Measuring Software Engineering Report

Cian Doyle

Contents

Introduction

Measuring an employee's productivity is a widespread practice in all industries and sectors of the economy. It provides employers the opportunity to measure and compare worker performance and try and extrapolate more efficient methods of labor so that costs can be reduced, and profits can be maximized. Famous examples of metrics used to measure worker productivity include the quotas placed on workers in heavy industries during the first five-year plan enacted by Stalin. These quotas helped to measure whether a worker can produce up to a certain metric. Depending on whether a worker surpasses their quotas they could be rewarded. The most famous example is Alexei Stakhanov who mined fourteen times the quota on the coal mine he was working at in a six-hour period and was awarded with "hero of socialist labor" in 1935. These quotas, though harsh and draconian, helped soviet industrial planners measure productivity of regions and workforces. Another famous example of measuring performance is the company of Enron. Enron was famed for having ruthless performance evaluation with all employees receiving a grade based on performance and once the end of the quarter came those at the lowest grade would be fired. Usually this amounted to about 10 percent of the company's workforce. This earned the company a reputation for being extremely efficient and highly motivated which in turn lead to its overvaluation at the stock market despite doubts about what the company did to make money resulting in the largest securities fraud in history. With over seventy-four billion dollars being invested in the company and lost once the fraud was revealed.

Based on the previous examples it can be concluded that measuring worker performance is seen as a big part of running an efficient and effective organization be it a state run mine or an energy company. In most industries it is easy to measure and follow worker performance. In steel mills it can be measured in tons of steel produced. In an accounting firm it can be measured by the size of the accounts being managed by a given accountant. In law firms it can be measured by the number of cases won or acquitted. Software engineering is different though. It can be hard to visualize what makes a good programmer. Is it lines of code written. Tests passed, products finished, customer reviews, etc. The answer is not clear as the previously mentioned methods collect data that may be useful in certain contexts and completely useless in others. This can result in misevaluations of workers' performance meaning that a more efficient worker may be dismissed over misinterpretation of their performance.

In the following report I will outline 4 main points regarding measuring software engineering activity.

1. Measuring Engineering Activity
2. platforms on which engineering activity data can be collected and processed
3. Computation that could be done over software engineering data
4. ethics and legal or moral issues

Part 1: Measuring Engineering Activity

As mentioned in the introduction there can be many ways to collect performance data on software engineers. Some methods are more effective than others and according to Abi Noda, Senior Product Manager at GitHub. Too many companies rely on the “flawed five”. This includes commits, lines of code, pull requests, velocity points and “impact”. The reason for these being called the flawed five is that many of these metrics can be gamed and deceived to give a false impression of how well a worker is performing.

No. Of code commits

This method is a nice example of a metric that an employer may think of initially as an effective way to measure performance. This has good qualities as it allows the employer to prioritize some level of activity for workers. Unfortunately, this only measures any level of activity which means that work could be completed at a terrible pace but still commitments are consistent.

Lines of code

This method is useful as it allows the employer to measure how much work a worker is doing, which improves upon the no. Of code commits method. The downside of this method though is that it can result in bloated code that runs poorly and is difficult to understand. Secondly this method is a poor way of measuring rate of completed as a worker can complete one large commit and game the system with his large commit but low productivity.

Pull requests

This method is good in some ways as it allows the employer to ensure release cadence and continuous delivery. Once again though, this method is a poor measure on its own as it doesn’t measure the amount of work that is to be completed by the person pulling and results in small pull requests.

Velocity points

Velocity points are good to a degree for forecasting progress for the future based on the work completed but is bad because sizing is done before work is completed not after and it undermines the usefulness of the estimation process.

“Impact”

Impact testing is a form of testing that is developed to be a way of measuring every metric involved in productivity measuring. This method's intention was to develop a universal system of scoring software developers so that there was a concrete measure of a software developers' performance. This metric ended up being a new interpretation of the lines of code metric with some additional criteria for increasing score. This metric is poor as it is too abstract for software developers to keep track of what is important to their managers. The method of getting a high score is not always clear and thus makes the employee disorientated and confused as to what the overall goal is. Without an obvious goal present it leads to a misinterpretation of a worker's productivity. This concludes the five flawed metrics now we can look at other not yet mentioned methods of measuring productivity.

Test cases passed

This method is relatively good for developers in being able to develop software that can work based on a series of inputs that is specified in a test file. This makes the process of developing software more straightforward and can be an invaluable tool for increasing productivity in an individual that is producing software. Test cases on the other hand are less prepared for being used as a metric in measuring productivity on their own. This is because it only gives you a measure of the quality of work done by the individual. It does not provide any information on the pace of the worker, nor does it give any indication of projections for future productivity. Furthermore, it can be abused by developing non comprehensive test cases to fool the metric which can lead to a misevaluation of a worker's productivity.

Reachable code

Just like the test cases, looking at what code is reachable gives the developer an idea of where an error or bug may lie in their code. It can also be useful for developing a small efficient code that relies less on specific cases and instead works in a general input case. This is useful for a developer because it helps them to achieve a higher quality of work. This is a poor metric for measuring worker productivity as it does not give very useful data. This method can be used to ensure a more efficient code that might have a lower running time but using this solely for measuring productivity is too short sighted and does not give a fair view of programmer's productivity.

