Measuring Software Engineering
CSU33012 | Hugo Bolger

Table of Contents

Introduction

What is Software Engineering?

Software Engineering is the application of systematic, disciplined and qualifiable approach to the development, operation and maintenance of software. The job of a software engineer is more than the writing of complex code, as is often the perception. Rather, software engineers work in collaborative teams with professionals and experts in certain fields to develop projects that can reach global audiences. Together, they identify the needs of the client and use their expertise to create a project that best serves the client’s desires. Software Engineering is about continuous planning and adapting, – continuous development – developing prototypes for clients and slowly moulding the project around their needs. Software Engineering is a process that has been developed to reduce cost, time, and improve the quality of end product. Since the 1960s, Software Engineering has played a role in changing in some the most important events in recent history. From the moon landing to the development of the World Wide Web, the fabric of how we communicate and interact in our daily lives has changed drastically due to the development of Software Engineering, and it has grown to become an essential part and have a practical role in almost every project or task that may be relevant in today’s environment. Yet the broad nature of software and its ever-changing atmosphere, means it can be quite hard to keep up. As Pepe Marañon was once quoted saying:

“If you are a developer, you always have to learn. You have to be permanently, all your professional life, learning new things.”

Now as we look forward to the future of Software Engineering, we wonder just how much of our lives could be changed by this rapidly developing process. However, from within the SE process there is still a large problem that has been fought to be overcome since its creation. After over 50 years since its inception, we still struggle to accurately quantify the work of a software engineer. We still ask ourselves; how do we measure software engineering; how do we quantify the productivity of a software engineer?

In this report, I will consider the ways in which the software engineering process can be measured and assessed in terms of measurable data, identify an overview of the computational platforms available to perform this work, and explain the algorithmic approaches available and the ethics concerns surrounding this kind of analytics.

Software Measuring Metrics

Traditional Metrics

Source Lines of Code

SLOC is one of the original metrics created to measure the efficiency of a software engineer. It measured the size of a computer programme by counting the number of lines in the source code. In theory, more difficult and complex codes would be longer and need more lines. It was an efficient and cost-effective way to measure software engineering and could be universally applied. But in practise, the flaws were immediately exposed.

Transcribed as nine flaws by Bhatt, Tarey and Patel,
These should be applied to all metrics to assess their limitations:

1) Lack of Accountability

There are significantly more steps to the engineering process that the writing of the code, which usually accounts for only 30-35% of the overall effort. It is unfair to judge a software engineer on a small part of their entire effort.

2) Lack of Cohesion with Functionality

More skilled developer may be able to create a programme with the same functionality but with far fewer lines of code. This discrepancy between functionality and effort means that the truly best programmers are not rewarded.

3) Adverse Impact on Estimation

Estimated based on lines of code can go adversely wrong, in all possibilities.

4) Developer's Experience

Implementation of specific logic differs based on the level of experience of the developer. A more experienced programmer can implement certain functionality in fewer lines than a less experienced programmer, yet the less experience and perhaps less efficient code is rewarded.

5) Difference in Language

Similar to the languages we speak, the languages computers speak can vary in length an efficiency. Programmes written in different languages can, not only be longer, but can require more effort and time, yet the number of function points remains the same. Lines of Code cannot measure the extra efficiency or inefficiency of different languages

6) Advent of GUI Tools

Programmers can now write relatively little code in return for high levels of functionality. We have access to in-built tools that shorten our source code and SLOC ignores the code automatically generated by a GUI tool. Again, the language problem arises as one language, with tools, can achieve in one line what another may do in multiple.

7) Problems With Multiple Languages

Often software is developed with more than one language. Tracking and reporting defect rates can be problematic, and defects cannot be attributed to a particular language subsequent to integration of the system.

8) Lack of Counting Standards

What is a line of code? Do comments count? Are data declarations included? What happens if a statement extends over several lines? With new languages constantly being created, it is difficult to standardize counting.

9) Psychology of a Programmer

Incentivises programmers to extend their code unnecessarily, creating complex and unintelligible results. This is in their best interest as they are essentially paid by-line. These programmes can be costly to maintain and require significant bug-fixing time.

Function Points

Function Points measure the amount of functionality in a software programme and is another metric for measuring software engineering. The functional user requirements of a programme are identified and categorised into inputs, outputs, inquiries, internal files and external interfaces. Each type is assessed for complexity and assigned function points. These are set by one of several ISO Standards for Function Points. In essence, the larger number if functions the more functionality, this allows for the measurement of the size of the programme. Unfortunately, there is currently no ISO Standard Method that includes algorithmic complexity in the sizing result. While the problems with SLOC are addressed with this metric, they are not entirely solved and many of the nine problems plague Functional Points also.

