MEASURING SOFTWARE ENGINEERING

Tom McManus

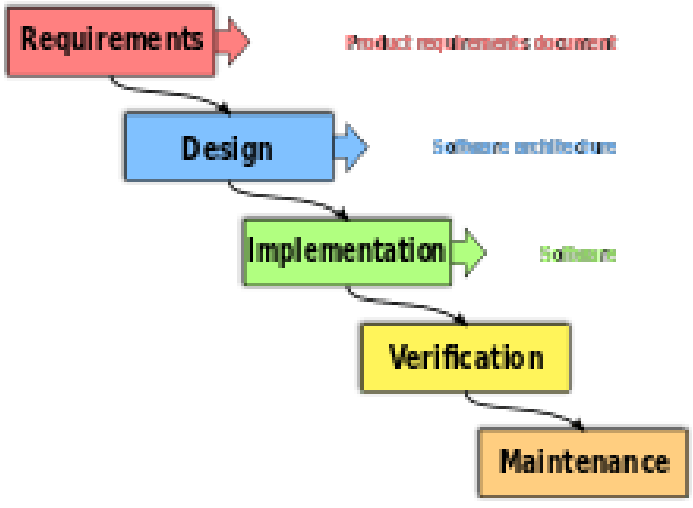
18323885

CSU33012

Introduction

In the following report I will be looking into how software engineering as a process can be measured and assessed. I will be focusing on the following four aspects in this report.

1. Methods used for measuring data
2. The platforms available to perform this work
3. The different algorithmic approaches available for software engineers
4. The ethics concerns surrounding this kind of analytics



Before I begin, I will start by explaining what Software Engineering is. Software engineering is the design, maintenance and development of software. It is all about keeping software efficient and up to date. There are methods used to ensure standards are kept up to date. They follow multiple steps in designing software starting with analysing and planning the software. This is essentially deciding on what the software will need to do and how it will be implemented. The following phase is the design phase. This is laying out the blueprint they will follow when they

start the implementation. It is essentially planning how they will approach the project. It is then implemented in the next phase. This is the coding part of the project and where they will actually have the software up and running. It is then followed by the next phase of testing. This is ensuring that the software runs without error and can handle any issues that could occur. The final phase is maintaining the software and dealing with any problems that may occur later on. It is also about developing the software. Linus Torvald created Linux, a softeware used masssively in the industry, in 1991 and he is still maintaing it today. These steps are followed to ensure that the software developed is as efficient as possible.

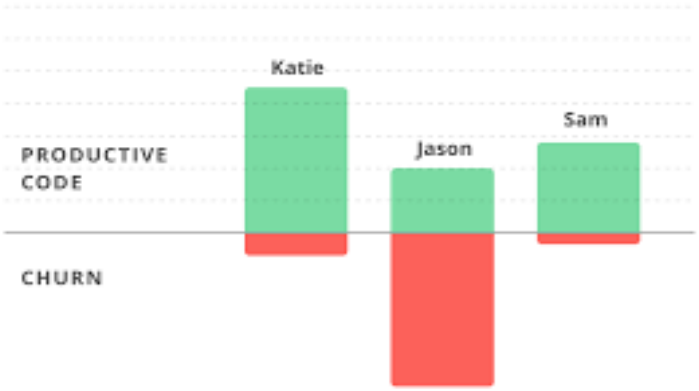
Measuring Data

Recently measuring the data has become increasingly important to assess the efficiency and reliability of the code. There are several different methods for measuring the data and all the methods approach it in a different way. The following points are all viable ways of measuring code.

- Code churning
- Number of lines
- Test coverage
- Amount of bugs
- Velocity
- Lead time

Code Churning

Code churning is looking at the productive code and comparing it to the amount of code written. It is a prime example of the famous saying “Quality over quantity”. Many developers will rewrite their code several times and use their best one. The code churning method is used for measuring data as instead of looking at who has written the most it looks at who has written the most productive and useful code.



The diagram above shows why churning is a useful method for measuring data. While Jason wrote the most code by far the majority of it wasn't used and is hence "churn". Sam was the opposite. He wrote the least code but almost all of it was used.

