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

**School of Computer Science and Statistics**

# Assessment Submission Form

| | |
|---|---|
| **Student Name** | Gavin Eccles |
| **Student ID Number** | 15318307 |
| **Course Title** | MSISS |
| **Module Title** | Software Engineering |
| **Lecturer(s)** | Stephen Barrett |
| **Assessment Title** | Software engineer report |
| **Date Submitted** | 23/11/17 |
| **Word Count** | 5200 |

**I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at: http://www.tcd.ie/calendar**

**I have also completed the Online Tutorial on avoiding plagiarism 'Ready, Steady, Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write**

**I declare that the assignment being submitted represents my own work and has not been taken from the work of others save where appropriately referenced in the body of the assignment.**

Signed .....Gavin Eccles..... Date .....23/11/17.....

“Consider the ways in which the software engineering process can be measured and assessed in terms of measurable data, overview of computational platforms available, algorithm approaches available and the ethics concerns surrounding this kind of analytics”

## Introduction

The plan of this report is to start by giving a brief background on the software engineering process, then to focus on the four parts of the task and ultimately provide a conclusion on my findings. I will stride to give my opinion on topics discussed while supporting with concrete sources.
Layout:

1) Software Engineering Process
2) What data is relevant?
3) Where to compute?
4) What Algorithms?
5) Ethics concerning this work
6) Conclusion

## 1.The Software Engineering Process

The software engineering process is a complicated process involving the application of a disciplined approach to be it design and development or operation and maintenance of the software. The aim of the process is to divide the work into distinct phases to improve designing, efficiency, product management and project management. [1]


The six steps are identified in the accompanying image and are all of equal importance. From identifying requirements to designing the software architecture to testing and maintenance, each step plays an imperative role in the software engineering process.

## 2.Measurable Data

<u>Collecting Software Engineering Data</u>

Collecting data regarding software engineers leads to challenges ensuring that the collected data can “provide useful information for project, process, and quality management and, at the same time, that the data collection process will not be a burden on development teams” (Kan, 2014). Therefore, it is important to carefully consider what kind of data to collect that will prove beneficial, for example improving performance. The metrics and models that the data is based on, are used to drive the improvements so before any data collection begins, the goals and interests of the data collection should be established. [2].

According to Kan, the most important thing is that the data gathered be focused, accurate, and useful than that it be plentiful. Without being metrics driven, over-collection of data could be time wasting and expensive. Data collection costs are less expensive for larger scale companies due to economies of scale but will never be insignificant. As well as expense costs, manual data collection is an unreliable activity, with missing data and error filled data affecting the analysis process.

A simple data collection methodology of six steps is put forward by Basili and Weiss (1984) included below and I feel is important to follow these steps to optimize the benefits from Data analytics. [3]

1. Establish the goal of the data collection.
2. Develop a list of questions of interest.
3. Establish data categories.
4. Design and test data collection forms.
5. Collect and validate data.
6. Analyse data.

Why Get this Data?

It is important to gather data from software engineers because the most useful resources in software companies are human resources. Software engineers are mostly concerned with "human effort needed to build a product, quality, easiness of maintenance, cost, and time required". [4] Therefore it is essential to understand how developers work and to encourage these developers to achieve the best results possible by adopting a new process.

According to Sillitti (2003), this data can be used to estimate variables such as time scheduling, quality of ongoing developing projects, identify improvements for development process and highlight existing problems. Another positive influence of data analytics is that it can be used to identify security issues that exist in the company i.e. if one employee has been accidently leaking company code etc. The main objective of data analysis is to establish patterns which will increase software development performance. Data which yields intelligence about the project and the development process is vital for business growth and future success.