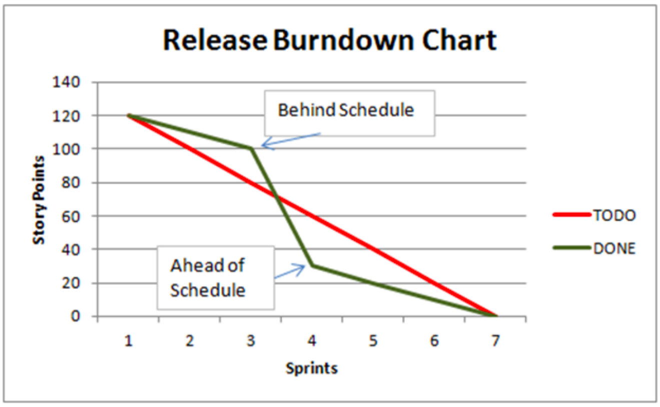
Number of Commits

This, in theory, should be a measure of the amount of time spent on a piece of code. Yet similarly to point 8 of the above flaws, what is a Commit? A commit can be anything, from a complex edit with extensive and vital input for the completion of a project to an arbitrary comment added to the end of the code. But this doesn't mean looking at commits is completely flawed. While it can't measure productivity, it can be show the activity levels of a software engineer, their commitment to the project and the collaborative effort of the team.

Agile Process Metrics

Burndown Charts

Measures how many tasks are completed over time – time measured in sprints. The tasks are units of software development needed to be completed in a sprint. The chart compares estimate work to days to complete it. As days pass in the sprint, the chart is amended and updated to track the work completed. It is very simple to understand and communicate to clients. It also helps in planning for the developers as they can track how much work still needs to be completed and an estimate of the time frame to complete it, planning that prevents disruptive mistakes that can slow the process.



Open / Close Rates

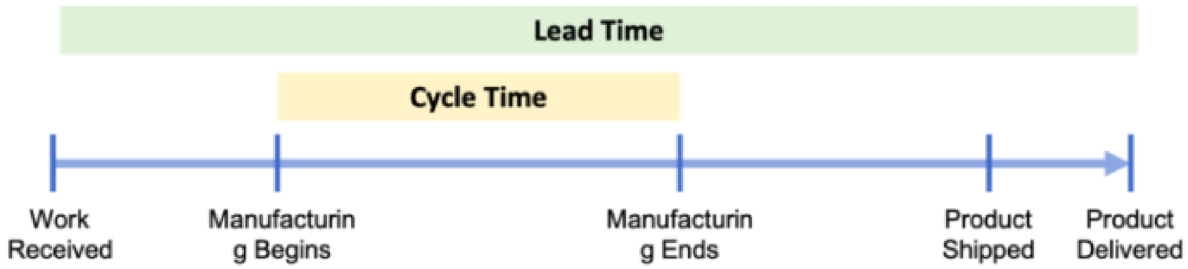
Addressing problem solving, this metric measures the number of issues reported (opened) and fixed (close) between a certain time frame. Think of it as the reverse cycle time and apply it in the same way.

Lead Time

This is the time period from the beginning of the development of a project to its delivery to the client. This helps in predicting how long the development process may be when discussing with the client. A lower lead time is not always a good sign and so this metric really must be used in tandem with a metric that measures the productivity of the code.

Cycle Time

Similar to lead time, cycle time is the time period taken to change or manufacture the system of a project. This can really show the levels of teamwork within the process and the level of focus on the maintenance of the finished product.

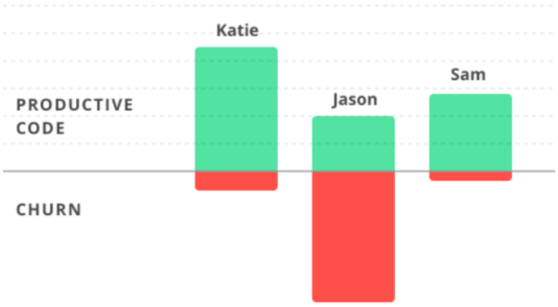


These agile process metrics reveal the productivity of the team but not necessarily the productivity of the code itself. The application of these metrics can give clients a clear picture of how focused the team are on their project and can be used to sway clients towards specific companies and teams.

Code Productivity Metrics

Code Churn

Code Churn is a software metric using lines of code that represents a proportion of a developer's code that is an edit to their recent work. It essentially measures the editing levels of a developer, but counting the number of lines that were edited, added or deleted. It can measure the productivity of a developer and in turn can be used as quality assurance by the managers and clients of the project. Higher churn can imply under-engagement of an engineer, excessive prototyping and polishing or frequently, a lack of clarity about what the clients need are – be it unclear requirements or just not being sure what they want.



