

---

# *Measuring Engineering – A report*

---

**Jakub Slowinski**

student number: 16319781

20/11/2018

## **Specifications:**

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**

A software engineer is responsible for the complete life cycle of a new/modified software product, from research and design to implementation, training and support.

Software engineers develop software applying engineering principles towards the project to fulfil the client's requirements and requests. Most software engineers work as employees with various businesses, government agencies and non-profit organizations.

This role involves: Analysing software user needs and designing, constructing and maintaining computer applications software; constructing working software products, running performance tests and finding ways to eliminate bugs in the software.

Heavy technical expertise, numerous personal accomplishments and experience with the use of open source tools are all expected from a software engineer. They should be proficient with automated testing, pattern design, and fault-tolerant systems.

They must be able to manage resources, coordinate development tasks and supervise technical project plans as part of a team. Collaboration with IT, operations and development staff through the software development life cycle is a must.

A short note on productivity:

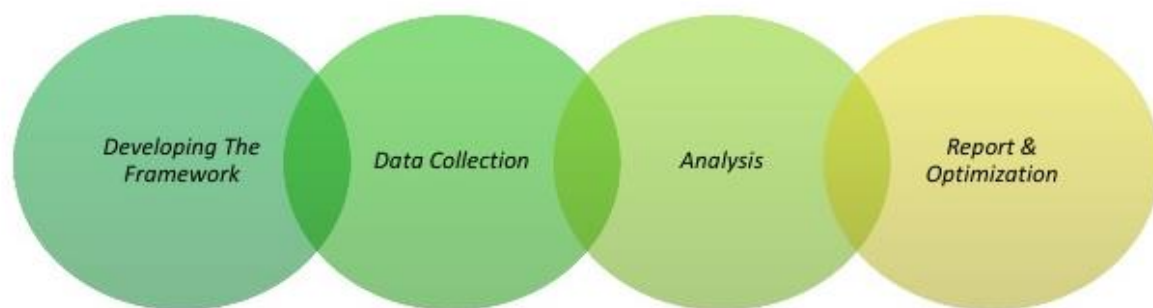
Productivity has been applied as a measurement instrument for assessing different decisions and preventing the waste of resources. Productivity is measured by the ratio of output to inputs used in a production process, i.e. output per unit of input. The bottom line, is that productivity is a crucial factor in the production performance of businesses as it helps them be more profitable. Productivity also helps the employees as it increases the amount of work they output, possibly leading to them being more noticed and appreciated by management and eligible for more promotions. Higher productivity would also lead to better time management, minimising the "crunch" and decreasing stress in employees and there would be less pressure from impending deadlines.

There are multiple forms of productivity and the choice among them depends on the purpose of the productivity measurement and the availability of the data.

Measuring the software engineering process is a method of increasing productivity in your own work cycles, as well as the work cycles of your employees and colleagues. This is accomplished using various methods that I will try to outline in this report. Tracking various outputs e.g. lines of code can be put into ratio with an input e.g. hours work upon, with the goal of assessing productivity and seeing the viable methods for increasing it, in the most effective way possible.

## **Ways in which the software engineering process can be measured and assessed in terms of measurable data**

To measure and assess software engineering we need to first plan and organise the process in a set manner as well as collect data in large quantity and high quality. The quantity is important for populating various algorithms and computational platforms that will analyse the said data. Quality is important as it will bring the outcome of our calculations to a higher standard and therefore makes the calculations more precise, leading to less errors.



Process of analysing data:

1. The framework for collection and reporting should be developed or chosen
2. Collect data from employees and contractors
3. Analyse the data collected
4. Report the finding from analysis while thinking of ways to optimise our process soon

This is an example of how data about the software engineering process could be measured.

