References

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

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

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

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

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

[6] 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.

[7] Aaron Cass, Barbara Lerner, Eric McCall, Leon Osterweil, Stanley Sutton, and Alexander Wise. Little-jil/juliette: A process definition language and interpreter. In *Proceedings of the 22nd International Conference on Software Engineering*, 2000.

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

[9] Jonathan E. Cook and Alexander L. Wolf. Automating process discovery through event-data analysis. In *ICSE '95: Proceedings of the 17th international conference on Software engineering*, pages 73–82, New York, NY, USA, 1995. ACM Press.

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

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

[12] 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.

[13] 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.

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

[15] 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.

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

[17] Dirk Jager, Ansgar Schleicher, and Bernhard Westfechtel. Using UML for software process modeling. 1999.

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

[19] Chris Jensen and Walt Scacchi. Experience in discovering, modeling, and reenacting open source software development processes. In *Proceedings of the International Software Process Workshop*, 2005.

[20] Philip M. Johnson. Hackystat Framework Home Page. http://www.hackystat.org/.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.

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

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

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

[33] Hongbing Kou and Philip M. Johnson. Automated recognition of low-level process: A pilot validation study of Zorro for test-driven development. In *Proceedings of the 2006 International Workshop on Software Process*, Shanghai, China, May 2006.

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

[35] Elisabetta Di Nitto, Luigi Lavazza, Marco Shiavoni, Emma Trananella, and Michelle Tombetta. Deriving executable process descriptions from UML. In *Proceedings of the 24th International Conference on Software Engineering*, 2002.

[36] Leon J. Osterweil. Unifying microprocess and macroprocess research. In *Proceedings of the International Software Process Workshop*, pages 68–74, 2005.

[37] 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.

[38] 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.

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

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

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

[42] Craig Schlenoff, Michael Gruninger, Florence Tissot, John Valois, Josh Lubell, and Jintae Lee. *The Process Specification Language (PSL) Overview and Version 1.0 Specification*. National Institute of Standards and Technology, 2000.

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

[44] Terry Shepard, Steve Sibbald, and Colin Wortley. A visual software process language. *Communications of the ACM*, April 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.

[46] Stanley Sutton and Leon Osterweil. The design of a next-generation process language. In *Proceedings of the Fifth International Symposium on Foundations of Software Engineering*, 1997.