**Measuring Software Engineering**

**By Cian Crowley**

**18328635**

Page 2 - Intro

Page 2-4 - Measurable data

Page 5-6 – Platforms/software available

Page 7-8 – Algorithmic approaches

Page 9-10 – Ethics

**Intro**

We live in a world where data is everything. Over the past decade and longer, society has started to realise how valuable data is. In the line of engineers and especially software engineers, these people are slowly being tracked and monitored more and more. People and business are pumping millions into getting the most out o their software engineers. Efficiency is key to these companies.

If I were to record everything I did in a day. If I were to analyse it to see how long I spent doing the work I needed to do versus being on my phone scrolling on Instagram or twitter. Would my day be efficient? What would I consider efficient? Would my future employer consider efficient? These are questions I seriously must ask myself as I pursue a career in this industry.

You may be asking, how they can do such a thing? What else they can employers find out about employees? How can they physically acquire this information and what algorithms are behind it? These are all questions that I will be answering in this report. As we move into the future, we must be aware of what data is been taken of us and how. We should also always question the morals and ethics of any of these schemes. It is and always will be important to question the boundaries of your situation.

**Measurable Data**

Yes, it is important that we question the boundaries of our situation. However, before we can do this, we must understand exactly the situation first. What information are these companies and businesses looking for exactly? These businesses are looking for any kind of pattern in engineers’ behaviours that can lead them to work with efficiency. A perfect example of this would be stress and anxiety. These are two very common things in the workplace that have been proven to slow down workers. This applies to software engineers just as much as anyone. Employers don’t want their employees to slow down in any way. They want employees working at maximum capacity. To do this they want to control everything they possibly can.

If an employer finds out one of their employees is stress or anxious, then it is definitely in the employers interest to give that person some time of or even figure out a way for them to let off some steam. This is only an example, but the real question is, what information is most valuable to employers? Employers want to know what their employees are doing at every second, however this is nearly impossible for employers to check their employees every move. They would like to know if employees are happy in their workplace or if their considering leaving.

Happiness is a huge factor in the workplace, countless research reports and experiments have shown that people who are happy are more likely to be successful and healthy. On the flip side of this mental illness is on the rise and affecting more and more people each day. This has a huge affect on workers. Workers take days off the more they feel sick and this ends up with less productivity in the workplace. This is why a lot of big firms have invested in a lot of mental health resources for workers. It is in both the workers and the employer’s best interest. Data surrounding employees mental and physical health is very valuable to businesses, they can anticipate a certain workload from people depending on their health.

This leaves employers asking, what else can we find out about software engineers? They want the best from their engineers, so they go above and beyond for this data. We live in a time where data is costly, and the value for data is just increasing. Companies are paying more and more for ways to calculate data of their workers and their customers. The end goal here is for all software engineers to be working at their very best.

In this data hungry world, what is it exactly that software engineer employers are looking for? Well the answer is simple, anything they can get their hands on. These people want to know as much about your background as possible. They will cover all your social media platforms to understand the type of person you are. This can be anything from, what sports or games you play competitively to your political beliefs. These employers want to know who they are hiring. This is for the majority of professional jobs too. However, there is a lot more that employers are able to record about software engineers in particular.

Software engineer employers begin by looking at what skills these potential employees have. They don’t just look at their CV’s and take their word for it. They research into these people. They checkout their Github accounts for example. Here, they can see exactly what software they are familiar with, how familiar they are with each software, who they have worked with before, are they used to working in groups or alone. This alone, helps software engineer employers to sculpt the team they would like. This is even used within companies and not just for hiring. When a manager would like to build a team of software engineers to perform a task, all they have to do is check the database of software engineers and see who is the best at working in a team and who are the best at any given software.

This may seem pretty basic, but it goes even further. These employers or even managers want everything perfect. They build statistic and data based on engineer’s code commits, their branch push and pulls. The number of commits help these employers see the regularity of code being written and also what exact work is being done. This gives them a better understanding of all their engineers and how they think. They look at these types of things to make sure good coding practices are taking place. Efficiency is everything to these people and rightly so. Software engineering runs on efficiency. So, if a code is inefficient and uses a lot of memory it this is no use to the business.

