ISBSG
Delivering IT Confidence

The global and independent source of data and analysis for the IT industry

Short paper

Measuring Team Performance

to understand High-Performing and Low-Performing teams

Published by the International Software Benchmarking Standards Group

2022

Measurement of software development

In many industries, decisions are made based on data. Important KPIs are measured over time and the data may show a certain event or pattern that is the trigger for an action. A simple example is the throughput (i.e. the number of products ready per minute) of an important assembly line in a factory. When the throughput drops below a certain threshold, this indicates that something is wrong. So, a repair crew will determine the problem and repair it.

In the software industry, often people speak about 'software factories'. These teams focus on software development. They help to develop the value that organizations offer their customers. Examples are: new features in a mobile application, new functionality on a website, or changes to the ERP system.

Managers of software development and maintenance teams also wish to make decisions based on data, but this is a lot more difficult than the factory example. How can you measure the output of software development so that effective decisions can be made?

To solve this issue, the ISO standards for functional size measurement have been developed. Currently there are 5 different methods that are ISO certified (in alphabetical order).

- COSMIC-FFP (ISO/IEC 19761)
- FiSMA (ISO/IEC 29881)
- IFPUG (ISO/IEC 20926)
- Nesma (ISO/IEC 24570)
- UKSMA-Mark II (ISO/IEC 20968

Of these methods, IFPUG, Nesma and COSMIC are the most commonly used, worldwide. These methods are used to measure the functional size of a piece of software by quantifying its amount of functionality, regardless of technology or non-functional requirements. Just as a brick wall of 100 square meters has the same surface area as a glass wall of 100 square meters, a software system written in Java of 1000 function points offers the same amount of functionality as an application of 1000 function points, developed in Cobol.

There are many experts worldwide that are certified analysts in one or more of the functional size measurement methods mentioned above. These analysts can analyze a set of functional user requirements (e.g. user stories, use cases, functional designs, etc.) and determine the functional size of those. This means that it is also possible to determine the functional size added, modified, and/or removed in a certain time-period, such as a sprint or release. This is developed by Nesma and part of its standard. Nesma calls this 'the project size', and the unit of measurement is called 'Enhancement Function Points (EFP)'. Therefore, there are 2 types of functional size:

- Application size: the functional size of an application at a certain moment in time. For example, Application XYZ is 1000 FP at the moment.

- Project size: the functional size added, modified and/or removed in a certain timeframe. In the example of sprint XYZ, 25 FP were added, 15 were modified and 5 were removed. The project size was 25 +15 +5 = 45 Enhancement Function Points (EFP).

Determining Team metrics

This is great, because now it becomes possible to measure the throughput of software development teams. By measuring the number of EFP per sprint or per release, differences in throughput per team can be understood and compared. This provides a better understanding when making informed decisions, especially when data is used. When for instance, sprint or release data is collected of the effort hours spent, the cost of these hours and the number of defects found can be found. The following standard metrics can be determined:

- Project Delivery Rate: effort hours spent per EFP
- Cost Efficiency: Cost of effort hours spent per EFP
- Process Quality: Defects per 1000 EFP
- Delivery Speed: EFP per month

Monitoring these metrics per team provides insight and understanding as to what is happening in the team. When fewer EFP are developed, this may signal that there is an issue.

Function Points can be regarded as a proxy for value produced. The more functionality a team produces, the more the users can use that functionality. Of course, prioritizing the functionality is an important prerequisite to use the proxy this way.

Once these metrics are measured, it becomes possible to compare the teams to each other. Management can then see the teams that produce better value than others.

Often, productivity is used as a basis of comparison because it is independent from hourly rates or internal cost. It's important to realize that in ISBSG terminology, productivity is often used instead of the correct term Project Delivery Rate (PDR). For humans it's hard to work with numbers with a lot of decimals, which is what you get when calculating function points per hour, so therefore this calculation is reversed into hours per function point and called PDR. An example of the comparison between teams' PDRs is shown in Figure 1.

