Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CSU330312 Software Engineering

Measuring Software Engineering Report

Liam Junkermann - 19300141

November 23, 2021

Contents

Introduction

Software engineering is a notoriously difficult to track for time and productivity. Some engineers may be more efficient with some technologies and not others, and tasks may have varying difficulty making engineering activity and productivity particularly difficult to accurately and consistently track productivity. This report will discuss various ways engineering activity can be measured, appropriate data collected, and finally processed to accurately profile engineers. In addition, the moral, ethical, and legal consequences of these methods will be explored.

1 Measurable Data

There are a number of metrics which can be used to determine engineering activity depending on the way a given repository or organisation is set up.

Commits and code frequency

The simplest way to track engineering productivity could be based on the commit and code frequency charts provided by most version control providers. These charts allow a supervisor, or even the engineer themselves, to see the number of commits they make over a period of time, as well as the number of lines added or removed from the files within the repository. This method of measuring engineering activity is rudimentary and does not account for tasks which may take more time to research, plan, or structure – all subtasks which would not be reflected in a simple commit or code frequency chart.

Pull Request creation or review frequency

Version control is used to allow many people to work on a project together at the same time. Organisations will use pull requests to combine the work of various engineers in order to begin the production pipeline (the flow within which the code is built and packaged for release to production or testing groups). While commit frequency and volume are important to track activity, and break down tasks, ultimately contribution to production environments is what dictates success and progress. Pull requests are the measure for those production contributions, they allow the engineer who has worked on a feature or task to share that work with their team, and allows the rest of the team to make comments about the functionality or approach used to solve the problem presented in the task or feature. It is important, for both the team and the product they are working on, that the new features sent to the production pipeline will work and solve the problem stated. Tracking engineering activity through pull requests is an effective way to track both individual engineering contributions to a production product, and the contributions from other team members through the code review process.

Task and time tracking systems

One of the more effective and holistic ways to track engineering activity and performance leverages and expands upon the previous two solutions. Using a system like JIRA, Azure Dev Ops (ADO) or even certain functions of

Github allow teams to track tasks that need to be completed and track progress through tasks broken down to different degrees of complexity, as well as link progress with commits and completed pull requests. This method of measuring engineering performance allows progress on tasks to be tracked in reference with actual effort and hours of work put into a given task or subtask. This also allows the management of the team to improve as time estimation becomes more accurate with iterations of using this kind of solution.

Testing Coverage

An important part of any project which looks to see a production environment is the testability and rigorous testing to ensure that the solution can run effectively in that production environment. Any solutions a developer writes must be tested, and many production pipelines provide engineers graphs or statistics about code coverage – as in which lines of code were tested and which were not. These statistics allow engineers to catch bugs and ensure their solutions is being tested appropriately.

2 Connecting and performing Calculations on Activity Data sets

3 Computation to profile software engineers

4 Morality, Ethics, and Legality of Collecting and processing engineer measurement