Using HPC Resources to Exploit Big Data for Code Review Analytics

Patanamon Thongtanunam[1], Shane McIntosh[2], Ahmed E. Hassan[1], and
Hajimu Iida[3]

[1] Queen's University, Canada
[2] McGill University, Canada
[3] Nara Institute of Science and Technology, Japan
p.thongtanunam@cs.queensu.ca, shane.mcintosh@mcgill.ca,
ahmed@cs.queensu.ca, iida@itc.naist.jp

Abstract. Code review is one of the crucial software activities where
developers and stakeholders collaborate with each other in order to assess
software changes. Since code review processes act as a final gate for new
software changes to be integrated into the software product, an intense
collaboration is necessary in order to prevent defects and produce a high
quality of software products. Recently, code review analytics has been
implemented in projects (for example, StackAnalytics[4] of the OpenStack
project) to monitor the collaboration activities between developers and
stakeholders in the code review processes. Yet, due to the large volume
of software data, code review analytics can only report a static summary
(e.g., counting), while neither insights nor instant suggestions are pro-
vided. Hence, to better gain valuable insights from software data and help
software projects make a better decision, we conduct an empirical inves-
tigation using statistical approaches. In particular, we use the large-scale
data of 196,712 reviews spread across the Android, Qt, and OpenStack
open source projects to train a prediction models in order to uncover the
relationship between the characteristics of software changes and the like-
lihood of having poor code review collaborations. We extract 20 patch
characteristics which are grouped along five dimensions, i.e., software
changes properties, review participation history, past involvement of a
code author, past involvement of reviewers, and review environment di-
mensions. To validate our findings, we use the bootstrap technique which
repeat the experiment 1,000 times. Due to the large volume of studied
data, and an intensive computation of characteristic extraction and find-
ing validation, the use of the High-Performance-Computing (HPC) re-
sources is mandatory to expedite the analysis and generate insights in
a timely manner. Through our case study, we find that the amount of
review participation in the past and the description length of software
changes are a significant indicator that new software changes will suffer
from poor code review collaborations [2017]. Moreover, we find that the
purpose of introducing new features can increase the likelihood that new
software changes will receive late collaboration from reviewers. Our find-
ings highlight the need for the policies of software change submission that

---

[4] http://stackalytics.com/

monitor these characteristics in order to help software projects improve the quality of code reviews processes. Moreover, based on our findings, future work should develop real-time code review analytics implemented on HPC resources in order to instantly provide insights and suggestions to software projects [2016, 2015, 2015, 2014, 2014].

Keywords: Big Data, Software Engineering, Code Review Analytics, Social Computing

References

[2017] Thongtanunam, P., Mcintosh, S., Hassan, A.E., Iida, H.: Review Participation in Modern Code Review. Empir. Softw. Eng. 22, 768–817 (2017).

[2016] Thongtanunam, P., Mcintosh, S., Hassan, A.E., Iida, H.: Revisiting Code Ownership and its Relationship with Software Quality in the Scope of Modern Code Review. In: Proceedings of the 38th International Conference on Software Engineering (ICSE). pp. 1039–1050 (2016).

[2015] Thongtanunam, P., Tantithamthavorn, C., Kula, R.G., Yoshida, N., Iida, H., Matsumoto, K.: Who Should Review My Code? A File Location-Based Code-Reviewer Recommendation Approach for Modern Code Review. In: Proceedings of the the 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SAnER). pp. 141–150 (2015).

[2015] Thongtanunam, P., McIntosh, S., Hassan, A.E., Iida, H.: Investigating Code Review Practices in Defective Files: An Empirical Study of the Qt System. In: Proceedings of the 12th International Working Conference on Mining Software Repositories (MSR). pp. 168–179 (2015).

[2014] Thongtanunam, P., Kula, R.G., Cruz, A.E.C., Yoshida, N., Iida, H.: Improving Code Review Effectiveness through Reviewer Recommendations. In: Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE?14). pp. 119–122 (2014).

[2014] Thongtanunam, P., Yang, X., Yoshida, N., Kula, R.G., Ana Erika, C.C., Fujiwara, K., Iida, H.: ReDA: A Web-based Visualization Tool for Analyzing Modern Code Review Dataset. In: The proceeding of the 30th International Conference on Software Maintenance and Evolution (ICSME). pp. 606–609 (2014).