CSU33012-Measuring-Software-Engineering

Lydia MacBride
19333944

December 2021

Contents

1 Introduction

This document is my submission for the Measuring Software Engineering Report for CS33012 Software Engineering. If for some reason you are reading it as a standalone PDF, you can find the most up to date version on the *Github* repo for this assignment.

This document is a companion piece to my own method of visualising a software engineer's productivity, which can be found in its own *Github* repo. Due to my own moral issues with attempting to create some usable metric of an individuals ”productivity”, further outlined in the **Ethical Considerations** section, I instead opted to create a light-hearted and novel use case for the data pulled from the *Github* API: choosing to map the data to the world generation parameters for *Minecraft*.

2 How to Measure Software Engineering

2.1 Monitoring Commits

One simple approach we could take to measure the productivity of a software engineer is to measure the number of commits they make to a code base. This would provide a very simple insight into which developers contributing to a repo are more productive than others, however on its own this is a very rudimentary method of measurement. This could be improved marginally by factoring the times commits are made, and then looking at the frequency of commits in particular. Together these metrics would give an over view to how much, and how regularly an engineer contributes to a project.

However there are flaws with this metric as it does not take into account code quality, code size and a developer's overall usefullness to a project. Additionally, this method will most likely end up punishing developers with workflows that do not align with whats expected of a ”typical” developer.

2.2 Code Complexity

A significantly more robust method of measuring productivity would be to analyse the code complexity and scale of each commit. This could then be used to mark engineers who commit larger and more complex code segments as being more valuable to a projects overall completion. However there are a large number of pitfalls to this approach.

First off, code analysis is more computationally expensive than some simple metric based on commit count/frequency. Additionally, knowing what ”good” code is for any given project often times requires some direct analysis from someone who is familiar with the specifications of the project itself, and to attempt measuring this algorithmically would almost require some form of digital crystal ball; a piece of software that knows exactly what ”good” code for a project should look like, and at that point why would we even need human developers. Lastly, and more crucially, more complex code is not universally a good thing, there are so, so many situations where a concise one-liner is far more preferable to a mess of nested loops and conditionals.

3 Tools for Measurement

When it comes to hosting and running in-house measurement methods, any organisation wishing to do so has a plethora of affordable for-rent web-hosting and computational power. Services such as *Amazon Web Services*, *Google Cloud*, *Microsoft Azure* and many, many more all offer everything necessary to quickly set up and run your employee monitoring method of choice. And as a bonus, if you are an Irish business wishing to monitor the productivity of your developers, all of the hosting will most likely be very local to you.

Additionally there lots of pre-made tools to aid in measuring the productivity of software engineers: ranging from simple libraries for a variety of programming languages, such as *PyGithub*, to full suites providing ”deep” insights into developer productivity. All that’s to say, it is far from a niche idea for a piece of software, and there are bucket-loads of startups attempting to pounce on this rapidly emerging demand.

4 Potential Sources of Data

Perhaps the most obvious source of data for measuring the productivity of workers are the various Version Control Systems utilised by any self respecting team of software engineers. With any decent VCS offering a number of key workflow features to allow multiple programmers to access and work on the same code-base simultaneously, a very key byproduct is created: data. Full logs of what, when and most crucial, who, made changes or contributions to a project. Bundle in the fact that all the main players in pre-built VCSs offer easy to use APIs to access this data with third party productivity measurement suites, of which there are many, many startups offering some form of way to visualise this data.

With a step up in intrusion, another source of employee data primed for scraping and processing are digital meeting spaces like *Slack*. Offering information on when employees are interacting with the platform, are at their devices or are away from them, and for how long at a time; providing data on which employees are communicating with one another, various avenues for processing individual messages. This opens up any new form of measuring productivity to factor in how sociable an employee is, their active hours at their devices, and perhaps even attempting to measure the mood of employees through the tone of their messages.