- First, it is necessary to set measurement targets
- Specify the extent of which metrics are set for data collection
- Designate the procedures for data collection and storage of said data
- Plan the appropriate methods for analysing collected

The best practices for performing measurement are as follows:

- Collect the measurement data required
- Analyse this data and determine outcome
- Safely store results and data
- Deliver results to team

Some of the test metrics that can be used for calculations along with the measurable data:

Reliability - Number of failures, time spent “down”

Performance efficiency - Stress testing, response time

Security - Time taken to fix failures, lines of error messages, bugs created (technical debt), issues resolved

Maintainability - Number of lines of code delivered, function implemented

Developer - Repository commits, test coverage, code complexity, meetings attended

Rate of delivery - User stories, total developer time required

Testability - Number of technologies needed to test, quality of documentation

Usability - Completion rate, client satisfaction level

Pros and cons to select measurable data are presented in the table below:

<u>Measurable data</u>	<u>Pros</u>	<u>Cons</u>
Unit test coverage	Asserts that code is usable and fits the purpose	Tests can be written in specific scenarios guaranteed to pass, defeating point of testing
Technical debt	A better developer will ship less technical debt and code reviews reduce the number of bugs pushed to the build	Can be hard to track developer due to bugs occurring in good code due to incorrect usage of said code in dependencies
Lines of code committed	A highly productive software engineer will ship more lines than a less productive engineer	Developers who write short simple and functional code would appear to not be making the same impact as someone writing extremely complex code
Frequency of repository commits	A highly engaged team member will be making more frequent commits, so he backs up his work and others are able to review his code	Developers who have a valid reason to not commit often are underperforming, while other statistics might show that they are not.
Client satisfaction level	Good way to determine was a project delivered in a polished and working manner. The higher the	Some clients might be overly satisfied or not enough as there could be inherent bias for the final

	client satisfaction, the better the code and therefore the functionality of the software.	working solution.
--	---	-------------------

Finding points of concern and ways of optimisation:

Finding bottlenecks is important as a single bottleneck can severely limit the performance of a system or program. Tracking down bottlenecks (sometimes known as "hot spots" - sections of the code that execute most frequently - i.e. have the highest execution count) is called performance analysis. This can be achieved with the help of specialized tools, known as performance analysing tools. The objective goal is to make these specific sections of code perform as fast as possible to improve overall algorithmic efficiency.

Analysing trends is a useful tool for determining amongst others whether a bad performance is temporary or if it is set to continue. If a certain developer has a long-standing trend of regression, then a confrontation might be in order, to determine the cause of this and to help improve his performance.

Analysing trends is a good idea to see if a period of extremely high productivity is sustainable. If a team is over performing and it is seen as a trend, then it is only temporary, and they shouldn't be expected to have this much efficiency all year round. It is important in times like these, to look for the cause of the high productivity, be it high morale or a more exciting project and try to replicate these certain things to try bringing back an abnormally high level of productivity when the trend is over.

Performing this analysis lets us find the root cause of a period of low productivity such as determining a time where there were unclear goals. This could be seen with developers having a low amount of code pushed as well as a lot of time pondering over the options they could pursue. This could be solved with more frequent but concise meetings so if anyone spotted something they were unsure of, they had to occasion to speak up.

The data used in the cases above can be collected using the computational platforms in the section below.

## **An overview of the computational platforms available to perform this work**

Toggl: <https://toggl.com/> - a tool for increasing productivity and timekeeping

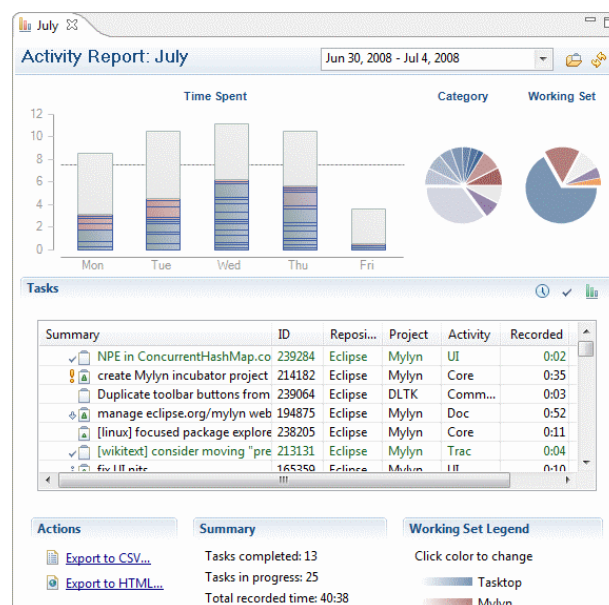
Toggl is a time-tracking app with the niche of being simple to use. It's ideal for agencies, teams and small businesses who utilise a variety of skills and tool for the development process at hand. It keeps track of project time and gets it together in one place, so you can keep an eye on it and keep your team on track.

