

---

# **Software Engineering**

## **Measuring Software Engineering**

---

Hailing Jiang 17337907

Nov 2020

### **Contents:**

#### **1 Introduction**

#### **2 Measurement**

2.1 Why should we measure

2.2 What and how can we measure and how can we assess them

2.21 Personal work efficiency

2.22 Team contribution

2.23 Body health

2.24 Work-life balance

#### **3 Computational platforms**

3.1 Platform to measure personal work efficiency

3.2 Platform to measure team contribution

3.3 Platform to measure body health

3.4 Platform to measure work-life balance

#### **4 Algorithmic approaches**

4.1 Self-evaluation with average and mean

4.2 Computational Intelligence (Artificial Intelligence)

#### **5 Ethics concerns**

#### **6 Resource**

# 1. Introduction

In this report, I will be talking about the ways in which the software engineering process can be measured and assessed in term of measurable data. An overview of the computational platforms available to perform this work will be discussed, as well as the algorithmic approaches available. This essay will also discuss the ethical concerns surrounding this kind of analytics.

## 2 Measurement

### 2.1 Why should we measure

Software engineering is a complex process. It involves so many aspects that might affect the efficiency of the work, reliability and stability for the final product.

For be able to push the team and help the members work efficiently, we could apply measurement to working areas where change might need. Since the more productive the software engineerings are, the more efficient for the team and will be able to know the problem in the product on time.

Also these days the health problem for software engineering got attention from the public. There are software engineering who stays up overnight and dead due to that. We know that once we got onto some problems, we might be able to solve that immediately and could spend time on that over night. To make sure the staff have a good life-work balance and good health, the company could introduce effective ways to help them set up the good work cycle and make sure they all follow that.

### 2.2 What and how can we measure and how can we assess them

There are quite a lot of data we can measure for software engineering. From the work perspective, we could measure the personal work efficiency and team contribution. From the staff health perspective, we could measure the body health and work-life balance.

## 2.21 Personal work efficiency

### Code quantity

#### How many lines of code?

```
1 <?php
2
3 /* this is a for-loop */
4 for ($i = 0; $i < 100; $i++)
5 {
6     printf("hello");
7 }
```

7 physical lines of code  
3 logical lines of code  
1 comment lines of code  
6 non-comment lines of code

#### Different types:

LOC: Physical lines of code  
LLOC: Logical lines of code  
CLOC: Comment lines of code  
NCLOC: Non-comment lines of code

*P1: Diagram on different types of measurement on a single piece of work*

For personal work efficiency, we could measure the code quantity by lines of effective code. Rather than just measure the lines of code itself, there is a better way to measure the effective lines. The company used to measure the lines of code written by an engineer, but it really depends on the engineer's personal habit. Some of them prefer to spread the lines out while others might prefer to squish the lines together. Even though there are standard formats to write the code, it's still able to run and work perfectly in most of the languages. Thus there is a better way called LLOC(Logical Lines of Code) to measure the volume of code produced by the engineer. It doesn't count the actual lines but counts the number of statements within the code. There could be few lines been written on one line and it still is able to count it out. There are other ways to measure the lines of code like CLOC(Comment lines of code) and NCLOC(Non-comment lines of code), but once one of the measurement is used, we are able to use that over time and measure the efficiency out.

In this method, it helps out the engineers better with their measurement about the quantity of code. Thus the engineers wouldn't need to try to spread out the code to get better with lines of code in the work.

## Code quality

However, it's also far not enough to measure the work efficiency. Engineers with the most quantity of code might not be the best one among the colleagues. Thus, it also a great idea to count the bugs per line to measure the quality of the code. The easiest way to analysis the quality of the code is to find the average rate of bugs in each line of codes.

For writing the code, fix bugs are the most often things to do. It may get involved more than the writing part. To measure the bug fixes time, it can respect the challenge level of the project and shows what's the problem for the current project process.

## Constancy of code

To measure the constancy of the code written by a software engineering, considering the code quantity and quality over a long period of time by each individual.

