Measuring Software Engineering

"To deliver a report that considers the ways in which the software engineering process can be measured and assessed in terms of measurable data, an overview of the computational platforms available to perform this work, the algorithmic approaches available, and the ethics concerns surrounding this kind of analytics."

Introduction

The correct or most efficient method for measuring software engineering seems ambiguous. Many people offer differing viewpoints on this topic. Nowadays, a good software engineer leaves a significantly large data footprint compared to software engineers of the past. With the introduction of version control software such as GitHub and Bitbucket, It's easy to see the work and experience of a software engineer over the years. With this information at hand we can measure the skills of each software engineer on GitHub. LinkedIn now provides a detailed source of information on past work experience and skills of software engineers. Detailed data analysis and workplace analytics can be used to measure the effectiveness of some software engineering practices such as the use of an Agile environment, different lengths of and workloads for sprints. As you can see there are many different approaches which can be taken when measuring software engineering.

How can Software Engineering be measured?

Software engineering is measured for many reasons, the main being to improve software productivity. Software productivity is the ratio between the software produced and the expense which went into producing it. There are a range of software productivity tools including Timular, Beeminder and Toggle which can help with these measurements. Each stage of the Software Development Lifecycle can and should be measured in order to boost productivity. Software metrics measure the characteristics which are quantifiable or countable.

