Jenna Lake 18303391

CSU33012 Software Engineering

Prof. Stephen Barrett

27 November 2020

Measuring Software Engineering

Measuring software engineering is a problem that is coming to light as software engineering has become a huge industry and is being developed in larger scale. Other industries have ways of tracking work, especially tangible products that are being produced. Now that software is mainstream, there needs to be a way to effectively measure it like any other field. This is hard, given the combination of qualitative, quantitative, and subjective data that could be considered. The definition accepted for software productivity is the "ratio between the functional value of software produced to the labor and expense of producing it" (Sidler, 2002). Basically, the value of the output compared to value of input. This is still a very general metric, as the output has no set way of being calculated.

Measurements

The naïve approach to measuring software engineering is measuring the lines of code (LOC) that each person contributes. This is how productivity used to be measured, but it was realized that it would not effectively give insight as to how productive an engineer was. As well as not being a good measurement it could potentially decrease productivity if people just add more

lines of code and make their programs as long as they can and potentially get rewarded for this by their employer for seeming more productive. It would be incentivizing the wrong aspects of coding and cause further inefficiency because programmers would be wasting their time writing unnecessarily long code rather than writing meaningful code in as few lines as possible and being able to add value to their current project or move on to a new project. Also, there are other things to consider when using such a simple measurement, like that LOC are not universal for all languages, for example, an assembly language will have far more lines than the same program in another language, say Python or Java (Parareda and Pizka, 2007). Lines of code can be used as a simple, cheap and easy measurement, but to get a better and more in-depth metric of software engineering, there are other measurements that can be used. To effectively measure software engineering, you must use more complex metrics, as being a software engineer is far more complex than writing lines of code.

Measuring software engineering is moving from quantitative metrics, such as number of lines of code, to qualitative metrics. The problem that is arising from these types of metrics is that there is no set way to measure them. It is a less straightforward measurement but can explain much more about the code quality and engineer productivity. The definition for software productivity mentioned previously is hard to measure because things like functionality and quality of code are not able to be counted or measured directly. Different engineers and their managers might have different definitions of functional and quality code and different things they prioritize in their measurements.

With the technologies now readily available to anyone, it is easier to find other ways to measure software. Using these tools, it is possible to access different kinds of data that they are

recording and giving back to the user. You can see an overview of the contributions of each person, including number of commits, lines of code added or taken away, when they were contributing most, and how many different files they are contributing to. As stated by Fenton and Neil (1999), “our objective is to handle the key factors largely missing from the usual metrics models: uncertainty and combining different (often subjective) evidence.” There are all different types of data being collected constantly on cloud source control tools, which can be used together in different ways to find specific and personalized ways of measuring productivity based on what metrics managers think are most relevant to what they are trying to accomplish with their measurements.

There are plenty of metrics as well that should be considered that are not so much software related but workspace and environment related as well. Tools to collect data about someone’s personal productivity and workflow are being used in many companies. Not thinking about the actual work that is being produced, managers need to think of the work environment as a social system and how it affects the workers’ productivity. How employees interact with each other and how they are influencing each other is important in understanding other aspects of each person’s productivity (Wei Pan et al., 2012). Things like how well employees are working in teams, how much effort are they putting in, and how focused they are is being tracked in every type of work environment, not just in software development. Data reported on how many tasks are being completed using project management tools, what employees are spending their time on during the day, whether it is emailing, bathroom breaks, or focusing on work, and many other things can be used to measure efficiency and productivity of workers. These measurements can be taken into account as well, alongside how efficient and productive the

actual code they are producing is to have a complete overview on the engineers productivity to efficiently measure software engineering.

Platforms

These platforms all track the types of measurements talked about in the above section. They record all the data, including git data, and give it back to the user so they can use it as they would like so they can get a complete understanding of how their team is doing. The platforms, as well, have their own algorithms and analyses of the data that they will produce and show to the user so they can make recommendations and so that they do not have to analyze the data themselves if they do not want to.

