

# References

- [Abde86a] Abdel-Ghaly, A.A., "Analysis of Predictive Quality of Software Reliability Models," Ph.D. dissertation, City University, London, 1986.
- [Abde86b] Abdel-Ghaly, A.A., Chan, P.Y., and Littlewood, B., "Evaluation of Competing Software Reliability Predictions," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 9, September 1986, pp. 950-967.
- [Abra87] Abraham, J.A., Metze, G., Iyer, R.K., and Patel, J.H., "The Evolution of Fault-Tolerant Computing at the University of Illinois," *The Evolution of Fault-Tolerant Computing*, vol. 1 of *Dependable Computing and Fault-Tolerant Systems*, A. Avizienis, H. Kopetz, and J.-C. Laprie (eds.), Springer-Verlag, Wien and New York, 1987, pp. 271-311.
- [Abramson, S.R., et al., "Customer Satisfaction-Based Product Development," *Proceedings International Switching Symposium*, vol. 2, Inst. Electronics, Information, Communications Engineers, Yokohama, Japan, 1992, pp. 65-69.
- [Abra92] Adams, E.N., "Optimizing Preventive Service of Software Products," *IBM Journal of Research and Development*, vol. 28, no. 1, January 1984, pp. 2—14.
- [Adam84] Adams, E., and Kulisch, U. (eds.), *Scientific Computing with Automatic Result Verification*, Academic Press, New York, 1993.
- [Adam93] Agresti, W.W., and Evanco, W.M., "Projecting Software Defects From Analyzing Ada Designs," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, November 1992, pp. 988-997.
- [Agre92] Ahmed, S., and Tesauro, G., "Scaling and Generalization in Neural Net works: A Case Study," *Advances in Neural Information Processing Systems 1*, D. Touretzky (ed.), Morgan Kaufmann, San Francisco, 1989, pp. 160-168.
- [Ahme89] American Institute of Aeronautic and Astronautics, *Recommended Practice for Software Reliability ANSI/AIAA R-013-1992*, February 1993.
- [AIAA93] Akaike, H., "A New Look at Statistical Model Identification," *IEEE Transactions on Automatic Control*, AC-19, 1974, pp. 716-723.
- [Akai74] Ammann, P.E., and Knight, J.C., "Data Diversity: An Approach to Software Fault Tolerance," *Proceedings of the 17th International Symposium on Fault-Tolerant Computing (FTCS-17)*, IEEE Computer Society Press, Pittsburgh, Pennsylvania, 1987, pp. 122-126.
- [Amma87] Anderson, T., and Lee, PA., *Fault Tolerance-Prentice-Hall*, 1981.
- [Ande81] Anderson, T, Barrett, P.A., Halliwell, D.N, and Moulding, M.R., "Software Fault-Tolerance: An Evaluation," *IEEE Transactions on Software Engineering*, vol. SE-11 (12), 1985, pp. 1502-1510.
- [Ande85]

- [Angu82] Angus, J.E., and James, L.E., "Combined Hardware/Software Reliability Models," *Proceedings of the Annual Reliability and Maintainability Symposium*, 1982, pp.176-181.
- [Anna80] Anna-Mary, B., "A Study of the Musa Reliability Model," M.S. thesis, Computer Science Department, University of Maryland, 1980.
- [ANSI86] ANSI/IEEE Std. 1012, "ANSI/IEEE Standard for Software Verification and Validation Plans," *Software Engineering Standards*, 3d ed., 1986.
- [ANSI91] ANSI/IEEE, "Standard Glossary of Software Engineering Terminology," *STD-729-1991*, ANSI/IEEE, 1991.
- [Ara90] Arlat, J., Kanoun, K., and Laprie, J.-C., "Dependability Evaluation of Software Fault-Tolerance," *Proceedings of the 18th IEEE International Symposium on Fault-Tolerant Computing (FTCS-18)*, Tokyo, June 1988, pp. 142-147; also in *IEEE Transactions on Computers*, vol. 39, no. 4, April 1990, pp.504-513.
- [Arno73] Arnold, T.F., "The Concept of Coverage and Its Effect on the Reliability Model of Repairable Systems," *IEEE Transactions on Computers*, vol. C-22, June 1973, pp.251-254.
- [Arti86] Artis, H.P., "Workload Characterization Using SAS PROC FASTCLUS," *Workload Characterization of Computer Systems and Computer Networks*, G. Serazzi (ed.), Elsevier Science Publishers, 1986.
- [Asch84] Ascher, H., and Feingold, H., *Repairable Systems Reliability: Modeling, Inference, Misconceptions and Their Causes*, lecture notes in statistics, vol. 7, Dekker, New York and Basel, 1984.
- [Atha89] Athavale, A., "Performance Evaluation of Hybrid Voting Schemes," M.S. thesis, North Carolina State University, Department of Computer Science, December 1989.
- [ATT90] AT&T Bell Laboratories, "Draft Software Reliability Engineering: Reliability Estimation Tools Reference Guide Version 3.7," August 1990.
- [Avey80] Aveyard, R.L., and Man, FT, "A Study on the Reliability of the Circuit Maintenance System I-B," *Bell System Technical Journal*, vol. 59, October 1980, pp.1317-1332.
- [Aviz77] Avizienis, A., and Chen, L., "On the Implementation of N-Version Programming for Software Fault-Tolerance during Program Execution," in *Proceedings COMPSAC 77*, 1977, pp. 149-155.
- [Aviz78] Avizienis, A., "Fault-Tolerance: The Survival Attribute of Digital Systems," *Proceedings of the IEEE*, vol. 66, no. 10, October 1978, pp. 1109-1125.
- [Aviz86] Avizienis, A., and Laprie, J.-C., "Dependable Computing: From Concepts to Design Diversity," *Proceedings of the IEEE*, vol. 74, no. 5, May 1986, pp. 629-638.
- [Aviz84] Avizienis, A. and Kelly, J.P.J., "Fault Tolerance by Design Diversity: Concepts and Experiments," *IEEE Computer*, August 1984, pp. 67-80.
- [Aviz85] Avizienis, A., "The N-Version Approach to Fault-Tolerant Software," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 12, December 1985, pp.1491-1501.
- [Aviz87] Avizienis, A., and Ball, D.E., "On the Achievement of a Highly Dependable and Fault-Tolerant Air Traffic Control System," *IEEE Computer*, vol. 20, no. 2, 1987, pp.84-90.

- [Aviz88] Avizienis, A., Lyu, M.R., and Schiitz, W., "In Search of Effective Diversity: A Six-Language Study of Fault-Tolerant Flight Control Software," *Proceedings of the 18th International Symposium on Fault Tolerant Computing*, Tokyo, Japan, June 1988, pp.15-22.
- [Barl75] Barlow, R.E., and Proschan, F., *Statistical Theory of Reliability and Life Testing*, Holt, New York, 1975.
- [Basi84a] Basili, V.R., and Perricone, B.T., "Software Errors and Complexity: An Empirical Investigation," *Communications of the ACM*, vol. 22, no. 1, January 1984, pp.42-52.
- [Basi84b] Basili, V., and Weiss, D., "A Methodology for Collecting Valid Software Engineering Data," *IEEE Transactions on Software Engineering*, vol. SE-10, no. 6, 1984, pp.728-738.
- [Basi88] Basili, V.R., and Rombach, H.D., "The TAME Project: Towards Improvement Oriented Software Environments," *IEEE Transactions on Software Engineering*, vol. 14, no. 6, June 1988, pp.758-773.
- [Bass94] Bassin, K., "The Butterfly Model. A Proven Approach to Software Quality Improvement," IBM Technical Report, Glendale Programming Lab., vol. 752-5582, January 1994.
- [Baue85] Bauer, H.A., Croxall, L.M., and Davis, E.A., "The 5ESS Switching System: System Test, First-Office Application, and Early Field Experience," *AT&T Technical Journal*, vol. 64, no. 6, 1985, pp. 1503-1522.
- [Beck90] Becker, G., and Camarinopoulos, L., "A Bayesian Estimation Method for the Failure Rate of a Possibly Correct Program," *IEEE Transactions on Software Engineering*, vol. 16, no. 11, November 1990, pp. 1307-1310.
- [Beiz90] Beizer, B., *Software Testing Techniques*, 2d ed., Van Nostrand, 1990.
- [BELL89] BELLCORE, *Network Switching Element Outage Performance Monitoring Procedures*, SR-TSY-000963, issue I, April 1989.
- [BELL90a] BELLCORE, *The Analysis and Use of Software Reliability and Quality Data*, SR-TSY-001547, issue 1, January 1990.
- [BELL90b] BELLCORE, *Reliability and Quality Measurements for Telecommunications Systems (RQMS)*, TR-TSY-000929, issue 1, June 1990.
- [BELL90c] BELLCORE, "Reliability Section 12, Issue 3," in *LSSGR—LATA (Local Access and Transport Area) Switching Systems Generic Requirements*, Bell-core, February 1990.
- [Bell90] Belli, F., and Jedrzejowicz, P., "Fault-Tolerant Programs and Their Reliability," *IEEE Transactions on Software Engineering*, vol. 16, no. 2, 1990, pp. 184-192.
- [Bell91] Belli, F., and Jedrzejowicz, P., "Comparative Analysis of Concurrent Fault-Tolerance Techniques for Real-Time Applications," *Proceedings of the Second International Symposium on Software Reliability Engineering*, Austin, Texas, 1991.
- [Benn92] Bennett, J., Denoncourt, M., and Healy, J.D., "Software Reliability Prediction for Telecommunication Systems," *Proceedings of the Second Bell-core/Purdue Symposium on Issues in Software Reliability Estimation*, October 1992, pp. 85-102.
- [Berg89] Bergen, L.A., "A Practical Application of Software Reliability to a Large Scale Switching System," *IEEE International Workshop: Measurement of Quality During the Life Cycle*, Val David, Quebec, Canada, April 25-27, 1989.

- [Beya81] [Beyaert, B.](#), [Florin, G.](#), [Lone, P.](#), and [Natkin, S.](#), "Evaluation of Computer Systems Dependability Using Stochastic Petri Nets," *Proceedings of the 11th IEEE International Symposium Fault-Tolerant Computing (FTCS-11)*, Portland, Maine, June 1981, pp. 79-81.
- [Bhan94] [Bhandari, I.](#), [Halliday, M.J.](#), [Chaar, J.](#), [Chillarege, R.](#), [Jones, K.](#), [Atkinson, J.S.](#), [Lepori-Costello, C.](#), [Jasper, P.Y.](#), [Tarver, E.D.](#), [Lewis, C.C.](#), and [Yonezawa, M.](#), "In-Process Improvements Through Defect Data Interpretation," *IBM Systems Journal*, vol. 33, no. 1, 1994, pp. 182-214.
- [Bhar81] [Bhargava, B.](#), and [Hua, C.](#), "Cost Analysis of Recovery Block Scheme and Its Implementation Issues," *International Journal of Computer and Information Sciences*, vol. 10, no. 6, 1981, pp. 359-382.
- [Biro74] [Biolini, A.](#), "Some Applications of Regenerative Stochastic Processes to Reliability Theory—Part One: Tutorial Introduction," *IEEE Transactions on Reliability*, vol. R-23, no. 3, August 1974, pp. 186-194.
- [Bish86] [Bishop, P.O.](#), [Esp, D.G.](#), [Barnes, M.](#), [Humphreys, P.](#), [Dahl, G.](#), and [Lahti, J.](#), "PODS—A Project on Diverse Software," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 9, 1986, pp. 929-940.
- [Bish88] [Bishop, P.O.](#), and [Pullen, P.D.](#), "PODS Revisited—A Study of Software Failure Behavior," *Proceedings of the 18th International Symposium on Fault-Tolerant Computing*, Tokyo, Japan, 1988, pp. 2-8.
- [Bish91] [Bishop, P.G.](#), and [Pullen, F.D.](#), "Error Masking: A Source of Failure Dependency in Multi-Version Programs," in *Dependable Computing for Critical Applications*, [J.-C. Laprie](#) and [A. Avizienis](#) (eds.), Springer-Verlag, Vienna, Austria, 1991, pp. 53-73.
- [Biya95] [Biyani, Shriram](#), "A Graphical Display for Categorical Data Series," *IBM Research Report*, 1995.
- [Bobb86] [Bobbio, A.](#), and [Trivedi, K. S.](#), "An Aggregation Technique for the Transient Analysis of Stiff Markov Chains," *IEEE Transactions on Computers*, vol. C-35, September 1986, pp. 803-814.
- [Boeh79] [Boehm, B.W.](#), "Guidelines for Verifying and Validating Software Requirements and Design Specifications," in *Proceedings EURO IFIP'79*, London, September 1979, pp. 711-719.
- [Boeh81] [Boehm, B.W.](#), *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, New York, 1981.
- [Boeh86] [Boehm, B.W.](#), "A Spiral Model of Software Development and Enhancement," *ACM SIGSOFT Software Engineering Notes*, vol. 11, no. 4, 1986, pp. 14-24.
- [Boeh89] [Boehm, B.W.](#), *Tutorial: Software Risk Management*, IEEE CS Press, 1989.
- [Boeh91] [Boehm, B.W.](#), "Software Risk Management: Principles and Practices," *IEEE Software*, January 1991, pp. 32-41.
- [Bour69] [Bouricius, W.G.](#), [Carter, W.C.](#), [Schneider, P.R.](#), "Reliability Modeling Techniques for Self-Repairing Computer Systems," *Proceedings of the 24th ACM National Conference* 1969, pp. 295-309.
- [Bria92] [Briand, L.C.](#), [Basili, V.R.](#), and [Hetmanski, C.J.](#), "Providing an Empirical Basis for Optimizing the Verification and Testing Phases of Software Development," *Proceedings of the Third International Symposium on Software Reliability Engineering* Research Triangle Park, North Carolina, October 1992, pp. 329-338.

- [Bria93] Briand, L.C., Thomas, W.M., and Hetsmanski, C.J., "Modeling and Managing Risk Early in Software Development," *Proceedings of the 15th International Conference on Software Engineering*, 1993, pp. 55-65.
- [Bric84] Brick, D.B., Draper, J.S., and Caulfield, H.J., "Computers in the Military and Space Sciences," *Computer*, vol. 17, no. 10, 1984, pp. 250-262.
- [Bril87] Brilliant, S.S., Knight, J.C., and Leveson, N.G., "The Consistent Comparison Problem in N-Version Software," *ACM SIGSOFT Software Engineering Notes*, vol. 12, no. 1, 1987, pp. 29-34.
- [Bril90] Brilliant, S.S., Knight, J.C., and Leveson, N.G., "Analysis of Faults in an N-Version Software Experiment," *IEEE Transactions on Software Engineering*, vol. 16, no. 2, 1990, pp. 238-247.
- [Broc87] Brocklehurst, S., "On the Effectiveness of Adaptive Software Reliability Modelling," Technical Report, Centre for Software Reliability, City University, London, 1987.
- [Broc90] Brocklehurst, S., Chan, P.Y., Littlewood, B., and Snell, J., "Recalibrating Software Reliability Models," *IEEE Transactions on Software Engineering*, vol. SE-16, no. 4, April 1990, pp. 458-470.
- [Broc92] Brocklehurst, S., and Littlewood, B., "New Ways to Get Accurate Reliability Measures," *IEEE Software*, vol. 9, no. 4, July 1992, pp. 34-42.
- [Broo80] Brooks, W.D., and Motley, R.W., *Analysis of Discrete Software Reliability Models*, Rome Air Development Center Technical Report, RADC-TR-80-84, April 1980.
- [BStd86] British Standard 5760, "Reliability of Constructed or Manufactured Products, Systems Equipments and Components, Part 4: Guide to Specification Clauses to the Achievement and Development of Reliability in New and Existing Items," British Standard Institution, 1986.
- [Budd80] Budd, T.A., "Mutation Analysis of Program Test Data," Ph.D. dissertation, Yale University, May, 1980.
- [Burk46] Burks, A.W., Goldstine, H.H., and von Neumann, J., *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*, U.S. Army Ordnance Department, 1946.
- [Bush90] Bush, M., "Improving Software Quality: The Use of Formal Inspections at the Jet Propulsion Laboratory," *Proceedings of the 12th International Conference on Software Engineering (ICSE12)*, Nice, France, March 26-28, 1990, pp. 196-199.
- [Butl91] Butler, R.W., and Finelli, G.B., "The Infeasibility of Experimental Quantification of Life-Critical Software Reliability," *Proceedings of ACM SIG-SOFT'91 Conference on Software for Critical Systems*, New Orleans, Louisiana, December 1991, pp. 66-76.
- [Butl93] Butler, R.W., and Finelli, G.B., "The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software," *IEEE Transactions on Software Engineering*, vol. SE-19, January 1993, pp. 3-12.
- [Carm95] Carman, D.W., Dolinsky, A.A., Lyu, M.R., and Yu, J.S., "Software Reliability Engineering Study of a Large-Scale Telecommunications Software System," *Proceedings 1995 International Symposium on Software Reliability Engineering*, Toulouse, France, October 1995.

- Carter, W.C., Jessep, D.C., Bourricius, W.G., Wadia, A.B., McCarthy, C.E., and Milligan, F.G., "Design Techniques for Modular Architectures for Reliable Computer Systems," IBM T.J. Watson Report No. 70.208.0002, 1970.
- Carter, W.C., "Architectural Considerations for Detecting Run-Time Errors in Programs," *Proceedings of the 13th International Symposium on Fault-Tolerant Computing (FTCS-13)*, IEEE Computer Society Press, Milano, Italy, 1983, pp. 249-256.
- Castillo, X., and Siewiorek, D.P., "Workload, Performance, and Reliability of Digital Computing Systems," *Proceedings of the 11th IEEE International Symposium on Fault-Tolerant Computing (FTCS-11)*, Portland, Maine, June 1981, pp. 84-89.
- Castillo, X., and Siewiorek, D.P., "A Workload Dependent Software Reliability Prediction Model," *Proceedings of the 12th International Symposium on Fault-Tolerant Computing*, June 1982, pp. 279-286.
- Chaar, J., Halliday, M., Bhandari, I., and Chillarege, R., "In-Process Evaluation for Software Inspection and Test," *IEEE Transactions on Software Engineering*, 19, no. 11 November 1993.
- Chan, F., Dasiewicz, P., and Seviara, R., "Metrics for Evaluation of Software Reliability Growth Models," *Proceedings of the Second International Symposium on Software Reliability Engineering*, Austin, Texas, May 1991, pp. 163-167.
- Charette, R.N., *Software Engineering Risk Analysis & Management*, McGraw-Hill, New York, 1989.
- Chen, L., and Avizienis, A., "N-Version Programming: A Fault-Tolerance Approach to Reliability of Software Operation," *Proceedings of the 8th International Symposium on Fault-Tolerant Computing*, Toulouse, France, 1978, pp.3-9.
- Chen, M., Horgan, J.R., Mathur, A.P, and Rego, V.J., "A Time/Structure Based Model for Estimating Software Reliability," Technical Report SERC-TR-117-P, Software Engineering Research Center, Department of Computer Science, Purdue University, W. Lafayette, Ind., 1992.
- Chen, M., Mathur, A.P, and Rego, V.J., "Effect of Testing Techniques on Software Reliability Estimates Obtained Using Time Domain Models," *Proceedings of the 10th Annual Software Reliability Symposium*, IEEE Reliability Society, Denver, Colo., June 25-26, 1992, pp. 116-123; also in *IEEE Transactions on Reliability*, vol. 44, no. 1, March 1995, pp. 97-103.
- Chen, M.H., Jones, M.K., Mathur, A.P, and Rego, V.J., "Terse: A Tool for Evaluating Software Reliability Models," *Proceedings of the 4th International Symposium on Software Reliability Estimation*, Denver, Colo, November 1993, pp. 274-283.
- Chen, M., "Tools and Techniques for Testing Based Reliability Estimation," Ph.D. dissertation, Purdue University, 1994.
- Chen, M., Mathur, A.P, and Rego, V.J., "A Case Study to Investigate Sensitivity of Reliability Estimates to Errors in Operational Profiles," *Proceedings of the Fifth International Symposium on Software Reliability Engineering*, Monterey, Calif. November 6-9, 1994.
- Cheung, R.C., "A User-Oriented Software Reliability Model," *IEEE Transactions on Software Engineering*, vol. SE-6, March 1980, pp. 118-125.
- Chillarege, R., and Siewiorek, D.P., "Experimental Evaluation of Computer Systems Reliability," *IEEE Transactions on Reliability*, vol. 39, no. 4, October 1990.



- Chillarege, R., Kao, W.-L., and Condit, E.G., "Defect Type and Its Impact on the Growth Curve," *Proceedings of the 13th International Conference on Software Engineering*, 1991.
- Chillarege, R., Bhandari, I.S., Chaar, J.K., Halliday, M.J., Moebus, D.S., Ray, B.K., and Wong, M.-Y., "Orthogonal Defect Classification—A Concept for In-Process Measurements," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, November 1992, pp. 943-956.
- Chillarege, R., and Bassin, K., "Software Triggers and Their Characteristics—A Case Study on Field Failures from an Operating Systems Product," *Fifth IFIP Working Conference on Dependable Computing for Critical Applications*, September 1995.
- Choi, B.J., DeMillo, R.A., Krauser, E.W., Mathur, A.P., Martin, R.J., Offutt, A.J., Pan, H., and Spafford, E.H., "The Mothra Toolset," *Proceedings of Hawaii International Conference on System Sciences*, HI, January 3-6, 1989.
- Christenson, D.A., "Using Software Reliability Models to Predict Field Failure Rates in Electronic Switching Systems," *Proceedings of the Fourth Annual National Joint Conference on Software Quality and Productivity*, Washington, D.C., 1988.
- Christenson, D.A., Huang, S.T., and Lamperez, A.J., "Statistical Quality Control Applied to Code Inspections," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 2, 1990, pp. 196-200.
- Christmansson, J., Kalbarczyk, Z., and Torin, J., "Dependable Flight Control System Using Data Diversity with Error Recovery," *Journal of Computer Systems Science Engineering*, 1994.
- Clarke, L.A., Podgruski, A., Richardson, D.J., and Zeil, S., "A Formal Evaluation of Data Flow Path Selection Criteria," *IEEE Transactions on Software Engineering*, vol. 15, no. 11, November 1989, pp. 1318-1332.
- Cleveland, W., *Visualizing Data*, AT&T Bell Laboratories and Hobard Press, 1993.
- Clement, G.F., and Giloth, P.K., "Evolution of Fault Tolerant Switching Systems in AT&T," in *The Evolution of Fault-Tolerant Computing*, A. Avizienis, H. Kopetz, J.-C. Laprie (eds.), Springer-Verlag, Wien, Austria, 1987, pp. 37-54.
- Cohen, J., *The Single Server Queue*, North Holland, Amsterdam, 1969.
- Costes, A., Landrault, C., and Laprie, J.-C., "Reliability and Availability Models for Maintained Systems Featuring Hardware Failures and Design Faults," *IEEE Transactions on Computers*, vol. C-27, June 1978, pp. 548-560.
- Cote, V., et al., "Software Metrics: An Overview of Recent Results," *Journal of Systems and Software*, vol. 8, 1988, pp. 121-131.
- Coudert, O., and Madre, J.C., "Fault Tree Analysis:  $10^{20}$  Prime Implicants and Beyond," *Proceedings of the Reliability and Maintainability Symposium*, January 1993, pp. 240-245.
- Courtois, P. 3., *Decomposability: Queuing and Computer System Application*, Academic, New York, 1977.
- Coutinho, J. de S., "Software Reliability Growth," *IEEE Symposium on Computer Software Reliability*, 1973.

- [Cox62] Cox, D.R., *Renewal Theory*, Methuen, London, U.K., 1962.
- [Cox66] Cox, D.R., and Lewis, P.A.W., *The Statistical Analysis of a Series of Events*, Methuen, London, 1966.
- [Cox78] Cox, D.R., and Lewis, P.A.W., *The Statistical Analysis of a Series of Events*, Wiley & Sons, New York, 1978.
- [Cram92] Cramp, R., Vouk, M.A., and Jones, W., "On Operational Availability of a Large Software-Based Telecommunications System," *Proceedings of the Third International Symposium on Software Reliability Engineering*, IEEE Computer Society, 1992, pp. 358-366.
- [Craw85] Crawford, S., McIntosh, A., and Pregibon, D., "An Analysis of Static Metrics and Faults in C Software," *The Journal of Systems and Software*, vol. 5, 1985, pp. 37-48.
- [Cris82] Cristian, F., "Exception Handling and Software Fault Tolerance," *IEEE Transactions on Computers*, vol. C-31, no. 6, June 1982, pp. 531-540.
- [Cris85] Cristian, F., "A Rigorous Approach to Fault-Tolerant Programming," *IEEE Transactions on Software Engineering*, vol. SE-11, 1985, pp. 23-31.
- [Crow74] Crow, L.H., "Reliability Analysis for Complex Repairable Systems," *Reliability and Biometry*, F. Proshan and R.J. Serfling (eds.), SIAM, Philadelphia, pp. 379-410.
- [Crow77] Crow, L.H., "Confidence Interval Procedures for Reliability Growth Analysis," *tech. report 1977*, U.S. Army Material System Analysis Activity, Aberdeen, Md., 1977.
- [Csen90] Csenki, A., "Bayes Predictive Analysis of a Fundamental Software Reliability Model," *IEEE Transactions on Reliability*, vol. R-39, 1990, pp. 177-183.
- [Curr86] Currit, P.A., Dyer, M., and Mills, H.D., "Certifying the Reliability of Software," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 1, January 1986, pp. 3-11.
- [Cusu95] Cusumano, M., and Selby, R., *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*, Free Press, Simon & Schuster, New York, 1995.
- [Dala88] Dalal, S.R., and Mallows, C.L., "When Should One Stop Testing Software," *Journal of the American Statistical Association*, vol. 83, no. 403, September 1988, pp. 872-879.
- [Dala90] Dalal, S.R., and Mallows, C.L., "Some Graphical Aids for Deciding When to Stop Testing Software," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 2, February 1990, pp. 169-175.
- [Dala92] Dalal, S.R., and Mallows, C.L., "Buying with Exact Confidence," *The Annals of Applied Probability*, vol. 2, no. 3, 1992, pp. 752-765.
- [Dala93] Dalal, S.R., Horgan, J.R., and Kettenring, J.R., "Reliable Software and Communication: Software Quality, Reliability, and Safety," *Proceedings of the 15th International Conference on Software Engineering*, Baltimore, Md., May 1993.
- [Dala94] Dalal, S.R., and McIntosh, A.A., "When to Stop Testing for Large Software Systems with Changing Code," *IEEE Transactions on Software Engineering*, vol. 20, no. 4, April 1994, pp. 318-323.
- [Dani78] Daniel, W.W., *Applied Nonparametric Statistics*, PWS-KENT Pub., Boston, Mass., 1978.



