



Software Phase II Reading List

This is the current Software Phase II Reading List. It is intended as a resource for students who are planning to take the Phase II Written Exam in the Software track of the Informatics concentration.

This list is subject to change.

Reading list updated: *July 2003*.

Current Topics

Architecture

[MT00] N. Medvidovic, R. Taylor. A Classification and Comparison Framework for Software Architecture Description Languages. *IEEE Transactions on Software Engineering*, Vol. 26, No. 1, pp. 70-93 (January 2000).

[GAO95] David Garlan, Robert Allen, and John Ockerbloom. Architectural Mismatch: Why Reuse Is So Hard. *IEEE Software* 12(6): 17-26, November 1995.

[PW92] Dewayne E. Perry and Alexander L. Wolf. Foundations for the Study of Software Architecture. *ACM Software Engineering Notes* 17(4):40-52, October 1992.

[TMA+96] Richard N. Taylor, Nenad Medvidovic, Kenneth M. Anderson, E. James Whitehead, Jr., Jason E. Robbins, Kari A. Nies, Peyman Oreizy, and Deborah L. Dubrow. A Component- and Message-Based Architectural Style for GUI Software. *IEEE Transactions on Software Engineering* 22(6):390-406, June 1996.

Events-based Architectures, Middleware, and Usage Monitoring

[CRW00] Carzaniga, A., Rosenblum, D.S., and Wolf, A.L. Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service. In *Proceedings of the Nineteenth ACM Symposium on Principles of Distributed Computing*, p. 219-227, ACM Press, Portland, OR, July, 2000.

[HR98] David Hilbert and David Redmiles. An Approach to Large-Scale Collection of Application Usage Data Over the Internet. In *Proc. 20th International Conference on Software Engineering*, pp. 136-145, Kyoto, Japan, April 1998.

Components and Programming Languages

[Fra97] M. Franz. Dynamic Linking of Software Components. *IEEE Computer*, 30:3, pp. 74-81; March 1997.

Formal Methods

[JD96] Jackson, D. and Damon, C.A. Elements of Style: Analyzing a Software Design Feature with a Counterexample Detector. *IEEE Transactions on Software Engineering*. 22(7), p. 484-495, 1996.

[Win90] J.M. Wing. A Specifier's Introduction to Formal Methods. *IEEE Computer*, 23(9):8-24, September 1990.

Validation

[AG93] J.M. Atlee and J.D. Gannon. State-Based Model Checking of Event-Driven System Requirements. *IEEE Transactions on Software Engineering*, 19(1):24-40, January 1993.

[Hol97] Gerard J. Holzmann. The Model Checker SPIN. *IEEE Transactions on Software Engineering*, 23(5):279-295, May 1997.

Testing

[Ric94] D.J. Richardson. TAOS: Testing with Analysis and Oracle Support. *Proceedings of the 1994 International Symposium on Software Testing and Analysis*, pp. 138-153, Seattle, WA, August 1994.

[Ros95] Rosenblum, D.S. A Practical Approach to Programming with Assertions. *IEEE Transactions on Software Engineering*. 21(1), p. 19-31, January, 1995.

[PY99] Pavlopoulou, C. and M. Young (1999). Residual test coverage monitoring. *Proceedings of the 1999 international conference on Software engineering*, 1999, Pages 277 - 284.

[RH96] Gregg Rothermel and Mary Jean Harrold. Analyzing Regression Test Selection Techniques. *IEEE Transactions on Software Engineering*, 22(8):529-551, August 1996.

[DC96] Dwyer, M. and Clarke, L.A. A Flexible Architecture for Building Data Flow Analyzers. *Proceedings of the Eighteenth International Conference on Software Engineering*. p. 554-564, ACM, 1996.

[Wey98] Weyuker, E.J. Testing Component-Based Software: A Cautionary Tale. *IEEE Software*. 15(5), p. 54-59, 1998.

Metrics

[BCH+95] B. Boehm, B. Clark, et al. Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering*, 1: 57-94, 1995.

[FN00] Norman E. Fenton and Martin Neil. Software Metrics: Roadmap. *Proceedings of ICSE - Future of Software Engineering Track*. p. 357-370, 2000.

