

Measuring Software Engineering

In this report ,we will have to deliver a report that considers the ways in which the software engineering process can be measured and assessed in terms of measurable data ,an overview of the computational platforms available to perform this work, the algorithmic approaches available and the ethics concerns surrounding this kind of analytics.

What is the software process?

For different people process means different things, especially in software context, so it is important to clearly understand what term software process means.



The photo clearly shows that the process is cyclic. 6 Steps (Assess Needs, Design Specifications, Develop Test Software, Implementation, Support Operations). The

6th Step is where measuring the software engineering is important, Evaluate performance. In this step ,software engineers will have to be able to measure key values such as code coverage and system health in order to implement further systems. This shows measuring software process is a key aspect of a software engineering

How are software engineering process can be measured and assessed?

Software Engineering Measurement is key in the ever changing IT field. For Project managers ,they will need to make sure that the job is done well to ensure that the product or service is done reliably and that the work is done accordingly and tested properly.

They are very simple way to measuring data such as:

Number of Line:

There is two ways to calculate lines of code. One way is to calculate the actual source lines of code excluding the comments and the blank lines. This is obviously the wrong way to do it and I'll tell you why. For example, for iterating a statement a while loop will take more lines than a for loop however as a program the lets say the difference in 5 lines will have no effect on our program effectiveness . So Sadly, just counting the source line of code is not the right to measure the process of a software engineer. The only way to calculate the lines of code is measuring the number of " statements " such as in C programming languages is the number of statement terminated by a semi-colon.

Number of commits:

The concept of measuring commits, is that the more commits the better more reliable the program is. Obviously, it's not as simple. The number of commits is not the right way of doing things. As a commit can simply mean changing a space or indentation ,which is still important ,but doesn't have the same impact of if someone committed an actual feature of the program. The only way calculating the number of commits, could be used is to have a standard value for a commit for every commit for every software engineers working in that set program.

To summarize, measuring the number of sales/profits such as in the commercial field will be enough. However in the IT field this is not enough to measure the quality of work of a software engineer. Here is some more complex ways of measuring the process of a software engineer.

Instruction Path Length:

This way, it calculates the number of machine code instructions that the program needs to execute a program. Instruction Path Length shows the performance of a program and the amount of time it takes. This is a much better way to measure the process of a software engineer, it directly related to the performance of the program unlike measuring the number of lines or statements of the source code. This is a good start in measuring the process of a software engineer.

Code Churn

The percentage of a developer's own code were modified, added and deleted over a set period. Code churn allows software engineers to manage the software development process, especially its quality. When the percentage of code churn starts to spike, this is an indicator that the development process going well and the program is being polished. Code Churn rate could also identifies problems with individual developers. An increase in churn rate may indicate that a developer is experiencing trouble in working on a particular instruction.

Code Coverage

This shows how much of the code is been covered by test, showing that the program is ultimately been engineered properly. If the test are done properly done, the code should be bug-free. However, like every method of measuring the process has its faults, code coverage is dependent on test made by the software engineering so much the firms are starting to use computational program to measure the process of a software engineer. In my opinion, Code coverage is the best way to measure the process of a software engineer. Gives initiative to the software engineer to make sure the program is done reliably and isn't done for sake of just running code.














































What are the Computational programs/services available?

HackyStat

Hackystat is a program which is used to collect data on the software process and provides software engineers with goals based on the data collected by the program. Hackystat provides monitors which gives the full story of the code coverage, devtime and a lot more on a set of developers on a program. This is useful for project managers to make weekly reports on each of his engineers.

This toolkit is very good for monitoring the process of a software engineer.

