

The Impact of Effectively Managing Requirements for Agile Projects



Introduction

As the ISBSG repository contains more data of projects carried out in an agile way of working, analysis of differences between traditional projects and agile projects becomes more significant.

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 'New Developments & Enhancements' repository 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.

Agile methodologies emphasize flexibility and continuous adaptation, but clear requirements are still crucial for efficient development.

The effective management of project requirements can:

- **contribute to a team's ability to deliver working software in short iterations.**
- **minimize rework caused by misunderstandings.**
- **ensure the final product aligns with stakeholder expectations.**

This short paper investigates the impact of the effort spent on the design phase and its impact on productivity, in agile software development projects.

¹ The PDR is 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.

The Agile Requirements Management Paradox

The Agile Requirements Management Paradox lies in the inherent tension between two core aspects of Agile development:

1. **Embracing Change and Flexibility:** Agile thrives on adapting to evolving needs and readily accepting changes in requirements. This allows teams to respond to new insights and user feedback throughout the development process.
2. **Need for Clarity and Focus:** Despite the embrace of change, some level of requirement clarity is necessary for efficient development. Without a baseline understanding of what needs to be built, teams can waste effort on features that ultimately don't meet user needs.

The paradox arises because **too much upfront documentation** can stifle the flexibility that's core to Agile. However, a **complete lack of requirements management** can lead to confusion, rework, and missed deadlines.

The key to resolving this paradox lies in finding the right balance. Agile teams employ **lightweight requirements management** techniques that focus on capturing the essence of a feature while allowing room for future adjustments. This might involve user stories, acceptance criteria, and collaborative tools that keep requirements visible and adaptable.

Now that many organizations have moved to an agile way of application development, the financial impact can be better understood. Agile delivery is faster: in theory every sprint (of 2 to 4 weeks) brings a new release of functionality to the users. Also in theory, we don't need to estimate project costs any longer, as we have a money budget bucket, which consists of effort hours and (internal or external) effort costs, which is very predictable. However, the Project Delivery Rate (PDR) determines how much value (functional size) is delivered per sprint or in this budget box. The PDR therefore remains a very important metric to use in project estimation, monitoring, benchmarking, and outsourcing activities.

Analysis of the impact of design effort on the PDR

The ISBSG collects Developments and Enhancements data and categorizes the effort in 6 activity classes: Plan, Specify, Design, Development, Test, and Implementation. These activities need to be performed in any new development project, functional release, and/or sprint(s) of agile application development teams.

For agile application development, we assume that the planning and specification is not done by the team. Usually, there is a product owner that carries out these activities.

For this short paper, we therefore look into the Design activity type. As described by the agile paradox, we expect that too little and too much requirements management result in higher PDR's (effort hours spent per function point). But how much is too much?

To answer this question, data from the ISBSG Repository is analysed. The following selection criteria is applied to the ISBSG data:

- Data Quality Rating = A or B
- Count Approach is Nesma or IFPUG 4+
- Development Methodologies = Agile Development
- Project year > 2020

When the criteria above is applied to the ISBSG Data Repository, a dataset of 317 data points is obtained. Figure 1 displays the distribution of the Project Delivery Rate for this dataset.

