International Journal of Innovative Research and Scientific Studies, 8(1) 2025, pages: 2762-2778

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

The case of evaluating the teamwork for performing new tasks under conditions of constraints: An ecosystem approach to achieving project success

Alla Polyanska[1*], Taras Mykhailov[2], Iryna Stankovska[3]

[1,2,3]*Ivano-Frankivsk National Technical University of Oil and Gas, Ukraine.*

Corresponding author: Alla Polyanska (*Email: parvs@ukr.net*)

Abstract

The article is devoted to substantiating a methodology for assessing the future performance of a project team under constraints. The proposed solution is considered in the context of using an ecosystem approach to management, namely, strengthening collaboration and engaging project team members, which affects their ability to solve tasks efficiently. The proposed assessment methodology is based on Scrum principles and incorporates elements of traditional project management, specifically the critical path method, which enables effective resource allocation for project tasks. Additionally, it considers the constraints on project implementation within the Theory of Constraints (TOC) framework. The proposed assessment methodology has both theoretical and practical significance. Theoretically, it provides a comprehensive approach to solving project tasks. From a practical perspective, the methodology evaluates the potential for fulfilling orders under constraints. This is achieved by analyzing the historical experience of the project team to utilize the knowledge gained for improving the evaluation of customer orders. The results of the calculations based on the proposed methodology made it possible to identify the gap between planned and forecasted work volumes, highlight constraints that need to be addressed to bridge this gap, propose strategies for overcoming these challenges, and evaluate the effectiveness of the methodology's application.

Keywords: Constraints, Ecosystem approach in project activities, Evaluation methodology, Future work estimation, Knowledge, Project team, Scrum, Story point.

DOI: 10.53894/ijirss.v8i1.5055
Funding: This study received no specific financial support.
History: Received: 9 January 2025 / **Revised:** 18 February 2025 / **Accepted:** 25 February 2025 / **Published:** 28 February 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Problem Statement

The study of project performance is the subject of research by many scholars, driven by the relevance and widespread adoption of the project-based approach as a flexible management method, the diverse composition of project teams, and the unique challenges that are often encountered for the first time. Today, the effectiveness of project work is widely examined based on the Scrum framework (Riabchykov and Hanushchak-Yefimenko [1] and Cartaxo, et al. [2]), the Agile approach (Beck, et al. [3] and Hema, et al. [4]), and the use of the critical path method [5, 6]. Well-organized work through these management practices contributes to the successful completion of project tasks (Flores-Garcia, et al. [7]), adherence to deadlines and budgets (Bukłaha [8] risk reduction Tavares, et al. [9]), and effective team performance (Torres and Cornide-Reyes [10]), among other benefits.

However, there is no ideal method that would suit all teams. Knowledge gained from performing certain tasks [11, 12] and studying cases (IBM [13]), which in most cases demonstrate successful examples of solving applied tasks, is gaining more importance.

In this article, we also propose to examine the experience of a team that faced a client's request to complete a large volume of work within an extremely short period of time. In project management, meeting tight deadlines while maintaining high-quality results is a constant challenge. Organizations often struggle to balance client demands with the team's ability to complete tasks under strict time constraints. The urgency of the request posed a significant challenge, requiring immediate attention and strategic and operational planning. Teams are in a complex situation involving many tasks, deadlines, and other constraints. The urgency necessitates carefully assessing whether the request can be fulfilled and whether the team can effectively operate within the given limitations. The challenge lies in the workload and organizing its execution efficiently to ensure precise and effective task completion. In these conditions, the risks of overestimating or underestimating tasks pose a significant threat, potentially impacting project timelines, budget constraints, and overall client satisfaction.

Therefore, the topic of this article involves justifying methods for evaluating work in project teams under established constraints, using an ecosystem approach to project management.

2. Analysis of Research and Literature Review

Successful project management requires an understanding of constraints. Constraints are defined by the project management triangle: time, scope, and budget. Balancing these constraints while ensuring stakeholder satisfaction remains a constant focus of project activities carried out by the project manager or project management office. Constraints act as limiting factors that influence project management, programs, or project portfolios. Every project has a defined scope of work that must be completed within a set budget and timeframe while ensuring that the product meets the agreed-upon quality standards [14]. However, traditional project management methods, such as EVM/ES, do not account for the impact of multitasking and uncertainty related to task duration estimation [15]. The idea of introducing buffers to mitigate the effects of multitasking and uncertainty was presented within the framework of the CCPM concept [15]. CCPM focuses on action planning in project activities and proposes estimating task duration with a 50% probability of on-time completion. Additionally, CCPM suggests using buffers to accommodate uncertainty [15]. The CCPM concept is one of the directions in developing the Theory of Constraints (TOC) for projects.

The main objective is to increase productivity or raise the percentage of completed tasks within the organization. In an environment of change and project implementation uncertainty, achieving project work productivity is realized through applying Scrum. Scrum relies on clear planning, flexibility, motivation, communication, and transparency—characteristics that help eliminate constraints in project execution. The Scrum framework identifies key metrics and evaluates and analyzes them to enhance productivity. The challenges of developing metrics for assessing productivity and project performance using Scrum have been explored in studies [12-18]. Performance indicators in software development within Scrum-based companies have been analyzed by [16]. In their article [17], the authors argue that the most significant factors influencing work performance and the perceived importance of adhering to metrics are the duration and experience of an employee within the team, regardless of their specific roles.

In the study Riabchykov and Hanushchak-Yefimenko [1] the authors focused on one of the key metrics for measuring team productivity - *'volatility'* and its relationship with the iteration risk metric. The study states that in the application of Scrum, sprint volatility may be influenced by risks arising from the dynamic nature of the project environment. According to the authors, this approach allows for establishing a connection between sprint riskiness and fluctuations in team productivity.

Authors Lysenko and Fediai [18] propose using the Agile Earned Value Management (Agile EVM) method in Scrum projects as a cost management tool. They justify how applying the Agile EVM method to projects implemented within the Scrum framework enables the project manager and team to track their progress and make decisions to adjust their actions for cost monitoring [18].

Another important metric proposed for use in determining the productivity of an Agile team is velocity, which includes the volumes and timelines for completing work in each sprint [19-21].

The authors of the work Pham and Neumann [22] note that measuring productivity is of great importance when using agile methods and justify the appropriateness of using indicators such as transparency, standardization, and accuracy, alongside productivity metrics like story points and work completion charts, which are widely used in practice.

Productivity of teamwork can be ensured by considering the categories of "Balancing between stability and change," which forms the substantive theory of constant balancing between stability and change during agile work from the perspective of the employee [23]. At the core of this theory is the model of Corbin and Strauss [24] for constructing substantiated theories, presented in the form of a three-level pyramid, where the concept of the lower level is at the bottom, categories are on the second level, and the main category is at the top.

Currently, researchers are focusing on issues related to answering the question: "Why is the use of agile methods positive for some software developers and negative for others?" Therefore, the authors of the study Venkatesh, et al. [25] proposed linking the use of the agile method with the individual skills of the developer and two established determinants of employee burnout: role conflict and role ambiguity, which, in our opinion, significantly impact productivity.

Buvik and Tvedt [26] emphasize how trust directly or indirectly affects knowledge exchange within a team Buvik and Tvedt [26] which, in our opinion, influences the team's productivity. According to the results of their research, they argue that in the context of a project team, dedication to the project is more important for knowledge exchange than dedication to the team. Considering the complexity of work, knowledge and competencies significantly improves personnel performance [27]. In examining the role of knowledge in management, the authors note that knowledge creates value and drives the company's development [28]. Another factor affecting the team's productivity is its level of digital maturity Polyanska, et al. [29] a necessary and mandatory condition for an IT project team.

Thus, the proposed approaches to evaluating project work productivity allow for a focus on constraints and the choice of relevant ways to manage them. By applying TOC, systemic constraints for projects are mainly defined as resources. To use constraints, tasks on the critical path take priority over other types of activities. Projects are generally planned and managed so that resources are available when it's time to start performing critical path tasks. The Theory of Constraints, developed by E. Goldratt in the 1980sWilliam Dettmer [30] offers an approach to management based on seeing the system as one or more constraints that reduce its efficiency [31-34].

A constraint is anything that prevents a system from achieving its goals. After identifying the constraint, the theory focuses on overcoming it, which will expand the system's throughput and improve its performance. The methodological approach within the Theory of Constraints includes several stages. First, it is necessary to identify the systemic constraint. This step defines one or more factors that limit the system's productivity. Secondly, deciding how the system can function more effectively within the existing constraint is necessary. Thirdly, everything else must be subordinated to the management [35]. Three types of uncertainties in project planning and scheduling are highlighted: "uncertainty of activity time," "uncertainty of path time," and "resource uncertainty". Buffers are used and applied in Critical Chain Project Management (CCPM) Vanhoucke [36] and Steyn [37] to reduce these uncertainties. Despite significant knowledge in project management, history shows the inefficiency of projects when they fail to meet one or more of these expectations. Research suggests that project managers should identify external factors and/or internal management of the triple constraints, including trade-offs that may impact project management [38].