From another prospect, the team could check the amount of the code written by the engineer that needs to be rewritten. It's called 'churn rate'. A high churn rate means the low quality of the code been written while a low churn rate indicates that the code quality is relatively high. Churn rate could also be an indicator of how the project goal changes as changing requirements would lead to a high churn rate.

## 2.22 Team contribution

Similar to the personal effectivity measurement, team contribution is to measure the productivity of an engineer which contributes to the group over the project.

## Code quantity

For measuring the code quantity in the project, we can measure the frequency of commit and amount of commit over time. Similar to the measurement in personal effectivity, measuring commit alone is not a good idea. It wouldn't be able to show the commit is important and useful to the project or not. Most of the time people commit their halfway work to the project repository but it still counts one commit to the project. Thus we need also measure the amount of useful code in one commit. This way could encourage the efficiency of the engineers and also could let the product

manager has a broad idea about the amount of each people's contribution over a time. Even though it's not the only way to define is that engineer a good contributor or not, the project manager still would be able to see the comparison between each team members.

### Code coverage

In the code quality part, we can consider the code coverage of a certain software engineer that contribute to the project. Measure the amount of code that reliable and be useful to the project.

### Test coverage

All code has to be tested before release to users or process to the next stage. The test is a big part to do. Though the amount of code been done is important, it has to go together with test coverage to make sure the code works. Especially to make useful tests, that be able to catch the edge cases which may cause the software or project to fail. It's possible to write a lot of tests but just test very little things. Thus measure the test coverage would be important to consider the contribution of a software engineer in the teamwork as well.

### Documentation in the code

Documentation makes the code more readable. In the large program, sometimes the people need to rewrite the code again as the requirements update. Thus the documentation is important to let the next person handle on it to know what parts have been done and what needs to update. If the code is unreadable and there's no documentation to explain it, the code is not reusable and it was a waste of time for the contributor. The more readable the code, the better note for the future colleague to work on the project.

## 2.23 Body health

Track the various data about the health of the engineers would be important. As different people have different health problems, the health data would allow detailed analysis of their performance and also for improving their health.

Some companies give engineers Fitbit watch to measure the health data like heart rate, breathing rate, blood oxygen level and stress level etc. Combining

these data would easily find out the relation between stress and work performance on them.

Not only related to work, but good health is also the base of everything else, including a better work status. If there's any problem been to figure out through this process, the engineer could find the solution way earlier and better for their own body.

## 2.24 Work-life balance

Some companies use the chair which is able to measure the sitting time and regular walk-off time and give the engineers the idea about how long time they were working and how long they started to walk. This product is pretty useful especially for the work that needs to sit for a long time and needs to set up a good habit about work-life balance.