Relevant Data on software engineers

In terms of data relating to the software development process, a wide range of data is available on the work of software engineers concerning their code. For example:

| | | |
|---|---|---|
| - Number of commits | -Number of contributions | - Frequency of contributions |
| -Team performances | - Technical Debt | -Individual Performances |
| -Projects involved in | - Bug Fixing | -Code churning |

Non-code related data:

| | | |
|---|---|---|
| -Meetings attended | -Communications | -Gender |
| -Emails | -Date joined Repository | -Company |

The mining of software repositories involves extracting all forms of data including basic and value-added information from the repositories. With all this available data, an important component in data analytics comes in to play called taxonomy. It is the classification of data once it's been gathered and can be achieved through various computational platforms such as SQL (structured query language) which helps in data simplification. Data is then stored in a large database before the next step of analysis begins.

Data Sources

Considering the problem of human data collection, automating this process will improve both the data quality and reduce the acquisition effort done by programmers. A process called PSP gives developers a guideline on how to track their work, resulting in large amounts of data where automated tools come into play such as Hackystat and PROM.

The Personal Software Process (PSP) provides engineers with a disciplined personal framework for developing software. "The PSP process consists of a set of methods, forms, and scripts that show software engineers how to plan, measure, and manage their work". It is important in the job of a software engineer when writing requirements, running tests, defining processes, and repairing defects. (Humphrey 2000) [6]. In the PSP, it is essential to collect "detailed metrics of the development time, bugs discovered and corrected at all development stages, as well as software size" [5]. One of the key components of the job of a software engineer is to consistently produce quality products and in order to achieve such high expectations the engineers must plan, measure and track product quality and focus on quality from the very beginning. The first step to improve productivity is to identify inefficiencies. Repeatedly monitoring can aid developers in tracing their own performance, match them to a planned schedule and find out what additional elements may affect them such as behaviours, interactions and different environments. To improve their personal processes in the long run, they must record and analyse each step of a job.

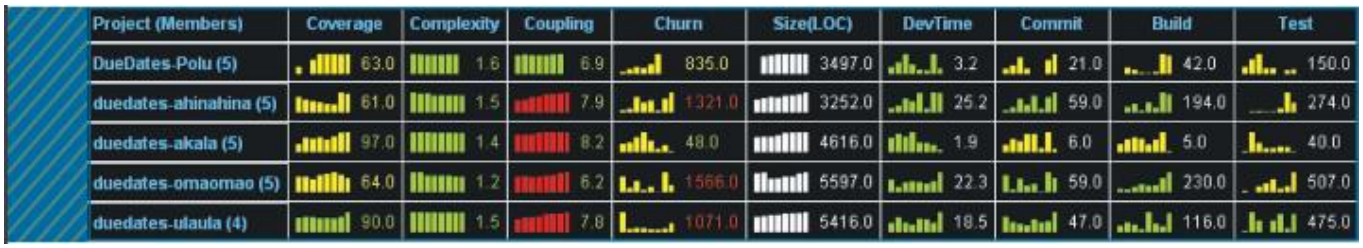
Data quality problems exist in PSP due to the manual nature, a tool called Leap toolkit was created to overcome problems with data quality with the use of automation and data normalisation. The developer is still responsible for entering the data but the benefit of Leap toolkit is the automation of PSP analysis and extending the degree of analysis than PSP by introducing more complicated regression models. Leap enables developers to stay private by not displaying individual's names. By enabling developers to control their own data files, it makes the first step to avoid measurement dysfunction. "It creates a repository of personal process data that developers can keep with them as they move from project to project and organization to organization", this kind of analysis is significant in improving individual's performance as a whole. [5] All this historical data collected is then statistically analysed by some of the following tools:

Hackystat is a data collection tool which was developed at the university of Hawaii. After the data is gathered in the PSP, Hackystat allows developers to process and analyse this data automatically. The tool is focused on the individual maintaining a high level of privacy which is very important to the company and individual to whom the data concerns.

PROM (PRO Metrics), an automated tool for collecting and analysing software metrics and PSP data. It collects both code and process measures by using plug-ins that collect data from the tools the developers use.