Process

[CW98-1] Cook, J.E. and Wolf, A.L. Discovering Process Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology*. 7(3), p. 215-249, 1998.

[BT98] G.A. Bolcer and R.N. Taylor. Advanced Workflow Management Technologies. *Software Process - Improvement and Practice*. 4, p. 125-171, 1998.

Workflow

[Nut96] G. Nutt. The Evolution Towards Flexible Workflow Systems. *Distributed Systems Engineering* 3(4):276-294, December 1996.

[Eng90] Y. Engeström. When is a Tool? Multiple Meanings of Artifacts in Human Activity. Chapter 8 of *Learning, Working and Imagining*, Painettu Kirjapaino Oma Kyssä, Jyväskylässä, 1990, pp. 171-195.

Configuration Management

[HMR+01] A. van der Hoek, M. Mikic-Rakic, R. Roshandel, and N. Medvidovic. Taming Architectural Evolution. In *Proceedings of the Sixth European Software Engineering Conference (ESEC) and the Ninth ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9)*. p. 1-10, Vienna, Austria, September 10-14, 2001.

[HCH+02] A. van der Hoek, A. Carzaniga, D.M. Heimbigner, and A.L. Wolf. A Testbed for Configuration Management Policy Programming. *IEEE Transactions on Software Engineering*. To Appear.

[CW98-2] R. Conradi and B. Westfechtel. Version Models for Software Configuration Management. *ACM Computing Surveys*. 30(2), p. 232-282, 1998.

Software Environments and Design Environments

[RHR98] Robbins, J., Hilbert, D., Redmiles, D. Extending Design Environments to Software Architecture Design, *Automated Software Engineering*, Vol. 5, No. 3, July 1998, pp. 261-290.

[TN92] Thomas and Nejme. Definitions of tool integration for environments. *IEEE Software*, 9(2):29-35, March 1992.

[Kad92] R. Kadia. Issues Encountered in Building a Flexible Software Development Environment: Lessons Learned from the Arcadia Project. *Proceedings of ACM SIGSOFT '92: Fifth Symposium on Software Development Environments*, pp. 169-180, Washington, DC, December 1992.

Requirements and Safety

[LT93] Leveson, N. G. and C. S. Turner. An investigation of the Therac-25 accidents. *Computer*, 26(7): 18-41, 1993.

Evaluation / Assessment

[McG94] J. McGrath. Methodology Matters: Doing Research in the Behavioral and Social Sciences. In R. Baecker et al., *Readings in human-computer interaction : toward the year 2000*, Morgan Kaufmann, 1995, pp. 152-169.

HCI

[Mae94] Maes, P. Agents That Reduce Work and Information Overload, *Communications of the ACM*, Vol. 37, No. 7, 1994, pp. 30-40.

[Hil93] W.C. Hill. A Wizard of Oz Study of Advice Giving and Following. *Human-Computer Interaction*, 8(1):57-81, 1993.

[Nie93] J. Nielsen. What is Usability. In J. Nielsen, *Usability Engineering*, Morgan Kaufmann, 1993, Ch. 2, pp. 23-48.

CSCW

[AP96] M.S. Ackerman and L. Palen. The Zephyr Help Instance: Promoting Ongoing Activity in a CSCW System. *Proceedings of the 1996 ACM Conference on Human Factors in Computing Systems (CHI '96)*, pp. 268-275, Vancouver, BC, April 1996.

[MC94] Malone, T. W. and K. Crowston (1994). The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1): 87-119.

Design (Design Rationale, Reuse)

[Kru92] C. W. Krueger. Software Reuse. *ACM Computing Surveys*, 24(2):131-184, June 1992.

[JLS92] A. Jarczyk, P. Loeffler and I.F. Shipman. Design Rationale for Software Engineering: A Survey. *Proceedings of the 25th Annual IEEE Computer Society Hawaii Conference on System Sciences*, pp. 577-586, January 1992.

[Bec99] Beck, K. Embracing Change with Extreme Programming. *IEEE Computer*. 32(10), p. 70-77, 1999.

Hypertext

[Eng95] Engelbart, D. C. Toward augmenting the human intellect and boosting our collective IQ. *Communications of the ACM*, 38(8): 30, 32-3, (1995).