Thus, the increasing complexity of modern project tasks and the conditions under which they are executed draw more attention to the ecosystem approach in organizational development [39]. The ecosystem approach has become a vital methodology in addressing the challenges of implementing modern projects. It offers a comprehensive framework for assessing and managing future work in various socio-ecological systems based on complex systems thinking, adaptive management, and transdisciplinary principles. By integrating these methodologies, the ecosystem approach ensures that projects are effective, resilient, and adapted to dynamic environments [40]. The increasingly complex business world presents companies with numerous new challenges, which they often cannot solve independently. To be successful now and in the future, companies must collaborate and engage in business ecosystems. This collaboration can lead to significant competitive advantages for the companies and benefits for customers. However, strategic alignment in the ecosystem landscape remains challenging for many companies and is characterized by significant uncertainty. This approach involves organizing activities, particularly in projects, based on inter-industry partnerships and the involvement of ecosystem agents, among other factors.

As the world becomes increasingly interconnected and interdependent, the question arises: how to balance the equilibrium between the demand for new products and services and the limited supply under constraints defined by competitive environments, consumer values, and the capabilities of project teams. This article is dedicated to addressing this issue. [39]

3. Research Objective

This study aims to explore and substantiate modern methods and approaches to project management in cases related to the challenges of evaluating project execution under time constraints. Specifically, the article focuses on developing effective strategies and practical solutions to enhance a team's productivity in a Scrum environment under restrictive conditions. To achieve this objective, the study proposes justifying the hypothesis that an ecosystem approach enables a comprehensive and holistic assessment of the constraints faced by a project team, developing strategies for the successful implementation of a project that takes these constraints into account, identifying the potential of Scrum environments in constraint-sensitive situations, and ensuring a more accurate understanding of their strengths and limitations.

4. Methods of Research and Data Management

The article employs the Comparison-based Estimation approach using Story Points (Mallidi and Sharma [41]) to conduct the research. This approach is based on comparing the characteristics of the current project with those of a previous similar project. Such a comparison enables the team to assess the complexity of a specific feature and assign a numerical value that reflects its difficulty. The research is based on a case study of a project team's work and proposes a methodology for estimating future tasks based on generalized experience. Resource estimation for completing future tasks is carried out through the following sequential steps:

Step 1. Defining the type of work and its scope based on task decomposition. To determine when the work can be completed, it is first essential to understand what is meant by "work." At this initial stage, the involvement of a business analyst and a product manager is critical to define the scope of the work clearly.

Step 2. Estimating the scope of work based on historical data. At this stage, team members often participate in expert evaluations, which is considered a reliable approach. However, the team in the case study discussed in this article faced time constraints and was reluctant to engage in such evaluations. They preferred to start working on new features immediately without distraction.

Step 3. Calculating the team's velocity. To assess the feasibility of a specific request, it is crucial to calculate the team's velocity and estimate for completing the request. Comparing these values provides an understanding of what is realistically achievable.

Step 4. Calculating the value generated by each sprint during the execution of all work. At this stage, it is important to compare the types of work performed within the planned scope and the tasks that have already been completed. The calculated value helps determine the level of the team's fulfillment of customer requests.

Step 5. Formulating the calculation results and identifying the gap between the desired and estimated task completion state.

Step 6. Proposing and substantiating strategies to improve task execution based on the use of an ecosystem approach.

To ensure the clear application of the proposed methodology for estimating future work for a project team under constraints, it is essential to define the key project elements with which the team interacts. It is important to note that these definitions are not universally recognized in the industry. Instead, they are the ones agreed upon by the studied team, and understanding this is critical for grasping the entirety of the article. The definitions are as follows:

Epic - An epic represents a large scope of work comprising multiple interrelated tasks.

Requirement - A requirement is a description of new functionality that the product must have to meet business objectives. Requirements serve as the foundation for product design, testing, and implementation.

Technical Improvements - Technical improvements refer to changes or enhancements aimed at improving the performance, stability, security, or usability of a system or product.

Bugs - Bugs are defects or malfunctions that cause a product to operate incorrectly. They may arise from developer errors, insufficient testing, or changes in requirements.

Investigation Tickets - Investigation tickets are tasks undertaken by the team to explore a specific problem, technology, or solution that requires deeper analysis before implementation.

Incidents - Incidents are unexpected events or issues that disrupt the normal functioning of a product. They usually originate from end users of the product and are treated with the highest priority.

5. Presentation of Core Material

All work requires an evaluation of its outcomes. This principle equally applies to project activities in the IT domain. Established and tested methodologies to optimize project team performance and offer measurement systems alongside the conditions necessary to achieve them. Forming functions and collaborative efforts in project execution utilize performance and efficiency criteria. Performance refers to how successfully a team achieves its objectives. For a Scrum team, the goal is to deliver value to the customer [2]. Efficiency reflects the ability to achieve results while minimizing the use of resources such as time, money, and effort. It measures how well something is accomplished. For Scrum teams, efficiency can be assessed by the amount of completed work, the quality of that work, and the speed of its completion. Thus, evaluating the work of a project team requires consideration of both criteria.

This article examines a case study involving a project IT team operating in the financial technology sector and using the Scrum framework in its work [42]. Unlike the classical Scrum framework, where the team consists of Developers, a Product Owner, and a Scrum Master, the studied team featured slightly modified roles:

Developers were distinctly divided into separate roles:

UI and API Developers - Developers specializing in different parts of the code;

QA Manual and QA Automation Engineers - Responsible for manual and automated testing;

Business Analysts - Tasked with writing specifications for all developers.

The *Product Owner* role remained unchanged. The *Scrum Master* was replaced by a *Project Manager*, who had broader and more traditional responsibilities. While differing from conventional Scrum roles, these adaptations reflect the team's unique dynamics and resource constraints in planning and executing future tasks. Despite these differences, the team adhered to all core Scrum events, including Sprint (2-week duration), Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective. Additionally, the team followed the fundamental principles of Scrum (transparency, inspection, adaptation), ensuring self-organization and flexibility in project execution [43]. The team recently completed its 100th sprint, signifying a high level of maturity and stability. Strong internal connections, well-established processes, and an ecosystem comprising team members, customers, partners, and other stakeholders contributed to the team's robust performance.

One of the key challenges faced by the team was estimating the timeline for completing a complex task. The team evaluated future work and determined the feasibility of delivering functionality within eight weeks (equivalent to four sprints). Initially, this seemed impossible due to tight deadlines, but the team had no option but to decline the request. Rising to the challenge, they managed to satisfy the requirement through concerted effort. This case study explores how the team assessed constraints and resolved the complex task using prior estimations and an ecosystem-based approach. The solution provides valuable insights into the adjustments made during planning and execution, as well as the use of the Scrum framework to ensure on-time delivery. The findings lay the groundwork for scaling the developed approach to evaluate the performance of the project team.

5.1. Step 1. Defining the Type of Work and its Scope Based on Task Decomposition.

To evaluate the team's capacity to complete a task, it is crucial first to define what is meant by the term "work." An initial analysis revealed that the development of four requested functionalities was required. Consequently, four epics with high-level descriptions of the work were created. The business analyst then decomposed each epic into smaller, executable tasks. A key condition for this decomposition was ensuring that each task could be completed within a single sprint (two weeks). While developer involvement is typically necessary at this stage, in this case, the business analyst performed the decomposition independently and presented the results to the team. This approach was adopted because the business analyst possesses extensive experience with the current project, and the team fully trusts their ability to perform such work accurately. The results of this step were as follows (Figure 1):

1. Planned Epic #1: 10 tasks;
2. Planned Epic #2: 16 tasks;
3. Planned Epic #3: 11 tasks;
4. Planned Epic #4: 11 tasks.

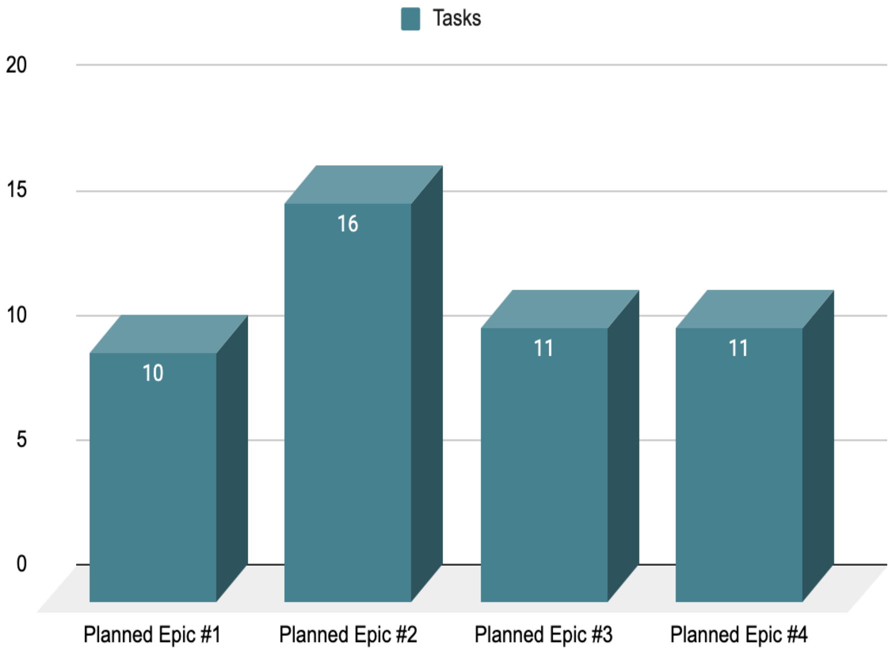

Figure 1.
Decomposition of work into planned epics and planned tasks.