Measurable Data

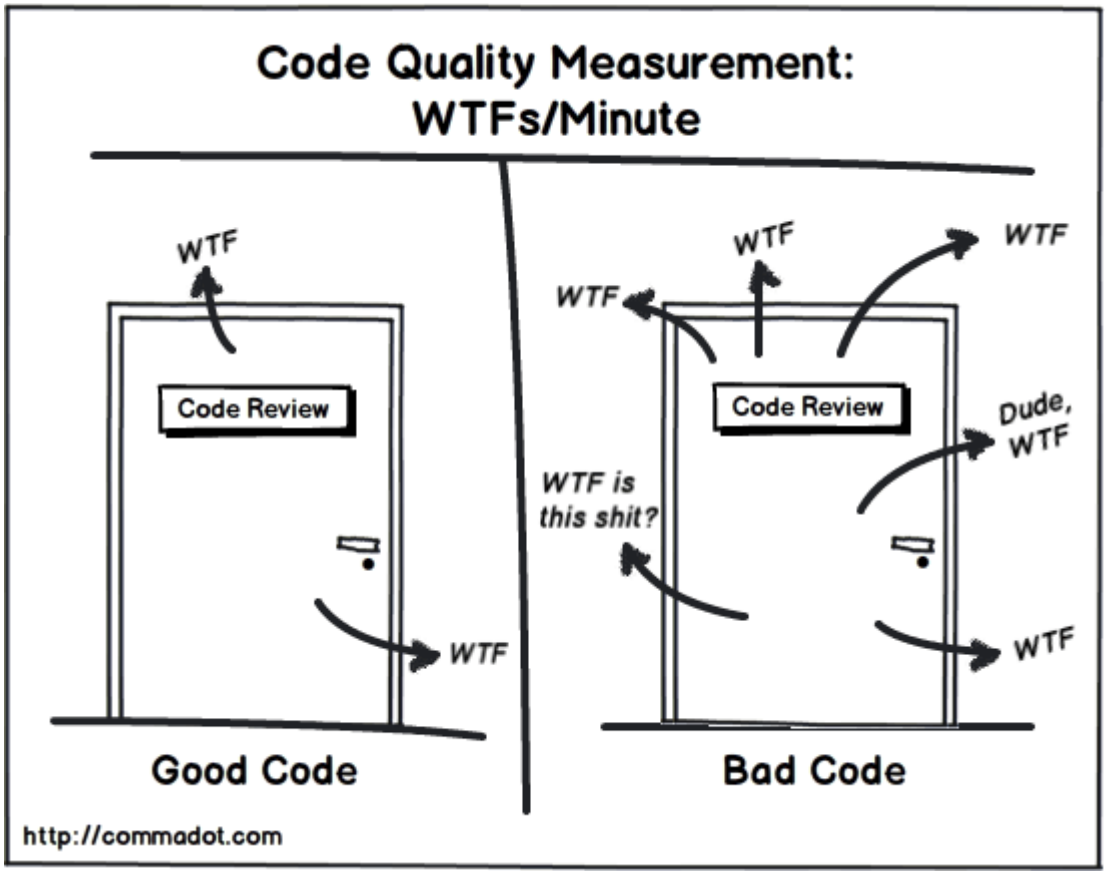
Code

Many would think the primary measurable factor of Software Engineering is the measurement of code written. This may not be the best place to look when seeking productivity. Using the quote from Bill Gates "*Measuring programming progress* by lines of code is like measuring aircraft building progress by weight.", we can assume, this should not be a measurement solely based on the amount of code written. The SLOC software metric counts the number of lines of code with text. LLOC is another metric which counts the number of logical statements. SLOC is considered bad practice as it can discourage programmers from writing efficient code.

Some coding tasks can be far more complex than others which must be taken into account. Measurement of code should be based on quality rather than quantity. The quality of the code written impacts the quality of the software produced. In general, high quality code has good readability, performs what and how it is meant to, is well tested and consistent. *Coding standards* should be specified to maintain consistency.

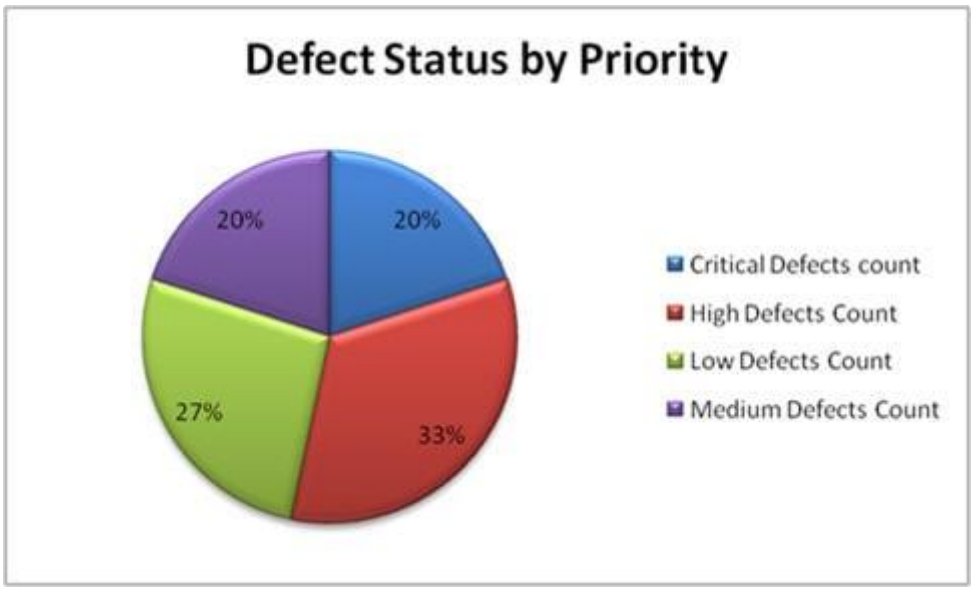
Code Reviews

Code reviews also allow time for pointers and corrections in code written by engineers. Code reviews are around hour long events which can help optimize code written by developers. Defects found in the code can be used to calculate the effectiveness of the code review.



