C., Eckerlin, R., Ferrara, F.: Hierarchical to relational database migration. IEEE Software **11**(3) (1994) 21–27 [110]
361. Menhoudj, K., Ou-Halima, M.: Migrating data-oriented applications to a relational database management system. In: Proc. Int'l Workshop on Advances in Databases and Information Systems, Moscow (1996) [110]
362. Mens, K., Brichau, J., Gybels, K.: Managing the evolution of aspect-oriented software with model-based pointcuts. In Thomas, D., ed.: Proc. European Conf. Object-Oriented Programming (ECOOP). Volume 4067 of Lecture Notes in Computer Science., Springer-Verlag (2006) 501–525 [228, 230, 261]
363. Mens, K., Kellens, A., Pluquet, F., Wuyts, R.: Co-evolving code and design with intensional views – a case study. Computer Languages, Systems and Structures **32**(2–3) (July-October 2006) 140–156 Special Issue: Smalltalk. [10, 213, 216]
364. Mens, K., Tourwé, T.: Delving Source Code with Formal Concept Analysis. Computer Languages, Systems and Structures **31**(3) (2004) 183–197 [211]
365. Mens, T., Demeyer, S., Janssens, D.: Formalising behaviour preserving program transformations. In Corradini, A., Ehrig, H., Kreowski, H.J., Rozenberg, G., eds.: Proc. Int'l Conf. Graph Transformation (ICGT). Volume 2505 of Lecture Notes in Computer Science., Springer-Verlag (2002) 286–301 [94, 146, 148]
366. Mens, T., Mens, K., Tourwé, T.: Aspect-oriented software evolution. ERCIM News (58) (July 2004) 36–37 [245]
367. Mens, T., Taentzer, G., Müller, D.: Challenges in model refactoring. In: Proc. 1st Workshop on Refactoring Tools, University of Berlin (2007) [96]
368. Mens, T., Taentzer, G., Runge, O.: Analyzing refactoring dependencies using graph transformation. Software and Systems Modeling (2007) [152, 165]
369. Mens, T., Tourwé, T.: A survey of software refactoring. IEEE Trans. Software Engineering **30**(2) (February 2004) 126–162 [94]
370. Mens, T., Van Eetvelde, N., Demeyer, S., Janssens, D.: Formalizing refactorings with graph transformations. Software Maintenance and Evolution: Research and Practice **17**(4) (July/August 2005) 247–276 [94, 148, 159]

371. Mens, T., Wermelinger, M., Ducasse, S., Demeyer, S., Hirschfeld, R., Jazayeri, M.: Challenges in software evolution. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE). (2005) [4]
372. Meszaros, G.: XUnit Test Patterns: Refactoring Test Code. Addison-Wesley (2007) [184]
373. Missaoui, R., Godin, R., Sahraoui, H.: Migrating to an object-oriented database using semantic clustering and transformation rules. Data Knowledge Engineering **27**(1) (1998) 97–113 [110]
374. Monden, A., Nakae, D., Kamiya, T., Sato, S., Matsumoto, K.: Software quality analysis by code clones in industrial legacy software. In: Proc. IEEE Symp. Software Metrics. (2002) 87–94 [22]
375. Monteiro, M.P., Fernandes, J.M.: Object-to-aspect refactorings for feature extraction. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM Press (2004) [218, 219, 220, 222]
376. Monteiro, M.P., Fernandes, J.M.: Towards a catalog of aspect-oriented refactorings. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM Press (2005) 111–122 [218, 228, 229]
377. Moonen, L.: Generating robust parsers using island grammars. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (Oct. 2001) 13–22 [27]
378. Moonen, L.: Exploring Software Systems. PhD thesis, Faculty of Natural Sciences, Mathematics, and Computer Science, University of Amsterdam (December 2002) [177]
379. Moore, I.: Jester, JUnit test tester. In: Proc. Int'l Conf. eXtreme Programming and Flexible Processes in Software Engineering (XP). (2001) 84–87 [192]
380. Mossienko, M.: Automated Cobol to Java recycling. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR), Washington, DC, USA, IEEE Computer Society Press (2003) 40–50 [139]
381. Muccini, H., Dias, M., Richardson, D.J.: Software architecture-based regression testing. Systems and Software **79**(10) (2006) 1379–1396 [199]
382. Muller, A., Caron, O., Carré, B., Vanwormhoudt, G.: On some properties of parameterized model application. In: Proc. European Conf. Model-Driven Architectures: Foundations and Applications. (2005) 130–144 [259]
