



---

# **CSU330312 Software Engineering**

## **Measuring Software Engineering Report**

Liam Junkermann - 19300141

December 27, 2021

---

### **Contents**

<b>1</b>	<b>Measurable Data</b>	<b>2</b>
	Commits and code frequency . . . . .	2
	Testing Coverage . . . . .	2
	Pull Request creation or review frequency . . . . .	3
	Task and time tracking systems . . . . .	3
	Health and Activity Tracking . . . . .	3
<b>2</b>	<b>Connecting and Performing Calculations on Activity Data sets</b>	<b>4</b>
	Integrated Platforms . . . . .	4
	Custom Pipelines . . . . .	5
<b>3</b>	<b>Computation to profile software engineers</b>	<b>7</b>
<b>4</b>	<b>Morality, Ethics, and Legality of Collecting and processing engineer measurement</b>	<b>8</b>
	<b>Conclusion</b>	<b>10</b>

## Introduction

Software engineering is a notoriously difficult to track for time and productivity. Some engineers may be more efficient with some technologies and not others, and tasks may have varying difficulty making engineering activity and productivity particularly difficult to accurately and consistently track productivity. This report will discuss various ways engineering activity can be measured, appropriate data collected, and then processed to accurately profile engineers. Finally, the moral, ethical, and legal consequences of these methods will be explored.

## 1 Measurable Data

There are a number of metrics which can be used to determine engineering activity depending on the way a given repository or organisation is set up.

### Commits and code frequency

The simplest way to track engineering productivity could be based on the commit and code frequency charts provided by most version control providers. These charts allow a supervisor, or even the engineer themselves, to see the number of commits they make over a period of time, as well as the number of lines added or removed from the files within the repository. This method of measuring engineering activity is rudimentary and does not account for tasks which may take more time to research, plan, or structure – all subtasks which would not be reflected in a simple commit or code frequency chart. In a team it is incredibly important for team members to make and label commits appropriately in order to allow other team members to see which tasks are being completed, how they are being completed, and to more easily find the necessary changes in lines or files during code reviews.

### Testing Coverage

An important part of any project which looks to see a production environment is the testability and rigorous testing to ensure that the solution can run effectively in that production environment. Any solutions a developer writes must be tested, and many production pipelines provide engineers graphs or statistics about code coverage – as in which lines of code were tested and which were not. These statistics allow engineers to catch bugs and ensure their solutions is being tested appropriately.

### **Pull Request creation or review frequency**

Version control is used to allow many people to work on a project together at the same time. Organisations will use pull requests to combine the work of various engineers in order to begin the production pipeline (the flow within which the code is built and packaged for release to production or testing groups). While commit frequency and volume are important to track activity, and break down tasks, ultimately contribution to production environments is what dictates success and progress. Pull requests are the measure for those production contributions, they allow the engineer who has worked on a feature or task to share that work with their team, and allows the rest of the team to make comments about the functionality or approach used to solve the problem presented in the task or feature. It is important, for both the team and the product they are working on, that the new features sent to the production pipeline will work and solve the problem stated. Tracking engineering activity through pull requests is an effective way to track both individual engineering contributions to a production product, and the contributions from other team members through the code review process.

### **Task and time tracking systems**

One of the more effective and holistic ways to track engineering activity and performance leverages and expands upon the previous two solutions. Using a system like JIRA, Azure Dev Ops (ADO) or even certain functions of GitHub allow teams to track tasks that need to be completed and track progress through tasks broken down to different degrees of complexity, as well as link progress with commits and completed pull requests. This method of measuring engineering performance allows progress on tasks to be tracked in reference with actual effort and hours of work put into a given task or subtask. This also allows the management of the team to improve as time estimation becomes more accurate with iterations of using this kind of solution.

### **Health and Activity Tracking**

There are many health tracking solutions on the market today which can be used to track engineers daily activity, sleep effectiveness and hygiene, among a myriad of other potential metrics to track. These health tracking solutions can provide insight into how certain habits affect overall health acutely – on a day-to-day or "once off" basis – or more chronically, effects that may build over time. Collecting and understanding this data can be very helpful for some people to see and change their behaviour to

achieve a more balanced life, which can only improve productivity.

