

REFERENCES

- [1] Anthony, L. Alwardt, Richard J. Pandorf & Philip R. Tarpley (2009). A Lean Approach to Designing for Software Testability. *IEEE International Conference on Software Engineering*, 978-1-4244-4981-1/09/IEEE, 19-28.
- [2] Amid D., Amin, & Somaye M. (2013). A Hybrid Evaluation Framework of CMM and COBIT for Improving the Software Development Quality. *Journal of Software Engineering and Applications*, 6, 280-288.
- [3] Azam, F., Qadri, S., Ahmad, S., Khan, K., Siddique, A. B. & Ehsan, B. (2014). Framework of Software Cost Estimation By Using Object Orientated Design Approach. *International Journal of Scientific & Technology Research*, 3 (8), 97- 100.
- [4] Amnon H. Eden & Tom Mens (2006). Measuring Software Flexibility. *IEE Software*. London, UK: The Institution of Engineering and Technology, 153(3), 113–126.
- [5] Arora, D., Khanna, P., Tripathi, A., Sharma, S., & Shukla, S. (2011). Software quality estimation through object oriented design metrics. *International Journal of Computer Science and Network Security*, 11(4), 100-104.
- [6] Bruce, W. Weide, Stephen H. Edwards, Wayne D. Heym, Timothy J. Long & William F. Ogden (1995). Characterizing Observability and Controllability of Software Components. *Technical Report OSU-CISRC-9/95-TR37*. The Ohio State University, Columbus, OH 43210, 1995, 1- 20.
- [7] Bashir, I., & Goel, A. L. (2012). *Testing object oriented software: life cycle Solutions*. Springer Science & Business Media.
- [8] Bruce, W.N.Lo & Haifeng S. (1998). A preliminary testability model for object-oriented software. *In Proceedings International Conference on Software Engineering, Education, Practice*. IEEE, 330-337.

- [9] Birolini, A. (2014). Design Guidelines for Reliability, Maintainability, and Software Quality. In *Reliability Engineering*, Springer Berlin Heidelberg, 144-168.
- [10] Bruntink, M. & Deursen, A.V. (2004). Predicting class testability using object-oriented metrics. In *Proceedings of IEEE International Workshop on Source Code Analysis and Manipulation*, 136-145.
- [11] Badri, L., Badri, M., & Toure, F. (2010). Exploring empirically the relationship between lack of cohesion and testability in object-oriented systems. In *Advances in Software Engineering*, Springer Berlin Heidelberg, 78-92.
- [12] Baudry, B. (2001). Towards a safe use of design patterns for object oriented software testability. In *proceedings of ISSRE 01*, Hong-Kong, 324-329.
- [13] Binder, R.V. (1994). Design for testability in object oriented systems. *Communications of the ACM*, 37(9), 87-101.
- [14] Badri, M. & Toure, F. (2012). Empirical Analysis of Object-Oriented Design Metrics for Predicting Unit Testing Effort of Classes. *Journal of Software Engineering and Applications*, 5, 513-526.
- [15] Baudry, B., Traon Le, Y. & Sunye, G. (2002). Testability Analysis of a UML Class diagram. *Proceedings of the Eighth IEEE Symposium on Software Metrics [METRICS.02]*, IEEE, 187-194.
- [16] Bruntink, M. & Van Deursen, A. (2004). Predicting Class Testability Using Object-Oriented Metrics. In *Proceedings of the Fourth IEEE International Workshop on Source Code Analysis and Manipulation*, Chicago, 136-145.
- [17] Behshid, B., Kahani, M. & Akbari, M. K. (2009). Customizing ISO 9126 quality model for evaluation of B2B applications. *Information and software technology* 51(3), 599-609.