383. Müller, H.: Understanding software systems using reverse engineering technologies: Research and practice.
http://www.rigi.csc.uvic.ca/UVicRevTut/UVicRevTut.html (1996) ICSE-18 Tutorial. [177, 302]
384. Munson, J.C., Elbaum, S.G.: Code churn: A measure for estimating the impact of code change. In: Proc. Int'l Conf. Software Maintenance (ICSM), Washington, DC, USA, IEEE Computer Society Press (1998) 24–31 [82, 87]
385. Murphy, G.C., Notkin, D.: Reengineering with reflexion models: A case study. IEEE Computer **8** (1997) 29–36 [99]
386. Murphy, G.C., Notkin, D., Sullivan, K.: Software reflexion models: Bridging the gap between source and high-level models. SIGSOFT Software Engineering Notes **20**(4) (1995) 18–28 [99]
387. Murray, A., Lethbridge, T.C.: Presenting micro-theories of program comprehension in pattern form. In: Int'l Workshop on Program Comprehension (IWPC), Washington, DC, USA, IEEE Computer Society Press (2005) 45–54 [100]
388. Musa, J.D.: Software Reliability Engineering. McGraw-Hill (1998) [72]
389. Nagappan, N., Ball, T.: Use of relative code churn measures to predict system defect density. In: Proc. Int'l Conf. Software Engineering (ICSE). (2005) 284–292 [82, 83, 84]

390. Nagappan, N., Ball, T., Zeller, A.: Mining metrics to predict component failures. In: Proc. Int'l Conf. Software Engineering (ICSE), New York, NY, USA, ACM Press (2006) 452–461 [75, 76]
391. Naur, P., Randell, B.: Software Engineering. NATO, Scientific Affairs Division, Brussels (1969) [1, 304]
392. Nelson, E.: Estimating software reliability from test data. Microelectronics and Reliability **17**(1) (1978) 67–74 [73]
393. Newcomer, E., Lomow, G.: Understanding SOA with Web Services. Addison-Wesley Professional (2004) [8, 140]
394. Nickell, E., Smith, I.: Extreme programming and software clones. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2003) [36]
395. Niere, J., Schäfer, W., Wadsack, J.P., Wendehals, L., Welsh, J.: Towards pattern-based design recovery. In: Proc. Int'l Conf. Software Engineering (ICSE), IEEE Computer Society Press (May 2002) 338–348 [163]
396. Object Management Group: Meta object facility (MOF) specification. formal/2002-04-03 (April 2002) [301]
397. Object Management Group: Interface definition language, version 2.0. http://www.omg.org/gettingstarted/omg_idl.htm (2003) [236]
398. Object Management Group: Unified Modeling Language: Superstructure version 2.0. formal/2005-07-04 (August 2005) [148, 240, 308]
399. Object Management Group: UML 2 Object Constraint Language Specification. (March 2006) Version 2.0. [241]
400. O'Brien, L., Smith, D., Lewis, G.: Supporting migration to services using software architecture reconstruction. In: Proc. IEEE Int'l Workshop on Software Technology and Engineering Practice (STEP). (2005) 81–91 [109]
401. Ogawa, M.: Visualizing the Eclipse bug data. http://vis.cs.ucdavis.edu/∼ogawa/eclipse/ (January 2007) [70]
402. Ohlsson, M.C., von Mayrhauser, A., McGuire, B., Wohlin, C.: Code decay analysis of legacy software through successive releases. In: Proc. IEEE Aerospace Conf. (1999) 69–81 [87]
403. Oman, P.W., Lewis, T.G.: Milestones in Software Evolution. IEEE Computer Society Press (1990) [2]
404. Opdyke, W.F.: Refactoring: A Program Restructuring Aid in Designing Object-Oriented Application Frameworks. PhD thesis, University of Illinois at Urbana-Champaign (1992) [93, 94, 218, 223]
405. Opdyke, W.F., Johnson, R.E.: Creating abstract superclasses by refactoring. In: Proc. ACM Computer Science Conf., ACM Press (1993) 66–73 [93]
406. Oreizy, P., Medvidovic, N., Taylor, R.N.: Architecture-based runtime software evolution. In: Proc. Int'l Conf. Software Engineering (ICSE), Washington, DC, USA, IEEE Computer Society Press (1998) 177–186 [241]
407. Organization for the Advancement of Structured Information Standards: Reference Model for Service Oriented Architecture 1.0. OASIS (July 2006) [140]
408. Ostrand, T.J., Weyuker, E.J., Bell, R.M.: Where the bugs are. In: Proc. ACM SIGSOFT Int'l Symp. Software Testing and Analysis (ISSTA), New York, NY, USA, ACM Press (2004) 86–96 [82, 87]
409. Papakonstantinou, Y., Gupta, A., Garcia-Molina, H., Ullman, J.: A query translation scheme for rapid implementation of wrappers. In: Proc. Int'l Conf. Declarative and Object-oriented Databases. (1995) [123]
410. Parnas, D.L.: Software aging. In: Proc. Int'l Conf. Software Engineering (ICSE), IEEE Computer Society Press (1994) 279–287 Sorento, Italy, May 16-21. [2]

411. Paulson, J.W., Succi, G., Eberlein, A.: An empirical study of open-source and closed-source software products. IEEE Trans. Software Engineering **30**(4) (2004) 246–256 [271, 273, 283]
412. Perez, J., Navarro, E., Letelier, P., Ramos, I.: A modelling proposal for aspect-oriented software architectures. In: Proc. IEEE Int'l Symp. and Workshop on Engineering of Computer Based Systems (ECBS), Washington, DC, USA, IEEE Computer Society Press (2006) 32–41 [244]