Health tracking options include common and integrated devices like Apple or Samsung watches, these watches allow users to very easily track metrics like sleep, day to day activity, exercise distance and intensity metrics, and some other health metrics depending on the model, all while enhancing the users experience and productivity with their attached mobile devices. These watches can serve engineers with reminders to drink water, stand up and walk around (particularly important given that most software developers spend many hours a day sitting), and even can have reminders to eat or drink if that is something a user may struggle with.

There are also more targeted devices which can be used, such as an Oura ring or Whoop strap. These devices are much more targeted in their intended use, and carry a different price tag accordingly. The Oura ring is geared toward analysing and improving sleep, reportedly being nearly as accurate as a full scale sleep lab in analysing your sleep. Boasting a high accuracy in measuring sleeping heart rate, heart rate variability, respiratory rate, body temperature, with more features such as sleeping blood oxygen coming soon, this option allows engineers to optimise their sleep and habits around sleep to maximise their productivity and recovery every day, particularly if they engage in consistent high intensity exercise. The Whoop Band is much more geared toward athletes. Whoop is a low profile strap worn around the wrist or bicep (depending on the user's preference) which records heart rate throughout the day and night, with the intention of being worn, like the Oura ring, 24/7. The data from the strap is then uploaded to the Whoop app, before then being analysed to determine the athlete's strain throughout the day. Whoop would be an especially good tool for engineers who exercise and are looking for an activity tracking solution which requires minimal effort. Whoop and Oura have the functionality to share activity data which is more universally accessible than the smart-watch options noted above which may be useful for an organisation looking to build an engineers profile using their activity data. The morality and legality of this will be discussed later on in this report.

## 2 Connecting and Performing Calculations on Activity Data sets

### Integrated Platforms

**Code analytics** There are various ways to connect and collect activity data with platforms which can analyse that data, many times these platforms – which handle collection and analysis of code productivity

data – are the same. Platforms like GitHub, JIRA, and ADO provide many charts and analytics options for specific repositories. GitHub also provides an API which allows anybody to view the commit, PR and Code review statistics of any given user. These can be collected and passed to other analytics services or to a custom service or pipeline which executes analytics on datasets. These integrated systems also produce data and graphics which can be used to determine the time it takes to complete a given task or the progress of tasks and epics through commits and PRs. Utilising a complete system like this (or with native integrations like JIRA and Bitbucket), can streamline the process of finding collecting and performing analytics on a set of data.

## Custom Pipelines

**Code Analytics** Popular version control providers, including many self-hosted solutions, provide APIs and dataset hooks to access data. This allows for many data analytics options. There are a few options for data analytics solutions.

**Excel** A basic Excel spreadsheet can be used to generate charts and data visualisations that may be used to profile an engineer. Excel also allows for data to be used offline and manipulated more interactively than the integrated platforms allow.

**Pandas Data analysis** Pandas is a python data analytics library which allows the users to manipulate and complete operations on data in a more extendable way than Excel. It also can be used alongside libraries like Matplotlib to generate charts and show any relationships found within the Pandas datasets. Pandas allows users to leverage skills they may already have in programming to create charts and regressions similarly to excel. Pandas can also be used to build an automatic pipeline that can be run to produce dataset outputs for other visualisation solutions. This automation allows managers to get updated statistics about an engineer on a regular schedule to provide feedback if needed with updated data from any number of sources. Potential sources could include data from the version control provider used by the organisation, any available health and activity data, as well as any other data collected by the organisation. The ease with which this data can be collected programmatically from various sources makes this analytics option particularly attractive if multiple platforms are used to track time, tasks completed, code contribution (commits and PRs), health and activity data, and any other data an organisation may find relevant in building an engineers profile.

**General Health Tracking** Beyond obvious productivity and time tracking requirements of any business, some organisations may choose to track health metrics, as noted above. A separate pipeline and process would need to be used in order to both connect to the relevant platforms, and extract the necessary data from those platforms in order to perform calculations.