The best method for measuring productivity

So far, the best method for measuring a worker's productivity is a mixture of all previous methods to a reasonable degree. Relying on one metric too much can lead to misevaluations. When using a mixture of all the previously mentioned metrics it allows for each metric to maximize the advantages of each metric and minimize the disadvantages. Using many metrics, management will be able to gather large amounts of data that can be used to form a strong picture of a measure of productivity. Secondly it is best to measure teams, not individuals, as the tasks that an individual may need to perform in a team can vary widely thus taking a measure of the team will be a more comprehensive measure of productivity.

To gather all this data a large apparatus of data gathering systems will need to be implemented so that all the individual software engineers can have every metric measured and stored so that the data can be interpreted later.

Part 2: platforms which engineering activity data is collected and processed

To be able to collect the data required for accurate comprehensive evaluation of software engineering activity, a system of data collection needs to be implemented. This system can include

GitHub (Cloud based applications for data storage)

Cloud based applications are applications that exist to store data in a data center which can be accessed by anyone with permission to access such data. These applications are used for a variety of purposes but most follow the same spirit providing a central database for information that can accessed and edited by anyone with the appropriate permissions. Usually, the data on who is responsible for an edits and uploading is recorded by the cloud platform. Cloud platforms are used by nearly all medium to large scale businesses in some capacity.

The first technology we can use to collect data on software engineers is GitHub. GitHub is a cloud-based hosting service that performs version control for a team of developers looking for a place to perform version control. This platform is used universally in the software engineering industry. On GitHub a project can be stored and pulled from. The work can be branched from a master branch so that modifications to the project can be made without disrupting the entire project for the team. When new updates are ready the work is committed to the branch and when the branch is no longer needed ideally the branch is merged with the master making the new updated changes part of the master branch. as commits are made the data on who and when it was committed is stored as part of the repository and can be reviewed by anyone who has permission to see the repository.

This platform allows those who are responsible for performance evaluation to see the direct influence that a given member of staff has contributed to a project. This can be done by looking at the number of commits of the individual in question and the quality of commits. This data is stored permanently in the GitHub repository forever and is never deleted.

Deployment of mobile infrastructure

The mobile phone has enabled those who are conducting performance analysis to be able to collect a vast amount of data on individuals who are analyzing. The mass adoption by the entire population (excluding several sub-Saharan countries) has allowed businesses and organizations to deploy phone-based applications that are able to interact with their employees and collect data on them with over 65% of companies using at least one social tool on mobile phones.

Amazon web services (for computing purposes)

Cloud computing has provided a platform for companies to process large amounts of data and extrapolate conclusions based on that data. Cloud computing is based on the concept of users being able outsource their computing needs to companies that can compute the data for them using computers in a data center. This setup is good as it highly flexible and does not require the users to commit to an investment into their own high-performance computing (HPC) cluster. It is also good as the centralization of computing power allows for a more efficient use of computing power. This means HPC computers spend less time idle. Finally due to the centralized nature of the service this means that the service is highly scalable and the user can simply request more computing power for their purposes and when the host company wants to expand their capacity, they are free to add onto the existing infrastructure. This helps to alleviate the problem of computers that were once considered high powered becoming obsolete and becoming useless to the user.

A good example of such a service is the Amazon web services. This service allows customers to purchase computing power and data storage on a pay as you go basis. This service has been highly successful and is now becoming a platform for high power computing that can be purchased as it is needed without a large initial investment being made. This platform provided the basis for companies to be able to process the vast amounts that have been collected by methods mentioned in part 1.

The advances in algorithms related to big data processing

In the years following the emergence of big data the development of new algorithms or the adaptation of existing algorithms to efficiently sort and analyze this data this includes, K-Means Clustering, Association Rule Mining Algorithm and Linear Regression. These algorithms are but a few examples of the algorithms that are used to process large data sets. These algorithms are used to extrapolate conclusions on the given data within a confidence interval and rely heavily on statistical analysis. The benefits of Big Data include the fact that it provides large sample sizes but its advantage is also a disadvantage as such large amounts of data can be difficult to interpret. This means that the use of computing power to process this data in a timely manner is essential. Due to the development of such algorithms means we are able to achieve this. Thanks to this development, employers and managers can use these algorithms to process the large amounts of data that is collected on employees.

The expansion of computer infrastructure to collect and process large data sets

Over the last 25 years computer infrastructure has been expanded to be able connect and communicate with more computers on the networks that it is connected to. This expansion has allowed for larger amounts of data to be able to be sent from computers on the network to a given central location. This has allowed a great deal of data to be able to be transmitted and collected. This gives employers an ability to collect the data that they need to establish an effective performance analysis of their employees as established in part 1.