Integrated in this project is the ability to create editors ,build tools .Events such as switching files editing methods, constructing test cases will all be monitored by the software

| Project (Members) | Coverage | Complexity | Coupling | Churn | Size(LOC) | DevTime | Commit | Build | Test |
|------------------------|--|---|---|--|--|--|--|---|---|
| DueDates-Polu (5) |  63.0 |  1.6 |  6.9 |  835.0 |  3497.0 |  3.2 |  21.0 |  42.0 |  150.0 |
| duedates-ahinahina (5) |  61.0 |  1.5 |  7.9 |  1321.0 |  3252.0 |  25.2 |  59.0 |  194.0 |  274.0 |
| duedates-akala (5) |  97.0 |  1.4 |  8.2 |  48.0 |  4616.0 |  1.9 |  6.0 |  5.0 |  40.0 |
| duedates-omaomao (5) |  64.0 |  1.2 |  6.2 |  1566.0 |  5597.0 |  22.3 |  59.0 |  230.0 |  507.0 |
| duedates-ulaula (4) |  90.0 |  1.5 |  7.8 |  1071.0 |  5416.0 |  18.5 |  47.0 |  116.0 |  475.0 |

The software UI gives indications of:

Dev Times: which portrays the time spent by the programmer on his IDE

Commit: the number of commits made in the repository

Built: the number of builds and how many of them were a success

Test: Success rate of the test cases

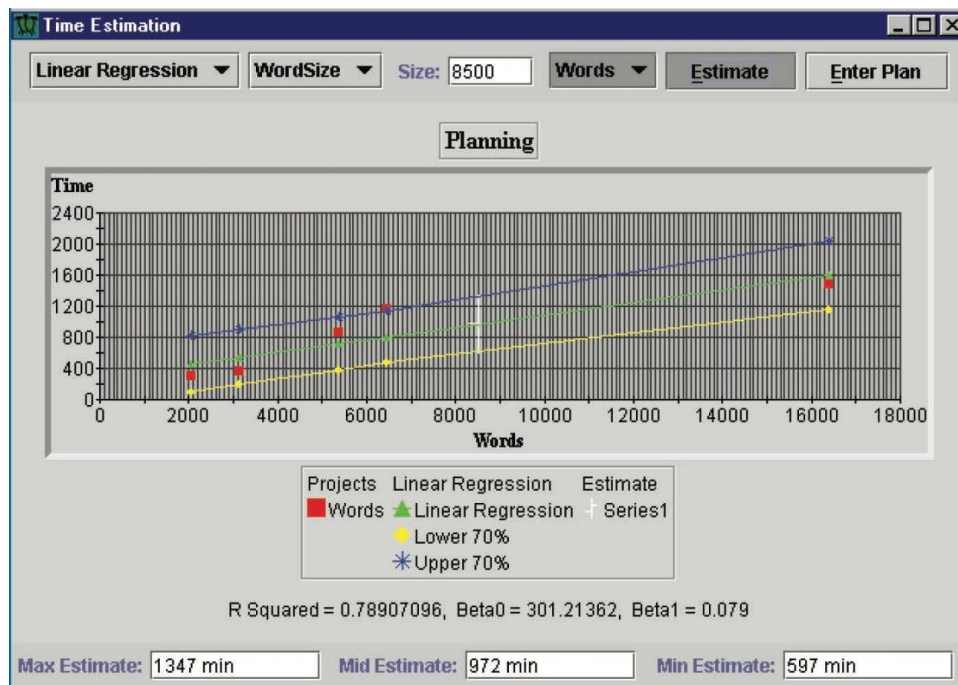
PSP

Created by Watts Humphrey, The Personal Software Process, assists software engineers to better understand and improve their performance by keeping track

of their predicted development of their code. The idea of this toolkit is to give software engineers the necessary skill to work on projects in a team

Leap

Developed by Philip M. Johnson, Lightweight, Empirical, Anti-measurement dysfunction and Portable tries to portray and analyze personal data concerning time, size, defects and checklist. It enables the developers to maintain the control of their data files and records data only on the individual. The programmer has to manually put in its data, which affects productivity. Unlike Hackstat this program would be used for individuals to monitor their own work, they might report back to a manager. Leap is a toolkit for more individuals based projects and less for big projects where there's more than one engineer working on the project. This tool attempted to rectify the problems with the quality of data being collected by its successor, through automating and normalizing data



The graph will give the estimate time that the developer will take to code a set number of lines/codes.

What are the algorithms used in the process?

The programs listed above only are used really as statistical analysis of measuring the software engineer. However, more analysis should be made by the project manager to effectively measure the software engineer process.