Number of Lines

Looking at the number of lines written, or time spent on the project is also a metric for measuring the process. This metric is often called LOC standing for lines of code. Personally, I believe that this method is not a very efficient method for measuring but is still sometimes used. It is essentially just looking at how many lines they have written or how much time they spent on the project. Many long pieces of code can be enhanced and shorten so this method for measuring a software engineering process so this method, in my opinion, is flawed. It does however show that there is lots of commitment from the engineer and he is working hard on the project.

Test Coverage

Another method used for measuring software engineering processes is seeing the test coverage. This is very measurable data so is useful. It is a good method for checking a process as it shows the parts of the code that is done correctly. In my opinion, the most important part in coding is that it is functional and can handle problems. Therefore, I think this is one of the best methods. This method says how well the code works and that it has been tested to deal with problems. It is pretty much measuring how well the code works which is very important and is a good method for measuring a software engineering process.

Amount of bugs

Counting the number of bugs that are reported is also a metric for software engineering processes. It is simple and can work due to the fact that bugs are what can ruin a piece of code.

However, there are also problems with this method. Counting the bugs can provide a false picture of a project. It is possible that a superior piece of code could be seen as worse just because it is more “buggy”.

Velocity

The time it takes to do the work is a metric used sometimes. Also referred to as the velocity, it is the time taken to do a project or the amount of work done in a certain time period. It is not very useful to compare teams’ times though as its based on non-objective estimates.

Lead time

The lead time is the time taken for the whole project to be complete. From when the software engineer starts the project up until the finished product is ready. It is an ineffective way to measure as it doesn’t take into account the quality of the code and also projects can have different lengths and difficulties. It is used more to compare it with the estimated lead time for the specific project.

Platforms

Many companies want to optimize their workers output and so there has been more demand for software engineering measurement systems. They are quite expensive due to the high complexities of them. They are useful for optimizing work patterns. Three examples of these platforms are

- GitHub
- GitPrime
- Personal Software Process
- Semantic Designs

GitHub

GitHub is possibly one of the most useful and well-known pieces of software in the world with over 40 million users as of January 2020. It allows users to collaborate on projects. It makes use of the Git. It allows users the option for source code management and distributed version control. It is a subsidiary of Microsoft. It not only makes use of the features of Git but also offers its own services such as bug tracking and task management. It allows users who have access to a repository the option to commit, pus and pull code from the repository.

GitPrime

GitPrime is used more for measing the software engineering processes. It makes use of GitHub and analyses performance. It analyses data from any git based repositay service such as GitHub, BitBucket and GitLab. It is very useful for companies as it allows them to optimize work patterns and hence increase the speed and efficiency of work.

Personal Software Process (PSP)

Personal Software Process was developed to assist software engineers in finding methods to measure and improve their way of working. It is made up of four levels. PSP 0 is the first level and is made up of personal measurement, size measures and coding standards. PSP 1 is about the planning time and scheduling. PSP 2 brings in personal quality management, design and code reviews. PSP 3 is the final stage and is for personal process evolution. It require manual entry into the system.

Semantic Designs

Semantic Designs is one of the global leaders for services to enhance the development of software systems. It offers an option outside of Git for analysing processes. It uses language specific metrics for several supported languages. It analyses the code and produces measurements from the data that it retrieves. It can be used on large scale projects which makes it useful to large companies. They are popular because it allows the creation of custom metrics for different modes.

Algorithmic Approaches

The metrics that are collected can be clearer when put into certain algorithms and can show where a process' downfalls occur by calculating specific attributes. The systems are now highly sophisticated and detailed. The number of different metrics they measure is as large as it has ever been now. Some examples of these algorithms are below:

- Machine Learning
- Principal Component Analysis
- K means clustering

Machine Learning

Machine learning is an application of artificial intelligence that allows systems to automatically learn and improve without being programmed to do so. It is trying to get computers to act as humans do and learn from what they are doing. It is divided into three sections:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning

This method applies what has been learned in the past to new data using past examples to predict future events. It is called supervised because it is trained to have a process of dealing with new input. It uses its past example to give the output desired. It is given inputs and known outputs at first which is the training. It then uses the link between these two to predict the output of other inputs with unknown outputs.