The difference between the two data tools is PROM analysis the whole development process including non-code related topics, providing an analysis of all members of the team of developers. Whereas Hackystat focuses primarily on code related activities and the individual software developers. The table to the right, illuminates some of the key differences between the two frameworks. The image below shows a Software ICU (intensive care unit) display based on Hackystat.

| Feature | PROM | Hackystat |
|---|---|---|
| Supported languages | C/C++, Java, Smalltalk, C# (planned) | Java |
| Supported IDEs | Eclipse, JBuilder, Visual Studio, Emacs (planned) | Emacs, JBuilder |
| Supported office automation packages | Microsoft Office, OpenOffice | - |
| Code Metrics | Procedural, object oriented and reuse | Object oriented |
| Process Metrics | PSP | PSP |
| Data aggregation | Views for developers and managers | Views for developers |
| Data Management | Project oriented | Developer oriented |
| Business process modeling | Under development | - |
| Data analysis and visualization | Predefined simple analysis and advanced customized analysis (both in beta) | Predefined simple analysis |




In my research into the kind of data tools for measuring the software engineering process I came across additional analytic technologies such as sonar and sixth sense analytics. To conclude, I feel there is a very large amount of data out there to be mined, involving a high scale to process it all. By the use of analysis tools available such as Hackystat and PROM, one can begin to "crunch" and analyse this data and ultimately pave way for a range of benefits improving overall efficiency, increased performance and improved product quality of software engineers.

## 3.Where to Compute?

Analytics as a service

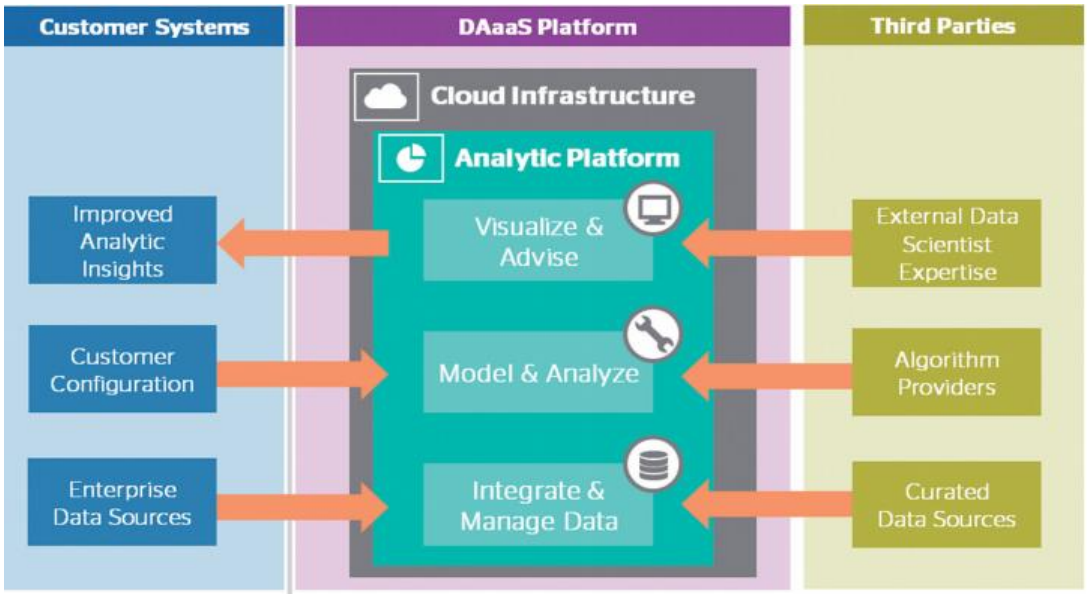
With the advancement of technology not only are we seeing historic jobs being automated but new jobs are created. Data Analytics is a brand-new space and has become a new "as-a-service" that companies are willing to spend on. "Analytics as a Service empowers knowledge workers by granting them personalized access to centrally managed information data sets" [10]. This prevents many of the delays that data scientists face and enables them to gain access into a larger database

to discover richer insights faster than before. This service has proven successful in business companies as business analysts have more information on consumers and can used this insight to drive revenue growth.