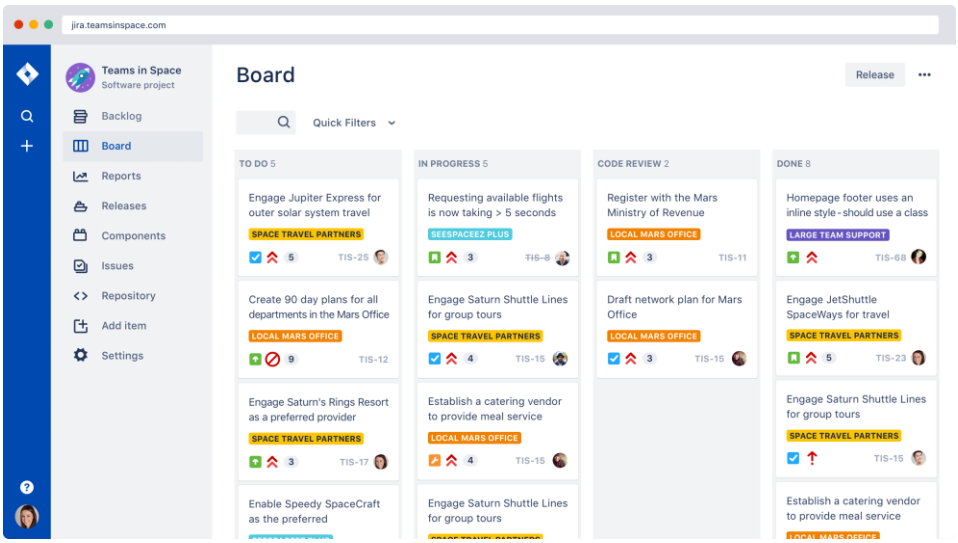
Defect Metrics

Defect metrics is a measurement of code quality. This includes measuring the number of defects within the code, how each defect originates, the place where the defect occurs and the severity of each. This can help show any bugs which are present in a software. In the example diagram below, there is a high percentage of Critical and High defects. This is a good way to flag these defects which need to be tended to quickly.



Task Tracking

In my own experience using a task tracker such as JIRA can be helpful. Including time estimations before attempting tasks and logging time taken as you work can help you as an developer better understand your limits. It can provide you with the necessary information to better predict the timeline of your future tasks. Kanban boards can provide a visual representation of a software projects status. Engineers may also see the status of their co-workers tasks too which can give a better overall understanding of where the project is at in terms of how many tasks remain on the backlog or if there are any bugs or issues still relevant.



Workplace Metrics

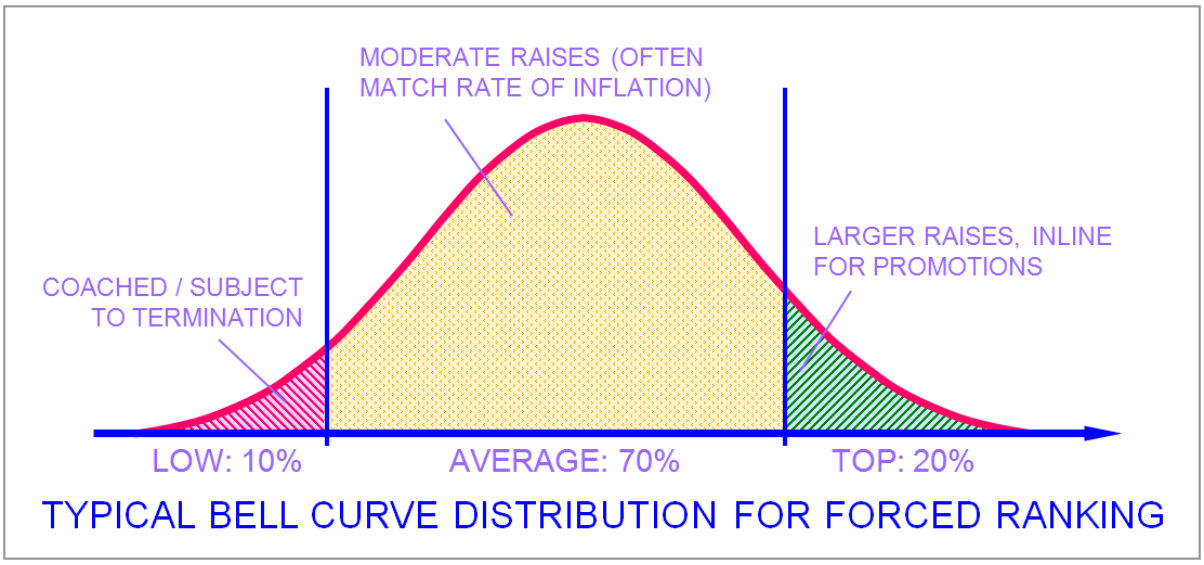
There are tonnes of different metrics used in the workplace all used to measure work quality, quantity of work and efficiency.

