



SOFTWARE ENGINEERING REPORT

CS3012



NUALA NÍ DHOMHNAILL
16321005

Abstract:

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

1. Importance of Software Measurement

Software Measurement is a discipline within the software development cycle and within the Software industry as a whole.

“Without measurement there can be no measurable improvement”.

In order to make an informed and effective decision concerning the further development of a project, it is crucial to record and analyse data acquired from the software development process. However, it is not always clear which data should be used in our measurements. Do we measure a projects progress based on the length of the code None the less, in an ever-changing world, improvement is a rare gem to find and in order to find it, we must measure our progress?

There are many arguments as to which method of measurement is the most effective in assessing the development of a software, I will discuss these further in this report.

2. Measurable Data

A. Source Lines of Code (SLOC)

Using the length code to assess a project is one of the most popular methods and is without a doubt the easiest method. Data such as the contribution made by project team member can be automatically accessed, processed and analysed. There are two major types of SLOC:

- Physical SLOC (LOC) - includes blank lines in the measurement, unless they contribute to greater than 25% of a block of code
- Logical SLCO (LLOC) - measure the amount of statements and excludes blank lines

Although comparing the length and complexity of the code to other similar projects provides insight as to how the project is performing in regard to its competition, it does not however provide any information on how the project has improved from its last evaluation. Therefore using the Lines of Code methodology has come under scrutiny by well known, successful developers.

“Measuring programming progress by lines of code is like measuring aircraft building progress by weight.” – Bill Gates

This is due to the many disadvantages associated with the SLOC metric.

- The experience of the developer can influence the volume of code used to complete a simple task
- The difference in languages – due to limitations in certain languages a higher volume of code may be needed to complete similar tasks
- A programmer whose value to the project is being measured on code creates an incentive for the programmer write unnecessarily long blocks of code

Programming language	SLOC/FP	Programming language	SLOC/FP
ASP / ASP.Net	69	Java	53
Assembly	119	Javascript	47
Shell / DOS Batch	128	PHP	67
C	97	Pascal	90
C#	54	Python	24
C++	50	SQL / make	21
HTML / CSS / XML / XSLT	34	Visual Basic	42

Table 2.1 SLOC to function-point ratio for various programming languages

B. Measure Outcomes

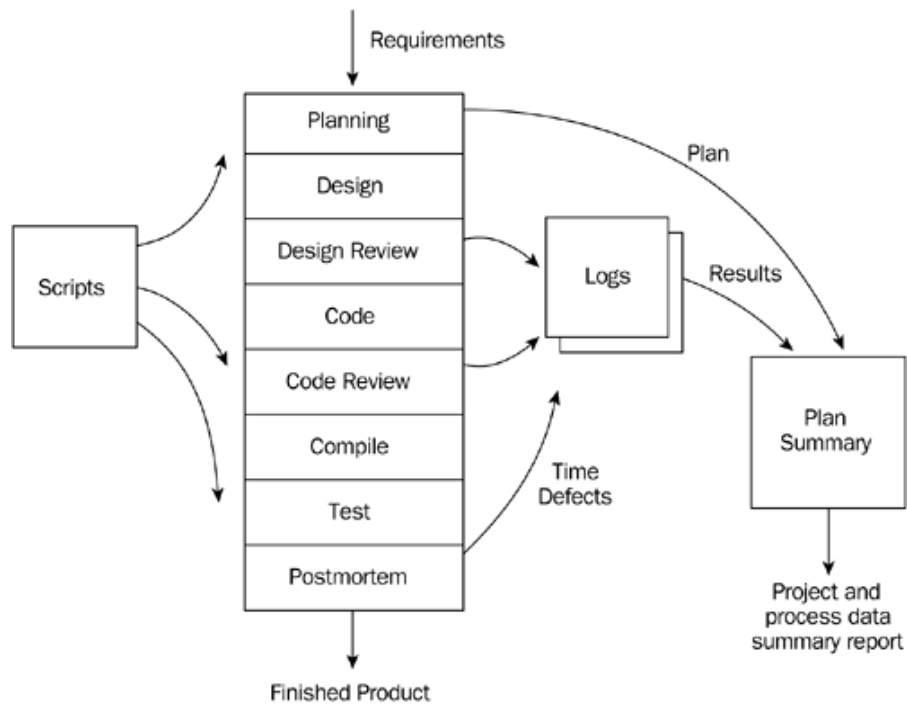
There is an argument to be made that in order to measure effective software engineering, we must stop measuring productivity. However, if we are not to measure productivity of a software team, how can we know if they are working effectively and efficiently? By measuring the outcomes and goals achieved. But how do we measure the outcomes of an engineering project?

In order to be able to measure and assess the outcomes of a software project we must combine the engineering department with other business functions to create a strong partnership. This links engineering to the business cases and it becomes evident what the outcomes are of the engineering project.