Code Coverage

Code Coverage is a software testing metric that determines the number of lines of code that are successfully validated under a test procedure and helps analyse how comprehensively a software is verified. Calculate by dividing the number of lines of code executed by a testing algorithm by the total number of lines of code in a system, it can be seen as an efficiency ratio for the entire system. It helps uncover any dead or bad code in the system and helps developers ensure code quality is maintained. It can be a useful metric when increasing code scalability and allows for more efficient use of time, getting the software to market quicker. However, test must comprehensively test all possible outcomes to reduce chance of bugs and given the high chance of unforeseeable outcome, code coverage can often result in sub-optimal debugging.

Computational Platforms

Personal Software Process

Founded in 1996 by Watts Humphry, PSP is a computational framework for managing the quality of a projects. It helps developers to measure their work output and improve efficiency by developing their personal planning skills and guiding them to meet estimations.

PSP helps SEs to:

- Improve Approximating and Planning Skill
- Fulfil Goals and Commitments
- Manage Standards of Quality of the Project
- Reduce Defects in their Work

This process has developed a Model Framework of Activities

- PSP 0 – Personal Measurement
    - Measure Current Process; Time Spent, Faults, Size of Programme; to establish a baseline,
    - Post Mortem; Ensure all data has been recorded and analysed,
    - Implement Coding Standard; Develop Process Improvement Plan (PIP).
- PSP 1 – Planning and Scheduling
    - Estimates for Size of new programme; Using Baseline in PSP 0,
    - Prepare Test Report,
    - Estimate Time-Frame using Accumulated Data from previous projects,
    - Develop Schedule Planning using estimates; High Level Design (HLD).
- PSP 2 – Quality Management
    - Begin HLD Review
    - Begin Code Review

While the goal of this process is to increase quality and reduce defects, it makes no room for time management. Developers have to record scores of data and log them in forms. They have to track defects, create project plans and checklists for design and code. And all this data has to be reviewed and evaluated. This is not only costly in terms of time, but also in human error. The state of manual input will always suffer from human error at some level and due to the heavy reliance of PSP on manual input, that level is high and often result in incorrect process conclusions.

Hackystat

Hackystat is an open source framework for collection, analysis, visualisation, interpretation, annotation and dissemination of software development process and product data. It is not a conventional monolithic system but rather a collection of small services that work together to provide the information and data on software engineering activities. It can be used to analyse the software engineering process. Users attach software sensors to their development tools and they 'unobtrusively' collect and send raw data about development to a web service – Hackystat SensorBase – for storage. This is done automatically, in real time, so cost of time is cheap. Acting in a similar way to PSP in subsequent steps, it provides feedback on the process of the developer and is devised to aid them in their planning. However, while the continuous observation of the developer reduces the cost of manual input, it can ethically grey to consider.

GitPrime / Pluralsight Flow

This software aggregates historical git data into easy-to-understand insights and reports. Managers of large teams can break down the data of its engineers into small, manageable, and most importantly, understandable chunks. It tackles issues on all stages of the software engineering process;

- Code
  - Identifies Bottlenecks
  - Compares Sprints and Releases over time
  - Commit Efficiency for each language used
  - Tracks Skill Development and Workflow Recommendations for the team
  - Breaks Down team's Contribution and Work Habits
  - Match Events to Activities – Project Timeline
  - Helps to Plan – Compare Historical Sprints and their successes
- Review
  - Provides Granular Visibility into end-to-end Pull Request Workflow
  - Broadens Team's Knowledge of Codebase
  - Promotes and Measures Collaboration
  - Provides Visualisation of Code Review Dynamics
  - Tracks team Discussions for productivity, uncertainty and disagreement in reviews
  - Observes Team Dynamics and Patterns
- Collaboration
  - Observes Responses to Commits and General Feedback
  - Observes how more senior team members Mentor with Feedback
  - Boosts Knowledge Distribution evenly across the engineering team

It's easy to see the massive impact that Flow has. Flow increases efficiency and accuracy of the entire process. It is simple to use and easier to understand. It allows for rapid navigation of Big Data and analyses and adapts these data to suit the user's needs. In seconds, it provides insights into data that would take hundreds of experts hours to decipher and in doing so, significantly reduces engineering cycle time. However, it is clear to see the adverse ethical impact that such an intrusive software provides.

WayDev

Waydev is a new agile data-driven method of tracking engineers' output directly from git repositories without any need for manual input. They analyse data across several platforms from Github to Azure DevOps and Bitbucket. Their goal is to bring out the best in engineers' work and improve efficiency on a global scale. They can be seen as a competitor to Flow and operates in a very similar way. They provide a broad range of services which can be directly bought and downloaded from its website. These services includes;

- Daily Engineer Effectiveness
  - Work Logs – Better Planning and Expectations
  - Daily Update – Track team velocity over sprints
  - Sprints – Mitigate Risk
  - Time Cards – Schedule Meetings in order not to be Disruptive
- Valuable Insights
  - Developer Summary – Efficiency of Developers to remove blockers
  - Compare Developers – Over sprints to understand dynamics
  - Developer Stats – Easy-to-use graphs aggregating engineer stats
- Code Review & Engineer Collaboration

    - Review Workflow
    - Review Collaboration
    - PR Resolution
    - Fundamentals
- Overview of Performance
    - Project Timeline
    - Retrospective
    - Repository Stats
    - Targets & Estimations

Code Climate Velocity

A third competitor in the Engineering Intelligence market is Code Climate. It is designed to provide meaningful and actionable engineering insights for the entire engineering organisation. Velocity, its flagship product, analyses all the data from a Github repos and provides real-time analytics, heads-up displays and custom reports to give a clearer perspective on how an engineering team is working.

