Measuring Software Engineering Report

CSU33012 Software Engineering

Honglin Li

Student ID: 19315272

Table of Contents

1. Introduction

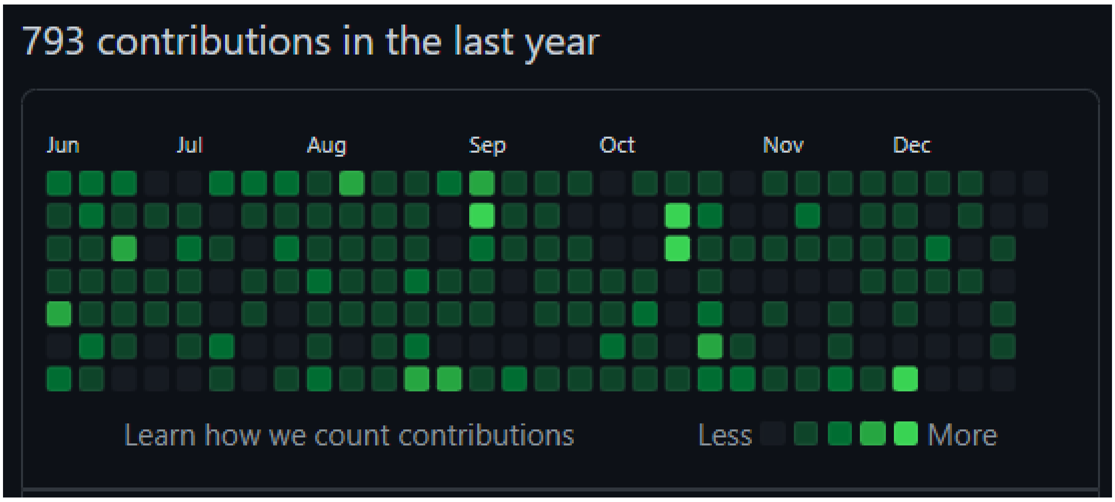
This report discusses the different aspects of measuring software engineering. We are going to focus on how one can measure software engineering activity, what platforms are used to process relevant data, and what kinds of computations could be made over the software engineering data. In addition, the last topic discusses the ethics and legal or moral issues related to the processing of such data.

2. Measuring engineering activity

Being able to measure engineering activity is vital to software engineers and their employers. It is fundamental to the efficiency of software development and maintenance.

2.1. Frequency of commits

In a software development team, when developers make changes to the current code base they are together working on and make pull requests, all other members of the team are then notified about these changes. Number of pull requests submitted is sometimes a quantifiable measurement of software engineering activity. The contributions graph on GitHub is a known example of this method of measuring engineering activity.



793 contributions in the last year

Contributions graph of GitHub user barrettotte showing frequency of commits in the last year[1]

However, quantity does not equal quality. One could make many commits with few necessary changes, but they can only be considered as little engineering activity.

2.2. Code review

Code Review, also known as Peer Code Review, is the act of consciously and systematically convening with one's fellow programmers to check each other's code for mistakes and has been repeatedly shown to accelerate and streamline the process of software development like few other practices can.[2] The main purpose of code review is to discover quality issues. Through code review, software engineers can achieve better code quality, find defects, and as well as acquire new knowledge from their peers.