Figure 1 illustrates the 4 epics, each representing a distinct functionality to be developed. Each epic is divided into specific tasks. In total, all four epics comprise 48 tasks.

5.2. Step 2. Estimating the Scope of Work Based on Historical Data.

At this stage, teams often rely on expert estimation to evaluate the scope of work, which is a traditional approach. However, the studied team faced time constraints and was reluctant to engage in such estimations because it would require additional unavailable time. The team preferred to begin working on the new tasks immediately without delays.

The team opted for a method based on historical data to avoid expert estimation. By analyzing previously completed functionalities, the team could identify a reliable benchmark for estimating the scope of upcoming work. The team reviewed the last four completed epics and calculated the number of tasks in each:

1. Completed Epic #1: 20 tasks;
2. Completed Epic #2: 30 tasks;
3. Completed Epic #3: 14 tasks;
4. Completed Epic #4: 4 tasks.

To further understand the process of scope estimation based on historical data, it is necessary to explain the concept of Story Points and how the studied team uses them. Story Points are a unit of measurement used to estimate the complexity and effort required to complete a task. They are not tied to specific time frames but rather reflect the relative complexity of a task as perceived by the team.

Understanding how Story Points are applied in the project is critical to comprehending the team's approach to work estimation. The team adheres to a set of guidelines for estimation while acknowledging the absence of strict rules regarding the use of Story Points, allowing flexibility to define practices tailored to their needs.

The studied team has established specific criteria for estimating tasks:

Requirements - These are thoroughly evaluated and serve as the foundation for project planning.

Technical Improvements - Although they generally do not directly impact business value, Technical Improvements are still assessed. The team treats technical debt as an integral component of such enhancements.

Bugs - Over time, the team has refined its approach to estimating future work by deciding to assign Story Points to bugs no longer. Instead, limiting the number of unresolved bugs has proven to be a more efficient alternative, saving valuable time on estimation.

Investigation Tickets - These requests are conservatively estimated, usually at 1 or 2 Story Points, serving as a time constraint to prevent the investigation from becoming overly time-consuming.

Incidents - Errors reported by end users, given their high priority, are not estimated. They are addressed promptly without undergoing the estimation process.

The team reviews each of the tasks during a Product Backlog Refinement meeting. The team conducts a voting process to estimate the tasks, discussing any differences in their evaluations. Discussions continue until all team members agree on a single, unified estimation.

As a result, the team calculated how many Story Points were allocated to each task within the Completed Epics. The results of this step are summarized below (Figure 2):

1. Completed Epic #1: 20 tasks = 47.5 story points;
2. Completed Epic #2: 30 tasks = 61.5 story points;
3. Completed Epic #3: 14 tasks = 29.5 story points;
4. Completed Epic #4: 4 tasks = 7 story points.

Summarizing the data, we obtained 68 tasks and 145.5 story points. This allowed us to determine that 1 task is approximately equivalent to 2.14 story points.

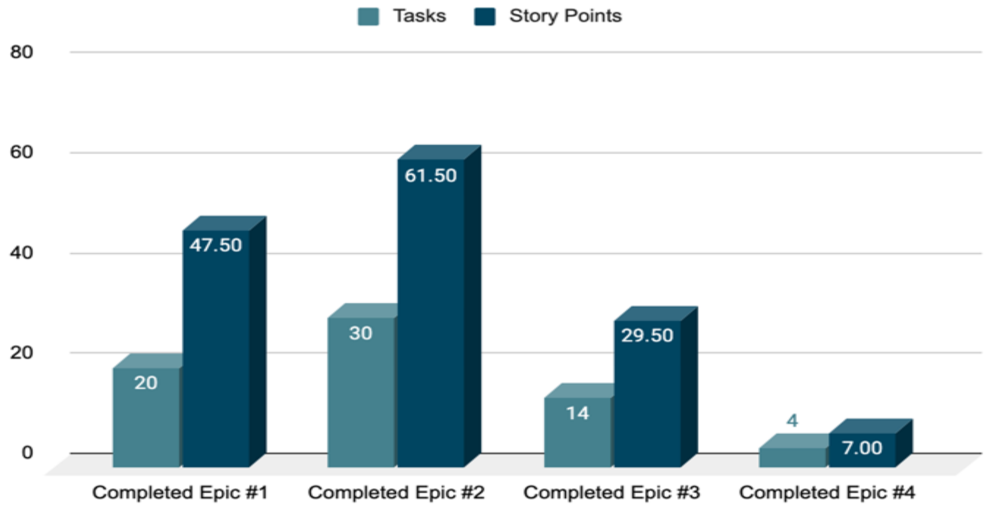

Figure 2.
Evaluation of completed epics.

This information became the basis for estimating the planned Epics (Figure 3):

1. Planned Epic #1: 10 tasks = 21.40 story points;
2. Planned Epic #2: 16 tasks = 34.24 story points;
3. Planned Epic #3: 11 tasks = 23.54 story points;
4. Planned Epic #4: 11 tasks = 23.54 story points.

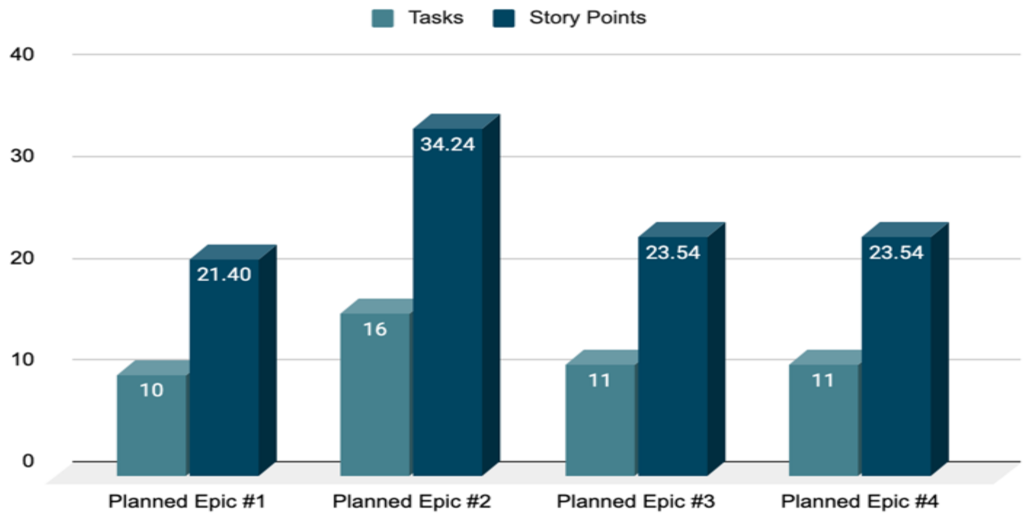

Figure 3.
Epics execution forecast in story points.

Overall, the estimated execution of the planned work amounted to 102.72 story points. Consequently, the execution of 1 task of planned work is equivalent to 2.14 story points.

5.3. Step 3. Calculating the Team's Velocity.

The velocity of the studied team is the number of story points in fully completed tasks. Velocity may vary between different sprints, so it is important to examine the average velocity, which reflects the mean value over several sprints.

To assess the feasibility of a specific request, it is crucial to calculate the team's velocity along with an estimate for the request. Comparing these values provides insight into what can realistically be achieved.

By analyzing the team's performance in the last 10 sprints, the number of story points in completed tasks was identified (Figure 4):

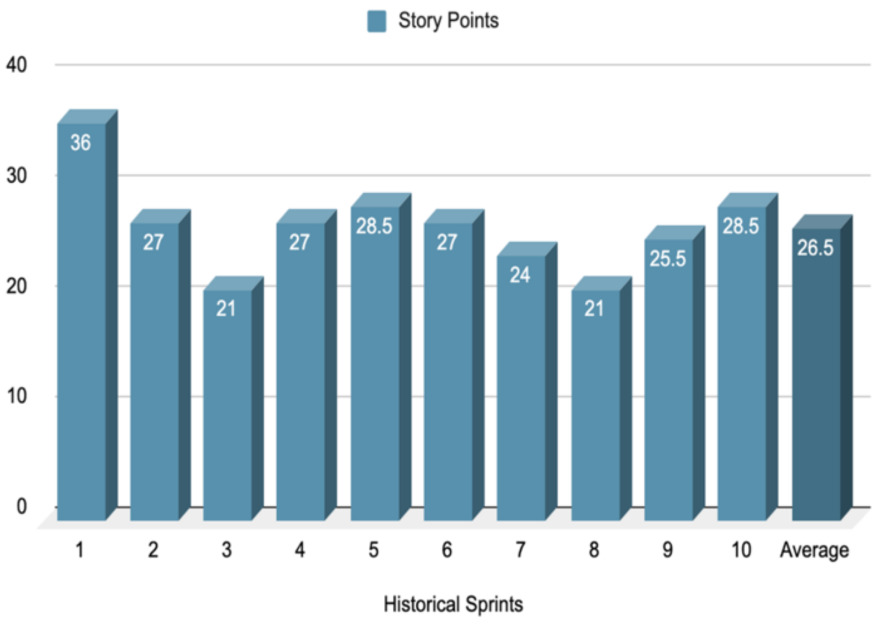

Figure 4.
Number of completed story points in the last ten sprints.