An advantage to using this method of measurement is that most outcomes are not hard to comprehend. For example, breaking the market successfully, growing in the market, gaining competitive advantages. Unfortunately, this method has disadvantages to level its advantages. When measuring outcomes software engineers can easily become focused on timelines, the amount of code written and features, rather than the wider perspective. The finished project.

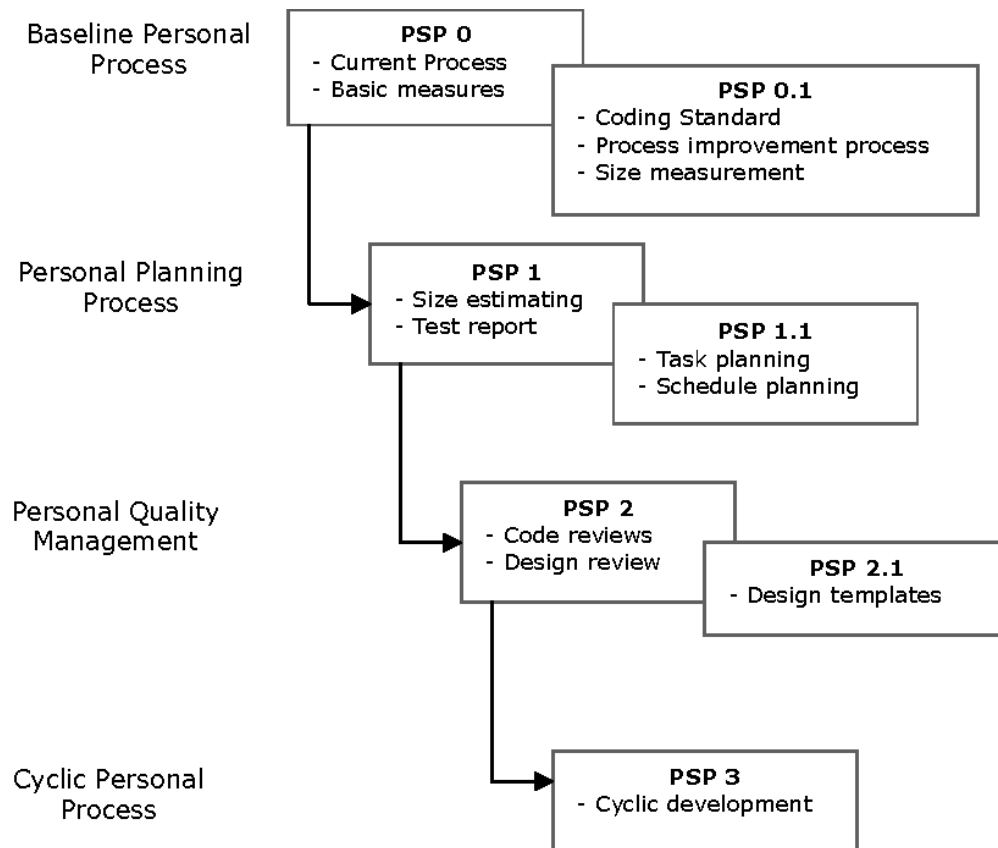
C. Personal Software Project.

The personal software process fixates on the individual developers on the team. It was developed to help the engineers to enhance their performance and productivity by constantly tracking and comparing the predicted and actual improvements carried out.



The process consists of a set of methods, forms, and scripts. It has been designed so that it is compatible with any programming language or design methodology. It can be used for the majority of software work, including writing requirements, running tests, defining processes, and repairing defects. This gives this metric of measurement immense flexibility.

The main objective of the personal software process is to minimise the defects of the final software product. The process is designed to enable the developers to spot the defects within the code at an earlier stage, thus decreasing the time spent in fixing them



3. Computational Platforms

Although software metrics are great for measuring and tracking an engineering teams progress, such metrics are not useful if there are not collected efficiently and analysed properly. Fortunately, there are a multitude of platforms which enable software engineers to track these metrics.