All these products operate in very similar ways and offer very similar products. And there are even more like them – Jira, Pinpoint, Lynda, SpeedCurve. Companies now have an abundance of platforms to help them analyse the engineering process. They metrics these platforms use to measure data are similar to those I mentioned before but span far beyond. The one thing that must be stressed and always considered, is the ethical concerns with observing engineers to this degree; something to keep in mind when considering how much these services have to offer, but for now we will discuss how to interpret this data with algorithmic approaches.

Algorithmic Approaches

Big Data can be complex to navigate and even harder to break down and present to a Layman. What is so special about the computational platforms mentioned above, is their ability to take something so complex and return simple graphs and insights which contain such in-depth data and analysis. So, how do they achieve this breakdown of the complex to the simple – complex algorithms.

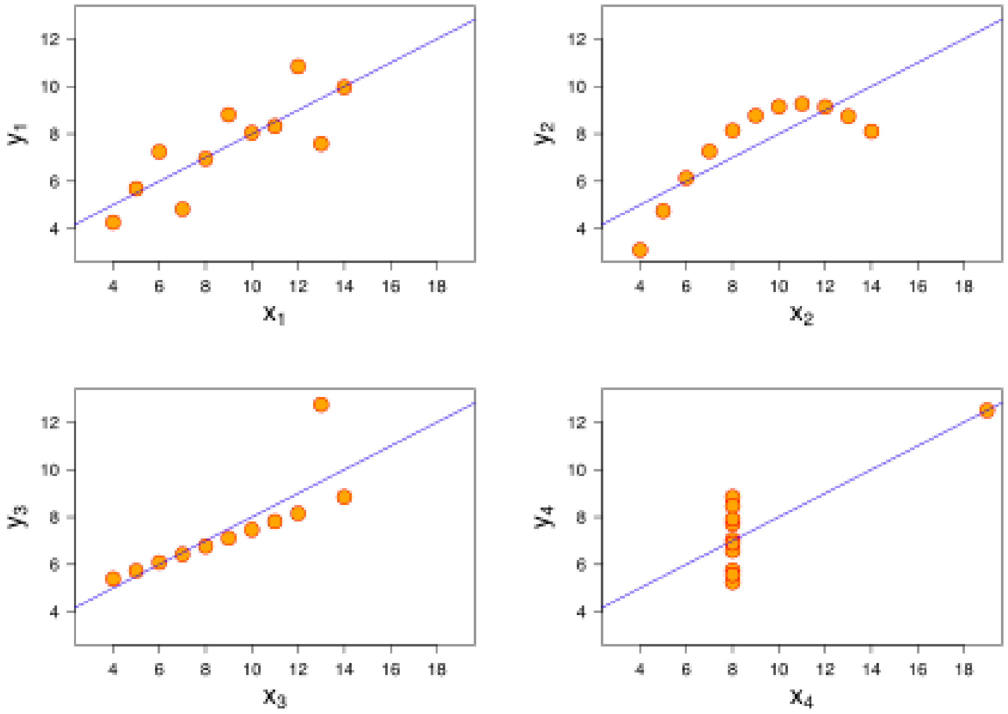
Machine Learning

Machine learning is vital to the development these algorithms. It attempts to allow computers to learn as humans do. The processing power of computers allows for massively complex tasks to be completed in fractions of seconds. There are three types of machine learning in which various algorithms lie.

1) Supervised Learning
   - Linear Regression

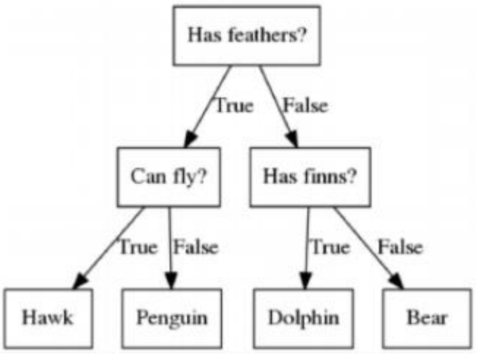
   A fundamental supervised machine-learning algorithm due to its simplicity and well-understood properties, Linear Regression is used to estimate real values based on continuous variables, by establishing a relationship between independent and dependent values and finding a line of best fit. LR allows for a quick and understandable graph that in many cases accurately depicts the correlation between the independent and dependent variables. The line of best fit tries to optimise the data and find a trend within the graph. However, when variables are highly uncorrelated or independent or don't follow a linear trajectory, this method can be misleading and inefficient,

x1 optimises the data whereas x[2:4] are misled.



