Measuring Software Engineering Introduction

by Arthur Sasunts

When working on a project, software engineers need to acknowledge the efficiency, reliability and cost of the software. However, while trying to meet these demands, problems often arise due to timelines and budgeting which usually results in a lower quality product. Software engineering was introduced to address these issues.

Software engineering is the process of analyzing user needs and designing, constructing, and testing end user applications that will satisfy these needs through the use of software programming languages. It ensures that the software is built consistently, correctly, on time and on budget and within requirements. Software engineering is usually used for more larger and complex systems, which are often the critical systems used by businesses and organizations. Technology in today's world is constantly changing and improving which results in the requirements of a software to be altered. Software engineers must adapt to these changes and work on a solution whilst also staying within the requirements of the end user. Therefore, a software engineer must have a wide range of skills and be able to work on these changes efficiently and effectively.

However, there must also be discipline and control when it comes to software engineering. Some measures are used to assess the quality of the software and to gain a better understanding of the work. Software is usually measured in two categories, direct and indirect measurement. Direct measurements include software processes such as cost and work ethic and products like lines of code, execution and others. Indirect measurements include products like functionality, quality, complexity, reliability, maintainability and many more.

There are four main aspects that I will explore in regards of measuring software engineering, measurable data, computational platforms, algorithmic approaches and ethics.

Measurable Data

Source Lines Of Code

There are many ways in which data can be measured. One of which is Source Lines of Code (LOC). This is a measure taken to count the size of a program or software by counting the amount of lines of text there is in the program's source code. LOC is used in various ways to assess a project, and there is debate on how effective this measurement is. It is typically used to predict the amount of effort that will be needed when creating a program, as well as to estimate programming productivity or maintainability once the software is produced. People often question how effective this measure is as it does not provide a reliable answer. This is because a program can have around a thousand lines of code and work correctly whereas another program that carries out the same function and gives the same result may only have a few hundred lines of code. Sometimes you may think that because a program has more lines of code, it means that it is better, but that is not the case. Most software engineers try to use the least amount of code as possible so their program can be simply but efficient. However, there are still advantages to using LOC as measurable data. Some advantages include:

- It is universal and widely used.
- It is easily measured.
- It allows comparison of size and productivity between diverse development groups.
- It is an automated process so a software engineer doesn't have to count each individual line himself.

There are two main types of SLOC measures, physical SLOC (LOC) and logical SLOC (LLOC). As mentioned before physical SLOC counts every line of code excluding comments, whereas logical SLOC counts the lines of executable statements in the code. LLOC defines a statement depending of the language type. For example, in C like languages, a statement is defined when a line ends in a semi-colon. We can use the following snippet of code to see the difference between LOC and LLOC.

```c
/* Now how many lines of code is this? */
for (i = 0; i < 100; i++)
{
    printf("hello");
}
```

- Physical SLOC will count 4 lines of code.
- Logical SLOC will count 2 lines of code.

Generally logical SLOC is better as it counts the exact executable statements however code can be written differently depending on the programmer. That piece of code can be written as:

```
for (i = 0; i < 100; i++) printf("hello"); /* How many lines of code is this? */
```

Now physical SLOC will count one line of code and logical SLOC will still count two. Overall, the lines of code measurement can be effective but not always accurate.

Function Point Analysis

Function Point Analysis (FPA) is a measure used to express the amount of business functionality a software provides to the user. It was first made public by Allan Albrecht of IBM in 1979 and has been proven to be a reliable method for measuring the size of computer softwares. It provides functionalities such as estimating projects, managing change of scope, measuring productivity, and communicating functional requirements. The FPA technique quantifies the functions contained within software in terms that are meaningful to the software users. The measure relates directly to the business requirements that the software is intended to address. Therefore, one of the primary goals of Function Point Analysis is to evaluate a system's capabilities from a user's point of view. FPA assists the user in managing their software by providing five basic functions categorized into two groups, Data Functions and Transactional Functions. Data functions address the data requirements of the end user and consist of:

- Internal logical files.
- External interface files.