Linear Discriminant Analysis

Linear Discriminant Analysis is an algorithm that falls under the supervised learning category. It is a method that finds a linear combination of features that characterizes or separates data. It uses the knowledge of the labelled data to group the unlabeled data. This algorithm looks for a linear combination of the variables in the data which best explains the dataset.

Unsupervised Learning

Unsupervised Learning is similar to supervised learning, but it is not given the training. The sample data used to test has unknown output. It studies how systems can infer a function to show a hidden structure from unlabeled data. It does not figure out the correct output but is used to describe hidden structures.

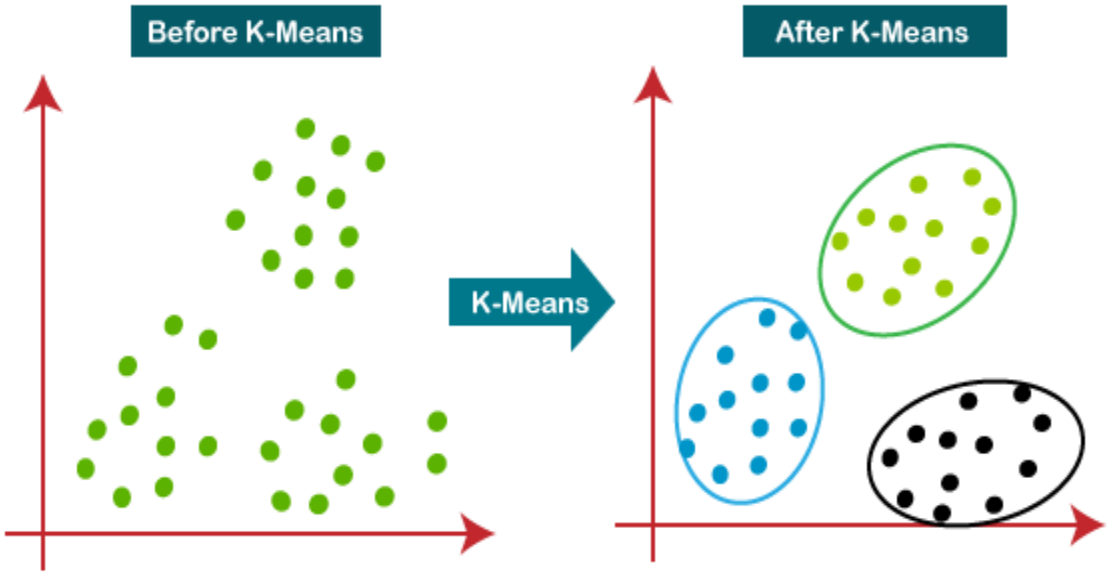
Principal Component Analysis

Principal Component Analysis is used to deal with large datasets that have many different variables. It falls under the unsupervised learning category but is similar to linear discriminant analysis in the way that they both search for linear combinations of the variable in the data. It

reduces the dimensions of the data and finds out which variables are the most relevant. It finds the variables that account have the most variance and thus covers the most of the data.

K-Means Clustering

K means clustering is one of the most popular unsupervised learning methods due to its simple nature. You select a target number which is the k. The algorithm then separates the data into k clusters based on characteristics. This is useful for identifying the structure of a data set. Similar data is put in clusters so all data in in cluster with the same characteristics. As you can see in the diagram below it groups similar data together.



Reinforcement Learning

This method interacts with the environment. It learns how to behave in a certain environment by performing actions and seeing what the results are. It is essentially trial and error and see what the best method is. It rewards the most relevant characteristics of the learning.

Q Learning

Q learning works by assigning each step of the process a reward value and finds the optimal policy by maximizing the expected value of the total rewards over all of the steps. It falls under

reinforcement learning as it keeps repeating all the combination of steps and learns the best method for the environment.