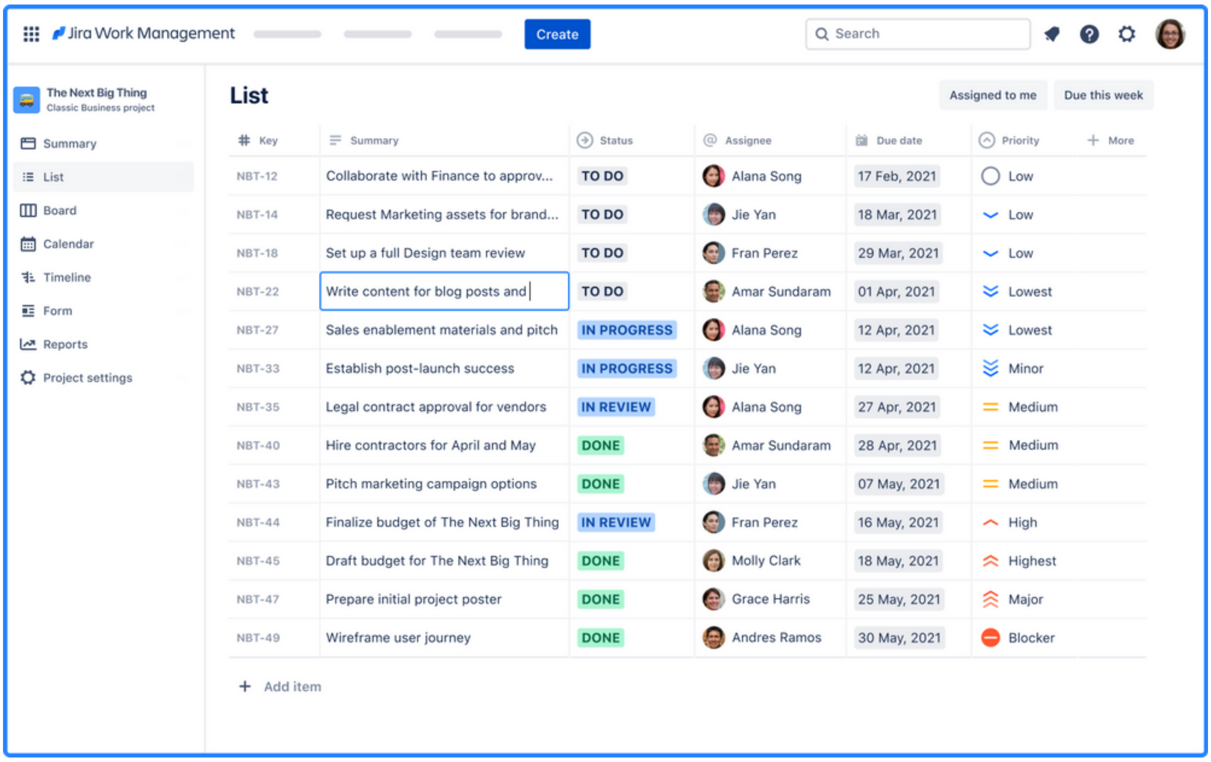
It is necessary to point out that the effectiveness of code review is related to the speed of reviewing. It is recommended that code review should be performed at a rate between 200 to 400 lines per hour.[3] Code review practices that are too fast would make it unlikely to spot errors.

3. Platforms

Various platforms for measuring software engineering have emerged as its importance is being recognized by developers and companies. These platforms provide services that help to analyze and visualize software engineering processes.

3.1. Jira

Jira is a software development platform made by Atlassian. Atlassian claims Jira to be "the #1 software development tool used by agile teams". Jira has released a number of products, namely Jira Software, Jira Service Management, Jira Work Management, and Jira Align.