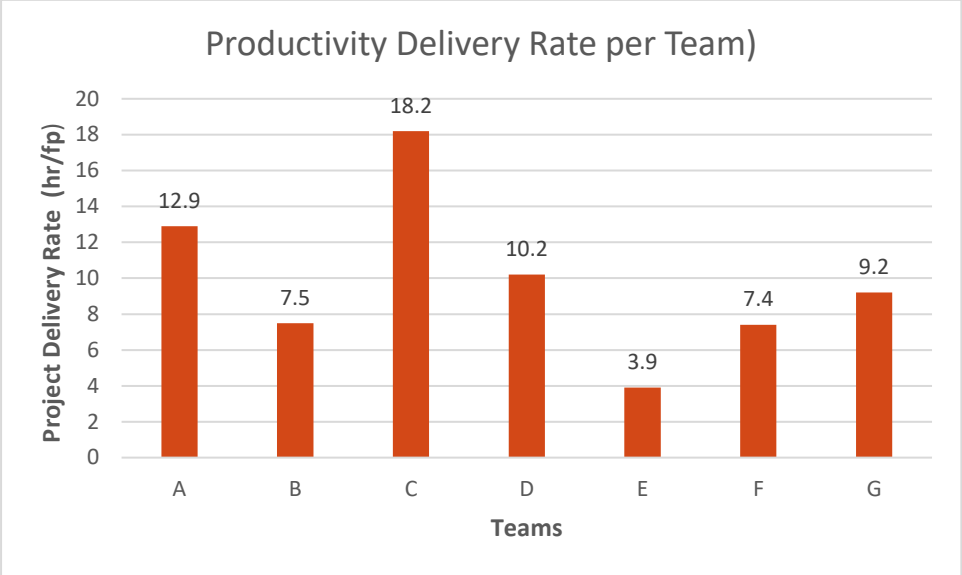

Figure 1: Comparing the project delivery rate of teams at a given time

Figure 1 provides limited insights on team productivity. Team A is, for example, a Java team of 8 people, while team C is a Cobol team of 5 people. Productivity of a team largely depends on the technical environment. Therefore, it helps to compare each team's productivity to market averages in the industry.

Using ISBSG data to understand team performance

The ISBSG collects industry data, where output is measured using ISO/IEC standardized and therefore objective, repeatable, auditable methods, such as Nesma, IFPUG and COSMIC function points. The ISBSG 'New Developments & Enhancements' repository contains thousands of completed projects, releases and sprints for which these metrics are calculated, enabling organizations to use this industry data for fact-based understanding and decision making.

To understand the PDR performance of team A, we can compare it to industry averages. Figure 2 shows how to filter the ISBSG dataset, which is distributed as a MS Excel file. In this case, the following filters are applied:

- Data Quality Rating = A or B
- Project Year > 2015
- Primary Programming Language = Java
- Count Approach: IFPUG 4+ or Nesma (These methods are basically identical)
- Relative Size: XS or S (10 – 100 FP)
- Project Type = Enhancement

