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

CSU33012 Software Engineering
Measuring Software Engineering Report

Patrick O'Callaghan, Std# 19334526

December 19th , 2021

Contents

1. Introduction

Over the course of this report, I intend to discuss the various challenges associated with measuring software engineering. This report will be split into four distinct sections. Section one will consist of a discussion of the methods used in measuring engineering activity along with any findings I have during my research of the topic. Specifically, I intend to look at frequency of code commits as a method and the inferred productivity analysis from this method.

In the second section of this report, I will look at the infrastructure involved with measuring engineering activity. I intend to look at the various platforms that allow for the gathering of engineering activity data along with the calculations that they need to perform over large data sets in order to produce results. During my research for this section, I want to explore the different kinds of infrastructures available along with the tools necessary to analyse the data provided by the platforms which make up the infrastructure.

During the third section of this report, I will examine the different kinds of computation that are used to profile software engineer's performance. I want to look at the different kinds of computations that are used in the industry, such as simple counting and machine learning analysis. I also want to look at with the strengths and weaknesses associated with each individual technique.

Finally, in section four I will look at the ethical issues surrounding the processing of this type of personal data. I will discuss if it is ethical to perform analysis on this personal data, along with giving my own personal opinion on the topic.

2. Measuring Engineering activity

There are many different methods used in the world of productivity and activity analysis to produce a result based on the work of a software engineer. When taking the nature of the industry into consideration, it is obvious why this poses such a challenge to those tasked with measuring engineering activity. Software engineering is collaborative and creative, which means that the quality of the work is more important than the quantity of it. However, as shareholders and business executives require performance measurement analysis in order to make personnel decisions, the measuring of software engineering activity is viewed as a necessity in the working world. I intend to discuss frequency of code commits as an example of a method used to measure engineering activity, and then discuss my findings around the value of doing such analysis.

2.1 Frequency of Code Commits

When considering methods to measure engineering activity, the most straightforward method is the frequency of code commits. This involves simply looking at how often an engineer commits code to the project. This method seems to be the most intuitive, as it looks at how much a given engineer is working on a project. This method gives the person performing the analysis an idea of how much code the given engineer has produced for a project, and how often they are working on the project. This can then be used by the analyzer to compare engineers to each other based on how many code commits they have contributed to the project, indicating how much work that have done on it. This metric is relatively simple to collect and gives a quick and easy insight into how much work an engineer is doing.

2.2 Productivity

Programming is not a discipline that can be measured in this way. The sheer volume of code written or commits to a project cannot be used as a measure of productivity. As stated before, programming is a creative and collaborative process. The value of each line of code is not equal.

As a simple example, two programmers were asked to produce an algorithm to do some complex computation. The first programmer wrote the code for a simple main method, taking in user input and outputting the result of the algorithm, and committed five times. The second programmer wrote the actual algorithm which provided the functionality and did this in just two commits.

Overall, it is fair to say that the second programmer's contribution to the program was more important than the first. However, based on the metrics of frequency of code commits and on lines of code, the first programmer would be considered to have contributed more. This simple example is scalable to large projects, in which a more advanced programmer could theoretically write code in less commits compared to a less experienced programmer. In the eyes of these metrics this could lead the analyzer to believe that the less experienced programmer is more valuable to the company as they are committing code more frequently. This is a significant flaw to this method however it is not useless as it can provide a quick and easy insight into the amount of time spend on a project by a given engineer.

3 Infrastructure

When discussing the infrastructure around measuring software engineering, there are many different platforms that make it up. These platforms form the basic infrastructure allowing for the analysis of metrics and calculate desired statistics based on available data. These platforms include Trello, JIRA and GitHub. These platforms keep track of various metrics used in code analysis or provide functionality to organize projects into separate tasks, allowing for effective collaboration. The analyzer can then pass this data to programs like excel, MATLAB and Pandas in order to analyse the data. These programs are examples of what makes up the current infrastructure surrounding measuring engineering activity.

3.1 Platforms

There are many platforms available that record the metrics necessary for doing the kind of data analytics necessary to measure engineering activity. As mentioned above, Trello, JIRA and GitHub are popular examples of these.

Trello allows for the organization of projects and gives an insight into how far developed a section of a project is. It also keeps track of who has contributed to each piece of work and who is currently working on a given piece of work.

JIRA is a work management tool that allows for agile development and project management. It is the backbone for Atlassian Open DevOps. This is a very well known set of practices combining software development and IT. A lot of its aspects came from the idea of an agile development model. JIRA allows for the tracking of agile metrics which give an insight into the quality of a product and also help to track team performance.