Transactional functions address the user’s need to access data and consists of:

- External inputs.
- External outputs.
- External inquiries.

Once the function is identified and categorized into a type, it is then assessed for complexity and assigned a number of function points. Each of these functional user requirements maps to an end-user business function, such as a data entry for an input or a user query for an inquiry. All of the functional components are analyzed and added together to derive a Function Point count. There are many advantages to Function Point Analysis, such as:

- It is a tool for estimating costs and resources for software development and maintenance.
- It is independent of the programming language, technology, techniques.
- Creation of more function points can define productivity goal as opposed to LOC.

Computational Platforms

Computational platforms are the environments used to create/run a software. There are many computational platforms available for software engineers, most of them open source, meaning they are free to use. They can be very diverse with different different platforms providing different functionalities and restrictions. The computational platforms that I will be looking at is:

- Git-repository hosting.
- IDEs.
- Operating systems.

Git-repository hosting

Git is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files. There are many git-repository hosting sites but the more popular ones include the likes of GitHub, Bitbucket and GitLab. All of these sites are usually open source but also offer some premium features such as unlimited private repos. These websites are great for software engineers as it allows them to share their code with the world. Most of the repositories are open source so anyone can see/use your code. There are also many open source projects where people from around the world can contribute to. However, teamwork is where these git-repository websites excel. If a group is working on a project and each member is given a different part of the program to complete sites like GitHub is the perfect place to work on. The group can create a repository and commit their individual work or commit their part of the program to the main file. This way each member of the team can see the changes in the main program and pull from the repository. Another great feature is that it records the commit history. So if a member of the group isn't contributing to the project, the rest of the group can see. This works well in college group work or in companies. All of the repositories can be accessed and edited using the git-command line interface. Another great feature about git-repository hosting is that they can be integrated with IDEs and text editors. For example, applications such as Atom and Microsoft Visual Studio have GitHub integration which means that users can easily commit their work through the app without having to use the terminal. This makes it so much easier for software engineers to get their work done and takes less time to do.

Integrated Development Environment (IDE)

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools, and a debugger.

IDEs are where most software engineers carry out their work. They are very efficient and simple to use as you can build, compile and run programs without the need for a terminal. There are a lot of IDEs available, usually open source. Every IDE differs from each other with specific functions and unique features. Some IDEs also specialise with only one programming language and are used to develop applications for that specific language. For example, the Eclipse IDE is primarily written in Java but it can support other programming languages through plug-ins.

IDEs are also very useful to run tests on programs. Software engineers can test their code by running unit tests that they can write themselves. This ensures that their code can pass any test which may cause the program to fail. If they find a test that fails, they can go back to their code, locate the problem and fix it. Software engineers can also test for code coverage of their program. Code coverage measures the extent of which the source code of a program is executed through unit testing. High code coverage is generally better as it means that most of the code was executed when the tests were run which in turn leads to less bugs in the code. If the code coverage is lower, this might mean that there are bugs in the code which have not been tested. IDEs such as Eclipse and Visual Studio allow for unit testing and code coverage testing.

Operating Systems

Although operating systems do not have a significant role in developing a software, they do allow software engineers to customise and code they way they prefer. Usually it's personal preference to which operating system you use. I believe that despite operating systems not having a huge impact on the way you code, it still may affect your work. If you do not like using a Windows operating system, you probably won’t enjoy coding on it too. For me personally, I do not like Windows systems but prefer Macs. I feel more comfortable using a Mac and enjoy doing my work on it.