Can tell you how valuable your time is or the time of a colleague. It can break up your hours between projects, customers and tasks to see what makes you money and what's slowing you down and therefore giving you an opportunity to increase efficiency. It contains many useful tools such as: tracking reminders; idle detection; offline mode; scheduling reports to email; billable rates; exporting report to PDF, CSV or XLS; estimating time estimates; user groups.

Can track time spent through your browser and stop through your phone. All the time entries are synced between the phone apps (iOS, Android), desktop apps (Windows, Mac OS, Linux), the Toggl Chrome and Firefox extension and the website itself.

Tasktop: <https://www.tasktop.com/>

Tasktop is an all-in-one tool which can connect all software delivery tools and automate information flow. It can improve cross-team collaboration as well as measure and collect software engineering data. We will be discussing the latter. Tasktop captures important data as well as populating and visualising a reporting database. This leads to the goal of visualising the important metrics such as development cycle time. Due to this process, it is useful in identifying bottlenecks in the development cycle.



*Sample of a version of Tasktop's activity reports from year 2008*

Tasktop can integrate a large number of tools including: Trello; Git; Microsoft Project Server; Microsoft Test Manager; Salesforce; SmartBear; QAComplete; Zendesk.

Aside from development it also allows you to integrate DevOps tools using Tasktop's Gateway Integration Style. It is also possible to have build management tools along with code review and code analysis tools. With Tasktop you're able to save your coding time for your applications instead of the code analysis tools. Being able to handle the complexity involved in tool integration is critically important. With Tasktop's web-based integration designer and administration interface you can configure all your integrations with just a few clicks. You're able to visually integrate your tools landscape as you define conditional routes for artefacts to flow to specific repositories and projects, define filters, and graphically map artefact state

transitions across all toolchains. It also provides an intuitive user experience for defining and tuning the flow of information across your integration landscape.

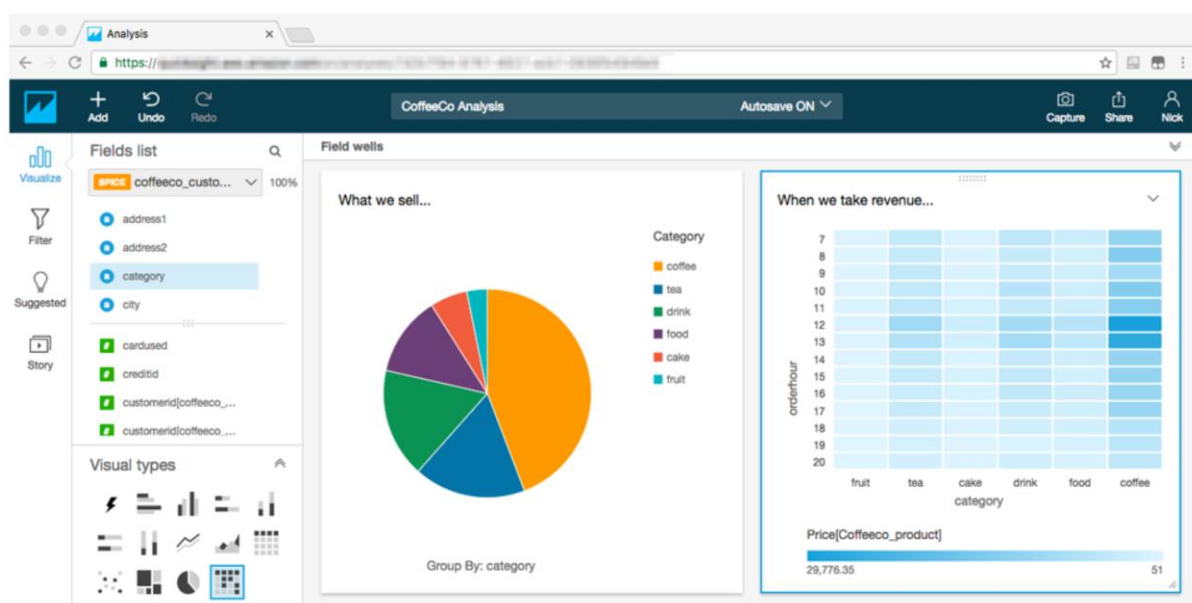
Hackystat: <https://hackystat.github.io/>