-The process of Aaas: When an analyst requires data, they first go to a web portal and request a personalised data sandbox with the relevant information needed from the master data warehouse. Upon being granted access, the analyst can use exploration tools to analyse the dataset and then use a visualisation kit to display this graphically. On completion of analysis, they release access from their sandbox back to the IT. [10]

-Benefits: Results in reduced costs and compliance risks with IT while increasing productivity of its users. This increase is evident as it grants workers the ability to make more informed decisions and to deliver more significant outcomes. Issues related to security remain intact even though data is more accessible, it remains physically stored and controlled by IT in the cloud. Without a doubt, in an ever-increasing data driven economy, companies that use Aaas will allow faster access to data for employees and will evolve their business quicker and to higher levels than obtainable before. [10]

DAaas (Data Analytics as a service) is a branch of Aaas and aims to "lower the barrier of entry to advanced analytical capabilities, without demanding that the user commits to large internal infrastructures and human resources to the project" [11]. The main concept is visualised below.




Using DAaas in the Real-world

Using data analytics is evident across many industries:

1) In retail, a model can be created from existing data for campaign management and customer behaviour. This behaviour would include both customer internet activities and instore activity via mobile devices and could be used to drive revenue growth.
2) In SmartCities, copious data sources are provided by the city such as sensor networks. Using analytical capabilities, DAaas can be used to provide an effective solution for cities under extreme pressure to reduce costs.
3) In the Electrical Utilities sector, it can be used to identify fraud and detect non-technical losses. A customer with a smart meter can upload their information to the system and in turn this information is processed analytically and consulted by experts.

These are few among many examples of where Big data "crunching" comes into play in the real world. The barriers to entry to advanced analytical capabilities are being lowered. Users are no longer demanded to large internal infrastructures and to committing human resources to the project of data analysis. [11]

Data Analytic companies

**6th Sense** are a data analytics consulting firm who strive to help customers gain a competitive advantage in their marketplace by using their customers data and transforming it into a strategic asset. This is achieved through rigorous analysis on high scale data. Some of their clients include SAS, Aptiva and Reliance life assurance.

**Idiro** target improving businesses decision making by analysing and understanding their data. It's head quarters are located in Dublin. The four concepts that Idiro claim to fame are better marketing insights, Operational Efficiency, Increased sales and predicting future trends, all achieved to gathering, analysing data and creating predictive models.

Business Analytics Case Studies

How is Big Data used in real-world situations?

- Google uses big data to refine its major search and ad-serving algorithms
- LinkedIn which is the social network for business orientated people uses big data to improve its product by predicting who you may know and how many degrees of separation you are from another person.
- McLaren's Formula One racing team use real-time car sensor data during races to anticipate issues with it's cars by using predictive analysis enabling them to take corrective actions before it's too late.
- 23andMe are a personal genomics and biotechnology company which provide customers 360-degree understanding of their genetic history by creating a model which extracts insights from big data.
- Tesla are a car manufacturer who are leading the development of self-driven cars. They use huge amounts of data from car sensors and mapping of roads to make this space the future of driving. They plan to have a car drive itself from New York to LA by the end of this month (November 2017). [14]

Companies involved in the measurement and analysis of Software Engineers

To bring this part of the question into the context of measuring and assessing the data solely on software engineers, I researched into the different companies that perform this kind of analysis. I have included the companies below:

**Code Climate**

The business model of Code Climate as stated on their website: "Code Climate incorporates fully-configurable test coverage and maintainability data throughout the development workflow, making quality improvement explicit, continuous, and ubiquitous". It is used by over 100,000 projects and analyses over 2 billion lines of code every single day. It works on a wide range of software languages such as Java, Python, PHP, Haskell etc…

Code Climate enables organisations and engineering teams to take control of the quality of code they produce by introducing "fully configurable test coverage and maintainability data throughout the development workflow". In modern day software development, a huge concern is technical debt which reflects the marginal cost of reworking code by choosing an easier solution now than using a more sustainable approach that would take longer. Choosing a package like code climate, the fight against technical debt can be combated as it identifies frequently changed files that add no increased value but instead have inadequate coverage and maintainability issues.