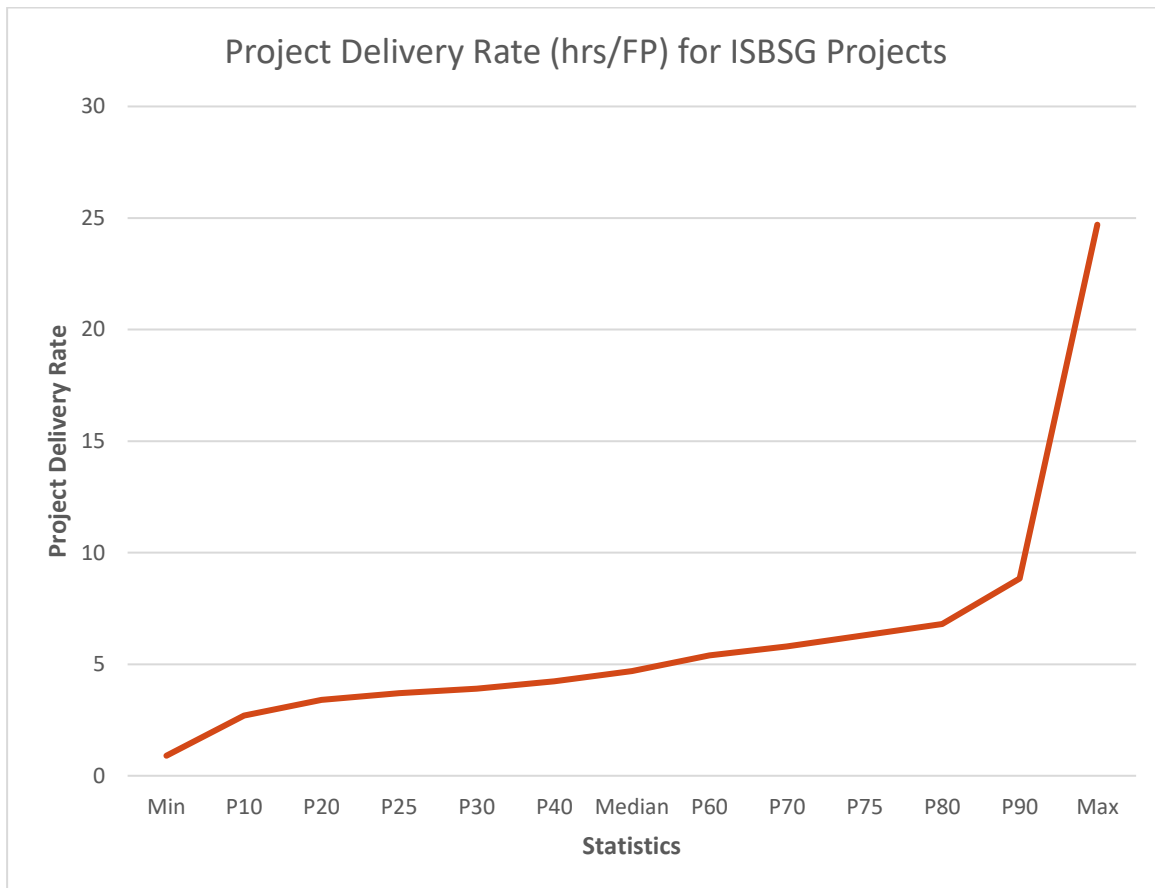


Figure 1: PDR of ISBSG Projects from chosen dataset in this analysis

To enable further analysis, the 317 data points, shown in Figure 1, are grouped according to the percentage of time spent on design (i.e. %Design), as follows:

- Very Low: 0%-15%
- Low: 15% - 20%
- Medium: 20% - 25%
- High: >25%

In Figure 2, the PDR per %Design class is shown.

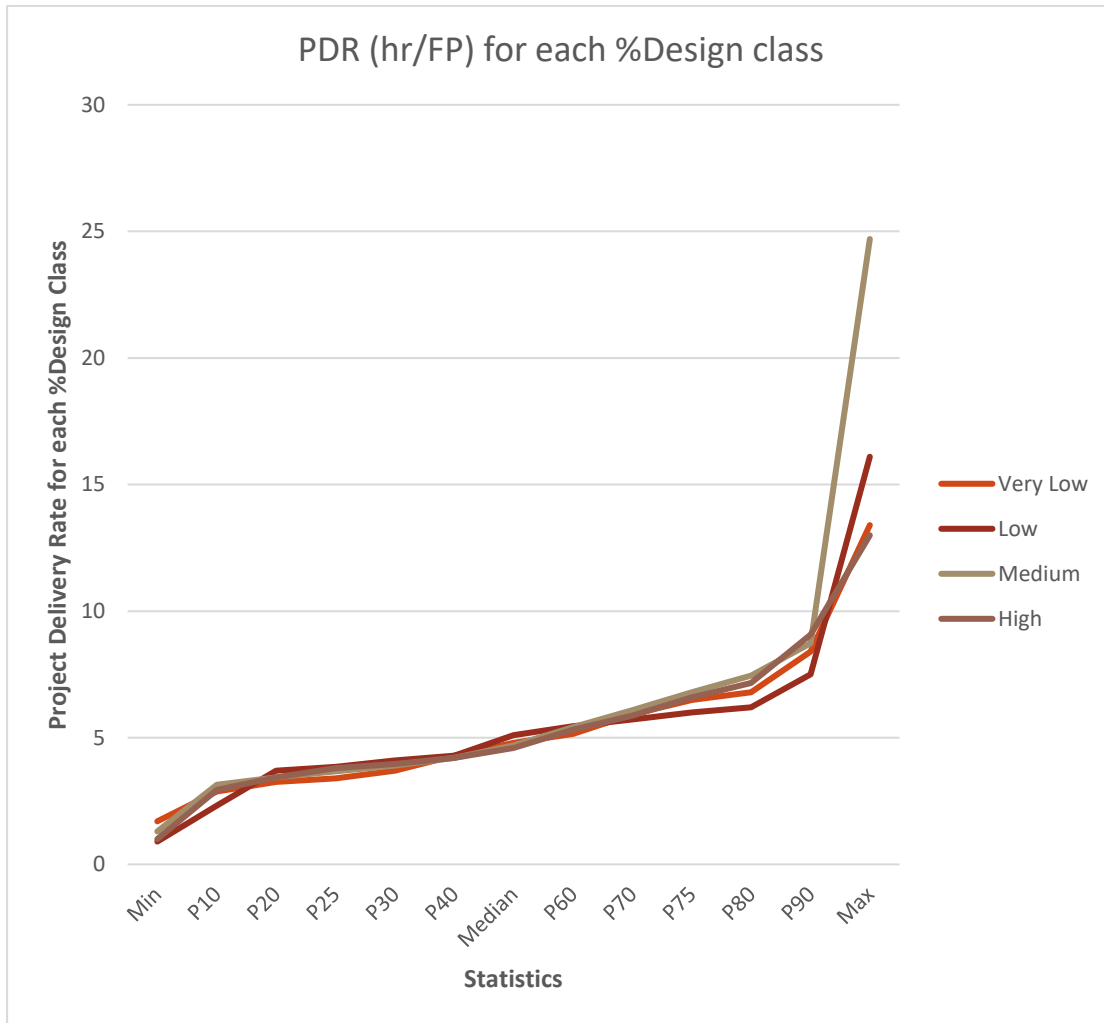


Figure 2: PDR of the dataset in this analysis per %Design class.

Although there are some small differences in the descriptive statistics, there does not seem to be a very big difference in the PDR per %Design class. Unexpectedly, however, the medium category shows a much higher P90 and max PDR than the other categories.

From this analysis and this dataset, the theory stated above is not supported. However, a lot of important factors have been left out of this analysis, e.g., size, primary programming language, team size, etc. Therefore, it is not possible to draw any conclusions from this analysis.

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. Their aim is 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. It 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 <https://www.isbsg.org/data-subscription-2/>

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/>