ISBSG
Delivering IT Confidence

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

Industry Report

Practical Guide: Productivity Measurement of Software Projects

Published by the International Software Benchmarking Standards Group

08-2019

Introduction

Report Description

In this industry report, Challenges in Productivity Measurement in the software industry are addressed and the way ISBSG data can be used to benchmark productivity of completed projects, releases and sprints is explained.

About the ISBSG

The International Software Benchmarking Standards Group is the international independent not-for-profit organization that collects data of software projects (new developments, enhancements, releases and sprints) from the industry.

The ISBSG Mission

The mission of ISBSG is to collect data in order to help decision makers in the software industry to make better decisions based on data. To quote William Edwards Deming: "*Without data, you are just another person with an opinion*". The ISBSG mission is supported by its partners, who represent IT and Metrics organisations and associations from around the world. Check the list with current partners here: https://isbsg.org/meet-isbsg-partners/

ISBSG Data

The ISBSG repositories can be obtained for a modest fee and provide a wealth of data: https://www.isbsg.org/software-project-data/

The ISBSG data is unique, as it is provided in Excel format, which makes it easy to analyse through the statistical tools in Excel. Although many aspects of the data have been analysed by software metrics experts in the industry and researchers worldwide, most of the time the data is used to analyse productivity figures of certain types of projects for instance. ISBSG data is used for essential software processes, like software cost estimation, supplier) performance measurement, contracting, agile team estimation, benchmarking. In this document, the view of ISBSG on productivity measurement is explained as well as the way the data can be used in decision making.

Notes

Please note that Productivity is universally defined as Output/Input. ISBSG uses the Project Delivery Rate (PDR) to express productivity, which is the inverse: Input/Output. Using the universal definition would result in figures that are very low, e.g. 0,0232 FP/hour, which is hard for human brains to process, therefore the PDR is used, resulting in figures like 7,8 hours/FP. The PDR is the inverse of Productivity but is in practice used for exactly the same types of analysis, and therefore PDR and Productivity are often used intertwined.

Table of Contents

Introduction 1

Productivity Measurement – an example 3

The Software Industry and Productivity Measurement 3

Productivity: measuring the input 4

Productivity: measuring the output 7

An example of using the ISBSG data 9

Appendix A: About the ISBSG 11

Productivity Measurement – an example

In industries other than the software industry, productivity measurement is a normal activity that drives the success of a company.

Example

For example, consider a one-man painting company. For a painter, it would be logical to measure his productivity in ***effort hours per square meter***. He probably wants to differentiate the measurement into some categories, like tools used (e.g. roller / brush / spray/ etc.) and paint object characteristics (e.g. wall/ stairs/ door/ etc.).

When the painter builds up a database with productivity figures, he can easily quote for new painting jobs, simply by measuring the paint surface in square meters, multiplying with the proper productivity rate and multiplying the result with the hourly rate he asks for.

If there is a (international) database available with the productivity rates of paint jobs performed by other companies in the industry, the painter can compare how he performs, on average, against the industry. In case he is not best-in-class already, he understands that to win new paint jobs he has to keep improving his productivity. This is because lowering his hourly rate is usually not a very good idea.

He can do all this, but only when:

- He uses a ***standard measurement unit***, e.g. square meter. Only by using standards, productivity rates can be compared (benchmarked) and used for estimating;
- He uses a ***standard way to record the effort hours***. For example, is the lunch break included or excluded? Is the time spent discussing customer requirements included/excluded?
- He uses ***meaningful categories*** that differentiate productivity. For a painter, it may not matter too much if the paint object (let's say a wall) is in a villa or in a fisherman's house. The type of house may not be a meaningful category. However, the tool he uses is probably a main productivity driver.

Now, let's take a look at the software industry.

The Software Industry and Productivity Measurement

The Need to Improve

Unfortunately, the IT (and software) industry is still quite immature when it comes to using standards and when it comes to productivity (performance) measurement, benchmarking and continuous improvement.

Even the large international system integrators often don't know their productivity, cost efficiency, quality and delivery speed, as they don't collect data based on standards.

Organizations have no idea how they perform against their competition. It's essential for organizations to understand their capabilities against the industry. Sometimes delivering

functionality faster and better than the competition is even the basis for organizational survival.