Ethics

There are several ethical concerns with measuring software engineering as you are handling someone's data. Most developers will be spending the majority of their time at work on a computer which makes it easy for companies to monitor what they are doing. However, it can lead to a sense of unease for the developer if they know they are under surveillance. The handling of other people's data is very controversial at the moment given the recent introduction of the GDPR laws. The main concern with the ethics is that people are entitled to their privacy. With this situation as well it can be possible that the figures don't show the real situation. In this report I will focus on looking at the ethical issues surrounding privacy and misleading figures.

Privacy

People are entitled to their own privacy and the question is how much is too much? Where is the line? There was lots of controversy when Amazon secured a patent for wrist watches that tracked their employee's movements and how long they were still for. Many people believed this was a breach in privacy and a step too far. Is this any different to companies monitoring their developers? In 2018 the European Union introduced the General Data Protection Regulation. This has had a huge affect on handling other peoples' data. It needs to be much clearer on what the data is for and why they need it. The individual most be aware that their data is being collected and the company must get the persons consent before they can collect the data. Privacy has become a big topic over the last few years especially with the controversy with Facebook.

Misleading Figures

Another ethical issue is the figures being incorrect. As we saw above there are many different metrics for measuring a software engineering process. None of them are show the full picture. Some developers might know how to mislead the system. This could be unfair on the good developers if they are appearing worse because someone has cheated the system. If a company solely looks at the figures and fires someone just from the figures and they were wrong, it is very unfair and ethically wrong. In team projects leadership and teamwork are essential skills that cannot be measured and thus not shown in the figures.

Conclusion

In conclusion, there are many ways to measure the quality of a software engineers work. I have only looked at a few of the metrics, platforms and algorithms in this report. There is still no definitive best method to use. This report has outlined several different methods for measuring software engineering process' and has also looked at some of the platforms available to do this. It has also looked at some of the algorithms used to assess employees and the ethical concerns involving monitoring employees. From what I have seen I believe there is no way to properly assess a developer as the methods do not take into account everything about their performance. There is, however, some very good methods and many different ways to look at it. The industry is growing massively and constantly improving and it will be interesting to see what new methods and platforms will be introduced in the next few years.

BIBLIOGRAPHY

- https://en.wikipedia.org/wiki/Software_engineering#:~:text=Software%20engineering%20is%20the%20systematic,engineering%20is%20a%20computing%20discipline.
- https://study.com/academy/lesson/software-engineering-definition-process-methods.html#:~:text=The%20software%20engineering%20process%20consists,approaching%20software%20development%20and%20delivery
- https://pdfs.semanticscholar.org/ca67/5537d748957c2f0a49e28df2265018f2f958.pdf
- file:///C:/Users/user/Downloads/Kneuper2018_Chapter_SoftwareAndSoftwareProcessMeas.pdf
- https://www.pluralsight.com/blog/teams/why-code-churn-matters
- https://www.techopedia.com/definition/8073/lines-of-code-loc#:~:text=The%20phrase%20%E2%80%9Clines%20of%20code,used%20to%20write%20a%20program.
- https://www.guru99.com/software-testing-metrics-complete-tutorial.html
- https://www.practical-agile.com/280-bugs-as-a-metric
- https://en.wikipedia.org/wiki/GitHub
- https://stackshare.io/stackups/github-vs-gitprime
- http://www.semanticdesigns.com/Company/
- https://www.expert.ai/blog/machine-learning-definition/#:~:text=Machine%20learning%20is%20an%20application,use%20it%20learn%20for%20themselves.
- https://en.wikipedia.org/wiki/Linear_discriminant_analysis#:~:text=Linear%20discriminant%20analysis%20(LDA)%2C,classes%20of%20objects%20or%20events.
- https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
- https://en.wikipedia.org/wiki/Q-learning#:~:text=Q%2Dlearning%20is%20a%20model,to%20take%20under%20what%20circumstances.
- https://digitalguardian.com/blog/what-gdpr-general-data-protection-regulation-understanding-and-complying-gdpr-data-protection
- https://www.nytimes.com/2018/02/01/technology/amazon-wristband-tracking-privacy.html
- https://www.geeksforgeeks.org/personal-software-process-psp/