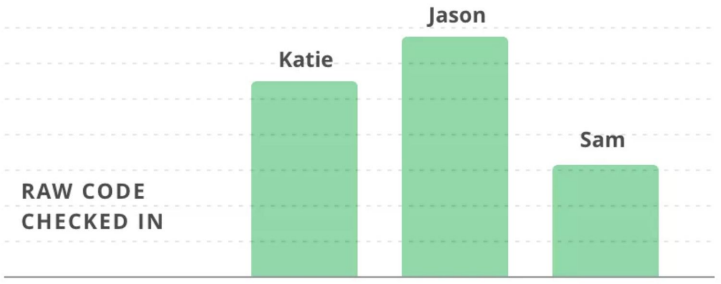
Sometimes operating systems do matter while programming. For example, if a software engineer wants to create applications for an Apple device, they can only do so by using a Mac operating system. It is not possible for them to create it on any other operating system as Apple do not allow it. However, one way to overcome this issue is by using a virtual machine. Software like VMware allows you to install another operating system within a virtual machine on your computer. This is a great tool because software engineers can install multiple operating systems and switch between them if needed.

Algorithmic Approaches

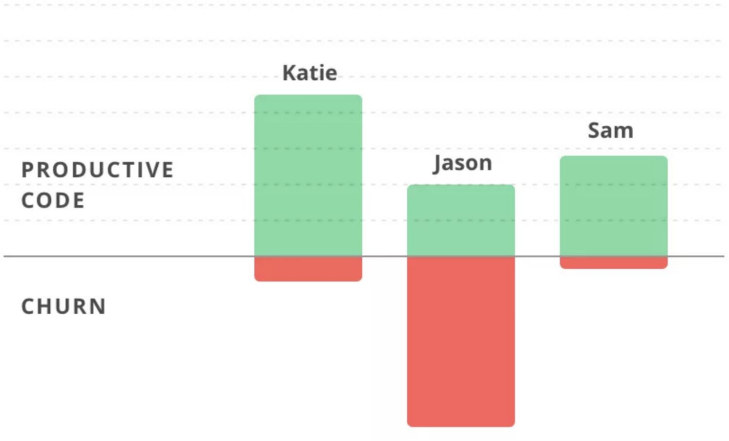
There is variety of algorithmic approaches a software engineer can take to calculate data being measured. These include approaches such as code churn and machine learning.

Code Churn

Code churn is the measure of the rate at which your code changes. This is a good way of looking at the productivity of a software engineer. For example, if there is a group of three people working on a project, normally you would deduce that the person with the most code written has contributed to the project the most. However, that is not the case. If we take a look at this diagram we can see that Jason has



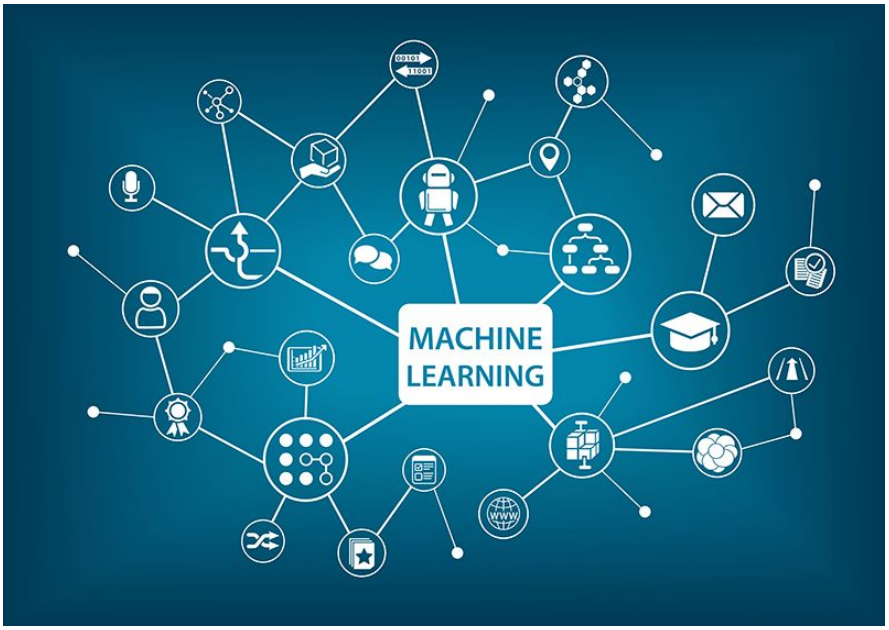
contributed to the project the most. However if we look at the diagram below, illustrating the code churn of each software engineer, we can clearly see that the result is completely different. Katie was the one that contributed the most to the project.