- [Davi84] Davies, P.A., "The Latest Developments in Automatic Train Control," in *Proceedings of the International Conference on Railway Safety Control and Automation Towards 21st Century*, London, U.K., September 1984, pp. 272-279.
- [Davi93] Davis, G.J., Earls, M.R., and Patterson-Hine, F.A., "Reliability Analysis of the X-29A Flight Control System Software," *Journal of Computer and Software Engineering*, vol. 1, no. 4, 1993, pp. 325-348.
- [Dawi84] Dawid, A.P., "Statistical Theory: The Prequential Approach," *Journal Royal Statistical Society*, series A, vol. 147, 1984, pp. 278-292.
- [Deb86] Deb, A.K., and Goel, A.L., "Model for Execution Time Behavior of a Recovery Block" *Proceedings COMPSAC 86*, 1986, 497-502.
- [Deb88] Deb, A.K., "Stochastic Modelling for Execution Time and Reliability of Fault-Tolerant Programs Using Recovery Block and N-Version Schemes," Ph.D. thesis, Syracuse University, 1988.
- [DeGr86] DeGroot, M.H., *Probability and Statistics*, Series in Statistics, Addison-Wesley, Reading Mass., 1986.
- [DeMi91] DeMillo, R.A., and Mathur, A.P., "On the Use of Software Artifacts to Evaluate the Effectiveness of Mutation Analysis for Detecting Errors in Production Software," Technical Report SERC-TR-42-P, Software Engineering, Research Center, Purdue University, W. Lafayette, Ind., 1991.
- [DU184] Dillon, W.R., and Goldstein, M., *Multivariate Analysis*, John Wiley & Sons, 1984.
- [DiMa91] DiMario, M.J., "Measuring Software Product Quality That Includes Customer Perception," *QAMC/IEEE International Workshop*, Val David, Quebec, Canada, April 24, 1991.
- [Doyl95] Doyle, S.A., and Mackey, J.L., "Comparative Analysis of N-Version Programming (NVP) Architectures," *Proceedings of the Reliability and Maintainability Symposium*, January, 1995.
- [Drap86] Draper, D., and Smith, H., *Applied Regression Analysis*, 2nd ed., Wiley, New York, 1986.
- [Duan64] Duane, J.T., "Learning Curve Approach to Reliability Monitoring," *IEEE Transactions on Aerospace*, vol. 2, 1964, pp. 563-566.
- [Duga89a] Dugan, J.B., and Trivedi, K.S., "Coverage Modeling for Dependability Analysis of Fault-Tolerant Systems," *IEEE Transactions on Computers*, vol. 38, no. 6, 1989, pp. 775-787.
- [Duga89b] Dugan, J.B., Veeraraghavan, M., Boyd, M., and Mittal, N., "Bounded Approximate Reliability Models for Fault Tolerant Distributed Systems," *Proceedings of the 8th Symposium on Reliable Distributed Systems*, 1989, pp. 137-147.
- [Duga91] Dugan, J.B., "Correlated Hardware Failures in Redundant Systems," *Proceedings of the 2nd IFIP Working Conference Dependable Computing for Critical Applications*, Tucson, Arizona, February 1991.
- [Duga92] Dugan, J.B., Bavuso, S., and Boyd, M., "Dynamic Fault Tree Models for Fault Tolerant Computer Systems," *IEEE Transactions on Reliability*, September 1992.
- [Duga93a] Dugan, J.B., Bavuso, S., and Boyd, M., "Fault Trees and Markov Models for Reliability Analysis of Fault Tolerant Systems," *Journal of Reliability Engineering and System Safety*, vol. 39, 1993, pp. 291-307.

- [Duga93b] Dugan, J.B., and Van Buren, R., "Reliability Evaluation of Fly-by-Wire Computer Systems," *Journal of Systems and Safety*, June, 1993.
- [Duga94a] Dugan, J.B., "Experimental Analysis of Models for Correlation in Multiversion Software," *Proceedings of the International Symposium on Software Reliability Engineering*, 1994.
- [Duga94b] Dugan, J.B., and Lyu, M.R., "System-Level Reliability and Sensitivity Analysis for Three Fault-Tolerant System Architectures," *4th IFIP International Working Conference on Dependable Computing for Critical Applications (DCCA4)*, San Diego, Calif, January 1994, pp. 295-307.
- [Duga94c] Dugan, J.B., and Lyu, M.R., "System Reliability Analysis of an N-Version Programming Application," *IEEE Transactions on Reliability*, December 1994, pp. 513-519; also in *Proceedings ISSRE'93*, 1993, pp. 103-111.
- [Duga95] Dugan, J.B., and Lyu, M.R., "Dependability Modeling for Fault-Tolerant Software and Systems," chapter 5 in *Software Fault Tolerance*, M. Lyu (ed.), John Wiley & Sons, Chichester, U.K., 1995.
- [Dura84] Duran, J.W., and Ntafos, S.C., "An Evaluation of Random Testing," *IEEE Transactions on Software Engineering*, vol. SE-10, 1984, pp. 438-444.
- [Eckh85] Eckhardt, D.E., and Lee, L.D., "A Theoretical Basis for the Analysis of Multi-version Software Subject to Coincident Errors," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 12, 1985, pp. 1511-1517.
- [Eckh91j] Eckhardt, D.E., Caglayan, A.K., Kelly, J.P.J., Knight, J.C., Lee, L.D., McAlister, D.F., and Vouk, M.A., "An Experimental Evaluation of Software Redundancy as a Strategy for Improving Reliability," *IEEE Transactions on Software Engineering*, vol. 17, no. 12, 1991, pp. 692-702.
- [IEEC91] *Information Technology Security Evaluation Criteria, Provisional Harmonised Criteria* Office for Official Publications of the European Communities, June 1991.
- [Efro79] Efron, B., "Computers and the Theory of Statistics: Thinking the Unthinkable," *SIAM Review*, vol. 21, no. 4, October 1979, pp. 460-480.
- [Ehre83] Ehrenberger, W., "Safety, Availability and Cost Questions About Diversity," *4th IFAC/IFIP/IFORS International Conference on Control in Transportation Systems*, Baden-Baden, Germany, April 1983.
- [Ehre85] Ehrenberger, W., "Statistical Testing of Real Time Software," *Verification and Validation of Real Time Software*, W.J. Quirk (ed.), Springer-Verlag, New York, 1985, pp. 147-178.
- [Ehr190] Ehrlich, W.K., Lee, K., and Molisani, R.H., "Applying Reliability Measurements: A Case Study," *IEEE Software*, March 1990.
- [Ehr190a] Ehrlich, W.K., Stampfel, J.P., and Wu, J.R., "Application of Software Reliability Modeling to Product Quality and Test Process," *Proceedings of the 12th International Conference on Software Engineering*, Nice, France, March 1990.
- [Ehr193] Ehrlich, W.K., Prasanna, B., Stampfel, J.P., and Wu, J.R., "Determining the Cost of a Stop-Test Decision," *IEEE Software*, March 1993, pp. 33-42.
- [Eick92] Eick, S.G., Steffen, J.L., and Sumner, E.E., "SeeSoft—A Tool for Visualizing Line-Oriented Software Statistics," *IEEE Transactions on Software Engineering*, vol. 18 no. 11, 1992, pp. 957-968.
- [Eick94] Eick, S.G., and Lucas, P.J., "Visualizing Run-Time Errors in Program Code," *Proceedings of the 5th International Symposium on Software Reliability Engineering*, Industrial Track, 1994.

- [Elma90] Elman, J., "Finding Structure in Time," *Cognitive Science*, no. 14, 1990, pp.179-211.
- [Endr75] Endres, A., "An Analysis of Errors and Their Causes in System Programs," *Proceedings of the International Conference on Software Engineering*, April 1975, pp.327-336.
- [ESA88] European Space Agency, "Software Reliability Modeling Study," Invitation to tender AO/I-2039/87/NL/IW, February 1988.
- [Ever90] Everett, W.W., Furlong, R.J., Klinger, D.J., Tortorella, L.M., and Vanderbei, K.S., *Reliability by Design*, Customer Information Center Code 010-810-105, Chap. 4, November 1990.
- [Ever92] Everett, W.W., "An 'Extended Execution Time' Software Reliability Model," *Proceedings of the Third International Symposium on Software Reliability Engineering*, IEEE Computer Press, 1992, pp. 14-22.
- [Ever93] Everett, W.W., and Musa, J.D., "A Software-Reliability Engineering Practice," *IEEE Computer*, vol. 26, no. 3, 1993, pp. 77-79.
- [Faga76] Fagan, M.E., "Design and Code Inspections to Reduce Errors in Program Development," *IBM Systems Journal*, vol. 15, no. 3, 1976, pp. 219-248.
- [Faga86] Fagan, M.E., "Advances in Software Inspections," *IEEE Transactions on Software Engineering*, July 1986, pp. 744-751.
- [Fahl88] Fahlman, S., "Faster-Learning Variations on Back-Propagation: An Empirical Study," *Proceedings of the Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski (eds.), Morgan Kaufmann, San Francisco, Calif, 1988, pp. 38-51.
- [Fahl90] Fahlman, S, and Lebiere, C., "The Cascaded-Correlation Learning Architecture," *Advances in Neural Information Processing Systems 2*, D. Touretzky (ed.), Morgan Kaufmann, San Francisco, Calif, 1990, pp. 524-532.
- [Fair85] Fairley, R., *Software Engineering Concepts*, McGraw-Hill, N.Y., 1985.
- [Farr83] Farr, W.H., *A Survey of Software Reliability Modeling and Estimation*, NSWC TR-171, Naval Surface Warfare Center, September 1983.
- [Farr88] Fair, W.H., and Smith, O.D., "Statistical Modeling and Estimation of Reliability Functions for Software (SMERFS) User's Guide," TR 84-373, Revision 1, NSWC December 1988.
- [Farr93a] Farr, W.H., and Smith O., *Statistical Modeling and Estimation of Reliability Functions for Software (SMERFS) User's Guide*, NSWCDD TR 84-373, Revision 3, Naval Surface Warfare Center Dahlgren Division, September, 1993.
- [Farr93b] Farr, W.H., and Smith O., *Statistical Modeling and Estimation of Reliability Functions for Software (SMERFS) Access Guide*, NSWC TR 84-371, Revision 3, Naval Surface Warfare Center Dahlgren Division, September, 1993.
- [FCC92] Federal Communications Commission, "Notification by Common Carriers of Service Disruptions," 47 CFR Part 63, Federal Register, vol. 57, no. 44, March 5, 1992, pp.7883-7885.
- [Fent91] Fenton, N.E., *Software Metrics: A Rigorous Approach*, Chapman and Hall, New York, 1991.
- [Fran88] Frankel, E.G., *Systems Reliability and Risk Analysis*, second revised edition, Kluwer Academic Publishers, 1988.

- [Fran93] Franklin, P.H., "Software-Reliability Prediction in a Multiple-Processor Environment," *Proceedings of the 1993 Reliability and Maintainability Symposium*, Atlanta, Georgia, January 1993.
- [Frie91] Fries, R.C., *Reliability Assurance for Medical Devices, Equipment and Software*. Interpharm Press, Inc., Buffalo Grove, 111., 1991.
- [Frie93] Friedman, M.A., "Automated Software Fault-Tree Analysis of Pascal Programs," *Proceedings of the 1993 Reliability and Maintainability Symposium*, Atlanta, Georgia, January, 1993, pp. 458-461.
- [Frul84] Frullini, R., and Lazzari, A., "Use of Microprocessor in Fail-Safe on Board Equipment," *Proceedings of the International Conference on Railway Safety Control and Automation Towards 21st Century*, London, U.K., September 1984, pp. 292-299.
- [Gan88] Gaffney, J.E., and Davis, C.F., "An Approach to Estimating Software Errors and Availability," SPC-TR-88-007, version 1.0, March 1988, *Proceedings of the Eleventh Minnowbrook Workshop on Software Reliability*, July 1988.
- [Gaff90] Gaffney, J.E., and Pietrolewicz, J., "An Automated Model for Software Early Error Prediction (SWEEP)," *Proceedings of the 13th Minnowbrook Workshop on Software Reliability*, July 1990.
- [Gant91] Gantenbein, R.E., Shin, S.Y., and Cowles, J.R., "Evaluation of Combined Approaches to Distributed Software-Based Fault Tolerance," *Pacific Rim International Symposium on Fault Tolerant Systems*, 1991, pp. 70-75.
- [GA092] United States General Accounting Office, *Patriot Missile Defense*, GAO, IMTEC 92-26, February 1992.
- [Gaud90] Gaudoin, O., "Statistical Tools for Software Reliability Evaluation," Ph.D. thesis, Joseph Fournier Univ. Grenoble I, December 1990; in French.
- [Geis90] Geist, R., and Trivedi, K., "Reliability Estimation of Fault-Tolerant Systems: Tools and Techniques," *IEEE Computer*, July 1990, pp. 52-61.
- [Geph78] Gephart, L.S., Greenwald, C.M., Hoffinan, M.M., and Osterfeld, D.H., *Software Reliability: Determination and Prediction*, Air Force Flight Dynamics Laboratory Technical Report, AFFDL-TR-78-77, June 1978.
- [Gers91] Gersting, J., Nist, R.L., Roberts, D.R., Van Valkenburg, R.L., "A Comparison of Voting Algorithms for N-Version Programming," *Proceedings of the 24th Annual Hawaii International Conference on System Sciences*, vol. II, 1991, pp. 253—262. Reprinted in *Fault-Tolerant Software Systems: Techniques and Applications*, Hoang Pham (ed.), IEEE Computer Society Press, 1992, pp. 62-71.
- [Gert94] Gertman, D., and Blackman, H.S., *Human Reliability and Safety Analysis Data Handbook*, John Wiley & Sons, Inc., New York, 1994.
- [Ghez91] Ghezzi, C., Jazayeri, M., and Mandrioli, D., *Fundamentals of Software Engineering*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [Gibb94] Gibbs, W.W., "Software's Chronic Crisis," in *Scientific American*, vol. 271, September 1994, pp. 86-95.
- [Girg86] Girgis, M.R., and Woodward, M.R., "An Experimental Comparison of the Error Exposing Ability of Program Testing Criteria," *Proceedings of the Workshop on Software Testing, Validation, and Analysis*, Banff, Canada, July 15-17, 1986.
- [Gmei79] Gmeiner, L., and Voges, U., "Software Diversity in Reactor Protection Systems: An Experiment," *Proceedings of the IFAC Workshop SAFECOMP'79*, 1979, pp. 75-79.

- Gnedenko, B.V., Belyayev, Y.K., and Solovyev, A.D., *Mathematical Methods of Reliability Theory*, Academic Press, New York, 1969.
- GNU, TkGnats Package, Free Software Foundation Inc., 1993.
- Goel, A.L., "Software Reliability Models: Assumptions, Limitations, and Applicability," *IEEE Transactions on Software Engineering*, vol. 11, no. 12 1985, pp.1411-1423.
- Goel, A.L., and Okumoto, K., "Time-Dependent Error -Detection Rate Model for Software and Other Performance Measures," *IEEE Transactions on Reliability*, vol R-28, no.5. August 1979, pp. 206-211.
- Goyal, A., Lavenberg, S.S., and Trivedi, K.S., "Probabilistic Modeling of Computer System Availability," *Annals of Operations Research*, no. 8, March 1987, pp.285-306.
- Graden, M.E., Horsley, P.S., and Pingel, T.C., "The Effects of Software Inspections on a Major Telecommunications Project," *AT&T Technical Journal*, vol. 65, issue 3, 1986 pp. 32-40.
- Grady, R.B., and Caswell, D.L., *Software Metrics: Establishing A Company-Wide Program*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- Grady, R.B., *Practical Software Metrics for Project Management and Process Improvement*, Prentice-Hall, Englewood Cliffs, New Jersey, 1992.
- Gray, J., "Why Do Computers Stop and What Can We Do about It?" Tandem Technical Report 85.7, June 1985.
- Gray, J.N., "Why Do Computers Stop and What Can Be Done About It?" *Proceedings of the 5th Symposium on Reliability in Distributed Software and Database Systems*, Los Angeles, January 1986, pp. 3-12.
- Gray, J., "A Census of Tandem System Availability Between 1985 and 1990," *IEEE Transactions on Reliability*, vol. 39, no. 4, October 1990, pp. 409-418.
- Griewank, A., and Corliss, G.F. (eds.), *Automatic Differentiation of Algorithms—Theory, Implementation and Application*, SIAM, Philadelphia, 1992.
- Gmarov, A., Arlat, J., and Avizienis, A., "On the Performance of Software Fault Tolerance Strategies," *Proceedings of the 10th IEEE International Symposium on Fault-Tolerant Computing (FTCS-10)*, Kyoto, Japan, October 1980, pp.251-253.
- Gross, D., and Miller, D.R., "The Randomization Technique as a Modeling Tool and Solution Procedure for Transient Markov Processes," *Operations Research*, vol. 32, no. 2, 1984, pp. 343-361.
- Hagelin, G. "ERICSSON Safety System for Railway Control," in [Vog87a], 1987, pp.11-21.
- Halstead, M.I., *Elements of Software Science*, Elsevier North-Holland, New York, 1977.
- Hamilton, P.A., and Musa, J.D., "Measuring Reliability of Computation Center Software," *Proceedings of the Third International Conference on Software Engineering*, 1978, pp.29-36.
- Hamlet, D., and Taylor, R., "Partition Testing Does Not Inspire Confidence," *IEEE Transactions on Software Engineering*, vol. SE-16, no. 12, 1990, pp. 1402-1411.
- Hamlet, D., and Voas, J., "Faults on Its Sleeve: Amplifying Software Reliability Testing," *Proceedings of the 1993 International Symposium on Software Testing and Analysis*, Cambridge, Mass., June 1993, pp. 89—98.

- [Hand66] *Handbook of Tables for Probability and Statistics*, Chemical Rubber Company, Cleveland, 1966.
- [Hans92] Hansen, J.R., and Siewiorek, D.R., "Models for Time Coalescence in Event Logs," *Proceedings of the 22nd International Symposium on Fault-Tolerant Computing*, July 1992, pp. 221-227.
- [Harr90] Harrington, P.V., "Applying Customer-Oriented Quality Metrics," *IEEE Software* November 1989.
- [Hech79] Hecht, H., "Fault-Tolerant Software," *IEEE Transactions on Reliability*, vol. R-28, August 1979, pp. 227-232.
- [Hech86] Hecht, H., and Hecht, M., "Fault-Tolerant Software," *Fault-Tolerant Computing: Theory and Techniques*, D.K. Pradhan (ed.), vol. 2, Prentice-Hall, Englewood Cliffs, New Jersey, 1986, pp. 658-696.
- [Hech87] Hecht, H., and Fiorentino, E., "Reliability Assessment of Spacecraft Electronics," *Proceedings of the Annual Reliability and Maintainability Symposium*, 1987, pp. 341-346.
- [Henl82] Henley, E.J., and Kumamoto, H., *Probabilistic Risk Assessment*, IEEE Press, New York, 1982.
- [Henr91] Henry, S.M., and Wake, S., "Predicting Maintainability with Software Quality Metrics," *Software Maintenance: Research and Practice*, vol. 3, 1991, pp. 129-143.
- [HU183] Hills, A.D., "A310 Slat and Flap Control System Management and Experience," *Proceedings 5th DASC*, November 1983.
- [Hoad81] Hoadely, B., "Quality Measurement Plan (QMP)," *Bell Systems Technical Journal*, vol. 60, no. 2, 1981, pp. 215-273.
- [Hoad86] Hoadely, B., "Quality Measurement Plan (QMP)," *Encyclopedia of Statistical Sciences*, Kotz, Johnson, and Read (eds.), John Wiley & Sons, Inc., New York, 1986, pp. 393-398.
- [Hogg83] Hogg, R.V., and Tanis, E.A., *Probability and Statistical Inference*, 2d ed., Macmillan Publishing Co., Inc., 1983.
- [Holl74] Hollander, M., and Proschan, F., "A Test for Superadditivity for the Mean Value Function of a Non-Homogeneous Poisson Process," *Stochastic Procedure and Their Applications*, vol. 2, 1974, pp. 195-209.
- [Holl78] Hollander, M., "Testing Whether More Failures Occur Later," *Proceedings of the Annual Reliability and Maintainability Symposium*, 1978, pp. 103-106.
- [Hopf86] Hopfield, J., and Tank, D., "Computing with Neural Circuits: A Model," *Science*, vol. 23, August 1986, pp. 625-633.
- [Horg91] Horgan, J.R., and London, S.A., "Dataflow Coverage and the C Language," *Proceedings of the Fourth Annual Symposium on Testing, Analysis, and Verification*, Victoria, British Columbia, Canada, October 1991, pp. 87-97.
- [Horg92] Horgan, J.R., and Mathur, A.P., "Assessing Tools in Research and Education," *IEEE Software*, May 1992, pp. 61-69.
- [Horg94] Horgan, J.R., London, S.A., and Lyu, M.R., "Achieving Software Quality with Testing Coverage Measures," *IEEE Computer*, vol. 27, no. 9, September 1994, pp. 60-69.
- [Howa71] Howard, R.A., *Dynamic Probabilistic Systems*, John Wiley & Sons, Inc., New York, 1971.



- Howden, W.E., "Functional Testing," *IEEE Transactions on Software Engineering*, vol. SE-6, no. 2, March 1980, pp. 162-169.
- Hsueh, M.C., and Iyer, R.K., "A Measurement-Based Model of Software Reliability in a Production Environment," *Proceedings of the 11th Annual International Computer Software & Applications Conference*, October 1987, pp. 354-360.
- Hsueh, M.C., Iyer, R.K., and Trivedi, K.S., "Performability Modeling Based on Real Data: A Case Study," *IEEE Transactions on Computers*, vol. 37, no. 4, April 1988, pp. 478-484.
- Huang, K-H., and Abraham, J.A., "Algorithms-Based Fault-Tolerance for Matrix Operations," *IEEE Transactions on Computers*, vol. C-33, 1984, pp. 518-528.
- Huang, Y., and Kintala, C., "Software Implemented Fault Tolerance: Technologies and Experience," *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing*, June 1993, pp. 2-9.
- Hudak, J., Suh, B., Siewiorek, D., and Segall, Z., "Evaluation and Comparison of Fault-Tolerant Software Techniques," *IEEE Transactions on Reliability*, vol. 42, no. 2, June 1993, pp. 190-204.
- Hudepohl, J., Snipes, W., Jones, W.D., and Hollack, T., "A Methodology to Improve Switching System Software Service Quality and Reliability," *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, December 1992.
- Hull, T.E., Fairgrieve, T.F., Tang, P.T.P., "Implementing Complex Elementary Functions Using Exception Handling," *ACM Transactions on Mathematical Software* vol. 20, no. 2, June 1994.
- Humphrey, W.S., *Managing the Software Process*, Addison-Wesley Publishing Company, Reading, Mass., 1989.
- Ibe, O.C., and Wein, A.S., "Availability of Systems with Partially Observable Failures," *IEEE Transactions on Reliability*, IEEE, vol. 41, no. 1, March 1992, pp. 92-96.
- IBM MHVPL PTM Work Group, *Requirements for Program Trouble Memo-randas*, IBM Mid Hudson Valley Programming Lab, Myers Corners Road, Wappingers Falls, N.Y., 1990.
- IBM, "ODC Data Requirements and Definitions," working document, ODC Project, IBM Watson Research, 1995.
- IEEE *Computer*, "Special issue on fault-tolerant computing," vol. 17, no. 8, 1984.
- American National Standards Institute/Institute of Electrical and Electronics Engineers: *A Standard for Binary Floating-Point Arithmetic*, ANSI/IEEE Std. 754-1985, New York, 1985 (reprinted in *SIGPLAN* vol. 22, no. 2, 1987, pp. 9-25).
- IEEE Std. 1012-1986, IEEE Standard, *Software Verification and Validation Plans*, IEEE 1986.
- American National Standards Institute/Institute of Electrical and Electronics Engineers: *A Standard for Radix-Independent Floating-point Arithmetic*, ANSI/IEEE Std. 854-1987, New York, 1987.
- IEEE Standard, "A Standard Classification of Software Errors, Faults and Failures," *Technical Committee on Software Engineering, Standard P-1044 IDS*, December 1987.

- [IEEE88a] IEEE Standard, *Dictionary of Measures to Produce Reliable Software* (ANSI), IEEE Std-982.1-1988, IEEE, New York, 1988.
- [IEEE88b] IEEE, *Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software* (ANSI), IEEE Std-982.2-1988, IEEE, New York, 1988.
- [IEEE90] IEEE Standard, *Glossary of Software Engineering Terminology* (ANSI) IEEE Std-610.12-1990, IEEE, New York, 1990.
- [IEEE90a] IEEE, "Theme Articles: Metrics," *IEEE Software*, vol. 7, no. 2, March 1990.
- [IEEE900] IEEE Standards, "Standard for Software Quality Metrics Methodology," *Software Engineering Standards Committee, Standard P-1061 ID21*, April 1990.
- [IEEE91] Institute of Electrical and Electronics Engineers, *ANSI/IEEE Standard Glossary of Software Engineering Terminology*, IEEE Std. 729-1991, 1991.
- [IEEE94] IEEE, "Special Issue on Safety-Critical Systems," *IEEE Software*, J. Knight and B. Littlewood (eds.), January 1994.
- [IEEE95] IEEE, *Charter and Organization of the Software Reliability Engineering Committee*, 1995.
- [Ishi91] Ishikawa, K., *Introduction to Quality Control*, 3A Corporation, Tokyo, 1991.
- [ISO87] ISO, "International Standard ISO 9001: Quality Systems—Model for Quality Assurance in Design/Development, Production, Installation and Servicing," /ISO 9002, Switzerland, 1987.
- [ISO91] ISO, "Quality Management and Quality Assurance Standards—Part 3: Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software," *ISO 9000-3*, Switzerland, June 1991.
- [Iyer82a] Iyer, R.K., and Rossetti, D.J., "A Statistical Load Dependency Model for CPU Errors at SLAC," *Proceedings of the 12th International Symposium on Fault-Tolerant Computing* June 1982, pp. 363-372.
- [Iyer82b] Iyer, R.K., Butner, S.E., and McCluskey, E.J., "A Statistical Failure/Load Relationship: Results of a Multi-Computer Study," *IEEE Transactions on Computers*, vol. C-31, July 1982, pp. 697-706.
- [Iyer85a] Iyer, R.K., and Velardi, P., "Hardware-Related Software Errors: Measurement and Analysis," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 2, February 1985, pp. 223-231.
- [Iyer85b] Iyer, R.K., and Rossetti, D.J., "Effect of System Workload on Operating System Reliability: A Study on IBM 3081," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 12, December 1985, pp. 1438-1448.
- [Iyer86] Iyer, R.K., Rossetti, D.J., and Hsueh, M.C., "Measurement and Modeling of Computer Reliability as Affected by System Activity," *ACM Transactions on Computer Systems*, vol. 4, no. 3, August 1986, pp. 214-237.
- [Iyer90] Iyer, R.K., Young, L.T., and Iyer, P.V.K., "Automatic Recognition of Intermittent Failures: An Experimental Study of Field Data," *IEEE Transactions on Computers*, vol. 39, no. 4 April 1990, pp. 525-537.
- [Jaco88] Jacobs, R., "Increased Rates of Convergence Through Learning Rate Adaptation," *Neural Networks*, vol. 1, 1988, pp. 295-307.
- [Jaco91] Jacob, J., "The Basic Integrity Theorem," *Proceedings of the IEEE International Symposium on Security and Privacy*, Oakland, Calif., May 1991, pp. 89-97.