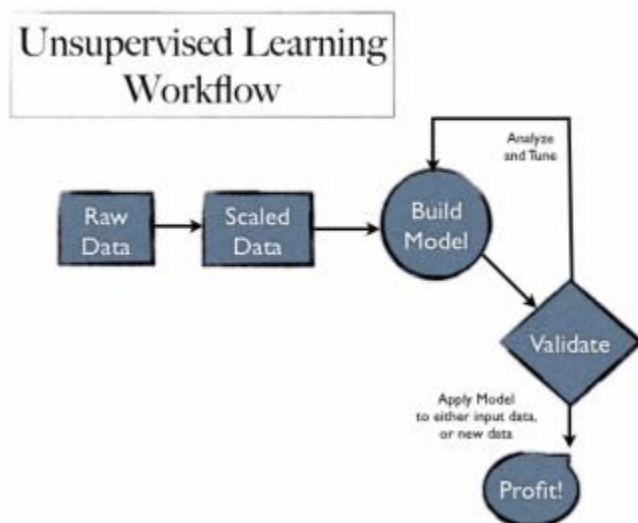
Machine Learning An Application of artificial intelligence that provides systems the ability to learn and improvise without being programmed

Unsupervised Learning:

Type of machine learning algorithm used to conclude trends from data without a labeled response.

Starts with analysis of a known trained dataset, the learning algorithm produces predictions about output values. The algorithm provides goals for any new input.

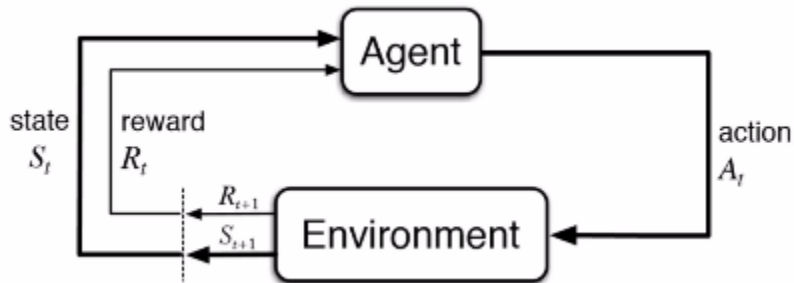
A big example of this cluster analysis used to find hidden patterns in data analysis. This will be very useful in measuring the process of a software engineer. As the project manager will be able predict outcomes and patterns to help prevent any mistakes in the processes



Reinforcement Learning:

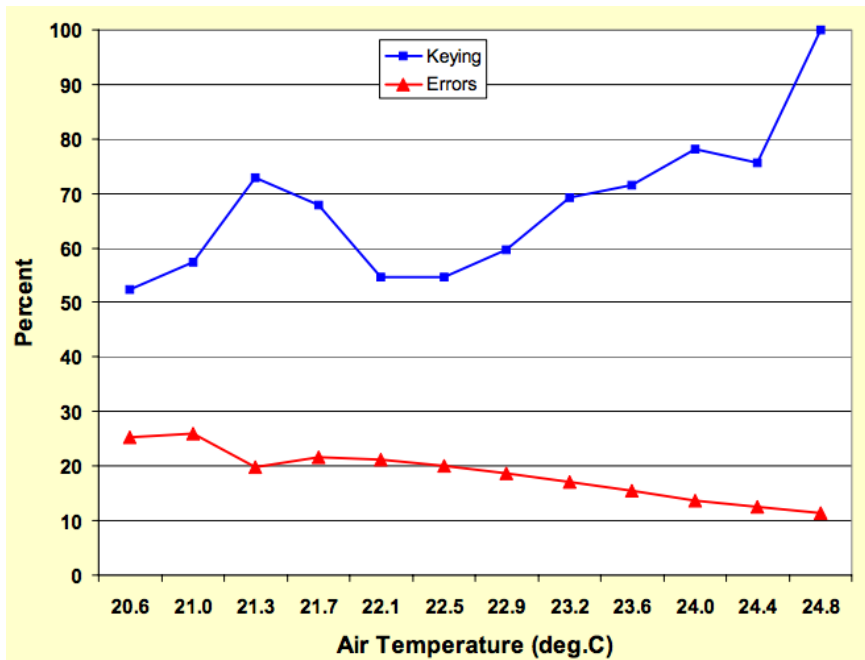
Type of machine learning algorithm allowing the machine to identify behaviors based on feedback from the environment. That behavior can be learnt or can

carry adapting as time goes by. If the problem is created cautiously ,some reinforcement learning can help prevent those problems happening again. Finding the ideal behavior that will maximizes the reward. This will pretty much make Project managers redundant or increases the productivity of the process as less mistakes are made.