Although Jason wrote a lot of code, most of it was deleted and as a result, his code churn was very high. Sam didn't write a lot of code but he was very productive and his code churn was the lowest and Katie wrote a lot of code but had little code churn. Code churn essentially shows which software engineer was the most productive.

Machine learning

Machine learning and artificial intelligence are one of the fastest growing sectors in computers today. Machine learning is a method of data analysis that is automated which uses artificial intelligence to learn from data, identify patterns and make decisions on its own. It provides the ability for systems to automatically learn and improve from experience without changing the program. There are many applications for machine learning in today's world. If we take a look at autonomous cars, a lot of the software being used is machine learning technologies. A company called Wayve created a software that allows a car to drive autonomously by only using a camera to detect it's surroundings. They claim that their software uses a unique end-to-end machine learning technology and can learn to drive in a new city with minimal new data or maps. This approach is also much cheaper than what the big car companies use, which is a system called LiDar and costs roughly €80,000. I find this absolutely fascinating. Another interesting application for machine learning is for games. AI bots can be trained using machine learning to be the very best at a game, even better than actual players. An example of this is when Elon Musk created a bot for DOTA 2 using machine learning tehcnologies. What's most interesting about this is the fact that they didn't program the bot to know the rules of DOTA 2, instead they let the bot play lifetime amount of 1v1s against itself and over time, the bot learned how to be really good at the game. They brought the bot to the DOTA 2 international championships in 2017 and let professional players play against it. To everyone's surprise, the bot kept beating the best players in the world and it never lost a game. This is incredible as it shows us how technology can teach itself to become good at something without being programmed to. Machine learning is also used in social media platforms. For example, Twitter processes lots of data to recommend tweets based on who people follow and what they tweet as well as detecting any racial or inappropriate content and removing it from their platform.



Ethics

There are huge ethical concerns when it comes to data collection and analysis, especially in today's world where a lot of data is being collected every single day. Two of the biggest ethical subjects involve privacy and data protection.

Privacy

One of the biggest ethical concerns regarding data is privacy. Any unauthorized access to information can be an invasion of privacy. Even when access is authorized, it still may lead to privacy issues. Data is usually collected autonomously and if you are a software engineer, a lot of data will be collected. Employers can evaluate their workers performance through methods described above and collect data about them, of course with the consent of the employee. Although this information being collected may be beneficial to the employer and the company, it can be dehumanising knowing that everything you do is recorded and analysed. However, an employer can use this information to help his employees. If they can see that one of their employees is falling behind or isn't doing the necessary work, they can help them out and steer them into the right direction. But if an employee uses this information to fire someone, it would be very unfair and unethical.

With all the privacy concerns in the last few years, the EU have released the General Data Protection Regulation (GDPR) law. This is a regulation for data protection and privacy for all individuals within the EU. The GDPR provides significant improvements to current data protection rules. It provides higher standards of data protection and aims to give control to individuals over their personal data. This is reassuring for customers as well as employees as it makes sure that companies are clear and intent with their policies and what data they are collecting. After the introduction of GDPR, if you enter a website, the terms and conditions come up and you must accept them in order to use the website. I believe that this is the right step as it bring more trust to customers and ensures that their information is safe.

