

Paper Review:

Assessing the Value of Branches with What-if Analysis

Braden D. Licastro
Department of Computer Science
Allegheny College
licastb@allegheny.edu
<http://www.fullforceapps.com/>

February 5, 2013

1 Summary

This paper analyses the costs of branch usage in source code management systems and assesses alternate branch structures to reduce delay involved with merging branches. [1]. With larger and larger code projects, many developers have implemented the use of source management systems to prevent keeping other developers from working efficiently. Using this system it is possible to take existing code, branch that code and work on a specific feature while another developer works on another feature using the same code base. Once modifications are complete the changes will be merged back to the master, but depending how the project was originally branched, conflicts, erroneous code, and other sub-par data may be present that worked in their respective branches but break when merged. The authors of the paper survey several hundred Microsoft developers and gather branching statistics. From this, they research and propose several methods of structuring branches to optimize time spent merging projects.

2 Critique

Throughout the research article Microsoft employees' experiences were continually referenced which raises concern. The employees that are hired at these companies are typically the best in their field. This raises some concern regarding the validity of the surveys. Though mentioned in the paper that the study is most geared toward benefiting this type of professional, many open source projects have a massive user base and could also benefit. Being able to gauge the results with a mixed group of developers would be far more effective in determining the performance of the proposed branch structures and techniques.

In addition to this, the authors outline the method used to normalize the data so various branch delays and isolations can be compared. The authors make an assumption where more code modifications equates to a longer delay. This may be true in many cases, but there are also cases where this assumption is invalid. It is possible that one code edit could be complex enough that it can take as long as n number of edits on more straightforward code. Because of this it is possible for the normalized code to be skewed just the same as it would have been before normalization.

3 Synthesis

After reading the research paper I concluded that additional research could be performed to substantiate the claims outlined in the paper, and possibly improve the effectiveness. As a future area of research it would be beneficial to verify their results on smaller but detailed projects. Windows is a complex piece of software with many parts, but a small program of high complexity could potentially yield different results.

Lastly an area for future research would be running the same tests with users of different levels of competency. An efficient and experienced developer could potentially develop a branched program in a way that it would ensure a seamless integration, while an engineer with less experience may write a functional program that needs additional code modifications in order to integrate seamlessly with the master. Merging these separate branches coded using different levels of experience and techniques can yield vastly different results than what were recorded previously.

References

- [1] Christian Bird and Thomas Zimmermann. Assessing the value of branches with what-if analysis. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, FSE '12, pages 45:1–45:11, New York, NY, USA, 2012. ACM.