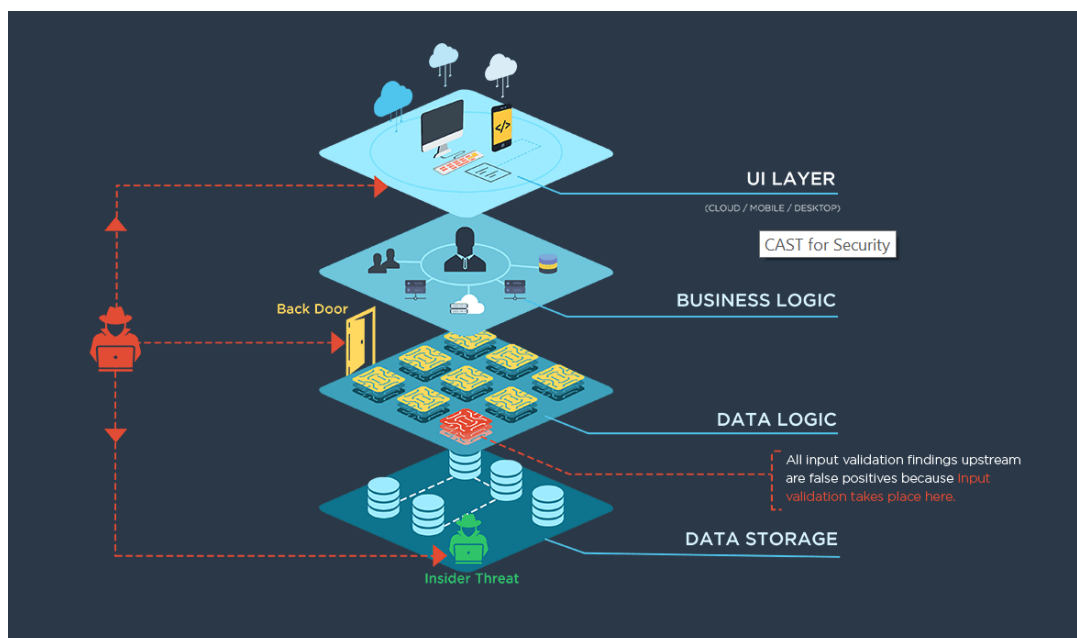
A. Hackystat

“Hackystat is an open source framework for collection, analysis, visualization, interpretation, annotation, and dissemination of software development process and product data”

It is designed to reduce the overhead costs of collecting and analysing of the raw, low level data by automating the process. The platform also provides a range of useful tools for creating software reports based on the data collected.

B. CAST API

CAST technology creates a digital image of the internal composition of software. The CAST solution inspects source code, identifies and tracks quality issues, and provides the data to monitor development performance. CAST measures the quality of the code and ensures that it mimics the coding standards by analysing all levels of complicated applications.



4. Ethics

“Ethics is a set of moral principles that govern the behaviour of a group or individual.”

With the wide availability of big data today, intellectual property and privacy concerns are constantly being discussed.

A. General Data Protection Regulation

The GDPR’s role in society is to reshape, maintain and regulate the way sectors such as managed their data, more importantly, data concerning the public.

“The aim of the GDPR is to protect all EU citizens from privacy and data breaches in today’s data-driven world. ”

If an organisation were to breach these regulations, they could be fined up to 4% of their annual turnover or €20 million, whichever sum is the highest. This is the maximum fine which an organisation can receive and is imposed for serious infringements.



References

- Barnes, D. and Barnes, D. (2018). *The Myth of Developer Productivity*. [online] Dev9. Available at: <https://dev9.com/blog-posts/2015/1/the-myth-of-developer-productivity> [Accessed 20 Nov. 2018].
- Citeseerx.ist.psu.edu. (2018). [online] Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.4352&rep=rep1&type=pdf> [Accessed 20 Nov. 2018].
- Code Red. (2018). *You Can't Measure Software Engineering Productivity, so Measure Job Satisfaction Instead*. [online] Available at: <https://redfin.engineering/measure-job-satisfaction-instead-of-software-engineering-productivity-418779ce3451> [Accessed 20 Nov. 2018].
- Eugdpr.org. (2018). *Key Changes with the General Data Protection Regulation – EUGDPR*. [online] Available at: <https://eugdpr.org/the-regulation/> [Accessed 20 Nov. 2018].
- Flylib.com. (2018). *Goals and Focus | Software Development for Small Teams: A RUP-Centric Approach (The Addison-Wesley Object Technology Series)*. [online] Available at: <https://flylib.com/books/en/4.311.1.104/1/> [Accessed 20 Nov. 2018].
- Hackystat. (2018). *Hackystat*. [online] Available at: <https://hackystat.github.io/> [Accessed 20 Nov. 2018].
- Irma-international.org. (2018). *IRMA-International.org: Open Innovation: Assessing the Socio-Economic Factors of Global Software Development: Noel Carroll: Book Chapters*. [online] Available at: <http://www.irma-international.org/chapter/open-innovation/117292/> [Accessed 20 Nov. 2018].
- Projectcodemeter.com. (2018). *Source lines of code (LOC, SLOC, KLOC, LLOC)*. [online] Available at: http://www.projectcodemeter.com/cost_estimation/help/GL_sloc.htm [Accessed 20 Nov. 2018].
- Team, E. (2018). *For Engineering Performance, Stop Measuring Productivity - WorkSpace Today*. [online] WorkSpace Today. Available at: <http://theworkspacetoday.com/2017/01/13/for-engineering-performance-stop-measuring-productivity/> [Accessed 20 Nov. 2018].