- [18] Bousquet, Lydie du, Delaunay Michel, Huy-Vu Do & Robach Chantal, (2010). Analysis of Testability Metrics for Lustre/Scade Programs. *Second International Conference on Advances in System Testing and Validation Lifecycle*, 26-31.
- [19] Baudry & Traon (2005). Measuring Design Testability of a UML Class Diagram. *Information and Software Technology*, 47(13), 859–879.
- [20] Chidamber, S. R. & Kemerer, C. F. (1994). A Metrics Suite for Object Oriented Design. *IEEE Transactions on Software Engineering*, 20, 2781-2789.
- [21] Cinneide, M.O., Boyle, D. & Moghadam, I.H. (2011). Automated Refactoring for Testability. *Software Testing, Verification and Validation Workshops (ICSTW), IEEE Fourth International Conference*, Ireland, 437 – 443.
- [22] Chauhan, R., Singh, R., Saraswat, A., Joya, A. H. & Gunjan, V. K., (2014). Estimation of Software size using Object Oriented Design Metrics. *International Journal of Innovative Research in Computer and Communication Engineering*, 2 (1), 2581-2586.
- [23] Dino Esposito (2008). Design Your Classes for Testability. Available at: <http://dotnetslackers.com/articles/nnet/Design-Your-Classes-for-Testability.aspx>.
- [24] Dromey, R.G. (1995). A Model for Software Product Quality. *IEEE Transaction on Software Engineering*, 21(2), 146–162.
- [25] Elish, K. O., & Alshayeb, M. (2011). A classification of refactoring methods based on software quality attributes. *Arabian Journal for Science and Engineering*, 36(7), 1253-1267.
- [26] Felderer, M. & Schieferdecker, I. (2014). A taxonomy of risk based testing. *International Journal on Software Tools for Technology Transfer*, 16(5), 559- 568.

- [27] Fu, Jianping & Minyan Lu (2009). Request Oriented Method of Software Testability Measurement. *Information Technology and Computer Science, ITCS 2009. International Conference on IEEE*, 2, 675- 682.
- [28] Goel Noopur & Gupta Manjari (2012). Testability Estimation of Framework Based Applications. *Journal of Software Engineering and Applications*, 5(11), 54-59.
- [29] Gupta, N. & Goyal, D. (2015). Modelling Software Quality with Quality Parameters for Object Oriented Design. *International Journal of Software Engineering Research and Practices*, 1(1), 1- 4.
- [30] Gupta, S. P. & M. P. Gupta (2014). *Statistical Methods, 43rd Edition*. Sultan Chand & Sons.
- [31] Gross, H. G. (2001). A prediction system for evolutionary testability applied to dynamic execution time analysis. *Information and Software Technology*, 43, 855-862.
- [32] Gao J. & Shih, M. C. (2005). Component testability model for verification and measurement. *In Proceedings of the 29th Annual International Computer Software and Applications Conference*, IEEE Computer Society, 211- 218.
- [33] Gonzalez, A., Piel, E. & Gross, H. G. (2009). A Model for the Measurement of the Runtime Testability of Component Based Systems. *IEEE International Conference on Software Testing Verification and Validation Workshops*, 19-28.
- [34] Genero, M., Olivas, J., Piattini, M. & Romero, F. (2001). Empirical validation of measures for class diagram structural complexity through controlled experiments. *In Proceedings of the ADIS2001, Workshop on Decision Support in Software Engineering*, Spain, 84.
- [35] Ghosh, S. (2002). Towards measurement of testability of concurrent object-oriented programs using fault insertion: a preliminary investigation. *In Proceedings Second IEEE International Workshop on Source Code Analysis and Manipulation*, 1 Oct. 2002, Los Alamitos, CA, USA, 17-25.

- [36] Gupta V., Aggarwal, K. K. & Y. Singh (2005). Fuzzy Approach for Integrated Measure of Object Oriented Software Testability. *Journal of Computer Science*, 1(2), 276-282.
- [37] Huang, R., Li, M., & Li, Z. (2013). Research of Improving the Quality of the Object-Oriented System. *International Journal of Information and Education Technology*, 3(4), 433.
- [38] ISO/IEC 25010 (2011). Systems and Software Engineering Systems and Software Quality Requirements and Evaluation System and Software Quality Models. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35733
- [39] Jianping, Fu, Liu Bin, & Lu Minyan (2010). Present and future of software testability analysis. *15th IEEE International Conference on Computer Application and System Modeling*, (ICCASM), 674-681.
- [40] Jimenez, G., Taj, S. & Weaver, J. (2005). Design for Testability. *In Proceedings of the 9th Annual NCIA Conference*, 342-349.
- [41] Jungmayr, S. (2002). Design for Testability. *CONQUEST*, 57-64.
- [42] Jungmayr, S. (2002). Testability Measurement & Software Dependencies. *In Proceedings of the 12th International Workshop on Software Measurement*, 179–202.
- [43] Jalote, P. (2012). *An integrated approach to software engineering*. Springer Science & Business Media.
- [44] Jungmayr, Stefan (2002). Testability during Design. *Software Technik-Trends. Proceedings of the GI Working Group Test, Analysis and Verification of Software*, Potsdam, 10-11.
- [45] John Hunt (2007). Designing Software for Testability. Available at: http://www.regdeveloper.co.uk/2007/10/29/design_for_testability/page2.html

