

## REFERENCES

- [1] Ducasse, S. and Pollet, D., 2009. Software architecture reconstruction: A process-oriented taxonomy. *IEEE Transactions on Software Engineering*, 35(4), pp.573-591.
- [2] Murphy, G.C., Notkin, D. and Sullivan, K.J., 2001. Software reflexion models: Bridging the gap between design and implementation. *IEEE Transactions on Software Engineering*, 27(4), pp.364-380.
- [3] Postma, A., 2003. A method for module architecture verification and its application on a large component-based system. *Information and Software Technology*, 45(4), pp.171-194.
- [4] Erich, F., Amrit, C. and Daneva, M., 2014. Report: Devops literature review. *University of Twente, Tech. Rep.*
- [5] Burd, E.L., Chan, P.S., Duncan, I.M.M., Munro, M. and Young, P., 1996. Improving visual representations of code. *University of Durham, Technical Report.*
- [6] Zhang, H.Y., Urtado, C. and Vauttier, S., 2010, July. Architecture-centric development and evolution processes for component-based software. In *Proc. of 22nd SEKE Conf., Redwood City, USA (July 2010).*
- [7] Ayewah, N., Pugh, W., Morgenthaler, J.D., Penix, J. and Zhou, Y., 2007, June. Evaluating static analysis defect warnings on production software. In *Proceedings of the 7th ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering* (pp. 1-8). ACM.
- [8] De Silva, L. and Balasubramaniam, D., 2012. Controlling software architecture erosion: A survey. *Journal of Systems and Software*, 85(1), pp.132-151.
- [9] Che, M. and Perry, D.E., 2014, April. Architectural design decisions in open software development: A transition to software ecosystems. In *2014 23rd Australian Software Engineering Conference* (pp. 58-61). IEEE.

- [10] Shahin, M., 2015, September. Architecting for DevOps and Continuous Deployment. In *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference* (pp. 147-148). ACM.
- [11] Garlan, D. and Schmerl, B., 2004, May. Using architectural models at runtime: Research challenges. In *European Workshop on Software Architecture* (pp. 200-205). Springer Berlin Heidelberg.
- [12] Stark, G., Skillicorn, A. and Ameen, R., 1998. An examination of the effects of requirements changes on software releases. *CROSSTALK The Journal of Defence Software Engineering*, pp.11-16.
- [13] Chen, L., 2015, May. Towards Architecting for Continuous Delivery. In *Software Architecture (WICSA), 2015 12th Working IEEE/IFIP Conference on* (pp. 131-134). IEEE.
- [14] <http://www.uml-diagrams.org/multi-layered-application-uml-model-diagram-example.html>
- [15] Hammer, D.K. and Ionita, M., 2005. SCENARIO-BASED SOFTWARE ARCHITECTURE EVALUATION METHODS An overview.
- [16] Van Gorp, J., Brinkkemper, S. and Bosch, J., 2005. Design preservation over subsequent releases of a software product: a case study of Baan ERP. *Journal of Software Maintenance and Evolution: Research and Practice*, 17(4), pp.277-306.
- [17] Williams, B.J. and Carver, J.C., 2010. Characterizing software architecture changes: A systematic review. *Information and Software Technology*, 52(1), pp.31-51.
- [18] Knodel, Jens, Dirk Muthig, and Matthias Naab. "Understanding Software Architectures by Visualization--An Experiment with Graphical Elements." *2006 13th Working Conference on Reverse Engineering*. IEEE, 2006.
- [19] Murphy, Gail C., and David Notkin. "Reengineering with reflexion models: A case study." *Computer* 30.8 (1997): 29-36.