1. Sprint #1: 36 story points;
2. Sprint #2: 27 story points;
3. Sprint #3: 21 story points;
4. Sprint #4: 27 story points;
5. Sprint #5: 28.5 story points;
6. Sprint #6: 27 story points;
7. Sprint #7: 24 story points;
8. Sprint #8: 21 story points;
9. Sprint #9: 25.5 story points;
10. Sprint #10: 28.5 story points;

The average sprint velocity is calculated as 265.5 / 10 = 26.55 story points.

A common mistake at this stage is to rely solely on the average velocity and multiply it by the number of planned sprints. The calculation would be 26.55 x 4 = 106.20 story points in this case. This could lead to the assumption that no additional work is necessary because the estimated 102.72 story points for the work, based on historical data, are lower than the 106.20 story points the team could deliver across four sprints. However, this calculation does not consider that full velocity includes new features, technical debt, improvements to existing functionality, defects, investigations, proof-of-concept work, and so on.

It is crucial to identify what contributes to full velocity. It is important to note that the team estimates Requirements, Technical Improvements, and Investigation Tickets but does not estimate Bugs and Incidents. Based on this, in addition to the full velocity, the team also calculated the specific implementation of functionality, where only Requirements are considered for the calculation.

Analyzing the team's performance over the last 10 sprints, the number of story points in completed requirements was identified (Figure 5):

1. Sprint #1: 31.5 story points;
2. Sprint #2: 18 story points;
3. Sprint #3: 14 story points;
4. Sprint #4: 20 story points;
5. Sprint #5: 22 story points;
6. Sprint #6: 27 story points;

7. Sprint #7: 17 story points;
8. Sprint #8: 14 story points;
9. Sprint #9: 20.5 story points;
10. Sprint #10: 21.5 story points.

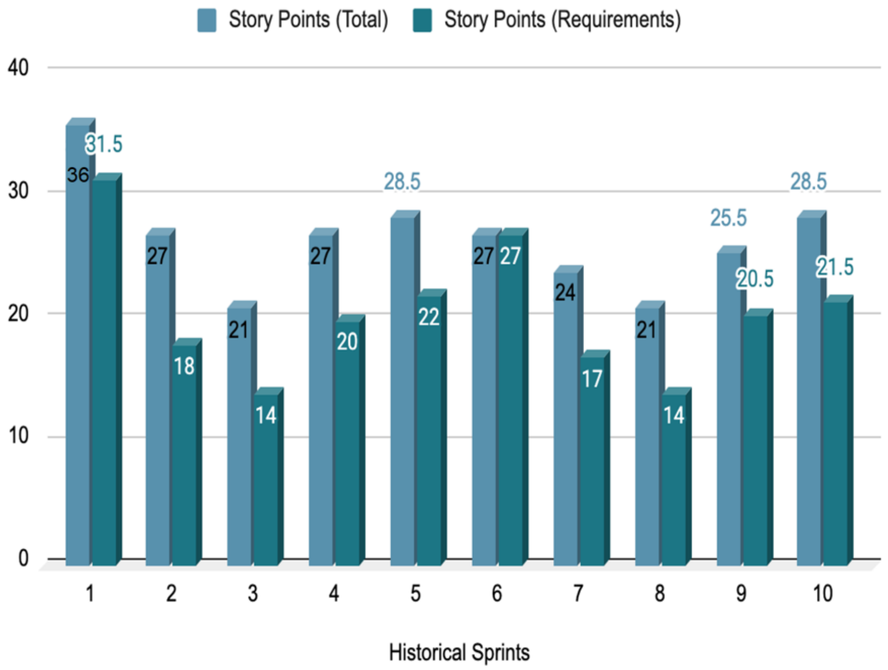

Figure 5.
Number of completed Story Points in total and specifically for Requirements in the last 10 sprints.

The average speed for requirements is 205.5 / 10 = 20.55 story points. Forecasting the implementation of functionality for the next 4 sprints at this clear speed results in 20.55 x 4 = 82.20 story points, which is significantly less than the estimated 102.72.

5.4. Step 4. Calculating The Value Generated by Each Sprint During the Execution of All Work.

By comparing the execution of Requirements to the total work completed, we can calculate this % (Figure 6).

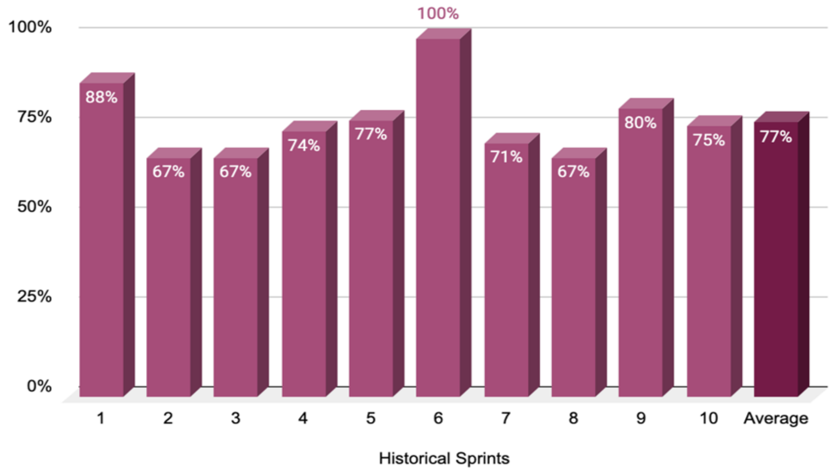

Figure 6.
The ratio of Requirements to total work completed in the last 10 sprints.

1. Sprint #1: 87.50%
2. Sprint #2: 66.67%
3. Sprint #3: 66.67%
4. Sprint #4: 74.07%
5. Sprint #5: 77.19%
6. Sprint #6: 100.00%
7. Sprint #7: 70.83%
8. Sprint #8: 66.67%

9. Sprint #9: 80.39%
10. Sprint #10: 75.55%
11. Total: 77.70% (205.50 / 265.50)
12. Average: 76.54%

Thus, the project team's capacity allowed them to execute, on average, 76.54% of the customer's requests and requirements.

5.5. Step 5. Formulating The Calculation Results and Identifying the Gap Between the Desired and Estimated Task Completion State.

These data show that, on average, only 77% of the work aligns with business value. To complete the necessary work without changing speed, we calculate what this figure should be, considering the obtained results:

- Overall evaluation of planned work completion: 102.72 story points;
- Planned sprints – 4;
- Desired speed of Requirements completion per sprint: 102.72 / 4 = 25.68 story points;
- Predicted speed based on historical results of Requirements completion per sprint: 205.5 / 10 = 20.55 story points;
- Ratio of predicted to desired speed: 20.55 / 25.68 = 80.02%.

Thus, the team's work evaluation is based on comparing the actual speed of work completion from previous projects with the speed calculated without considering the need to address defects, errors, and other refinements.

This step demonstrates that the team cannot complete the planned work within the next four sprints without accounting for risks and constraints by applying changes to the development process and increasing the clear percentage of completed work. Based on the results obtained, strategies were implemented to strengthen the team's ability to work on complex tasks under time constraints using an ecosystem approach.

5.6. Strategy #1. Collaborative Alignment of Tasks with the Entire Team

Experience has shown that in many teams, the full scope of tasks is often not disclosed to all team members. Typically, a specific task is assigned to an individual developer, and that's the end of it. Recognizing the limitations of this approach, it became evident that it would not lead to success in the current context. To address this limitation, a decision was made to share the planned scope of tasks with all team members during a collective meeting. This ensured that everyone clearly understood what needed to be developed and within what time frame. This approach eliminates uncertainties such as "What's next?" and fosters a collaborative environment where the entire team works toward a common goal [44]. We view alignment as critical for fostering collaboration and engagement in task execution [45]. Additionally, transparency was enhanced by disseminating information about task scope and estimated completion speed to all team members. This contributed to a shared understanding of which requested functionalities might be unachievable within the specified time frame without appropriate adjustments. Keeping the entire team informed about the scope of work is critically important, as it fosters a unified understanding of project objectives. This minimizes misunderstandings, aligns everyone around a common goal, and increases the team's overall effectiveness. When every team member has a complete picture of the work scope, it encourages collaboration, reduces the likelihood of miscommunication, and ultimately supports the team's success. Figure 7 and 8 compare traditional and ecosystem approaches to aligning tasks and objectives among team members. At its core, the ecosystem approach emphasizes the interdependent nature of project components. Changes in system modeling, problem assessment, or management methods do not occur in isolation but instead affect the overall structure and success of the project. This interconnection allows for a more comprehensive understanding of how various subsystems interact and contribute to broader objectives. In the example considered, problem-solving is more effective in a unified subsystem that brings project participants together for better task understanding and alignment of work requirements.