- [46] Joshi Megha & Sardana Neetu (2014). Design and Code Time Testability Analysis for Object Oriented Systems. *IEEE International Conference on Software Engineering*, 590-593.
- [47] Kolb R. & Muthig, D. (2006). Making testing product lines more efficient by improving the testability of product line architectures. *In Proceedings of the ISSTA 2006 workshop on Role of Software Architecture for Testing and Analysis*, (ROSATEA '06), ACM Press, 22–27.
- [48] Kiewkanya, M. & Muenchaisri, P. (2011). Constructing Modifiability Metrics by Considering Different Relationships. *Chiang Mai J. Sci.*, 38, 82-9. www.science.cmu.ac.th/journal-science/josci.html
- [49] Khatri, S., Chhillar, R.S. & Singh, V. B. (2011). Improving the Testability of Object Oriented Software during Testing and Debugging Processes. *International Journal of Computer Applications*, 35 (11), 0975 – 8887.
- [50] Kansomkeat, S. & Rivepiboon, W. (2008). An analysis technique to increase testability of object oriented components. *Software Testing Verification and Reliability*, 18, 193-219.
- [51] Khan, R. A., Mustafa, K. & Ahson, S.I. (2007). An empirical validation of object oriented design quality metrics. *Journal of King Saud University Computer and Information Sciences*, 19, 1-16.
- [52] Kumar, E. M., Sharma, P., & Sadawarti, H. (2010). Measuring testability of aspect oriented programs. In *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on IEEE*, 345-349.
- [53] Karoui, K., Dssouli R. & Cherkaoui, O. (1996). Specification transformations and design for testability. *In Proceedings of GLOBECOM'96. 1996 IEEE Global Telecommunications Conference*, 18-22 Nov, 1996, New York, NY, USA, 680- 687.

- [54] Khalid, S., Zehra, S. & Arif, F. (2010). Analysis of object oriented complexity and testability using object oriented design metrics. *Proceedings of the 2010 National Software Engineering Conference on - NSEC '10*, 1–8.
- [55] Khan, R.A. & Mustafa, K. (2009). Metric based Testability Model for Object Oriented Design (MTMOOD). *SIGSOFT Software Engineering Notes*, 34 (2).
- [56] Lee & Ming Chang (2014). Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance. *British Journal of Applied Science & Technology*, 4 (21).
- [57] Limin Shen & Shangping Ren (2006). Analysis and measurement of software flexibility based on flexible points. *Published in the Proceedings of SMEF*, 331- 341.
- [58] Lo, B.W.N. & Shi, H. (1998). A Preliminary Testability Model for Object Oriented Software. *In Proceedings of International Conference on Software Engineering, Education, Practice*, IEEE, 330–337.
- [59] Lu, B., Shi, Q. & Cao, Q. J. (2005). Research on Computation Model for Software Testability Based on Random Graph Theory. *Bulletin of the College of Computer Engineering, University of Shanghai for Science and Technology*, 27 (6), 551-555.
- [60] Lulu He & Jeffrey Carver (2009). Modifiability Measurement from a Task Complexity Perspective: A Feasibility Study. *In Proceedings IEEE, Third International Symposium on Empirical Software Engineering and Measurement*, 430-434.
- [61] Ma, L., Wang H. & Li, Y. (2007). Construct metadata model based on coupling information to increase the testability of component based software. *IEEE/ACS International Conference on Computer Systems and Applications*, AICCSA 2007, Amman, Jordan, 24-31.