Hackystat is an open source framework for collection, analysis, visualization, interpretation, annotation, and dissemination of software development process and product data. The Hackystat Framework can be used to measure to process of software engineering. It can be used as infrastructure to support professional development, either open or non-open source by facilitating the collection and analysis of information important for quality assurance, project planning, and resource management. Hackystat users typically attach software ‘sensors’ to their development tools, that collect and send “raw” development data to a web service called the Hackystat SensorBase.

Hackystat uses client and server-side data collection to gain a more complete view of the development process. The SensorBase can be queried by other web services to form higher level abstractions of this data and integrate it with other internet-based communication or coordination mechanisms. It is also able to generate visualizations of the data, abstractions, or annotations provided.

Amazon QuickSight:

Amazon QuickSight is a fast, cloud-driven business intelligence service that makes it easy to build visualization, perform analysis and quickly gain insights from your data. QuickSight provides pay-per session pricing so you only get charged for using the data, making it possible to get no charges if you decide to abolish this method of measuring engineering. It performs visualisation based on your data, so it saves managers time in interrogating and interpreting the otherwise raw data.

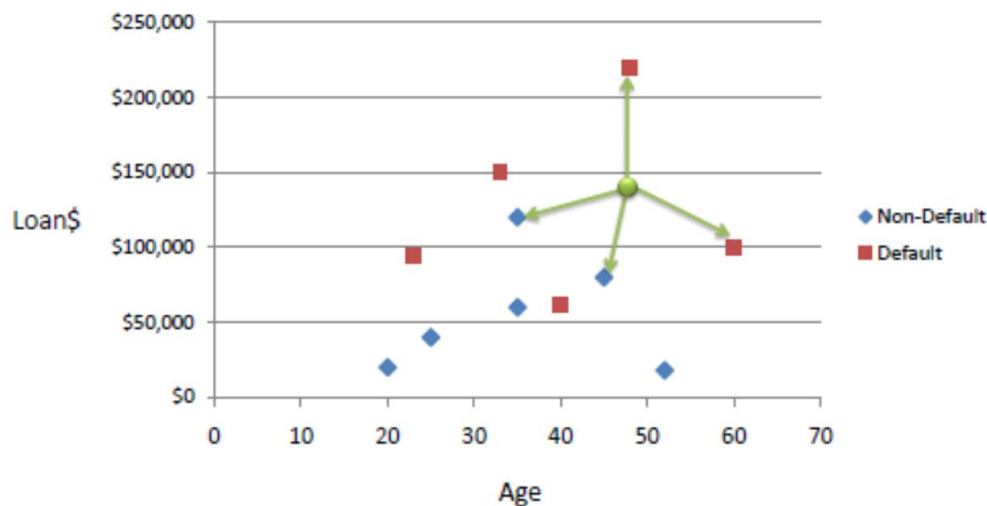


*Sample visualisation using Amazon QuickSight.*

As this is provided by Amazon Web Services, you don't need to set up your own servers for performing the analytics of potentially thousands of developers, as it is all computed in the cloud, allowing you to upscale as needed without the need for additional infrastructure.

## The algorithmic approaches available

Machine learning is a data analysis method that automates the development of analytical models. It is a branch of artificial intelligence that is based on the idea that systems can learn from data, identify patterns and make decisions with little to no human intervention.