GitHub is the largest code hosting platform in the world. It allows for anyone to view anyone else's public repositories and see who committed to them. The GitHub API can be used to access many different types of data related to productivity such as commit data or repo data. GitHub also provides simple version control through git, which allows for easy collaboration across teams of software engineers.

3.2 Analytical Tools

Examples of analytic tools used to perform the complex and computationally expensive calculations required to analyse the data provided on these platforms are excel, MATLAB and pandas.

Excel is a spreadsheet program which provides many functions to perform calculations on loaded data. The spreadsheets created by excel can be used to easily compare the performance of software engineers and it can be used to perform calculations on large datasets such as those found in performance analysis.

MATLAB is a tool which allows for the analysis of data and the production of graphs and mathematical models. It can be used in the analysis of a developer's performance in a project by creating graphs or models based on the data provided by the platforms discussed above.

Pandas is a python library built to analyse data in the Python programming language. It allows for the generation of charts and graphs to display relationships in the data. It is similar to excel in a way as it allows for the creation of similar results as seen in excel. However, pandas is faster than excel and much better for the visualization of charts and patterns. Pandas can do more than excel but excel is easier to use.

When combined with the discussed platforms, these tools provide a clear profile of an engineer and give a good insight into their activity on a project. The combination of platforms and tools create the infrastructure seen today for measuring engineering activity.

4 Computations to Measure Productivity

There are many different kinds of computations that can be done to measure software engineering. They can be used to create a profile for each engineer or give each engineer a score. These can then be used to determine how productive an engineer is, or to see how valuable they are to the company. There are many different techniques to do this, such as simple counting and machine learning analysis. Each of these techniques have their own advantages and disadvantages.

4.1 Techniques

Simple counting is relatively straightforward. It involves counting data for various metrics in order to create a score for an engineer. For example, the counting of commits or number of hours spent on a project can be used to build a productivity score for a given engineer. Simple counting could also take the number of hours spend working at a given time or on a given day to create a profile of an engineer. The analyzer could use this data to create a profile which gives an insight into the days of the week or the hours of the day in which an engineer is most productive.

There are many different machine learning algorithms which can be used to analyse software engineer data. Machine learning algorithms are programs which learn and adjust to enhance performance as they process more data. This means that the algorithm improves in correlation with the amount of data that they process, making them good for processing large software engineer data. They can be used to create a productivity score for a given engineer and constantly update the score as they receive more data. This allows for the creation of dynamic productivity scores, making it easy for the analyzer to see if the engineer is improving or regressing in their productivity.

Intelligence augmentation will soon be used to integrate machine learning into daily planning and productivity. This promises to simplify complex decisions and planning into straightforward decisions, allowing for tasks such as meeting planning or assignment of roles to be automated in the near future.

5 Ethics

There are many ethical issues with the collecting and processing of software engineering data. When processing personal data in a company or organization, it is vital to keep it secure as any leak could be damaging to both the individual and the company. There are many problems related to this collection and ethical questions that must be answered before the data can be collected.

5.1 The ethical decision behind processing personal data

The processing of personal data in a company or organization comes with an ethical decision. The question is if it is ethically correct to collect data on an individual inside a company and use the data to compare them to their peers. If this data was leaked it would lead to employees knowing all about their peers, along with the data associated with their work. This could lead to a big issue inside the company along with possible fallout with employees. The other issue is employees not wanting their performance to be monitored by their employers. It can lead to a lack of trust between team members, and it can add unnecessary stress to the employee. This negatively impacts both employer and employee as the employee's performance is directly related to the company's performance. However, if it is implemented correctly the processing and measuring of performance can be used as incentive for the employee to work harder for the company which is mutually beneficial.

The encryption of data is an important step to processing personal data ethically. When a company uses an employee's personal data to analyse their performance, it should be fair to assume that the company will encrypt the data to prevent against data leaks. This gives the employee confidence in the company and in the analysis. In a best case scenario, the data should not be saved at all, with only the final score or profile being stored for later analysis. This would benefit both the company and the employee. This would also help to prevent against data leaks and give the employee more security.

5.2 My Opinion

In my opinion, I believe that processing personal data in order to analyse performance is necessary. I think that although it poses ethical questions and creates moral dilemmas, it is required in order to make decision inside a company accurately and fairly. I think that the benefits of analyzing personal data within a company outweighs the negative effects that it can have when done incorrectly. It is clear to me that the power held by a company analyzing the data of its employees can be exploited, however I believe that if you are working for a company you must fully believe in them and have the confidence that this processing of personal data is done correctly and ethically. A mutual trust can be built between the employer and employee through the use of encryption or by not saving personal data, which would lead to the ethical processing of employee data. This would be both ethical and mutually beneficial, which is why I believe that processing personal data is necessary.

6 Discussion