A platform that is designed specifically for software development team managers trying to effectively measure how their team is performing is Pluralsight Flow. The four headers they focus on providing information for are Code, Review, Upskill, and Collaborate. (Pluralsight.com, 2020) The headers each return different data that has been collected and sectioned to provide easy access for the manager. Flow measures not just the code, but they measure the proficiency of each coder in different languages, as to show each person's strengths. This data is provided to "connect the dots between code and skill" (Pluralsight.com, 2020), which is what most measurements are trying to do as they move away from measuring LOC. They show project timelines and a breakdown of what is accomplished during different sprints or periods of time. They assess how the team is coding and how they are working together, including each person's contribution to code and who interacts best with who. They return all this data, as well

as give recommendations on how to make the projects flow better or the teams work more effectively together.

Waydev is another platform that tracks how software development teams are working. For software engineers, the data Waydev focuses on is under the sections Work Log, Daily Update, Teams Stats, Team Compare, Timecard, Inactivity, Sprints. Each section gives different data and different insights into the work being produced and how it is being produced. A particularly interesting one is Team Stats. It is a feature that allows you to view the specific data you want and get customizable performance reports showing whichever stats the manager chooses and have them displayed in different ways, making it quick and easy for managers. According to their website, waydev.co (2020), they allow managers to "Get the complete visibility of [their] team. With a data-driven story of how everyone is doing."

Code Climate's Velocity is a third platform that measures software engineering. Their website presents the three advantages of using Velocity. The first being "End-to-end visibility into continuous delivery", showing the advantages of collecting pull request data, and using it to show whether productivity is trending up or down. Advantage two is "Full context on where your team is working", which shows how the platform manages code reviews and making sure work is distributed equally. Advantage three is "Flexibility to focus on your team's unique challenges", which highlights how the platform takes into account how everyone is different and the adaptability to compensate for this. (Codeclimate.com, 2020)

Not necessarily a platform, but a framework that software engineers can use personally to keep themselves working productively is the Personal Software Process (PSP). PSP is based on "to be

most effective, engineers must plan their work and they must base their plans on their own personal data" (Humphrey, 2000). The platforms talked about previously were all dealing with teams and bigger scale projects, where PSP works on a smaller level. To have a productive team, you must have productive members. PSP is a way for each software engineer to be personally responsible for and keep track of their own productivity so they can effectively add the most value to groups.

Tracking general productivity in the workplace involving employee health, emotions, and whereabouts is becoming increasingly popular also. " 'Technological advancements in several fields—big data analytics, communications capture, mobile device design, DNA testing, and biometrics—have dramatically expanded capacities for worker surveillance both on and off the job' " (Wartzman, 2019) . Employees are being tracked now in every aspect of their lives while they are at work. Video surveillance is used, along with other tools. For example, Vibe monitors Slack interactions between employees to track the general moods across the company (Sabel, 2017). Three Square Market had even gone as far as having a few employees volunteer to get chips in their hands to monitor their work. It has not been deployed throughout the company (Spicer, 2017). Amazon has wristbands as well for similar purposes and also does not use them throughout the company, but they are still there if they ever need them (Miller, 2018). Technology is available to any company that wants to monitor practically any portion of their employee's workdays and even lives, to an extent. This type of data is being used to keep employees on track and keep the companies running efficiently.

Algorithms

Data is always being collected, but how the manager chooses to use the data will determine whether it becomes valuable to them. Data alone is not helpful in determining productivity, but when evaluated properly it can be an accurate measurement of software engineering.

Using the LOC metric to measure software engineering is not just counting how many lines each person has written. When calculating what lines are counted and how they are counted, there's a few guidelines. The first being only counting Redundancy Free Source Lines of Code (RFSLOC) – which excludes repeated or unused code, keeping only the important code being analyzed. Then, RFSLOC per working day are measured to get an accurate measurement with a time constraint as well. This gives the metric more context and makes it more applicable to using it as a viable measurement. While this measures productivity, the formula of defects per RFSLOC is used to indicate quality. These algorithms used together can be used to follow whether enough code is being written per day, and whether the code is of a certain quality. This prevents lots of poor-quality code being inputted as well as stopping only a small bit of working code (Parareda and Pizka, 2007).

The three main platforms talked about above all include forms of analyzing the activity in version control to learn about the groups. Using kinds of network analysis techniques, the data from here can be shaped into indicators for productivity as a group, which is a feature those three all have different variations of (Dittrich, 2013).

