Defining and Optimizing the "North Star" Metric of Measuring Software

Anonymous Author(s)

I. Background

Defining the adequate software metrics can be used to measure and optimize the productivity of software development and maintenance, which can further enable the organizations, managers, and developers with the ability of building higher quality software with higher efficiency.

The "north star" metric is one metric that matters, that refers to the absolute core metrics related to business/strategy in the current phase of a product. Once established, it shines like the Polaris in the air, which can guide the team in the same direction to improve the related metrics. In theory, the "north star" can be explicitly measured, and can reflect the product value and customer value with the leading indicator of benefits. In the domain of software engineering, the "north start" can be composed of multi-dimensional metrics, like the business value, the quality and the cost of the software, of which related metrics have been explored in previous studies. For example, the SonarQube tool has been studied with metrics related to software quality, that can be used to automatically detect bugs, vulnerabilities, code smells, coverage, etc.. There are also some standards, like ISO 25010, that can help define the cornerstone model for the given system to be measured. And Google also used the DORA (metrics) [1] to measure and optimize its software development tasks.

II. Key Challanges

Although the metrics of measuring software has been widely studied in the community, it still is an open research question. This challenge aims to identify, recognize, or define the north star metric, that can be used to accurately capture the core value in software development. It can be optimized to further guide practitioners in understanding and improving software effectiveness and efficiency in a data-driven way. To this end, the corresponding difficulties may stem from several aspects:

- **Difficulties in understanding**: Although we have massive metrics generated by analysis tools, it is difficult for developers to find right metrics and identify key problems. Many metrics or problems reported, such as bad smells, are intermediate metrics or problems, which are difficult to measure the actual impact on software quality.
- **Difficulties in optimization**: A global perspective is needed in software development. Improving a single problem may not improve the overall quality, especially when the problem is not a bottleneck. Additionally, improving a single problem may introduce more problems in some cases. We need to identify key problems and make continuous improvements.
- **Difficulties in measuring and unifying**: It is difficult to establish unified metrics for horizontal comparison of software development. For example, line of code (LOC) can be used to estimate the workload, but it does not take into account the complexity of programming languages, business logic, and implementation algorithms in different project. Story point also helps development team to plan their work and improve delivery predictability, but estimating story points with multiple teams is a challange.

[1] https://www.devops-research.com/research.html

III. Expecting Requirements

The north star metric can capture the core value of software thate should be identified in software development. It could be applicable to different software development scenarios, for example in embedded or cloud-based software development, and agile or waterfall development process with its measuability and optimizability. In addition, an automated approach should be provided to help figure out the north star metric in different scenarios. The effectiveness of the north star metric should be evaluated with empirical studies, experiments etc.

IV. Related Works

Recent works have focused on the quality [1] and productivity [2] in software development. Some famous software companies, like Google [3] and Microsoft [4] , are also interested in identifying and predicting the develops' productivity. Open source software is available to analyze software development artifacts and processes, and provide a set of metrics for evaluation.

References

[1] N Tsuda, H Washizaki, K Honda, H Nakai, Y Fukazawa, M Azuma, T Komiyama, T Nakano, H Suzuki, S Morita, K Kojima. Wsqf: Comprehensive software quality evaluation framework and benchmark based on square. In2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) 2019 May 25 (pp. 312-321). IEEE.

[2] MA Storey, T Zimmermann, C Bird, J Czerwonka, B Murphy, E Kalliamvakou. Towards a theory of software developer job satisfaction and perceived productivity. IEEE Transactions on Software Engineering. 2019 Sep 27;47(10):2125-42.

[3] E Murphy-Hill, C Jaspan, C Sadowski, D Shepherd, M Phillips, C Winter, A Knight, E Smith, M Jorde. What predicts software developers' productivity?. IEEE Transactions on Software Engineering. 2019 Feb 19;47(3):582-94.

[4] N Forsgren, MA Storey, C Maddila, T Zimmermann, B Houck, J Butler. The SPACE of Developer Productivity: There's more to it than you think. Queue. 2021 Feb 28;19(1):20-48.