The industry got away with that for a long time, because:

- It is difficult to measure ***output*** (software is not a physical thing, can't be touched and measured with conventional measurement instruments).
- Software projects are much more like an R&D project than manufacturing a product. R&D is incredibly hard to measure. It is relatively easy to measure the inputs, but the outputs are hard to measure and unpredictable by nature.

Looking Ahead

Now, slowly the industry is becoming more and more transparent. Customer organizations ask potential suppliers more and more to quantify their performance based on historical data. This way, it becomes possible to select the best supplier for the job. Please note that the best choice is usually **not** the least expensive choice (often resulting in project failures…or even disasters). The best offer is the most realistic one.

Especially in the agile era, organizations are looking for ways to understand the performance of the teams in order to understand the high performers and the low performers.

Senior management in organizations that 'went agile' struggle with the grip they have on the teams. As Agile teams usually don't use standardized metrics, and use story point metrics instead, it's impossible to compare these metrics between the teams. Therefore the call for standardized productivity measurement processes from senior management is getting louder nowadays.

ISBSG data helps organizations that measure productivity, based on standards, to understand their performance against the industry. Teams that appear to be low performers may, in practice, perform well when compared to the proper peer group.

Productivity: measuring the input

Meaningful Metrics

While it may seem easy to implement a Productivity Measurement process in an organization, reality shows that it is more difficult than one may think.

Just like the painter, it is sufficient to ***measure inputs*** (usually effort hours) ***and outputs*** (Units of Measurement, UoM) per software project, while ***using meaningful categories to differentiate the projects***, like technology (Java/.Net/Oracle/Etc.), project type (new development/enhancement) and/or implementation (Package implementation/modification/custom made software).

Important Decisions

To be able to build up meaningful and comparable productivity metrics, it is critical that **(international) standards** are used.

Some decisions that have to be made:

- Effort hours in/out scope of measurement, for instance

- Technical design, coding, unit test, systems test, other supplier tests, overhead in scope;
- Functional design, support acceptance test, implementation activities out of scope.
- Overtime in/out scope of measurement;
- Travel hours, meeting hours, overhead hours in/out scope;
- In case of packages, Portals/CMS or other configurable software, it may be necessary to have separate effort registration activities for customization, setting parameters and custom made software not part of the package.

Effort Registration

To be able to analyze the productivity of a supplier, department or team, the effort registration system should be implemented in a standard way.

If the choice is made that functional design hours are out of scope, all projects should register their effort of functional design separately from the other effort hours. It is strongly recommended to draw up a standard 'Work Breakdown Structure (WBS)' per project type and implement this WBS in the effort registration system. Everybody who registers effort hours should be aware of the importance of booking their effort correctly in the system.

ISBSG data

ISBSG Data is collected about the people whose time is included in the work effort data reported. Four levels are identified in the ISBSG data repository.

1. development team effort (e.g. project team, project management, project administration)

2. development team support (e.g. database administration, data administration, quality assurance, data security, standards support, audit & control, technical support)

3. computer operations involvement (e.g. software support, hardware support, information centre support, computer operators, network administration)

4. end users or clients (e.g., user liaisons, user training time, application users and/or clients)

