ISBSG
Delivering IT Confidence

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

Short paper

Effort Distribution in Agile Teams

Published by the International Software Benchmarking Standards Group

2024

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.

The ISBSG collects industry data, where output is measured using ISO/IEC standardized and therefore objective, repeatable, auditable methods. These methods include: Nesma, IFPUG and COSMIC. 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 Development & Enhancement' repository has 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 (i.e., effort distribution) between traditional and agile projects is analyzed.

[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.

Optimal Effort in Agile Development Teams

Understanding the optimal effort distribution in agile development teams is crucial for several reasons:

Increased Efficiency and Productivity:

- **Balanced workload:** By allocating tasks based on individual strengths and capacities, you prevent overburdening specific team members and ensure everyone contributes effectively. This leads to a smoother workflow and faster completion of tasks within each sprint.

- **Reduced bottlenecks:** Identifying areas where effort is concentrated helps you anticipate potential bottlenecks and proactively address them by redistributing tasks or seeking additional resources.

- **Improved focus:** When team members understand their specific responsibilities and the effort expected, they can focus their energy on completing those tasks efficiently, reducing distractions and wasted effort.

Enhanced Team Morale and Collaboration:

- **Fairness and ownership:** A balanced distribution fosters a sense of fairness and ownership among team members, leading to increased motivation and engagement. Everyone feels valued and contributes meaningfully to the project's success.

- **Improved communication and collaboration:** When team members understand each other's workload and capacity, they can communicate more effectively, collaborate seamlessly, and offer support when needed, fostering a stronger team spirit.

- **Reduced burnout:** By preventing individuals from being overloaded, you minimize the risk of burnout and ensure team members maintain their energy and enthusiasm throughout the project.

Meeting Project Goals and Deliverables:

- **Predictable outcomes:** Understanding the effort required for each task allows for more accurate estimation and better project planning. This leads to increased predictability in meeting deadlines and delivering commitments.

- **Improved quality:** When team members are not overloaded and can focus on their assigned tasks, they are more likely to produce higher-quality work and deliver results that meet the project's standards.

- **Increased adaptability:** Agile methodologies emphasize flexibility and they respond to change. Understanding effort distribution allows the team to adapt quickly to changing priorities or unexpected challenges by adjusting task assignments and workload as needed.

While there's no one-size-fits-all "optimal" effort distribution, understanding individual strengths, task complexities, and project goals allows agile teams to strive for a balanced allocation that maximizes efficiency, promotes collaboration, and ultimately leads to successful project outcomes.

Effort captured in the ISBSG Development & Enhancement Data (D&E) Repository

ISBSG collects data from the IT industry using data collection forms, which can be found here: https://www.isbsg.org/submit-data/.

Anyone who sends in a data collection form to ISBSG receives a free benchmark report in return. This shows the performance of their project, release or sprint against industry averages performance.

If you are not already doing so, please help ISBSG to collect more data!

Understanding team performance is a very powerful tool to improve the competitiveness of companies. High performing teams deliver more value, faster and with better quality, leading to increased company profits.

Effort is captured by activity type:

1. Plan: This category encompasses activities related to overall project planning and management.

Examples:
- Defining project scope and objectives
- Creating project schedule and budget
- Identifying risks and mitigation strategies
- Establishing communication plans and stakeholder management

2. Requirements: This category focuses on gathering, analyzing and documenting user needs and functional specifications.

Examples:
- Conducting user interviews and workshops
- Analyzing user stories and requirements documents
- Defining acceptance criteria
- Creating system use cases and data flow diagrams

3. Design: This category involves translating requirements into detailed technical blueprints for the application.

Examples:
- Designing system architecture and user interface mockups
- Defining database schema and data models
- Developing technical specifications and algorithms

4. Development: This category encompasses the actual coding and implementation of the application functionalities.

Examples:

- Writing code in a specific programming language
- Unit testing and integration testing of individual components
- Building and deploying the application to different environments

5. Test: This category involves testing the application for functionality, performance, and security vulnerabilities.

Examples:

- Conducting system testing, integration testing, and user acceptance testing
- Identifying and fixing bugs and defects
- Performance testing and optimization

6. Implementation: This category covers activities related to deploying the application to production and providing ongoing support.

Examples:

- Installing and configuring the application in the production environment
- User training and documentation
- Providing ongoing maintenance and bug fixes

It's important to note that these are just general examples, and the specific activities within each category may vary depending on the project's complexity, methodology used, and the specific ISBSG definitions.