- [Jeli72] Jelinski, Z., and Moranda, P.B.: "Software Reliability Research," *Proceedings of the Statistical Methods for the Evaluation of Computer System Performance*, Academic Press, 1972, pp. 465-484.
- [Jewe85] Jewell, W.S., "Bayesian Extensions to a Basic Model of Software Reliability," *IEEE Transactions on Software Engineering*, SE-11, 1985, pp. 1465-1471.
- [JohK82] Johnson, N., and Kotz, S. (eds.), *Encyclopedia of Statistical Sciences*, Wiley & Sons Inc., New York, 1982.
- [John82] Johnson, R.A., and Wichern, D.W., *Applied Multivariate Statistical Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [John88] Johnson, A.M., and Maiek, M., "Survey of Software Tools for Evaluating Reliability, Availability, and Serviceability," *ACM Computing Surveys*, vol. 20, no. 4, pp. 227-269. December 1988.
- [Jone91] Jones, W.D., "Reliability Modeling for Large Software Systems in Industry," *Proceedings of the Second International Symposium on Software Reliability Engineering*, May 1991, pp. 35-42.
- [Jone92] Jones, W.D., "Reliability of Telecommunications Software: Assessing Sensitivity of Least Squares Reliability Estimates," *Proceedings of TRICOM*, February 1992.
- [Jone93] Jones, W.D., and Gregory, D., "Infinite Failure Models for a Finite World: A Simulation of the Fault Discovery Process," *Proceedings of the Fourth International Symposium on Software Reliability Engineering*, November 1993, pp. 284-293.
- [Jord86] Jordan, M., "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine," *Proceedings of the 8th Annual Conference of the Cognitive Science*, 1986. pp. 531-546.
- [JPL91] *Handbook For Software Reliability Measurement*, Software Product Assurance Section Jet Propulsion Laboratory, JPLD-8672, June 20, 1991.
- [Juhl92a] Juhlin, B.D., "Implementing Operational Profiles to Measure System Reliability," *Proceedings of the 3rd International Symposium on Software Reliability Engineering* Research Triangle Park, North Carolina, October 7-10, 1992, pp. 286-295.
- [Juhl92b] Juhlin, B.D., "Applying Software Reliability Engineering to International PBX Testing." Presentations at the *9th International Conference on Testing Computer Software*, Washington, D.C., June 16-18, 1992, pp. 165-176.
- [Juhl93] Juhlin, B.D., "Software Reliability Engineering in the System Test Process," presentations at the *10th International Conference on Testing Computer Software* Washington, D.C., June 14-17, 1993, pp. 97-115.
- [Kan91] Kan, S.H., "Modeling and Software Development Quality," *IBM Systems Journal*, vol. 30, no. 3, 1991, pp. 351-362.
- [Kana95] Kanawati, A., Kanawati, N., and Abraham, J., "FERRARI: A Flexible Software-Based Fault and Error Injection System," *IEEE Transactions on Computer*, vol. 44, no. 2. February 1995.
- [Kano87] Kanoun, K., and Sabourin, T., "Software Dependability of a Telephone Switching System," *Proceedings of the 17th IEEE International Symposium on Fault-Tolerant Computing (FTCS-17)*, Pittsburgh, June 1987, pp. 236-241.
- [Kano88] Kanoun, K., Laprie, J.-C., and Sabourin, T., "A Method for Software Reliability Growth Analysis and Assessment," *Proceedings of the 1st Interna-*

- tional Workshop on Software Engineering and Its Applications*, Toulouse, Dec. 1988, pp. 859-878.
- [Kano89] Kanoun, K., "Software Dependability Growth—Characterization, Modeling, Evaluation," Doctor es-Sciences thesis, Toulouse Polytechnic National Institute, September 89 LAAS Report no. 89.320; in French.
- [Kano91a] Kanoun, K., Bastos Martini, M., and Moreira De Souza, J., "A Method for Software Reliability Analysis and Prediction—Application to the TROP-ICO-R Switching System," *IEEE Transactions on Software Engineering*, April 1991, pp. 334-344.
- [Kano91b] Kanoun, K., and Laprie, J.-C., "The Role of Trend Analysis in Software Development and Validation," *Proceedings IFAC International Conference on Safety, Security and Reliability (SAFECOMP'91)*, 30 October-1 November, 1991, Trondheim, Norway.
- [Kano92] Kanoun, K., Kaaniche, M., Laprie, J.-C., and Metge, S., "SoRel: A Tool for Reliability Growth Analysis and Prediction from Statistical Failure Data," Centre National de la Recherche Scientifique, LAAS Report 92.468, December 1992. Also in *FTCS-23 Proceedings*, Toulouse, France, June 1993, pp. 654-659.
- [Kano93a] Kanoun, K., Kaaniche, M., Beounes, C., Laprie, J.-C., and Arlat, J., "Reliability Growth of Fault-Tolerant Software," *IEEE Transactions on Reliability*, vol. 42, no. 2, June 1993, pp. 205-219.
- [Kano93b] Kanoun, K., Kaaniche, M., and Laprie, J.-C., "Experience in Software Reliability: From Data Collection to Quantitative Evaluation," *Proceedings of the Fourth International Symposium on Software Reliability Engineering (ISSRE'93)*, Denver, Colo., November 1993, pp. 234-246.
- [Kao93] Kao, W., Iyer, R., and Tang, D., "FINE: A Fault Injection and Monitoring Environment of Tracing the UNIX System Behavior under Faults," *IEEE Transactions on Software Engineering*, vol. 19, no. 11, November 1993, pp. 1105-1118.
- [Kap185] Kaplan, G., "The X-25: Is It Coming or Going?," *IEEE Spectrum*, vol. 22, no. 6, 1985, pp. 54-60.
- [Kare90] Kareer, N., Kapur, P.K., and Grover, P.S., "An S-Shaped Software Reliability Growth Model with Two Types of Errors," *Microelectronics and Reliability*, vol. 30, no. 6, 1990, pp. 1085-1090.
- [Karu91] Karunanithi, N., Malaiya, Y.K., and Whitley, D., "Prediction of Software Reliability Using Neural Networks," *Proceedings of the 2nd International Symposium on Software Reliability Engineering*, IEEE Computer Society Press, May 1991, pp. 124-130.
- [Karu92a] Karunanithi, N., Whitley, D., and Malaiya, Y.K., "Prediction of Software Reliability Using Connectionist Approaches," *IEEE Transactions on Software Engineering*, vol. 18, no. 7, July 1992, pp. 563-574.
- [Karu92b] Karunanithi, N., Whitley, D., and Malaiya, Y.K., "Using Neural Networks in Reliability Prediction," *IEEE Software*, vol. 9, no. 4, July 1992, pp. 53-59.
- [Karu92c] Karunanithi, N., and Malaiya, Y.K., "The Scaling Problem in Neural Network for Software Reliability Prediction," *Proceedings of the Third International Symposium on Software Reliability Engineering*, IEEE Computer Society Press, October 1992, pp. 76-82.
- [Karu92d] Karunanithi, N., "Generalization in the Cascade-Correlation Architecture: Some Experiments and Applications," Ph.D. dissertation, Computer Science Department, Colorado State University, Fort Collins, Colo., December 1992.

- [Karu93a] Karunanithi, N., "A Neural Networks Approach for Software Reliability Growth Modeling in the Presence of Code Churn," *Proceedings of the 4th International Symposium on Software Reliability Engineering*, Denver, Colo., November 3-6, 1993, pp. 310-317.
- [Karu93b] Karunanithi, N., "Identifying Fault-Prone Software Modules Using Connectionist Networks," *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications, (IWANN'T'93)*, October 18-20, 1993, pp. 266-272.
- [Karu93c] Karunanithi, N., "Identifying Fault-Prone Software Modules Using Feed-Forward Networks: A Case Study," *Advances in Neural Information Processing Systems 6*, J. Cowan, G. Tesaruo, and J. Alspector (eds.), Morgan Kaufmann, San Francisco, Calif, 1993, pp. 793-800.
- [Keen92] Keene, S., and Chris, L., "Combining Software and Hardware Aspects of Reliability," *Quality and Reliability Engineering International*, vol. 8, no. 4, 1992, pp. 419-426.
- [Keil83] Keiller, PA., Littlewood, B., Miller, D.R., and Sofer, A., "Comparison of Software Reliability Predictions," *Proceedings of the 13th IEEE International Symposium on Fault-Tolerant Computing (FTCS-13)*, Milano, Italy, June 1983, pp. 128-134.
- [Kell86] Kelly, J.P.J., Avizienis, A., Ulery, B.T., Swain, B.J., Lyu, R.T., Tai, A.T., and Tso, K.S., "Multiversion Software Development," *Proceedings IFAC Workshop SAFECOMP'86*, Sarlat, France, October 1986, pp. 43-49.
- [Kell88] Kelly, J., Eckhardt, D., Caglayan, A., Knight, J., McAllister, D., and Vouk, M., "A Large Scale Second Generation Experiment in Multi-Version Software: Description and Early Results," *Proceedings of the 18th International Symposium on Fault-Tolerant Computing*, Tokyo, Japan, June 1988, pp. 9-14.
- [Kend77] Kendall, M.G., *The Advanced Theory of Statistics*, Oxford University Press, 1977.
- [Kenn92] Kenney, G.Q., and Vouk, M.A., "Measuring the Field Quality of Wide-Distribution Commercial Software," *Proceedings of the 3rd IEEE International Symposium on Software Reliability Engineering (ISSRE'92)*, Research Triangle Park, North Carolina, October 1992, pp. 351-357.
- [Kenn93a] Kenney, G.Q., "Estimating Defects in Commercial Software During Operational Use," *IEEE Transactions on Reliability*, vol. 42, no. 1, 1993, pp. 107-115.
- [Kenn93b] Kenney, G.Q., "The Next Release Effect in the Field Defect Model for Commercial Software," Ph.D. dissertation, North Carolina State University, Department of Computer Science, 1993.
- [Khos89] Khoshgoftaar, T.M., and Woodcock, T.G., "A Simulation Study of the Performance of the Akaike Information Criterion for the Selection of Software Reliability Growth Models," *Proceedings of the 27th Annual South East Region ACM Conference*, April 1989, pp. 419-423.
- [Khos90] Khoshgoftaar, T.M., and Munson J.C., "Predicting Software Development Errors Using Software Complexity Metrics," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 2, February 1990, pp. 253-261.
- [Khos91] Khoshgoftaar, T.M., and Woodcock, T.G., "Software Reliability Model Selection: A Case Study," *Proceedings of the Second International Symposium on Software Reliability Engineering*, May 1991, pp. 183-191.

- [Khos92a] Khoshgoftaar, T.M., and Munson, J.C., "A Measure of Software System Complexity and Its Relationship to Faults," *Proceedings of the 1992 International Simulation Technology Conference*, Houston, Texas, November 1992, pp.267-272.
- [Khos92b] Khoshgoftaar, T.M., and Munson, J.C., "Applications of a Relative Complexity Metric for Predicting Source Code Complexity at the Design Phase," *Proceedings of the Second Software Engineering Research Forum*, Melbourne, Florida, November 1992, pp 191-198.
- [Khos92c] Khoshgoftaar, T.M., Munson, J.C., Bhattacharya, B.B., and Richardson, G., "Predictive Modeling Techniques of Software Quality from Software Measures," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, November 1992, pp.979-987.
- [Khos92d] Khoshgoftaar, T.M., Pandya, A.S., and More, H.B., "A Neural Network Approach for Predicting Software Development Faults," *Proceedings of the Third International Symposium on Software Reliability Engineering*, IEEE Computer Society Press, October 1992, pp. 83-89.
- [Khos93a] Khoshgoftaar, T.M., Munson, J.C., and Lanning, D.L., "Dynamic System Complexity," *Proceedings of the IEEE-CS International Software Metrics Symposium*, Baltimore, Md., May 1993, pp. 129-140.
- [Khos93b] Khoshgoftaar, T.M., Munson, J.C., and Lanning, D.L., "A Comparative Study of Predictive Models for Program Changes During System Testing and Maintenance," *Proceedings of the Conference on Software Maintenance*, Montreal, Quebec, Canada September 1993, pp. 72-79.
- [Khos93c] Khoshgoftaar, T.M., Lanning, D.L., and Pandya, A.S., "A Neural Network Modeling Methodology for the Detection of High-Risk Programs," *Proceedings of the Fourth International Symposium on Software Reliability Engineering*, Denver, Colo., November 1993, pp.302-309.
- [Khos94a] Khoshgoftaar, T.M., Lanning, D.L., and Pandya, A.S., "A Comparative Study of Pattern Recognition Techniques for Quality Evaluation of Telecommunications Software," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 2, February 1994, pp. 279-291.
- [Khos94b] Khoshgoftaar, T.M., Munson, J.C., and Lanning, D.L., "Alternative Approaches for the Use of Metrics to Order Programs by Complexity," *Journal of Systems and Software*, vol. 24, no. 3, March 1994, pp. 211-221.
- [Kim89] Kim, K.H., and Welch, H.O., "Distributed Execution of Recovery Blocks: An Approach for Uniform Treatment of Hardware and Software Faults in Real-Time Applications," *IEEE Transactions on Computers*, vol. 38, no. 5, May 1989, pp.626-636.
- [Kits93] Kitson, D., and Masters, S., "An Analysis of SEI Software Process Results: 1987-1991," *Proceedings of the Fifteenth International Conference on Software Engineering*, IEEE Computer Society Press, 1993, pp. 68-77.
- [Klat91] Klatte, R., Kulisch, U., Neaga, M., Ratz, D., and Ullrich, C., *PASCAL-XSC—Language Description with Examples*, Springer-Verlag, New York, 1991.
- [Klat92] Klatte, R., Kulisch, U., Lawo, C., Rauch, M., and Wiethoff, A., *C-XSC*, Springer-Verlag, New York, 1992.
- [Knaf92] Knaf, G., "Overview of the Software Reliability Models, Estimation Procedures, and Predictive Performance Measures," lecture notes for Tutorial Track 2a for the *Third International Symposium on Software Reliability Engineering*, October 1992.



- Knight, J.C., and Leveson, N.G., "An Experimental Evaluation of the Assumption of Independence in Multiversion Programming," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 1, 1986, pp. 96-109.
- Knuth, D., *The Art of Computer Programming: Semi-Numerical Algorithms*, Addison-Wesley Book Company, Reading, Mass., 1970, pp. 550-554.
- Knuth, D.E., "A Torture Test for TEX," Technical Report No. STAN-CS-84-1027, Department of Computer Science, Stanford University, Stanford, Calif, 1984.
- Knuth, D.E., *TEX: The Program*, Addison-Wesley, Reading, Mass., 1986.
- Knuth, D.E., "The Errors of TEX," *Software Practice and Experience*, vol. 19, no. 7, July 1989, pp. 607-685.
- Kozlov, B.A., and Ushakov, U.A., *Reliability Handbook*, L.H. Koopmans and J. Rosenblat (eds.), Holt Rinehart and Winston, 1970.
- Kreutzer, W., *System Simulation: Programming Styles and Languages*, International Computer Science Series, Addison-Wesley Publishing Company, Menlo Park, Calif 1986.
- Kruger, G., "Project Management Using Software Reliability Growth Models," *Hewlett-Packard Journal*, June, 1988, pp. 30-35.
- Kulisch, U. and Miranker, W.L., *Computer Arithmetic in Theory and Practice*, Academic Press, New York, 1981.
- Kulisch, U., private communication, 1993.
- Kyparisis, J., and Singpurwalla, N.D., "Bayesian Inference for the Weibull Process with Applications to Assessing Software Reliability Growth and Predicting Software Failures," *Computer Science and Statistics*, 16th Symposium Interface, Atlanta, Georgia, 1984, pp. 57-64.
- Langberg, N., and Singpurwalla, N.D., "A Unification of Some Software Reliability Models," *SIAM Journal of Scientific and Statistical Computation*, vol. 6, no. 3, 1985, pp. 781-790.
- Laprie, J.-C., "Dependability Modeling and Evaluation of Hardware-and-Software Systems," *Proceedings of the 2nd GIINTGIGMR Conference on Fault-Tolerant Computing*, Bonn, Germany, September 1984, pp. 202-215.
- Laprie, J.-C., "Dependability Evaluation of Software Systems in Operation," *IEEE Transactions on Software Engineering*, vol. SE-10, no. 6, November 1984, pp. 701-714.
- Laprie, J.-C., "Dependable Computing and Fault Tolerance: Concepts and Terminology," *Proceedings of the 15th International Symposium on Fault-Tolerant Computing (FTCS-15)*, Ann Arbor, Michigan, 1985, pp. 2-11.
- Laprie, J.-C., Arlat, J., Beounes, C., Kanoun, K., and Hourtolle, C., "Hardware and Software Fault-Tolerance: Definition and Analysis of Architectural Solutions," *Proceedings of the 17th International Symposium on Fault-Tolerant Computing (FTCS-17)*, Pittsburgh, Pennsylvania, 1987, pp. 116-121.
- Laprie, J.-C., "Hardware-and-Software Dependability Evaluation," *Proceedings of the IFIP 11th World Congress*, San Francisco, Calif, August 1989, pp. 109-114.
- Laprie, J.-C., Arlat, J., Beounes, C., and Kanoun, K., "Definition and Analysis of Hardware- and Software-Fault-Tolerant Architectures," *IEEE Computer*, vol. 23, no. 7, July 1990, pp. 39-51. Reprinted in *Fault-Tolerant Software Systems: Techniques and Applications*, Hoang Pham (ed.), IEEE Computer Society Press, 1992, pp. 5-17.

- Laprie, J.-C., Beounes, C., Kaaniche, M., and Kanoun, K., "The Transformation Approach to the Modeling and Evaluation of the Reliability and Availability Growth," *Proceedings 20th IEEE International Symposium on Fault-Tolerant Computing (FTCS-20)*, IEEE Computer Society, 1990, pp. 364-371.
- Laprie, J.-C., Kanoun, K., Beounes, C., and Kaaniche, M., "The KAT (Knowledge-Action-Transformation) Approach to the Modeling and Evaluation of Reliability and Availability Growth," *IEEE Transactions on Software Engineering* vol. 17, no. 4 April 1991, pp. 370-382.
- Laprie, J.-C., *Dependability: Basic Concepts and Terminology, Dependable Computing and Fault-Tolerant Systems*, vol. 5, J.-C. Laprie (ed.), Springer-Verlag, Wien, New York, 1992.
- Laprie, J.-C., and Kanoun, K., "X-Ware Reliability and Availability Modeling," *IEEE Transactions on Software Engineering* vol. 18, no. 2, February 1992, pp. 130-147.
- Laprie, J.-C., "For a Product-in-a-Process Approach to Software Reliability Evaluation," *Proceedings of the 3rd International Symposium on Software Reliability Engineering ISSRE'92*, Research Triangle Park, North Carolina, October 1992, pp. 134-139.
- Laprie, J.-C., "Dependability: From Concepts to Limits," *Proceeding's SAFE-COMP'93*, Springer-Verlag, Poznan, Poland, 1993, pp. 157-168.
- Lee, P.A., and Anderson, T., *Fault Tolerance: Principles and Practices*, Springer-Verlag, New York, 1990.
- Lee, I., Iyer, R.K., and Tang, D., "Error/Failure Analysis Using Event Logs from Fault Tolerant Systems," *Proceedings 21st International Symposium on Fault-Tolerant Computing*, June 1991, pp. 10-17.
- Lee, I., and Iyer, R.K., "Analysis of Software Halts in Tandem System," *Proceedings of the 3rd International Symposium on Software Reliability Engineering*, October 1992, pp. 227-236.
- Lee, L., *The Day the Phones Stopped: How People Get Hurt When Computers Go Wrong*, Donald I. Fine, Inc., New York, 1992.
- Lee, I., Tang, D., Iyer, R.K., and Hsueh, M.C., "Measurement-Based Evaluation of Operating System Fault Tolerance," *IEEE Transactions on Reliability*, vol. 42, no. 2 June 1993, pp. 238-249.
- Lee, I., and Iyer, R.K., "Faults, Symptoms, and Software Fault Tolerance in the Tandem GUARDIAN90 Operating System," *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing*, Toulouse, France, June 1993, pp. 20-29.
- Lee, I., and Iyer, R.K., "Identifying Software Problems Using Symptoms," *Proceedings of the 24th International Symposium on Fault-Tolerant Computing*, San Antonio, Texas June 1994, pp. 320-329.
- Lee, I., "Software Dependability in the Operational Phase," Ph.D. dissertation. Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, October 1994.
- Lee, I., and Iyer, R.K., "Software Dependability in the Tandem GUARDIAN System," *IEEE Transactions on Software Engineering*, vol. 21, no. 5, May 1995, pp. 455-467.

- [LeGa90] LeGall, G., Adam, M.F., Derriennic, H., Moreau, B., and Valette, N., "Studies on Measuring Software," *IEEE Journal of Selected Areas in Communications*, vol. 8, no. 2, 1990, pp. 234-245.
- [Leve86] Leveson, N., "Software Safety: What, Why and How," *ACM Computing Surveys*, June 1986, pp. 125-164.
- [Leve87] Levendel, Y., "Quality and Reliability Estimation for Large Software Projects Using a Time-Dependent Model," *Proceedings of COMPSAC 87*, Tokyo, Japan, October 1987, pp. 340-346.
- [Leve89] Levendel, Y., "Defects and Reliability Analysis of Large Software Systems: Field Experience," *Proceedings of the 19th IEEE International Symposium on Fault-Tolerant Computing (FTCS-19)*, Chicago, June 1989, pp. 238-244.
- [Leve90] Levendel, Y., "Reliability Analysis of Large Software Systems: Defect Data Modeling," *IEEE Transactions on Software Engineering*, vol. SE-16, no. 2, February 1990, pp. 141-152.
- [Leve91] Levendel, Y., "Software Quality Improvement Process: When to Stop Testing," *Proceedings Software Engineering & Its Applications*, Toulouse, France, December 1991, pp. 729-749.
- [Leve91a] Leveson, N.G., "Safety Verification of Ada Programs Using Software Fault Trees," *IEEE Software*, July 1991, pp. 48-59.
- [Leve93] Levendel, Y., "Fault-Tolerance Cost Effectiveness," *Proceedings of the Third Workshop on Issues in Software Reliability Workshop Program*, U.S. West Advanced Technologies, Boulder, Colorado, November 1-2, 1993.
- [Leve95] Levendel, Y., "The Cost Effectiveness of Telecommunication Service Dependability," *Software Fault Tolerance*, Lyu, M.R. (ed.), Wiley & Sons, Inc., January 1995, pp. 279-314.
- [Lewi64] Lewis, P.A., "A Branching Poisson Process Model for the Analysis of Computer Failure Patterns," *Journal of Royal Statistical Society*, series B, vol. 26, no. 3, 1964, pp. 398-456.
- [Li93] Li, N., and Malaiya, Y., "Enhancing Accuracy of Software Reliability Prediction," *Proceedings of the 4th International Symposium on Software Reliability Engineering*, November 4-6, 1993, pp. 71-79.
- [Lin83] Lin, J., and Costello, Jr., D.J., *Error Control Coding—Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.
- [Lin90] Lin, T.T., and Siewiorek, D.P., "Error Log Analysis: Statistical Modeling and Heuristic Trend Analysis," *IEEE Transactions on Reliability*, vol. 39, no. 4, October 1990, pp. 419-432.
- [Lind89] Lind, R., and Vairavan, K., "An Experimental Investigation of Software Metrics and Their Relationship to Software Development Effort," *IEEE Transactions on Software Engineering*, vol. 15, no. 5, May 1989, pp. 649-653.
- [Litt73] Littlewood, B. and Verrall, J., "A Bayesian Reliability Growth Model for Computer Software," *Journal of the Royal Statistical Society*, series C, vol. 22, no. 3, 1973, pp. 332-346.
- [Litt74] Littlewood, B., and Verrall, V., "A Bayesian Reliability Model with a Stochastically Monotone Failure Rate," *IEEE Transactions on Reliability*, vol. R-23, no. 2, 1974, pp. 108-114.

- [Litt78] Littlewood, B., "Validation of a Software Model," Software Life Cycle Management Workshop, Atlanta, Georgia, 1978, published by the International Business Services Inc., pp. 146-152.
- [Litt79a] Littlewood, B., "How to Measure Software Reliability and How Not To," *IEEE Transactions on Reliability*, vol. R-28, no. 2, June 1979, pp. 103-110.
- [Litt79b] Littlewood, E., "A Software Reliability Model for Modular Program Structure," *IEEE Transactions on Reliability*, vol. R-28, no. 2, June 1979, pp. 241-246.
- [Litt80a] Littlewood, B., "The Littlewood-Verrall Model for Software Reliability Compared with Some Rivals," *Journal of Systems and Software*, vol. 1, no. 3, 1980, pp. 251-258.
- [Litt80b] Littlewood, B., "Theory of Software Reliability: How Good Are They and How Can They Be Improved?," *IEEE Transactions on Software Engineering*, vol. SE-6, 1980, pp. 489-500.
- [Litt80c] Littlewood, E., "A Bayesian Differential Debugging Model for Software Reliability," *Proceedings of the International Computer Software Applications Conference*, IEEE Computer Society, 1980, pp. 511-519.
- [Litt81] Littlewood, B., "Stochastic Reliability Growth: A Model for Fault-Removal in Computer Programs and Hardware Designs," *IEEE Transactions on Reliability*, vol. R-30, no. 4, October 1981, pp. 313-320.
- [Litt86] Littlewood, E., Abdel-Ghaly, A.A., and Chan, P.Y., "Tools for the Analysis of the Accuracy of Software Reliability Predictions," *Software System Design Methods*, Springer-Verlag, Heidelberg, 1986, pp. 299-335.
- [Litt87] Littlewood, I., and Miller, D.R., "A Conceptual Model of Multi-Version Software," *Proceedings of the 17th International Symposium on Fault-Tolerant Computing*, IEEE Computer Society Press, July 1987, pp. 150-155.
- [Litt87a] Littlewood, B., *Software Reliability: Achievement and Assessment*, Blackwell, Oxford, 1987.
- [Litt87b] Littlewood, E., "How Good are Software Reliability Predictions?" *Software Reliability: Achievement and Assessment*, Blackwell, Oxford, 1987, pp. 172-191.
- [Litt87c] Littlewood, B., and Sofer, A., "A Bayesian Modification to the Jelinski-Moranda Software Reliability Growth Model," *Journal of Software Engineering*, no. 2, 1987, pp. 30-41.
- [Litt88] Littlewood, B., "Forecasting Software Reliability," *Software Reliability Modeling and Identification*, S. Bittanti (ed.), Springer-Verlag, Berlin, 1988, pp. 140-209.
- [Litt89] Littlewood, E., and Miller, D.R., "Conceptual Modeling of Coincident Failures in Multiversion Software," *IEEE Transactions on Software Engineering*, vol. 15, no. 12, 1989, pp. 1596-1614.
- [Litt91] Littlewood, B., "Limits to Evaluation of Software Dependability," in *Software Reliability and Metrics*, B. Littlewood and N. Fenton (eds.), Elsevier, Amsterdam, 1991.
- [Litt93] Littlewood, B., and Strigini, L., "Validation of Ultra-High Dependability for Software-Based Systems," *Communications of the ACM*, vol. 36, no. 11, November 1993.
- [Liu87] Liu, G., "A Bayesian Assessing Method of Software Reliability Growth," *Reliability Theory and Applications*, S. Osaki and J. Cao (eds.), World Scientific, 1987, pp. 237-244.