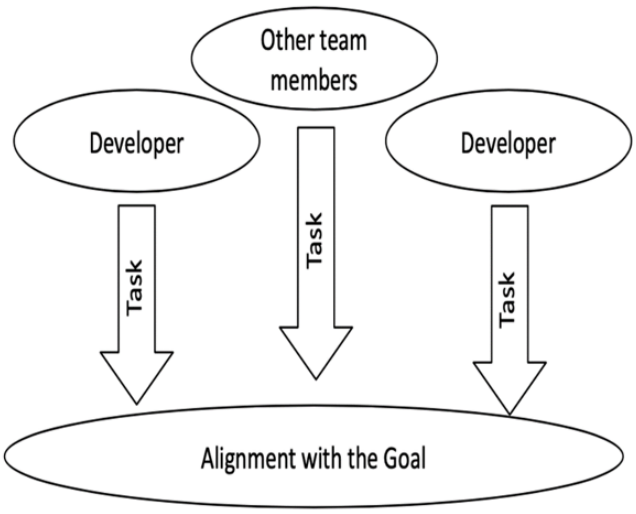

Figure 7.
Alignment with the goal after task completion.

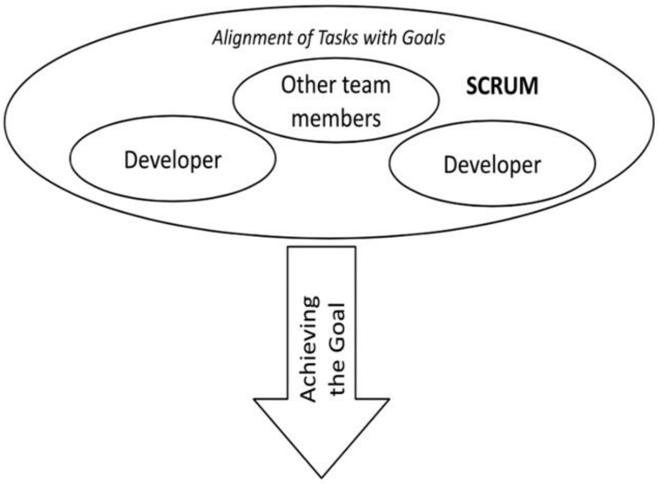

Figure 8.
Achieving the goal because of task alignment using the ecosystem approach.

5.7. Strategy #2. Focus on quality.

Software development quality is critical as it directly affects various aspects of the product lifecycle and the project's overall success. Even when time is limited and certain functionality must be delivered quickly, prioritizing quality remains crucial. In the quality assessment process, the team uses the total number of reworks as a key metric reflecting the current state of development. Reworks in this context refer not only to refinements but also to correcting defects that arise during the execution of assigned tasks. This includes fixing functionality errors, ensuring the product meets initial requirements, and confirming its proper operation under specified conditions. Defect corrections often result from shortcomings in initial analysis, design, or testing, necessitating a review of previously completed work. Figure 9 illustrates the current process of completing and verifying tasks, showing the conditions under which a task may require rework.

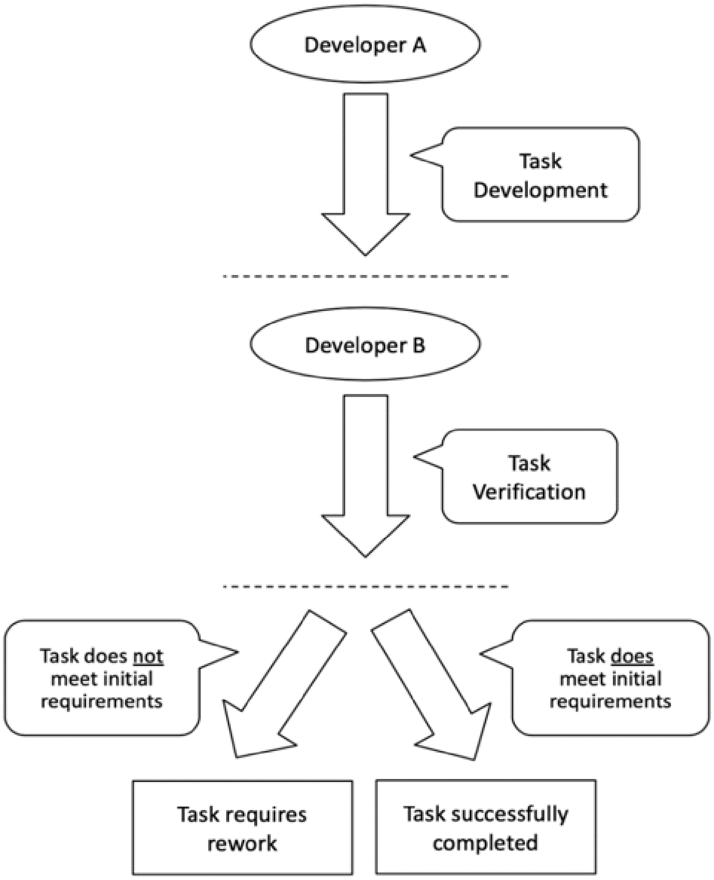

Figure 9.
The process of completing and verifying the task before implementing the strategy.

It was decided to evaluate the average number of reworks per sprint to implement future improvements. This evaluation enables all team members to share a common understanding of the current state.

As a result, the team analyzed the number of rework instances over the last ten sprints (Figure 10):

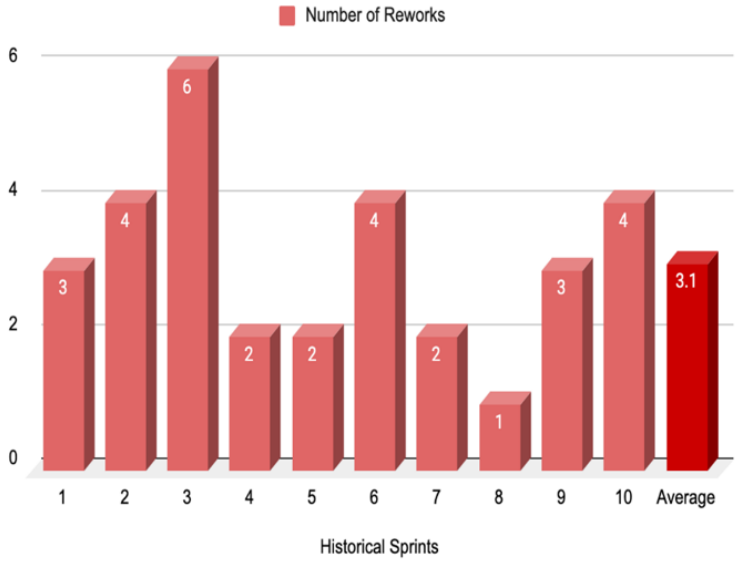

Figure 10.
The number of reworks in the last 10 sprints.

The average number of reworks over the past 10 sprints is 3.1 reworks per sprint. The team accepted this as a baseline for assessing task quality. As a result, the team aimed to reduce the number of reworks while working on the planned epics. To achieve this, it was decided that the developer performing the task would also carry out an additional check. Thus, an extra step was added to the task execution and verification process (Figure 11).

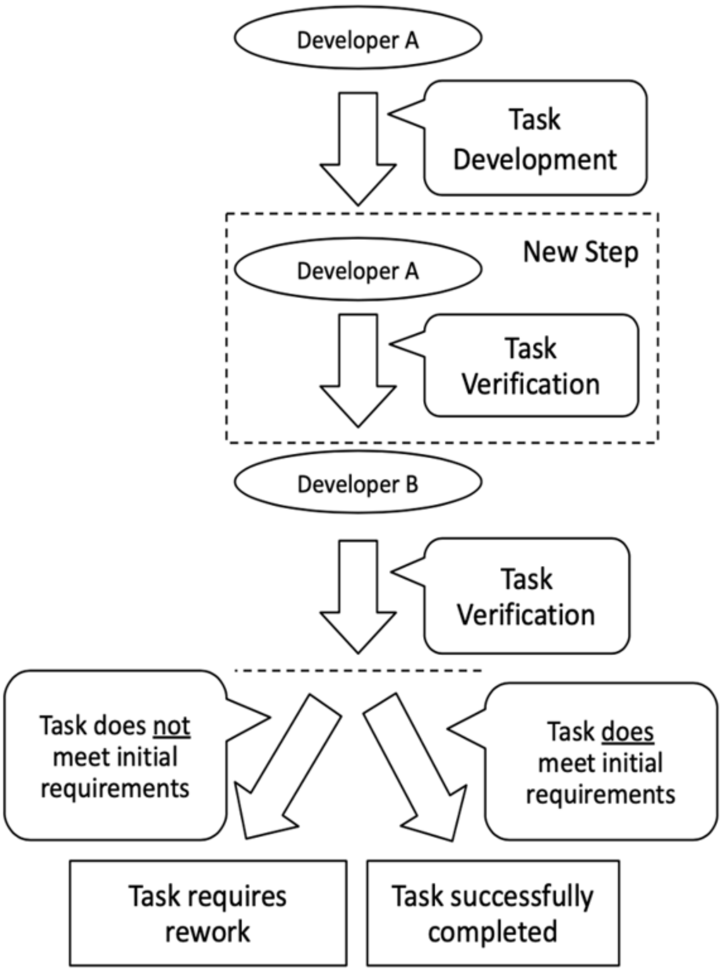

Figure 11.
The process of completing and verifying the task after implementing the strategy.