Part 3: Computation that could be done over software engineering data

Defeasible argumentation (expert systems)

The definition of defeasible is “open in principle to revision, valid objection, forfeiture, or annulment” and defeasible argumentation is an argument that is rationally compelling but not deductively feasible. An example of a defeasible argument is “the sun will come up tomorrow”, although rationally compelling and taken as fact this statement does not hold up when subject to deduction. There is no way of proving that the sun will come up tomorrow despite us taking for granted the idea that the sun will come up tomorrow because it would be rational to believe that sun will come up tomorrow.

Defeasible argumentation can be used in developing expert systems. Expert systems are systems that take the knowledge and experience of a professional and translate it into a software system that can take the place of the expert system. Conceptually the system has 3 main parts. First the knowledge is translated into a knowledge base. Then an inference engine that mimics the behavior of a defeasible argument will use the knowledge to be able to form its own ideas and directions based on the input and data it receives.

Supervised learning algorithms (Neural networks)

A supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. Learning algorithms are very good at extrapolating behaviors and learning to produce appropriate outputs based on a set of inputs.

Neural networks are systems that analyze a large data set of inputs and outputs and form a method replicating the outputs based on the inputs. A typical neural network consists of an input layer where input data exists, an output layer where the goal exists and a hidden layer where the path from the input to the output is mapped through. This acts similarly to the human brain. Our brain can see inputs and outputs and then extrapolate a given path or method to make the inputs into outputs.

Neural networks can be used to develop a system that can predict a team's progress based on their current progress. This analysis can be used to take a clear picture of the team's progress and give a projection of the team's finish time. Using the data, a manager can make more efficient use of resources when it knows what teams are struggling.

Unsupervised Learning algorithms (K means clustering)

Unsupervised learning algorithms are learning algorithms that detect patterns in data sets. They do not need to be provided with a given output to produce a conclusion.

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

This helps to inform management about the habits and patterns of their employees and teams working for them. This results in a more well-informed decision given the data that has been collected in part 1.

Part 4: ethics and legal or moral issues

Privacy

Privacy is the primary concern when you consider the volume of data being extracted from the software engineering workforce. Such in depth scrutinization of a person's work can lead to massive breaches of privacy. Never in human history has a management structure been given the ability to scrutinize every employee in their employment to such a large extent.

Privacy is a right afforded to everyone according to the United Nations Universal declaration of Human rights. Despite these rights not being backed by and form of concrete and widely respected authority it does set a precedent for some degree of protection against infringement of privacy. Privacy of the workers needs to be a responsibility of the employer. Sadly, due to the nature of private enterprise, prioritizing privacy is not always the most profitable course of action for a company to take. When harvesting large amounts of data from a workforce looks like it may be profitable then usually management is encouraged to pursue this course of action by the shareholders and the board of directors in major companies.

Labour rights

Workers are members of an organization that trades time and skills for money so that they can use the money to continue living and providing for themselves and their families. Privacy outside of the work environment was never a major concern of labor rights movements as there was no widespread method of extracting data on a worker after they had left the workplace. Now that frameworks for extracting and processing this data exist, it can lead to organizations profiting from the data that workers provide to their data collection. It can be argued that a company can be entitled to the data on working patterns of its employees due to the work being done on company time. This sets a poor example though since such a concession could be seen as the groundwork for further extraction of data that a company or organization is not morally entitled to.

This is a major problem as there is very little labor laws being routinely enforced as it is so adding another will be ineffective. For example, workers in the Irish construction industry have a strong dislike for unions due to the employees that are found to have joined them being fired from their jobs despite this being in blatant breach of labor laws. This overlook of labor laws is due to preferential treatment of employers over employees. Even though privacy is a right afforded to employees there is very little a worker can do to combat an inappropriate encroachment by management into someone else's private life.

Possible solutions

Possible solutions to this problem are a reinforcement of labor laws so that workers can feel empowered to challenge management on an issue which can include privacy. When a worker feels powerless and disrespected in their employment, they can feel alienated and fed up with management leading to resentment and poor performance overall. Even though management should know this by now due to the large proportion of reasons for leaving a job include poor management, management continues to persist in this manner.

Secondly the signing of proposals to bar companies from extracting data from their employees without their consent. Breaking these rules will result in a substantial fine. This will go a long way in helping discourage a company from unlawfully extracting data from their employees. This is because the bottom line is the only thing that can manipulate a private company.

Lastly is the employment of managers that are from within a company. This will give the manager a familiarity with the employees. This comes with the added benefit of having instant trust between workers and managers due to seeing the relationship less as an “us and them” structure and instead as a fellow programmer who understands the importance of the privacy of software engineers.

Conclusion

In conclusion, measuring software engineering activity is far more difficult than measuring the progress of traditional industries. Despite this, measuring activity is needed to be able to run a large software engineering company when management is not intimately aware of every employee in their staff. The methods that a company can use are numerous and flawed on their own. When combined with each other and observed in the proper context it can provide a relatively good picture of a software engineer's progress and productivity. There are many new technologies and frameworks that allow companies to be able to extract this data and form this picture. Extracting this data can be a slippery slope paved with opportunities for abuse of privacy. This can lead to workers feeling that their privacy has been breached. There are solutions to these problems that may be difficult to implement but they need to be if workplaces intend to continue a positive relationship between management and employees.