Effort in Agile Development Teams

While there's no single "typical" effort distribution in agile development teams, some general trends emerge:

Focus on Value Delivery:

- **Development and Testing:** A significant portion of effort, often exceeding 50%, is typically dedicated to development and testing activities. This ensures features are built and thoroughly tested within each sprint, meeting user needs and delivering value incrementally.

Collaboration and Communication:

- **Planning and Requirements:** A considerable amount of effort, often around 20-30%, goes towards planning. This includes backlog refinement, user story estimation, and sprint planning. Additionally, effort is invested in requirements gathering and communication with stakeholders to ensure clarity and alignment.

Design and Implementation:

- **Design and Implement:** Depending on the project's complexity and chosen methodology, design and implementation activities might take up

10-20% of the effort. This includes designing user interfaces, system architecture, and deploying the application to production.

Factors Influencing Distribution:

It's important to remember that these are just general estimates, and the actual effort distribution can vary significantly depending on several factors:

- **Project complexity:** More complex projects might require a higher proportion of effort dedicated to planning, design, and testing to ensure quality and manage risks.
- **Team composition and skills:** The team's skillset and experience can influence effort allocation. Teams with strong design expertise might spend more time on upfront design, while others might prioritize rapid development and testing cycles.
- **Agile methodology used:** Different methodologies within the agile umbrella might emphasize different activities. For example, Kanban might focus on continuous flow and minimize upfront planning, while Scrum might dedicate more time to sprint planning and backlog refinement.
- **Project priorities and goals:** The project's specific goals and priorities can dictate effort allocation. If delivering a functional prototype quickly is crucial, more effort might be directed towards development and testing in initial sprints.

Key Takeaways:

- Value delivery through development and testing is central.
- Collaboration and communication are crucial for effective planning and requirement gathering.
- Design and implementation efforts vary based on project specifics.
- The optimal distribution is dynamic and adapts to project needs and team capabilities.

By understanding these factors and continuously monitoring effort distribution, agile teams can strive for an allocation that maximizes efficiency, fosters collaboration, and ultimately delivers successful project outcomes.

Match Agile effort to ISBSG activity types.

In the next table, the match of the typical agile team effort distribution[2] to the ISBSG activity types, is shown.

ISBSG Activity Type	Agile Effort Distribution	Explanation
Plan	Planning and Requirements (20-30%)	This category aligns with activities like backlog refinement, user story estimation, and sprint planning within the agile context.
Requirements	Planning and Requirements (20-30%)	This category encompasses activities like user interviews, gathering user stories, and defining acceptance criteria, which are crucial for agile requirements gathering and backlog management.
Design	Design and Implementation (10-20%)	While not always a separate stage in agile, design efforts are often integrated into development sprints. This includes designing user interfaces and system architecture.
Development	Development and Testing (50%+)	This forms the core of agile development, encompassing writing code, unit testing, and integration testing to deliver working functionalities within each sprint.
Test	Development and Testing (50%+)	Agile emphasizes continuous testing throughout the development lifecycle. This category includes system testing, user acceptance testing, and identifying and fixing defects.
Implementation	Design and Implementation (10-20%)	This category might be less prominent in pure agile methodologies but can encompass deployment activities and initial user training in some cases.

Table 1: Match Agile effort to ISBSG activity types

[2] Source: Google Gemini 25-02-2024.

Now, let's look at the effort distribution of data in the ISBSG D&E repository. Unfortunately, not all teams register their effort hours consistently and effort of certain phases is not always submitted.

For the analysis, the following data set from the ISBSG D&E Repository is selected:

Data Quality Rating = A or B.

Year of Project > 2020.

Count approach = Nesma or IFPUG 4+.

All 6-effort type: >0.