Reviewing code is not the most exciting task so having a company do this kind of analysis on your code can simplify the job of a software engineer enabling them to focus their time on more important development problems. Code climate has seamless integration with Github and provides line by line technical debt and code coverage on Github, therefore identifying the value to which code churning actual adds. Other features which are provided include a week by week progress tracker which zooms in on each Github contributor to a project. This identifies their performance to a particular project by showing a summary of the most important changes that they merged. Also containing a trend section which analyses the overall performance of your code on a long-term basis determining if your software is getting better or worse. It identifies areas in your code across all projects which require the most attention by correlating information against areas of high churn.

In the real world, Code climate is used by companies such as thoughtbot, chargify, DNSimple, Harvest, Litmus and many more. [7]

**Codebeat** is a similar company which gives instant feedback on a developer's code to decrease technical debt and find refactoring opportunities costing only $20 dollars per user. It is integrated with Github, Bitbucket and Gitlab and once your repository is connected, Codebeat begins aiding the creation of clean code. Companies using Codebeat include JTribe, Perk and netguru. Codebeat claim that they help these companies by decreasing "technical debt and thanks to its multiple language support, allows them to consolidate all of their code quality metrics into one, easy-to-use-tool". [8]

**Codacy** is another alternative: "Check code style, security, duplication, complexity and coverage on every change while tracking code quality throughout your sprints.". Companies such as adobe, Paypal and Toptal are customers. [9]


Both CodeFactor and Hound specialise in code review for Github pull requests and have been gaining popularity. To conclude Software Engineers are measured and assessed to a large degree already. There is a vast range of computational platforms available to analysis data on Software Engineers. The reason for the abundance of companies is due to the copious amount of code written daily, increasing the demand for tests to ensure code is clean and that code churning essential aids in the development and performance of software engineers rather than impeding progression.

## 4.The algorithmic approaches available

Humans vs Machine

Artificial intelligence is intelligence displayed by machines in contrast with natural intelligence displayed by humans, these machines mimic human thinking functions such as learning and problem solving. But AI machines are becoming more and more capable, reducing the need for human intellect and replacing historic human jobs with automation. Daniel Susskind preaches that AI will help aid professionals and will transform professionalism in the long run by displacing their work. In an interesting YouTube video, he presents the idea of AI fallacy:

*"The mistaken assumption that the only way to develop systems that perform tasks at the level of experts or higher is to replicate the thinking processes of human species"*

This is a very powerful statement which leads to serious ethical issues addressed in the next section but also makes me ponder about what extent can we use this computational power and efficient algorithms to reduce and replace routine tasks. Daniel identifies human qualities such as creativity, judgement and empathy which machines do not possess. These qualities are what makes humans different to routine based machines but effective computing gaining insights into correlations have led AI to become more intelligent than a human mind. Since the learning curve of AI is fast, it is a mistake to assume machines can't be creative cause they can't think or that they can't feel so they can't be emphatic. With machines becoming increasingly capable these qualities are not far away and they will take on more and more of today's tasks. [17]

Computational Intelligence

"Computational Intelligence comprises concepts, paradigms, algorithms, and implementations of systems that are supposed to exhibit intelligent behaviour in complex environments", (Kruse et al., n.d.). Intelligent systems are desired to support decision making, be in charge of controlling processes and to identify and interpret patterns or to be responsible for manoeuvring vehicles or robotics autonomously in different situations.

Involved in the problem-solving strategy is integrating uncertain, vague and incomplete knowledge, since machines focus on perfect knowledge, algorithmic approaches must be adopted to adapt and continue the AI learning curve. Two successful approaches are the Fuzzy systems and the Bayesian and Markov networks. In Fuzzy systems, imperfect knowledge is formalized and can be used to derive inference mechanisms, leading to approximate reasoning methods. Bayesian networks are attempts to store and reason with uncertain knowledge in complex application areas. Bayesian networks contain a probalistic graphical model which are well suited for dependence analysis and learning from the data.