Looking ahead, this decision demonstrated significant improvements in terms of reworks. During the planned four epics, which spanned four sprints, only seven reworks were recorded. This resulted in a low rework rate of 1.75 reworks per sprint compared to 3.1 in historical sprints. As a result, the average number of reworks per sprint amounted to 56% of the baseline level.

5.8. Strategy #3. Identifying Bottlenecks.

During team discussions about factors hindering progress, an unexpected fact emerged: the main obstacle to improving efficiency was the Super Review Process for API tasks. To further understand the issue, it's necessary to clarify the types of tasks assigned to developers—namely API and UI tasks, which are carried out by API and UI developers, respectively. An API developer cannot perform UI tasks, and a UI developer cannot handle API tasks.

It is also necessary to explain the process of reviewing the code written by a developer. Before a task can be verified for compliance with initial requirements, the code for that task is reviewed by other developers. This process is called Peer Review. After a successful Peer Review, a Super Review is conducted, and only then can the task proceed to verification. The key issue lies in the fact that, just like development, both Peer Review and Super Review follow the same specialization constraints: API tasks can only be reviewed by API developers, and UI tasks only by UI developers. A schematic representation of the review process is shown in Figure 12.

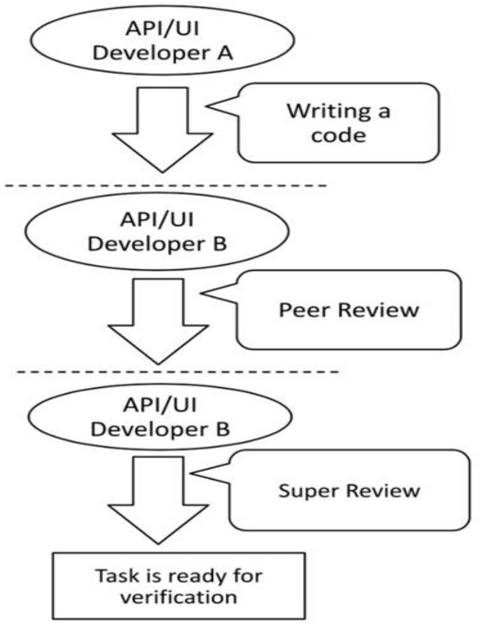

Figure 12.
Tasks review process.

After analyzing historical data, it was revealed that the Super Review process for API tasks takes an average of 1.9 days (Figure 13). During discussions about this issue, it became clear that the developer responsible for conducting the Super Review for API tasks works in a different time zone, with a 7-hour time difference from most of the team's developers. The solution to this issue is based on addressing the specifics of remote work, as discussed in the referenced source [46].

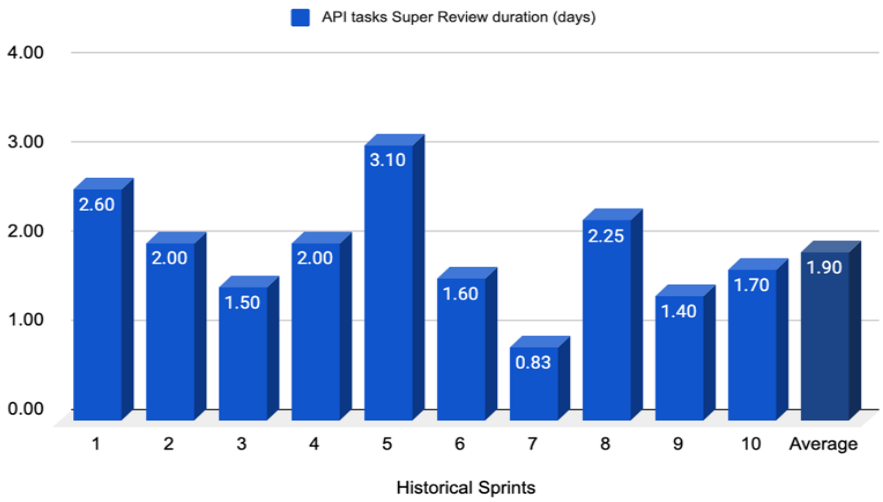

Figure 13.
API tasks Super Review duration in the last 10 sprints.

To address this issue, an additional API developer was added to the team working in the same time zone as most. This adjustment aimed to eliminate the problems associated with the previous time difference and enhance the overall efficiency of the Super Review process for API tasks. As a result, the team improved the average duration of the Super Review process for API tasks during the work on the planned epics. This figure dropped to 0.51 days, which is a 27% reduction compared to the historical data.

5.9. Strategy #4. Adding an Additional Developer.

In the field of software development, there is a common perception that adding additional developers often slows down processes, as the team goes through stages of forming, storming, norming, and performing [47]. Forming refers to team members getting to know each other and establishing basic rules of collaboration; storming is characterized by clashes of different perspectives and work styles; norming involves reaching mutual understanding and creating unified standards; and performing is the productive cooperation to achieve goals. However, the studied team faced a unique situation—they decided to add another developer who had previously worked on this project. This developer was already familiar with the existing structure, codebase, and team dynamics. Such prior knowledge allowed the team to skip the typical stages of forming, storming, and norming and quickly transition to performing. The seamless integration of this experienced developer enabled rapid and effective collaboration, contradicting the usual expectations of team slowdowns when introducing new members.

5.10. Strategy #5. Implementing "After Start Time" Approach.

While Scrum does not require the completion of all planned tasks within a sprint or the immediate implementation of new tasks, it strongly encourages timely completion. As a result, developers often select tasks based on their likelihood of being completed within the sprint. This approach has several drawbacks. The primary disadvantage is that priority is given not to the most important tasks but to those that can be completed by the end of the sprint. Understanding that this decision-making process might not always align with the team's goals, the team decided to implement the "After Start Time" approach, which is unique to this team. This approach addresses the issue of prioritizing the most critical tasks, allowing developers to take on the next task under specific conditions.

1. There are no current sprint tasks where the developer can assist;
2. There are no bugs (defects) in the current sprint where the developer's help is needed;
3. None of the colleagues require assistance with ongoing tasks.

Only after these conditions are met a developer can take on the next planned task, with the understanding that it will not be completed within the current sprint.

5.11. Strategy #6. Requesting Vacation Postponement.

It is important to note that this action is highly contentious and generally not recommended due to its potential impact on the current and future trajectory of the project. In exceptional circumstances, where the urgency of the project is particularly high, teams may consider the possibility of postponing planned vacations. It should be emphasized that this measure must be approached with utmost caution, as it may negatively affect both the immediate and long-term success of the project. The decision to request team members to defer their vacations was made collectively, ensuring open dialogue and a clear understanding of the potential consequences. This action highlights the exceptional nature of the circumstances under which such a request can be made, recognizing the possible drawbacks and the necessity of carefully considering the overall dynamics of the team.

6. Results

Thanks to their dedicated efforts, the team successfully achieved the assigned goal, specifically developing the requested functionality within the following eight weeks, equivalent to four sprints. This accomplishment highlights the team's ability to work effectively under time constraints while maintaining high task quality, utilizing human resources efficiently, and adhering strictly to Scrum principles.

Let's delve deeper into the results. After completing the planned functionality, the team's velocity during its execution was assessed. As mentioned earlier, the planned work was completed within four sprints.

For simplicity, a new term is introduced here and going forward: "Observed Sprints," referring to the sprints during which the planned work was carried out.

Analyzing the team's metrics for these four sprints revealed the number of story points completed in the finished tasks (Figure 14):

The increase in velocity, especially during the first two sprints, demonstrates the team's ability to achieve higher productivity in the short term. For comparison, the average velocity in the Historical Sprints was 26.5 story points for total tasks and 20.55 story points for requirements. However, it is important to note that such growth is sustainable only for a limited period, as evidenced by the gradual leveling off of velocities in sprints 3 and 4.

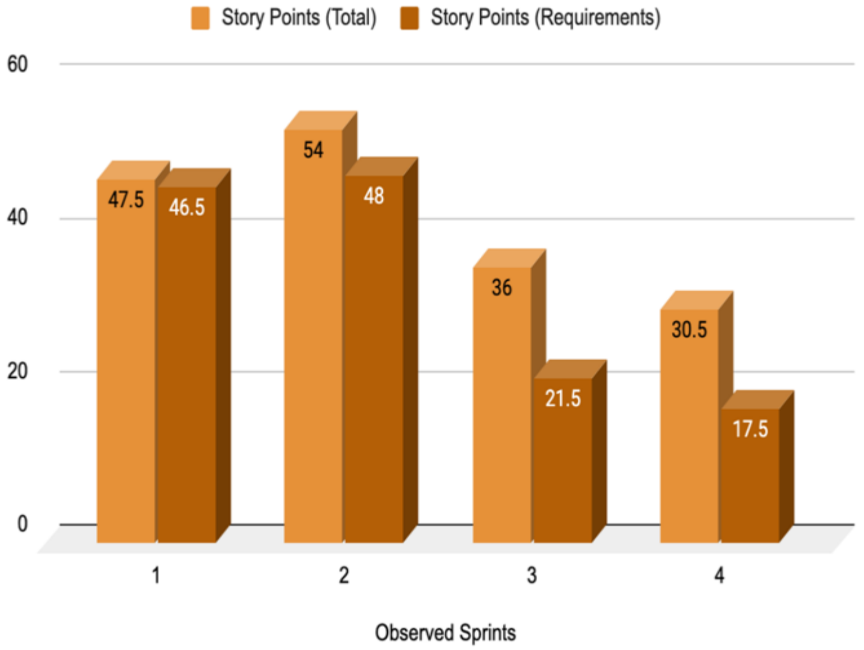

Figure 14.
The total number of completed story points, specifically for requirements in the four observed sprints.