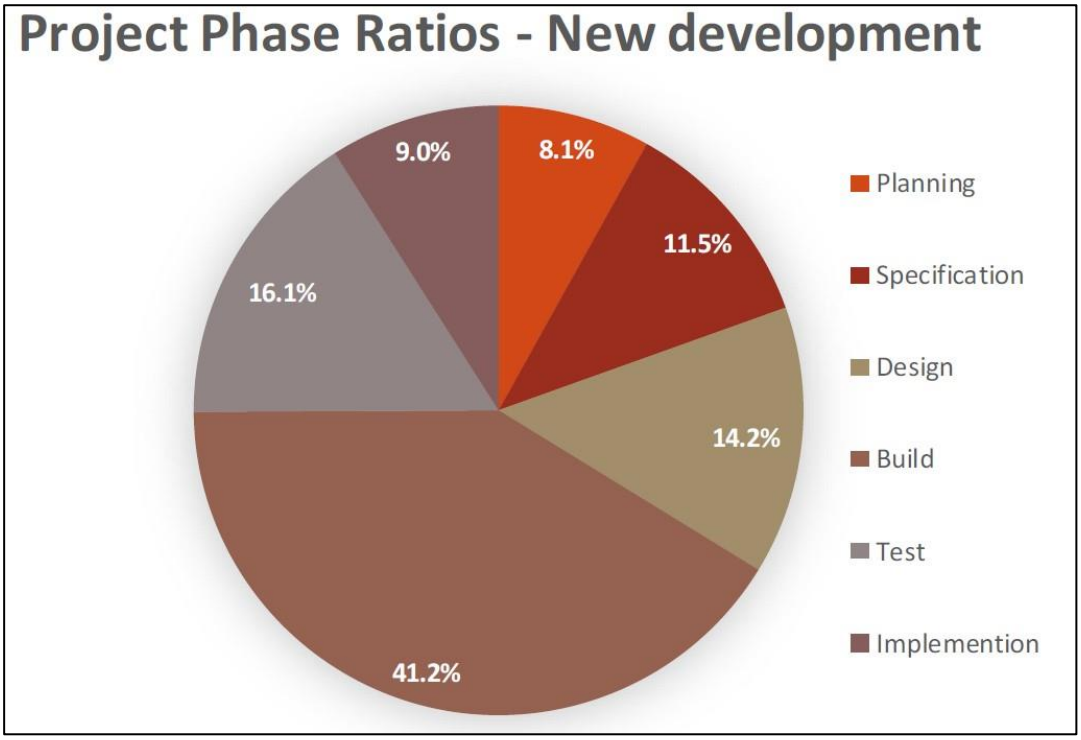
In practice, usually only level 1 effort is used to analyze the productivity of software development teams. To use the data in all the different lifecycle methodologies in the industry, ISBSG uses 6 meta phases to which all activities can be mapped. These activities are:

- Planning
- Specification
- Design
- Build
- Test
- Implementation

When comparing productivity rates, it is essential to understand the activities that are included in the PDR. In the Development & Enhancement repository, the Project Activity Scope field is used to show this.

To compare apples with apples, ISBSG normalizes project data that has not submitted data for all the phases. Therefore the **Normalised Work Effort Level 1** includes the data for all the phases, including the missing ones. The ISBSG repository manager normalizes the data using the percentages of data for which all the phases is available.

In the Special Analysis Report 'Project Planning - Insight into project phase ratios for planning validation and management (2017)' the most recent percentages are given. An example: new development projects is given in the next figure:



So, even if your organization is not carrying out all the activities, it is possible to normalize the effort data to cover the full lifecycle using the ISBSG percentages. This way, the normalized level 1 effort can be compared to the ISBSG normalized level 1 effort.

An Agile Example

Consider the following scenario:

- An agile team is carrying out the activities Design, Build and Test.
- They have recorded 1000 hours for their project (a number of sprints).
- Use the percentages above (just for this example, there is a separate pie chart for agile development available). This is 14.2% + 41.2% + 16.1% = 71.5% of the full lifecycle.
- If they want to normalize the effort to the full lifecycle, they need to multiply their effort with 100 / 71.5, which results in 1.399 effort hours.
- The Normalized Work Effort for this team is 1.399 effort hours (level 1).
- If they have measured the output of the project in function points, for example 300 FP, their Normalized Level 1 PDR = 1399/300 = 4.7 hours per FP.

Productivity: measuring the output

ISO/IEC Functional Sizing Methods

ISBSG strongly recommends using ISO/IEC standards for functional size measurement in order to be able to use the data and compare between projects. However, these methods are often regarded as old fashioned: 'Something we did 30 years ago'.

However, software development is still about developing functionality. Like a brick wall of 100 square meters has the same surface as a glass wall of 100 square meter, a 100 FP Cobol system offers the same amount of functionality to its users as a 100 FP Java system. Functional Size is independent of technical way of implementation and independent of the development method (traditional, agile, etc.).

Therefore is a ***highly recommended practice*** to use an ISO/IEC standard for functional size measurement in Productivity Measurement of software projects. There are five functional size measurement methods that comply to the ISO/IEC standard:

- COSMIC function points (ISO/IEC 19761);
- IFPUG function points (ISO/IEC 20926);
- Mark II function points (ISO/IEC 20968);
- NESMA function points (ISO/IEC 24570);
- FiSMA function points (ISO/IEC 29881).