Workplace communication

Workplace communication is a factor which can majorly affect the productivity of a working team and in my experience, especially so for software development teams. Since COVID and the introduction for many people to working from home, this factor plays as vital a role as ever in software productivity. Not having the opportunity to ping questions to co-workers for instant responses is a noticeable disadvantage to being in-office. This can lead to having problems or errors last longer than they would have in-office as the environment is a little less integrated. There are however many ways for employers to track their developer's communication. Microsoft Teams gathers meeting data such as number of hours each person has spent in meetings during work hours and after work hours. Collaboration hours with other developers, emails sent, IM's sent, hours spent on an IM service and averages for all developers are different types of data which are collected. Using this data, an employer can create a visual representation of their developers and see what they are good at and where they may be lacking. Employee KPI's may show where improvements can be made.

Forced Ranking

Forced Ranking of developers is a new trend which has led to big improvements in productivity. All developers are evaluated based on performance and ranked. This seems to be an unethical approach however and could lead to stress and mental health detriments caused by the work environment. It can however increase the drive of some workers to rank higher than their colleagues. Some companies fire the bottom 10% of this ranking and hire top applicants which leads to increased productivity. Developers higher in the rankings would be closer to a promotion or raise.



Health

It's not surprising nowadays that the *health* of a workplaces developers is measured. Improving a developer's health can lead to an improvement in productivity. Poor posture can affect a developer's performance, so companies may choose to provide ergonomic chairs for developers to increase their comfort and performance. Physical health is just one aspect which can be measured to do with health. Mental health plays a hugely significant role in a developer's activity and focus. Poor mental health can lead to absenteeism, it can generally have a negative effect on developer productivity by lowering developer morale. Therefore, it is in the employer's best interest to support and help their developers look after their mental health.

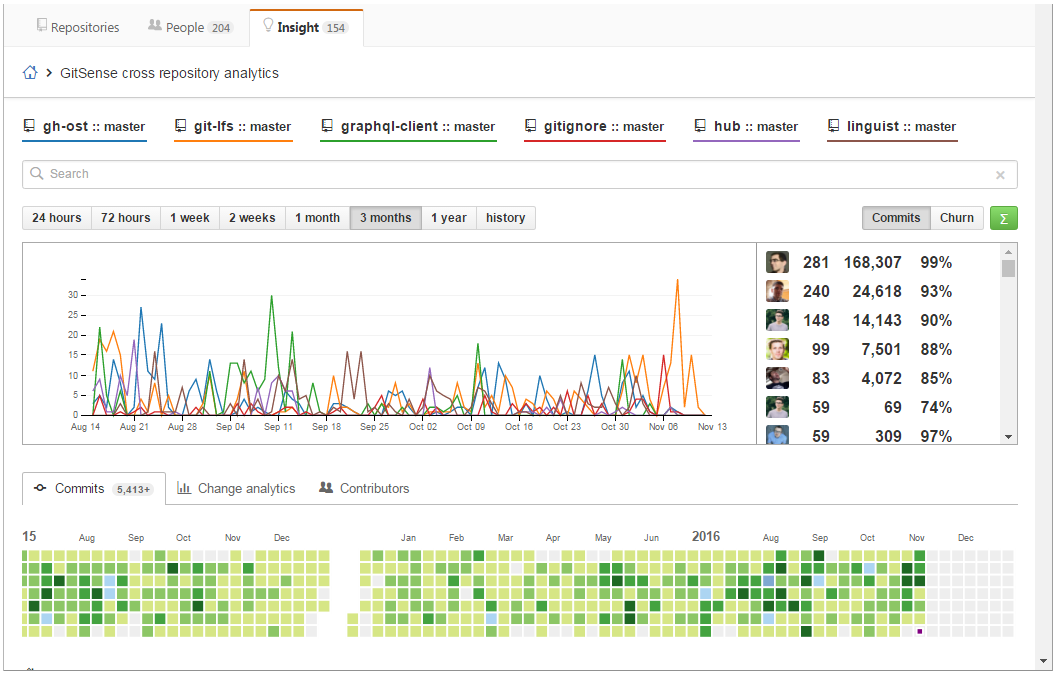
What platforms can be used to gather and process data?