- Decision Tree

Used for classification problems, this is a decision support tool that allows computers to map possible outcomes onto a tree to help identify a strategy most likely to reach a goal. They are very useful in the planning aspect of the SE process, where decision trees can be used as a visual and analytical decision support tool. Trees consist of decision nodes, chance nodes and end nodes, that portray the possibility of possible outcome in a simple graph.

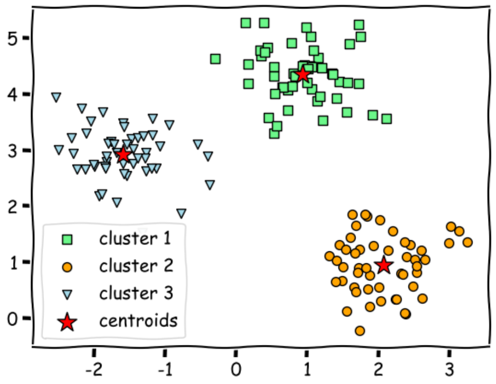


- K-Nearest Neighbours

Simple to implement and with low processing time, kNN classes similar points in a test set together and assigns any new points to the class most common to its nearest k neighbours. K-Clustering is a part of Multivariate Analysis, a set of statistical models that examine patterns in multidimensional data by considering several data variables. Its ability to handle multiple variables and dimensions at once is what make MvA so efficient and simple

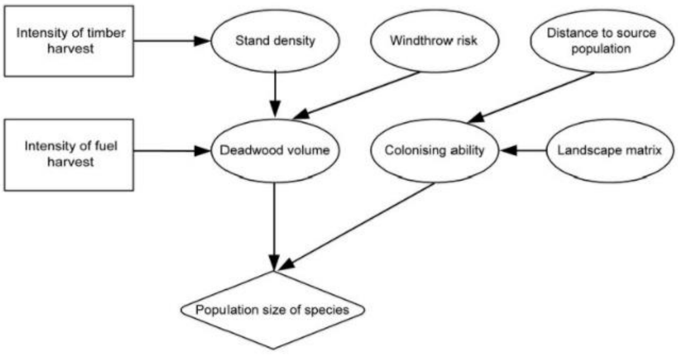
2) Unsupervised Learning
   - K-Means Clustering

Another algorithm using clustering of Multivariate Analysis, K-Means is one of the most popular of unsupervised learning algorithms due to its simplicity. Again grouping similar data points together, like in kNN, K-Means tries to find an underlying pattern. It begins by creating a first group of k data points which act a centroid for each cluster group. It then performs iterative calculations to optimise the position of each ‘centroid’.



3) Reinforcement Learning
   - Bayesian Networks

Bayesian Networks are probabilistic graphical models that represent a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Accounting for uncertainty and bias, it can take an outcome and can make assumptions about the possibility of causes for that outcome - assumptions that traditional statistical theory may have overlooked or been unable to uncover. Each causal node is given a probabilistic value. Bayesian network models capture both conditionally dependent and conditionally independent relationships between random variables.


Intensity of timber harvest
Stand density
Windthrow risk
Distance to source population
Intensity of fuel harvest
Deadwood volume
Colonising ability
Landscape matrix
Population size of species

Depicted; Bayesian Belief Network
Outcome: Population Size of Threatened Fungus Species

Ethical Concerns

As the data collection process becomes more complex and comprehensive, ethical issues must come to the fore. Monitoring employee daily performances and behaviours and storing that data for further analysis does feel like a complete invasion of privacy. But with the benefits that these analyses provide to the system as a whole, they have to be put in place. It is then our job to draw a line and never cross it. In Europe, our line is formed by the regulations put in place by the General Data Protection Regulation (GDPR).