Productivity:

Happiness and heat to list a few are outside factors that will greatly effect productivity. The 1/T rule (the probability of a person changing from active to inactive is inversely proportion to the duration of the activity) shows that when employees are happy they are found to have a lower degree of deviation .



The graph clearly shows that heat has a smaller impact on error-making in productivity as well in Summer, so increase in temperature and daylight (less melatonin directly related to cause depression) productivity is at its highest theory wise. However, distractions such as employees take more holidays during summer has a say in this.

System Health:

Using the statistical analysis from the computational programs, If the project managers sees week by week the level of code coverage regressing or decreasing such as new test-cases not being dealt with. This will cause problems in the software process. System health could identify major problems in productivity and the level of commitment of the programmers.

What are the ethics involved in the software process?

Ethics are moral principles that govern a person's behavior or the conducting of an activity

Security& Privacy:

Using unauthorized transmission and use of personal data, many apps/software ,such as UBER ,will have to know the users personal information like location and credit card information. For privacy issues, the software process will have to ensure that they will have the necessary level of security. If any leaks are made especially credit card numbers leaks. The company that would leak such information will lose the trust of their clients. If the company doesn't have the sufficient Public relation resources they would lose big in a business point of view as well. For the client ,a mistake like that could mean if not insured the client could lose money out of a mistake that he hasn't made. For me, that shows that something is ethically wrong when someone losing out for something he hasn't any say in. Sometimes for less known apps ,they will collect personal information and sell them to marketing companies. Although ,there is a mandate of laws that doesn't permit this but Privacy should be a big factor in the software process. Measures such as encryption ,authentication and firewalls will have to be implemented.

To summarize, the impact of Ethics on the software process shouldn't be put to side. Outside factors such as cyber-terrorism will always try to find a way to break those firewalls and crack the encryption. Cyber-terrorist will then sell this data to the black market or keep it for their own benefit . So the software engineer will have to keep up to date with the new technology of stronger security. To ensure that the software process can stay alive. If Ethics are under valued by the software engineers the program will ultimately suffer and might die. This will lead the software engineer being more motivated that his work is at stake if he doesn't respect the code of ethics stated by the company. However who is determining the violations? How are we ensuring he is also complying with the set code of ethics?. Ethics will always be a grey area. However, I know as a software engineer respect is key .We need to respect each other and that for me is what ethics is all about.

Now Ethics directly concerning the analytics taken from the measurement of the process of a software engineer. Plagiarizing is a big issue, project managers will have to make sure work from a certain developer doesn't get exploited, as to make the plagiarizer look better than he is. Also him not knowing the way the code works can lead to more bugs and errors.

It is to the project manager responsible to set a code of ethics. This code should ensure that the privacy of the analytics taken should be kept as a secret from other developers ,so only the managers will be able to see the measurements. The code of ethics should also ensure that all parties know the violations and the punishments. Ethics is domain dependent in my view but once a code of ethics is stated it should be respected. However, Respect goes both ways in which the code of ethics should protect and outline the core values of both the managing values and the developers values. Values of such are of course Respect, Privacy and more

References:

https://www.youtube.com/watch?v=Dp5_1QPLps0

https://en.wikipedia.org/wiki/Source_lines_of_code

https://www.researchgate.net/publication/228731372_Experiences_withhackystat_as_a_service-orientred_architecture

<https://pdfs.semanticscholar.org/d0c3/5a51a50bd8f348a8bb2d0298b5ba83cec0.pdf>

https://en.wikipedia.org/wiki/Personal_software_process

<https://www.mathworks.com/discovery/unsupervised-learning.html>

<https://www.slideshare.net/lemiorhan/professional-code-of-ethics-in-software-engineering>