413. Pessemier, N., Seinturier, L., Coupaye, T., Duchien, L.: A model for developing component-based and aspect-oriented systems. In: Proc. Int'l Symp. Software Composition (SC). Volume 4089 of Lecture Notes in Computer Science., Vienna, Austria, Springer-Verlag (mar 2006) 259–273 [244]
414. Pfleeger, S.L.: Software Engineering: Theory and Practice. Prentice Hall (1998) [266]
415. Pickin, S., Jard, C., Jéron, T., Jézéquel, J.M., Traon, Y.L.: Test synthesis from UML models of distributed software. IEEE Trans. Software Engineering **33**(4) (2007) 252–269 [199]
416. Pigoski, T.M.: Practical Software Maintenance: Best Practices for Managing your Software Investment. John Wiley & Sons (1997) [7]
417. Pinzger, M., Gall, H., Fischer, M., Lanza, M.: Visualizing multiple evolution metrics. In: Proc. ACM Symp. Software Visualization, St. Louis, Missouri, ACM Press (2005) 67–75 [60]
418. Pirzada, S.S.: A Statistical Examination of the Evolution of the Unix System. PhD thesis, Department of Computing, Imperial College, London (1988) [270]
419. Plump, D.: Hypergraph rewriting: Critical pairs and undecidability of confluence. In Sleep, M.R., Plasmeijer, M.J., van Eekelen, M.C., eds.: Term Graph Rewriting: Theory and Practice. John Wiley & Sons (1993) 201–213 [150]
420. Prechelt, L., Malpohl, G., Philippsen, M.: JPlag: Finding plagiarisms among a set of programs. Technical report, University of Karlsruhe, Department of Informatics (2000) [29, 34]
421. Putnam, L.: A general empirical solution to the macrosoftware sizing and estimating problem. IEEE Trans. Software Engineering **4**(4) (1978) 345–61 [195]
422. Quintero, C.E.C., Rodríguez, M.P.R., de la Fuente, P., Barrio-Solórzano, M.: Architectural aspects of architectural aspects. In: Proc. European Workshop on Software Architecture (EWSA). (2005) 247–262 [245]
423. Rahm, E., Do, H.: Data cleaning: Problems and current approaches. Data Engineering Bulletin **23** (2000) 3–13 [121]
424. Rajlich, V.: A model for change propagation based on graph rewriting. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (1997) 84–91 [6]
425. Rajlich, V., Gosavi, P.: Incremental change in object-oriented programming. IEEE Software **21**(4) (2004) 62–69 [6]
426. Ramalingam, G., Komondoor, R., Field, J., Sinha, S.: Semantics-based reverse engineering of object-oriented data models. In: Proc. Int'l Conf. Software Engineering (ICSE), New York, NY, USA, ACM Press (2006) 192–201 [163]
427. Ramil, J.F., Lehman, M.M.: Metrics of software evolution as effort predictors – a case study. In: Proc. Int'l Conf. Software Maintenance (ICSM). (October 2000) 163–172 [7]
428. Ramil, J., Capiluppi, A.: Metric-based studies of open source software evolution. Presentation Charts, Research Seminar, University of Leicester, UK (November 2004) [277]
429. Ramil, J.F., Smith, N.: Qualitative simulation of models of software evolution. Software Process: Improvement and Practice **7**(3–4) (September–December 2002) 95–112 [279, 280]

430. Randell, B.: System structure for software fault tolerance. IEEE Trans. Software Engineering **1**(2) (1975) 221–232 [80]
431. Rashid, A., Sawyer, P., Moreira, A.M.D., Araújo, J.: Early aspects: A model for aspect-oriented requirements engineering. In: Proc. Joint Int'l Conf. Requirements Engineering (RE), IEEE Computer Society Press (2002) 199–202 [213]
432. Ratzinger, J., Fischer, M., Gall, H.: Evolens: Lens-view visualizations of evolution data. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), Lisbon, Portugal, IEEE Computer Society Press (September 2005) 103–112 [59]
433. Ratzinger, J., Fischer, M., Gall, H.: Improving evolvability through refactoring. In: Proc. Int'l Workshop on Mining Software Repositories (MSR), New York, NY, USA, ACM Press (2005) 1–5 [59]
434. Raymond, E.S.: The cathedral and the bazaar: musings on Linux and open source by an accidental revolutionary. Revised edn. O'Reilly & Associates, Inc. (2001) [269]
435. Reddy, Y.R., Ghosh, S., France, R.B., Straw, G., Bieman, J.M., McEachen, N., Song, E., Georg, G.: Directives for composing aspect-oriented design class models. Trans. Aspect-Oriented Software Development (2006) 75–105 [259, 260]
436. Reiss, S.P.: Constraining software evolution. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2002) 162–171 [174]
437. Rieger, M., Ducasse, S., Lanza, M.: Insights into system-wide code duplication. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2004) 100–109 [32]
438. Rieger, M.: Effective Clone Detection Without Language Barriers. Dissertation, University of Bern, Switzerland (2005) [30]
439. Riel, A.J.: Object-Oriented Design Heuristics. Addison-Wesley, Boston MA (April 1996) [57, 102]
440. Roberts, D., Brant, J., Johnson, R.E.: A refactoring tool for Smalltalk. Theory and Practice of Object Systems **3**(4) (1997) 253–263 [93]
