Paper Five Summary

Matthew Neal, Joseph Sankar, and Alexander Sobran

October 18, 2015

Reference

Nagappan et al. [1] listed below.

Important Keywords

Organizational metrics Metrics designed to give a balanced view of organizational complexity with respect to software development. The authors argue that these metrics, as a measure of organizational structure, greatly affect product quality.

Failure-proneness The probability that a particular piece of software will fail in production. The authors hope to reduce this by incorporating organizational metrics into models.

Software Metrics The authors define them as things like code churn, code complexity, code coverage, and code dependencies. These metrics have been used to determine fault proneness.

Conway's Law Organizations that design systems are constrained to produce the systems which are copies of the communication structures of these organizations.

Feature Extraction

Motivational Statements The authors perspective is that software metrics have been used to a great degree to predict fault proneness in software, but that organizational metrics have not been used to try to predict failure proneness. Their main motivation is determining whether organizational metrics have a comparable efficacy in predicting failure proneness and also whether it is possible to accurately define and measure organizational metrics.

Future Work The authors hope to replicate the study with other Microsoft projects as well as organizations outside of Microsoft. They have mentioned that they are already begun collaborating with the Fraunhofer research institute (pg. 10) for this purpose. The authors also hope to research open source teams and their "virtual organizations", global software development, and even the social and cognitive aspects of the engineers themselves.

Baseline Results The authors present strong baseline results when compared with software metrics in the area of fault predictions. The main comparison metrics used in the paper are precision and recall. The authors found that models using organizational metrics (specifically organizational structure) outperformed all software metric based prediction models. The closest competitor for precision was code coverage at 83.8% vs organizational structure at 86.2%, while for recall the closest competitor was code churn at 79.9% vs organizational structure at 84.0%.

Statistical Tests The authors use Spearman's rank correlation, which is an assessment of how correlated two specific variables are. They wanted to prove that the predicted faults and the actual faults of Windows Vista were highly correlated to show the sensitivity of their method. They found, as shown in Figure 4 in the paper that there was a statistically significant high correlation between predicted and actual failures with 99% confidence.

Possible Improvements

- The phrasing in particular sections is grammatically incorrect. Here are a couple examples: "The deeper in the tree is the ownership the more focused the activities, communication, and responsibility" (pg. 4) and "The lower level is the ownership the better is the quality" (pg. 5). More editing should have been done to fix these errors.

Connection to Other Papers

In [2], this paper was referenced by the authors as previous research. The organizational metrics described in this paper were used as a set of features in [2]. It was also used in that paper to reinforce the conclusion that organizationally distributed development affects the number of post-release defects.

References

[1] Nachiappan Nagappan, Brendan Murphy, and Victor Basili. The influence of organizational structure on software quality: An empirical case study. In *Proceedings of the 30th International Conference on Software Engineering*, ICSE '08, pages 521–530, New York, NY, USA, 2008. ACM.

[2] Christian Bird, Nachiappan Nagappan, Premkumar Devanbu, Harald Gall, and Brendan Murphy. Does distributed development affect software quality?: An empirical case study of windows vista. *Commun. ACM*, 52(8):85–93, August 2009.