Unfortunately, this results in only 3 data points, and only 2 of these are for agile teams. The average effort distribution of these 2 data points is the following figure:

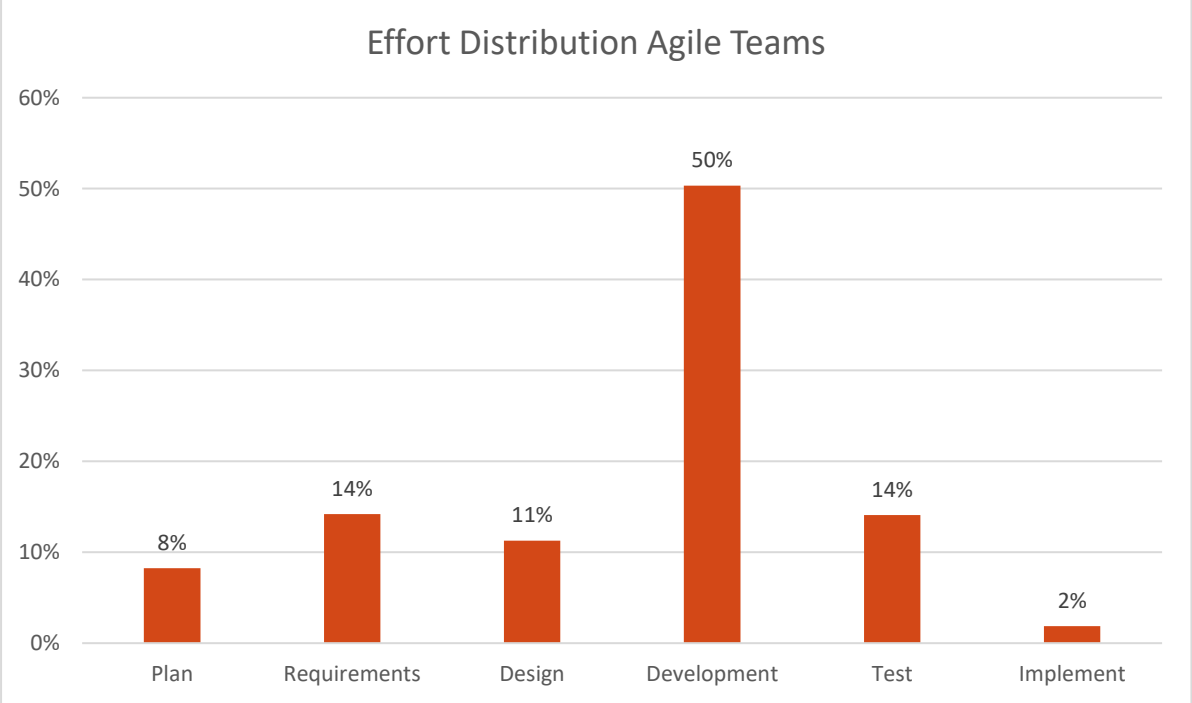

Figure 1: Conclusion on Effort Distribution Compared to Agile Norms:

The data from the ISBSG repository **partially aligns** with the typical effort distribution observed in agile development teams. Here's a breakdown of the observations:

Typical distribution	→	**Actual percentages**
Planning and Requirements (20-30%)	-	8+14 = 22%. In the typical range
Design and Implementation (10-20%)	-	11+2 = 13%. In the typical range
Development and Testing (50%+)	-	50+14 = 64%. In the typical range

Even though there are only 2 data points, the average distribution of these points are in the typical agile range.

Overall:

While the data deviates slightly from the "typical" agile distribution in some areas, it's important to consider the specific context of the project and the reasons behind these variations.

Here are some additional points to consider:

- **Project complexity:** More complex projects might require adjustments in effort distribution, potentially justifying a higher allocation for design or planning activities.
- **Methodology variations:** Different agile methodologies might emphasize specific activities differently, leading to variations in effort distribution.
- **Data accuracy:** The ISBSG data might not perfectly capture the nuances of effort allocation within the specific project context.

It's crucial to analyze the data within the context of the project and consider potential reasons for the observed deviations. If possible, gaining further insights into the project's specific methodology, complexity, and team dynamics could provide a more complete understanding of the effort distribution and its effectiveness in achieving project goals.

Analysis of Data Points with only Design, Development and Test effort

Most of the time, effort that is submitted is only of the following categories - Design, Development and Test. In these cases, the ISBSG repository manager needs to normalize the effort to incorporate the activities not submitted, for instance to calculate the Normalized Project Delivery Rate (hours per FP).

When we look at the dataset with the following difference:

Design, Development and Test effort are all > 0

Development Methodologies = “Agile development”

The number of datapoints is 317, with the following size distribution in Nesma or IFPUG function points. See Figure 2 for size statistics for the projects.