441. Roberts, D.B.: Practical Analysis for Refactoring. PhD thesis, University of Illinois at Urbana-Champaign (1999) [94, 95]
442. Robillard, M.P., Murphy, G.C.: Concern graphs: Finding and describing concerns using structural program dependencies. In: Proc. Int'l Conf. Software Engineering (ICSE), ACM Press (2002) 406–416 [213, 216]
443. Robles, G., Amor, J.J., Gonzalez-Barahona, J.M., Herraiz, I.: Evolution and growth in large libre software projects. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), Lisbon, IEEE Computer Society Press (September 2005) 165–174 [270]
444. Rosenblum, D.S., Weyuker, E.J.: Using coverage information to predict the cost-effectiveness of regression testing strategies. IEEE Trans. Software Engineering **23**(3) (1997) 146–156 [7, 200]
445. Rothermel, G., Harrold, M.J.: Empirical studies of a safe regression test selection technique. IEEE Trans. Software Engineering **24**(6) (1998) 401–419 [200, 202]
446. Royce, W.W.: Managing the development of large software systems: concepts and techniques. In: Proc. IEEE WESTCON, IEEE Computer Society Press (August 1970) Reprinted in Proc. Int'l Conf. Software Engineering (ICSE) 1989, ACM Press, pp. 328-338. [1]
447. Runeson, P., Andersson, C., Höst, M.: Test processes in software product evolution: a qualitative survey on the state of practice. Software Maintenance and Evolution: Research and Practice **15**(1) (2003) 41–59 [174]
448. Saff, D., Ernst, M.D.: An experimental evaluation of continuous testing during development. In: Proc. ACM/SIGSOFT Int'l Symp. Software Testing and Analysis (ISSTA), ACM (2004) 76–85 [201]

449. Sahraoui, H.A., Lounis, H., Melo, W., Mili, H.: A concept formation based approach to object identification in procedural code. Automated Software Engineering **6**(4) (1999) 387–410 [109]
450. Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., Lakhani, K.: Understanding free/open source software development processes. Software Process: Improvement and Practice **11**(2) (March/April 2006) 95–105 [286, 287]
451. Schärli, N., Ducasse, S., Nierstrasz, O., Black, A.: Traits: Composable units of behavior. In: Proc. European Conf. Object-Oriented Programming (ECOOP). Volume 2743 of Lecture Notes in Computer Science., Springer-Verlag (July 2003) 248–274 [9]
452. Schleimer, S., Wilkerson, D.S., Aiken, A.: Winnowing: local algorithms for document fingerprinting. In: Proc. ACM SIGMOD Conf. (2003) 76–85 [29, 34]
453. Schmidt, D.C.: Model-driven engineering. IEEE Computer **39**(2) (2006) 25–31 [199]
454. Schneider, A.: JUnit best practices. Java World **12** (2000) [184]
455. Schröter, A., Zimmermann, T., Zeller, A.: Predicting component failures at design time. In: Proc. Int'l Symp. Empirical Software Engineering. (September 2006) 18–27 [70, 77, 79]
456. Schürr, A., Winter, A., Zündorf, A.: The PROGRES approach: Language and environment. In Ehrig, H., Engels, G., Kreowski, H.J., Rozenberg, G., eds.: Handbook of Graph Grammars and Computing by Graph Transformation: Applications, Languages, and Tools. Volume 3. World Scientific (1999) 487–550 [150]
457. Seacord, R.C., Plakosh, D., Lewis, G.A.: Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices. 1st edn. Addison-Wesley Professional (2003) [7, 292]
458. Serrano, M.A., Carver, D.L., de Oca, C.M.: Reengineering legacy systems for distributed environments. Systems and Software **64**(1) (2002) 37–55 [109]
459. Shaw, M., Garlan, D.: Software Architecture – Perspectives on an Emerging Discipline. Prentice Hall, Upper Saddle River, NJ, USA (1996) [297]
460. Shepherd, D., Pollock, L., Tourwé, T.: Using language clues to discover crosscutting concerns. In: Proc. workshop on Modeling and Analysis of Concerns in Software (MACS), New York, NY, USA, ACM Press (2005) 1–6 [211]
461. Smith, N., Capiluppi, A., Ramil, J.F.: A study of open source software evolution data using qualitative simulation. Software Process: Improvement and Practice **10**(3) (July/September 2005) 287–300 [8, 279, 280]
462. Smith, N., Capiluppi, A., Ramil, J.F.: Agent-based simulation of open source evolution. Software Process: Improvement and Practice **11**(4) (July/August 2006) 423–434 [279, 280, 281, 287]
463. Sneed, H., Sneed, S.: Creating web services from legacy host programs. In: Proc. Workshop on Website Evolution (WSE), Los Alamitos, CA, USA, IEEE Computer Society Press (2003) 59–65 [139]
464. Sneed, H.M.: Encapsulation of legacy software: A technique for reusing legacy software components. Annals of Software Engineering **9**(1-4) (2000) 293–313 [122]
465. Sneed, H.M.: Integrating legacy software into a service oriented architecture. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR), Los Alamitos, CA, USA, IEEE Computer Society Press (2006) 3–14 [109, 168]
466. Sneed, H.: Estimating the costs of software maintenance tasks. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (1995) 168–181 [7]