ISBSG Project ID	Rating	Rating	Software Age	Software Age	Major Grouping	Major Grouping	Major Grouping	Major Grouping	Major Grouping	Gro	Major Grouping	Major Grouping	Major Grouping	Sizing	Sizing
	Data Quality Rating	UFP rating	Year of Project	Year Range	Industry Sector	Organisation Type	Application Group	Application Type	Development Type	De	Language Type	Primary Programming Language	Count Approach	Functional Size	Relative Size
10007	B	B	2016	2016-2020	Communication	Telecommunications;	Business Application	Customer relationship management;	Enhancement		3GL	Java	IFPUG 4+	51	S
10049	B	A	2018	2016-2020	Construction		Business Application	Financial Transactions;	Enhancement		3GL	Java	NESMA	1.323	L
10331	B	A	2018	2016-2020	Communication	Telecommunications;	Business Application	Other;	Enhancement		3GL	Java	IFPUG 4+	119	M1
10335	B	B	2018	2016-2020	Services		Business Application	Business Intelligence;	Enhancement		3GL	Java	NESMA	1782	L
10392	B	B	2016	2016-2020	Communication	Telecommunications;	Business Application	Customer relationship management;	Enhancement		3GL	Java	IFPUG 4+	132	M1
10467	B	A	2018	2016-2020	Communication	Telecommunications;	Business Application	Other;	Enhancement		3GL	Java	IFPUG 4+	52	S
10546	B	B	2016	2016-2020	Communication	Telecommunications;	Business Application	Customer relationship management;	Enhancement		3GL	Java	IFPUG 4+	86	S
10639	B	B	2017	2016-2020	Manufacturing		Business Application	Document management;	Enhancem	Plot Area	3GL	Java	NESMA	366	M2
10784	B	B	2018	2016-2020	Mining		Business Application	Logistics;	Enhancement		3GL	Java	NESMA	214	M1
10960	B	A	2018	2016-2020	Communication	Telecommunications;	Business Application	Other;	Enhancement		3GL	Java	IFPUG 4+	107	M1
11014	B	B	2016	2016-2020	Communication	Telecommunications;	Business Application	Customer relationship management;	Enhancement		3GL	Java	IFPUG 4+	66	S
11058	B	B	2018	2016-2020	Manufacturing		Business Application	Business Intelligence;	Enhancement		3GL	Java	NESMA	889	M2
11145	B	A	2020	2016-2020	Communication	Telecommunications;	Business Application	Other: Service Order & Activation Man	Enhancement		3GL	Java	IFPUG 4+	322	M2
11266	B	B	2019	2016-2020	Finance		Business Application	Database System;	Enhancement		3GL	Java	NESMA	413	M2
11457	B	B	2019	2016-2020	Utilities		Business Application	Report Generation;	Enhancement		3GL	Java	NESMA	357	M2
11530	B	B	2016	2016-2020	Communication	Telecommunications;	Business Application	Customer relationship management;	Enhancement		3GL	Java	IFPUG 4+	118	M1
11542	B	B	2018	2016-2020	Manufacturing		Business Application	Logistics;	Enhancement		3GL	Java	NESMA	292	M1
11678	B	A	2017	2016-2020	Communication	Telecommunications;	Business Application	Other;	Enhancement		3GL	Java	IFPUG 4+	70	S
11684	B	A	2019	2016-2020	Communication	Telecommunications;	Business Application	Other;	Enhancement		3GL	Java	IFPUG 4+	104	M1
12014	B	A	2019	2016-2020	Government		Business Application	Financial Transactions;	Enhancement		3GL	Java	NESMA	2.041	L
12083	B	B	2018	2016-2020	Wholesale		Business Application	Transaction Processing;	Enhancement		3GL	Java	NESMA	722	M2
12117	B	B	2016	2016-2020	Communication	Telecommunications;	Business Application	Other;	Enhancement		3GL	Java	IFPUG 4+	174	M1
12274	B	A	2017	2016-2020	Communication	Telecommunications;	Business Application	Content Management;	Enhancement		3GL	Java	IFPUG 4+	76	S
12661	B	A	2020	2016-2020	Finance		Business Application	Expert System;	Enhancement		3GL	Java	NESMA	1.274	L
12744	B	B	2016	2016-2020	Communication	Telecommunications;	Business Application	Other;	Enhancement		3GL	Java	IFPUG 4+	91	S
12798	B	B	2016	2016-2020	Communication	Telecommunications;	Business Application	Customer relationship management;	Enhancement		3GL	Java	IFPUG 4+	110	M1
12849	B	B	2017	2016-2020	Utilities		Business Application	Customer relationship management;	Enhancement		3GL	Java	NESMA	765	M2
12954	B	B	2019	2016-2020	Retail		Business Application	Document management;	Enhancement		3GL	Java	NESMA	562	M2
12989	B	B	2016	2016-2020	Communication	Telecommunications;	Business Application	Customer relationship management;	Enhancement		3GL	Java	IFPUG 4+	98	S
13018	B	A	2017	2016-2020	Communication	Telecommunications;	Business Application	Content Management;	Enhancement		3GL	Java	IFPUG 4+	221	M1
13056	B	B	2018	2016-2020	Utilities		Business Application	Business Intelligence;	Enhancement		3GL	Java	NESMA	360	M2
13121	B	B	2019	2016-2020	Manufacturing		Business Application	Computer aided design;	Enhancement		3GL	Java	NESMA	964	M2

Figure 2: screenshot of the filtering of the data in the ISBSG repository

Filtering the data results in 341 data points. Figure 3 shows the distribution of this data.