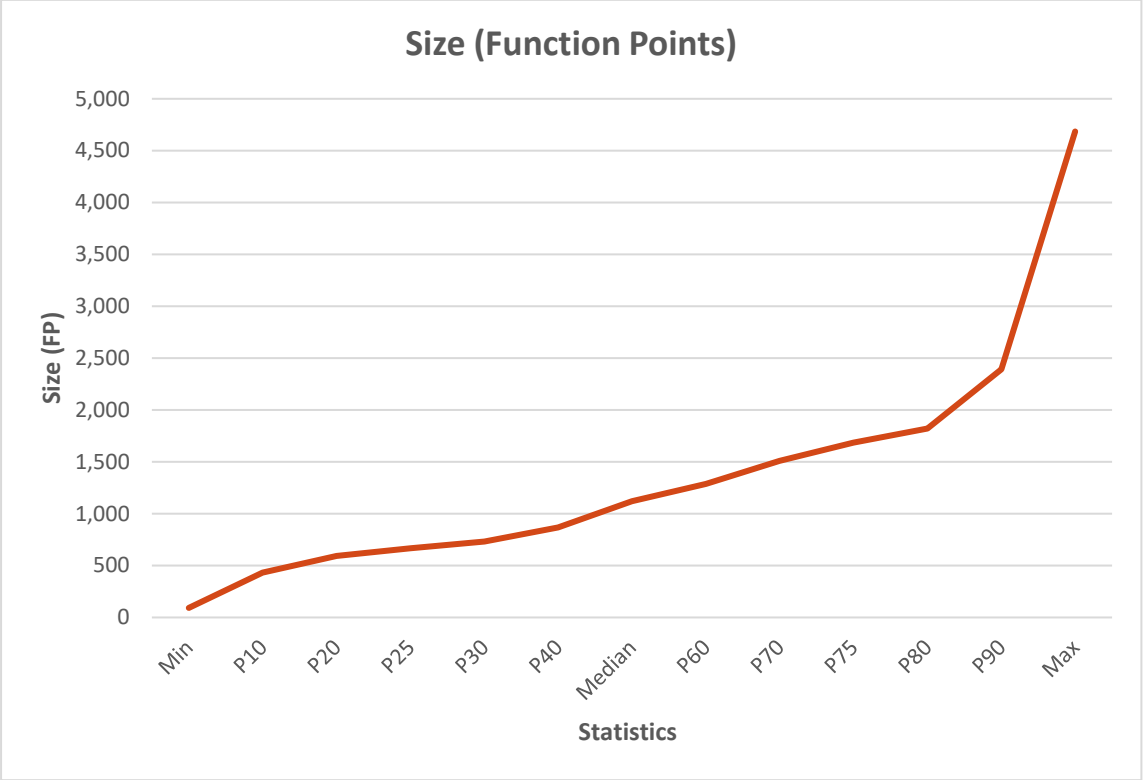

Figure 2: Size Statistics for Agile projects from the ISBSG repository

The median size of the 317 data points is approximately 1000 function points.

Figure 3 shows the effort distribution percentages of these data points.

