Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CSU33012 Software Engineering
Measuring Software Engineering Report

Merlin Prasad Std# 19333557

November 16,2021

Introduction

This report outlines the varies methods that are used to measure software engineering , the platforms that carry out this action , the

Measurable Data

Software engineers produce vast amounts of data as they work however what to actually analyse to figure out if they are efficient is conflicting [1] .

Here are some common metrics used to measure the information software engineers produce and how this data is used to aid the software engineering process.

https://waydev.co/software-development-metrics/

Agile Process Metrics

Agile process metrics aid agile teams to make plans and decisions . While they don't measure the software itself they aim to help improve the software development process by tracking the progress of a team in creating practical, high quality software that is ready to be delivered.

Lead time

Lead time is a measure of how long it takes to develop an idea to have it as software running in production. Usually you aim to have a shorter wait time to be more responsive to customers . Looking at a team's lead time history can help predict how long a project will take to complete .

Cycle time

Cycle time is the amount of time it takes to implement a change to the software system and deliver that change into production. An easy way to game this metric would be to delay when to first commit and then store up commits locally and push them all in one go before next review. This would hinder the team work process as other developers would be unable to interact with work as it is in process .

Insert pic

https://stackify.com/track-software-metrics/

Team velocity
Team velocity accesses how many software units a team normally completes in an iteration or sprint. It aids in planning the amount of iterations that would be required. It is however quite futile to compare team velocities as the deliverables of each team would be different .

Open/close rates
Open/ close rates are a measure of the number of production issues that are reported and closed within a specific time period.

Production Metrics

These metrics aim to measure how much software is being created and determine the efficiency of the software development teams.

Active Days
Active days is the amount of time a programmer contributes code to the project . While spending more time could mean the programmer is putting more effort into work there is also the question on whether spending more time on something should be rewarded. If one programmer is able to write a solution in 2 hours instead of 8 should they be punished if the result of their work is the same ? It can be easily gamed by spending idle hours trying to solve something just to increase your hours . Active days are useful to measure to see how much time is wasted on non-engineering tasks such as planning and meetings. It can help plan for more productive meetings and see the cost of interruptions.

Code churn
Code churn indicates the amount of change that takes place in a particular area of code for a short period of time. It's useful for gauging code quality. Some level of reworking of a program is expected no matter how skilled the programmer .
Version control systems automatically track code churn using following formula.[6]

Code churn = Lines Added + Lines Deleted + Lines Modified

Add a picture

It is considered bad if a piece of code undergoes changes too frequently as it's usually a symptom of another issue such as an engineer struggling to write code required . A high amount of code churn could altert that a project needs attention.

Lines of code (LoC)

A common and somewhat naive approach to measure a software engineer's productivity would be counting the number of lines of code they produce. The idea being that if a programmer was solving more things they would produce more lines . However this approach contains many flaws and can be easily gamed. It is possible to solve a problem with 50 lines of code instead of 5 well written lines however this type of measurement would reward the former. This results in people trying to game the system by inserting pointless lines of code simply to be seen as being as a good software engineer . This increase in useless code would also be detrimental to the project as these extra lines will require more maintenance [2] . It is quite clearly a poor and unequal way to measure productivity. We wouldn't judge the quality of a painting with how many paint strokes but rather how good the final piece is .

Impact

Impact aims to measure the effect that code changes on the project. Code to affect multiple files would score higher. A few lines of code alterations in multiple files could have more impact then adding many lines to a single file. It aims to measure the efficiency of code changes in a more complex way than LoC

Efficiency

Efficiency is the amount of contributed code that's productive which tends to involve balancing coding output against code longevity. An engineer who is trying a new solution may have lower efficiency rate over an engineer contributing small commits however both are valuable

Mean Time Between Failures (MTBF) /Mean Time to Recover (MTTR)

These technical incident metrics aim to measure how well software recovery and preserves fata when a software failure occurs. It can be useful in assessing the stability of a team.

[1] https://harness.io/blog/developer-productivity/
[2] https://blog.ndepend.com/alternatives-lines-of-code-loc/

https://www.pluralsight.com/blog/teams/5-developer-metrics-every-software-manager-should-care-about

https://techbeacon.com/app-dev-testing/9-metrics-can-make-difference-todays-software-development-teams

https://www.gitclear.com/popular_software_engineering_metrics_and_how_they_are_gamed

https://www.gitclear.com/popular_software_engineering_metrics_and_how_they_are_gamed

https://waydev.co/software-development-metrics/

https://techbeacon.com/app-dev-testing/9-metrics-can-make-difference-todays-software-development-teams

[4] https://www.whatmatters.com/faqs/okr-meaning-definition-example/

[5] https://www.productplan.com/glossary/technical-debt/

[6] https://gitential.com/code-churn-the-good-the-bad-and-the-perfect/

[7] https://linearb.io/blog/what-is-code-churn/

https://blog.pragmaticengineer.com/can-you-measure-developer-productivity/
https://www.gitclear.com/popular_software_engineering_metrics_and_how_they_are_gamed

What platforms can be used to gather and process data / Development Analysis Platforms and Frameworks

OKRs should it be frame work? Yes move to frameworks look up waydev.

Objectives and Key Results (OKR) are a goal setting tool to communicate what someone wants to accomplish and what milestones need to be met to do this[4].
It was created at Intel by Andy Groven and many business such as Google and Netflix .(remove line?)

One of issues with trying to use OKR to measure productivity is that it contributes to an increase in technical debt for the company.Technical debt is the cost of additional rework caused by choosing a simple solution at the moment instead of using the more effient approach that would take longer .[5] . Many companies also lack the resources to be able to implement ACK and are ubavkr to move as quick as Google and this puts a lot of pressure on software engineer..

Pluralsight

SPACE

Cloud services i.e GitPrime

Pluralsigh

Code Climate

Flow

Waydev

What algorithms can we use/ algorithmic approach

Is it ethical?l Ethical analysis

Conclusion