Advantages of using one of these functional size measurement methods for productivity measurement are:

- Objective, repeatable, verifiable, defensible way to determine the size of the software.
- A clear relation between functional size and effort needed to realize the application. This has been studied and verified many times.
- The measure is clear for both customer organizations and supplier organizations. More functionality means more value, more effort needed and a higher price;
- Functional size is independent of the technical solution and/or the non-functional requirements. An application of 500 NESMA function points realized in Java is just as big as a Wordpress website of 500 FP. This enables comparison and benchmarking over technical domains and the use of historical project data (when properly classified) in estimating new software projects.

Other Functional Sizing Methods

Although ISBSG does capture the data of non-standardized methods, these are difficult to use in practice, just because of this lack of standardization. Examples of Other size measures that are often used in the industry, but are ***not recommended*** to use in productivity measurement:

- ***Story points (SP)*** in agile projects: A very subjective measure that only has value within one team. Comparison to other teams, departments and organizations is not possible. Please note that SP are useful to plan sprints and to track velocity for one team, but for productivity measurement SP are close to useless.

- ***Usecase Points (UCP)***: Only applicable when the documentation consists of usecases. UCP is also a highly subjective method, especially when it comes to

establishing the Technical Complexity Factor and the Environmental Factor. Also, there is no standard way to write usecases.

- ***Complexity Points***: Subjective and not standardized method to measure the complexity of an application.
- ***IBRA Points***: Not standardized method to measure the business rules in an application. When applied according to the manual, the result is zero for all applications.
- ***Fast Function Points (FFPA)*** (by Gartner): A measurement method deployed by Gartner that cannot be compared to the ISO standardized function point analysis methods. FFPA is perceived to be a commercial method that lacks a theoretical base and is partly subjective. The method has not proved to be faster than the Nesma estimated method and has not proved to be more accurate. Unfortunately it is often pushed on higher management level without the support of the specialists who have to work with it.

An example of using the ISBSG data

In this chapter an example is demonstrated of how to use the ISBSG data to measure the productivity of a completed software development project and to benchmark this productivity against the ISBSG Development & Enhancements repository.

The project

This new Development project was carried out using the traditional waterfall approach by an internal team of a bank. It is a business application. The technology was Java and the effort spent:

- Design: 1110 hours
- Build: 4210 hours
- Test 1530 hours
- Implementation: 230 hours

The total number of effort hours is 7080 hours.

During the project, a few RFC's were implemented which have been taken into account in the project size. The project size measured is 850 FP (Nesma or IFPUG)

The Analysis

First, the effort data is normalized to cover the full lifecycle as described. In this case, the team did not carry out Plan and Specify activities. Using the percentages given before, this means that 81.4% of the full lifecycle effort was spent. This means the normalize level 1 effort is 7080 / 81.4 *100 = 8698 hours.

Calculating the Project Delivery Rate: 8698 / 850 = 10.2 hours per FP.

The next step is to select the right peer group in the ISBSG repostory. For this analysis, the following selection parameters have been used:

- Data Quality Rating: A or B. This means only high quality data is used.
- Industry Sector: Banking or Financial
- Application Group: Business application
- Project type = New Development
- Primary Programming Language: Java.
- Relative Size: M2 (300 FP – 1000 FP) to compare with projects of similar size
- Count approach: Nesma or IFPUG 4+.

For data analysis purposes it is important to realize that the Nesma method and the IFPUG method produce similar results, as the counting guidelines are roughly the same. It is therefore possible to select Nesma and IFPUG 4+ data when analysing.

The Result

This selection of the ISBSG Development & Enhancement repository (2019: In total over 9000 data points) results in the following data set regarding the Normalised Level 1 PDR (ufp).

Statistic	Value
N	14
Min	1.1
P10	2.2
P25	2.6
Median	3.6
P75	6.2
P90	13.6
Max	24.7
Average	6.3

In total 14 projects are selected. The minimum value is 1.1 and the maximum value us 24.7. The Median (percentile 50% = P50) is half of the average, indicating a non-normal distribution. Possible outliers can be identified and excluded, however when working with the Median, the effect of outliers is usually much smaller compared to working with the average. In this example possible outlier analysis and removal has not been carried out.

Based on this analysis the project PDR of 10.2 hours/FP falls between the Percentile 75% and the Percentile 90%. This project was carried out against a higher PDR (equals lower productivity) than market average.

Appendix A: About the 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. 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 help YOU and your organization improve the estimation, planning, control and management of your 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, you receive 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/

Partners

This page will help you to find an ISBSG partner in your country http://isbsg.org/meet-isbsg-partners/