It is almost becoming second nature that companies are collecting data from workplace activities. Most of these tools have different approaches to measurement of workplace activity such time tracking, visualization tools, code tracking, monitoring of computer usage and task trackers. From company to company the use of tools can vary. Microsoft use their own tool Microsoft Workplace Analytics to track their workers activity.

Git

Git is an open-source version control system for tracking changes in any set of files. It is the most common version control system. Git runs locally but can be connection to online hosts such as GitHub as discussed earlier. It increases the ease of collaboration with other developers. It also allows more than one developer to be working on a piece of code at once by merging changes automatically. It tracks all changes made to a project which can make painting a picture of the entire process a lot easier as well as maintaining versions for

rollbacks. Git is extremely easy to use and is integrated into most IDE's. Git keeps track of all commits(changes) made by developers. This can be used to measure a developers contribution to a project by measuring the frequency and size of their commits. GitHub provides good visualization of this.

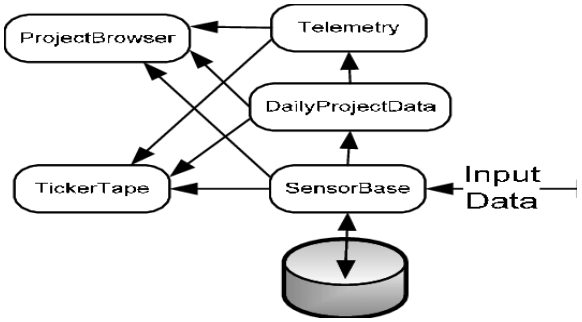


PSP

Personal Software Process is a framework designed by the Software Engineering Institute (SEI) which helps software engineers to find a way of measuring and improving the way they work. It helps them in developing their respective skills at a personal level and introduces them to the ways of planning, estimations against the plans and quality control methods. Result from the PSP training have shown in increase in the use of disciplined methods as well as productivity and quality improvements. It aims to reduce the number of faults in their work. They are not industry standard however.

Hackystat

Hackystat is an open source framework for collection, analysis, visualization, interpretation, annotation, and dissemination of software development process and product data. It allows you to attach software plugins(sensors) to your development tools which unobtrusively collects and sends data about your actions to the Hackystat SensorBase, which they use for storage. The SensorBase can then be queried which can be used to create a visual of the raw data. Google charts display the data in a graphical format. Hackystat is a flexible software and users have complete control on what they wish to analyse. There are no limits with this open source software, its goal is to facilitate 'collective intelligence' in software development.



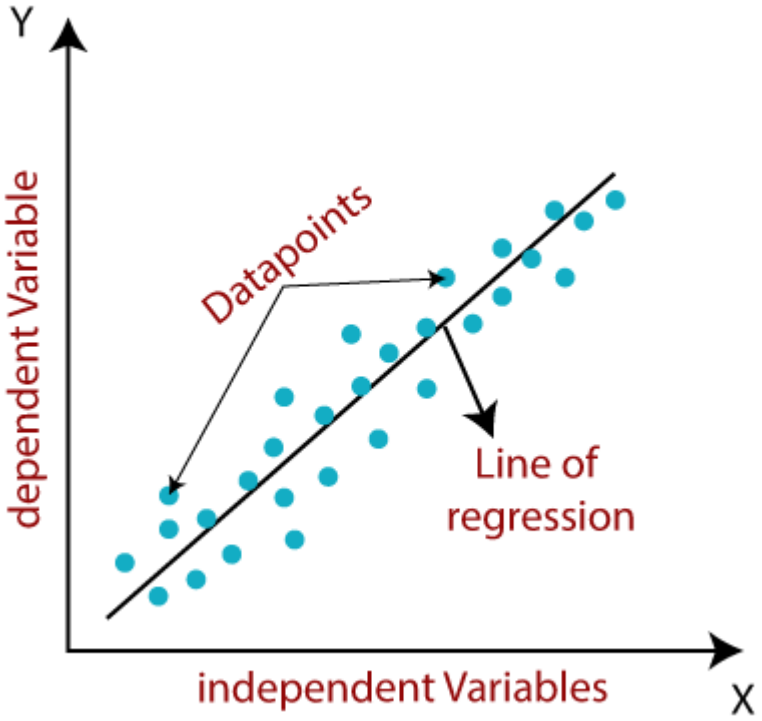
What algorithms can we use?

