Tanmay Kaushik
18308341
Software Engineering
20 November 2020

Measuring Software Engineering

Over the last few decades, software engineering has become one of the most popular fields that people are working in. A survey tells us that there are over 40 million users of Github with more than 190 million repositories. This data is enough to tell us that there are a lot of software engineers out there with a lots of lines of code. In this competitive market of software engineering, measuring a person's or a team's ability to develop software has never been more needed. There are a lot of companies and organisations who are doing this. But there are some questions that must be answered regarding this practice.

How can software engineering be measured?

It has been established that the software engineering has to be measured. But what should the metrics be to measure that?

Length of Code:

The most basic approach to measure software engineering is seeing the length of the code the engineer has committed. This approach may be applicable in a different fields - like measuring the success of a salesman by comparing the sales figures - but, comparing the length of code is not a realistic measure of efficiency and accuracy.

Writing optimized and readable code is a preferred in software engineering so this approach may impact the developer in a negative way.

Hours Worked

Another simplistic measure could be a record of how much time an engineer or a team spent towards developing a specific part of the project. This measure has several disparities like who should decide the number of hours that needs to be spent. Another problem with this approach is the impact it has on the developer because of the pressure created because of deadlines and limits.

Number of Commits:

This approach can be a good measure to check the accuracy of the code. However, along with the number of commits, one should additionally measure the impact of the new code -

check if it creates conflicts, how deep the change reaches the rest of the code. It should also measure the size of the commit as one developer may add many changes in one commit or add small changes many time.

Complexity of the code (McCabe Number) :

A highly complex code can be very difficult to understand for people other than the creator. This could result into a big problem for people working in a team when developing software thus inhibiting the process. Therefore, we can measure a good software code by McCabe Number - which is a measure which quantifies the complexity of a software program. This number is determined by measuring the number of linearly independent paths through the program. A good code would have a low cyclomatic complexity.

Ratio of lines of Code to lines of Test :

A good measurement of software engineering could be the number of tests that the software passes, and if all the edge cases are passed and if the code is reliable. I believe that this approach is the one of the best among all because testing is one of the most important part of software development.

Contributions in Team, Discussions and Forums :

A person's ability to communicate, involvement in discussions and development groups and their participation in various forums could be used as a measure too. In a team, all person's ability to working with their partners and communicating their ideas with each other is really important for a team to perform well. However, in my opinion, this measure would not accurately measure the person's development skill. This could be an additional measure but not the only one.

Technologies and Platforms used to measure it

We have reflected upon the data points that can be used to measure a person

To solve this problem, there are many companies which have made technologies to measure software engineering and improve team efficiency -

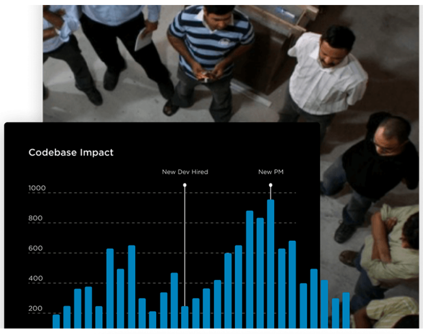
Flow by Pluralsight

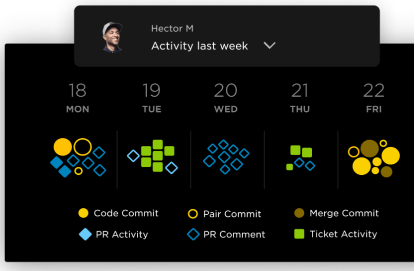
This product takes data from code commits, tickets, pull requests of the team and visualises workflow and shows where conflicts and problems exist. It sees the behaviour of an individual and the team and analyses that data and suggests sprint plans and also summarises how an individual participates in the team.