[Nel95] Nelson, T. H. The heart of connection: hypermedia unified by transclusion. *Communications of the ACM*, 38(8): 31-3 (1995).

[HS94] Halasz, F. and M. Schwartz. "The Dexter Hypertext Reference Model." *Communications of the ACM* 37(2): 30-9 (1994).

Web

[FWA+98] R. Fielding, E.J. Whitehead, K. Anderson, G. Bolcer, P. Oreizy and R. Taylor. Web-based Development of Complex Information Products. *Communications of the ACM*, 41(8):84-92, August 1998.

[B-LC+94] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret. The World-Wide Web. *Communications of the ACM*, 37(8): 76-82, August 1994.

[FT00] R. Fielding and R.N. Taylor. Principled Design of the Modern Web Architecture. *Proceedings of the 2000 International Conference on Software Engineering (ICSE 2000)*. p. 407-416, Limerick, Ireland, 2000.

Open Source

[Ray98] Raymond, E. S. The cathedral and the bazaar [Linux operating system]. *First Monday*, 3(3), 1998.

Foundations / Classics

Process

[Bro87] Frederick P. Brooks. No Silver Bullet: Essence and Accident in Software Engineering. *IEEE Computer* 20(4):10-19, April 1987.

[Ost87] Leon J. Osterweil. Software Processes Are Software Too. In *Proceedings of the 9th International Conference on Software Engineering*, pp. 2-13, Monterey, CA, March 1987.

[PC86] D. L. Parnas and P. C. Clements. A Rational Design Process: How and Why to Fake It. *IEEE Transactions on Software Engineering*, SE-12(2):251-257, February 1986.

[Boe88] B.W. Boehm. A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21(5):61-72, May 1988.

Evaluation / Assessment

[BSH86] V.R. Basili, R.W. Selby, and D.H. Hutchens. Experimentation in Software Engineering. *IEEE Transactions on Software Engineering*, SE-12(7):733-743, July 1986.

Components and Programming Languages

[McI68] M.D. McIlroy. 'Mass Produced' Software Components. In *Software Engineering: A Report on a Conference Sponsored by the NATO Science Committee*, P. Naur and B. Randell (eds.), pp. 138-155, Garmisch, Germany, October 1968.

Safety

[Lev86] N.G. Leveson. Software Safety: What, Why, and How, *ACM Computing Surveys*, 18(2):125-163, June 1986.

Testing

[RW85] Rapps, S. and Weyuker, E.J. Selecting Software Test Data Using Data Flow Information. *IEEE Transactions on Software Engineering*. SE-11(4), p. 367-375, April 1985.

[Wei84] M. Weiser. Program Slicing, *IEEE Transactions on Software Engineering*, SE-10(4):352-357, July 1984.

Verification and Validation

[ABC82] W. Richards Adrion, Martha A. Branstad, and John C. Cherniavsky. Validation, verification, and testing of computer software. *ACM Computing Surveys*, 14(2):159-192, June 1982.

Environments

[TM81] W. Teitelman and L. Manister. The Interlisp Programming Environment. *IEEE Computer*, 14(4):25-33, April 1981.

Program Transformation / Generation

[Nei84] Neighbors, J. M. (1984). The Draco approach to constructing software from reusable components. *IEEE Transactions on Software Engineering*, SE-10(5): 564-74.

Design

[Wir71] N. Wirth. Program Development by Stepwise Refinement. *Communications of the ACM*, 14(4):221-227, April 1971.

[CKI88] B. Curtis, H. Krasner and N. Iscoe. A Field Study of the Software Design Process for Large Systems. *Communications of the ACM*, 31(11):1268-1287, November 1988.

Books

[Som01] Sommerville, I. *Software Engineering*: 6th ed. Addison-Wesley. 1996.

[Bro95] F.P. Brooks. *The Mythical Man-Month, 25th Anniversary Edition*. Addison-Wesley, Reading, MA, 1995.

[Jac95] Jackson, M. *Software Requirements & Specifications: A Lexicon of Practice, Principles and Prejudices*. p. 228+xvi, ACM Press/Addison-Wesley. 1995.