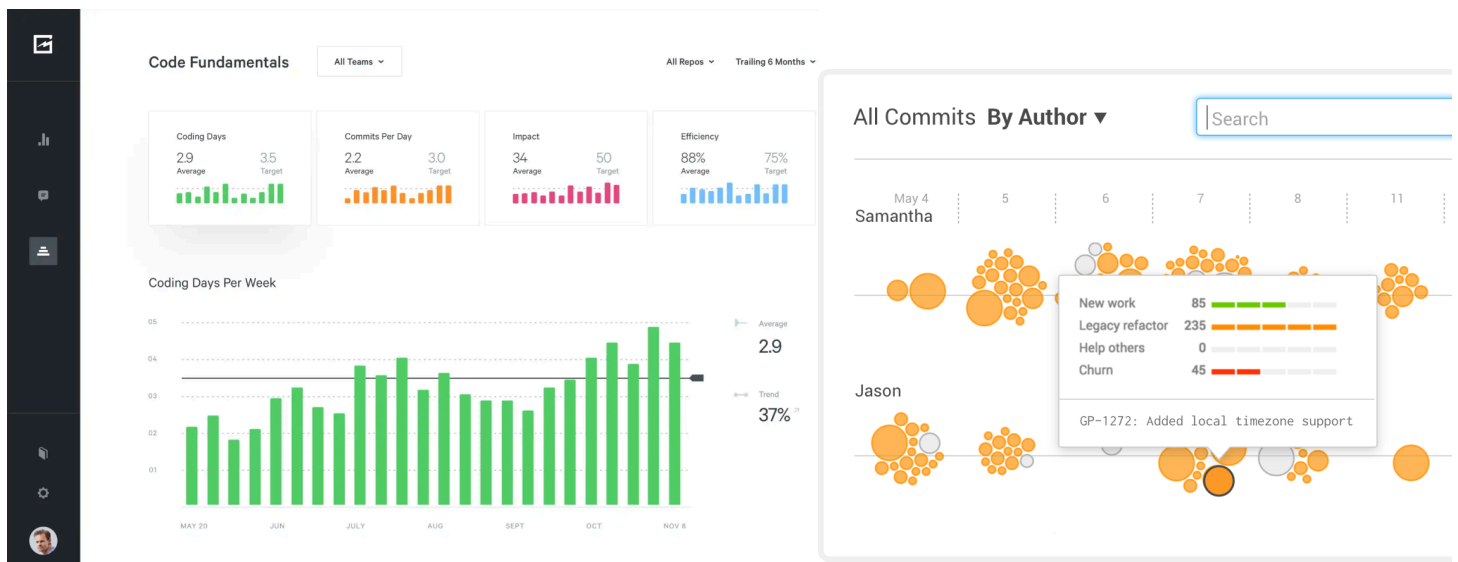
## 3 Computational platforms

There are lots of computational platforms that are able to measure the above data for the engineers and their companies.

### 3.1 Personal work efficiency

For the personal work efficiency measurement, there are products like measuring cube that the workers are able to measure by themselves. The cube could record the daily data and produce the report for a period of time for the engineer themselves.

There are also lots of ready products like GitPrime. GitPrime is a company that develops an engineering intelligence platform. Compare to other companies, GitPrime is more focus on individual performance metrics, equips a manager to keep closer track of their engineers so they have a clear idea of the strongest and weakest performers of the team.



The GitPrime originally imported data from Git alone, therefore it primarily works off of source-code level data, not the collaborative work data. This action enabled them to integrate with more tools(Github, bitbucket, GitLab etc.) and the engineers can only see their own performance in the work and adjust themselves from that.

Since GitPrime is mainly focused on individual measurement, the solution is useful for the manager to determine who to go to solve the problem when something particular went wrong. It let the manager know the personal strength of each people.

### 3.2 Team contribution

Although GitPrime measures more individual data from the engineers, there are also functions which could measure for the team contribution.

It offers a large range of tools and analysers to analyses personal performance and the team impact, team growth and depends on the measurement could predict the outcomes of scaling efforts and suggests sprint plans to the team. It could also measure how collaborative are people within and between teams. Thus it has been great usage among the teamwork and collaboration.

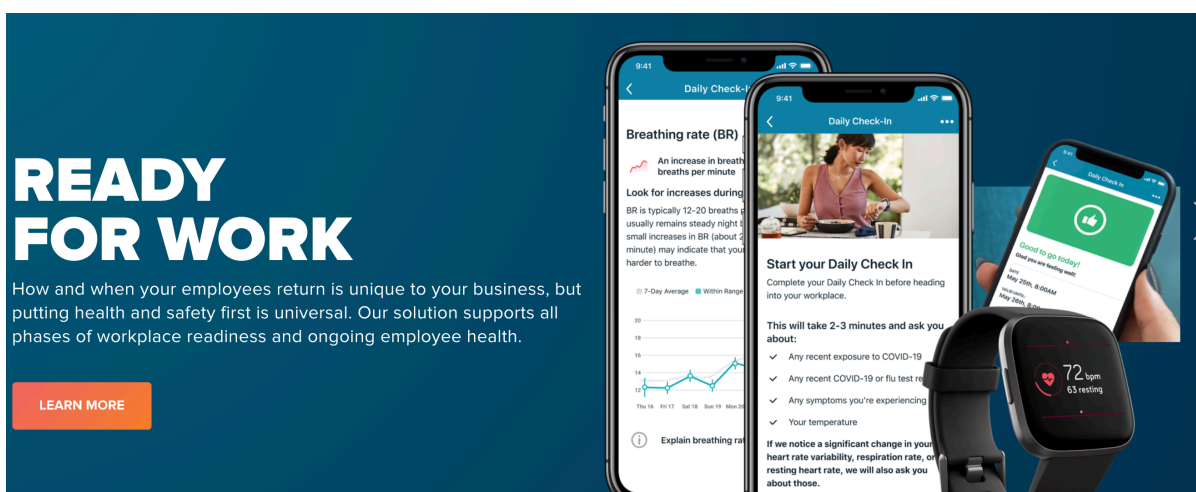
There is also another product called Velocity which has a similar preference as GitPrime. However. Velocity has put collaboration at the centre of its produce. As Velocity mainly focuses on pull requests, it imports data from both Git and Git applications like GitHub and Bitbucket. Then the engineers will have access to more insights into the code review process and into PR - related details. The manager will be able to see the team's efficiency is increasing week over week or not and able to see how much time a team member is spending writing new code vs. refactoring.