- [62] Malla, P. & Gurung, B. (2012). Adaptation of Software Testability Concept for Test Suite Generation - A Systematic Review. *School of Computing Blekinge Institute of Technology*, Karlskrona Sweden, SE-371 79, 1 -74.
- [63] Mourad Badri & Fadel Toure (2012). Empirical Analysis of Object-Oriented Design Metrics for Predicting Unit Testing Effort of Classes. *Journal of Software Engineering and Applications*, 5, 513-526.
- [64] Mao, Chengying, Yansheng Lu, & Jinlong Zhang (2007). Regression testing for component-based software via built-in test design. *In Proceedings of the 2007 ACM symposium on applied computing*, 565-571.
- [65] Mouchawrab, S., Briand, L.C. & Labiche, Y. (2005). A measurement framework for object oriented software testability. *Information and Software Technology*, 47, 979-997.
- [66] Nikfard, P., Najafabadi, M. K. & Rouhani, B.D. (2013). An Empirical Study into Model Testability. *In Proceedings of IEEE, International Conference on Informatics and Creative Multimedia*, 85-92.
- [67] Nguyen T.H., Delaunay M. & Robach C. (2010). Testability Analysis Integrated Into Scicos Development Environment. *In Proceedings of IEEE, International Conference on Software Engineering and Measurement*, 143-147.
- [68] Pizzi, Nick J. (2013). A Fuzzy Classifier Approach to Estimating Software Quality. *Information Sciences*, 241, 1–11.
- [69] Phogat, M. & Kumar, D. (2011). Testability of Software System. *International Journal of Computational Engineering & Management*, 14, 84-92.
- [70] Panigrahi, Chhabi Rani & Rajib Mall, (2013). An approach to prioritize the regression test cases of object-oriented programs. *CSI Transactions on ICT*: 1-15.
- [71] Pettichord, B. (2002). Design for Testability. *In Proceedings of Pacific Northwest Software Quality Conference*, 786- 793.

- [72] Bengtsson, P., Lassing, N., Bosch, J. & Vliet H. (2004). Architecture-level modifiability analysis (ALMA). *The Journal of Systems and Software*, Elsevier, 69, 129-147.
- [73] Shaheen, M.R. & Bousquet, L. D. (2009). Analysis of the introduction of testability antipatterns during the development process, *In Proceedings of the Fourth International Conference on Software Engineering Advances*, (ICSEA), 20-25 Sept. 2009, Piscataway, NJ, USA, 128-133.
- [74] Singh, P. K., Sangwan, O. P., Pratap, A., & Singh, A. P. (2014). An Analysis on Software Testability and Security in Context of Object and Aspect Oriented Software Development. *International Journal of Security and Cybercrime, Romania*, 3(1), 17-28.
- [75] Sabane, A. (2010). Improving System Testability and Testing with Micro Architectures. *17th Working Conference on Reverse Engineering*, 309-312.
- [76] Tiwari, Rajeev & Goel Noopur (2013). Reuse: reducing test effort. *ACM SIGSOFT Software Engineering Notes*, 38 (2), 1-11.
- [77] Tsung-Han Tsai, Chin-Yu Huang & Jun-Ru Chang (2009). A Study of Applying Extended PIE Technique to Software Testability Analysis. *In Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference*.
- [78] Voas, J.M. (1996). Object-Oriented Software Testability. *In Proceedings of International Conference on Achieving Quality in Software*.
- [79] Voas & Miller (1992). Improving the Software Development Process using Testability Research. *IEEE Software*, IEEE, 114–121.
- [80] Venkatesan, K. G. S., Elamurugaselvam, M., & SCHOLAR, P. (2013). Design based object oriented Metrics to measure coupling & cohesion. *International journal of Advanced & Innovative Research*, 2(5), 778-785.
- [81] Voas & Miller (1995). Software Testability: The New Verification. *IEEE Software*, 12(3), 17-28.

- [82] V. Chowdhary, (2009). Practicing Testability in the Real World. *International Conference on Software Testing, Verification and Validation*, IEEE CSP, Washington, 260-268.
- [83] Voas & Miller (1993). Semantic metrics for software testability, *Journal of Systems and Software*, 20 (3), 207-216.
- [84] Wang S.J. & Yeh, T.H. (2009). High-level test synthesis with hierarchical test generation for delay-fault testability. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28, 1583-1596.
- [85] Wang, Y. (2003). Design for Test and Software Testability. *University of Calgary*, Available at: <http://www.ucalgary.ca/~ageras/wshop/abstracts/2003/design-for-testability.htm>
- [86] Xie, T., Zhang, L., Xiao, X., Xiong, Y. F., & Hao, D. (2014). Cooperative software testing and analysis: Advances and challenges. *Journal of Computer Science and Technology*, 29(4), 713-723.
- [87] Xiaoguang Zhou & Bo Lv (2013). ERP System Flexibility Measurement Based On Fuzzy Analytic Network Process. *Journal of Software*, 8 (8), 1943-1951.
- [88] Zheng, W. & Bundell, G. (2008). Contract-Based Software Component Testing with UML Models. *Computer Science and its Applications*, CSA. International Symposium, 13-15 October 2008, 83–102.
- [89] Zhao, L. (2006). A new approach for software testability analysis. *International Conference on Software Engineering*. In *Proceeding of the 28th international conference on Software Engineering*, Shanghai, 985–988.