The total number of completed story points in the observed sprints is 47.5 + 54 + 36 + 30.5 = 168 story points for total tasks. Similarly, the total number of story points for requirements is calculated as 46.5 + 48 + 21.5 + 17.5 = 133.5 story points.
By dividing these totals by the number of observed sprints, the average velocity is calculated as:
168 / 4 = 42 story points - the average team velocity in the observed sprints for total tasks.
133.5 / 4 = 33.38 story points - the average team velocity in the observed sprints for requirements.
Figure 15 shows a comparison of this data with historical metrics:

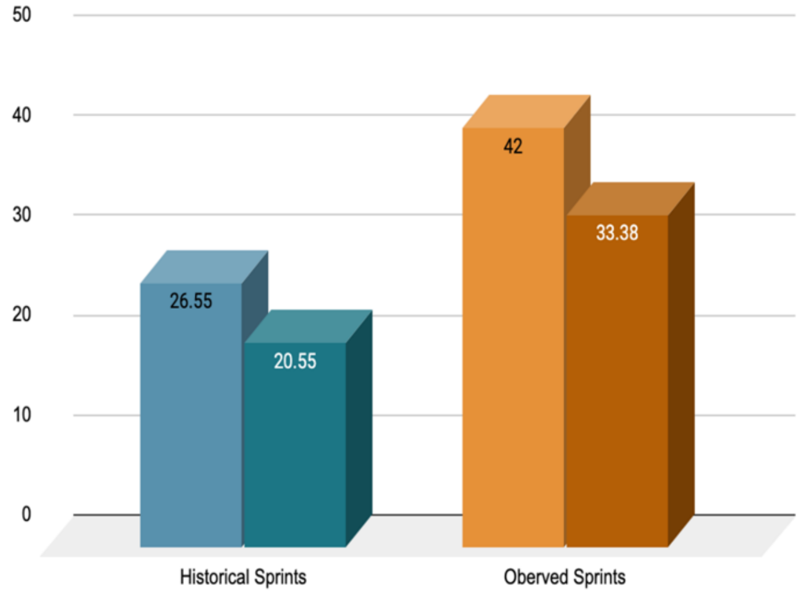

Figure 15.
Comparison of the team's average velocity between historical and observed sprints.

The average ratio of requirements completion to the total completed work is calculated as follows:
20.55 / 26.55 = 77.40% - historical sprints.
33.38 / 42 = 79.48 - observed sprints.
This ratio can also be calculated for each of the studied sprints (Figure 16):

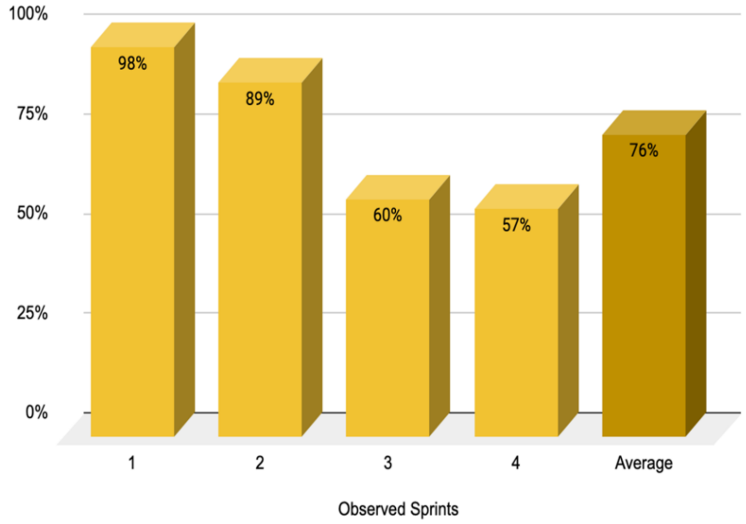

Figure 16.
The ratio of requirements to total completed work in the 4 observed sprints.

The results obtained confirm the necessity of evenly distributing tasks throughout the entire process of their execution. As shown in Figure 16, the maximum focus on requirements was placed in the first two sprints. However, the ratio of such tasks to the total sharply decreased in the next two sprints. This indicates the necessity of performing all types of tasks.

It is important to emphasize once again that the task set for the team was to complete certain functionality in a very short period. By following standard methods, the team would not have been able to achieve this goal. The steps and strategies described in this article helped the team reach the set objective.

7. Conclusions

The case considered in the article describes approaches to assessing a complex project under time constraints and environmental uncertainty. The focus is on practical project management methods, particularly the Scrum framework. The analysis demonstrates how the team effectively addressed the client's tasks under constraints such as limited time, human resources, and specific requirements.

The calculations conducted within the proposed methodology identified key constraints requiring strategic resolution. The results allowed for the assessment of task completion speed, identification of bottlenecks, and the development of an appropriate strategy for team functioning oriented toward an ecosystem approach.

In summary, the ecosystem approach provides a reliable foundation for assessing future work in project implementation. Its emphasis on integration, innovation, and adaptability contributes to the successful execution of projects and meeting and exceeding client expectations. The proposed strategies enable teams to overcome modern challenges, reduce time costs, and create long-term value.

Based on the case analysis, it is concluded that Scrum is a key tool for improving team productivity in the short term. Task alignment, a focus on quality, bottleneck elimination, involvement of additional developers, and changes in task execution sequences increased team efficiency. This confirms Scrum’s adaptability in situations with strict time constraints and its ability to deliver outstanding results.

Consistently following the stages of assessment confirmed the hypothesis that the ecosystem approach promotes a comprehensive solution to tasks. The methodology is based on analyzing historical experience and allows for the development of strategies to achieve set goals.

7.1. Theoretical and Empirical Significance

The theoretical significance lies in developing a flexible methodology for evaluating the scope of client tasks. The approaches to assessing task complexity improve the efficiency of Scrum in time-sensitive projects, ensuring a more precise understanding of its capabilities and limitations.

Empirical significance is based on a real case of a project team's work. The obtained results confirm the positive impact of the proposed methodology on team productivity and achieving desired outcomes.

7.2. Limitations and Prospects

The main limitation lies in the difficulty of predicting future client requirements. However, the analyzed case demonstrates that simple calculations can help identify historical prerequisites, determine client parameters, and adjust indicators while considering the value of work for the client and the team's ability to deliver it.

Taking the results into account, the experience of task completion forms a knowledge curve that allows for optimizing time costs and achieving expected results. This experience creates new growth points, contributing to sustainable development and improving project management practices.

References

[1] O. M. Riabchykov and L. M. Hanushchak-Yefimenko, "The impact of iteration risk coefficient on the volatility of scrum team work in the scrum methodology," *Journal of Strategic Economic Research*, vol. 1, pp. 94–99, 2024. https://doi.org/10.30857/2786-5398.2024.1.10

[2] B. Cartaxo, A. Araújo, A. S. Barreto, and S. Soares, "The impact of scrum on customer satisfaction: An empirical study," in *2013 27th Brazilian Symposium on Software Engineering. https://doi.org/10.1109/SBES.2013.10*, 2013, pp. 129-136.

[3] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, and M. Fowler, "Manifesto for agile software development," Retrieved: https://agilemanifesto.org, 2001.

[4] V. Hema, S. Thota, S. N. Kumar, C. Padmaja, C. B. R. Krishna, and K. Mahender, "Scrum: An effective software development agile tool," in *IOP Conference Series: Materials Science and Engineering. https://doi.org/10.1088/1757-899X/981/2/022060*, 2020, vol. 981, no. 2, p. 022060.

[5] F. K. Levy, G. L. Thompson, and J. D. Wiest, "The ABCs of the critical path method harvard business review," Retrieved: https://hbr.org/1963/09/the-abcs-of-the-critical-path-method, 1963.

[6] W. P. Millhiser and J. G. Szmerekovsky, "Teaching critical chain project management: The academic debate and illustrative examples," *INFORMS Transactions on Education*, vol. 12, no. 2, pp. 67-77, 2012. https://doi.org/10.1287/ited.1110.0083

[7] E. Flores-Garcia, J. Bruch, M. Wiktorsson, and M. Jackson, "Decision-making approaches in process innovations: an explorative case study," *Journal of Manufacturing Technology Management*, vol. 32, no. 9, pp. 1-25, 2021. https://doi.org/10.1108/JMTM-03-2019-0087

[8] E. Bukłaha, "The origin and types of controlling in project management," *Journal of Management and Financial Sciences*, vol. 32, pp. 27-37, 2018. https://doi.org/10.33119/JMFS.2018.32.2

[9] B. G. Tavares, C. E. S. Da Silva, and A. D. De Souza, "Risk management in scrum projects: A bibliometric study," *Journal of Communications Software and Systems*, vol. 13, no. 1, pp. 1-8, 2017. https://doi.org/10.24138/jcomss.v13i1.241

[10] P. L. Torres and H. Cornide-Reyes, "Team performance assessment and improvement in an agile methodology framework," presented at the 2022 XVLIII Latin American Computer Conference (CLEI), Armenia, Colombia, pp. 1–10, 2022. DOI: https://doi.org/10.1109/CLEI56649.2022.9959959, 2022.