Compare to individual measurement, the team performance would be important to the business. Engineers are been expected to continuous deliver business value to the companies. However, a small bottleneck could hold up the entire team progress during any given sprint. The larger the team is, the harder to manage, and it even harder to discover where the work stuck in the pipeline and why. The manager could use the activity log from Velocity to see do the engineers are working on meets the manager's expectations. From the work in progress, the manager could see how far along each PR is from being merged and able to see which PRs are at risk and who's working on them. Velocity is an analytics tool for top-down

management. With all these trackings, the whole team will stay on track easily.

### 3.3 Body health

For the body health of the engineers, as mentioned above, Fibbit watch is one of the methods to monitor engineers health data and doesn't add extra pressure to them. There're companies already using them. The data have been measured could use the related application from Fitbit to collect and analyse and report on a regular base. The following image is the advertisement of Fitbit for prompting the body health check solution on employees.



P3: Fitbit advertisement online to promote employers to concern about their employees' health

### 3.4 Work-life balance

For the work-life balance, Microsoft already introduced their own platform in measuring the time they spend on coding, e-mailing, and all other activities. This kind of platform is great to let the engineer themselves to know their working habit and could make changes if they like to.

The chair mentioned in the previous section would also have platform be able to analyse the data been recorded by the chair, to make the product more useful to use.

## **4 Algorithmic approaches**

### Self-evaluation with average and mean



The easiest way is to use the measured data to get an average and mean from everyone. Compare the data among the team to see who has a better performance and who need improving on certain aspects. For individual performance could compare with their previous own data to see have the improved since last period of time or not. It might takes some times but could let the engineers themselves to engage with the process of evaluation and to know what they need to improve on. By this way, there's no risk of leaking personal information including work performance and health measurement.

For this way of analyse, the data could be analysed by using tools like Excel, Matlab, Python and R. There also need a large amount of work to be done before hand to use these tools to do the proper statistical analysis of the data in the way that they want to.

### Computational Intelligence (Artificial Intelligence)

Artificial Intelligence and Machine Learning are increasingly being used to analyse large amount of data. Specially Computational Intelligence is a subset of Artificial Intelligence with mainly consists of three pillars including neural networks, fuzzy systems and evolutionary computation.

Neural Networks are used for data analysis and classification, associative memory, clustering generation of patterns and control. Nerual Network techniques share with the fuzzy logic ones the advantage of enabling data clustering.

Fuzzy logic system solves uncertain problem based on a generalisation of traditional logic, which could let us perform approximate reasoning.

Evolutionary computation solves optimisation problems by generating, evaluating and modifying a population of possible solutions. It uses the biological evolution as a source of inspiration. It includes genetic algorithms, evolutionary programming, evolution strategies and swarm intelligence.

Computational Intelligence is the theory, design, application and development of biologically and linguistically motivated computational paradigms. With the increasing popularity of deep learning and artificial intelligence, many companies are looking at computational intelligence to see if they can modify a system which can automatically make decisions based on engineers work data and give feedback based on their performance.

Computational Intelligence workers focus on problems that are difficult to solve using artificial systems and are required to solve by humans, which the problem requiring intelligence.