Supervised vs un-supervised machine learning

Supervised learning is a type of machine learning algorithm that uses a training dataset to make predictions. Included in this dataset are input data and response values. The algorithm then seeks to create a model that makes predictions of the response variable for a new dataset. The larger the test dataset, the more accurate the model will be.

Unsupervised learning is where the test file only has input data and no corresponding output values. The algorithm models the underlying structure in order to learn more about the relevant data. There are no historical answers to teach the machine in this case. Both supervised and unsupervised machine learning algorithms play a huge part in modern society, using regressive models to make powerful predictions by identifying patterns that could be invisible to the human eye. [19]

Problems with solely relying on algorithms

Although algorithms are highly powerful, they can sometimes provide false positives and false negatives, probably due to data errors. Depending on what the algorithm is being used to identify, it could be a huge problem which I will discuss in the ethics section.

Software Intelligence

To bring this section back to the question, I feel these algorithms can be used to measure and assess the performance of software engineers. Perhaps, by identifying most productive developers, which employees add the most valuable code, differentiating between good and bad employees etc. An interesting article by Hassan I felt highlights the use of algorithms on software repositories:

| Software Engineering Data | Mining Algorithms | Software Engineering Tasks |
|---|---|---|
| **Sequences**: execution/static traces, co-changes, etc. | association rule mining, frequent itemset/subseq/ partial-order mining, seq matching/clustering/classifi-cation, etc. | programming, maintenance, bug detection, debugging, etc. |
| **Graphs**: dynamic/static call graphs, program dependence graphs, etc. | frequent subgraph mining, graph matching/clustering/ classification, etc. | bug detection, debugging, etc. |
| **Text**: bug reports, emails, code comments, documentations, etc. | text matching/clustering/ classification, etc. | maintenance, bug detection, debugging, etc. |

Hassan coins the term Software Intelligence (SI) as the “future of mining software engineering data, within modern software engineering research, practice, and education”. He claims that the future of software engineering will change dramatically as SI will provide researchers and developers concrete facts and evidence to help them design evolutionary or transformative approaches such as new languages and tools, instead of decision making on their gut feeling. [15] In current state, SI is used for MSR (Mining software Repositories) and the table above shows the main data mined from such. Although MSR is a transforming field instead of primarily being static record-keeping repositories they are transforming into active ones thanks to the “rich, extensive and readily available” software repositories. The lifecycle of a project can be improved by using SI to look beyond the coding phase and examine all stakeholders including managers, testers, developers and all support teams. It should be integrated into daily work environments and Integrated development environments. Instead of just historical data, SI should examine data including developer interactions with IDEs, notes taken in meetings (Using voice recognition tech) and support call recordings. Hassan outlines the steps that MSR should follow to enable the full capacity of the SI algorithm: To empirically investigate problems in software engineering domain, to highlight mining requirements regarding such problems and to adopt advanced mining algorithms [14]. SI can help on small scale by reviewing a particular code alteration to large scale issues such as re-designing a portion of the software. SI should leverage all types of repositories and enable a platform for various research into the work of Software engineers to revolutionise the way we develop code.

To Conclude, the vast variety of algorithms available are training algorithms to be capable of thinking and decision making even when knowledge is not perfect. The limits to these computations exist due to lack of emphatic and creative qualities but with an increase in artificial intelligence autonomous robotics will be able to think independently without human interaction. The algorithmic approaches will continue to simplify data and existing challenges and will help to identify trends that will improve society as a whole.

## 5. The ethics concerns surrounding this kind of analytics

Data Sovereignty

Data Sovereignty is the concept that data in digital form is subject to the laws of the country it is located. Countries such as Canada and EU members have strict data residency to ensure it remains within the country to protect their citizen's personal information. These sovereignty laws are a barrier to organisations adopting the cloud as it requires data to be managed and controlled to avoid breaching any rules and regulations.

Data Ownership

An important question nowadays is who actually owns the data online? If you make a post on Facebook do you retain ownership or do Facebook claim the rights? Having researched into ownership laws, it is evident that this is a grey area and there are no clear ownership rights. Our historic laws have focused oh physical assets which were non-replicable, so should we have a case to own our data online too? Most users of Facebook aren't aware that they recently changed their terms of service meaning users information is no longer erased when they delete their account and remains in the hands of the company forever.

