ISBSG
Delivering IT Confidence

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

Short paper

Analysis of Productivity of Projects Developed in Low Code Development Technology

Published by the International Software Benchmarking Standards Group

2021

Introduction

As the ISBSG repository contains more data for projects carried out in an agile way of working, analysis of differences between traditional projects and agile projects becomes more significant. 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. Typical key metrics based on function points are:

- Project Delivery Rate (PDR)[1]: Hours spent per function point
- Cost efficiency: Cost (or Price) per function point
- Quality: Defects per function point (in test and/or 1st month of production)
- Speed: Function points delivered per calendar month.

The ISBSG repository 'New Developments & Enhancements' contains thousands of completed projects for which these metrics are calculated. This enables organizations to use this industry data for fact-based understanding and decision making. In this short paper, the difference in productivity between traditional and agile projects is analyzed.

Low Code

In the last decade, software development became one of the key drivers of organizational success. To deliver functionality to the user, earlier than the competition, became an important business driver. Therefore, many organizations struggled as their application portfolios consisted of many legacy applications, for which it is difficult and expensive to implement changes. Also, it became harder to hire qualified resources for the projects.

Low-code technologies are one of the industry answers to these challenges. These platforms promise high productivity, and fast delivery, while these changes can be made by people with less technical backgrounds (e.g. citizen developers). By using dragging, dropping and visual modelling instead of coding, organizations can implement new functionality fast and with relatively low cost.

Low-code software is changing how enterprise applications are created and who is creating them. In this short paper, a high-level analysis is given of the differences in productivity between traditional languages, such as Java or .Net and low code technologies.

[1] The PDR is actually the inverse of the universal concept of Productivity (output/input) as it is easier to process for human minds, which usually struggles with metrics with many decimals.

Productivity in traditional projects

In this short paper, only the difference in productivity between Low-code in traditional projects and releases is shown.

For the traditional project dataset, the following selection criteria were applied:

- Primary Programming Language: .Net or Java
- Count approach: Nesma or IFPUG 4+

This results in 1921 data points. Figure 1 shows the PDR distribution of this data set.

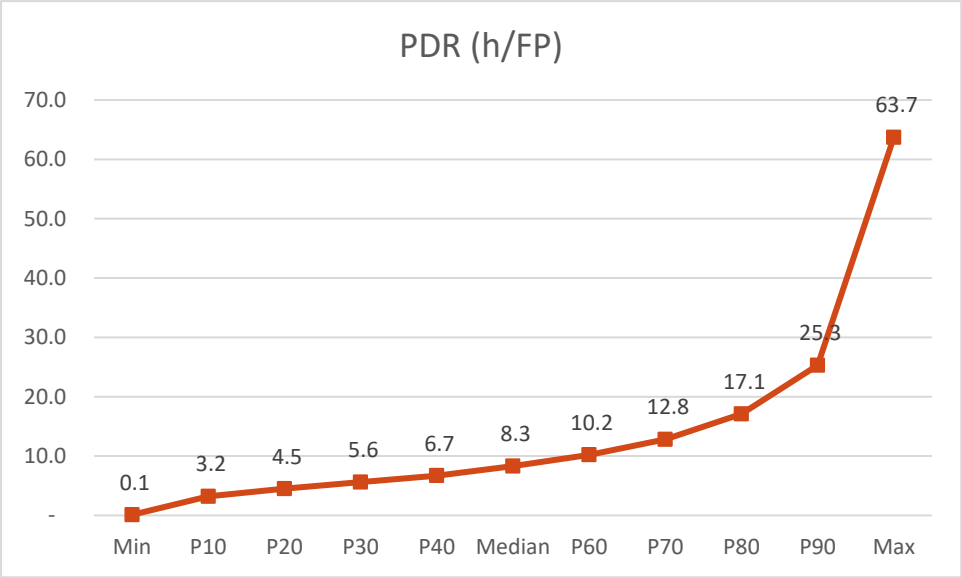

Figure 1: The distribution of Project Delivery Rate (PDR) of traditional languages in the ISBSG 2021 D&E repository.

The median Project Delivery Rate (PDR, hours per function point) of traditional languages is 8.3 hours per function point. Only 10% of the projects delivered functionality more productive than 3.2 hours per function point. Now let's take a look at the low-code dataset.