- 6 data processing principles of GDPR:

1) Processed lawfully, fairly and in a transparent manner
2) Collected for specified, explicit and legitimate purposes
3) Adequate, Relevant and limited to what is necessary
4) Accurate and, where necessary, kept up to date
5) Retained only for as long as necessary
6) Processed in an appropriate manner to maintain security

- 6 lawful bases for processing:
  Personal Data can only be processed

1) If it's necessary to meet Contractual Obligations entered into by the data subject
2) To comply with the data control and its Legal Obligations
3) To protect the data subject's Vital Interests
4) For tasks in the Public's Interest
5) For purposes of Legitimate Interests pursued by the data controller
6) If the subject gives their Explicit Consent

- Data Subjects Rights

1) Right to be Informed
   - Of any details relating to the data controller
   - Reasons for processing
   - Recipients of the personal data
   - Retention Period
2) Right to Access Information
   - Confirmation of whether or not personal data is being processed
   - Where personal data is being processed relating to enquirer
   - Copy of personal info relating to enquirer
3) Right to Rectification
   - Rectification of inaccurate or outdated data
4) Right to Erasure
   - Erasure of personal data where it is no longer necessary in relation to purpose for which it was collected
   - Withdrawal of consent
   - Where personal data has been unlawfully processed
5) Right to Object to processing of personal data
   - Where processing relates to tasks in the public interest, under official authority or in the legitimate interest of others
   - Where data controller is using personal data for purposes of marketing directly to enquirer

6) Right to Data Portability
    - Obtain personal data from a data controller in a format that makes it easier to reuse your information in another context, and to transmit this data to another data controller of your choosing without hindrance.
    - Only applies when processing is automated
7) Right of Restriction
    - Restriction of processing of personal data, but personal data may still be stored
    - Where data is inaccurate
    - For legal reasons
8) Right to Automated Decision-Making and Profiling
    - Automated processing is only permitted with consent, with exception to safeguarding rights freedoms and legitimate interests

- Data Security

  All organisations must implement appropriate and proportionate technical and organisational measures to protect personal data

  Reporting all personal data breaches to data controller is mandatory

- Fines relating to companies who fail to follow the GDRP regulations are sizeable €20 million or 4% of Total Revenue – Whichever is higher

While GDPR covers a broad side of data protection and has made huge strides from the Data Protection Act, there are still a few aspect that may be cause for concern:

- Data Collection

  There is technically no limit to the amount of data a company can collect about its employees. While customer data is secured, employee data can be monitored with ease. Employee interactions, characteristics and attributes can be observed, analysed and quantified. Employee output, work rate, health, status, social impact, commitment; with all the data available, so much of an employee's way of being can be analysed and quantified.

- Data Usage

  With all the data available, perhaps the goals of the GDPR, is not to limit access to data, but rather limit usage and processing ability of firms. And while consent has become a huge part of preventing excessive processing of personal data, there are other regulations that allow firms to process these data. For example, 'Legitimate Interest of data controller' or 'Vital Interest of data subject' are ways in which firms are able to justify this intense processing and analysis of the personal data of their employees.

Ultimately, the ethical aspects of data collection are entirely subjective. It is up to the subject to decide how to weigh the benefits of data collection with the need for personal privacy. It can be frightening to comprehend the level to which corporations can analyse us but if we educate ourselves on the protection policies of the GDPR and other institutions outside the EU, we can feel more comfortable to never be surprised and trust that corporations will continue to adhere to the regulations as they are amended.

Personally, I am conflicted. The urge to try to 'remove all trace of my presence on the net' is hard to resist. But beyond my selfish desires to find myself in a real life spy film, there are some real fears; I don't like the idea of my interactions being monitors, my moods being analysed. I don't need my employer to receive a report on my health and social status, my work ethic and collaborative spirit. And while I hope my personal data will continue to be protected, the whole idea of Big Data processing and individual analysis is so new and so vast, that it is hard not to worry about the implications on your individuality.
But another side of me says 'No!'. I am hard working, healthy and outgoing. So, if my employer wants to analyse my day-to-day, then it can only be to the benefit of my future career.

Conclusion

In this report, I learned of the rapid development of the field of software engineering and more broadly, the field of data analysis. From the primitive software measuring metrics of SLOC and number of commits, to development to the complex metrics we use today, the innovation process is huge and immediate. With immensely impressive computational platforms, we can easily process and analyse data like never before. The strides being made in the world of statistical analysis allow us to generate more accurate conclusions to the data we source.

The analytical landscape we live in today is unrecognisable to the one just a couple of decades ago. With the state of innovation, who can say where we will stand 20 years from now. And the biggest question of all, what role will data protection regulations have in the future; will it continue to keep our personal data restricted and our rights reserved; or will innovation and regulatory loopholes find a way to shift the landscape once again?

Bibliography

What is Software Engineering

Codecademy News. 2020. *What Does A Software Engineer Do?*. [online] Available at: <https://news.codecademy.com/what-does-a-software-engineer-do/> [Accessed 14 November 2020].

2020. [online] Available at: <https://www.researchgate.net/publication/279259268_Software_Developers'_Perceptions_of_Productivity> [Accessed 14 November 2020].

En.wikipedia.org. 2020. *Software Engineering*. [online] Available at: <https://en.wikipedia.org/wiki/Software_engineering> [Accessed 14 November 2020].

Finding Metrics

Stackify. 2020. *Software Development Productivity: Most Important Metrics To Track*. [online] Available at: <https://stackify.com/measuring-software-development-productivity/> [Accessed 17 November 2020].