[11] Q. Zhou, X. Deng, G. Wang, and A. Mahmoudi, "Linking elements to outcomes of knowledge transfer in the project environment: Current review and future direction," *Frontiers of Engineering Management*, vol. 9, no. 2, pp. 221-238, 2022. https://doi.org/10.1007/s42524-022-0195-3

[12] P. Tominc, D. Oreški, and M. Rožman, "Artificial intelligence and agility-based model for successful project implementation and company competitiveness," *Information*, vol. 14, no. 6, p. 337, 2023. https://doi.org/10.3390/info14060337

[13] IBM, "Use key performance indicators to measure and guide progress," Retrieved: https://www.ibm.com/garage/method/practices/learn/kpis-measure-guide-progress/, n.d.

[14] Project Management Institute, " A guide to the project management body of knowledge (PMBOK® guide)," 2017.

[15] B. Little, "The principles of successful project management: It takes careful planning, skilful leadership… and a little bit of luck," *Human Resource Management International Digest*, vol. 19, no. 7, pp. 36-39, 2011. https://doi.org/10.1108/09670731111175597

[16] F. Almeida and P. Carneiro, "Performance metrics in scrum software engineering companies," *International Journal of Agile Systems and Management*, vol. 14, no. 2, pp. 205-223, 2021. http://dx.doi.org/10.1504/IJASM.2021.10041601

[17] F. Almeida and P. Carneiro, "Perceived importance of metrics for agile scrum environments," *Information*, vol. 14, no. 6, p. 327, 2023. https://doi.org/10.3390/info14060327

[18] D. E. Lysenko and Y. V. Fediai, "The Agile earned value management method as a cost management tool in scrum projects," *Information Systems and Technologies*, vol. 3, no. 163, pp. 159–164, 2021. https://doi.org/10.33042/2522-1809-2021-3-163-159-164

[19] S. Bansai, "Velocity - an agile metric," Retrieved: https://www.izenbridge.com/blog/velocity-in-agile-scrum, n.d.

[20] Agile Pinoy, "Measuring team velocity," Retrieved: https://agilepinoy.wordpress.com/2018/01/11/measuring-team-velocity, n.d.

[21] P. Doshi, "Agile metrics: Velocity," Retrieved: https://www.scrum.org/resources/blog/agile-metrics-velocity, n.d.

[22] K. P. Pham and M. Neumann, "How to measure performance in agile software development? A mixed-method study," in *2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). https://doi.org/10.1109/SEAA64295.2024.00074*, 2024, pp. 443-450.

[23] C. Lindskog and J. Netz, "Balancing between stability and change in Agile teams," *International Journal of Managing Projects in Business*, vol. 14, no. 7, pp. 1529-1554, 2021. https://doi.org/10.1108/IJMPB-12-2020-0366

[24] J. Corbin and A. Strauss, "Basics of qualitative research: Techniques and procedures for developing grounded theory," 4th ed. Thousand Oaks, CA: Sage Publications, 2014, p. 77.

[25] V. Venkatesh, J. Y. Thong, F. K. Chan, H. Hoehle, and K. Spohrer, "How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills," *Information Systems Journal*, vol. 30, no. 4, pp. 733-761, 2020. https://doi.org/10.1111/isj.12282

[26] M. P. Buvik and S. D. Tvedt, "The influence of project commitment and team commitment on the relationship between trust and knowledge sharing in project teams," *Project Management Journal*, vol. 48, no. 2, pp. 5-21, 2017. https://doi.org/10.1177/875697281704800202

[27] A. Polyanska and R. Psiuk, "Cognitive methods of manager behavior formation in the conditions of international enterprise activities," in *Advances in Manufacturing II: Volume 3-Quality Engineering and Management. https://doi.org/10.1007/978-3-030-17269-5_14*, 2019, pp. 194-206.

[28] A. Polyanska and O. Malynka, "Knowledge management as a basis of modern development of the companies," *Journal of Eastern European and Central Asian Research*, vol. 1, no. 2, pp. 9-9, 2014. http://dx.doi.org/10.15549/jeecar.v1i2.19

[29] A. Polyanska, S. Savchuk, I. Zapukhliak, Y. Zaiachuk, and I. Stankovska, "Digital maturity of the enterprise as an assessment of its ability to function in industry 4.0," presented at the In International Scientific-Technical Conference MANUFACTURING (pp. 209-227). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-99310-8_17, 2022.

[30] H. William Dettmer, *Goldratt's theory of constraints a systems approach to continuous improvement*. ASQ Quality Press Milwaukee: Wisconsin6, 1997.

[31] K. B. Hendricks and V. R. Singhal, "The long-run stock price performance of firms with effective TQM programs," *Management Science*, vol. 47, no. 3, pp. 359-368, 2001. http://dx.doi.org/10.1287/mnsc.47.3.359.9773

[32] D. Hodgson, "Disciplining the professional: The case of project management," *Journal of Management Studies*, vol. 39, no. 6, pp. 803-821, 2002. http://dx.doi.org/10.1111/1467-6486.00312

[33] V. R. Kannan and K. C. Tan, "Just in time, total quality management, and supply chain management: Understanding their linkages and impact on business performance," *Omega,* vol. 33, no. 2, pp. 153-162, 2005. https://doi.org/10.1016/j.omega.2004.03.012

[34] P. Kumar, "Analysis the impact of classical management approaches on the management practices," *Asian Journal of Management,* vol. 8, no. 3, pp. 841-853, 2017. https://doi.org/10.5958/2321-5763.2017.00132.9

[35] T. Petrenko, "Working with constraints in project management," *Zeszyty Naukowe Wyższej Szkoły Bankowej w Poznaniu,* vol. 100, no. 1, 2023. https://doi.org/10.58683/dnswsb.633

[36] M. Vanhoucke, "Critical chain/buffer management: Protecting the schedule against project delays," Retrieved: http://www.pmknowledgecenter.com/dynamic_scheduling/risk/critical-chain, n.d.

[37] H. Steyn, "Project management applications of the theory of constraints beyond critical chain scheduling," *International Journal of Project Management,* vol. 20, no. 1, pp. 75-80, 2002. https://doi.org/10.1016/S0263-7863(00)00054-5

[38] F. Rugenyi and H. Bwisa, "Effects of triple constraint on the management of projects in Nairobi, the project manger's perspective," *Strategic Journal of Business and Change Management,* vol. 3, no. 2, 2016. http://dx.doi.org/10.61426/sjbcm.v3i2.241

[39] C. Park, "Project ecosystem competency model," *Procedia-Social and Behavioral Sciences,* vol. 226, pp. 116-123, 2016. http://dx.doi.org/10.1016/j.sbspro.2016.06.169

[40] C. H. Richter, J. Xu, and B. A. Wilcox, "Opportunities and challenges of the ecosystem approach," *Futures,* vol. 67, pp. 40-51, 2015. https://doi.org/10.1016/j.futures.2014.12.002

[41] R. K. Mallidi and M. Sharma, "Study on agile story point estimation techniques and challenges," *International Journal of Computer Applications,* vol. 174, no. 13, pp. 9-14, 2021.

[42] Scrum Guide, "Scrum framework poster," Retrieved: https://www.scrum.org/resources/scrum-framework-poster, n.d.

[43] D. McGreal and R. Jocham, *The professional product owner: Leveraging scrum as a competitive advantage*. Boston: Addison-Wesley Professional, 2018.

[44] A. S. Polyanska, V. O. Martynets, V. R. Psyuk, Y. R. Kashchenko, and A. O. Maksymchuk, "Transformation of the business ecosystem model into energy enterprise's strategy," presented at the In IOP Conference Series: Earth and Environmental Science (Vol. 1415, No. 1, p. 012055). IOP Publishing. http://dx.doi.org/10.1088/1755-1315/1415/1/012055, 2024.

[45] V. Petrenko, A. Polyanska, L. Verbovska, and G. Bodnar, "A view on harmonization of interaction of business entities in conditions of change," innovations in industrial engineering. icieng 2021," *Lecture Notes in Mechanical Engineering,* pp. 342–359, 2021. https://doi.org/10.1007/978-3-030-78170-5_30

[46] A. Polyanska, Y. Pazynich, Z. Poplavska, Y. Kashchenko, V. Psiuk, and V. Martynets, "Conditions of remote work to ensure mobility in project activity. in international scientific-technical conference manufacturing." Cham: Springer Nature Switzerland, 2024, pp. 151-166.

[47] G. Stavrou, P. Adamidis, J. Papathanasiou, and K. Tarabanis, "Team formation: A systematic literature review," *International Journal of Business Science and Applied Management,* vol. 18, no. 2, pp. 17–34, 2023. http://dx.doi.org/10.69864/ijbsam.18-2.174