Artificial Intelligence now has a significant impact on the world and its impact is rapidly growing. An ever-growing chunk of data is being generated from different sources and applications daily. AI/Machine Learning systems can learn from this data and perform tasks. There are three main types of machine learning: Supervised, Unsupervised and Reinforcement learning.

For *Supervised learning*, the data used for training consists of inputs matched to correct outputs. Once the training is complete the algorithm takes unseen before inputs and tries to determine the correct output based on its prior training. It deals with the labelled data. Supervised learning consists of two main groups: regression and classification. Classification is when the output is a category, such as “green” and “black”. Regression is when the output is a real number/value, examples include “euro” or “height”.

Support Vector Machine is a classification algorithm used to find a hyperplane. The machine support uses data points and outputs a hyperplane which is a line which separates all points. This splits the dataset into two classifications, one on one side of the line and vice versa. The hyperplane is chosen as from points on both sides where the distance to the hyperplane is the largest.

Linear Regression is an example of a popular Supervised learning algorithm. It is a statistical method that is used for predictive analysis. The linear regression model is the graph generated through using this algorithm and it shows a sloped line called the line of regression as the relationship between the variables. Linear regression uses a linear relationship between dependent and independent variables. The algorithm predicts real or numeric variables for example as sales, price of a product and salary.



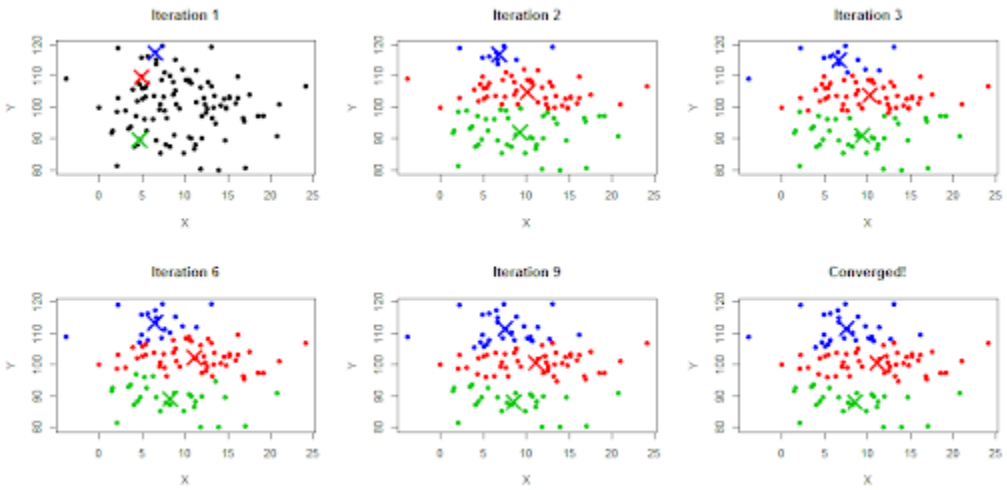
For Unsupervised learning users do not need to supervise the model. Instead, it allows the algorithm to work alone to uncover information and patterns which were not previously detected. It mainly deals with the unlabelled data. Unsupervised learning consists of two groups: association and clustering.

Clusters are groups of data points joined together because they have similarities.

There are different types of clustering which can be utilized: Agglomerative which groups each data into a cluster and forms unions between near clusters, reducing the amount of clusters, Exclusive which groups data in one cluster only, Probabilistic which uses probability distribution to find clusters and Overlapping which uses fuzzy sets to cluster the data. Data can belong to two or more clusters with different degrees.

Hierarchical clustering is a method which creates a hierarchy of clusters. Each datapoint is assigned to their own datapoint initially and close clusters are moved to the same cluster. The algorithm continues until there is one cluster remaining.