Productivity in low-code projects

Low-code platforms promise the very productive delivery of functionality, which can be an important business driver nowadays. In many organizations, the business case for low-code is easily positive.

In the 2021 D&E release of ISBSG, there are 57 projects that were completed in a low-code technology: Mendix and Outsystems. Figure2 shows the the distribution of the PDR of this data set.

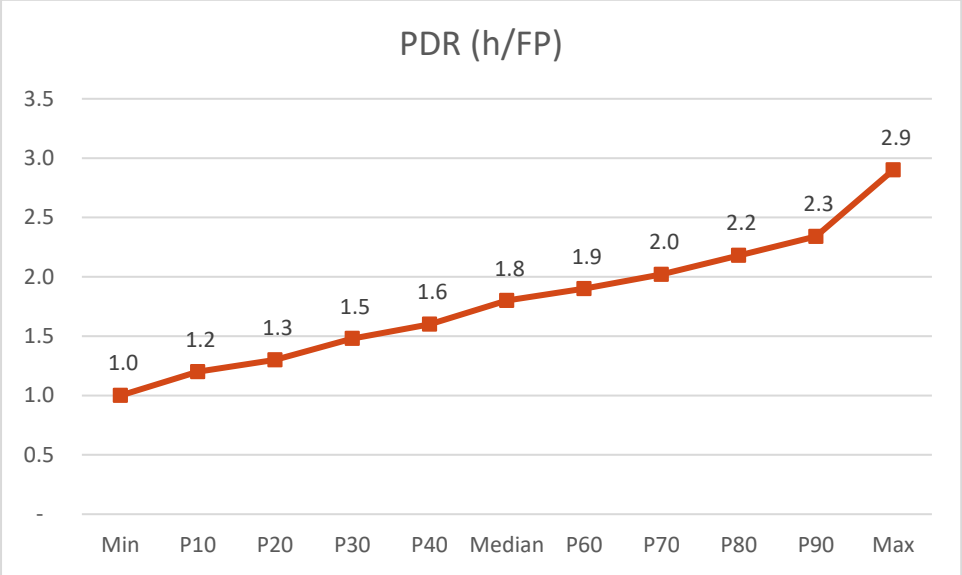

Figure 2: The distribution of Project Delivery Rate (PDR) of low-code languages in the ISBSG 2021 D&E repository.

While for the traditional languages, the 10th percentile was 3.2 hours per function point, all of the 57 low-code projects were completed more productively. The median is 1.8 hours per function point which means there is a factor 4.5 in average productivity.

Simple example

A simple example: to deliver a project of 100 function points, in allow-code technology one would spent about 180 effort hours, at maybe 100 USD per hour, equals 18000 USD for the project. Instead realizing the same project in .Net or Java would cost around 830 effort hours. Using the same hour rate, this would mean 83000 USD for the project. Of course, the low-code project would also be completed faster, resulting in benefits for the business or users. Also, maybe these hours could be spent by people with a lower hourly rate, which would make the difference in cost even higher.

Low-code seems to be changing the industry and every organization should switch to one of these platforms as soon as possible. The cost of application development, however, is only a part of the Total Cost of Ownership of an application. Licenses, cloud consumption and maintenance & support costs are other cost drivers that need to be taken into account.

The license model and the cost drivers in this model play an important role. If the license cost, for instance, depends on the number of end-users of the application, there could be an issue when you develop a free app, put it in the app stores, only to find out it becomes very popular worldwide and your license costs go through the roof. When making technology choices, it remains important to carefully understand the differences in license models, maintenance costs, application development speed and productivity, as well as quality and risk.

Conclusion

Low-code platforms promise highly productive software development and the analysis in this short paper supports this. There are other factors to take into account before deciding to use a low-code platform, such as the license model and the cost drivers that impact the license prices. When purely looking at productivity, however, there is a large difference between the visual drag-and-dropping in low-code platforms and the manual coding of Java or .Net code.

If you wish to do your own analysis, or if you are interested to use the ISBSG data for Cost estimation, benchmarking, performance measurement, procurement, etc., please subscribe to the data here: https://www.isbsg.org/project-data/

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:

http://isbsg.org/meet-isbsg-partners/