References

[1] Alain Abran and James Moore, editors. *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, 2005.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the 11th International Conference on Data Engineering*, Taipei, Taiwan, 1995.

[3] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. *Encyclopedia of Software Engineering*, chapter Experience Factory. John Wiley and Sons, 1994.

[4] Peter Baxter. The MetricCenter toolkit. Distributive Software, Fredricksburg, Virginia, 2001.

[5] Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2000.

[6] Kent Beck. *Test-Driven Development by Example*. Addison Wesley, 2003.

[7] Barry Boehm, Chris Abts, A. Winsor Brown, Sunita Chulani, Bradford Clark, Ellis Horowitz, Ray Madachy, Donald Reifer, and Bert Steece. *Software Cost Estimation with COCOMO II*. Prentice Hall, 2000.

[8] Mike Chapman. NASA MDP repository. http://mdp.ivv.nasa.gov/, 2004.

[9] Noopur Davis. Team Software Process tool. http://www.sei.cmu.edu/tsp, 2004.

[10] Michael E. Fagan. Design and code inspections to reduce errors in program development. *IBM Systems Journal*, 15(3):182–211, 1976.

[11] Stuart Faulk, John Gustafson, Philip M. Johnson, Adam A. Porter, Walter Tichy, and Larry Votta. Toward accurate HPC productivity measurement. In *Proceedings of the First International Workshop on Software Engineering for High Performance Computing System Applications*, Edinburgh, Scotland, May 2004.

[12] Stuart Faulk, Philip M. Johnson, John Gustafson, Adam A. Porter, Walter Tichy, and Larry Votta. Measuring HPC productivity. *International Journal of High Performance Computing Applications*, December 2004.

[13] Ernest Friedman-Hill. *JESS in Action*. Mannig Publications Co., Greenwich, CT, 2003.

[14] Boby George and Laurie Williams. An Initial Investigation of Test-Driven Development in Industry. *ACM Sympoium on Applied Computing*, 3(1):23, 2003.

[15] Boby George and Laurie Williams. A Structured Experiment of Test-Driven Development. *Information & Software Technology*, 46(5):337–342, 2004.

[16] A. Geras, M. Smith, and J. Miller. A Prototype Empirical Evaluation of Test Driven Development. In *Software Metrics, 10th International Symposium on (METRICS'04)*, page 405, Chicago Illionis, USA, 2004. IEEE Computer Society.

[17] E. Heierman, G. Youngblood, and D. Cook. Mining temporal sequences to discover interesting patterns. In *Proceedings of the 2004 International Conference on Knowledge Discovery and Data Mining*, Seattle, Washington, 2004.

[18] Lorin Hochstein, Victor Basili, Marvin Zelkowitz, Jeffrey Hollingsworth, and Jeff Carver. Combining self-reported and automatic data to improve effort measurement. In *Proceedings of the 2005 Conference on Foundations of Software Engineering*, 2005.

[19] Watts S. Humphrey. *A Discipline for Software Engineering*. Addison-Wesley, New York, 1995.

[20] A. Jedlitschka and M. Ciolkowski. Towards evidence in software engineering. In *Proceedings of the 2004 International Symposium on Empirical Software Engineering*, 2004.

[21] Philip M. Johnson. Hackystat system. http://www.hackystat.org/.

[22] Philip M. Johnson. The Hackystat-JPL configuration: Overview and initial results. Technical Report CSDL-03-07, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, October 2003.

[23] Philip M. Johnson and Anne M. Disney. The personal software process: A cautionary case study. *IEEE Software*, 15(6), November 1998.

[24] Philip M. Johnson, Hongbing Kou, Joy M. Agustin, Christopher Chan, Carleton A. Moore, Jitender Miglani, Shenyan Zhen, and William E. Doane. Beyond the personal software process: Metrics collection and analysis for the differently disciplined. In *Proceedings of the 2003 International Conference on Software Engineering*, Portland, Oregon, May 2003.

[25] Philip M. Johnson, Hongbing Kou, Joy M. Agustin, Qin Zhang, Aaron Kagawa, and Takuya Yamashita. Practical automated process and product metric collection and analysis in a classroom setting: Lessons learned from hackystat-uh. In *Proceedings of the 2004 International Symposium on Empirical Software Engineering*, Los Angeles, California, August 2004.

[26] Philip M. Johnson, Hongbing Kou, Michael G. Paulding, Qin Zhang, Aaron Kagawa, and Takuya Yamashita. Improving software development management through software project telemetry. *IEEE Software*, August 2005.

[27] Philip M. Johnson, Carleton A. Moore, Joseph A. Dane, and Robert S. Brewer. Empirically guided software effort guesstimation. *IEEE Software*, 17(6), December 2000.

[28] Philip M. Johnson and Michael G. Paulding. Understanding HPCS development through automated process and product measurement with hackystat. In *Second Workshop on Productivity and Performance in High-End Computing (P-PHEC)*, February 2005.

[29] Aaron Kagawa. Hackystat MDS supporting MSL MMR. Technical Report CSDL-04-06, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, June 2004.

[30] Aaron Kagawa and Philip M. Johnson. The Hackystat-JPL configuration: Round 2 results. Technical Report CSDL-03-07, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, May 2004.

[31] Gerold Keefer. Extreme programming considered harmful for reliable software development. Technical report, AVOCA GmbH, 2003.

[32] B. Kitchenham. Systematic reviews. In *Proceedings of the 2004 International Symposium on Software Metrics*, 2004.

[33] Barbara Kitchenham, Tore Dyba, and Magne Jorgensen. Evidence-based software engineering. In *Proceedings of the 2004 International Conference on Software Engineering*, 2004.

[34] Nancy Leveson and Clark Turner. An investigation of the Therac-25 accidents. *IEEE Computer*, July 1993.

[35] H. Mannila, H. Toivonen, and A. Verkamo. Discovering frequent episodes in sequences. In *Proceedings of the 1995 International Conference on Knowledge Discovery and Data Mining*, Montreal, Canada, 1995.

[36] M. Matthias Muller and Oliver Hagner. Experiment about Test-first Programming. In *Empirical Assesment in Software Engineering (EASE)*. IEEE Computer Society, 2002.

[37] Matjaz Pancur and Mojca Ciglaric. Towards empirical evaluation of test-driven development in a university environment. In *Proceedings of EUROCON 2003*. IEEE, 2003.

[38] Michael G. Paulding. Measuring the processes and products of HPCS development: Initial results for the optimal truss purpose-based benchmark. Technical Report CSDL-04-13, Department of Information and Computer Sciences, University of Hawaii, Honolulu, Hawaii 96822, September 2004.

[39] Lutz Prechelt. The 28:1 Grant/Sackman legend is misleading, or: How large is interpersonal variation really? Technical Report 1999-18, University of Karlsruhe, 1999.

[40] Ken Raisor and David Tuma. Process dashboard for PSP. http://processdash.sourceforge.net/, 2001.

[41] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Company, 1989.

[42] Walker Royce. CMM vs. CMMI: From conventional to modern software management. *The Rational Edge*, February 2002.

[43] H. Sackman, W. Erikson, and E. Grant. Exploratory experimental studies comparing online and offline programming performance. *Communications of the ACM*, 11(1), 1968.

[44] G. Schulmeyer. The net negative producing programmer. *American Programmer*, June 1992.

[45] Alberto Sillitti, Andrea Janes, Giancarlo Succi, and Tullio Vernazza. Collecting, integrating and analyzing software metrics and personal software process data. In *Proceedings of the 29th Euromicro Conference*, 2003.