467. SOA, S.A.: Simple blog about service oriented architecture (SOA), its tooling and delivery or realisation.
http://soa-testing.blogspot.com (January 20 2007) [200]

468. Sommerville, I.: Software Engineering. 6th edn. Addison-Wesley (2001) [266]
469. Sourceforge: Emma.
http://emma.sourcefourge.net (January 20 2007) [192]
470. Sourceforge: Jester.
http://jester.sourcefourge.net (January 20 2007) [192]
471. Spanoudakis, G., Zisman, A.: Inconsistency management in software engineering: Survey and open research issues. In: Handbook of Software Engineering and Knowledge Engineering. World scientific (2001) 329–380 [10, 300]
472. Sprott, D., Wilkes, L.: Understanding service-oriented architecture. CBDI Forum (January 2004) [140]
473. Srivastava, A., Thiagarajan, J., Schertz, C.: Efficient integration testing using dependency analysis. Technical Report MSR-TR-2005-94, Microsoft Research (2005) [79]
474. Stahl, T., Völter, M.: Model Driven Software Development: Technology, Engineering, Management. John Wiley & Sons (2006) [8]
475. Stevens, P., Pooley, R.: System reengineering patterns. In: Proc. Foundations of Software Engineering Conf. (FSE), ACM-SIGSOFT (1998) 17–23 [98, 102]
476. Stewart, K.J., Darcy, D.P., Daniel, S.L.: Opportunities and challenges applying functional data analysis to the study of open source software evolution. Statistical Science **21**(2) (September 06 2006) 167–178 [272, 274]
477. Stoerzer, M., Graf, J.: Using pointcut delta analysis to support evolution of aspect-oriented software. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2005) 653–656 [228, 230]
478. Stroulia, E., El-Ramly, M., Iglinski, P., Sorenson, P.: User interface reverse engineering in support of interface migration to the web. Automated Software Engineering **10**(3) (2003) 271–301 [108]
479. Subramanyam, R., Krishnan, M.: Empirical analysis of CK metrics for object-oriented design complexity: Implications for software defects. IEEE Trans. Software Engineering **29**(4) (2003) 297–310 [73, 74]
480. Subversion: Subversion.
http://subversion.tigris.org (2006) [40]
481. Succi, G., Paulson, J.W., Eberlein, A.: Preliminary results from an empirical study on the growth of open source and commercial software products. In: Proc. Int'l Workshop on Economics-Driven Software Engineering Research, Toronto, Canada (14–15 May 2001) [8]
482. Sullivan, K., Griswold, W., Song, Y., Chai, Y., Shonle, M., Tewari, N., Rajan, H.: On the criteria to be used in decomposing systems into aspects. In: Proc. European Software Engineering Conf. and Foundations of Software Engineering (ESEC/FSE), ACM Press (2005) [228, 230]
483. Sun Microsystems: The SOA platform guide: Evaluate, extend, embrace. White Paper (February 2006) [140]
484. SWI-Prolog: Swi-prolog.
http://www.swiprolog.org/ (2006) [258]
485. Synytskyy, N., Cordy, J.R., Dean, T.: Resolution of static clones in dynamic web pages. In: Proc. Workshop on Website Evolution (WSE). (2003) 49–56 [27]
486. Szyperski, C.: Component Software: Beyond Object-Oriented Programming. ACM Addison-Wesley (1998) [9]
487. Taentzer, G.: AGG.
http://tfs.cs.tu-berlin.de/agg/index.html (2007) [150, 258]
488. Taentzer, G.: Tiger EMF Transformation.
http://tfs.cs.tu-berlin.de/emftrans (2007) [158]

489. Tairas, R., Gray, J., Baxter, I.: Visualization of clone detection results. In: Proc. OOPSLA workshop on Eclipse technology eXchange. (2006) 50–54 [33, 34]
490. Tarr, P., Ossher, H., Harrison, W., Sutton, S.M.: N degrees of separation: multi-dimensional separation of concerns. In: Proc. Int'l Conf. Software Engineering (ICSE), IEEE Computer Society Press (1999) 107–119 [203, 216, 234, 299]
491. Taylor, R.N., Medvidovic, N., Anderson, K.M., Whitehead, E.J., Robbins, J.E., Nies, K.A., Oreizy, P., Dubrow, D.L.: A component- and message-based architectural style for GUI software. IEEE Trans. Software Engineering **22**(6) (1996) 390–406 [236]
492. Tekinerdogan, B., Aksit, M.: Deriving design aspects from canonical models. In Demeyer, S., Bosch, J., eds.: Workshop Reader of the 12th European Conf. Object-Oriented Programming (ECOOP). Lecture Notes in Computer Science, Springer-Verlag (1998) 410–413 [213]
493. Terekhov, A.A., Verhoef, C.: The realities of language conversions. IEEE Software **17**(6) (2000) 111–124 [108]
494. The Eclipse Foundation: Eclipse. http://www.eclipse.org/ (2007) [165]
495. The Eclipse Foundation: Eclipse modeling framework. http://www.eclipse.org/emf/ (2007) [167]
496. The JBoss Community: Drools 3.0. http://labs.jboss.com/jbossrules/ (2007) [258]
497. Thiran, P., Hainaut, J.L., Houben, G.J., Benslimane, D.: Wrapper-based evolution of legacy information systems. ACM Trans. Software Engineering and Methodology **15**(4) (October 2006) 329–359 [123]