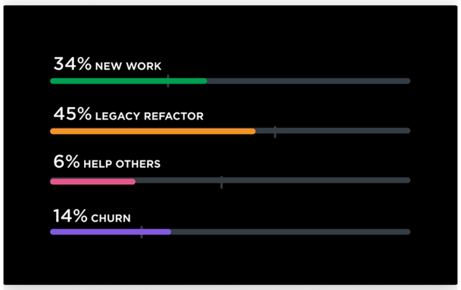
Core Features -

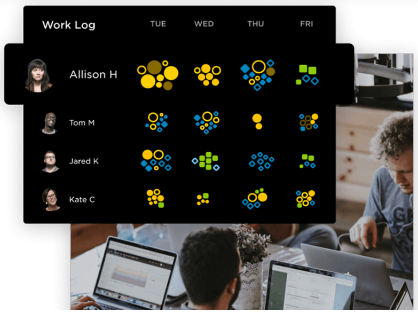
- Measures programmer's efficiency by analysing how many times the code written by the individual is rewritten.
- Measures how much new code and work has been done by the programmer by recording the number of commits, number of days spent on the work and other things.
- Determines which team member helps others by measuring the modifications made to other people's code.

Pluralsight Snapshots







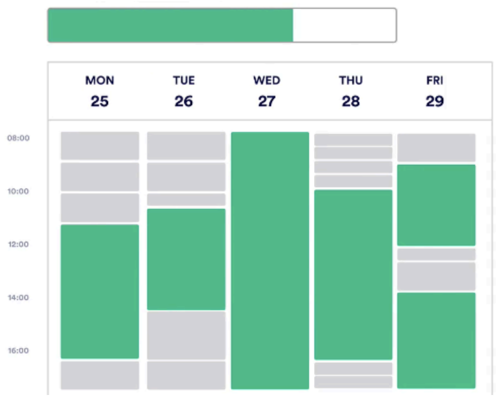


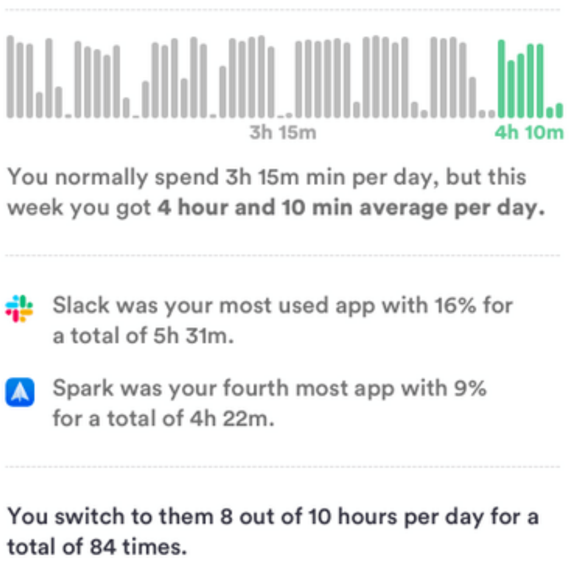
Dewo by Memory

This is a tool developed by Memory which analyses the behaviour of the individual by tracking the person’s desktop and web activity and suggests ways to the individual on how to improve their productivity. By tracking app switches, email distractions and meeting schedules, it helps the individual organise their schedule and help them setup for regular intervals of deep work.

Core Features -

- Tracks and visualises time spent of unproductive apps and content switching.
- Gives suggestions to organise meetings providing long windows of deep work to the programmer.
- Manager can track the time spent by the team by integrating Dewo with other platforms.





You normally spend 3h 15m min per day, but this week you got **4 hour and 10 min average per day.**

Slack was your most used app with 16% for a total of 5h 31m.

Spark was your fourth most app with 9% for a total of 4h 22m.