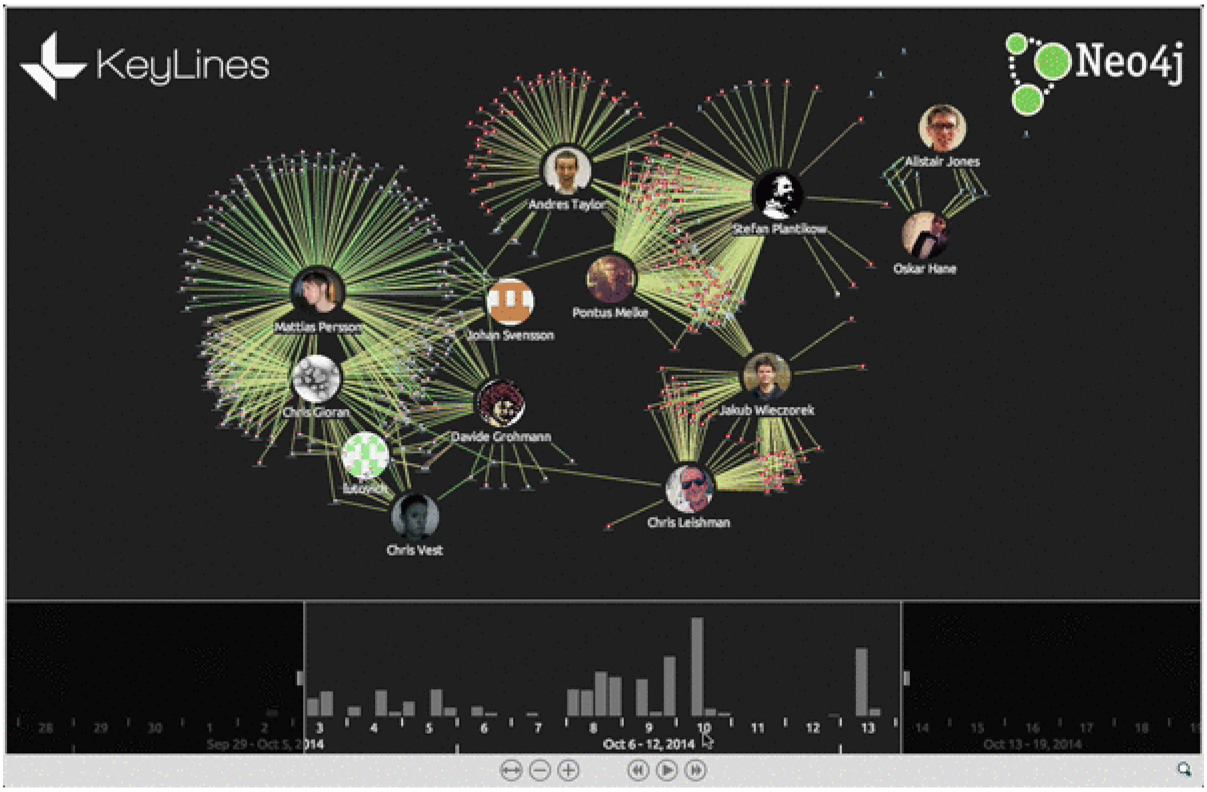


Key	Summary	Status	Assignee	Due date	Priority	+ More
NBT-12	Collaborate with Finance to approv...	TO DO	Alana Song	17 Feb, 2021	○ Low	
NBT-14	Request Marketing assets for brand...	TO DO	Jie Yan	18 Mar, 2021	Low	
NBT-18	Set up a full Design team review	TO DO	Fran Perez	29 Mar, 2021	Low	
NBT-22	Write content for blog posts and	TO DO	Amar Sundaram	01 Apr, 2021	Lowest	
NBT-27	Sales enablement materials and pitch	IN PROGRESS	Alana Song	12 Apr, 2021	Lowest	
NBT-33	Establish post-launch success	IN PROGRESS	Jie Yan	12 Apr, 2021	Minor	
NBT-35	Legal contract approval for vendors	IN REVIEW	Alana Song	27 Apr, 2021	= Medium	
NBT-40	Hire contractors for April and May	DONE	Amar Sundaram	28 Apr, 2021	= Medium	
NBT-43	Pitch marketing campaign options	DONE	Jie Yan	07 May, 2021	= Medium	
NBT-44	Finalize budget of The Next Big Thing	IN REVIEW	Fran Perez	16 May, 2021	High	
NBT-45	Draft budget for The Next Big Thing	DONE	Molly Clark	18 May, 2021	Highest	
NBT-47	Prepare initial project poster	DONE	Grace Harris	25 May, 2021	Major	
NBT-49	Wireframe user journey	DONE	Andres Ramos	30 May, 2021	⛔ Blocker	

List view of Jira Work Management[4]

3.2. GitHub API

GitHub API is a well-known platform used for extracting software engineering data from GitHub. Utilizing GitHub API, one can gather information about any GitHub user's repositories, given that they are public or have been authorized access to. GitHub API allows developers to extract raw data regarding software engineering activity and makes measuring software engineering activity more intuitive and customizable.