498. Tibermacine, C., Fleurquin, R., Sadou, S.: Preserving architectural choices throughout the component-based software development process. In: Proc. IEEE/IFIP Working Conf. Software Architecture (WICSA). (2005) 121–130 [241]
499. Tichelaar, S.: Modeling Object-Oriented Software for Reverse Engineering and Refactoring. PhD thesis, University of Bern (2001) [93, 94]
500. Tilley, S.R., Smith, D.B.: Perspectives on legacy system reengineering. Technical report, Software Engineering Institute, Carnegie Mellon University (1995) [107]
501. Tip, F., Kiezun, A., Baumer, D.: Refactoring for generalization using type constraints. In: Proc. ACM SIGPLAN Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), New York, NY, USA, ACM Press (2003) 13–26 [94]
502. Tip, F., Snelting, G., Johnson, R., eds. In Tip, F., Snelting, G., Johnson, R., eds.: Schloss Dagstuhl: Program Analysis for Object-Oriented Evolution. (February 2003) http://www.dagstuhl.de/de/programm/kalender/semhp/?semid=-2003091. [93]
503. Tonella, P., Ceccato, M.: Aspect mining through the formal concept analysis of execution traces. In: Proc. Working Conf. Reverse Engineering (WCRE), Washington, DC, USA, IEEE Computer Society Press (2004) 112–121 [109]
504. Toomim, M., Begel, A., Graham, S.: Managing duplicated code with linked editing. In: Proc. IEEE Symp. Visual Languages: Human Centric Computing, IEEE Computer Society Press (2004) 173–180 [26]
505. Tourwé, T., Brichau, J., Gybels, K.: On the existence of the AOSD-evolution paradox. In Bergmans, L., Brichau, J., Tarr, P., Ernst, E., eds.: SPLAT: Software engineering Properties of Languages for Aspect Technologies. (March 2003) [226, 245, 261]
506. Tourwé, T., Mens, K.: Mining aspectual views using formal concept analysis. In: Proc. Workshop Source Code Analysis and Manipulation (SCAM), IEEE Computer Society Press (September 2004) 97–106 [109]

507. Tsai, W.T., Paul, R., Song, W., Cao, Z.: Coyote: An XML-based framework for web services testing. In: Proc. IEEE Int'l Symp. High Assurance Systems Engineering (HASE), IEEE Computer Society Press (2002) 173–174 [200]
508. Tu, Q.: On navigation and analysis of software architecture evolution. Master's thesis, University of Waterloo (1992) [34]
509. Tu, Q., Godfrey, M.W.: An integrated approach for studying architectural evolution. In: Int'l Workshop on Program Comprehension (IWPC), IEEE Computer Society Press (June 2002) 127–136 [35]
510. Cubranic, D., Murphy, G.C., Singer, J., Booth, K.S.: Learning from project history: a case study for software development. In: Proc. ACM Conf. on Computer supported cooperative work (CSCW), New York, NY, USA, ACM Press (2004) 82–91 [43]
511. Cubranic, D., Murphy, G.C., Singer, J., Booth, K.S.: Hipikat: A project memory for software development. IEEE Trans. Software Engineering **31**(6) (2005) 446–465 [43, 44]
512. Ueda, Y., Kamiya, T., Kusumoto, S., Inoue, K.: Gemini: Maintenance support environment based on code clone analysis. In: Proc. IEEE Symp. Software Metrics, IEEE Computer Society Press (2002) 67–76 [31]
513. van den Brand, M., Heering, J., Klint, P., Olivier, P.: Compiling language definitions: the ASF+SDF compiler. ACM Trans. Programming Languages and Systems **24**(4) (July 2002) 334–368 [164]
514. van den Brand, M., Klint, P.: ASF+SDF Meta-Environment User Manual. (2005) [129]
515. van den Brand, M., van Deursen, A., Heering, J., de Jong, H., de Jonge, M.T.K., Klint, P., Moonen, L., Olivier, P., Scheerder, J., Vinju, J., Visser, E., Visser, J.: The ASF+SDF Meta-Environment: A component-based language development environment. In Wilhelm, R., ed.: Proc. Int'l Conf. Compiler Construction (CC). Volume 2027 of Lecture Notes in Computer Science., Springer-Verlag (2001) 365–370 [129]
516. van Deursen, A.: Program comprehension risks and benefits in extreme programming. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2001) 176–185 [174]
517. van Deursen, A., Kuipers, T.: Identifying objects using cluster and concept analysis. In: Proc. Int'l Conf. Software Engineering (ICSE), Los Alamitos, CA, USA, IEEE Computer Society Press (1999) 246–255 [109]
518. van Deursen, A., Moonen, L., van den Bergh, A., Kok, G.: Refactoring test code. In Marchesi, M., ed.: Proc. Int'l Conf. eXtreme Programming and Flexible Processes in Software Engineering (XP). (2001) [183, 186, 191]
519. Van Eetvelde, N., Janssens, D.: Extending graph transformation for refactoring. In: Proc. Int'l Conf. Graph Transformation (ICGT). Volume 3256 of Lecture Notes In Computer Science., Springer-Verlag (2004) 399–415 [94]