From the previous definition of software productivity, the value of input does not have a set definition or formula to find it. This allows the method of evaluating it to be up to the managers

to decide how it is measured. However, artificial intelligence is starting to play is role in how it is measured. Using AI, the massive amounts of data available now can be analyzed and metrics can be figured out automatically. Systems like neural networking systems for pattern recognition can be used to go through the data and see patterns that emerge and use these relationships between data to create value measurements. Automation and the ability to look at larger amounts of data give AI developed measurements the advantage over other measurements that have been developed. Whereas older techniques involving returning pre-defined information based on the data (like expert systems could produce), machine learning allows the system to learn from the random data that has been collected and create the patterns that may have not already been realized and put together more complex relationships between the data, creating more valuable algorithms (Council, 2018).

Ethics

With the advancements in technology, data privacy has never been a bigger issue. Being monitored so closely by employers is a very new thing. Obviously, people don't always trust it right away and might not be comfortable with it. Personally, I do not think this type of monitoring is a big deal, within bounds. When it comes to the git analytics, it is like tracking any other kinds of work. Software engineering is far behind other work areas in the world of productivity tracking. Watching activity on git is ethically the same as keeping tabs on the work any other employees are producing. As for other types of data being stored about employees, personally I would not mind if it was me. However, younger generations are generally desensitized to data collection because of growing up in the digital age (Wartzman, 2019). Even

though younger employees might not mind, there are still lots of things that can be done with the amount and depth of data that is stored, that most people probably don't even realize.

Companies cover their methods of tracking with the claims that they have good intentions, only to improve work environment and productivity. But data that was intended to be used for company improvements could be used against employees in the end. Humans are too complex to be stripped down to data and the assessment of their value to the company left to algorithms' perceptions of them as employees. There are intangible aspects to what someone brings to a work environment and the type of work they are doing that cannot be measured completely accurately.

The issue of ethics in tracking employees work is not black and white, it seems like a good idea in theory and could be used in so many ways to help employees and their companies. But things can easily go wrong. Like in the case of Myrna Arias, who got fired for deleting an employee tracking app (Pagliery, 2015). Also, laws don't always give full protection against every issue, especially if it is a grey area or a new issue that has never been dealt with before. "People don't realize that there aren't a lot of laws protecting privacy in the workplace. Part of the issue is that employees may not even be aware of how all the data is being tracked and how it's being used." (Miller, 2018)

With the integration of technology into every part of our lives, monitoring like this was bound to happen at some point. What is needed is full coverage laws around all types of scenarios. There is the GDPR in Europe, but the US has nothing specifically dedicated to data privacy like GDPR. The monitoring technology just needs to be kept in check by laws and it could be ethical

and benefit everyone involved. “We live in an age of ‘biocratic capitalism’ where the economy is based around harvesting one commodity – data about our everyday lives… Implanting chips under the skin of employees is just one more way that firms can continue to generate the resource they need most – data about ourselves” (Spicer, 2017). Although there may be ethical concerns, I think this is the way society and the economy will be going forward and there is no stopping or escaping it so the only way to ensure it stays ethical is to enact laws that regulate data privacy.

Bibliography

Codeclimate.com. (2020). Understand Each Step of Software Development Life Cycle | Velocity. [online] Available at: https://codeclimate.com/velocity/understand-diagnose/#report_builder [Accessed 23 Nov. 2020].

Council, G. (2018). Machine Learning AI vs Expert Systems AI. [online] Parascript. Available at: https://www.parascript.com/blog/machine-learning-ai-vs-expert-systems-ai/ [Accessed 24 Nov. 2020].

Dittrich, Andrew, Mehmet Hadi Gunes, and Sergiu Dascalu. (2013) "Network Analysis of Software Repositories: Identifying Subject Matter Experts." Complex Networks. Springer Berlin Heidelberg. pp. 187-198.

Fenton, N. E. and Neil, M. (1999) 'Software metrics: successes, failures and new directions', The Journal of Systems & Software, 47(2), pp. 149–157. doi: 10.1016/S0164-1212(99)00035-7. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.2683&rep=rep1&type=pdf

Fenton, N. E. and Neil, M. (Sept.-Oct. 1999) "A critique of software defect prediction models," in IEEE Transactions on Software Engineering, vol. 25, no. 5, pp. 675-689, doi: 10.1109/32.815326. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.548.2998&rep=rep1&type=pdf

Gitclear.com. (2020). Measuring Developer Productivity: A Comprehensive Guide for the Data Driven - GitClear. [online] Available at: https://www.gitclear.com/measuring_developer_productivity_a_comprehensive_guide_for_the_data_driven [Accessed 24 Nov. 2020].