- Lorzak, P.R., Caglayan, A.K., and Eckhardt, D.E., "A Theoretical Investigation of Generalized Voters for Redundant Systems," *Proceedings 19th Annual International Symposium on Fault-Tolerant Computing (FTCS-19)*, 1989, pp.444-451.
- Lu, M., Brocklehurst, S., and Littlewood, B., "Combinations of Predictions Obtained from Different Software Reliability Growth Models," *Proceedings 10th Annual Software Reliability Symposium*, 1992, pp. 24-33.
- Lyu, M.R., and Kinlaw, J., "ASAS Software Reliability Prediction: Methodology and Result," in *Proceedings ATTHPIJPL Software Reliability Measurement Technica Exchange Meeting*, Pasadena, Calif., June 1989.
- Lyu, M.R., and Nikora, A.P., "Software Reliability Measurements Through Combination Models: Approaches, Results, and a Case Tool," *Proceedings of the 15th Annual International Computer Software and Applications Conference (COMPSAC'91)*, Tokyo, Japan, September 1991, pp. 577-584.
- Lyu, M.R., "Measuring Reliability of Embedded Software: An Empirical Study with JPL Project Data," *Proceedings of the International Conference on Probabilistic Safety Assessment and Management*, Beverly Hills, Calif, February 1991, pp.493-500.
- Lyu, M.R., and Nikora, A.P., "A Heuristic Approach for Software Reliability Prediction: The Equally Weighted Linear Combination Model" *Proceedings of the 1991 International Symposium on Software Reliability Engineering*, IEEE Computer Society Press, May 1991, pp. 172-181.
- Lyu, M.R., "Software Reliability Measurements in N-Version Software Execution Environment," *Proceedings of the 1992 International Symposium on Software Reliability Engineering*, Raleigh, North Carolina, 1992, pp.254-263.
- Lyu, M.R., and Avizienis, A., "Assuring Design Diversity in N-Version Software: A Design paradigm for N-Version Programming," *Dependable Computing and Fault-Tolerant Systems*, J.F. Meyer and R.D. Schichting (eds.), Springer-Verlag, Wien, Austria, 1992, pp. 197-218.
- Lyu, M.R., and Nikora, A.P., "Using Software Reliability Models More Effectively," *IEEE Software*, July 1992, pp. 43-52.
- Lyu, M.R., and Nikora, A.P., "CASRE—A Computer-Aided Software Reliability Estimation Tool," *CASE 92 Proceedings*, Montreal, Canada, July 1992, pp.264-275.
- Lyu, M.R., and He, Y., "Improving the N-Version Programming Process Through the Evolution of Design Paradigm," *IEEE Transactions on Reliability*, vol. 42, no. 2, 1993 pp. 179-189.
- Lyu, M.R., Nikora, A.P, and Farr, W.H., "A Systematic and Comprehensive Tool for Software Reliability Modeling and Measurement," *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing (FTCS-23)*, Toulouse, France June 1993, pp. 648-653.
- Lyu, M.R., Chen, J., and Avizienis, A., "Experience in Metrics and Measurements for N-Version Programming," *International Journal of Reliability, Quality and Safety Engineering*, vol. 1, no. 1, 1994, pp. 41-62.
- Lyu, M.R., Horgan, J.R., and London, S., "A Coverage Analysis Tool for the Effectiveness of Software Testing," *IEEE Transactions on Reliability*, December 1994, pp. 527-535.
- Lyu, M.R. (ed.), *Software Fault Tolerance*, John Wiley & Sons, Chichester, U.K., January 1995.

- Lyu, M.R., Yu, J., Keramidas, E., and Dalal, S., "ARMOR: Analyzer for Reducing Module Operational Risk," *Proceedings of the 25th International Symposium on Fault Tolerant Computing (FTCS-25)*, Pasadena, Calif., June 1995, pp.137-142.
- Madden, W.A., and Rone, K.Y., "Design, Development, Integration: Space Shuttle Primary Flight Software System," *Communications of the ACM*, vol. 27, no. 8, 1984, pp 902-913.
- Malaiya, Y.K., Karunanithi, N., and Verma, P., "Predictability Measures for Software Reliability Models," *Proceedings of the 14th IEEE International Computer Software and Applications Conference (COMPSAC90)*, October 1990, pp.7-12.
- Malaiya, Y.K., and Srimani, P.K. (eds.), *Software Reliability Models: Theoretical Developments, Evaluation and Applications*, IEEE Computer Society Press, 1990.
- Malaiya, Y.K., Karunanithi, S. and Verma, P., "Predictability of Software-Reliability Models," *IEEE Transactions on Reliability*, vol. R-41, December 1992, pp.539-546.
- Martin, D.J., "Dissimilar Software in High Integrity Applications in Flight Controls," *Proceedings AGARD-CP 330*, September 1982, pp. 36.1-36.13.
- Martini, M.R., Kanoun, K., and de Souza, J.M., "Software Reliability Evaluation of the TROPICO-R Switching System," *IEEE Transactions on Reliability*, vol. 33, no. 3, 1990 pp.369-379.
- Martini, M.R., and de Souza, J.M., "Reliability Assessment of Computer Systems Design," *Microelectronics Reliability*, vol. 31, no. 2/3, 1991, pp. 237-244.
- Mathur, A.P., "On the Relative Strengths of Data Flow and Mutation Testing," *Proceedings of the Ninth Annual Pacific Northwest Software Quality Conference*, October 7-8, Portland, Oreg., 1991.
- Matsumoto, K., Inoue, K., Kikuno, T., and Torii, K., "Experimental Evaluation of Software Reliability Growth Models," *Proceedings of the 18th International Symposium on Fault-Tolerant Computing*, Tokyo, Japan, June 1988, pp.148-153.
- von Mayrhauser, A., and Keables, J., "A Data Collection Environment for Software Reliability Research," *Proceedings of the 2nd International Symposium on Software Reliability Engineering*, IEEE Computer Society Press, 1991.
- vonMayrhauser, A., and Keables, J., "A Simulation Environment for Early Life Cycle Software Reliability Research Prediction," *Proceedings of the International Tes. Conference*, IEEE Computer Society Press, 1992.
- vonMayrhauser, A., Malaiya, Y.K., Keables, J., and Srimani, P.K., "On the Need for Simulation for Better Characterization of Software Reliability," *Proceedings of the 4th International Symposium on Software Reliability Engineering*, Denver, Colo., 1993.
- Mays, R., Jones, C., Holloway, G., and Studinski, D., "Experiences with Defect Prevention," *IBM Systems Journal*, vol. 29, no. 1, 1990.
- Mazzuchi, T.A., and Soyer, T., "A Bayes Empirical-Bayes Model for Software Reliability," *IEEE Transactions on Reliability*, vol. R-37, pp. 248-254.
- McAllister, D.F., "Some Observations on Costs and Reliability in Software Fault-Tolerant Techniques," *Proceedings TIMS-ORSA Conference*, Boston, Mass., April 1985.



- McAllister, D.F., Sun, C.E., and Vouk, M.A., "Reliability of Voting in Fault-Tolerant Software Systems for Small Output Spaces," *IEEE Transactions on Reliability*, vol. 39, no. 5, 1990, pp. 524-534.
- McAllister, D.F., and Scott, R.K., "Cost Models for Fault-Tolerant Software," *Journal of Information and Software Technology*, vol. 33, no. 8, October 1991, pp. 594-603.
- McCabe, T.J., "A Complexity Metric" *IEEE Transactions on Software Engineering*, vol. 2, no. 4, December 1976, pp. 308-320.
- McConnell, S.R., Siewiorek, D.R., and Tsao, M.M., "The Measurement and Analysis of Transient Errors in Digital Computer Systems," *Proceedings of the 9th International Symposium on Fault-Tolerant Computing (FTCS-9)*, Madison, Wisconsin, June 1979, pp. 67-70.
- Mellor, P., "Failures, Faults, and Changes in Dependability Measurement," *Information and Software Technology*, vol. 34, no. 10, October, 1992, pp. 640-654.
- Mercer, "Product Advisor Survey," Trouble Reports, Mercer Analysis, 1994.
- Meyer, J.F., "On Evaluating the Performability of Degradable Computing Systems," *Proceedings of the 8th IEEE International Symposium on Fault-Tolerant Computing (FTCS-8)*, Toulouse, France, June 1978, pp. 44-49.
- Meyer, J.F., and Wei, L., "Analysis of Workload Influence on Dependability," *Proceedings of the 18th International Symposium on Fault-Tolerant Computing (FTCS-18)*, Tokyo, June 1988, pp. 84-89.
- Meyer, J.F., "Performability: A Retrospective and Some Pointers to the Future," *Performance Evaluation*, vol. 14, February 1992, pp. 139-156.
- Mili, A., *Introduction to Program Fault-Tolerance—A Structured Programming Approach*, Prentice-Hall International Ltd., U.K., 1990.
- Miller, D.R., "Exponential Order Statistic Models of Software Reliability Growth," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 1, January 1986, pp. 12-24.
- Miller, G., "The Magic Number Seven," *Psychological Review*, vol. 63, 1990.
- Miller, K.W., Morell, L.J., Noonan, R.E., Park, S.K., Nicol, D.M., Murrill, B.W., and Voas, J.M., "Estimating the Probability of Failure When Testing Reveals No Failures," *IEEE Transactions on Software Engineering*, vol. 18, no. 1, 1992, pp. 33-43.
- Mood, A., Graybill, F., and Boes, D., *Introduction to the Theory of Statistics*, 3d ed., McGraw-Hill, Inc., New York, 1974.
- Moranda, P.L., and Jelinski, Z., *Final Report on Software Reliability Study*, McDonnell Douglas Astronautics Company, MADC Report Number 63921, 1972.
- Moranda, P.B., "Software Reliability Predictions," *Proceedings of the Sixth Triennial World Congress of the International Federation of Automatic Control*, 1975, pp. 34.2-34.7.
- Moranda, P.B., "Predictions of Software Reliability During Debugging," *Proceedings of the Annual Reliability and Maintainability Symposium*, Washington, D.C., 1975, pp. 327-332.
- Moranda, P.B., "Event-Altered Rate Models for General Reliability Analysis," *IEEE Transactions on Reliability*, vol. R-28, no. 5, 1979, pp. 376-381.

- [Morg89] Morgan, N., and Bourlard, H., "Generalization and Parameter Estimation in Feedforward Nets: Some Experiments," *Advances in Neural Information Processing Systems 2*, D. Touretzky (ed.), Morgan Kaufman, San Francisco, 1989, pp. 630-637.
- [Mosl94] Mosleh, A., *Reliability Data Collection and Analysis System (REDCAS)*, Center for Reliability Engineering, University of Maryland, College Park, Md., 1994.
- [Mcur87] Mourad, S., and Andrews, D., "On the Reliability of the IBM MVS/XA Operating System," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 10, October 1987, pp. 1135-1139.
- [Muns89] Munson, J.C., and Khoshgoftaar, T.M., "The Dimensionality of Program Complexity," *Proceedings of the 11th Annual International Conference on Software Engineering*, Pittsburgh, May 1989, pp. 245-253.
- [Muns90a] Munson, J.C., and Khoshgoftaar, T.M., "Regression Modeling of Software Quality Empirical Investigation," *Journal of Information and Software Technology*, vol. 32, no. 2, March 1990, pp. 105-114.
- [Muns90b] Munson, J.C., and Khoshgoftaar, T.M., "Applications of a Relative Complexity Metric for Software Project Management," *Journal of Systems and Software*, vol. 12, no. 3, 1990, pp. 283-291.
- [Muns90c] Munson, J.C., and Khoshgoftaar, T.M., "The Relative Software Complexity Metric: A Validation Study," *Proceedings of the Software Engineering 1990 Conference*, London, 1990, pp. 89-102.
- [Muns91] Munson, J.C., and Khoshgoftaar, T.M., "The Use of Software Complexity Metrics in Software Reliability Modeling," *Proceedings of the International Symposium on Software Reliability Engineering*, Austin, Texas, May 1991, pp. 2-11.
- [Muns92] Munson, J.C., and Khoshgoftaar, T.M., "The Detection of Fault-Prone Programs," *IEEE Transactions on Software Engineering*, vol. 18, no. 5, 1992, pp. 423-433.
- [Muns93] Munson, J.C., and Khoshgoftaar, T.M., "Measurement of Data Structure Complexity," *Journal of Systems and Software*, vol. 12, no. 3, March 1993, pp. 217-225.
- [Musa75] Musa, J.D., "A Theory of Software Reliability and Its Applications," *IEEE Transactions on Software Engineering*, vol. SE-1, no. 3, 1975, pp. 312-327.
- [Musa77a] Musa, J.D., "Program for Software Reliability and System Test Schedule Estimation—User's Guide," Bell Telephone Laboratories, Inc., 1977.
- [Musa77b] Musa, J.D., and Hamilton, PA., "Program for Software Reliability and System Test Schedule Estimation—Program Documentation," Bell Telephone Laboratories, Inc., 1977.
- [Musa78] Musa, J.D., "Progress in Software Reliability Measurement," *Second Software Life Cycle Management Workshop*, published by the IEEE, New York, 1978, pp. 153-155.
- [Musa79] Musa, J.D., "Software Reliability Data," Technical Report, Data and Analysis Center for Software, Rome Air Development Center, Griffins AFB, New York, 1979.
- [Musa79a] Musa, J.D., "Validity of Execution—Time Theory of Software Reliability," *IEEE Transactions on Reliability*, vol. R-28, no. 3, 1979, pp. 181-191.
- [Musa79b] Musa, J.D., "Software Reliability Measures Applied to System Engineering," *Proceedings of the National Computer Conference*, 1979, pp. 941-946.

- Musa, J.D., "The Measurement and Management of Software Reliability," *Proceedings of the IEEE*, vol. 68, no. 9, 1980, pp. 1131-1143.
- Musa, John D., and Okumoto, K., "Software Reliability Models: Concepts, Classification, Comparisons, and Practice," *Electronic Systems Effectiveness and Life Cycle Costing*, J.K. Skwirzynski (ed.), NATO ASI Series, F3, Springer-Verlag, Heidelberg, pp. 395-424.
- Musa, J.D., and Okumoto, K. "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement," *Proceedings Seventh International Conference on Software Engineering*, Orlando, Florida, pp. 230-238.
- Musa, J.D., "Software Reliability," *Handbook of Software Engineering*, C.R. Vick and C.V. Ramamoorthy (eds.), Van Nostrand Reinhold, New York, 1984, pp. 392-412.
- Musa, J.D., Iannino, A., and Okumoto, K., *Software Reliability—Measurement, Prediction, Application*, McGraw-Hill, New York, 1987.
- Musa, J.D., and Everett, W.W., "Software-Reliability Engineering: Technology for the 1990s," *IEEE Software*, vol. 7, November 1990, pp. 36-43.
- Musa, J.D., "Reduced Operation Software," *ACM Software Engineering Notes*, vol. 16, no. 3, July 1991, p. 78.
- Musa, J.D., "Rationale for Fault Exposure Ratio K," *ACM Software Engineering Note*, vol. 16, no. 3, July 1991, p. 99.
- Musa, J.D., "Operational Profiles in Software Reliability Engineering," *IEEE Software*, vol. 10, no. 2, March 1993, pp. 14-32.
- Musa, J.D., "Sensitivity of Field Failure Intensity to Operational Profile Errors," *Proceedings of the 5th International Symposium on Software Reliability Engineering Monterey*, Calif, November 6-9, 1994, pp. 334-337.
- Myers, R.H., *Classical and Modern Regression with Applications*, Duxbury Press, Boston, Mass., 1990.
- Nagel, P.M., and Skrivan, J.A., "Software Reliability: Repetitive Run Experimentation and Modeling," Report NASA CR-165836, February 1982.
- Nakamura, K. (ed.), *Ayumi: Continuous Improvement of Software Quality Is Key at Fujitsu in Japan*, JUSE Press, 1992.
- Navlakha, J.K., "Measuring the Effect of External and Internal Interface on Software Development," *Proceedings of the 20th Annual Hawaii International Conference on System Sciences*, 1987, pp. 127-136.
- Nelson, V.P., and Carroll, B.D. (eds.), *Tutorial: Fault-Tolerant Computing*, IEEE Computer Society Press, 1987.
- Neufelder, A.M., *Ensuring Software Reliability*, Marcel Dekker, Inc., New York, 1993
- Nicola, V.F., and Goyal, A., "Modeling of Correlated Failures and Community Error Recovery in Multi-Version Software," *IEEE Transactions on Software Engineering* vol. 16, no. 3, 1990.
- Nikora, A.P., Lyu, M.R., and Antczak, T.M., "A Linear Combination Software Reliability Modeling Tool with a Graphically Oriented User Interface," *Proceedings of Symposium on Assessment of Quality Software Development Tools*, New Orleans Louisiana, May 1992, pp. 21-31.
- Nikora, A.P., "CASRE User's Guide," Jet Propulsion Laboratories, August 1993.

- [Nils90] Nilsson, J.N., *The Mathematical Foundations of Learning Machines*, Morgan Kaufmann San Francisco, 1990, chapters 2 and 3.
- [Ohba84] Ohba, M., "Software Reliability Analysis Models," *IBM Journal of Research and Development*, vol. 21, no. 4, July 1984, pp. 428-443.
- [Ohte90] Ohtera, H., and Yamada, S., "Optimum Software-Release Time Considering an Error Detection Phenomenon During Operation," *IEEE Transactions on Reliability*, vol. R-39, 1990, pp. 596-599.
- [OkumSO] Okumoto, K., and Goel, A., "Optimum Release Time for Software Systems Based on Reliability and Other Performance Measures," *The Journal of Systems and Software*, vol. 1, no. 4, 1980, pp. 315-318.
- [Onom93] Onoma, A.K., and Yamaura, T., "Practical Quantitative Methods of Improving Software Quality: Based on Decades of Experience," *NRC Workshop on Statistics in Software Engineering*, NRC, Washington, D.C., October 1993.
- [Onom95] Onoma, A.K., and Yamaura, T., "Practical Steps Toward Quality Development," *IEEE Software*, vol. 12, no. 5, September 1995, pp. 68-77.
- [PageSO] Pages, A., and Gondran, M., *System Reliability*, Eyrolles, Paris, 1980; in French.
- [Pant91] Pant, H., "Tracking Quality from Verification to Customer," *Software Quality from the Customer Viewpoint*, Globcom, Phoenix, Ariz., December 2—5, 1991.
- [Panz81] Panzl, D.J., "A Method for Evaluating Software Development Techniques," *The Journal of Systems Software*, vol. 2, 1981, pp. 133-137.
- [Papo65] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill Book Company, New York, N.Y., 1965, pp. 534-551.
- [Park82] Parker, D., "Learning Logic," *Invention Record S81-64*, File 1, Stanford University, Stanford, Calif., 1982.
- [Parn74] Parnas, D.L., "On a 'Buzzword': Hierarchical Structure," in *Proceedings of the 1974 IFIP Congress*, pp. 336-339.
- [Paul93] Paulk, M.C., Curtis, B., Chrissis, M.B., and Weber, C.V., "Capability Maturity Model, Version 1.1," *IEEE Software*, July 1993, pp. 18-27.
- [Pign88] Pignal, P.I., "An Analysis of Hardware and Software Availability Exemplified on the IBM 3725 Communication Controller," *IBM Journal of Research and Development*, vol. 32, no. 2, March 1988, pp. 268-278.
- [Prue95] Pruet, W.R., *Personal Communication and Data Clarification Report*, NASA, January 26, 1995.
- [RADC87] RADC, *Methodology for Software Reliability Prediction and Assessment*, RADC Technical Report TR-87-171, Rome Air Development Center, 1987. Revised in [RL92].
- [Rama82] Ramamoorthy, C.V., and Bastani, F.B.: "Software Reliability—Status and Perspectives," *IEEE Transactions on Software Engineering*, vol. SE-8, no. 4, July 1982, pp. 354-371.
- [Rand75] Randell, B., "System Structure for Software Fault Tolerance," *IEEE Transactions on Software Engineering*, vol. SE-1, no. 2, June 1975, pp. 220-232.
- [Rapp90] Rapp, B., "Application of Software Reliability Models in Medical Imaging Systems," *Proceedings of the 1990 International Symposium on Software Reliability Engineering*, Washington, D.C., April 1990.

- Rauzy, A., "New Algorithms for Fault Tree Analysis," *Reliability Engineering and System Safety*, vol. 40, 1993, pp. 203-211.
- Roberts, N., et al., *Introduction to Computer Simulation*, Addison-Wesley Book Company, Reading, Mass., 1983.
- Rodriguez, V., and Tsai, W., "A Tool for Discriminant Analysis and Classification of Software Metrics," *Information and Software Technology*, vol. 29, no. 5, April 1987, pp. 137-149.
- Rohn, W.B., and Arnold, T.F., "Design for Low Expected Downtime Control Systems," *Proceedings of the 4th International Conference on Computer Communications*, Philadelphia, Pennsylvania, June 1972, pp. 16-25.
- Rome Laboratory (RL), *Methodology for Software Reliability Prediction and Assessment*, Technical Report RL-TR-92-52, volumes 1 and 2, 1992.
- Ross, N., "The Collection and Use of Data for Monitoring Software Projects," *Measurement for Software Control and Assurance* B.A. Kitchenham and B. Littlewood (eds.), Elsevier Applied Science, London and New York, 1989, pp. 125-154.
- Roy, B., *Modern Algebra and Graph Theory*, Dunod, Paris, 1969; in French.
- Rudin, H., "An Informal Overview of Formal Protocol Specification," *IEEE Communications Magazine*, vol. 23, no. 3, March 1985, pp. 46-52.
- Rumelhart, D., Hinton, G., and Williams, R., "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, vol. I, MIT Press, 1986, pp. 318-362.
- SAE RMS Guidebook, *Reliability, Maintainability, Supportability Guidebook*, Society of Automotive Engineers (SAE) G-12 RMS Committee. To order, call (412) 776-4970 using #M-102.
- Saglietti, F., and Ehrenberger, W., "Software Diversity—Some Considerations about Benefits and Its Limitations," *Proceedings IFAC SAFE-COMP'86*, 1986, pp. 27-34.
- Sahinoglu, M., "Compound-Poisson Software Reliability Model," *IEEE Transactions on Software Engineering*, SE-18, no. 7, 1992, pp. 624-630.
- Sahner, R.A., and Trivedi, K.S., "Reliability Modeling Using SHARPE," *IEEE Transactions on Reliability*, vol. R-36, no. 2, June 1987, pp. 186-193.
- Sahner, R.A., Trivedi, K.S., and Puliafito, A., *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package*, Kluwer Academic Publishers, Boston, Mass., 1995.
- Sandier, G.H., *Systems Reliability Engineering*, Prentice-Hall, Englewood Cliffs, New Jersey, 1963.
- Sargent, R.G., "An Introduction to the Selection and Use of Simulation Languages," Syracuse University, TR-79-1, Department of Industrial Engineering.
- SAS User's Guide: Basics, SAS Institute, 1985.
- Sayet, C., and Pilaud, E., "An Experience of a Critical Software Development," *Proceedings of the 20th IEEE International Symposium on Fault-Tolerant Computing (FTCS-20)*, Newcastle, U.K., July 1990, pp. 36-45.

- Schick, G.J., and Wolverton, R.W., "Assessment of Software Reliability," *Proceedings of the Operations Research*, Physica-Verlag, Wurzburg-Wien, 1973, pp.395-122.
- Schneidewind, N.F., "Analysis of Error Processes in Computer Software," *Sigplan Note*, vol. 10, no. 6, 1975, pp. 337-346.
- Schneidewind, N.F., "Methodology for Validating Software Metrics," *IEEE Transactions on Software Engineering*, vol. 18, no. 5, May 1992, pp. 410-421.
- Schneidewind, N.F., and Keller, T.W., "Application of Reliability Models to the Space Shuttle," *IEEE Software*, July 1992, pp. 28-33.
- Schneidewind, N.F., "Minimizing Risk in Applying Metrics on Multiple Projects," *Proceedings of the Third International Software Reliability Engineering*, Research Triangle Park, North Carolina, October 1992, pp. 173-182.
- Schneidewind, N.F., "Software Reliability Model with Optimal Selection of Failure Data," *Proceedings of the 1993 Annual Oregon Workshop on Software Metrics*, March 23, 1993.
- Schneidewind, N.F., "Optimal Selection of Failure Data for Predicting Failure Counts," *Proceedings of the 1993 Complex Systems Engineering Synthesis and Assessment Technology Workshop (CSESAW '93)*, Naval Surface Warfare Center Dahlgren Division July 20-22, 1993, pp. 141-157.
- Schneidewind, N.F., "Software Reliability Model with Optimal Selection of Failure Data," *IEEE Transactions on Software Engineering*, vol. 19, no. 11, November, 1993 pp. 1095-1104.
- Schneidewind, N.F., "Optimal Selection of Failure Data for Predicting Failure Counts," *Proceedings of the Fourth International Symposium on Software Reliability Engineering*, November, 1993, pp. 142-149.
- Scott, R.K., Gault, J.W., and McAllister, D.F., "The Consensus Recovery Block," *Proceedings of the Total Systems Reliability Symposium*, 1983, pp.3-9.
- Scott, R.K., Gault, J.W., McAllister, D.F., and Wiggs, J., "Investigating Versior Dependence in Fault-Tolerant Software," *AGARD 361*, 1984, pp. 21.1-21.10.
- Scott, R.K., Gault, J.W., McAllister, D.F., and Wiggs, J., "Experimental Validation of Six Fault-Tolerant Software Reliability Models," *Proceedings of the IEEE 14th Fault-Tolerant Computing Symposium*, 1984, pp. 102-107.
- Scott, R.K., Gault, J.W., and McAllister, D.F., "Fault-Tolerant Reliability Modeling," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 5, 1987, pp.582-592.
- Shen, V, Yu, T, Thebaut, S., and Paulsen, L., "Identifying Error-Prone Software: An Empirical Study," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 4, April 1985, pp. 317-323.
- Shimeall, T.J., and Leveson, N.G., "An Empirical Comparison of Software Fault-Tolerance and Fault Elimination," *Proceedings of the 2nd Workshop on Software Testing, Verification and Analysis*, Banff, IEEE Computer Society, July 1988, pp.180-187.