But why stop at monitoring these activities in the digital world. Provided ethics are far from being on the table at this point, through a combination of any of the recent and emerging technologies like facial recognition, GPS tracking, voice processing, etc an employee's whole time in a place of work could be monitored and used to generate some form of metric for productivity. Every employees whereabouts known at any given moment, whether they are firmly planted at their work station or wandering around the office chatting with coworkers; an individuals mood determined by their facial expression and tone of voice. Whilst seemingly endless in potential for management to keep a close eye on its workforce, there are so many sacrifices to privacy, freedom and comfort for employees, that to seriously pursue this kind of monitoring would require a complete lack of conscience.

5 Ethical Considerations

Fundamentally I believe that trying to reduce an individuals worth to an easy to recount numeral or metric is an inherently amoral view to human life. An individual person is so complex, multi-faceted and unique that any attempts to reduce some monolithic and free standing value will by nature require gross oversimplification and compromise. On its own this isn't cause for concern, but when these metrics are held with such reverence that they are used to determine a person's employment status and financial status, and consequently their quality of life; alarm bells should be ringing for any sensible individual.

Additionally, there's biases in any form of ”productivity” metric against minority members. Both specifically crafted models based on volume of commits, code complexity, commit regularity, etc, and through machine learning and AI based approaches frequently punish minority members. The former punishing those who may have an ”atypical” workflow or ability, such as negatively scoring disabled engineers who may have a reduced working capacity due to having to also manage a disability (and trust me, managing a disability is a job in and of itself). With the latter similar issues arise, especially given how homogeneous training data sets tend to be for machine learning, which leads to incidents similar to when *Amazon* tried to filter job applications with AI, which learned to filter out predominantly women who applied for positions (CITE THIS). I am of the belief that any devisable model for measuring the ”productivity” of a software engineer could never fully encapsulate the scope of individual engineers' experiences and lives, and unfortunately that will guarantee that there will always be individuals who fall between the cracks and receive negative productivity scores solely for matters outside their own control.

Furthermore, these systems of productivity measurement can create an all round negative atmosphere within a workplace for all employees. Having to work with management constantly lurking over your shoulder, with every single piece of code you contribute to a project and the specifics of when, where and how you created it being analysed by management under the microscope provided by these tools. This is oppressive enough as is when the data being processed is solely in regards to one's produced code, but with multiple startups promising to expand their systems of measuring ”productivity” out into day to day communication between employees through monitoring online communication through the likes of *Slack* channels or through tracking employees through the physical spaces of workplaces (CITE THIS). These issues are compounded by the ranking or ”gamification” systems offered by many of these services, providing a cutesy leader-board of employee productivity, supposedly to encourage healthy competition; but in my eyes this wholly misses the point of a workplaces, a space that should ultimately be co-operative, not one where employees feel the need to compete against their coworkers at every given moment. Whilst these are all good things from the perspective of management and the rapid production of a product, there is a persistent threat that these systems of measurement will only succeed in turning workplaces into data driven panopticons.

This is all to say, we should be avoiding reducing an engineers value down to an easy to calculate, but utterly devoid of context, metric at all costs; and we should especially be in opposition to the use of these metrics during the process of hiring, promotion, wage allocation, job security, etc. No matter how much a startup wraps their employee management suite in slick, corporate buzzwords, any attempt and measuring an individual worker's worth will, at best, miss out on some key contexts of them as a person, and at worst, actively punish minority members in the workplace.

6 Conclusion

To tie this all up, I feel that there are multiple major flaws with the pursuit of a metric for productivity, and I think it is solely to the fact that what we consider ”productive” within a workplace, or society as a whole, is reduced to the ability to turn a profit for an employer. With all these discussions of a metric, the thought of worker health, equal pay, diversity in the workplace, etc, are all at best forgotten, and in many cases these are at direct odds with what it means to be ”productive” in a deeply capitalist society. When any organisation is considering rolling out any system of measuring an individual’s productivity, there should be immediate criticism and concern from anyone who will be subject to it.