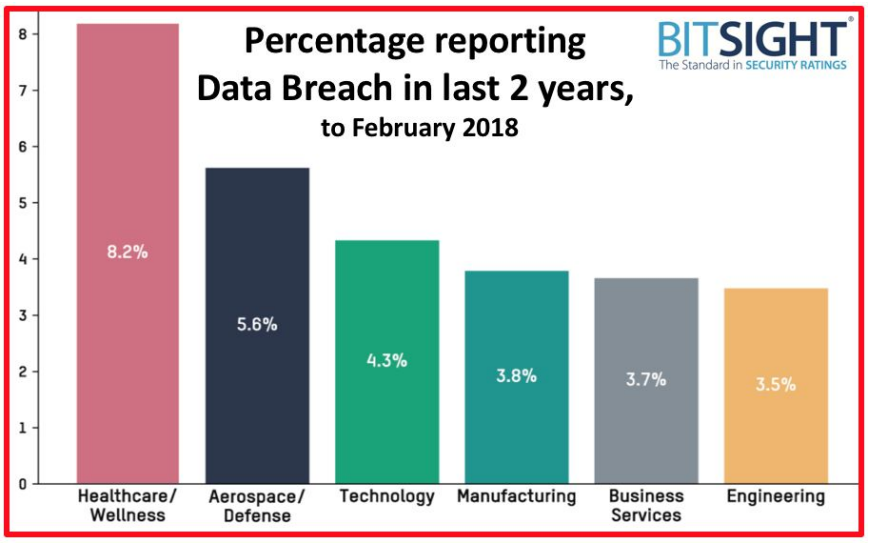
Data protection

Another big concern regarding data collection and analysis is the guaranteed protection of this data. Any platform that collects your data will claim that everything is safe and protected. Although this is true, there has been many cases of data breaches and leaks within major companies as well as hospitals and banks. These breaches can occur intentionally by hackers or unintentionally by employees accidentally leaking information. There are many examples of large data breaches over the last few years and the rate at which it's happening is worrying. Nobody wants their information being stolen and released to the world.

Impacts caused by data breaches include:

- Loss or compromisation of customers' data.
- Employees' data is put at risk.
- Companies could suffer from DDoS attacks.
- Companies can lose a lot of money.
- Companies can suffer damaging downtime.
- Companies may lose their trust and reputation.

As you can see, companies can suffer a lot from data breaches. One of the biggest breaches in the world occured in 2017 to Equifax. In September 2017, approximately 145.5 million people were potentially affected by this breach. Valuable information about the customers of Equifax such as their first and last names, social security numbers, birth dates, addresses and driving license numbers were accessed by the hackers. More recently in March 2018, Under Armour, revealed that a data breach occurred affecting 150 million accounts on MyFitnessPal. Also in March, the major political scandal of Facebook-Cambridge Analytica. It was revealed Cambridge Analytica had harvested the personal data of millions of people's Facebook profiles without their consent and used it for political purposes. All of these examples are major data breaches and have affected many people. From the graph below, we can see the sectors that suffer the most data breaches.



Conclusion

In this report, I have discussed how data can be measured for software engineers, the computational platforms available to carry out this work, the algorithmic approaches available and the ethical concerns surrounding data collection and analysis.

In conclusion, there are lots of ways to measure software engineering. Companies rely on this information to monitor their employees and make sure that they are working to the best of their abilities. However, as they do this they must also make sure that they have an ethical approach and ensure that everyone's data is protection and is used in the right ways.

References

- www.techopedia.com/definition/13296/software-engineering
- economictimes.indiatimes.com/definition/software-engineering
- ecomputernotes.com/software-engineering/software-measurement
- https://en.wikipedia.org/wiki/Source_lines_of_code
- www.careerride.com/pmp-advantages-of-using-line-of-code.aspx
- www.qpmg.com/fp-intro.htm
- https://en.wikipedia.org/wiki/Function_point
- https://www.careerride.com/pmp-advantages-of-function-points-analysis.aspx
- https://blog.gitprime.com/why-code-churn-matters/
- https://wayve.ai
- https://en.wikipedia.org/wiki/Data_breach#2018
- https://www.cyberrescue.co.uk/library/threat
- https://dataconomy.com/2018/03/12-scenarios-of-data-breaches/