- [Shin84] Shin, K.G., and Lee, Y.-H., "Evaluation of Error Recovery Blocks Used for Cooperating Processes," *IEEE Transactions on Software Engineering*, vol. SE-10, no. 6, November 1984, pp. 692-700.
- [Shoo72] Shooman, M.L., "Probabilistic Models for Software Reliability Prediction," *Statistical Computer Performance Evaluation*, Academic Press, New York, June 1972, pp. 485-502.
- [Shoo73] Shooman, M.L., "Operational Testing and Software Reliability Estimation During Program Developments," *Record of 1973 IEEE Symposium on Computer Software Reliability*, IEEE Computer Society, New York, 1973, pp. 51-57.
- [Shoo76] Shooman, M.L., "Structural Models for Software Reliability Prediction," *Proceedings of the 2nd International Conference on Software Engineering*, IEEE Computer Society, New York, October 1976.
- [Shoo77a] Shooman, M.L., "Spectre of Software Reliability and Its Exorcism," *Proceedings of the 1977 Joint Automatic Control Conference*, IEEE, New York, 1977, pp.225-231.
- [Shoo77b] Shooman, M.L., and Naturaja, S., *Effect of Manpower Deployment and Bug Generation on Software Error Models*, Rome Air Development Center Technical Report, RADC-TR-76-400, 1977.
- [Shoo83] Shooman, M.L., *Software Engineering*, McGraw-Hill, New York, 1983.
- [Shoo90] Shooman, M.L., *Probabilistic Reliability: An Engineering Approach*, McGraw-Hill, New York, 2d ed., 1990.
- [Siew82] Siewiorek, D.P., and Swarz, R.S., *The Theory and Practice of Reliable System Design* Digital Press, Bedford, Mass., 1982.
- [Siew92] Siewiorek, D.P., and Swarz, R.S., *Reliable Computer Systems—Design and Evaluation*, 2d ed, Digital Press, Bedford, Mass., 1992.
- [Siew93] Siewiorek, D., Hudak, J.J., Suh, B., and Segall, Z., "Development of a Benchmark to Measure System Robustness," *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing*, 1993, pp. 88-97.
- [Sing85] Singpurwalla, N.D., and Soyer, R., "Assessing (Software) Reliability Growth Using a Random Coefficient Autoregressive Process and Its Ramifications," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 12, December 1985, pp.1456-1464.
- [Sing91] Singpurwalla, N.D., "Determining an Optimal Time Interval for Testing and Debugging Software," *IEEE Transactions on Software Engineering*, vol. 17, no.4,1991, pp. 313-319.
- [Sing92] Singpurwalla, N.D., and Soyer, R., "Nonhomogenous Auto-Regressive Process for Tracking (Software) Reliability Growth, and Their Bayesian Analysis," *Journal of the Royal Statistical Society*, series B, vol. 54, 1992, pp. 145-156.
- [Smit88] Smith, P.M., Trivedi, K.S., and Ramesh, A.V., "Performability Analysis: Measures, an Algorithm, and a Case Study," *IEEE Transactions on Computers*, vol. 37, no. 4, April 1988, pp. 406-417.
- [Smit93] Smith, W., "Software and Switching System Reliability: Cause, Cure, or Both?" *Proceedings of Network Reliability: A Report to the Nation*, Symposium of the National Engineering Consortium, Washington, D.C., June 10-11, 1993.
- [Sof93] Software Edge, *Defect Control System for Windows*, The Software Edge, Inc., Colorado Springs, Colo., 1993.

- [Spat80] Spath, H., *Cluster Analysis Algorithms*, Ellis Horwood, West Sussex, U.K., 1990.
- [Spec84] Spector, A., and Gifford, D., "The Space Shuttle Primary Computer System," *Communications of the ACM*, vol. 27, no. 8, 1984, pp. 874-900.
- [SRMP88] *Software Reliability Modeling Programs*, Reliability and Statistical Consultants Ltd., Version 1.0, May 1988.
- [Stam93] Stamen, J.P., "Structuring Data for Analysis," *IEEE Spectrum*, October 1993, pp. 55-58.
- [Star87] Stark, G.E., "Dependability Evaluation of Integrated Hardware/Software Systems," *IEEE Transactions on Reliability*, vol. R-36, no. 4, 1987, pp. 440-444.
- [Star91] Stark, G.E., "A Survey of Software Reliability Measurement Tools," *Proceedings of the International Symposium on Software Reliability Engineering*, Austin, Texas, May 1991 pp. 90-97.
- [Ster78] Sterner, B.J., "Computerized Interlocking System—A Multidimensional Structure in the Pursuit of Safety," *IMechE Railway Engineer International*, November/December 1978, pp. 29-30.
- [Ston74] Stone, M., "Cross-Validatory Choice and Assessment of Statistical Predictions," *Journal of the Royal Statistical Society, series B*, vol. 36, no. 2, 1974, pp. 111-147.
- [Stri85] Strigini, L., and Avizienis, A., "Software Fault-Tolerance and Design Diversity: Past Experience and Future Evolution," *Proceedings IFAC SAFE-COMP'85*, 1985, pp. 167-172.
- [Suke76] Sukert, A., *A Software Reliability Modeling Study*, Rome Air Development Center Technical Report, RADC-TR-76-247, 1976.
- [Suke80] Sukert, A., and Goel, A., "A Guidebook for Software Reliability Assessment," *1980 Proceedings of the Annual Reliability and Maintainability Symposium*, 1980, pp. 186-190.
- [Sull91] Sullivan, M.S., and Chillarege, R., "Software Defects and Their Impact on System Availability—A Study of Field Failures in Operating Systems," *Proceedings of the 21st International Symposium Fault-Tolerant Computing*, June 1991, pp. 2-9.
- [Swee95] Sweet, W., "The Glass Cockpit," *IEEE Spectrum*, September 1995, pp. 30-38.
- [SWTR93] SWTRHA, *Report of the Inquiry into the London Ambulance Service*, South West Thames Regional Health Authority, February 1993.
- [Tai93] Tai, A.T., Meyer, J.F., and Avizienis, A., "Performability Enhancement of Fault-Tolerant Software," *IEEE Transactions on Reliability*, vol. 42, no. 2, June 1993, pp. 227-237.
- [Take86] Takeda, M., and Goodman, J., "Neural Networks for Computation: Number Representations and Programming Complexity," *Applied Optics*, vol. 25, no. 18, September 1986, pp. 3033-3045.
- [TA ND85] Tandem Computers Inc., *Product Reporting System User's Guide*, June 1985.
- [TAND89] Tandem Computers Inc., *Tandem Maintenance and Diagnostic System Reference Manual* March 1989.

- Tang, D., Iyer, R.K., and Subramani, S., "Failure Analysis and Modeling of a VAXcluster System," *Proceedings of the 20th International Symposium on Fault-Tolerant Computing*, June 1990, pp. 244-251.
- Tang, D., and Iyer, R.K., "Impact of Correlated Failures on Dependability in a VAXcluster System," *Proceedings of the 2nd IFIP Working Conference on Dependable Computing for Critical Applications*, Tucson, Arizona, February 1991.
- Tang, D., and Iyer, R.K., "Analysis and Modeling of Correlated Failures in Multicomputer Systems," *IEEE Transactions on Computers*, vol. 41, no. 5, May 1992, pp. 567-577.
- Tang, D., and Iyer, R.K., "Analysis of the VAX/VMS Error Logs in Multicomputer Environments—A Case Study of Software Dependability," *Proceedings of the Third International Symposium Software Reliability Engineering*, Research Triangle Park, North Carolina, October 1992, pp. 216-226.
- Tang, D., "Measurement-Based Dependability Analysis and Modeling for Multicomputer Systems," Ph.D. thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1992.
- Tang, D., and Iyer, R.K., "Dependability Measurement and Modeling of a Multicomputer Systems," *IEEE Transactions on Computers*, vol. 42, no. 1, January 1993, pp. 62-75.
- Tang, D., and Iyer, R.K., "MEASURE-1—A Measurement-Based Dependability Analysis Package," *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, Santa Clara, Calif., May 1993, pp. 110-121.
- Tausworthe, R.C., "A General Software Reliability Process Simulation Technique," Technical Report 91-7, Jet Propulsion Laboratory, Pasadena, Calif., March 1991.
- Tausworthe, R.C., and Lyu, M.R., "A Generalized Software Reliability Process Simulation Technique and Tool," *Proceedings of the 5th International Symposium on Software Reliability Engineering*, Monterey, Calif., November 1994.
- Tausworthe, R.C., and Lyu, M.R., "A Generalized Technique for Simulating Software Reliability," *IEEE Software*, March 1996.
- Taylor, D.J., Morgan, D.E., and Black, J.P., "Redundancy in Data Structures: Improving Software Fault-Tolerance," *IEEE Transactions on Software Engineering*, vol. SE-6, 1980, pp. 585-594.
- Thayer, T.A., Lipow, M., and Nelson, E.C., *Software Reliability*, North-Holland Publishing Company, New York, New York, 1978.
- Thevenod-Fosse, P., "Software Validation by Means of Statistical Testing: Retrospect and Future Direction," *Dependable Computing and Fault-Tolerant Systems, Vol. 4*, A. Avizienis and J.-C. Laprie (eds.), Springer-Verlag, Wien, Austria and New York, 1991, pp. 25-48.
- Thompson, W.E., and Chelson, P.O., "On the Specification and Testing of Software Reliability," *Proceedings of the 1980 Annual Reliability and Maintainability Symposium*, IEEE, New York, 1980, pp. 379-383.
- Tian, J., and Lu, P., "Measuring and Modeling Software Reliability: Data, Models Tools, and a Support Environment," Technical Report, TR-74.117, IBM Canada Laboratory, March 1993.

- [Tian93b] Tian, J., and Lu, P., "Software Reliability Measurement and Modeling for Multiple Releases of Commercial Software," *Proceedings of the Fourth International Symposium on Software Reliability Engineering*, Denver, Colo., November 3-6, 1993, pp. 253-260.
- [Tohm89] Tohma, Y, Tokunaga, K., Nagase, S., and Murata, Y., "Structural Approach to the Estimation of the Number of Residual Faults Based on the Hyper-Geometric Distribution," *IEEE Transactions on Software Engineering*, vol. SE-15, no. 3, March 1989, pp. 345-355.
- [Tohm89a] Tohma, Y., Jacoby, R., Murata, Y., and Yamamoto, M., "Hyper-Geometric Distribution Model to Estimate the Number of Residual Software Faults," *Proceedings COMPSAC 89*, IEEE CS Press, 1989, pp. 610-617.
- [Tohm91] Tohma, Y, Yamano, H., Ohba, M., and Jacoby, R., "Parameter Estimation of the Hyper-Geometric Distribution Model for Real Test/Debug Data," *Proceedings of the 1991 IEEE International Symposium on Software Reliability Engineering*, May 1991, pp. 28-34.
- [Tome93] Tomek, L.A., Muppala, J.K., and Trivedi, K.S., "Modeling Correlation in Software Recovery Blocks," *IEEE Transactions on Software Engineering*, vol. 19, no. 11, 1993, pp. 1071-1086.
- [Toy85] Toy, W.N., "Modular Redundancy Concept, Problems and Solutions," *EPRI Seminar: Digital Control and Fault-Tolerant Computer Technology*, Scottsdale, Arizona, April 1985.
- [Trac90] Trachtenberg, M., "A General Theory of Software Reliability Modeling," *IEEE Transactions on Reliability*, R-39, no. 1, 1990, pp. 92-96.
- [Trav87] Traverse, P., "AIRBUS and ATR System Architecture and Specification," in [Voge87a], 1987, pp. 95-104.
- [Triv75] Trivedi, A.K., "Computer Software Reliability: Many-State Markov Modeling Techniques," Ph.D. dissertation, Polytechnic Institute of Brooklyn, June, 1975.
- [Triv82] Trivedi, K.S., *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [Triv92] Trivedi, K.S., Muppala, J.K., Woollet, S.P., and Haverkort, B.R., "Composite Performance and Dependability Analysis," *Performance Evaluation*, vol. 14, February 1992, pp. 197-215.
- [Troy85] Troy, R, and Baluteau, C., "Assessment of Software Quality for the Airbus A310 Automatic Pilot," *Proceedings of the 15th International Symposium on Fault Tolerant Computing*, Ann Arbor, Mich., IEEE CS Press, June 1985, 438-443.
- [Tsao83] Tsao, M.M., and Siewiorek, D.P., "Trend Analysis on System Error Files," *Proceedings of the 13th International Symposium on Fault-Tolerant Computing*, June 1983, pp. 116-119.
- [Tso86] Tso, K.S., Avizienis, A., and Kelly, J.P.J., "Error Recovery in Multi-Version Software," *Proceedings IFAC SAFECOMP'86*, Sarlat, France, 1986, pp. 35-41.
- [Tso87] Tso, K.S., and Avizienis, A., "Community Error Recovery in N-Version Software: A Design Study with Experimentation," *Proceedings of the 17th Fault-Tolerant Computing Symposium*, 1987, pp. 127-133.
- [Tuke77] Tukey, J.W., *Exploratory Data Analysis*, Addison-Wesley, Reading, Mass., 1977.

- [Turn87] Turner, D.B., Burns, R.D., and Hecht, H., "Designing Micro-Based Systems for Fail-Safe Travel," *IEEE Spectrum*, vol. 24, no. 2, February 1987, pp. 58-63.
- [Uemu90] Uemura, M., Yamada, S., and Fujino, K., "Software Reliability Method: Application of Delayed S-Shaped NHPP Model and Other Related Models," *International Symposium on Reliability and Maintainability*, Tokyo, 1990, pp.467-472.
- [Upad86] Upadhyaya, S., and Saluja, K., "A Watchdog Processor Based General Rollback Technique with Multiple Retries," *IEEE Transactions on Software Engineering*, vol. SE-12, 1986, pp. 87-95.
- [USAT93] *USA Today*, April 15, 1993, pp. 6B-7B.
- [Vale88] Valette, V., "An Environment for Software Reliability Evaluation," *Proceedings of Software Engineering & Its Applications*, Toulouse, France, December 1988, pp.879-897.
- [Vale92] Valette, V., and Vallee, F., "Software Quality Metrics in Space Systems," *Proceedings of the Third International Symposium on Software Reliability Engineering*, Research Triangle Park, North Carolina, October 1992, pp. 296-302.
- [Vela84] Velardi, P., and Iyer, R.K., "A Study of Software Failures and Recovery in the MV5 Operating System," *IEEE Transactions on Computers*, vol. C-33, no. 6, June 1984, pp.564-568.
- [Vinn87] Vinneau, R., *User's Guide to a Computerized Implementation of the Goel-Okumoto Non-Homogeneous Poisson Process Software Reliability Model*, IIT Research Institute. Data Analysis Center for Software (DACS) report 90-0027, November 1987.
- [Voge87a] Voges, U. (ed.). *Software Diversity in Computerized Control Systems*, vol. 2 of *Dependable Computing and Fault-Tolerant Systems*, A. Avizienis, H. Kopetz, and J.-C. Laprie (eds.), Springer-Verlag, Wien and New York, 1987.
- [Voge87b] Voges, U., "Flight Applications," in [Voge87a], 1987, pp. 87-93.
- [Voge87c] Voges, U., "Use of Diversity in Experimental Reactor Safety Systems," in [Voge87a] 1987, pp. 29-49.
- [Vouk85] Vouk, M.A., McAllister, D.F., and Tai, K.C., "Identification of Correlated Failures of Fault-Tolerant Software Systems," in *Proceedings COMPSAC 85*, 1985, pp. 437-444.
- [Vouk86a] Vouk, M.A., McAllister, D.F., and Tai, K.C., "An Experimental Evaluation of the Effectiveness of Random Testing of Fault-Tolerant Software," *Proceedings of the Workshop on Software Testing*, Banff, Canada, IEEE CS Press, July 1986, pp.74-81.
- [Vouk86b] Vouk, M.A., Helsabeck, M.L., Tai, K.C., and McAllister, D.F., "On Testing of Functionally Equivalent Components of Fault-Tolerant Software," *Proceedings COMPSAC 86*, 1986, pp. 414-419.
- [Vouk88] Vouk, M.A., "On Engineering of Fault-Tolerant Software," *10th International Symposium*, "Computer at the University," Cavtat 88, Cavtat, Croatia, 1988.
- [Vouk90a] Vouk, M.A., Caglayan, A., Eckhardt, D.E., Kelly, J., Knight, J., McAllister, D., and Walker, L., "Analysis of Faults Detected in a Large-Scale Multiversion Software Development Experiment," *Proceedings DASC'90*, 1990, pp. 378-385.

- [Vouk90b] Vouk, M.A., Paradkar, A., and McAllister, D., "Modeling Execution Time of Multistage N-Version Fault-Tolerant Software," *Proceedings COMPSAC 90*, October 1990, pp. 505-511. Reprinted in *Fault-Tolerant Software Systems: Techniques and Applications*, HoangPham (ed.), IEEE Computer Society Press, 1992, pp. 55-61.
- [Vouk90c] Vouk, M.A., "Back-to-Back Testing," *Information and Software Technology*, vol. 32, no. 1, 1990, pp. 34-15.
- [Vouk92] Vouk, M.A., "Using Reliability Models During Testing with Non-Operational Profiles," *Proceedings Second Workshop on Issues in Software Reliability Estimation*, October 12-13, 1992, Bellcore, Livingston, N.J.
- [Vouk93a] Vouk, M.A., McAllister, D.F., Eckhardt, D.E., and Kim, K., "An Empirical Evaluation of Consensus Voting and Consensus Recovery Block Reliability in the Presence of Failure Correlation," *Journal of Computer and Software Engineering*, vol. 1, no. 4, 1993, pp. 367-388.
- [Vouk93b] Vouk, M.A., and Tai, K.C., "Some Issues in Multi-Phase Software Reliability Modeling," *Proceedings CASCON'93*, October 1993.
- [Wago73] Wagoner, W., *The Final Report on a Software Reliability Measurement Study*, The Aerospace Corp., El Segundo, Calif, Report No. TOR-0074 (4112)-1, August 15, 1973.
- [Wall84] Wallace, J.J., and Barnes, W.W., "Designing for Ultrahigh Availability: The Unix RTR Operating System," *IEEE Computer*, August 1984, pp. 31-39.
- [Wals85] Walsh, P.J., "A Measure of Test Case Completeness," Ph.D. dissertation, State University of New York, Binghamton, N.Y., 1985.
- [Wang93] Wang, Y.M., Huang, Y., and Fuchs, W.K., "Progressive Retry for Software Error Recovery in Distributed Systems," *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing*, June 1993, pp. 138-144.
- [Wats91] Watson, G.F., "Service Disrupted in Cities: Bell Companies, Manufacturers Join to Cure Phone Outages," *The Institute*, IEEE Press, September 1991, pp. 1 and 7.
- [Weer94] Weerahandi, S., and Hausman, R.E., "Software Quality Measurement Based on Fault-Detection Data," *IEEE Transactions Software Engineering*, September 1994, pp. 665-676.
- [Wei91] Wei, L., "A Model Based Study of Workload Influence on Computing System Dependability," Ph.D. thesis, University of Michigan, 1991.
- [Weib51] Weibull, W., "A Statistical Distribution Function of Wide Application," *J. Appl. Mech.*, vol. 18, 1951, pp. 293-297.
- [Weig90] Weigend, A., Huberman, B., and Rumelhart, D., "Predicting the Future: A Connectionist Approach," Stanford University, TR Stanford-PDP-90-01, April 1990.
- [Wein90] Wein, A.S., and Sathaye, A., "Validating Complex Computer System Availability Models," *IEEE Transactions on Reliability*, vol. 39, no. 4, October 1990, pp. 468-479.
- [Weis88] Weiss, S.N., and Weyuker, E. J., "An Extended Domain-Based Model of Software Reliability," *IEEE Transactions on Software Engineering*, SE-14, no. 12, 1988, pp. 1512-1524.
- [Werb74] Werbos, P., "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Ph.D. thesis, Harvard University, Cambridge, Mass., 1974.



- [WH183] Williams, J.F., Yount, L.J., and Flanningan, J.B., "Advanced Autopilot Flight Director System Computer Architecture for Boeing 737-300 Aircraft," *Proceedings of the 5th Digital Avionics Systems Conference*, Seattle, Wash., 1983.
- [Will89] Williams, R., and Zipser, D., "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, vol. 1, no. 2, 1989, pp.270-280.
- [Wong93] Wong, WE., "On Mutation and Data Flow," dissertation, available as Technical Report SERC-TR-149-P, Software Engineering Research Center, Pur-due University, W. Lafayette, Ind., 1993.
- [Wrig86] Wright, N.C.J., "Dissimilar Software," *Workshop on Design Diversity in Action*, Baden, Austria, June 1986, see summary in [Voge87b].
- [Wrig88] Wright, D.R., "A Modified u-Plot Applied to Failure Count Prediction," Technical Report, Centre for Software Reliability, City University, London, 1988.
- [Wrig93] Wright, D.R., *Recalibrated Prediction of Some Software Failure-Count Sequences*, PDCS2 Project (ESPRIT Project 6362) First Year Report, chap. 4, part 4.1, Centre for Software Reliability, City University, London, 1993.
- [Xie91a] Xie, M., *Software Reliability Modeling*, World Scientific Publishing Company, 1991.
- [Xie91b] Xie, M., and Zhao, M., *On Some Reliability Growth Models with Simple Graphical Interpretations*, Institute of Technology, Linköping, Sweden, ISSN 0281-5001, 1991.
- [Xie93a] Xie, M., "Software Reliability Models—A Selected Bibliography," *Journal of Software Testing, Verification and Reliability*, no. 3, 1993, pp. 3-28.
- [Xie93b] Xie, M., and Zhao, M., "On Some Reliability Growth Models with Simple Graphical Interpretations," *Microelectronics and Reliability*, vol. 33, no. 2, 1993, pp.149-167.
- [Yama83] Yamada, S., Ohba, M., and Osaki, S., "S-Shaped Reliability Growth Modeling for Software Error Detection," *IEEE Transactions on Reliability*, vol. R-32, no. 5, December 1983, pp.475-478.
- [Yama85] Yamada, S., and Osaki, S., "Software Reliability Growth Modeling: Models and Assumptions," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 12, December 1985, pp. 1431-1437.
- [Yama86] Yamada, S., Ohtera, H., and Narihisa, H., "A Testing-Effort Dependent Reliability Model for Computer Programs," *Tree Transactions of the IECE of Japan*, vol. E 69, no. 11, pp. 1217-1224, 1986.
- [Yama91] Yamada, S., Tanio, Y., and Osaki, S., "Software Reliability Measurement and Assessment Methods during Operational Phase and Their Comparisons," *Transactions Inst. Electronic Inf. Communication Engineering*, vol. J74-DI, March 1991, pp. 240-248 in Japanese.
- [Yau75] Yau, S.S., and Cheung, R.C., "Design of Self-Checking Software," *Proceedings of the 1975 International Conference on Reliable Software*, 1975, pp.450-457.
- [Youn86] Yount, L.J., "Use of Diversity in Boeing Airplanes," *Workshop on Design Diversity in Action*, Baden, Austria, June 1986, see summary in [Voge87b].
- [Your89] Yourdon, E., *Modern Structured Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey 1989.

## 820 References

- [Zave93] Zave, P., "Feature Interactions and Formal Specifications in Telecommunications," *IEEE Computer*, August 1993, pp. 20-29.
- [Zinn90a] Zinnel, K.C., "Using Software Reliability Growth Models to Guide Release Decisions," *Proceedings of the 1990 International Symposium on Software Reliability Engineering*, Washington, D.C., April 1990.
- [Zuse91] Zuse, H., *Software Complexity: Measures and Methods*, Walter De Gruyter, New York, 1991.