## 5 Ethics concerns

### About work data

Measure the data related to coding wouldn't be an ethical concern as this data is provided by the software engineer for the work. The data is been measured during the work and is only for work.

The only worries about measuring these kind of data is how we use that data. It's not really like the exams we have in the school, most of the questions have certain answers and the result could be valued as right or wrong. Everyone is different and we all have great skills in some areas and weakness in the others. Someone might not be that great in coding but they could make great analysis and report for the team. Julia Liuson has said that in the team we don't have to have everyone to be majored in the same area, people with different skill set would make the team growth better. Under a certain standard on measure the productivity only on the level of coding would let people great in other metrics have a related low score on this analysis.

From all above point, the measurement shouldn't just based on the coding level but also consider other aspect. Or this measurement shouldn't be use strictly with their promotion or stay or not. The employer could use the measured data and result to raise the productivity of the team on the project.

### About personal life data

Measuring the personal data related to personal life would be highly unethical. Though gathering data was to help understand the relation between working performance and their health status, I still think this kind of data is too personal to the employer. The collection of this data could allow employers to control employees personal life outside of their job. Like Fitbit watch is a watch and normally people wouldn't take it off on purpose to not been measured. Constantly gathering data such as heart rate or blood pressure would let the employer have too much access to engineers' personal life. It could result in that the unhealthy engineers been fired as the employer think there's a high percentage that this employee will progress less in the future work. This would be leading to the huge problem that the employees will feel they are been suspended all days and could result in other problem like the mental health problem.

In my opinion, measuring data to improve productivity wouldn't be a bad thing. But it's important to use the proper way to measure the limit necessary data during the work to improve the work efficiency and give the comfort personal space for the engineers.

## 6 Resource

<https://www.aivosto.com/project/help/pm-loc.html#:~:text=A%20logical%20line%20is%20a,not%20a%20line%20of%20code.>

[https://techcrunch.com/2019/05/01/pluralsight-will-acquire-gitprime-for-170m/?guccounter=1&guce\\_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce\\_referrer\\_sig=AQAAALBq32PnT5wSK5TX5B5dzDINGYI07bbLqzyQaVi5n2mCRA9Vcy9JQGma\\_ziudQlarZuPLmE0Jd4WQAiQ4wE5Yaq7IWGeuZrHRY8yg=p09eN0MbNBKnO5wCZArLb8cWrzq0O3eZXfVFSwhl5sTeG1dy7iL1xluWzfKYVRXcmLntjY](https://techcrunch.com/2019/05/01/pluralsight-will-acquire-gitprime-for-170m/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAALBq32PnT5wSK5TX5B5dzDINGYI07bbLqzyQaVi5n2mCRA9Vcy9JQGma_ziudQlarZuPLmE0Jd4WQAiQ4wE5Yaq7IWGeuZrHRY8yg=p09eN0MbNBKnO5wCZArLb8cWrzq0O3eZXfVFSwhl5sTeG1dy7iL1xluWzfKYVRXcmLntjY)

<https://www.softwareadvice.at/software/83124/gitprime>

<https://codeclimate.com/blog/velocity-vs-gitprime/>

<https://www.fitbit.com/global/ie/home>

[https://www.pluralsight.com/blog/tutorials/code-churn#:~:text=In%20benchmarking%20the%20code%20contribution,churn%20\(75%25%20Efficiency\).](https://www.pluralsight.com/blog/tutorials/code-churn#:~:text=In%20benchmarking%20the%20code%20contribution,churn%20(75%25%20Efficiency).)

<https://towardsdatascience.com/programming-languages-for-data-scientists-afde2eaf5cc5>

[https://en.wikipedia.org/wiki/Computational\\_intelligence](https://en.wikipedia.org/wiki/Computational_intelligence)

[https://cis.ieee.org/about/what-is-ci#:~:text=Computational%20Intelligence%20\(CI\)%20is%20the,Fuzzy%20Systems%20and%20Evolutionary%20Computation.](https://cis.ieee.org/about/what-is-ci#:~:text=Computational%20Intelligence%20(CI)%20is%20the,Fuzzy%20Systems%20and%20Evolutionary%20Computation.)