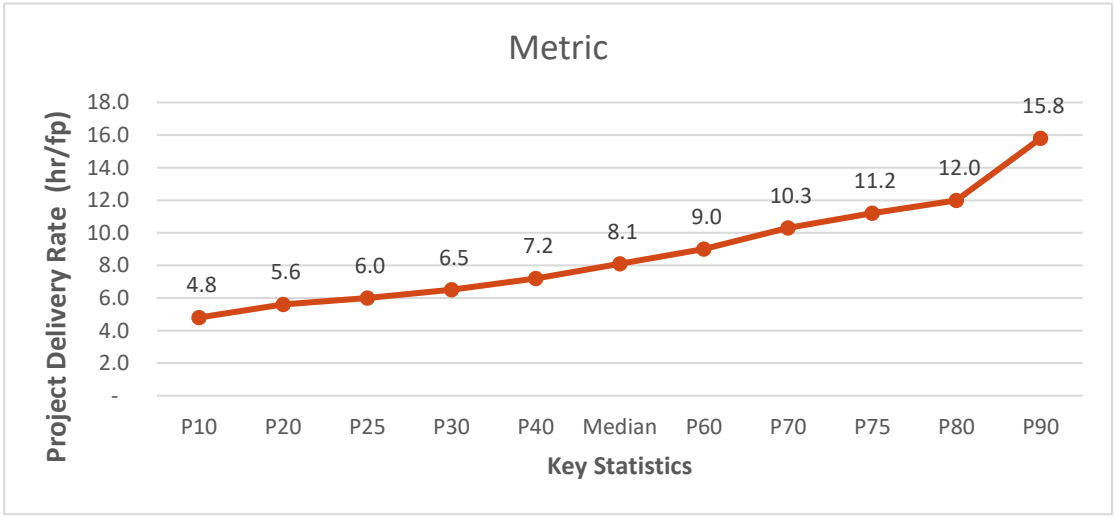

Figure 3: the distribution in PDR of Java projects in the ISBSG 2021 D&E Repository

As these distributions are often skewed to the right, we use the median to indicate the market average instead of the average. So for this dataset, the median is 8.1 hours/EFP, which we then refer to as the 'market average'.

Now we can assess the performance of the Java team (Team A), which is 12.9 hours/EFP, which falls between the 80th percentile and 90th percentile in the distribution. When analyzing PDR values, a lower value indicates that fewer hours were spent to create a function point. So, the performance of team A is not very good compared to the market average in the industry.

The productivity index is a metric that can be used to compare the team PDR to the industry average. This metric is calculated: 1- (Team PDR / Market Average PDR). In

this case, the productivity of team A is 1 – (12.9 / 8.1) = -60%. So, the team performs 60% worse than market average.

When calculated for all the teams, it highlights the high-performing and low performing teams. An example is given in Figure 4, where the trends in the productivity index of 12 teams are shown.