# Index

- Absolute performance, 141 Accelerated testing, 175 Acceptance test (*see* Acceptance testing)
- Acceptance testing, 570-571, 581 Acceptance voting, 584—585
  - reliability modeling, 603-604
- Accessibility matrix, 49 Accuracy:
  - model, 281
  - operational profile, 171
- Acquired software, reliability measurement, 235
- Action model, 58-60
- Activation function, 700-701
- Active fault, 28
- Adaptive maintenance, 32
- Adequacy criterion for test set, 534
- Adjudication:**
  - adjudicator, 575
  - algorithm, 577
  - by voting, 577-578
- Adjusted relative complexity, 503
- Adjustment of failure time, 240-241
- Administrator, system, 174
- Advantage:
  - implicit profile, 178-180
  - operational profile, 167—168
- Advice to reliability users, 162-163
- Akaike** information criterion, 297, 458
- Algorithm defect, 375
- Allocation of development resources, 172
- Alternates, 570
- Alternatives of model combinations, 295-298
- Analysis:
  - cluster, 326, 348
  - confirmatory, 450
  - environment, 448-449
  - exploratory, 449-450
- Analysis (*Cont.*):
  - failure dependency, 324-327
  - failure modes and effects, 233
  - fault tree, 233, 619-622, 624
  - Markov reward, 334
  - multiple discriminant, 511
  - multiple regression, 512
  - multivariate**, 326, 776-779
  - neural network model, 718
  - performability**, 351-352
  - principal component, 326
  - principal components, 495-499
  - sensitivity, 338-339
  - subadditive** property, 406-407
  - trend, 410
  - trend test, 410
  - variance, 716
- Application result for reliability measurement, 285-289
- Approach:
  - classification, 719
  - data domain, 73
  - early prediction, 162
  - error seeding and tagging, 73
  - estimative**, 719
  - experimental, 723
  - linear combination model, 289—299, 739
  - software reliability engineering, 220-226
  - time domain, 73
  - tree-based, 202
- Approximation of operational profile, 168
- Approximation of execution time, 240
- Architecture discovery process, 211—212
- Arithmetical mean test, 411—412
- Artifact, software, 663
- Artifact simulation (*see* Artifact-based simulation)

## 822 Index

- Artifact-based simulation, 665, 669-676
  - execution harness, 674
  - inputs, 671-672
  - reliability assessment, 674-675
  - results, 675-676
  - self-driven, 672-673
  - simulator architecture, 670—675
  - trace-driven, 674
- Artificial neural networks (see Neural networks)
- Artificial neuron, 700
- Assessing reliability capabilities of competitors, 227
- Assessment of quality factor, 222
- Assignment defect, 374
- Assumptions:
  - independence, 403, 580, 588
  - interversion failure correlation, 589
  - reliability growth, 450
  - stochastic independence, 403, 580, 588
- AT&T software reliability engineering toolkit, 732
- Atomic effort, 548
- Atomic systems, 35—41
- Attribute of dependability, 29, 32-33
- Attribute-value set, 394
- Authorized Program Analysis Report, 268, 362
- Autocorrelation coefficient, 777
- Automation:
  - diagnosis environment, 331
  - diagnosis of recurrences, 331-333
  - failure data collection, 245
  - selection of best predictor, 162
  - test, 238
  - verification of numerical results, 573-574
- Availability, 479-486
  - average, 62, 481
  - consideration in setting reliability objectives, 231
  - definition of, 14, 29, 35
  - instantaneous, 480-481
  - ra-step approximation, 484—485
  - pointwise, 61
  - prediction, 485-486
  - steady-state, 483 approximation, 484
  - system, 483
  - uptime, 481, 483
- Availability modeling, 60-64, 483^86
  - average availability, 62
  - Laplace transform, 62
  - Availability modeling (*Cont.*):
    - pointwise availability, 61
    - unavailability, 62-64
  - Average, 714
  - Average error, 714-715
  - Average failure rate, 481
  - Average recovery rate, 481
- Back-propagation learning algorithm, 705-707
- Back-to-back testing, 606
- Back-to-back voting, 582
- Background:
  - event, 207
  - model, 72-73
- Backward compatibility trigger, 388
- Backward elimination, 519
- Backward error recovery, 570
- Basic assumption, 78
- Basic block, 555
- Basic event in fault tree, 622
- Basis for characterizing system mode, 174-175
- Bath tub curve, 753-754
- Bayesian model, 104-109, 524-526
- Littlewood-Verrall (see Model: Littlewood-Verrall)
- Behavior (see Service)
- Belady's bandwidth metric, 513
- Bell-shaped response curve, 702
- Benefit of software reliability engineering, 210, 220-226
- Bernoulli distribution, 749
- Best current practice, 219
- Best estimate of reliability, 259
- Best predictor, automatic selection, 162
- Beta test, 241
- Bias:
  - average, 714
  - definition of, 139, 144-145
  - detection of, 145
  - trend (see y-plot)
  - unit, 701, 703
- Bimodal failure rate curve, 268-270
- Binomial distribution, 749
- Binomial model, 75, 77
- Birth rate, 762
- Birth-death model, calendar-time, 468
- Birth-death process, 762-765
- Block, 555
- Book overview, 8-12
- Book scope, 21-23
- Boundary value, 775

- Boundary version reliability, 592 **Braun**  
 statistic, 130 Brooks and Motley model,  
 92-93 Bug, 13, 362  
 (See a/so Fault) Busy  
 hour, 201 Butterfly model,  
 360
- $\chi^2$  statistic (see **Chi-square** statistic) **C-use**  
 (see Computational use) Calendar time:  
 birth-death model, 468  
 data requirements, 88  
 definition of, 14  
 reliability analysis, 461, 463-469 Call trees,  
 178-179, 202 Cascade architecture, 707  
 Cascade-correlation learning algorithm,  
 707-709 Catastrophic hazard, 624  
 Categorization of fault, 314-315 data  
 definition fault, 314-315 data fault, 314-315  
 incorrect computation, 314-315 missing  
 operation, 314-315 side effect of code update,  
 314-315 unexpected situation, 314-315 Causal  
 analysis, 364-366 Cause:  
 identifier, 395  
 system outages, 439-440 Certification of  
 reliability objectives,  
 240-241
- Change:  
 control, 235  
 in defect-type distribution with phase, 370  
 definition of, 363  
 during testing, 269-271  
 in evolving software, 276  
 local trend, 406  
 in trend, 407<sup>08</sup>
- Chapman-Kolmogorov** dynamics, 763
- Characteristics:  
 code structure, 670  
 error model, 343  
 project, 281-284  
 recovery model, 343  
 software environment, 669
- Characterization:  
 project, 278-279  
 system modes, 174-175 Checkpointing  
 and restart, 328
- Chi-square statistic:  
 definition of, 130  
 distribution, 750, 770  
 test, 714, 774-775 Classification:  
 accuracy of classifiers, 724  
 approach, 719  
 data, 280, 312, 517-518  
 error, 314-317  
 errortype, 516  
 failure, 31, 227  
 fault, 30-31, 314-317  
 model, 59  
 opinion-based, 373  
 severity, 12, 227-228, 477-478  
 semantic, 373
- Classification scheme, model, 73-76
- Classifier:  
 classification accuracy, 724  
 general, 442-443  
 minimal network growth using  
 cross-validation, 721-722  
 minimum distance, 722  
 multilayer network, 721  
**perceptron**, 721  
 software-specific, 442-443
- Clean-room development, 222 Clock  
 time, 14
- Cluster analysis, 326, 348, 778-779 Code  
 churns, 275
- Code identifiers for software systems, 448 Code  
 structural characteristics, 670 Coincident failure,  
 575, 596, 628, 635  
**Mold**, 575
- Collection of data, 159-160, 284-285
- Combination techniques for reliability  
 measurement, 161-162
- Combinatorial model, 758
- Command-driven system, 191
- Commercial tool, 730 Common-cause  
 fault, 576, 594 Comparison:  
 model, 120-127  
 model performance, 286-288, 292-295  
 sequences of predictions, 131
- Complexity:  
 control flow, 497, 522  
 coupling, 500  
 data structure, 521  
 dynamic, 504-510  
 functional, 505-506  
 metric, relative, 501  
 operational, 509-510

## 824 Index

### Complexity (*Cont.*):

program size, 496, 522 relative complexity, 501-503 adjusted, 503 metric, 501 scaled, 501 relative subgraph complexity, 509 space, 681 system, 500 time, 679, 681, 683  
Component configuration change in evolving software, 276 Component failure rate, 46 interface failure probability, 46 internal component failure rate, 46 Component model, 289 Compression ratio, 548 selection, 551-553 Computation complexity (see Time complexity)  
Computation error model, 646 Computation diagnostics, 457-458 Computational use, 533 Computer-Aided Software Reliability Estimation tool (**CASRE**), 739-740 Concurrent engineering, 667 Conditional probability, 749-750 Conditional probability density function, 749-750  
Conditional reliability, 55, 57—58  
Confidence: bound, 260 coefficient, 769 interval, 260, 769 for means, 769-770 for proportions, 771 for variances, 770-771 Confidentiality, 29 Configuration: change in evolving software, 276 management system, 503 profile, 202 vectors, 503 Confirmatory analysis, 450 Consensus recovery block, 583-584 reliability modeling, 601-603 Consensus voting, 578 reliability, 592-593, 599-601 versus majority voting, 600 Consistency check, 320 data, 448 external, 571-573

### Constant:

**failure-rate model**, 474-475  
**hazard**, 754  
**subadditivity** factor, 408-409 Continuous distribution, 750 Continuous random variable, 748-749 Continuous-space Markov model, 758 Continuous-time Markov model, 758-759 Continuous-time system reliability, 36-37, 40  
Control: charts, 423 error, 315-317 testing progress, 422 Control flow: complexity, 497, 522 testing, 532-533 Convolution operation, 54 Corrective maintenance, 32 Corrective service, 329 Correlated elements in functional and operational profiles, 183 Correlated failure, 324-325, 576, 598-599 Correlated fault, 596 Correlation analysis, 777 Correlation coefficient, 325-326, 777 Corruption, data, 197, 332, 449 Cosine arch moving average, 376  
Cost: fault -tolerant software, 607-609 testing, 265 life -cycle, 229-230 operational failure, 264—265 operational profile, 168, 171, 222 orthogonal defect classification, 397 processing and analyzing failure data, 223 reduction, 168 root -cause analysis, 397 software reliability engineering consultant, 223 Coupling complexity, 500 **Covariance** matrix, 497, 516 **Covariance**, 449, 708  
Coverage: definition of, 34 failure, 647 incorporation in reliability estimation, 546-547 information for failure data filtering, 547-551 model, 646-647

- Coverage (*Cont.*):
  - parameter, 337, 647
  - statistics, 544 *Cp*
- statistic, 520
- CPU-bound workload, 349-350
- Criteria for selecting software *reliability* tool, 731-732
- Critical hazard, 624
- Critical operation, 175
- Criticality*:
  - category, 176
  - measurement, 175
- Cross-referencing between logs, 311
- Cross-validation:
  - minimal network growth, 721-722
  - neural network, 724
  - set, 722
- Cumulative distribution function, 403, 748
  - empirical, 773
- Cumulative failure function, 14
- Cumulative failure plot, 452
- Cumulative reward, 352
- Customer:
  - definition of, 173, 175
  - expectations, 206
  - partnership, 444
  - perspective for quality measurement, 221
  - satisfaction tracking, 245
  - testing, 202-203
  - type, 173
- Cut-off point, 485
- Cutset:
  - definition of, 617
  - generation, 617-619
  - minimal, 617, 630-634
- Cycle-time reduction, 360
- Cyclomatic* complexity number (see *Cyclomatic* number)
- Cyclomatic* number, 497, 500, 513, 521
- Data aging, 454
- Data analysis:
  - detailed analysis, 323-333
  - preliminary analysis, 312-323
  - principles, 449-459
- Data analysis techniques:
  - data modeling, 450
  - data resistance, 450
  - data transformation, 450, 458-459, 739
  - plots and graphs, 450
- Data and Tool Disk, 12
- Data behavior, 454
- Data classification, 280, 312, 517-518
- Data coalescing, 313
- Data collection and analysis environment:
  - data collection system, 449
  - manipulation, filtering, and data merging system, 449
  - software reliability modeling tool, 449
  - statistical analysis and visualization system, 449
- Data collection and analysis process,
  - standardization, 440
- Data collection, 159-160, 284-285
  - automation, 245
  - form, 285
  - in field, 244-245
  - in system test, 239
  - items, 271
  - principle, 441-449
  - process, 266-267
  - plan, 266
  - system, 449
  - trial-run, 267
  - with multiple functional area, 272
- Data consistency checking, 448
- Data conversion (see *Data transformation*)
- Data corruption, 197, 332, 449
- Data definition, 267-272
- Data diagnostics, 454-455
- Data Disk (see *Data and Tool Disk*)
- Data domain:
  - approach, 73
  - reliability modeling (see *Reliability modeling in data domain*)
- Data-driven system, 193
- Data extraction, 313
- Data fault, 314-315
- Data filtering:
  - definition of, 449
  - filtering using coverage information, 547-551
  - process, 549-550
- Data flow:
  - diagrams, 211
  - testing and coverage, 532-534
- Data granularity, 446-447
- Data maintenance, 447-448
- Data modeling, 450, 454-455
- Data preprocessing, 312-314
- Data *recalibration* with respect to usage,
  - 454
- Data recording, 190
- Data references, global, 521
- Data representation, 712-713, 722-723
- Data resistance, 450



## 826 Index

### Data set:

fitting, 513 validation, 513 Data smoothing, 454, 739 Data-splitting, 513 Data structure complexity, 521 Data transformation, 17,450,458-459, 739 Data validation, 447-448 Data variability, 439 Deadlock error, 315-316 Death rate, 762 Debugging phase, 753 Decision coverage, 533-534 Decision-making process, 367 Def-use pair, 533 Defect: definition of, 13, 361 density, 113 latent, 465 multidimensional analysis, 393—396 reduction, 360 reintroduction rate, 468 tool for tracking, 394 Defect model, 364-366 Defect prevention process, 397 Defect simulation status profile, 691-693 Defect trigger, 384-393 attribute, 384-393 function test triggers, 388-389 combination path coverage trigger, 389 simple path coverage trigger, 389 test coverage trigger, 389 test interaction trigger, 389 test sequencing trigger, 389 test variation trigger, 389 review and inspection triggers, 387-388 backward compatibility trigger, 388 concurrency trigger, 388 design conformance trigger, 388 document consistency/completeness trigger, 388 lateral compatibility trigger, 388 operational semantics trigger, 388 rare situation trigger, 388 system test triggers, 387 hardware configuration and software configuration trigger, 387 normal mode trigger, 387 recovery/exception handling trigger, 387

Defect trigger, attribute, system test triggers (Cont.):  
systemstart-up and restart trigger, 387 workload volume/stress trigger, 387 definition of, 385 difference with symptom, 386 distribution, 391-393 use of, 389-393 Defect type, 368, 373-384 association with process, 376 attribute, 374-376 algorithm defect, 375 assignment defect, 374 build/package/merge defect, 374 checking defect, 374 defect due to something incorrect, 374 defect due to something missing, 374 documentation defect, 374 function defect, 374 interface defect, 374 timing/serialization defect, 374 definition of, 368, 373 distribution, 390-391 change with phase, 370 relative risk assessment, 376-384 Definition-free path, 533 Delayed S-shaped model (see Model: S-shaped model) Density function (see Probability density function) Dependability, 28-35 attribute of, 29, 32-33 impairment to, 28-32 means for, 33-35 requirement, 65-66 Dependence of faults/failures, 403, 586-589 Description statistics, 423 Design and implementation phase: software reliability engineering, 232-235 trend analysis on inspection data, 431-434 Design conformance trigger, 388 Design constraints, 261 Design fault, 19 Design methodology versus operational profile, 173 Determination: functional profile, 226-227 new feature introduction, 245—246 operational profile, 236-237

- Development:  
 developer-oriented measures, 221  
 improvements of developer and customer
- communication, 199  
 methodology, 234  
 operational profile, 206-208, 211-214  
 practices for fault management, 234-235  
 resources allocation, 172  
 test environment, 208
- Diagnosis,  
 symptom-based, 332
- Diagnostics for model  
 determination, 455-458  
 computational, 457-458  
 graphical, 455-457
- Differential-difference  
 equation, 764
- Discovery reviews, 231
- Discrete  
 distribution, 749
- Discrete random variable,  
 748-749
- Discrete-event simulation, 665
- Discrete-space Markov model, 758
- Discrete-time Markov model, 759
- Discrete-time  
 system reliability, 36, 39-40
- Discriminant  
 model, 515-516
- Discrimination technique, 515
- Disk (see Data and Tool Disk)
- Distributed  
 recovery block:  
 definition of, 581, 627, 629  
 fault tree model, 629-630  
 reliability model, 630  
 system reliability model, 647-648  
 system safety model, 647-648
- Distributed  
 systems, operational profile,  
 198-199
- Distribution:  
**characterization**, 772-776  
 function (see Cumulative distribution  
 function) fitting,  
 773-774
- sample, 135
- Bernoulli, 749  
 binomial, 749  
**chi-square**, 750, 770  
 continuous, 750  
 defect trigger, 391-393  
 defect type, 390-391  
 discrete, 749  
**empirical**, 772-773  
 exponential, 750  
 Gamma, 750  
 normal, 750  
**Poisson**, 749  
**Rayleigh**, 94, 755
- Documentation:  
 defect, 374  
 document completeness trigger, 388  
 document consistency trigger, 388  
 operational profile, 203-204
- Domain coverage  
 and predictive model,  
 520-523
- Domain loading, 498
- Domain metric, 498
- Domain pattern, 498, 514,  
 522-523
- Domain testing, 554
- Dormant fault, 28
- Dot-product function, 701
- Doubly stochastic  
 process, 346
- Duane** model (see Model: **Duane**)
- Duration of outage, 13
- Dynamic aspect of  
 functional complexity,  
 507-509
- Dynamic consensus voting, 584
- Dynamic majority voting, 584
- Dynamic program  
 complexity, 504-510
- Dynamic simulation:  
 approaches, 664-665  
 need, 663-664
- Dynamically weighted linear combination  
 model:  
 basic model, 291  
 model with fixed window, 296  
 model with sliding window, 296
- π** point, 378-379, 382
- Early prediction model,  
 17, 111-114,  
 162
- phase-based model, 111-112
- predicting defects  
 from Ada designs,  
 112-113
- Rome Laboratory model, 113-114
- Economic Stop Testing Model (ESTM)**  
 tool, 740-741
- Effect identifier, 395
- Effort:  
 atomic, 548  
 estimated, 496  
 filtered, 548  
 testing, 540
- Element, 168
- I—point symmetrical  
 moving-average,  
 481
- Elman** network (see Recurrent  
 network:  
 simple recurrent network)
- End-point prediction, 711, 714
- End-to-end  
 reliability process, 691
- Entity relationship  
 diagram, 211

## 828 Index

### Environment:

- automatic diagnosis, 331
- characteristics, 669
- consideration, 688-689
- data collection and analysis, 448-449
- profile, 181
- variable, 175-176, 180-181, 190, 192 Equally  
    **wrighted** linear combination  
    model, 290

### Error:

- average, 714
- definition of, 13, 28-29, 362-363
- detected, 28
- latent, 28
- relative**, 458
- Error bursts (see Multiple error) Error  
compensation, 570 Error confinement area,  
627 Error distributions, 320-323 Error entry,  
sample, 308-309 Error frequency, 320-321  
Error log, preprocessing 313 Error model,  
319, 337-344, 349-351 Error propagation,  
317-320 Error record, 308-309 Error  
recovery, 340, 570
- distributions, 320-323 Error report:  
    Authorized Program Analysis Report (**APAR**),  
268, 362  
    Modification Request (**MR**), 235, 268, 284  
    Problem Failure Report (**P/FR**), 267  
    Tandem Project Report (**TPR**), 311 Error  
seeding and tagging, 73 Error states, 340  
    mean waiting time, 340, 342 Error statistics,  
320 Error types, 315-317  
    control error, 315-317  
    deadlock error, 315-316  
    I/O and data management error, 316-317  
    memory error, 317  
    multiple error, 316, 322  
    programming exception error, 316, 344  
    storage exception error, 316, 344  
    storage management error, 316  
    Establishment of software reliability  
    requirements, 259—266
- Estimated effort, 496 Estimation:  
    least-squares, 520, 768-769  
    number of sites, 475  
    software reliability, 17

### **Estimative** approach, 719

- Estimator, 139, 766
  - consistent, 766
  - least-squares, 768-769
  - maximum likelihood, 767
  - unbiased minimum-variance, 766 **Euclidean**

### distance, 778 Evaluation:

- fault tolerance, 328, 570
- model, 342
- operational phase, 306
- reliability, 64-68 Events, **reportable**, 473, 483

### Event process:

- simulation, 665, 676
- statistics, 677 Evolving software:
  - complexity metrics, 502
  - component configuration change, 240, 276
  - failure time adjustment, 240, 276-277
  - ignoring changes, 240, 276 Exception  
handling, 328, 571
  - trigger, 387
  - operating system exceptions, 573 Exclusion:
    - remaining set of run types, 188-189
    - transient behavior data, 454 Execution of  
programs, 209—210 Execution profile, 505
    - relationship to relative complexity,  
506-507

### Execution time:

- approximation, 240
- component data requirements, 88
- definition of, 14
- measurement, 239-240 Expected life of  
system, 756 Expected value (see Mean)

### Experimental approach, 723 Explanatory

- variables, 162 Explicit versus implicit  
functional profile,  
177-180

### Exploratory analysis, 449-450 Exponential

- distribution, 750 Exponential failure time  
class model, 77-93
  - Brooks and Motley (see Model: Brooks and  
Motley)

- extended execution time (see Model:  
    extended execution time)

- hyperexponential** (see Model:  
hyper-exponential)

- Exponential failure time class model  
(*Cont.*):
- Inflection **S-shaped** (see Model: Inflection **S-shaped**) **Jelinski-Moranda** (see Model: **Jelinski-Moranda**) **Laprie hyperexponential** (see Model: **Laprie hyperexponential**) **Musa basic execution time** (see Model: **Musa basic execution time**) **nonhomogeneous Poisson process** (see Model: nonhomogeneous Poisson process) **Schneidewind** (see Model: **Schneidewind**)
  - Yamada modified exponential growth** (see Model: Yamada modified exponential growth) Exponential order statistic model class, 111
- Exponential random variables, 140  
Exposure period, 540  
Expressing software reliability, 259–261  
Extended **excution** time model, 92 Extensions of model combinations, 295-298  
Eyeball regression, 694
- Factor analysis, 326, 777-778  
common factor, 778  
**communality**, 778  
factor loading, 778  
unique factor, 778  
unique variance, 778 Factors affecting software reliability, 339 Factors in reliability objectives, 230-231 Fail-safe state, 647 Failure:
- coincident (see Coincident failure)
  - correlated (see Correlated failure)
  - definition of, 12, 28-29, 227, 362-363
  - examples of, 4-5
  - fatigue, 754
  - interweaving, 53
  - load-induced, 348
  - median time to next, 128
  - multiple component, 325
  - single point, 619
  - wear-out, 754
  - workload-related, 345-352 Failure classification, 31, 227 Failure data: automated collection, 245
- Failure data (*Cont.*):  
definition of, 15, 442-444  
failure-count, 15, 268, 271  
failures per time period, 15, 268, 271  
filtering process, 549–550  
format, 410  
**interfailure times**, 16, 268, 271  
**time-between-failures**, 16, 268, 271 Failure density function, 751 Failure dependence, 586-589 Failure dependency analysis, 324–327 **multiway**, 326-327 two-way, 325-326 Failure domain, 36, 38-39 Failure frequency, 320-321 Failure functions, 14  
cumulative failure function, 14  
failure intensity function, 14  
failure rate function, 14  
mean time between failures (see Mean time to failure)  
mean time to failure, 14  
mean value function, 14  
**MTBF** (see Mean time between failures)  
**MTTF** (see Mean time to failure)  
rate of occurrence of failures, 14, 55, 128  
**ROCOF** (see Rate of occurrence of failures)
- Failure impact, 333-335, 386 Failure independence, 580 Failure intensity:  
decay parameter, 470  
definition of, 757-758  
function, 14  
improvement factor, 258, 263  
influence of usage, 462  
Laplace test, 413  
model, 59  
optimum objective value, 229-230  
plots, 452-453 Failure model, 337-344, 349-351, 385-386  
Failure mode, 31  
Failure mode and effect analysis, 233 Failure rate:  
average, 481  
**bimodal curve**, 268-270  
definition of, 752-756  
field, 482  
function, 14  
instantaneous, 480  
model, 59

## 830 Index

Failure rate (*Cont.*):

per-office, 475

system, 49 Failure rediscovery, 53 Failure

severity classification, 12, 227-228,

477-478 Failure source, 65-66

Failure statistics, 320 Failure symptom, 385

Failure time adjustment, 240-241

in evolving software, 276-277 Failure-based

measure, 221 Pan-in and fan-out connections,

700-701 Fatigue failure, 754 Fault:

active, 28

common-cause, 576, 594

definition of, 13, 28-29, 362-363

design, 19

dormant, 28

first occurrence of, 315

physical, 18

regression, 403, 423

related, 594, 596, 628, 636

similar, 576, 594, 596

spawned, 196 Fault avoidance, 34 Fault-based

measures, 221 Fault classification, 30-31,

314-317 Fault categorization, 314-315

data definition fault, 314-315

data fault, 314-315

incorrect computation, 314-315

missing operation, 314-315

side effect of code update, 314-315

unexpected situation, 314-315 Fault data,

442-444 Fault density, 112-114 Fault

dependence, 403 Fault discovery rate, 688-689

Fault-exposure ratio, 673 Fault forecasting, 19,

21, 29 Fault injection, 673 Fault management,

234-235 Fault masking, 570 Fault model,

Gompertz curve, 467 Fault pathology, 28-29

Fault prediction model, 518-520 Fault

prevention, 19-20, 29 Fault profile, 314

Fault-prone module identification, 710,

718-725 Fault rate,

112 Fault recurrence, 315

Fault removal, 19-20, 29 Fault

sensitization, 402 Fault simulation,

691-693 Fault tolerance, 19-20, 29

(See *ffl/so* Fault-tolerant software)

Fault-tolerant software:

advanced techniques, 583-585 acceptance

voting, 584-585 consensus recovery block,

583-584 dynamic consensus voting, 584

dynamic majority voting, 584 *N*

self-checking programming,

583-586 applications,

568-569 basic techniques,

581-583

distributed recovery block, 581, 627,

629

*N*-version programming, 582-583, 627 recovery

blocks, 581-582 comparative system analysis,

642-645 cost, 607-609 definition of, 567-568

design, 605-606 development, 604-609

evaluation, 328, 570 experiment, 596-598 fault

tree analysis, 627 hybrid techniques, 583-585

parameter estimation, 639 *performability*

analysis, 628 present status, 568-569

quantitative analysis, 635-645 reliability in

presence of *interversion*

failure correlation, 596-604

reliability growth modeling, 65,

585-596 requirement, 605-606 testing,

604-609 validation and verification,

606-607 Fault tree:

analysis of *fault-tolerant* software, 627

analysis, 233, 619-622, 624 assignment

statement, 624 basic event, 622 construction,

616 as design aid, 622-623 *if-then-else*

statement, 624 modeling, 615-617, 629-634

distributed recovery block, 629-630 *N*

self-checking programming,

632-634

*JV*-version programming, 630-632

recovery block, 629-630 safety validation,

623-627

- Feasibility and requirement phase of software reliability engineering, 226-231
  - Feature:
    - coarse, 538
    - definition of, 536-537
    - granularity, 537
    - interacting, 538
    - testing, 238
    - tasks, 175
  - Feed-forward network, 703-704, 712
  - Field data, 244-245, 440
  - Field failure rate, 482
  - Field failure ratio, 467
  - Field reliability:
    - compared with objectives, 243—245
    - data variability, 439
    - projected, 259
    - reliability growth, 475-476
  - Field repair rate, 482
  - Field trial, 241
  - Field-defect model, **Weibull-type**, 464-465
  - Field required for reliability analysis, 443
  - Filtering of failure data, 547-551
  - Finite failure model, 73
  - First occurrence:
    - error, 318
    - failure, 244-245
    - fault, 315
  - First office application, 241
  - Fitting data set, 513
  - Fluctuation, 404, 433
  - Forecasting:
    - failure, 19, 21, 29
    - fault, 19, 21, 29
    - long-range, 447
  - Formal design review, 622
  - Formal inspection, 20
  - Formal methods, 19-20
  - Forward error recovery, 570
  - Forward inclusion, 519
  - F-statistics**, 717
  - Function, 168-170, 176
    - mapping to operation, 184
    - number of, 177, 181
  - Function defect, 374
  - Function list, 180-181
  - Function test trigger, 388-389
  - Functional complexity, 505—506
    - dynamic aspect, 507-509
    - variability, 509
  - Functional profile, 169, 176-183, 211-212
    - determination, 226-227
  - Functional profile (*Cont.*):
    - explicit versus implicit, 177-180
    - number of functions, 177
    - occurrence probability, 181-183
    - Functional testing, 268-270, 427, 534
  - Gain factor, 702
  - Gamma distribution, 750
  - Gaussian** function, 702
  - General rate-based simulation, 684-686
  - General-purpose tool, 730
  - Generalization training, 712
  - General classifiers, 442-443
  - Generalized exponential model class, 109-110
  - Generalized squared distance, 515
  - Generic users, 201
  - Geometric model (see Model: geometric)
  - Geometric **Poisson** model (see Model: geometric **Poisson**)
  - Global data reference, 521
  - Gnedenko** test, 411
  - Golden answer (see Correct answer)
  - Gompertz** curve fault model, 467
  - Goodness-of-fit**, 714
  - Gradient descent procedure, 705-706
  - Graphical diagnostics, 455-457
  - Graphical interpretation of **subadditive** property, 404-406
  - Graphical trend test, 410
  - Graph, 450-454
  - Group run type, 169
  - Halstead** number, 496, 513
  - Handicap value, 292
  - Hard-limiting threshold function, 701
  - Hardware:
    - configuration trigger, 387
    - recovery, 350
    - reliability, relationship to software reliability, 18-19
    - transient fault, 628
  - Hardware-related software failures, 327-328
  - Hazard, 624
    - catastrophic, 624
    - constant, 754
    - critical, 624
    - definition of, 752
    - function, 677
    - linearly increasing, 754—755
    - per-fault** rate, 689
  - Heisenbug**, 574



