REFERENCES

[1] L. Bass, P. Clements, and R. Kazman. Software Architecture in Practice (2nd Edition). Addison-Wesley Professional, 2 edition, April 2003

[2] M. De Silva and I. Perera, "Preventing software architecture erosion through static architecture conformance checking", in IEEE 10th International Conference on Industrial and Information Systems (ICIIS), 2015.

[3] Parnas, D. L. Software aging. In Proceedings of the 16th international conference on Software engineering (Los Alamitos, CA, USA, 1994), ICSE '94, IEEE Computer Society Press, pp. 279–287.)

[4] van Gurp, J., Brinkkemper, S., and Bosch, J. Design preservation over subsequent releases of a software product: a case study of baan erp: Practice articles. J. Softw. Maint. Evol. 17 (July 2005), 277–306.

[5] Guo, G. Y., Atlee, J. M., and Kazman, R. A software architecture reconstruction method. In Proceedings of the TC2 First Working IFIP Conference on Software Architecture (WICSA1) (Deventer, The Netherlands, The Netherlands, 1999), WICSA1, Kluwer, B.V., pp. 15–34.

[6] Murphy, G. C., Notkin, D., and Sullivan, K. J. Software reflexion models: Bridging the gap between design and implementation. IEEE Trans. Softw. Eng. 27, 4 (Apr. 2001), 364–380.

[7] Lung, C.-H., Zaman, M., and Nandi, A. Applications of clustering techniques to software partitioning, recovery and restructuring. J. Syst. Softw. 73, 2 (Oct. 2004), 227–244.

[8] Sartipi, K. Software architecture recovery based on pattern matching. In Proceedings of the International Conference on Software Maintenance (Washington, DC, USA, 2003), ICSM '03, IEEE Computer Society, pp. 293

[9] Jansen, A., Bosch, J., and Avgeriou, P. Documenting after the fact: Recovering architectural design decisions. J. Syst. Softw. 81, 4 (Apr. 2008), 536–557

[10] Abi-Antoun, M., Aldrich, J. (2009). Static extraction and conformance analysis of hierarchical runtime architectural structure using annotations. In Proceedings of the 24th ACM SIGPLAN conference on Object oriented programming systems languages and applications (OOPSLA '09). ACM, New York, NY, USA, 321-340. [Online]. Available from: DOI=http://dx.doi.org/10.1145/1640089.1640113

[11] Len Bass, Paul Clements, R. K. Software Architecture in Practice. 2000

[12] M. Mirakhorli, "Preserving the Quality of Architectural Tactics in Source Code", 2014.

[13] Rosik, J., Le Gear, A., Buckley, J., and Ali Babar, M. An industrial case study of architecture conformance. In Proceedings of the Second ACM-IEEE

international symposium on Empirical software engineering and measurement (New York, NY, USA, 2008), ESEM '08, ACM, pp. 80–89

[14] Bennett, K. (1996). Software evolution: past, present and future. Information and software technology, 38(11), 673-680.

[15] Paul C. Clements. "A survey of architecture description languages". In Proceedings of the Eighth International Workshop on Software Specification and Design. IEEE Computer Society Press, 1996

[16] L. de Silva, "A Rationale-based Architecture Description Language using the Oslo Modelling Platform," Master's thesis, University of St Andrews, 2008

[17] L. de Silva and D. Balasubramaniam, "A model for specifying rationale using an architecture description language," in Software Architecture. Proceedings of the 5th European Conference on Software Architecture (ECSA 2011) (I. Crnkovic, V. Gruhn, and M. Book, eds.), pp. 319–327, Springer Berlin Heidelberg, 2011

[18] J. Knodel, D. Muthig, and M. Naab. Understanding software architectures by visualization–an experiment with graphical elements. In WCRE '06: Proceedings of the 13th Working Conference on Reverse Engineering (WCRE 2006), pages 39–50, Washington, DC, USA, 2006. IEEE Computer Society

[19] G. C. Murphy and D. Notkin. Reengineering with reflexion models: A case study. Computer,30(8):29–36, 1997.

[20] L. Hochstein and M. Lindvall. Diagnosing architectural degeneration. sew, 00:137, 2003.

[21] J. Knodel, D. Muthig, M. Naab, and M. Lindvall. Static evaluation of software architectures. In CSMR '06: Proceedings of the Conference on Software Maintenance and Reengineering, pages 279–294, Washington, DC, USA, 2006. IEEE Computer Society.

[22] U. Liyanage and I. Perera, "Traceability Model For Viewing Architectural Tactics Using Code Comments", 2017.

[23] Continuous Integration and Its Tools. (2014). IEEE Software, 31(3), pp.14-16.

[24] Smith, T. (2010). Protecting the process [source code management]. Engineering & Technology, 5(4), pp.51-53.

[25] Quibeldey-Cirkel, K. and Thelen, C. (2012). Continuous Deployment. Informatik-Spektrum, 35(4), pp.301-305.

[26] CSS-Tricks. (2018). Why You Should Use Continuous Integration and Continuous Deployment | CSS-Tricks. [online] Available at: https://css-tricks.com/continuous-integration-continuous-deployment/ [Accessed 22 Feb. 2018].

[27] Kim, J. and Garlan, D. (2010). Analyzing architectural styles. Journal of Systems and Software, 83(7), pp.1216-1235.

[28] Monroe, R., Kompanek, A., Melton, R. and Garlan, D. (1997). Architectural styles, design patterns, and objects. IEEE Software, 14(1), pp.43-52.
[29] Mehta, N. and Medvidovic, N. (2003). Composing architectural styles from architectural primitives. ACM SIGSOFT Software Engineering Notes, 28(5), p.347.
[30] Thongkum, S. and Vatanawood, W. (2014). An Approach of Software Architectural Styles Detection Using Graph Grammar. International Journal of Engineering and Technology, 6(2), pp.123-127.
[31] Darshan, K. and P., S. (2017). Json is Efficient over the XML in Native Application. International Journal of Computer Applications, 165(8), pp.14-17.
[32] Pandey, M. and Pandey, R. (2017). JSON and its use in Semantic Web. International Journal of Computer Applications, 164(11), pp.10-16.
[33] Architectural Design Patterns for Language Parsers. (2014). Acta Polytechnica Hungarica, 11(5).
[34] Lyon, D. (2010). Semantic Annotation for Java. The Journal of Object Technology, 9(3), p.19.
[35] Ahuja, K. (2018). Are annotations bad?. [online] Java Code Geeks. Available at: https://www.javacodegeeks.com/2015/08/are-annotations-bad.html [Accessed 22 Feb. 2018].
[36] dzone.com. (2018). How Do Annotations Work in Java? - DZone Java. [online] Available at: https://dzone.com/articles/how-annotations-work-java [Accessed 22 Feb. 2018].
[37] ZHANG, L. (2008). Software Architecture Evaluation. Journal of Software, 19(6), pp.1328-1339.
[38] Garlan, D. and Shaw, M. (1994). An Introduction to Software Architecture. Pittsburgh: Carnegie Mellon University.
[39] Weyuker, E. and Vokolos, F. (2000). Experience with performance testing of software systems: issues, an approach, and case study. IEEE Transactions on Software Engineering, 26(12), pp.1147-1156.
[40] Marshall, A. (1991). A conceptual model of software testing. Software Testing, Verification and Reliability, 1(3), pp.5-16.