Platforms like Apple Health and Samsung Health require integrated solutions in order to access their respective health datasets. In cases where engineers would use Apple or Samsung Watches, the organisation would need to build purpose built apps in order to get more complete and seamless access to the data these devices produce. There are methods to manually export the data, but these methods are cumbersome and add a barrier to performing calculations on the data. Ideally a process involving an Apple or Samsung product would be combined with an app to allow organisations to more seamlessly collect the agreed data from the relevant devices before then performing the necessary calculations in conjunction with the coding data listed above.

If the engineer is using an Oura ring, Oura provides an API with which the organisation can interact in order to access a collection of metrics which Oura exposes. Oura does not grant full access to the data (like the Apple or Samsung solutions would allow locally), but does give access to the Readiness, Sleep, Activity, and Bedtime statistics which users can see from within the app. The overall scores and some supporting scores and metrics are shared through the API and can still be used to track health outputs in relation to productivity using a solution like Pandas listed above, in a seamless and relatively private method.

Finally, if an engineer uses a Whoop band, there is no official API available. There is an unofficial API that could theoretically be used, but with no official API a third party could not be authorised to access the data like the Oura ring and watch solutions allow. That being said, this unofficial API could still be used by an engineer looking to find their own correlations between the various metrics discussed so far. The unofficial API provides many more metrics than Oura, as well as the scores generated by the Whoop system. Providing enhanced functionality over the two previous solutions.

All these solutions provide engineers or their organisations methods with which to track raw engineering productivity with health metrics to statistically show a balanced, healthy life to promote healthy sleep and activity habits. Including Health metrics as part of a review method either through organisation policy or for an engineer's curiosity is possible with each of these potential solutions. There are also third party solutions which allow users of these products and many more to create their own health dash-

boards which could no doubt be integrated into other dashboards which share engineer code productivity metrics and analytics.

### 3 Computation to profile software engineers

There are a variety of ways to build effective profiles of a given software engineer. Using a combination of the platforms to collect and perform analytics on the various datasets an organisation can build a score or profile for their engineers. Software engineering is not just about constantly producing code (regardless of the quality). Other metrics which may be more difficult collect and quantify such as the ability for an engineer to work in a team and collaborate, or the health, activity, and sleep metrics that one might not see as being relevant to the work day. A collection of these statistics allows an organisation to build a more complete picture of an engineer that would not be possible otherwise. An engineer's profile should be personalised as each engineer would have different priorities both for their career and life. Some engineers may prioritise working in a team and participating in code reviews – either as a result of their position or their skills may be more suited to that – while other engineers may find it easier to churn out work-product until a sprint or task is completed, and code reviews may be the last thing they get to, finally you might find an engineer who switches between each quite seamlessly. Each of these engineers should not be profiled the same way so any profile or scoring system should allow for those priorities to be judged proportionally. Next, different engineers have different expectations of work-life balance. As remote work becomes more common through Covid, especially in software engineering, employees are able to build their own schedules more easily resulting in some unconventional working times depending on the engineer and organisation policy. For example, an engineer working in Europe may need to correspond with a team working on the US west coast and may find that shifting their work hours to be later in their day allows them to more efficiently communicate with the team in the US. As a result they may come "online" later than some of their European co-workers. This work-time flexibility allows engineers to explore other hobbies and potential activity outside work – for example sport. Having access to activity and health metrics allows these engineers to maximise their daytime to balance work and their activity while also maximising sleep performance. A subjective score may allow an engineer to personally judge and maximise their own life. It is important to encourage engineers to live a life outside work. Software engineering is a field where many engineers can fall into the trap of working all the time to hit deadlines under excess stress. While it can be hard to objectively track

mental health, trends in activity, sleep, and other collected health metrics can allow an organisation, or the engineer themselves, catch or validate when they may be struggling with mental health.

Through the collection of different types and sources of data combined using a solution outlined above, an organisation or engineer can strong profile made up of the various derived scores with the relevant weightings depending on the engineers preferences.