The question of who owns your data is tricky to solve as you create data every time you leave your house or use your phone. Potential data points could be the number of steps you take, if you look at the ground when walking, what kind of clothes you wear etc, the average person is not aware of the extent to which they are being measured. Arguably, we haven't even discovered the vast data that could be recorded.

Boundaries

Location services on mobile phones track where you are at all times of the day. Snapchat introduced a new feature called "SnapMap" where you can tell where everyone of your friends are at an instantaneous moment in time. This raises the questions: Do we have any privacy anymore? Does it breach privacy boundaries?

An interesting law case concerning Mark Savage when he ran as a candidate for public office as councillor in 2014. 'Reddit.com' contained a URL that identified Mark as a homophobic candidate, which he felt affected his chances of election as well as presenting inaccurate information. Google refused to delete this URL as it was in public interests. The data protection law addressed the following questions in court:

1. Does the data subject play a role in public life? Is the data subject a public figure?
2. Is the data accurate?
3. Is the data relevant and not excessive?
4. Is it clear that the data reflects an individual's personal opinion, or does it appear to be verified fact?

5. Is it clear that the data reflects an individual's personal opinion, or does it appear to be verified fact? [22]

These questions I feel address the ethics of everyone's concerns about their data online. People don't mind being recorded but being recorded incorrectly is a major upset and can have disastrous effects. The court declared that Mr Savage's fundamental rights and interests had been prejudiced. In this example I feel that his 'right to be forgotten' was correctly judged as misleading information can destroy a person reputation.

Personal opinion

There is a company called target that used data in pharmacy stores analysing customers purchases. They managed to predict that a teenage girl was pregnant before her father found out. I feel stories like this show how effective data analytics can be but places a seed of worry for our privacy as humans.

In my honest opinion I am not against companies tracking my data provided it is an accurate measurement. As discussed already, algorithms can sometimes lead to false positives or false negatives due to inaccurate data. Bad effects of such could for example lead to dismissing an employee, refusal of a bank loan or false diagnosis of a patient. Therefore, I feel it is right to use algorithms and personal data to some extent but we have to have a boundary on how much can be tracked.

I believe that data analytic services can certainly offer organisations significant value by identifying trends and predicting consumer behaviour, ultimately making these companies faster, more competitive and more profitable. Concerning AI, I feel that humanity could start to become in danger. Autonomous robotics will replace our jobs and lead to higher levels of unemployment. Furthermore, Machine thinking could become so powerful even perhaps beyond human control.

Concerning Software repositories, there is a trade-off between "easily obtained analytics and richer analytics with privacy and overhead concerns". In general, the easier an analytic is to gather and the less controversial it is to use, the more limited its usefulness. Social acceptability is a key consideration when mining repositories. Ethics concerns involving software engineering would arise from algorithms distinguishing between good and bad employees or laziest employees etc. I believe it is in the company's right to know who their most productive employees are as analysing the way they go about their work is the best way for the company to develop. Analytics which I wouldn't agree with would be basing results of discriminating issues such as sex, age or nationality. If an engineer would like to keep some of their own data confidential, it should be the responsibility of the company to respect this. Philip Johnson proposes an approach to privacy concerns with collecting and analysing behavioural data. "Consider a cloudbase, independent, privacy-oriented analytics repository in which developers could maintain complete control over data and choose whether to provide management access" [23]. An ideology like this could be the way forward to continue to revolutionise software development through mining data but at the same time respect ethical concerns.

## 6.Conclusion

The software engineering process has developed over the past number of years due to metrics that have been in place that have facilitated the growth of software development. For this surge to continue, it is necessary to identify what makes a software engineer productive and beneficial. Data retrieval is very important in the process as the most useful resources in software companies are human resources. The first step is to collect the relevant data but in order for this to be successful, the goals and interests of the data collection should be established beforehand to reduce costs. Specific process such as PSP guide developers on how to track and record their work and nowadays analytical tools such as Hackystat and PROM automate this process providing analysis of the collected data. These autonomous tools will enable increased performance and product quality of software engineers.