## 832 Index

- Hidden layer, 703 Hierarchy of interpreters, 43
- High-risk area identification, 233
- Histogram, 772 Hollander test, 411, 414
- Homogeneous Markov process, 759
- Homogeneous Poisson process, 74
- Homogeneous run types, 194 Human triggers, 388
- Hyperexponential model (see Model: hyperexponential)
- Hypothesis test, 410-411, 450, 774
- Identical and wrong answer, 575-576, 598
- Identification:
  - customer reliability needs, 227-228
  - fault-prone software modules, 710, 718-725
  - high-risk area, 233
  - Identifier:
    - cause, 395
    - effect, 395
    - subpopulation, 395
  - Imminent-failure state, 450
  - Impact of failures on performance, 333—335
  - Impact of system activity, 345—352
  - Impairment to dependability, 28-32
  - Implicit profile advantage, 178-180
  - Improvement:
    - developer and customer communication, 199
    - method, 160-163
    - product and process, 246
    - quality, 222
  - Inclusion-exclusion method, 619
  - Incorrect computation, 314—315
  - Incremental build problem, 526—527
  - Independence:
    - assumption, 403, 580, 588
    - failure, 580
    - fault, 628
    - input sequence, 40
  - Index of stability, 231
  - Infant mortality, 753
  - Infinite failure class model, 73, 98-104
    - Duane (see Model: Duane)
    - geometric (see Model: geometric)
    - Lipow (see Model: Lipow)
    - Moranda geometric Poisson (see Model: Moranda geometric Poisson)
    - Musa-Okumoto logarithmic Poisson (see Model: Musa-Okumoto logarithmic Poisson)
  - Inflection point in reliability growth curve, 378
  - Inflection S-shaped model, 92, 381
  - In-process measurement, 367
  - Input domain partitioning, 551
  - Input layer, 703
  - Input space, 170
  - Input state, 169
  - Input trajectory, 38-39, 402
    - (See also Failure: domain)
    - Input variable, 169
      - indirect, 197
      - key, 177
      - number reduction, 186
  - In-service time, 445, 460
  - Inspections and reviews, 234, 431-432
  - Instantaneous availability, 480-481
  - Instantaneous failure rate, 480
  - Instantaneous recovery rate, 480
  - Instantaneous reward rate, 760
  - Integrity, 29, 32
  - Interaction matrix, 191
  - Interdependencies among project elements, 668-669
  - Interface:
    - defect, 374
    - failure probabilities, 46
    - interpretive, 43
    - Interfailure times:
      - data, 16
      - Laplace test, 412
    - Interpreters:
      - hierarchy, 43
      - system, 43
    - Interpretive interface, 43
  - Interrupt, externally initiated, 169
  - Interval estimation, 769-772
  - Interval reliability, 55, 58
  - Interval reward rate, 760
  - Interversion failure correlation assumption, 589
  - Interweaving of failure, 53
  - I/O and data management error, 316-317
  - I/O workload, 349
  - Iterative development process, 363
  - Jelinski-Moranda model, 79
  - Jensen's estimator, 513
  - Jordan network (see Recurrent network: semirecurrent network)
  - modified, 712
  - Jump-start consulting, 223

- Kendall** test, 411, 737 Key input variables, 177  
 indirect, 197  
 interaction matrix, 191 Key operational input selection, 196 Key quality objectives, 257 *K-fold* coincident failure, 575 **X-means** algorithm, 349, 779 Knee of reliability growth curve, 378 Knowledge model, 58-60 **Kolmogorov-Smirnov** distance, 136, 738 **Kolmogorov-Smirnov** test, 714, 775-776 **KS** distance (*see* Kolmogorov-Smirnov distance)
- Laplace factor, 412-413  
 Laplace test, 411, 737  
 failure intensity, 413  
 interfailure times, 412  
 relation to **subadditive** property, 414-419
- Laplace transform:  
 availability, 62  
 failure intensity, 56  
**Laprie hyperexponential** model, 92, 468 Latent defect, 465 Latent discovery effect, 348 Latent error, 28  
 Lateral compatibility trigger, 388 Law of large numbers, 693 Layer, 41-42, 703 Learning algorithm, 700, 705, 707  
 back-propagation, 705-707  
 cascade-correlation, 707-709 Learning rate, 706 Least significant difference, 717  
 Least-squares estimation, 520, 734, 739, 768-769 License  
 information, 445 Life-cycle cost, 229-230  
 optimization for reliability specification, 263-266 Likelihood function, 767 Limitation: models, 76-77  
 present prediction techniques, 159-160  
 testing methods, 542-543 Limited growth simulation, 683 Linear combination model, 289-299, 739 Linear discriminant model, 515  
 Linear relationships in reliability growth models, 459 Linear threshold function, 701
- Linearly increasing hazard, 754-755 Lines of code, 495 **Lipow** model, 102 List:  
 customertype, 173  
 system mode, 174-176  
 usertype, 173-174 Listing operations, 184-186 **Littlewood-Verrall** model, 107-108  
 Load hazard model, 346-348 Load test program execution, 209-210 Load-induced effects, 348  
 Load-induced software failures, 348 Local fluctuations, 404 Local trend changes, 406 Log, cross-referencing, 311 Logging:  
 challenges in on-line evaluation, 310  
 on-line event, 308-309  
 on-line machine, 308-310  
 operator, 311 Logical time, 445 Logistic function, 702 Long-term prediction, 123-126, 159,  
 298-299, 447, 456-457, 711 Loss in performance measure, 761
- Maintainability, 29, 32  
 Maintenance:  
 adaptive, 32  
 corrective, 32  
 perfective, 32  
 personnel, 174  
 policy, 51-53 Majority voting:  
 comparison with consensus voting, 600  
 definition of, 577  
 reliability, 592-593 Management of fault, 234-235 Manipulation, filtering, and data merging  
 system, 449 Manual reporting, 308, 310-312 Mapping from functions to operations, 184  
 Markov chain, 43-44, 47, 63 Markov model, 646, 758-759  
 continuous-space, 758  
 continuous-time, 758-759  
 discrete-space, 758  
 discrete-time, 759 Markov process:  
 homogeneous, 759  
**nonhomogeneous**, 759

## 834 Index

- Markov reward analysis, 334, 759-762 Matrix:  
accessibility, 49  
covariance, 497, 516  
rediscovery, 329-330  
service utilization, 48  
standardized transformation, 498-499,  
514-515 Maximum likelihood  
estimation, 734, 736,  
738-739, 766-767 Mazzuchi's modification of  
Littlewood-  
Verrall model, 108 McCabe's cyclomatic  
number, 497, 500,  
513, 521
- Mean, 748  
Mean recurrence time, 342 Mean square error,  
299 Mean time between failures (see Mean  
time to failure) Mean time to failure,  
14, 756-757 Mean time to repair, 14 Mean value  
function, 14 Mean value voting, 578 Mean  
waiting time in error states, 340,  
342
- Means for dependability, 33-35
- Measure:  
developer-oriented, 221  
failure-based, 221  
fault-based, 221  
predictability evaluation, 713-715  
static, 502  
validation of, 521 Measurement:  
connection with semantics, 372  
criticality, 175  
customer perspective of quality, 221  
definition of, 360-361  
execution time, 239-240 in-process, 367  
process-based, 372  
reliability for acquired software, 235  
software reliability, 17  
(See also Software reliability measurement)
- Measurement-based analysis:  
overall framework, 304  
steps, 304-305  
studies, 307
- Measurement plan quality, 471 Measurement  
program, operational-profile, 198
- Measurement system, multidimensional,  
360
- Measurement techniques, 307-312 Median  
time to next failure, 128 Median voting, 577,  
583 Median-weighted linear combination  
model, 291
- Meeting reliability objectives, 232-233
- Memory error, 317 Merge defect, 374 Method:  
analyzing predictive accuracy,  
128-145 improvement, 160-163  
inclusion-exclusion, 619 least square,  
768-769 maximum likelihood, 766-767  
moments, 767-768 Metrics:  
Belady's bandwidth metric, 513 definition  
of, 495-496 domain, 498 Jensen's estimator,  
513 path, 449 usage, 499-500 Middleware  
software, 390 Minimal cutset, 617, 630-634  
Minimal network growth using  
cross-validation, 721-722 Minimizing  
number of recurrences,  
329-331
- Minimum distance classifier, 722 Missing  
operation, 314-315 Mistake, 13  
(See also Error) Mode  
of failure, 31 Model:  
action, 58-60 Bayesian, 104-109,  
524-526 binomial type, 75, 77 Brooks  
and Motley, 92-93 Butterfly, 360  
computation, 646 constant-failure-rate,  
474-475 coverage, 646-647  
Duane, 98-99 assumptions, 98 data  
requirements, 98 model estimation, 99  
model form, 99 overview, 98  
reliability prediction, 99

- Model (*Cont.*):
- dynamically weighted linear combination, 291, 296
  - equally weighted linear combination, 290
  - error, 337-344, 349-351
    - multiple occurrence, 344-345
    - propagation, 319
  - exponential failure time class, 77-93
  - extended execution time, 92
  - failure, 337-344, 349-351
    - finite, 73
    - infinite, 73
    - intensity, 59
    - no-failure, 474
    - rate, 59
    - geometric, 99-102
      - assumptions, 100
      - data requirements, 100
      - model estimation, 101
      - model form, 100-101
      - overview, 99-100
      - reliability prediction, 101
    - hyperexponential, 90-92
      - assumptions, 90-91
      - data requirements, 91
      - model estimation, 91
      - model form, 91
      - overview, 90
      - reliability prediction, 91
  - infinite failure class, 98-104
  - inflection S-shaped, 92, 381
  - interconnection structure, 700
  - Jelinski-Moranda, 77-80
    - assumptions, 78
    - data requirements, 78
    - model estimation, 79
    - model form, 79
    - overview, 77-78
    - reliability prediction, 79
  - knowledge, 58-60
  - Laprie hyperexponential, 92, 468
  - linear combination, 289-299, 739
  - Lipow, 102
  - Littlewood-Verrall, 105-109, 524
    - assumptions, 106
    - data requirements, 106
    - model estimation, 107-108
    - model form, 106-107
    - linear form, 107
    - quadratic form, 107
    - overview, 105
  - Model, Littlewood-Verrall (*Cont.*):
    - reliability prediction, 107-108
    - load hazard, 346-348
    - median-weighted linear combination, 291
  - Moranda geometric Poisson, 102
  - Musa
    - basic execution time, 87-90
      - assumptions, 87-88
      - data requirements, 88
        - calendar time component, 88
        - execution time component, 88
      - model estimation, 89-90
      - model form, 88-89
      - overview, 87
      - reliability prediction, 89-90
  - Musa-Okumoto logarithmic Poisson, 102-104
    - assumptions, 102
    - data requirements, 102
    - model estimation, 103
    - model form, 102-103
    - overview, 102
    - reliability prediction, 103
  - nonhomogeneous Poisson process (NHPP), 80-82
    - assumptions, 80
    - data requirements, 80
    - model estimation, 81
    - model form, 81
    - overview, 80
    - reliability prediction, 81-82
  - nonparametric, 699
  - neurons, 700
  - parametric, 709
  - Poisson type, 74-75, 77
  - Rayleigh, 94
  - rare-event, 473-476
  - recovery, 337-344
  - resource-usage, 349-351
  - Schick-Wolverton, 94
  - Schneidewind, 82-87
    - assumptions, 83
    - data requirements, 83
    - model estimation, 84-86
    - model form, 84
    - overview, 82-83
    - reliability prediction, 84-86
  - simulation, 662
  - software quality, 511
  - software reliability, 17-18
  - S-shaped, 95-98
    - assumptions, 95-96

## 836 Index

- Model, **S-shaped** (*Cont.Y* data requirements, 96 model estimation, 97-98 model form, 96-97 overview, 95 reliability prediction, 97-98 system, 41-43, 646 **ultrareliability**, 277-278 unequally weighted linear combination, 291 weight determination based on ranking
  - model results, 290-291 **Weibull** and gamma failure time class, 93-98, 756 **Weibull**, 93-95, 756 assumptions, 93-94 data requirements, 94 model estimation, 95 model form, 94-95 overview, 93 reliability prediction, 95 workload, 348-349 **Yamada** modified exponential growth, 91
- Model accuracy, 126, 281 Model application issues, 273-276 Model assumptions:
  - basic, 78 **Duane** model, 98 general, 274-275 geometric model, 100 **hyperexponential** model, 90-91 **Jelinski-Moranda** model, 78 **Littlewood-Verrall** model, 106 **Musa** basic execution time model, 87-88 **Musa-Okumoto** logarithmic **Poisson** model, 102 **nonhomogeneous** **Poisson** process model, 80 **S-shaped** model, 95-96 **Schneidewind** model, 83 standard, 78 **Weibull** model, 93-94 Model background, 72-73 Model bias (see **u-plot**) Model classification, 59, 73-76 Model combinations, extensions and alternatives, 295-298 Model comparison, 120-127 Model determination diagnostics (see Diagnostics for model determination)
- Model disagreement, 121 Model estimation:
  - Duane** model, 99 geometric model, 101 hyperexponential model, 91 **Littlewood-Verrall** model, 107-108 **Musa** basic execution time model, 89-90 **Musa-Okumoto** logarithmic **Poisson** model, 103 **Nonhomogeneous** **Poisson** process model, 81 **S-shaped** model, 97-98 **Schneidewind** model, 84-86 **Weibull** model, 95 Model evaluation, 342
- Model form:
  - Duane** model, 99 geometric model, 100-101 hyperexponential model, 91 **Jelinski-Moranda** model, 79 **Littlewood-Verrall** model, 106-107 **Musa** basic execution time model, 88-89 **Musa-Okumoto** logarithmic **Poisson** model, 102-103 **nonhomogeneous** **Poisson** process model, 81 **S-shaped** model, 96-97 **Schneidewind** model, 84 **Weibull** model, 94-95
- Model limitations, 76-77 Model performance comparisons, 286-288, 292-295 Model relationship, 109-111 Model selection, a priori, 126-127 Model trade-offs study, 230 Modification Request (MR), 235, 268, 284 Modified renewal process, 56-57 Monitoring field reliability versus objectives, 243-245 Monotone reliability growth, 404 Moving average:
  - cosine arch, 376 **11-point** symmetrical, 481 three-point symmetric, 465-466 MR (see Modification Request) **MTBF** (see Mean time between failures) **MTTF** (see Mean time to failure) **MTTR** (see Mean time to repair)
- Multi-interpretor system, 47-49 **Multicollinearity**, 499

- Multidimensional defect analysis, 393-396
- Multidimensional measurements system, 360
- Multilayer network, 703
  - classifier, 721
  - Multiple checks, 332-333
- Multiple component failure, 325
- Multiple correct output, 577
- Multiple discriminant analysis, 511
- Multiple error, 316, 322, 344-345
- Multiple operational profile, 173, 237
- Multiple regression analysis, 512
- Multivariate analysis, 326, 776-779
  - software quality control, 514-518
- Musa basic execution time model, 87-90
- Musa's modification of Littlewood-Verrall model, 108-109
- Musa-Okumoto logarithmic Poisson model, 102-104
- Mutant, 533
- Mutation testing, 532-533, 554
- N* self-checking programming, 583-586, 627
  - fault tree model, 632-634
  - reliability model, 633
  - system reliability model, 650
  - system safety model, 650
- Nature:
  - prediction error, 136, 151
  - reliability process, 666-667
  - software faults, 385
- Near homogeneous run category, 194-195
- Necessary change, 363
- Need:
  - simulation, 663-664
  - software reliability, 3-5
- Network (see Neural network)
- Neural network, 699-718
  - application for software reliability, 709-710
  - architecture, 702-704, 706-707
  - cross-validation, 724
  - feed-forward, 703-704, 712
  - initialization, 708
  - mapping, 711
  - model analysis, 718
  - multilayer, 703
  - processing unit, 700-702
  - recurrent (see Recurrent network)
  - single-layer, 703
- Neural network (Co/If):
  - software reliability growth model, 711
  - software reliability growth modeling, 710-718
- New feature introduction, 245-246
- Next-release effect, 465, 478-479
- Next-step prediction, 714
  - (See also Short-term prediction)
- NHPP model (see Model: non-homogeneous Poisson)
- JV-modular-redundancy, 582
- Noise, 128, 144
- Nonhomogeneous Markov process, 759
- Nonhomogeneous Poisson process, 59, 74
- Nonhomogeneous Poisson process model (see Model: nonhomogeneous Poisson process)
- Nonparametric model, 699
- Nonstationary Markov process (see Non-homogeneous Markov process)
- Normal distribution, 750
- Normal mode trigger, 387
- Notion of prediction error, 145
- N*-step approximation to availability, 484-485
- N*-step-ahead prediction (see Long-term prediction)
- Null hypothesis, 410
- Number of remaining defects, 463
- Number of sites estimation, 475
- AT-version programming, 582-583, 627
  - fault tree model, 630-632
  - reliability model, 631
  - system reliability model, 648-650
  - system safety model, 648-650
- Objectives:
  - project application, 278, 281
  - quality, 257
  - reliability, 257-258
  - software reliability engineering, 219
- Occurrence probabilities:
  - functional profile, 181-183
  - operational profile, 189-194
- ODC (see Orthogonal defect classification)
- One-step-ahead prediction, 122
- On-line machine logging, 308-310
- Operating phase, normal, 754
- Operation, 15, 168-170, 184
  - critical, 175, 189
  - listing, 184-186
  - number reduction, 186-189
  - selection, 195-196



## 838 Index

Operation, selection (*Cont.*):

with an explicit operational profile,  
195 with an implicit operational profile,  
195

key input selection, 196 verification of,  
185-186 Operational architecture, 174, 184

Operational complexity, 509—510 Operational  
development, **prioritization**,  
199

Operational failures cost, 264-265

Operational phase:

evaluation, 306 reliability modeling, 335-339

Operational profile, 15, 183-194,

257-258, 534-535 accuracy,

171 advantage, 167-168

application:

**DEFINITY**, 200-204 **FASTAR**, 204-210

**PQRS**, 210-214 approximation, 168 as  
educational aid, 200 concepts, 168-170

determination, 236-237 development,  
206-208, 211-214 cost, 168, 171 procedure,  
170-194 documentation, 203—204

distributed systems, 198-199 elimination of  
steps, 172 estimation difficulty, 535-537

feature definition, 536-537 feature granularity,  
537 multiple and unknown user groups,

537 operational profile for new features,

536 operational profile for new software,

536 granularity, 171 introduction,

167-168 multiple, 173, 237 new software,

536 occurrence probabilities, 189-194 range

of operation, 168 reliability estimation,

537-539 coarse features, 538 inadequate test

set, 538 interacting features, 538 relationship

to design methodology, 173

Operational profile (*Cont.*):

selection of operation, 195-196

selection of test, 194-197

simple case, 168

updating, 197-198, 213

uses, 199-200

Operational-profile measurement program,

198 Operational-profile-based

testing,

202-203, 222 Operational

semantics trigger, 388 Operator log, 311

Opinion-based classification, 373 Optimal

release time, 82 Optimistic prediction,

122-123, 137, 151 Optimum network

architecture, 706-707 Optimum value of failure

intensity objective, 229-230 Oracle capability,

569-570, 573 Orthogonal defect classification:

collapsing of defect categories, 379

combination with growth modeling, 379-384

cost, 397

defect trigger (see Defect trigger attribute)

defect type (see Defect type attribute)

definition of, 367,

deployment, 396-398

design, 370-371

intuition, 367-370

necessary condition, 371-373

pilot, 396-397

principles, 367-374

production, 396-397

staged production, 396-397

sufficient conditions, 373-374 Orthogonality,

370 Outage, 12-13

duration, 13

**reportable**, 473

system, 13 Outlier,

779 Output layer, 703

Overall project results for reliability modeling,

292-295 Overview of book, 8-12

**P-use** (see Predicate use) Package

defect, 374 Parameter estimation,

766-772

**Jelinski-Moranda** model, 79

methodology, 635-639

- Parameter:  
 coverage, 337,647  
 failure intensity decay, 470  
 reliability profile, 669 Parametric model,  
 709 Partnering with customers, 444 Path  
 metric, 449 Pathology, fault, 28-29 **PBX** (see  
 Private branch exchange) People involved in  
 software reliability  
     engineering, 247-248 **Per-office**  
 failure rate, 475 **Perceptron**, 703, 721  
 Percolation process, 340 Perfective  
 maintenance, 32 **Performability** analysis, 35,  
 351-352  
     fault-tolerant software, 628 Performance:  
     absolute, 141  
     degradation by recording function, 198  
     load testing, 239  
     loss measure, 761  
     relative, 141  
     stress testing, 214 Personnel, ma intenance,  
 174 Pessimistic prediction, 122-123,137,  
 151  
 Phase-based model, 111-112 Physical  
 fault, 18 Plots, 450-454  
     cumulative failure, 452  
     failure intensity, 452-53 **PLR** (see  
**Prequential** likelihood ratio) Plurality decision,  
 600 Point estimation, 766-769 **Pointwise**  
 availability, 61 **Poisson** distribution, 749, 765  
**Poisson** process, 74, 765  
     homogeneous Poisson process, 74  
     mean, 765  
     **nonhomogeneous** Poisson process, 74  
     variance, 765  
 Poisson type model, 74-75, 77  
 Population, 762  
     moment, 768 **Postdelivery** phase of software  
 reliability  
     engineering, 242-246  
 Postprocessing method, 710 **Postrelease**  
 staff needs, 242-243 Predictability:  
     next-step, 714  
     variable-term, 714
- Prediction:  
 availability, 485-486  
 based on **strubtural** decomposition, 162  
 defects from Ada designs, 112-113  
 end-point, 711, 714  
 error:  
     nature, 136, 151  
     notion, 145 stationary,  
     140  
     fault model, 518-520  
     long-term, 123-126,159, 298-299, 456^57,  
 711  
     **next-step**, 714  
     noise, 128  
     one-step-ahead, 122  
     optimistic, 122-123,137,151  
     pessimistic, 122-123, 137,151  
     recursive comparison, 128-130  
     sequential, 711  
     short-term, 120-123, 711  
     software reliability, 17  
     techniques, limitation, 159—160  
     time to next failure, 447  
     training, 712  
     use, 182 Predictive accuracy, methods of  
 analyzing,  
     128-145 Predictive model and domain  
 coverage,  
     520-523  
 Preliminary analysis of data, 312-323  
 Preliminary selection of software reliability  
 models, 272-273 Premature convergence, 706  
 Preprocessing:  
     automatically generated error logs, 313  
     data,312-314  
     human-generated problem reports,  
     313-314 **Prequential**  
 likelihood, 281, 291  
     ratio,131-134  
     statistic, 734, 736, 738-739 Prevention of  
 fault, 19-20, 29 Preventive service, 329  
 Principal component analysis, 326,  
 495^99  
 Prioritized operational development, 199 Private  
 branch exchange, 171, 200 Probability:  
     conditional, 749-750  
     occupancy, 342  
     software failure, 628  
     system failure, 620  
     theory, 751

## 840 Index

- Probability (*Cont.*):
  - transition, 342
  - unacceptable result, 646
- Probability density function, 54, 748
  - conditional, 749-750
  - empirical, 773
  - joint, 767
- Probability integral transform, 135
- Problem, 362
- Problem/Failure Report (*P/FR*), 267
- Problem reports, preprocessing, 313-314
- Problem with root-cause phase analysis, 372-373
- Process:
  - data recording, 190
  - decision-making, 367
  - defect prevention, 397
  - iterative development model, 363
  - reliability (see Reliability process)
  - semantic extraction, 366
  - software reliability engineering, 257
  - spiral model, 667
  - usage prediction, 182
  - waterfall model, 363
- Process-based measurement, 372
- Process pairs, 328
- Process progress for product, 369
- Process simulation:
  - discrete-event, 665
  - single-event, 678
  - transfer, 396
- Processing time, temporal distribution, 505
- Processor halt data, 326
- Product and process improvement, 246
- Product development plan, 228
- Product requirements specification, 228
- Product, progress through process, 369
- Product-in-a-process*, 68
- Profile, 168
  - configuration, 202
  - environmental, 181
  - execution, 505
  - fault and error, 314
  - functional, 169, 176-183, 211-212
  - supersystem, 173
  - user, 201-202
- Program complexity, relative, 495-504
- Program evaluation and review technique, 291
- Program length, 496, 513
- Program module functionality, 505
- Program risk classification, 514
- Program size complexity, 496, 522
- Program subgraph, 509
- Program volume, 496
- Programming exception error, 316, 344
- Progress of product through process, 369
- Project:
  - CDS, 695-696
  - DEFINITY, 200-204
  - FASTAR, 204-210
  - MIS, 512-515, 719-720
  - PQRS, 210-214
  - Space Shuttle flight software, 476-479
- Project application objectives, 278, 281
- Project characteristics, 281-284
- Project characterization 278-279
- Project elements, *interdependencies*, 668-669
- Project investigation, software reliability, 278-299
- Project resource schedule, 667
- Projection:
  - field reliability, 259
  - staffing needs, 242-243
- Proliferation of environmental variables, 199
- Propagation characteristics of first error, 318-320
  - further corruption mode, 318-320
  - no propagation mode, 318-320
  - quick detection mode, 318-320
- Pure birth process, 765
- Qualitative causal analysis, 364-366
- Quality and performance tests, 427
- Quality aspect in discriminant model, 516
- Quality attribute, 221
- Quality factor assessment, 222
- Quality function deployment, 227
- Quality improvement, 222 results, 167-168
- Quality measurement from customer perspective 221
- Quality objectives, key, 257
- Quality of measurement plan, 471
- Quality versus reliability 221
- Quantitative analysis of fault-tolerant software, 635-645
- Quantitative reliability requirements, 256
- Quantitative requirements, 256
- Quantitative statistical defect model, 364-366
- $R^2$  statistic, 458, 519
- Random sample, 766

- Random selection of run types, 194
- Random variable:
  - continuous, 748-749
  - definition of, 748
  - discrete, 748-749
  - exponential, 140
- Range of operations in operational profile, 168
- Rapid prototyping, 667
- Rare events handling, 553-554
- Rare situation trigger, 388
- Rare-event model, 473-476
- Rate function of software reliability
  - model, 686-687
  - Rate of occurrence of failures, 14, 55, 128
- Rate-based simulation:
  - event simulation, 665
  - reliability simulation, 686-690
  - reliability simulator architecture, 687-689
  - simulation algorithms, 676-686
- Ratio, **prequential** likelihood, 131-134
- Rayleigh** density function, 755
- Rayleigh** distribution function, 94, 755
- Realization (see Sample function)
- Recalibrated median, 147
- Recalibration** of data with respect to usage, 454
- Recalibration of software reliability model, 145-149
  - power of, 147-149
  - technique, 146-147
- Record of error, 308-309
- Record of intent, 363
- Recording function, performance degradation, 198
- Recovery:
  - error, 340
  - hardware, 350
  - software-controlled, 350
- Recovery blocks, 581-582
- Recovery fault tree model, 629-630
- Recovery distribution, 320-323
- Recovery model, 337-344
- Recovery point, 570
- Recovery rate:
  - average, 481
  - instantaneous, 480
  - software, 482
- Recovery routine, 328
- Recovery strategy, 232-233
- Recovery trigger, 387
- Recurrence, 329-333
  - automatic diagnosis, 331-333
  - fault, 315
  - minimizing, 329-331
  - time, 342
- Recurrent event simulation, 681-682
- Recurrent event statistics, 679-681
- Recurrent network, 703
  - fully recurrent network, 704
  - semirecurrent** network, 704
  - simple recurrent network, 704
- Recursive comparison of prediction, 128-130
- Rediscovery (see Recurrence)
- Rediscovery matrix, 329-330
- Rediscovery of failure, 53
- Reduced-operation software, 186-187
- Reducing interaction between successive runs, 187
- Reducing number of input variables, 186
- Reducing number of operations, 186-189
- Reducing number of run types, 186
- Reducing unnecessary interaction, 187
- Reduction:
  - cost, 168
  - defect, 360
  - time, 168, 360
- Redundant software elements, 233
- Regression coefficient, 347
- Regression faults, 403, 423
- Regression model selection, 518-520
- Regression test, 196-197, 238
- Regression:
  - eyeball, 694
  - stepwise**, 519
- Related faults, 594, 596, 628, 636
- Related types of testing, 238
- Relative complexity metric, 501
- relationship to execution profile, 506-507
- Relative error, 128, 458
  - long-term prediction, 456-457
- Relative performance, 141
- Relative program complexity, 495-504
- Relative subgraph complexity, 509
- Release criterion for reliability, 222
- Release date approach for reliability specification, 262-263
- Release time, 82
- Reliability allocation among components, 232