TechBeacon. 2020. *9 Metrics That Can Make A Difference To Today'S Software Development Teams*. [online] Available at: <https://techbeacon.com/app-dev-testing/9-metrics-can-make-difference-todays-software-development-teams> [Accessed 17 November 2020].

Pluralsight.com. 2020. *5 Developer Metrics Software Managers Should Watch | Pluralsight*. [online] Available at: <https://www.pluralsight.com/blog/teams/5-developer-metrics-every-software-manager-should-care-about> [Accessed 17 November 2020].

Source Lines of Code

2020. [online] Available at: <https://www.researchgate.net/publication/281840565_Analysis_Of_Source_Lines_Of_CodeSLOC_Metric> [Accessed 16 November 2020].

En.wikipedia.org. 2020. *Source Lines Of Code*. [online] Available at: <https://en.wikipedia.org/wiki/Source_lines_of_code#Advantages> [Accessed 16 November 2020].

Function Points

En.wikipedia.org. 2020. *Function Point*. [online] Available at: <https://en.wikipedia.org/wiki/Function_point> [Accessed 16 November 2020].

Youtube.com. 2020. [online] Available at: <https://www.youtube.com/watch?v=X0WLX8iAFb0&ab_channel=functionpoints> [Accessed 16 November 2020].

Number of Commits

Gitclear.com. 2020. *The 4 Worst Software Metrics Agitating Developers In 2019 - Gitclear*. [online] Available at: <https://www.gitclear.com/blog/the_4_worst_software_metrics_agitating_developers_in_2019> [Accessed 18 November 2020].

Burndown Charts

En.wikipedia.org. 2020. *Burn Down Chart*. [online] Available at: <https://en.wikipedia.org/wiki/Burn_down_chart> [Accessed 18 November 2020].

Agilenutshell.com. 2020. *Burndown Charts*. [online] Available at: <http://www.agilenutshell.com/burndown> [Accessed 18 November 2020].

Lead / Cycle Time

Toggl.com. 2020. *Takt Time Vs Cycle Time Vs Lead Time*. [online] Available at: <https://toggl.com/track/takt-time-cycle-time-lead-time/#:~:text=Lead%20Time%20vs.,it%20is%20ready%20for%20delivery.> [Accessed 18 November 2020].

Code Coverage

coverage, Better, H., Development, 2., review, I. and Reviewer, F., 2020. *Everything You Need To Know About Code Coverage*. [online] Codegrip. Available at: <https://www.codegrip.tech/productivity/everything-you-need-to-know-about-code-coverage/> [Accessed 16 November 2020].

En.wikipedia.org. 2020. *Code Coverage*. [online] Available at: <https://en.wikipedia.org/wiki/Code_coverage#:~:text=In%20computer%20science%2C%20test%20coverage,a%20particular%20test%20suite%20runs.> [Accessed 16 November 2020].

test IO. 2020. *Code Coverage Vs. Test Coverage* | *Test IO*. [online] Available at: <https://test.io/blog/code-coverage-vs-test-coverage/#:~:text=For%20those%20who%20aren't,is%20actually%20covered%20by%20tests*.> [Accessed 16 November 2020].

Code Churn

Pluralsight.com. 2020. *5 Developer Metrics Software Managers Should Watch* | *Pluralsight*. [online] Available at: <https://www.pluralsight.com/blog/teams/5-developer-metrics-every-software-manager-should-care-about> [Accessed 17 November 2020].

Pluralsight.com. 2020. *Code Churn - Defining And Resolving* | *Pluralsight*. [online] Available at: <https://www.pluralsight.com/blog/teams/why-code-churn-matters#:~:text=Code%20churn%20is%20when%20an,it%20again%2C%20and%20then%20again.> [Accessed 17 November 2020].

Pluralsight.com. 2020. *6 Causes Of Code Churn & What To Do About It* | *Pluralsight*. [online] Available at: <https://www.pluralsight.com/blog/teams/6-causes-of-code-churn-and-what-you-should-do-about-them> [Accessed 17 November 2020].

Personal Software Process

Humphrey, W., 2000. *The Personal Software Process*. [online] Available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.9598&rep=rep1&type=pdf> [Accessed 18 November 2020].

En.wikipedia.org. 2020. *Personal Software Process*. [online] Available at: <https://en.wikipedia.org/wiki/Personal_software_process#:~:text=The%20Personal%20Software%20Process%20(PSP,actual%20development%20of%20the%20code.> [Accessed 18 November 2020].

GeeksforGeeks. 2020. *Personal Software Process (PSP) - Geeksforgeeks*. [online] Available at: <https://www.geeksforgeeks.org/personal-software-process-psp/> [Accessed 18 November 2020].

Hackystat

Hackystat. 2020. *Hackystat*. [online] Available at: <https://hackystat.github.io/> [Accessed 18 November 2020].