Without a doubt, there is a high scale of data to be processed from software repositories such as Github and as a result vast companies exist to analyse this data. Companies like Sixth Sense try to turn this data into a strategic asset while Code Climate combats the fight against technical debt highlighting code that fails to add value but instead contains inadequate coverage and maintainability issues. This simplifies the software development process by making the process simpler and more efficient than before.

Machines are becoming more and more capable in modern society, with AI replacing the need for routine workers. Supervised and unsupervised algorithms empower the learning curve of these machines by making more powerful predictive models by analysing trends that a human might not have been able to spot. SI will aid developers with concrete facts and evidence concerning the development process permitting them to design evolutionary approaches. The lifecycle of a project can be improved by using SI to look beyond the coding phase and examine all stakeholders including managers, testers, developers and all support teams.

In terms of ethics, it is frightening the extent to which humans are recorded to in the digital age, with features such as location services perhaps violating privacy rights. In my opinion, I don't mind being recorded provided the data is accurate and doesn't damage one's reputation. Consumer data is immensely valuable but the important question isn't who owns the data but who owns the means to analysis, this is where Data Sovereignty comes into play. Concerning Software repositories, there is a trade-off between easily obtainable analytics and more advantageous analytics which touch on privacy concerns. I believe a software engineer should be able to keep some of their data confidential which the company should respect but the engineer must realise that analysing such data could add significant value to both the company's profits and his/her own productivity.

## References


*[1] https://www.linkedin.com/pulse/20140702104209-232985602-5-steps-of-software-engineering-process/*

*[2] Kan, S. (2014). Metrics and models in software quality engineering. [Place of publication not identified]: Addison-Wesley.*

*[3] Basili, V. and Weiss, D. (1984). A Methodology for Collecting Valid Software Engineering Data. IEEE Transactions on Software Engineering, SE-10(6), pp.728-738.*

*[4] Anon, (2017). [online] Available at: http://Collecting, Integrating and Analyzing Software Metrics and Personal Software Process Data [Accessed 21 Nov. 2017].*

*[5] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.6806&rep=rep1&type=pdf*

*[6] Humphrey, W. (2000). The Personal Software Process(sm) (PSP(sm)). Pittsburgh, Pa.: Carnegie Mellon University, Software Engineering Institute.*

*[7] https://codeclimate.com/*

*[8] https://codebeat.co/*

*[9] https://www.codacy.com/*

*[10]https://www.forbes.com/sites/oracle/2014/09/26/the-road-to-analytics-as-a-service/#174ee0a53622*

*[11]https://atos.net/wp-content/uploads/2017/10/01032013-AscentWhitePaper-DataAnalyticsAsAService.pdf*

*[12] https://6th-sense.in/*

*[13] http://idiro.com/*

*[14] http://bigdata-madesimple.com/17-important-case-studies-on-big-data/*

*[15]https://www.nitrd.gov/nitrdgroups/images/f/f1/Software_Intelligence_The_Future_of_Mining_Software_Engineering_Data_p161.pdf*

*[16] Kruse, R., Borgelt, C., Braune, C., Mostaghim, S. and Steinbrecher, M. (n.d.). Computational intelligence: A methodological Introduction.*

*[17] https://www.youtube.com/watch?v=Dp5_1QPLps0*

*[18] Sedgewick, R. and Wayne, K. (n.d.). Algorithms.*

*[19]https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/*

*[20] https://www.forbes.com/sites/forbestechcouncil/2017/04/11/is-data-sovereignty-a-barrier-to-cloud-adoption/*

*[21] https://lifehacker.com/you-dont-own-your-data-1556088120*

*[22] https://www.mhc.ie/latest/blog/first-irish-right-to-be-forgotten-case*

*[23] http://www.citeulike.org/group/3370/article/12458067*