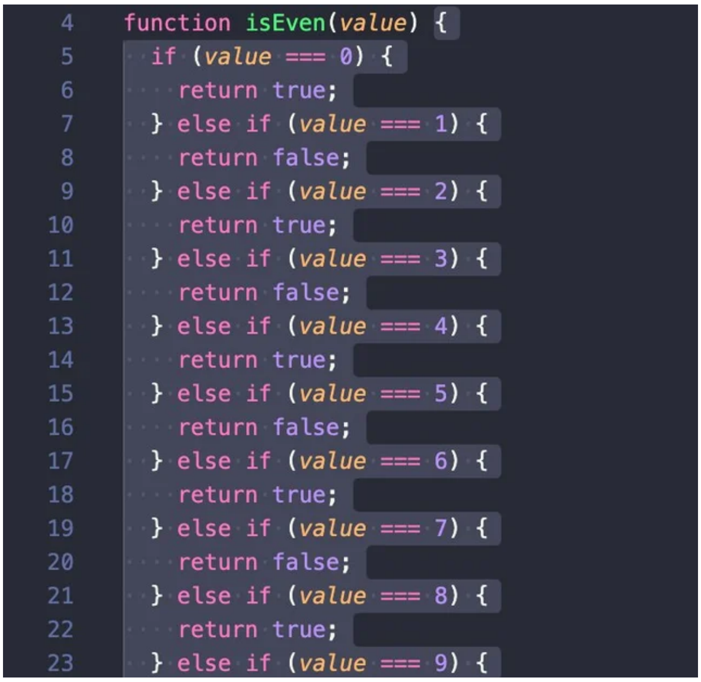
Graph visualization of activity over time in a GitHub repository[5]

4. Computation of data

Platforms like the ones discussed in the previous section enable easy access to software engineering data. To better measure software engineering performance using such data, raw data usually need to be processed.

4.1. Simple counting

In short, simple counting means counting lines of code. It is the most primitive and straightforward method to measure software engineering performance. Though intuitive and low-cost as it is, the use of simple counting in measuring software engineering encourages certain negative behaviors. As quantity is deemed equivalent to quality in this measure, it will lead to valuing complexity in code over efficiency.



```
function isEven(value) {
  if (value === 0) {
    return true;
  } else if (value === 1) {
    return false;
  } else if (value === 2) {
    return true;
  } else if (value === 3) {
    return false;
  } else if (value === 4) {
    return true;
  } else if (value === 5) {
    return false;
  } else if (value === 6) {
    return true;
  } else if (value === 7) {
    return false;
  } else if (value === 8) {
    return true;
  } else if (value === 9) {
```

An overly complex function with many lines of code[6]

4.2. Machine learning

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.[7] Just like its definition, the utilization of machine learning in measuring software engineering can help automate the process.

A 2008 research by Steffen Herbold on machine learning based on software metrics for process assessment shows that machine learning based methods perform better than methods that only take the status quo as basis for their hypothesis, since the history often contains valuable information. Furthermore, Herbold points out that “machine learning has the advantage of being unbiased, whereas experts instinctively use their intuition and expertise, which may be biased.”[8]

However, machine learning could be costly, which may put software engineering teams with small budgets in a place of disadvantage.

5. Ethics

The processing of software engineering data, like any data, could sometimes raise ethical, legal or even moral concerns. Although the automation of data processing eliminates the potential of bias in the processing of software engineering data itself, the end result of such processing still leaves room for misinterpretation or manipulation, which is a threat to the equity in the process of measuring software engineering. The use of third-party platforms could, moreover, potentially cause legal and even moral issues such as confidentiality and intelligence property violations.

6. Conclusion

The key to efficiently and correctly measure software engineering is multi-faceted. As discussed in the report, there are various methods and tools available for various software engineering needs. In order to maximize software engineering productivity and efficiency, it is ultimately up to specific developers or companies to determine the appropriate methodologies for themselves.

7. References

1. https://github.com/barrettotte
2. https://smartbear.com/learn/code-review/what-is-code-review
3. Kemerer, C.F.; Paulk, M.C. (2009-04-17). "The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data". *IEEE Transactions on Software Engineering*. 35 (4): 534–550
4. https://community.atlassian.com/t5/Jira-Work-Management-Articles/Announcing-the-waitlist-for-Jira-Work-Management/ba-p/1627313
5. https://cambridge-intelligence.com/graphing-github
6. https://www.reddit.com/r/ProgrammerHumor/comments/occadl/github_copilot_will_rule_us_all

7. https://www.sas.com/en_ie/insights/analytics/machine-learning.html

8. https://www.swe.informatik.uni-goettingen.de/sites/default/files/publications/research_summary.pdf