*Sample use of K-nearest neighbour algorithm to classify an unknown case (Age=48 and Loan=\$142,000) using Euclidean distance. If K=1 then the nearest neighbour is the last case in the training set with Default=Y.*

Instead of using one of the computational methods above, it is possible for a company to analyse the data themselves using a machine learning algorithm. One of such algorithms is the K nearest neighbour algorithm, which is a pattern-recognising algorithm used for finding regression and classification. Regression is the idea of giving a set of data and finding the best relationship that represents this set of data. While classification is being given a known relationship, the goal is to identify the specific class that the data belongs to.

This is, in essence, a machine learning algorithm which uses classification based on finding the most similar data points in the training data, and making an educated guess based on their classifications.

It is often used for anomaly detection, therefore able to notice an anomaly in a software engineers work behaviour and performance. Some advantages of using this method are amongst others: increasing productivity in development, improving production system, more precise planning and responding to fluctuation in the market.

K nearest neighbour algorithm is easy to implement and highly accurate. It is powerful as it makes no assumptions about data.



A downside is that it is computationally expensive, due to the fact that it stores all its training data. The algorithm also has a high memory requirement along with a relatively slow prediction stage.

## Ethics regarding analytics

There is a large ethic obstruction to the mass collection of software engineering data, which companies must do their utmost best to not infringe upon.

### Right to privacy

Employers must take care to only monitor aspects that are relevant to the work the employee might do and not from their personal life. Employees need to know and consent to the extent of the data being collected. People are entitled to location privacy; an employer shouldn't be tracking the location of its employees without a good reason. Using location-based tracking systems affect the security and privacy of user as other apps can access your location thus harvesting your data and being able to process this data as they see fit.

The way in which this data is stored would have to be GDPR compliant if dealing with the data of EU citizens. Organizations in breach of GDPR can be fined up to 4% of annual global turnover or €20 Million (whichever is larger). This is the maximum penalty that can be imposed for the most serious infringements which includes; that they do not have sufficient customer consent to process data. It should be noted that both controllers and processors are subject to these rules, which means that “clouds” are not exempt from the implementation of GDPR. This is a powerful deterrent for companies who would otherwise harvest our data without our utmost consent and every company is now forced to implement these consumer protection laws.

**What the new EU GDPR means in 1 minute**

The EU GDPR will increase privacy for individuals and give regulatory authorities greater powers to take action against businesses that breach the new laws. Here's what it means for your business:

**Tough penalties:** fines of up to **4%** of annual global revenue or **€20 million**, whichever is greater.

The regulation also applies to **non-EU companies** that process personal data of individuals in the EU.

The **definition of personal data** is now broader and includes identifiers such as:

- genetic
- mental
- cultural
- economic
- social identity.

The **international transfer of data** will continue to be governed under EU GDPR rules.

**Obtaining consent** for processing personal data must be clear, and must seek an affirmative response. **Yes**

Parental consent is required for the processing of **personal data of children** under age 16.

Data subjects have the **right to be forgotten** and erased from records.

Users may request a copy of personal data in a **portable format**.



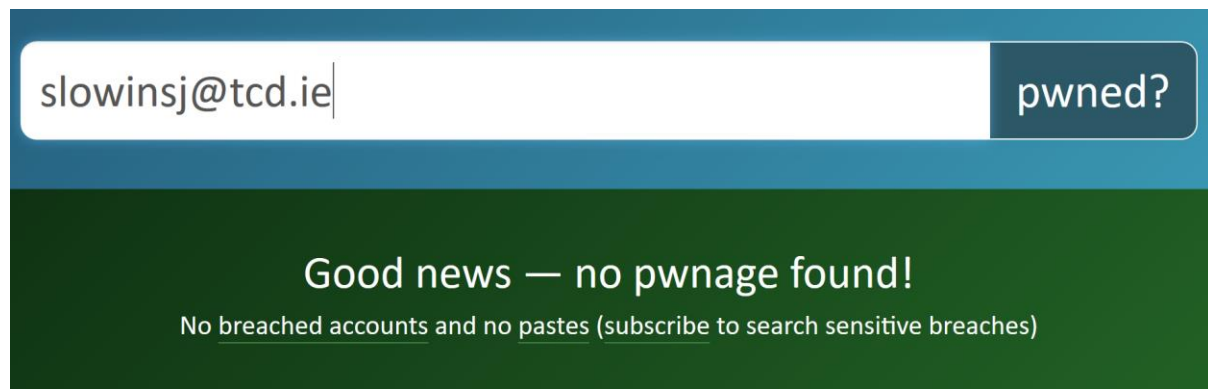
### *Infographic of GDPR and how it affects companies and the rights of users*