Youtube.com. 2020. [online] Available at: <https://www.youtube.com/watch?v=NrLFIpm0wps&ab_channel=hackystat> [Accessed 18 November 2020].

GitPrime / Pluralsight Flow

Pluralsight.com. 2020. *Pluralsight Flow* | *Overview*. [online] Available at: <https://www.pluralsight.com/product/flow> [Accessed 18 November 2020].

WayDev

Waydev. 2020. *About Us - Waydev*. [online] Available at: <https://waydev.co/about-us/?gclid=CjwKCAiAzNj9BRBDEiwAPsL0d6kETwcYMhM8l6d0i_HxlRk-c8GF-7CV-A8iNyNyMBDAtIQrnZ_7PRoC9fMQAvD_BwE> [Accessed 19 November 2020].

Code Climate Velocity

Codeclimate.com. 2020. *Engineering Metrics To Improve Continuous Delivery Practices* | *Velocity*. [online] Available at: <https://codeclimate.com/> [Accessed 19 November 2020].

Comparing Platforms

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/#:~:text=Waydev%20draws%20attention%20to%20both,oriented%20towards%20Project%20Management%20issues.> [Accessed 19 November 2020].

Finding Algorithms

2020. [online] Available at: <https://www.researchgate.net/publication/220808343_A_framework_for_computational_thinking_across_the_curriculum> [Accessed 19 November 2020].

Codes), C., 2020. *Commonly Used Machine Learning Algorithms* | *Data Science*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/> [Accessed 19 November 2020].

Transforming Data with Intelligence. 2020. *5 Advanced Analytics Algorithms For Your Big Data Initiatives* | *Transforming Data With Intelligence*. [online] Available at: <https://tdwi.org/articles/2018/07/02/adv-all-5-algorithms-for-big-data.aspx#:~:text=Linear%20regression%20is%20one%20of,sets%20of%20continuous%20quantitative%20measures.> [Accessed 19 November 2020].

Linear Regression

En.wikipedia.org. 2020. *Linear Regression*. [online] Available at: <https://en.wikipedia.org/wiki/Linear_regression#Machine_learning> [Accessed 19 November 2020].

Decision Tree

En.wikipedia.org. 2020. *Decision Tree*. [online] Available at: <https://en.wikipedia.org/wiki/Decision_tree#:~:text=A%20decision%20tree%20is%20a,only%20contains%20conditional%20control%20statements.> [Accessed 19 November 2020].

Multivariate Analysis

Medium. 2020. *An Introduction To Multivariate Data Analysis*. [online] Available at: <https://towardsdatascience.com/an-introduction-to-multivariate-data-analysis-ece93ceb1ed3> [Accessed 19 November 2020].

kNN

En.wikipedia.org. 2020. *K-Nearest Neighbors Algorithm*. [online] Available at: <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm> [Accessed 19 November 2020].

K-Means Clustering

Medium. 2020. *Understanding K-Means Clustering In Machine Learning*. [online] Available at: <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1#:~:text=5%20min%20read-,K%2Dmeans%20clustering%20is%20one%20of%20the%20simplest,popular%20unsupervised%20machine%20learning%20algorithms.&text=In%20other%20words%2C%20the%20K,centroids%20as%20small%20as%20possible.> [Accessed 19 November 2020].

Medium. 2020. *A Beginners Practical Guide: Apply K-Means Clustering Algorithm*. [online] Available at: <https://levelup.gitconnected.com/a-beginners-practical-guide-apply-k-means-clustering-algorithm-d57295699e61> [Accessed 19 November 2020].

Bayesian Network

Brownlee, J., 2020. *A Gentle Introduction To Bayesian Belief Networks*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/introduction-to-bayesian-belief-networks/> [Accessed 19 November 2020].

En.wikipedia.org. 2020. *Bayesian Network*. [online] Available at: <https://en.wikipedia.org/wiki/Bayesian_network#Developing_Bayesian_networks> [Accessed 19 November 2020].

2020. [online] Available at: <https://www.researchgate.net/figure/Illustration-of-a-simple-Bayesian-Belief-Network-The-example-depicted-is-an-influence_fig1_262522486> [Accessed 19 November 2020].

Ethical Concerns

General Data Protection Regulation (GDPR). 2020. *General Data Protection Regulation (GDPR) – Official Legal Text*. [online] Available at: <https://gdpr-info.eu/> [Accessed 25 November 2020].

Mater.ie. 2020. [online] Available at: <https://www.mater.ie/data-protection/Rights-of-Individuals-under-the-General-Data-Protection-RegulationAmended.pdf> [Accessed 27 November 2020].

2020. [online] Available at: <https://www.youtube.com/watch?v=NxgZ57BTkFQ&ab_channel=PhilLee> [Accessed 25 November 2020].