K Means clustering is an example of an Unsupervised Exclusive clustering algorithm. This is an iterative method which finds the highest value for each iteration. A value for k must be selected initially which represents the number of clusters wanted. The higher the value for k the smaller each cluster will be. The larger the value for k the bigger each cluster will be. Each cluster defines a centroid which is the central piece of data for that cluster. Each datapoint is moved to a cluster group connected to this centroid. This algorithm is used to find types of groups which are unknown or exist in complex data sets.



Reinforcement Learning is training for the machine model which teaches it how to make decisions and take action to maximize reward. For this no training data is used. Its goal is to determine optimized behaviour for the machine for certain situations. The machine uses trial and error to figure the best route through certain scenarios. Once set up, no further input is required for the algorithm.

Is this ethical? Is it reasonable?

Many believe employee monitoring software is intrusive and excessive. In my opinion it can be performed in an ethical manner. Secretly monitoring employees is undoubtably unethical and sometimes even illegal. Employees must be aware of any software being used to monitor them and must agree to being monitored.

The employee's privacy must be considered the utmost importance as it is private information being collected. Companies must be careful not to break any laws as they can find themselves in a legal battle swiftly. Monitoring employee health I would consider to be unethical but can be legal when the employee agrees. Fitbit allows the tracking of sleep and exercise. I do not believe companies should have the right to decide and guide their employees towards certain lifestyle choices. Screen-recording, screen-shotting and monitoring keystrokes can lead to the exposing of sensitive details such as passwords or leaking of bank details if the employee is on his banking website. These are pieces of data which companies should look to avoid as it is a breach. GDPR has become extremely influential in making companies compliant in their monitoring of employees with fines being introduced.

The data which companies are attempting to mine from monitoring their employees can be extremely valuable however and can help to find any workers performing inefficiently. Improvement actions can be taken at that point. Some data can be misleading however if we consider the SLOC approach. Developers can take advantage of this and make it seem like they have written more code than they have. I believe any data monitoring should be

discussed and agreed upon by both the employer and the employee with the employee having the final say on what is being monitored.

Conclusion

To conclude, measurement of software engineering can be complex and sometimes abstract. When done right, it benefits the hard-working employees and can point out some stragglers. The developments of AI have led to an immense improvement in the field of software engineering and how it can be measured.

It may be alienating for some, but it can accurately depict the level of productivity in the workplace and show where improvements will come. I think employees should be given clear warnings on the findings of their activity monitoring and given the option to improve before they are punished as the details can be hard to see. There are many tools which help the employer and developer communicate for this purpose.

In today's world data monitoring seems to be intertwined with the future of software engineering as the benefits and improvements made from interpreting this data intuitively are clear to be seen. It would be silly for companies to not look to improve productivity and efficiency through employee data monitoring. There must however be regulations in place which can safeguard the privacy of the developer. Without these safeguard's employees could be susceptible to stress and straining of their mental health.

Bibliography

https://www.perforce.com/blog/sca/what-code-quality-and-how-improve-code-quality

https://techbeacon.com/app-dev-testing/9-metrics-can-make-difference-todays-software-development-teams

https://www.sealights.io/agile-testing/test-metrics/

https://www.softwaretestinghelp.com/software-test-metrics-and-measurements/

https://connecteam.com/how-to-measure-communication-in-the-workplace/

https://docs.microsoft.com/en-us/workplace-analytics/use/metric-definitions

https://www.analyticsinhr.com/blog/employee-performance-metrics/

https://www.glowtouch.com/best-practices-and-metrics-for-code-review/

https://stackify.com/track-software-metrics/

https://www.youtube.com/watch?v=NrLFIpm0wps

https://hackystat.github.io/

https://www.nobledesktop.com/blog/what-is-git-and-why-should-you-use-it

https://gdpr.eu/what-is-gdpr/

https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

https://www.geeksforgeeks.org/what-is-reinforcement-learning/

https://www.youtube.com/watch?v=h5CRzj3z75k

https://www.youtube.com/watch?v=Dp5_1QPLps0

http://patentimages.storage.googleapis.com/pdfs/US20130275187.pdf

https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/

Written By: Killian Ronan

Student ID: 18328687