Using the data collected in a negative way has the potential to inflict tremendous stress on employees. As an example, the process of using stack ranking has been documented to have a negative impact upon employee morale, as well as holding potential for abusing the system alongside possibility for sabotage. Many companies have abandoned this competitive system of ranking employees in favour of systems that encourage co-operation rather than sabotage.

A fear culture could occur at a company that for instance tracked technical debt and ranked a developer against his peers. This could lead to the developer not being as productive to push code, due to the fear of making a mistake, thus decreasing the whole effectiveness of this method of analysis.

#### Security:

Security should be of utmost importance when collecting people's data. Potential unsafe storage of data is another ethical factor to take in mind. The discovering of sensitive information such as for example a heart problem, if leaked could have consequences for the victim such as increased problems in the acquiring life or health insurance. If a security breach occurs and data is leaked and or stolen, the employees and all persons affected have a right to know the degree of how this affect them, i.e. the data stolen and precautions which they might want to partake in. Due to the near inability for some companies in preventing data breaches, the data harvested should only be the data necessary not all the data the company can get their hands on. On a personal level, a website that I have found is useful is: <https://haveibeenpwned.com/>, which checks if you have an account linked to your emails which has been compromised in a data breach.



#### *Message of no breaches found on website haveibeenpwned.com*

A company should store data securely i.e. salt and hash passwords yet even as a very cautious, computer savvy user, my other email has been involved in numerous data breaches all of which were out of my control. This affects all users and some companies do not try hard enough to protect their clients. The GDPR, while “annoying users”(annoying predatory companies more) has European citizens best interest at heart, to protect us from predatory practises and other important factors such as the right to be forgotten, requesting companies to delete the data of anyone who request this, as well as the ability to request your “file” of everything the company knows and has recorded about you. This could even be numerous Gigabytes!

Companies also need a designated data protection officer, the DPO. This is similar to a compliance officer and is also expected to be proficient at managing the IT processes, data security (including dealing with cyberattacks) and other sensitive business continuity issues around the holding and processing of personal and sensitive data.

## Ending

The ability to track productivity is still in its infancy, yet many individuals have strong arguments for or against it. In my opinion, if used correctly, it is a great way for an individual programmer to decide where he could be more efficient, and it could lead to the possibility of self-improvement on the developer side. I believe that there is no single all-in-one application which can harvest and display all the data detailing a software engineers productivity, which means that every organisation must decide to use the appropriate algorithms on an ad-hoc basis, as different people might view different metrics in higher regard to others. A basis of trail an error would be useful to finding the right platform to analyse and measure the engineering process, but in the end settling for one (or more) is a necessity.

## References

[https://resources.sei.cmu.edu/asset\\_files/WhitePaper/2013\\_019\\_001\\_296863.pdf](https://resources.sei.cmu.edu/asset_files/WhitePaper/2013_019_001_296863.pdf)

<https://www.altexsoft.com/blog/engineering/what-software-quality-really-is-and-the-metrics-you-can-use-to-measure-it/>

<http://csdl.ics.hawaii.edu/research/hackystat/>

<https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7>

<https://eugdpr.org/>

<http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/>

<https://gdpr-info.eu/issues/right-to-be-forgotten/>

<http://gdprandyou.ie/data-protection-officer/>