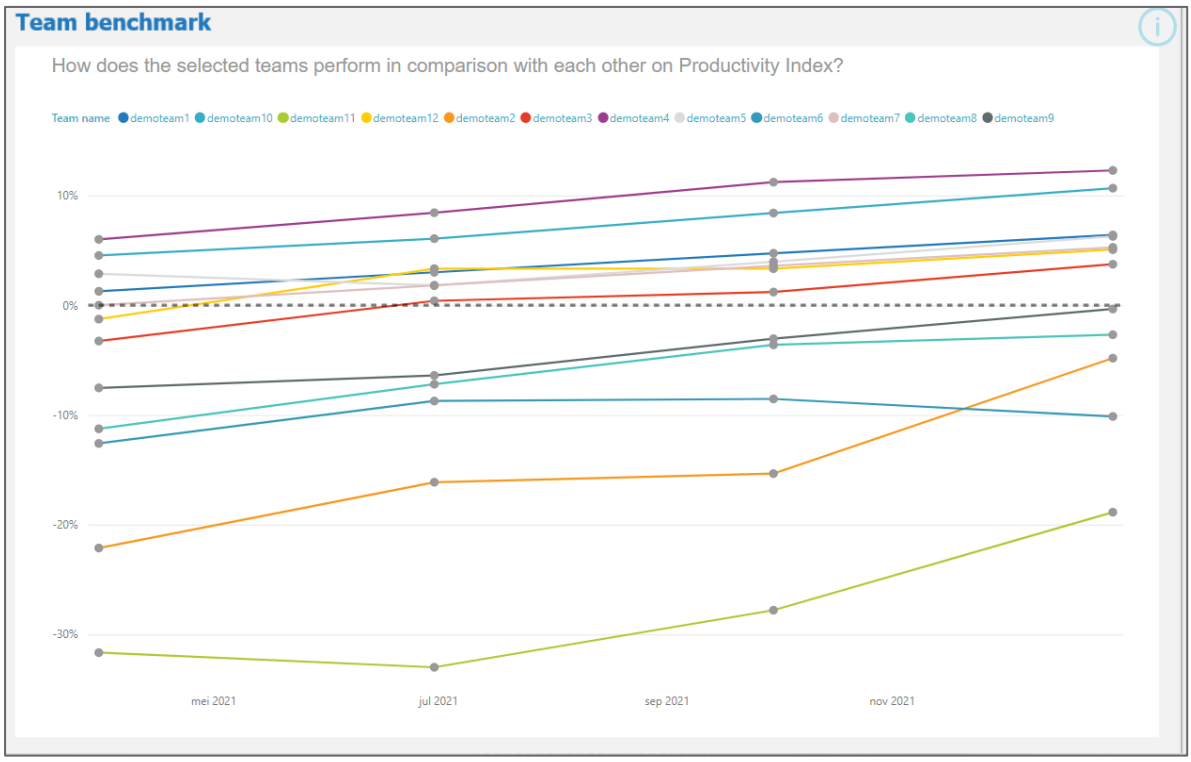

Figure 4: comparing the productivity index of 12 teams to understand high and low performance teams

In this figure the productivity index is measured 4 times during the year. The dotted line in the middle at 0% indicates the market average, which is 0%. In this figure, it's easy to understand which teams outperform the market and which are less productive. This insight can be the starting point to understanding what the reasons are for this. For instance, maybe the high performers have adopted certain best practices, or maybe they have a different mix of experience and skills. However, these measurements should never be used to penalize teams or individuals. Instead they can be used as a basis for improvement. Also, in the case of completely external teams, it becomes possible to use these metrics in contracts, so that the vendor is paid based on value produced, instead of the hours of effort expended.

Conclusions

It's hard to measure the output of software development teams, but the introduction of Enhancement Function Points (EFP) by Nesma has helped to establish the amount of functionality (value) produced by the teams.

Combined with data, it is possible to determine several important metrics: project delivery rate (the inverse of productivity), cost efficiency, delivery speed and process quality.

The trends in these metrics already provide much insight to management, but when combined with a comparison to the industry data of **ISBSG**, the value increases enormously.

Selecting a relevant dataset is easy in the ISBSG excel spreadsheet, and the distribution of the data shows where the teams have performed.

Then it becomes possibly to calculate the performance against the industry average (median). This results in an understanding per team how they have performed against the industry per metric. Especially the trends in these metrics give important management information whether the teams are improving or not.

This insight helps for management to understand which teams are performing well and which teams are not performing so well. This should never be used to punish but should be the starting point to improve.

The International Software Benchmarking Standards Group (ISBSG)

The ISBSG is a not-for-profit organization founded in 1997 by a group of national software metrics associations. Their aim was to promote the use of IT industry data to improve software processes and products.

ISBSG is an independent international organization that collects and provides industry data of software development projects and maintenance & support activities in order to help all organizations (commercial and government, suppliers and customers) in the software industry to understand and to improve their performance and decision making. ISBSG sets the standards of software data collection, software data analysis and software project benchmarking processes and is considered to be the international thought leader in these practices.

The ISBSG mission is to support commercial and public organizations to improve the estimation, planning, control and management of IT software projects and/or maintenance and support contracts.

To achieve this:

ISBSG maintains and grows 2 repositories of IT software development/maintenance & support data. This data originates from trusted, international IT organizations and can be obtained for a modest fee from the website www.isbsg.org/project-data/

Help us to collect data

ISBSG is always looking for new data. In return for your data submission, we issue a free benchmark report that shows the performance in your project or contract against relevant industry peers.

Please submit your data through one of the forms listed on http://isbsg.org/submit-data/

A specific Agile/Scrum data collections questionnaire can be downloaded here: https://cutt.ly/4vnuXVT

Partners

This page will help you to find an ISBSG partner in your country:
https://www.isbsg.org/board/