## 4 Morality, Ethics, and Legality of Collecting and processing engineer measurement

All the data that is collected and processed is highly privileged and private to the engineers and organisations it is vital that the information is collected and processed in a robust and secure way regardless of the morality, ethics, or legality of the individual parts of the data. The system which is used to collate and process all the metrics used in analytics would ideally be run entirely on the organisations internal infrastructure or personally by the engineer. This helps mitigate any moral or legal issues of the data being leaked as a result of an unsecured pipeline. Next, the only data that should be saved (encrypted at rest), should be the raw derived scores and the weights which the engineer has selected. That way the only data that is accessible does not contain all the supporting personal data, again limiting the possible fallout if the data was leaked. Each of the metrics collected has a different set of moral, ethical, and legal implications for its collection and processing

**Repository Data** All commits, code frequency, testing coverage, pull request creation and review statistics are openly available to any member of the team on the git repository, and more than likely anybody in the organisation and potentially even the public. Many git repositories are shared across teams in an organisation as some teams may rely on that repository within their own projects. There are no major moral, ethical, or legal consequences for collecting this data beyond mainting the same level of security that the original repository has.

**Task and time tracking systems** This feature may differ from organisation to organisation. Some organisations may track time based on productivity expecting a certain number of "engineering hours" each day from their engineers. Organisations that follow this method normally use platforms like JIRA or Azure Dev Ops (ADO) to track estimated time of completion for a task and then actual time of completion will be tracked. These kinds of statistics can be exported to various



dashboards and viewed by most members of the team as time tracking statistics are wrapped up as part of feature and task specifications in these systems. These statistics should again be collected and processed with the same level of security as the task and feature boards are, keeping view access limited to the team or the manager level depending on the organisations structure. For organisations that may use a separate time tracking structure, the same privacy limitations should be expected, and the data should be handled or anonymised accordingly.

**Health and Activity Tracking** This data is significantly more privileged than the previous two sets of data as it includes highly personal and constant statistics about the user and their daily habits. Having an organisation ask their employees about their health and activity habits is not an entirely new concept, but it is highly regulated. This is because normally it is not the business of a manager or co-worker to understand the intricacies of one's health history and life as it does not affect work. These kinds of details are normally collected to provide health insurance or similar services and only a select group of people would be allowed to see this kind of information. That being said, the health information normally shared with an organisation rarely contains the breadth and depth of data which would be available as a result of using one of the platforms listed above. There are many moral, ethical, and legal implications involved in collecting and processing this kind of data:

**Moral and ethical implications** When an organisation wants to ask for this kind of information it needs to be made clear that divulging this information will not affect the engineers standing in the organisation in any way. Therefore, it is easier if this data not be included in any scores or profiles given to the engineers from the organisation, and rather be available as an option for the engineer to add to the score or profile themselves if they feel it is appropriate or helpful.

**Legal implications** Legally there is precedent for organisations to collect an employees personal health information beyond the standard questions asked on behalf of a potential insurance provider. With the rise of telecommuting due to the COVID-19 pandemic companies have begun to use products which were originally geared toward improving telemedicine to monitor employee health through the pandemic. The primary legal concerns surround GDPR with regard to the storage and the security of the data transfer mechanisms. Ensuring data

security end-to-end will help mitigate any potential legal issues assuming employees of the organisation are open to having this data included in their engineering scores and profile.

## Conclusion

There are many interesting metrics, statistics and relationships which can be explored in relation to or using various methods of data collection and production, whether it be raw code related data or health and life data. These data collections can be used to create custom scores and metrics for an engineer which can be used personally to enhance daily productivity both in and out of work, or by the organisation to ensure that their employees are living balanced lives. The implications of collecting some data vary, but all data must be secured and handled appropriately to maintain an environment of trust between the organisation and their engineers. Software engineering can be a very stressful field to work in and combining raw productivity data, in the form of code related analytics described above, with other nonwork related data, activity data listed above, can allow organisations to measure themselves and their engineers and implement policies to improve productivity by improving the health of the engineers and in turn their work environment.