Humphrey, W. (2000). The Personal Software Process SM (PSP SM). [online] Software Engineering Institute. Carnegie Mellon University. Available at: https://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_13751.pdf [Accessed 23 Nov. 2020].

Miller, A. (2018). More companies are using technology to monitor employees, sparking privacy concerns. [online] ABC News. Available at: https://abcnews.go.com/US/companies-technology-monitor-employees-sparking-privacy-concerns/story?id=53388270#:~:text=US%20tech%20company%20offers%20to%20implant%20microchips%20in%20employees&text=For%20example%2C%20a%20recent%20patent,its%20employees%20in%20July%202017. [Accessed 24 Nov. 2020].

Nguyen, P. T. et al. (2021) 'Convolutional neural networks for enhanced classification mechanisms of metamodels', The Journal of Systems & Software, 172. doi: 10.1016/j.jss.2020.110860.

Orosz, G. (2020). Can You Really Measure Individual Developer Productivity? - Ask the EM. [online] The Pragmatic Engineer. Available at: https://blog.pragmaticengineer.com/can-you-measure-developer-productivity/ [Accessed 24 Nov. 2020].

Pagliery, J. (2015). Woman fired after disabling GPS on work phone. [online] CNNMoney. Available at: https://money.cnn.com/2015/05/13/technology/fired-gps-app/index.html [Accessed 24 Nov. 2020].

Parareda, B.M.Y. and Pizka, M. (2007). Measuring Productivity Using the Infamous Lines of Code Metric. [online] Inestra. Kaufering, Germany: itestra GmbH. Available at: https://itestra.com/wp-content/uploads/2017/08/07_itestra_measuring_productivity_using_lines_of_code.pdf

Pluralsight.com. (2020). Work on What Matters. [online] Available at: https://www.pluralsight.com/product/flow/code [Accessed 23 Nov. 2020].

Sabel, J.-M. (2017). Slack Can Now Monitor Employee's Emotions. [online] hirevue.com. Available at: https://www.hirevue.com/blog/hiring/slack-can-now-monitor-employees-emotions [Accessed 24 Nov. 2020].

Sidler, R. (2002). *Software Productivity*. [online] www.umsl.edu. Available at: http://www.umsl.edu/~sauterv/analysis/488_f02_papers/SoftwareProductivity.html.

Spicer, A. (2017). Surveillance used to be a bad thing. Now, we happily let our employers spy on us. [online] the Guardian. Available at: https://www.theguardian.com/commentisfree/2017/aug/04/surveillance-employers-spy-implanted-chipped [Accessed 24 Nov. 2020].

Wartzman, R. (2019). Workplace tracking is growing fast. Most workers don't seem very concerned. [online] Fast Company. Available at:

https://www.fastcompany.com/90318167/workplace-tracking-is-growing-fast-most-workers-dont-seem-very-concerned [Accessed 24 Nov. 2020].

Waydev.co. (2020). Git Analytics Platform for Engineering Productivity - Waydev. [online] Available at: https://waydev.co/ [Accessed 23 Nov. 2020].

Waydev. (2020). Pluralsight Flow (ex Gitprime) vs. Waydev vs. Code Climate: Tips to Choose the Best Engineering Productivity Software - Waydev. [online] Available at: https://waydev.co/gitprime-vs-waydev-vs-code-climate/ [Accessed 24 Nov. 2020].

Wei Pan, Wen Dong, M. Cebrian, Taemie Kim, J. H. Fowler, and A. S. Pentland. (March 2012) "Modeling Dynamical Influence in Human Interaction: Using Data to Make Better Inferences About Influence Within Social Systems." IEEE Signal Processing Magazine 29, no. 2: 77–86. Available at: https://dspace.mit.edu/bitstream/handle/1721.1/92443/Pentland_Modeling%20dynamical.pdf?sequence=1&isAllowed=y