- [20] Hochstein, Lorin, and Mikael Lindvall. "Diagnosing architectural degeneration." *28th Annual NASA Goddard Software Engineering Workshop, 2003. Proceedings.* IEEE, 2003..
- [21] Knodel, Jens, et al. "Static evaluation of software architectures." *Conference on Software Maintenance and Reengineering (CSMR'06).* IEEE, 2006.
- [22] Bandara, Vidudaya, and Indika Perera. "Identifying Software Architecture Erosion Through Code Comments." *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer).* IEEE, 2018.
- [23] Meyer, Mathias. "Continuous integration and its tools." *IEEE software* 31.3 (2014): 14-16.
- [24] Smith, Tony. "Protecting the process [source code management]." *Engineering & Technology* 5.4 (2010): 51-53.
- [25] Venkitachalam, Hariharan, et al. "Automated Continuous Evaluation of AUTOSAR Software Architecture for Complex Powertrain Systems." *INFORMATIK 2017* (2017).
- [26] Vasilescu, Bogdan, et al. "Quality and productivity outcomes relating to continuous integration in GitHub." *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering.* ACM, 2015.
- [27] Kim, J. and Garlan, D. (2010). Analyzing architectural styles. *Journal of Systems and Software*, 83(7), pp.1216-1235.69
- [28] Monroe, Robert T., et al. "Architectural styles, design patterns, and objects." *IEEE software* 14.1 (1997): 43-52.
- [29] Mehta, Nikunj R., and Nenad Medvidovic. "Composing architectural styles from architectural primitives." *ACM SIGSOFT Software Engineering Notes.* Vol. 28. No. 5. ACM, 2003.
- [30] Vatanawood, Songpon Thongkumand Wiwat. "An Approach of Software Architectural Styles Detection Using Graph Grammar."

- [31] Katte Darshan, A., and P. Suganya. "Json is Efficient over the XML in Native Application." *International Journal of Computer Applications* 165 (2017): 545-586.
- [32] Pandey, Mrinal, and Rajiv Pandey. "JSON and its use in Semantic Web." *International Journal of Computer Applications* 164.11 (2017): 10-16.
- [33] Kövesdán, Gábor, Márk Asztalos, and László Lengyel. "Architectural design patterns for language parsers." *Acta Polytechnica Hungarica* 11.5 (2014): 39-57.
- [34] Lyon, Douglas A. "Semantic annotation for Java." *Journal of Object Technology* 9.3 (2010).
- [35] Buitrago Flórez, Francisco, et al. "Changing a generation's way of thinking: Teaching computational thinking through programming." *Review of Educational Research* 87.4 (2017): 834-860.
- [36] Mei, Hong, and Jun-Rong Shen. "Progress of research on software architecture." *Ruan Jian Xue Bao(Journal of Software)* 17.6 (2006): 1257-1275.
- [37] Shaw, Mary, and David Garlan. *Characteristics of Higher-Level Languages for Software Architecture*. No. CMU-CS-94-210. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1994.
- [38] Weyuker, Elaine J., and Filippos I. Vokolos. "Experience with performance testing of software systems: issues, an approach, and case study." *IEEE transactions on software engineering* 26.12 (2000): 1147-1156.
- [39] Marshall, Adam C. "A Conceptual model of software testing." *Software Testing, Verification and Reliability* 1.3 (1991): 5-16.
- [40] <http://theagileadmin.com/what-is-devops/>
- [41] <https://insights.sei.cmu.edu/devops/2015/01/continuous-integration-in-devops-1.html>
- [42] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.5526&rep=rep1&type=pdf>

[43] <https://www.thinkhdi.com/~media/HDICorp/Files/White-Papers/whtppr-1112-devops-kim.pdf>

[44] [http://www.sparxsystems.com/enterprise\\_architect\\_user\\_guide/9.2/software\\_engineering/reverseengineersourcecode.html](http://www.sparxsystems.com/enterprise_architect_user_guide/9.2/software_engineering/reverseengineersourcecode.html)

[45] [http://www.codingthearchitecture.com/2014/06/24/software\\_architecture\\_as\\_code.html](http://www.codingthearchitecture.com/2014/06/24/software_architecture_as_code.html)

[46] <http://www.architexa.com/>

[47] <http://www.stack.nl/~dimitri/doxygen/>

[48] <http://softarch.usc.edu/projects/automated-recovery-of-software-system-designs/>