

# Week 2 Journal

Rodger Byrd

Choose a top journal or conference in your field with  $\approx$  45 papers one or two 2018/2019 issues. (DO NOT search for papers with keywords.. but can search for journals/conf title).

## I. PROCESS

The first section in the journal should discuss your process and your learning about the process (do not discuss the content of the papers).

## II. RAW NOTES

The second section is your raw notes from critical/creative reads as a latex document, and include all the papers in the references. Not expecting/wanting full sentences, grammar, want your raw notes. If you use handwritten notes, you can take pic and include them as figures.

The last section is references.. which should start with the citation for the primary conference, plus the citations for have all the papers from the browse, with the timing notes.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21]

## REFERENCES

- [1] F. Jaafar, F. Khomh, Y.-G. Guhneuc, and M. Zulkernine, "Anti-pattern mutations and fault-proneness." 14th International Conference on Quality Software, 2014, p. 10.
- [2] W. Ma, L. Chen, Y. Zhou, and B. Xu, "Are anti-patterns coupled? an empirical study." IEEE International Conference on Software Quality, Reliability and Security, 2015, p. 10.
- [3] B. Chen and Z. M. J. Jiang, "Characterizing and detecting anti-patterns in the logging code." IEEE/ACM 39th International Conference on Software Engineering, 2017, p. 11.
- [4] W. Fenske, "Code smells in highly configurable software." International Conference on Software Maintainence, 2015, p. 4.
- [5] F. Palomba, D. A. Tamburri, A. Serebrenik, A. Zaidman, F. A. Fontana, and R. Oliveto, "How do community smells influence code smells?" ACM/IEEE 40th International Conference on Software Engineering: Companion Proceedings, 2018, p. 2.
- [6] F. Ebert, F. Castor, N. Novielli, and A. Serebrenik, "Confusion detection in code reviews." IEEE International Conference on Software Maintenance and Evolution, 2017.
- [7] —, "Confusion in code reviews: Reasons, impacts, and coping strategies." IEEE International Conference on Software Analysis, Evolution and Reengineering, 2019.
- [8] N. Nahar and K. Sakib, "Acdpr: A recommendation system for the creational design patterns using anti-patterns." IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering, 2016.
- [9] C. Denzler and D. Gruntz, "Design patterns: Between programming and software design." 30th International Conference on Software Engineering, 2008.
- [10] M. Huaxin and J. Shuai, "Design patterns in software development."
- [11] J. S. Fant, "Building domain specific software architectures from software architectural design patterns."
- [12] M. Aoyama, "Evolutionary patterns of design and design patterns."
- [13] D. J. Ferreira, "Work in progress: Exploring programming anti-patterns as emphasis on creativity in the teaching of computer programming."
- [14] C. Izurieta and J. M. Bieman, "How software designs decay: A pilot study of pattern evolution." First International Symposium on Empirical Software Engineering and Measurement.
- [15] R. F. G. Antoniol and L. Cristoforetti, "Using metrics to identify design patterns in object-oriented software."
- [16] T. Feng, J. Zhang, H. Wang, and X. Wang, "Software design improvement through anti-patterns identification."
- [17] F. ZENG, A. CHEN, H. Wang, and X. TAO, "Study on software reliability design criteria based on defect patterns."
- [18] K. Z. Sultana, "Towards a software vulnerability prediction model using traceable code patterns and software metrics."
- [19] Z. Zemin, "Study and application of patterns in software reuse." IITA International Conference on Control, Automation and Systems Engineering, 2009.
- [20] A. Ghazarian, "Work in progress: Transitioning from novice to expert software engineers through design patterns: Is it really working?"
- [21] A. Tahmid, N. Nahar, and K. Sakib, "Understanding the evolution of code smells by observing code smell clusters." IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering, 2016.