These businesses want software engineers who are fully committed to their work. They look at interaction statistics, whether that be just checking emails and responding frequently or involvement in feedback and commits. It is clear to see that nearly everything you do as a software engineer is being monitored. Business want to hire and promote software engineers who are going to be effective in their work. That could be in terms of code or in terms of teamwork and involvement. The data obtainable to business is endless and going to keep expanding in this field of work.

**Platforms/software available**

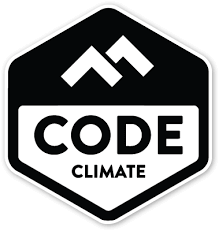
Granted that is a lot to process. I found it easier to understand why they wanted this data when I learned how to retrieve this data. Software engineers may be some of the most creative people this world has to offer, yet it is these people that create platforms or software to retrieve data from other software engineers. A simple way to do this on a small scale would be to record all engineer’s personal software process. A personal software process is a way for software engineers can understand their routine and discipline themselves in a way that makes them write code and execute tasks at a more efficient rate than they normally do.

Timeulars and Fitbits are just two simple devices that some software engineer employers provide to keep an eye on their engineers. It is a well known saying that time is money, and this is definitely valid for software engineers. Timeular is a simple 8-sided dice device that tracks the amount of time engineers spend and different activities. These are usually actions such as writing code, debugging, sending emails etc. It breaks down for the employer to see how well these people are spending their time and how they could maximise their workload.

The Fitbit is a device that tracks a person’s health. This device records the length of this person’s sleep, calories burnt, exercise done and more. Now you may question, well what does this have to do with a software engineer? A person whose job is not based physical activity. Millions has been invested into research in how to get the most out of employees and one of the biggest factors regardless of profession is always physical health. The reason being, if a person is physically active it induces endorphins which help with mental health and therefore help the software engineer or whomever it may be, keep a clear head whilst working.

These are only some of the common software used on employees and not even in particular software engineers. There is software out their available to business or companies, such as Pluralsight, Waydev and Code Climate. These software are used to track and record software engineers. These are the type of software that retrieve the type of data which I mentioned previously. The idea of these software is simple. They take in any project which is being worked on by a group on a certain platform and it analyses it. The software is able to compare the work of everybody against each other. This makes it very easy for managers to see who is pulling their weight, who is under-performing and who is over-achieving.



These are only a few of the available software out there. There is plenty other software that can help an employer compare their software engineers. The examples I have given are just a few, yet they are the more popular ones. However, what I find most interesting about this is the software used to track and predict behaviour using algorithms.

**Algorithmic approaches**

There are algorithms to nearly everything in live, the only thing is, some have been discovered and some haven’t. Any company nowadays, that is trying to retrieve data from people needs and algorithm to do so. There are many examples of business and employers using these algorithms through software to measure their software engineers. These algorithms are solely built to retrieve the data I talked about in the section ‘Measurable Data’. Any data that can help the productivity of a company or the efficiency of code.

Any algorithm that is going to predict the actions of humans needs research before-hand. In the past experts have been able to see patterns in code from people who have coded near enough to the time of quitting, so much so that they are able to predict to a high level of accuracy when someone is considering quitting that company or job based on their style of code. This caused for algorithms, based on these predictions. Managers would want to know if their employers are considering leaving before they do so. This would allow managers to act appropriately.

We have seen how managers and companies want all their staff to be happy. This is only because of the research done to show how a happy employee will work better and more productive. The research come before the algorithms, but the algorithms are made to determine the outcome. There is plenty more examples of this. The algorithms continue extend their complexity as research becomes more and more parallel to the algorithms in question. For software engineers it goes a bit further. These algorithms are made to calculate a score returning function to calculate different measurements of work or even code in these cases.

After, even, a small bit of research into measuring software engineers, and the algorithms that surround these measurements. You will find that ABC metric and cyclomatic complexity are two of the most common algorithms associated with software engineering. Both use metric scales and sometimes even vectors to measure code, however there is obviously some differences between the two. They are used in the line of work, to compare the work of the software engineers, hence creating a competitive atmosphere.