520. van Glabbeek, R.: The linear time – branching time spectrum I. The semantics of concrete, sequential processes. In J.A. Bergstra, A.P..S.S., ed.: Handbook of Process Algebra, Elsevier Science (2001) 3–99 [256]
521. Van Rompaey, B., Du Bois, B., Demeyer, S.: Characterizing the relative significance of a test smell. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2006) 391–400 [184]
522. Van Rysselberghe, F.: Lessons Learned Studying Historic Change Operations. PhD thesis, University of Antwerp (September 2007) [95]
523. van Rysselberghe F., M.R., Demeyer, S.: Detecting move operations in versioning information. In: Proc. European Conf. Software Maintenance and Reengineering (CSMR), IEEE Computer Society Press (2006) 271–278 [95]

524. Van Rysselberghe, F., Demeyer, S.: Reconstruction of successful software evolution using clone detection. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE). (2003) 126–130 [35]
525. Van Rysselberghe, F., Demeyer, S.: Evaluating clone detection techniques from a refactoring perspective. In: Proc. Int'l Conf. Automated Software Engineering (ASE). (2004) [30]
526. Vestal, S.: Fixed-priority sensitivity analysis for linear compute time models. IEEE Trans. Software Engineering **20**(4) (1994) [239]
527. Visaggio, G.: Ageing of a data-intensive legacy system: symptoms and remedies. Software Maintenance and Evolution: Research and Practice **13**(5) (2001) 281–308 [110]
528. Voinea, L., Telea, A.: CVSgrab: Mining the history of large software projects. In: Eurographics / IEEE VGTC Symp. Visualization (EuroVis), Lisbon, Portugal, IEEE VGTC,EG,, Eurographics (2006) 187–194 [51]
529. Voinea, L., Telea, A., van Wijk, J.J.: CVSscan: visualization of code evolution. In: Proc. ACM Symp. Software Visualization, New York, NY, USA, ACM Press (2005) 47–56 [51]
530. von Mayrhauser, A., Vans, A.M.: Program comprehension during software maintenance and evolution. IEEE Computer **28**(8) (August 1995) 44–55 [177]
531. Wahler, V., Seipel, D., von Gudenberg, J.W., Fischer, G.: Clone detection in source code by frequent itemset techniques. In: Proc. Workshop Source Code Analysis and Manipulation (SCAM). (2004) 128–135 [28]
532. Walenstein, A., Jyoti, N., Li, J., Yang, Y., Lakhotia, A.: Problems creating task-relevant clone detection reference data. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2003) 285–294 [17]
533. Walenstein, A., Lakhotia, A., Koschke, R.: 2nd int'l workshop on detection of software clones: workshop report. SIGSOFT Software Engineering Notes **29**(2) (2004) 1–5 [36]
534. Warren, I.: The Renaissance of Legacy Systems: Method Support for Software-System Evolution. Springer-Verlag, Secaucus, NJ, USA (1999) [109]
535. Waters, R.C.: Program translation via abstraction and reimplementation. IEEE Trans. Software Engineering **14**(8) (1988) 1207–1228 [108]
536. Wattenberg, M.: Map of the market. http://www.bewitched.com/ (1998) [70]
537. Wattenberg, M.: Arc diagrams: Visualizing structure in strings. In: Proc. InfoVis, IEEE Computer Society Press (2002) 10–17 [33]
538. Weiser, M.: Program slicing. IEEE Trans. Software Engineering **10**(4) (July 1984) 352–357 [138, 218]
539. Weyuker, E.J.: Component-based software engineering: Putting the pieces together. Addison-Wesley (2001) 499–512 [200]
540. Whittaker, J.: Markov Chain Techniques for Software Testing and Reliability Analysis. PhD thesis, Department of Computer Science, University of Tenn (1992) [73]
541. Wiederhold, G.: Modeling and system maintenance. In: Proc. Int'l Conf. Object-Oriented and Entity-Relationship Modeling, Berlin (1995) [107]
542. Wiki, C.: Refactor broken unit tests. http://c2.com/cgi/wiki?RefactorBrokenUnitTests (January 20 2007) [184]
543. Wilkes, L., Veryard, R.: Service-oriented architecture: Considerations for agile systems. CBDI Forum (April 2004) [140, 143]
544. World Wide Web Consortium (W3C): Web services glossary. http://www.w3.org/TR/ws-gloss/ (February 2004) [305]

545. Wu, B., Lawless, D., Bisbal, J., Grimson, J., Wad, V., O'Sullivan, D., Richardson, R.: Legacy system migration: A legacy data migration engine. In Czechoslovak Computer Experts, ed.: Proc. Int'l Database Conf. (DATASEM), Brno, Czech Republic (1997) 129–138 [111]
546. Wu, B., Lawless, D., Bisbal, J., Richardson, R., Grimson, J., Wade, V., O'Sullivan, D.: The butterfly methodology: A gateway-free approach for migrating legacy information systems. In: Proc. IEEE Conf. Engineering of Complex Computer Systems, Italy (September 1997) [107]
547. Wu, J.: Open Source Software Evolution and Its Dynamics. PhD thesis, University of Waterloo, Ontario, Canada (2006) [275]