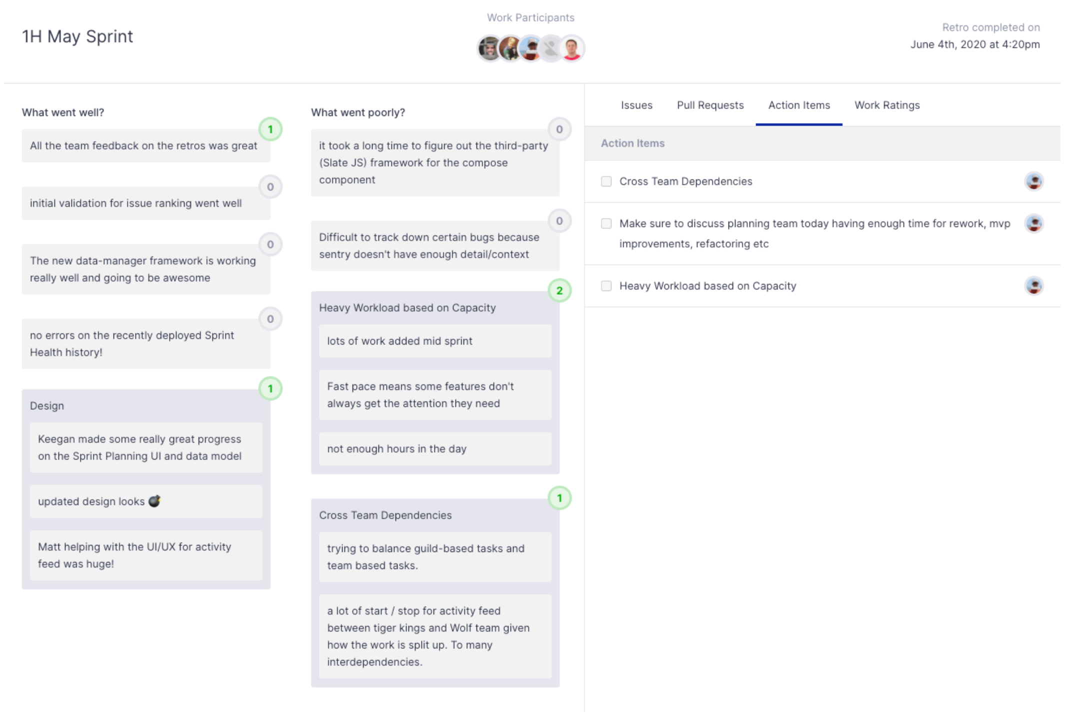
You switch to them 8 out of 10 hours per day for a total of 84 times.

Pinpoint

This solution uses machine learning provides customised recommendations and predicts risk by tracking individual’s and team’s code ownership, changes per commits and code churn (i.e. code rewritten). Pinpoint is used my many companies because of its simple interface and exceptional features.

Core features -

- Predicts the time it will take to complete an issue.
- Highlights blockers and capacity issues of the team
- Forecasts the likelihood of completion of planned issues in a sprint.
- Automatically assigns and tags issues from backlog to team members.



1H May Sprint

Work Participants

Retro completed on
June 4th, 2020 at 4:20pm

What went well?

All the team feedback on the retros was great 1

initial validation for issue ranking went well 0

The new data-manager framework is working really well and going to be awesome 0

no errors on the recently deployed Sprint Health history! 0

Design 1

- Keegan made some really great progress on the Sprint Planning UI and data model
- updated design looks 💣
- Matt helping with the UI/UX for activity feed was huge!

What went poorly?

it took a long time to figure out the third-party (Slate JS) framework for the compose component 0

Difficult to track down certain bugs because sentry doesn't have enough detail/context 0

Heavy Workload based on Capacity 2

- lots of work added mid sprint
- Fast pace means some features don't always get the attention they need
- not enough hours in the day

Cross Team Dependencies 1

- trying to balance guild-based tasks and team based tasks.
- a lot of start / stop for activity feed between tiger kings and Wolf team given how the work is split up. To many interdependencies.

Action Items

- ☐ Cross Team Dependencies
- ☐ Make sure to discuss planning team today having enough time for rework, mvp improvements, refactoring etc
- ☐ Heavy Workload based on Capacity

Pinpoint Snapshot

Algorithms & Techniques Used

Ethical and Moral Implications

Content

References

- https://medium.com/scope-ink/measuring-creativity-dfd1168d7c91
- https://www.tasktop.com
- https://www.chambers.com.au/glossary/mc_cabe_cyclomatic_complexity.php#:~:text=(Alias%3A%20McCabe%20number)&text=Some%20can%20avoid%20it.&text=McCabe's%20cyclomatic%20complexity%20is%20a,the%20more%20complex%20the%20code.
- https://wiki.c2.com/?ProductionCodeVsUnitTestsRatio
- https://memory.ai/timely-blog/productivity-tools-for-developers
- https://help.pluralsight.com/help/metrics
- https://memory.ai/dewo

- https://pinpoint.com/product/for-developers
- https://www.gitclear.com/pluralsight_gitclear_pinpoint_code_climate_code_development_kpi_metric_alternatives
-
-