## 842 Index

### Reliability:

advice to users, 162—163  
as release criterion, 222  
best estimate of, 259  
boundary version, 592  
definition of, 29, 35  
majority voting, 592-593  
module, 589  
inpresenceof **interversion** failure correlation, 596-604  
stable, 50, 57-58, 409, 423-424  
system, 36-37, 39<sup>0</sup>  
trade-off with functionality, 229  
**ultrahigh**, 159, 472  
versus quality, 221 Reliability analysis:  
calendar-time, 461, 463-469  
required fields, 443  
usage-based, 461, 469-472 Reliability  
assessment in artifact simulation, 674-675  
Reliability decrease, 50 Reliability  
demonstration:  
chart, 236  
testing, 235  
Reliability discontinuity, 427 Reliability effect  
on architecture, 231 Reliability estimates,  
variation, 550-551 Reliability estimation with  
inaccurate operational profiles, 537-539  
coarse features, 538  
inadequate test set, 538  
interacting features, 538 Reliability  
estimation incorporating coverage, 546-547  
Reliability evaluation, 64—68 Reliability  
fluctuation, 433 Reliability function, definition  
of, 752 Reliability growth, 50, 404, 423  
definition of, 403  
fault-tolerantsoftware, 65  
monotone, 404  
strict, 404  
Reliability growth assumption, 450  
Reliability growth curve:  
inflection point, 378  
knee, 378  
subjective aspects, 377-379 Reliability  
growth factor, 286 Reliability growth in field,  
475-476 Reliability growth modeling, 709-710  
linear relationships, 459 Reliability growth  
testing in system test stage, 237-240

Reliability growth modeling, combination with  
orthogonal defect classification,  
379-384

Reliability measurement for evolving software  
(see Evolving software) Reliability  
model:

acceptance voting, 603-604 comparison with  
simulation, 694-696 consensus recovery block,  
601—603 consensus voting, 592-593, 599-601  
**N** self-checking programming, 633 **A**-version  
programming, 631 distributed recovery block,  
630 fault-tolerant software, 585-596 trend  
analysis, 424 Reliability modeling, 53-60,  
257-258 conditional reliability, 55, 57—58 data  
domain, 588—594 **TV**-version programming,  
591-593 comparison, 593-594 recovery block,  
590-591 interval reliability, 55, 58 operational  
phase, 335-339 time domain, 588, 594-596  
reliability decrease, 50 reliability growth, 50  
results, 291-292 stable reliability, 50, 57-58  
tools, 273 with software complexity metrics,  
524-526

Reliability objectives, 230-231, 257-258  
achieving, 232-233 availability consideration,  
231 certification, 240-241 **influen**cial factors,  
230-231 reliability effect on architecture, 231  
Reliability **overestimation** due to saturation,  
545-546

Reliability prediction, 81-82, 713-718 definition  
of, 17 **Duane** model, 99 experiment, 713-718  
geometric model, 101 **hyperexponential** model,  
91 **Jelinski-Moranda** model, 79

**Littlewood-Verrall** model, 107-108 **Musa** basic  
execution time model,

89-90 **Musa-Okumoto** logarithmic

**Poisson**

model, 103 **nonhomogeneous**

**Poisson** process

model, 81-82 **Schneidewind** model,  
84-86

- Reliability prediction (*Cont.*):
  - S-shaped** model, 97-98
  - Weibull** model, 95
- Reliability process, 661, 665-669
  - definition of, 661
  - end-to-end, 691
  - fault identification, 667
  - fault repair, 667
  - nature, 666-667
  - primary event, 689
  - repair validation, 667
  - retest**, 667
  - schedule items, 668
  - secondary event, 689-690
  - simulation for project application, 690-696
  - software construction, 666
  - software correction, 666
  - software inspection, 666
  - software integration, 666
  - structures and flows, 667-668
  - test preparation, 667
  - testing, 667
- Reliability profile parameters, 669
- Reliability requirements specification, 256, 259-266
  - definition of, 256
  - life-cycle cost optimization, 263-266
  - product, 228
  - release date approach, 262-263
  - system balance method, 261-262
  - testable requirements, 260
  - untestable** requirements, 261
- Reliability simulation, 662-665, 686-690
  - rate-based, 686-690
  - simulator architecture, 687-689
- Reliability theory, 751-758
- Reliability trend test (see Trend test)
- Reliability validation, 257, 259
- Remaining set of run types, exclusion, 188-189
- Removal of
  - fault, 19-20, 29
  - Renewal process, 56
  - Renewal theory, 54
  - Repair rate, field, 482
  - Repetitions of failures, 244
  - Reportable** events, 473, 483
  - outages, 473
- Reporting, manual, 308, 310-312
- Request for proposal, 226
- Requirement of dependability, 65-66
- Residual risk, 561
- Residuals of fit, 455
- Residue, 738
- Resource, usage function, 346
- Resource-usage model, 349-351
- Restart and checkpointing, 328
- Restoration, 34, 49-50
- Result verification, 570-574
- Results:
  - application, 285-289
  - artifact simulation, 675-676
  - neural network for error-prone module identification, 723-725
  - neural network for reliability prediction, 715-718
  - overall project for reliability modeling, 292-295
  - quality-improvement, 167-168
  - reliability modeling, 291-292
  - simulation experiments, 691-694
- Reuse, software, 20, 234
- Review and inspection, 234
- Review and inspection trigger, 387-388
- Review of discovery, 231
- Reward functions, 334, 351
- Reward, cumulative, 352
- Reward rate, 351-352
  - instantaneous, 760
  - interval, 760
  - steady-state, 335, 760-762
- Reward-loss rate, steady-state, 335
- Risk, 472
  - residual, 561
  - testing based** model, 554-555
  - testing based** notation, 555
  - untested code, 554
- Risk assessment, 376-384, 472-473, 555-558
  - using defect types, 376-384
- Risk browser, 560-561
- Risk classification, 514-515
- Risk computation example, 558-559
- Risk confidence growth model, 562
- Risk model, 472, 554-563
  - software reliability, 561-563
- Risk prediction result, 715-718
- ROCOF** (see Rate of occurrence of failures)
- Rome Laboratory model, 113-114
- Root-cause analysis cost, 372-373, 397
- r-squared** statistic, 458, 519
- Run, 169, 185
- Run category, 194
  - near homogeneous, 194-195
- Run-time overhead, 563
- Run type, 169
  - group, 169

## 844 Index

### Run type(Cont.):

homogeneous, 194 number  
reduction, 186

Safety, 29, 35

Safety analysis, 624

Safety validation using fault trees,  
623-627

Sample distribution function, 135 Sample error  
entry, 308-309 Sample function, 677 Sample  
moment, 768 Sample space, 747 Sampling users,  
190 Saturation effect, 541-542, 562 empirical  
basis, 543-544 reliability **overestimation**,  
545-546 Scaled relative complexity, 501  
**Schick-Wolverton** model, 94 **Schneidewind**  
model (see Model:

**Schneidewind**)

Scope of book, 21-23

Secondary event simulation, 682-683 Security,  
29

**SEI** software process, 477

Selection:

operation, 195-196 regression model,  
518-520 with replacement, 195 without  
replacement, 195 Self-assessment in software  
reliability

engineering, 250-252 Self-checking  
capability, 569-570 Self-checking software  
component, 633 Self-driven simulation, 664,  
672-673 Semantic classification, 373 Semantic  
extraction of process, 366, 372,  
375

Semantic information, 368 Semantics,  
connection with measurement,  
372

Sensitivity analysis, 338-339 Sequential  
prediction, 131, 711 Serialization defect,  
374 Service, 12, 34 corrective, 329  
preventive, 329 Service utilization matrix,  
48 Setting up data collection process,

266-267 Severity assignment, 244

Severity classification (see Severity classi-  
fication of failure)

Severity classification of failure, 12,  
227-228, 477-478 Severity of failures,  
477 Short-term prediction, 120-123, 711, 714

Side effect of code update, 314-315 **Sigmoidal**  
unit, 702 Significance level, 775 Significant  
environmental conditions,  
174

Similar faults, 576, 594, 596 Simple operational  
profile, 168 Simple path coverage trigger, 389  
Simple short-term prediction, 120-123

Simulated code generation, 672-674 Simulated  
defect status profile, 691-693 Simulated fault  
status profile, 691-693 Simulation, 664

artifact-based, 665, 669-676

compared with reliability model, 694-696

discrete-event, 665

event process, 676

limited growth, 683

rate-based, 665

recurrent event, 681-682

reliability, 662-665

secondary event, 682-683

self-driven, 664, 672-673

trace-driven, 664 Simulation algorithms,

rate-based,

676-686

Simulation approaches, dynamic, 664-665

Simulation experiments and results, 691-694

display of results, 689-690 Simulation model,

206, 662 Simulation program structure for  
single

event occurrence, 678-679

Simulation time choice, 678-679 Simulator

architecture:

artifact-based, 670-675

rate-based, 687-689 Simulator program,

685 Single-event occurrence simulation,

678-679

Single-interpreters system, 43<sup>17</sup> Single-layer  
network, 703 Single point of failure, 619

Smallest divisions of work, 169 Smoothing

of software behavior, 409 Smoothing

technique, 161 Soak test, 197 Soak time, 197



- Software:
- data recording, 190
  - middleware, 390
  - reduced-operation, 186-187
  - Software artifact, 663
  - Software behavior, smoothing, 409
  - Software complexity metrics for reliability modeling, 524-526
  - Software complexity and software quality, 510-523
  - Software configuration trigger, 387
  - Software defects, 361-364
  - Software diversity, 567
  - Software elements, redundant, 233
  - Software environment characteristics, 669
  - Software error model, 341
  - Software evolution (see Evolving software)
  - Software failure correlation, 324-325, 598-599
  - Software failure data collection and analysis, 440
  - Software failure mode, 754
  - Software failure:
    - hardware-related, 327-328
    - load-induced, 348
    - probability, 628
    - with workload parameters, 345-352
    - (See *also* Failure)
    - Software fault:
      - model, 318
      - nature, 385
    - Software fault tolerance (see Fault-tolerant software)
    - Software field data, 440
    - Software life cycle, 306
    - Software metrics, 495-496
    - Software process, SEI, 477
    - Software quality:
      - models, 511
      - multivariate analysis, 514-518
      - relationship to software complexity, 510-523
      - relationship to software reliability, 6
    - Software recovery, 350
      - model, 341
      - recovery rate, 482
    - Software reliability:
      - definition of, 5, 540
      - early prediction, 17
      - expression, 259-261
      - factors affecting, 339
      - Software reliability (Cont.):
        - need for, 3-5
        - prediction, 17
        - relationship to software quality, 6
        - Software reliability and neural networks, 709-718
        - Software reliability and risk model, 561-563
        - Software reliability estimation, 17
          - with time and structure, 539—554
      - Software reliability engineering:
        - activities, 223-225
        - approaches, 220-226
        - benefits, 210, 220-226
        - cost, 222-223
          - operational profile cost, 222
          - processing and analyzing failure data cost, 223
        - software reliability engineering consultant cost, 223
        - definition of, 8
        - design and implementation phase, 232-235
        - design stage, 232-233
        - feasibility and requirement phase, 226-231
        - feasibility stage, 226-228
        - field trial stage, 241-242
        - getting started, 246-252
        - implementation on existing projects, 224
        - implementation on short-cycle projects, 226
        - implementation stage, 233-235
        - incremental implementation, 223, 225
        - information and support, 250
        - involved people, 247-248
        - objective, 219
        - organization preparation, 247-250
        - postdelivery and maintenance phase, 242-246
        - practice, 219-220
        - process, 247, 257
        - requirement stage, 228-231
        - self-assessment, 250-252
        - success story, 221-222
        - system test and field trial phase, 235-242
        - system test stage, 237-241
        - technical areas, 19-22
        - fault prevention, 19-20, 29
        - fault removal, 19-20, 29
        - fault tolerance, 19-20, 29
        - fault failure forecasting, 19, 21, 29

## 846 Index

- Software reliability measurement, 17 early prediction, 17 estimation, 17 framework, 256-278 prediction, 17 reliability prediction, 17 Software reliability modeling tool, 449 (See *a/so* Software reliability tool) Software reliability model, 17-18 preliminary choice, 272 preliminary selection, 272—273 rate function, 686-687 relationship to hardware reliability, 18-19 tool, 449 (See *also* Software reliability tool) Software reliability prediction, 17 Software Reliability Program (**SoRel**) tool, 737-739 Software reliability project investigation, 278-299 Software reliability requirement, 67 Software reliability requirements, establishment, 259-266 Software Reliability Simulator (**SoftRel**) tool, 688 Software reliability study goals, 441-442 Software reliability testing, 68 Software reliability tool:  
AT&T toolkit (see AT&T software reliability engineering toolkit) **CASRE** (see Computer-Aided Software Reliability Estimation tool) **ESTM** (see Economic Stop Testing Modeltool) **SMERFS** (see Statistical Modeling and Estimation of Reliability Functions for Software tool) **SoftRel** (see Software Reliability Simulator tool) **SoRel** (see Software Reliability Program tool) **SRMP** (see Statistical Modeling and Reliability Program tool) tool comparisons, 741-744 Software reuse, 20, 234 Software risk (see Risk) Software science metrics (see **Halstead** number) Software system, 12 Software testing, 20 Software-specific classifiers, 442-443 Source of failure, 65-66 Space complexity, 681 Space Shuttle flight software, 476-479 Spawned faults, 196 Spearman test, 411, 737 Special events, 472-479 Specification: product requirements, 228 reliability requirements, 261-266 (See *a/so* Requirements specification) Spectrum of defect analysis, 364-367 Spiral model, 667 Squared distance, generalized, 515 **SRE** (see Software reliability engineering) **S-shaped** model (see Model: **S-shaped** model) Stability index, 231 Stability test program execution, 209-210 Stable reliability, 50, 57-58, 409, 423-424 Stack trace, 332 Standard assumptions of software reliability models, 78 Standardization of data collection and analysis process, 440 Standardized transformation matrix, 498-499, 514-515 State space, 751 State transition equation, 759 State-transition-rate diagram, 764 Statement coverage testing, 533 State of error, 340 Static analysis and trend analysis, 431<sup>^</sup>34 Static analysis conduct, 431-433 Static measure, 502 Stationary prediction error, 140 Stationary process, 56 Statistic: **Braun**, 130 **chi-square** (see **Chi-square** statistic)  $X^2$  (see **Chi-square** statistic)  $C$ , 520 **F-statistics**, 717 **prequential** likelihood (see **Prequential** likelihood: statistic)  $R^2$  (**r-squared**), 458, 519 Statistical analysis and visualization system, 449 Statistical cluster analysis, 348 Statistical defect model, 364-366 Statistical Modeling and Estimation of Reliability Functions for Software (**SMERFS**) tool, 733-735 Statistical Modeling and Reliability Program (**SRMP**) tool, 736

- Steady-state availability, 483-484 Steady-state reward rate, 335, 760-762 **Stepwise** regression, 519 Stochastic independence assumption, 403, 580, 588 Stochastic process, 750-751 doubly, 346 Stop testing criteria, 467, 740 Stopping rule in selecting principal components, 498 Storage exception error, 316, 344 Storage management error, 316 Stress test program execution, 209—210 Strict reliability growth, 404 Structural decomposition for prediction, 162 Structure-based software reliability estimation, 539-554 Structures and flows in reliability process, 667-668 Student's ( $t$ -distribution, 770 **Subadditive** property, 404, 407-408 analysis, 406-407 graphical interpretation, 404-406 relationship to Laplace test, 414—419 summary, 409—410 **Subadditivity**, 404 **Subadditivity** factor, 406 constant, 408-409 for validation phase, 417, 419 Subgraph of program, 509 Subjective aspects of growth curves, 377-379 **Subpopulation** identifier, 395 **Subprocess** representation, 689 Success story, software reliability engineering, 221-222 Sum of squared errors, 706, 769 **Superadditive** property, 404 (See also Subadditive property) **Superadditivity**, 404 (See also Subadditivity) **Supersystem** profile, 173 Survival function, 54 Symptom, difference with trigger, 386 Symptom-based diagnosis, 332 System, 174 command-driven, 191 data-driven, 193 System activity impact, 345-352 System administrator, 174 System availability, 483 System balance method for reliability specification, 261—262 System complexity, 500 System dependability, relationship to software reliability, 7 System dynamics, 665 System failure rate, 49 System failure, probability, 620 System for manipulation, filtering, and data merging, 449 System for statistical analysis and visualization, 449 System interpreters, 43 System-level analysis: case study, 651—657 hardware and software system, 645—657 System-level software reliability: model, 336, 338 parameters, 336 System life, 756 System made up of components, 41-49 System mode, 174-176 System model, 41-43 System outage, 13 causes, 439-440 System reliability, 36-37, 39-40 continuous-time, 36-37, 40 discrete-time, 36, 39-40 System reliability model:  $N$  self-checking programming, 650  $\wedge$ -version programming, 648-650 distributed recovery block, 647-648 System safety model:  $N$  self-checking programming, 650  $\wedge$ -version programming, 648-650 distributed recovery block, 647-648 System start-up and restart trigger, 387 System structure model, 646 System test phase of software reliability engineering, 235-242 reliability growth modeling, 241-242 reliability growth testing, 237-240 test data collection, 239 System test triggers, 387 System unavailability, 483  $\wedge$ -distribution, 770 Tandem project report (**TPR**), 311 Task computation, 628 Task, feature-oriented, 175 Technical areas for software reliability engineering, 19-22 fault prevention, 19-20, 29 fault removal, 19-20, 29 fault tolerance, 19-20, 29 fault/failure forecasting, 19, 21, 29

## 848 Index

### Technique:

- combination, 161-162
- data analysis, 450
- discrimination, 515
- historical validation, 456
- smoothing, 161 Technology transfer, 396

Temperature coefficient, 702 Temporal distribution of processing time, 505

Temporary behavior of software, 409 Test:

- arithmetic mean, 411—412
- chi-square**, 714, 774-775
- Gnedenko** (see **Gnedenko** test)
- Hollander** (see **Hollander** test)
- Kendall** (see **Kendall** test)
- Kolmogorov-Smirnov**, 714, 775-776

Laplace (see Laplace test)

Spearman (see Spearman test) Test

automation, 238 Test compression factor,

259 Test coverage trigger, 389 Test efficiency, 422

Test environment development, 208 Test

interaction trigger, 389 Test selection, 194-197

- coarse grain selection, 194
- fine grain selection, 194
- random selection of run types, 194
- selection with replacement, 195
- selection without replacement, 195 Test

sequencing trigger, 389 Test set:

- adequacy criterion, 534
- inadequate, 538 Test variation trigger, 389

Testable reliability requirement, 260 Testing:

- accelerated, 175
- acceptance, 570-571, 581
- back-to-back, 606
- control flow, 532-533
- customer-like, 202-203
- data flow coverage, 533
- data flow, 532-533
- decision coverage, 533
- domain**, 554
- functional, 268-270, 427, 534
- increased, 175
- kinds of, 532
- mutation, 532-533, 554
- operational-profile-based**, 202-203, 222

### Testing (*Cont.*):

- performance and stress, 214
- progress control, 422
- quality and performance, 427
- related types, 238
- reliability demonstration, 235
- software, 20
- software reliability, 68
- statement coverage, 533
- varying occurrence probabilities, 214
- white-box, 532

Testing and operation trade-off, 265

**Testing-based** notation of risk, 555 Testing cost,

265 Testing effort, 540-541, 547 Testing

hypothesis, 410<sup>11</sup>, 450, 774 Testing methods, limits, 542—543 Testing overview, 532-534

Testing progress tracking, 240 Testing tool,

213-214 Testing with replacement, 277 Testing

without replacement, 277 Three-point

symmetric moving average, 465-466

Threshold function:

- hard-limiting, 701
- linear, 701 Time, 13-14, 444-445
- calendar time, 14
- clock time, 14
- execution time, 14
- in-service, 445, 460
- logical, 445
- translation, 14
- usage, 445

Time choice, simulation, 678-679 Time

complexity, 679, 681, 683 Time domain

approach, 73 Time measurement, 239-240

Time reduction, 168 Time to error, 320-322

Time to failure, 320, 447 Time to recovery,

322-323 Time to repair, 320 Time units, 419

**Time-between-failures** data, 16, 268, 271

**Time-between-failures** models vs. failure-count

models, 288-289 Time-domain analysis (see Time-domain

reliability modeling) Time-domain reliability modeling, 588,

594-596

Timing defect, 374

- Tolerance, 578-581 Tolerance of fault, 19-20, 29 Tool:
- commercial, 730
  - defect-tracking, 394
  - general-purpose, 730
  - reliability modeling, 273
  - testing, 213-214 Tool comparison, 703
  - software reliability tool, 741—744 Tool for software reliability modeling, 449
- Tool selection criteria, 731-732 Total hazard, 677
- Trace-driven simulation, 664, 674
- Trachtenberg's general theory, 464 Tracking customer satisfaction, 245 Tracking testing progress, 240 Trade-off between testing and operation, 265 Trade-offs studies, 229-230
- reliability and functionality, 229 Traffic condition, 175 Training data selection, 723
  - Training output layer, 708 Training regime, 712
  - Training set, 705, 722 Trajectory of input, 38-39, 402
- (See a(so Failure: domain) Transfer, process, 396 Transfer, technology, 396 Transformation of data (see Data transformation)
- Transformation of reliability-related functions, 753
- Transformation between data types, 17
- Transient behavior of software, 409 Transient hardware faults, 628 Transient region, 482, 484
- Transition probability, 342 Translation, time, 14
- Tree-based approach, 202 Trend analysis:
- definition of, 410
  - reliability models, 424
  - static analysis, 431-434
  - trouble reports, 433-434
- Trend change, 407-408 Trend test, 410-419, 737
- analytical, 410
  - graphical, 410
- Trial-run for data collection, 267
- Trigger (see Defect: trigger)
- Trouble reports, 6, 244, 431
- trend analysis, 433-434
  - Trouble tracking system, 244
- Two-class risk classification model, 515 Two-out-of-A<sup>n</sup> voting, 577 Type of classification error, 516 Type of connectivity, 703
- u-plot, 135-140, 281, 734, 736, 739 Ultrahigh reliability (see Ultrareliability) Ultrareliability, 159, 472
- modeling, 277-278
  - operation, 237
- Unacceptable result probability, 646
- Unavailability, 62-64, 483
- empirical, 481-482
  - uptime, 485
- Understanding customer expectations, 206 Unequally weighted linear combination model, 291 Unique factor, 778 Unique variance, 778 Units of time, 419
- Unnecessary interaction, reduction of, 187
- Unrelated fault (see Independence: fault)
- Unreliability calculation, 621 Untestable reliability requirement, 261 Updating operational profile, 197—198, 213
- Uptime availability, 481, 483 Uptime unavailability, 485 Usage, 445-446
- fields, 446
  - function of resource, 346
  - metrics, 499-500
  - profile of users, 463
  - time, 445
- Usage-based reliability analysis, 461, 469-472
- User, 173, 175
- experience, 175
  - generic, 201
  - profile, 201-202
  - sampling, 190
  - telecommunications, 174
  - type, 173-174
  - usage profile, 463
- Utilization environment, variation, 403
- V&V (see Verification and validation)
- Validating data set, 513 Validating new measure, 521

## 850 Index

- Validation phase, **subadditivity** factor, 417, 419
- variability, 128, 281
  - field reliability data, 439
  - functional complexity, 509
- Variable-term predictability, 714
- Variables, environmental, 175—176, 180-181, 190, 192
- Variance, 748
  - analysis of, 716
  - Poisson** process, 765
- Variants (see Alternates)
- Variation:
  - between-cluster**, 779
  - within-cluster, 779
  - model accuracy, 126
  - reliability estimates, 550—551
  - utilization environment, 403
- Varying occurrence probabilities for testing, 214
- Verification:
  - operation, 185-186
  - result, 570-574
- Verification and validation, 33—34
- Version, 52
- Version control, 235
- Versions (see Alternates)
- Voter, 577, 582
- Voting, 577
  - back-to-back, 582
  - consensus, 578
  - majority, 577
  - mean value, 578
  - median, 577, 583
  - two-out-of-TV**, 577
- Walk-through, 432
- Waterfall process, 363
- Wear-out failure, 754
- Weibull** and gamma failure time class
  - model, 93-98
  - Rayleigh** (see Model: **Rayleigh**)
  - Schick-Wolverton** (see Model: **Schick-Wolverton**)
  - S-shaped** (see Model: **S-shaped**)
  - Weibull** (see Model: **Weibull**)
  - Weibull-type** field-defect model, 464-465
- Weight determination based on ranking
  - model results, 290-291
- White-box testing, 532
- Work, smallest division, 169
- Workload:
  - CPU-bound, 349-350
  - I/O, 349
  - Workload-dependent **cyclostationary** model, 346
  - Workload model, 348-349
  - Workload parameters and software failures, 345-352
  - Workload stress trigger, 387
  - Workload volume trigger, 387
  - Weight adjustment phase, 705
  - Weight determination based on changes, 291
- X-ware**, 27, 35-49
  - atomic systems, 35—41
  - system made up of components, 41-49
    - multi-interpretor system, 47-49
    - single-interpretor system, 43-47
  - system models, 41<sup>13</sup>
- Yamada** modified exponential growth
  - model, 91
  - 3'-plot, 140-141, 281, 734, 736, 739

## ABOUT THE EDITOR

Michael R. Lyu is currently a Member of the Technical Staff at AT&T Bell Laboratories, Murray Hill, focusing on software reliability engineering, software fault tolerance, and distributed systems. He has published fifty papers in these fields. He is also the editor of *Software Fault Tolerance* (Wiley, 1995). Dr. Lyu received his B.S. in Electrical Engineering from National Taiwan University in 1981, M.S. in Electrical and Computer Engineering from University of California, Santa Barbara, in 1984, and Ph.D. in Computer Science from UCLA in 1988. He previously worked at JPL, Bellcore, and taught at the University of Iowa.