548. Wu, J., Spitzer, C.W., Hassan, A.E., Holt, R.C.: Evolution spectrographs: Visualizing punctuated change in software evolution. In: Proc. Int'l Workshop on Principles of Software Evolution (IWPSE), Kyoto, Japan, IEEE Computer Society Press (September 2004) 57–66 [66, 275, 276]
549. Xie, T., Zhao, J.: A framework and tool supports for generating test inputs of AspectJ programs. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD). (2006) 190–201 [223]
550. Xie, Y., Engler, D.: Using redundancies to find errors. In: Proc. Foundations of Software Engineering Conf. (FSE), ACM Press (2002) 51–60 [16]
551. Xie, Y., Engler, D.: Using redundancies to find errors. IEEE Trans. Software Engineering **29**(10) (October 2003) 915–928 [16]
552. Xing, Z., Stroulia, E.: Refactoring detection based on UMLDiff change-facts queries. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2003) 263–274 [35]
553. Xing, Z., Stroulia, E.: Analyzing the evolutionary history of the logical design of object-oriented software. IEEE Trans. Software Engineering **31**(10) (October 2005) 850–868 [35]
554. Xing, Z., Stroulia, E.: Refactoring practice: How it is and how it should be supported – an Eclipse case study. In: Proc. Int'l Conf. Software Maintenance (ICSM), IEEE Computer Society Press (2006) 458–468 [95]
555. Xu, D., Xu, W.: State-based incremental testing of aspect-oriented programs. In: Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD), ACM Press (2006) 180–189 [223]
556. Yan, H., Garlan, D., Schmerl, B., Aldrich, J., Kazman, R.: Discotect: A system for discovering architectures from running systems. In: Proc. Int'l Conf. Software Engineering (ICSE), Edinburgh, Scotland (23-28 May 2004) [235, 236]
557. Yang, W.: Identifying syntactic differences between two programs. Software – Practice and Experience **21**(7) (July 1991) 739–755 [28]
558. Yasumatsu, K., Doi, N.: SPiCE: A system for translating Smalltalk programs into a C environment. IEEE Trans. Software Engineering **21**(11) (1995) 902–912 [108]
559. Yau, S.S., Colofello, J.S., MacGregor, T.: Ripple effect analysis of software maintenance. In: Proc. Int'l Computer Software and Applications Conf. (COMPSAC), IEEE Computer Society Press (1978) 60–65 [2, 6]
560. Yeh, A.S., Harris, D.R., Reubenstein, H.B.: Recovering abstract data types and object instances from a conventional procedural language. In: Proc. Working Conf. Reverse Engineering (WCRE), Washington, DC, USA, IEEE Computer Society (1995) 227 [109]
561. Ying, A.T.T., Murphy, G.C., Ng, R., Chu-Carroll, M.C.: Predicting source code changes by mining change history. IEEE Trans. Software Engineering **30**(9) (September 2004) 574–586 [59]

562. Zaidman, A., Van Rompaey, B., Demeyer, S., Van Deursen, A.: Mining Software Repositories to Study Co-Evolution of Production and Test Code. Proceedings of the 1st International Conference on Software Testing, Verification, and Validation (ICST), IEEE Computer Society (2008) [201]
563. Zhang, C., Jacobsen, H.: Extended aspect mining tool. http://www.eecg.utoronto.ca/∼czhang/amtex (August 2002) [213]
564. Zhang, C., Jacobsen, H.A.: PRISM is research in aspect mining. In: Proc. ACM SIGPLAN Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA), ACM Press (2004) 20–21 [213]
565. Zhang, Z., Yang, H., Chu, W.C.: Extracting reusable object-oriented legacy code segments with combined formal concept analysis and slicing techniques for service integration. In: Proc. Int'l Conf. Software Quality (QSIC), Washington, DC, USA, IEEE Computer Society Press (2006) 385–392 [168]
566. Zimmermann, T., Weißgerber, P.: Preprocessing CVS data for fine-grained analysis. In: Proc. Int'l Workshop on Mining Software Repositories (MSR), Los Alamitos CA, IEEE Computer Society Press (2004) 2–6 [40, 53]
567. Zimmermann, T., Weißgerber, P., Diehl, S., Zeller, A.: Mining version histories to guide software changes. In: Proc. Int'l Conf. Software Engineering (ICSE), Los Alamitos CA, IEEE Computer Society Press (2004) 563–572 [59]
568. Zou, L., Godfrey, M.: Detecting merging and splitting using origin analysis. In: Proc. Working Conf. Reverse Engineering (WCRE), IEEE Computer Society Press (2003) 146–154 [16, 35]
569. Zou, Y., Kontogiannis, K.: A framework for migrating procedural code to object-oriented platforms. In: Proc. IEEE Asia-Pacific Software Engineering Conf. (APSEC), Los Alamitos, CA, USA, IEEE Computer Society Press (2001) 408–418 [109]
570. Zowghi, D., Gervasi, V.: On the interplay between consistency, completeness and correctness in requirements evolution. Information and Software Technology **45**(14) (2003) 993–1009 [8]
571. Zowghi, D., Offen, R.: A logical framework for modeling and reasoning about the evolution of requirements. In: Proc. IEEE Int'l Symp. Requirements Engineering, IEEE Computer Society Press (1997) 247–259 [8]