Cyclomatic complexity or otherwise known as CYC for short, is a metric system used to test the complexity of a code. This is calculated by the number of decisions inside the source. The more decisions the code has to make the more complex the code is. This code will always calculate the number of decisions with one more added on. These decisions are the while and for loops, and also the if and else statements. The higher the CYC does not mean the code is better, actually quite the opposite. Employers are always looking for the least complex codes. The actual formula is as follows:

**CYC = E – N + 2P**

When E = number of edges

N = number of nodes

P = number of parts to the flow graph which are disconnected

ABC metric is a bit different. Similar to cyclomatic complexity, however ABC metric is used to compute the size of a source code. Again this is used to compare workers, and to create the most efficient code possible. This means that the smaller the size of the source code the better. As you will see the variable B is quite similar to that of CYC, so therefor, it takes complexity into account. This can be computed using a simple algorithm however, it does differ slightly over software. Most popular in Java, C+ and C++. The formula is as followed

**|ABC| = sqrt((A\*A) + (B\*B) + (C\*C))**

When A = Assignments e.g. data to variable. +=, \*=, %=

B = Branch e.g. function call

C = Condition e.g. Boolean. ==, >, <

These are just two of the most popular algorithms for measuring software engineer’s work. Its almost as if they are being graded like they were in school. The most efficient code is rewarded.

**Ethics**

There are two sides to the argument of the ethics of this. On one side you have the productivity of a company and the product or service they provide. They are thriving for the best possible output. Then on the other side you the argument that everything an employee does is being monitored; you might consider this an invasion of privacy. Both sides have valid arguments and it is a matter of personal opinion whether you think it is right or wrong. I personally see no problem in it.

It is somewhat ironic how software engineers are building software and program to track themselves. Granted, it is their job, and this is what they are paid to do but it is just feeding into the problem even more. Humans for all of existence have been greedy. We always want more, from eve eating fruit from the forbidden tree, to the destroying of our own planet to even Jeff Bezos exploiting the majority of the world for his own wealth. Yes, you can argue whether it is ethically right or not, but at the end of the day, these things are going to happen anyway.

The most competitive and hardest working people always rise to the top. This is why you see firms all over the world looking for competitive people over everything else. Businesses asking potential employees their history in competitive sports or gaming. These examples might not even have anything to do with the line of work, yet you see with these measuring tools how exactly a competitive atmosphere can be created in the workplace. There is the example of the 4 stages of team productivity are, forming, storming, norming and performing. This shows how most team projects should go and indicates how storming is healthy for a team project. Although you may not agree with arguments amongst your profession, they happen anyway, and the employers are more than happy to see it. This is one of those things that are not liked by many yet happen anyway, just like the measurement of software engineers.

You can blame it on the management of companies or the employers, yet you can see that their roles are being tested just as much as software engineers, if not more. Management is under serious threat of being overthrown by AI. The more AI learns about human patterns and these algorithms of efficiency the more likely it will be that they finally overthrow these managerial positions. Is AI as a boss ethical? Consider hiring patterns in the past, where white males would receive the majority of jobs. This trend would just be followed by AI, as it leans more toward previous human patterns. I don’t believe this is fair however it is out of my hands. Humanities need to be more efficient and better than each other prevails. This leaves me with the option to follow this pattern which leads me to a job and a good life in this society, or I can revolt and stand up for what is right while risking the luxurious life. It’s a choice we all have to make but there is more problems than just this.

It’s the sad truth that our privacy is lessoning more and more as time goes on. The value of data is increasing exponentially. It is my belief that data is the next big form of currency. We can even see it coming into play with cryptocurrencies. Soon enough companies will be paying millions into discovering all they can possibly find about their employees. The margin for error in our personal lives is becoming smaller, day by day and its scary. Considerably unethical however, the decision is out of my hands, I must adapt and learn to be efficient in whatever I do.

***Resources***

* <https://www.youtube.com/watch?v=Dp5_1QPLps0>
* <https://waydev.co/how-it-works/>
* <https://en.wikipedia.org/wiki/Personal_software_process#:~:text=The%20Personal%20Software%20Process%20(PSP,actual%20development%20of%20the%20code>.
* <https://uk.pcmag.com/security-5/124891/the-quantified-employee-how-companies-use-tech-to-track-workers>
* <https://www.pluralsight.com/product/flow>
* <http://www.hitachi.com/rev/pdf/2015/r2015_08_116.pdf>
* <https://www.perforce.com/blog/qac/what-cyclomatic-complexity>
* <https://wiki.c2.com/?AbcMetric>