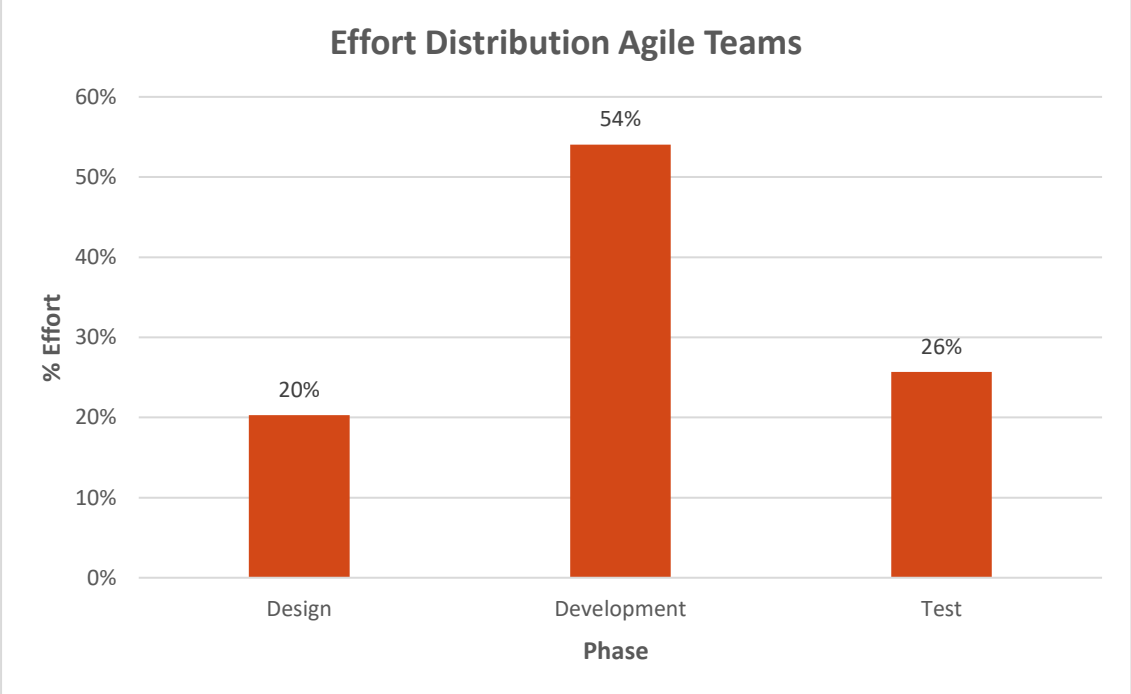

Figure 3: Effort distribution for ISBSG Agile projects for Design, Develop & Test Phase

When comparing the effort distribution shown in Figure 3 to the expected effort distributions in Table 1, the following observations can be made.

Alignments:

- **Development:** The **50%** effort allocated to development falls within the **expected range** for agile projects, highlighting the focus on delivering working functionalities in each sprint.

- **Test:** The **20%** effort dedicated to testing aligns with the emphasis on continuous testing throughout the development lifecycle in agile methodologies.

Deviations:

- **Design:** The **20%** effort for design remains **slightly higher** than the typical range of 10-20% observed in agile, which includes implementation. This could indicate a more upfront design approach or a complex project requiring more detailed design work.

Missing data:

- **Plan and Requirements:** As mentioned earlier, agile teams often exclude these phases from their reported data. However, their absence makes it difficult to assess the overall effort distribution and adherence to typical agile practices.

- **Implementation:** Like Plan and Requirements, the lack of data on implementation hinders a complete understanding of the project's effort allocation across the entire lifecycle.

Overall:

The data of the 317 data points still aligns quite well with the typical distribution observed in agile projects. The focus on development and testing reflects the core principles of delivering working software iteratively and incrementally. However, the missing data on other phases still limits a definitive conclusion about the project's overall adherence to agile practices.

Conclusion

The provided data on effort distribution in an agile development project offers **mixed insights** when compared to the typical breakdown observed in agile methodologies.

Key observations:

- **Development (50%)** aligns well with the expected emphasis on delivering functionalities within sprints.

- **Testing (20%)** aligns with the importance of continuous testing in agile.

- **Design (20%)** is slightly higher than the typical range, potentially indicating a more upfront design approach or project complexity.

- **Data is often missing** for **Plan, Requirements, and Implementation**, hindering a complete understanding of the overall effort distribution and adherence to agile practices. Often these activities are not carried out by the core agile development team: Plan and Requirements by the Product Owner, who is often a business representative and not considered part of the team. Implementation is often done by a separate operations team.

Therefore:

- While the data suggests a focus on core agile principles like development and testing, the lack of information on other phases limits a definitive conclusion.

- To gain a more comprehensive understanding, consider:
  - **Context:** Analyze the data within the specific project's context, including methodology, complexity, and team composition.
  - **Missing data:** Seek additional information on excluded phases for a holistic view.
  - **Comparison limitations:** Remember that the table provides a general guideline, and deviations can occur based on specific project circumstances.
  - **Use the data wisely:** When using the data, for instance for estimating or benchmarking, make sure you know if the Product Owner activities (Plan and Requirements) and/or the Implementation effort is in scope or out of scope for the estimate or the benchmark.

By considering these points and potentially seeking further context, you can gain a deeper understanding of the project's effort distribution and its effectiveness in achieving desired outcomes within the chosen agile framework.

If you wish to do your own analysis, or if you are interested in using the ISBSG data for cost estimation, benchmarking, performance measurement, procurement, etc., please subscribe to the ISBSG data by clicking on the link: 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 maintains industry data for software development projects and maintenance & support activities. This 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. ISBSG 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/