VALIDATING METRICS OF A MODEL FOR AGILE SOFTWARE DEVELOPMENT TEAM PERFORMANCE (MASTEP)

W. Smit[1*] and M. de Vries[2]

1 University of Pretoria, South Africa | Email address u20494522@tuks.co.za

2 University of Pretoria, South Africa | Email address marne.devries@up.ac.za

ABSTRACT

Maintaining consistent performance across Agile Software Development (ASD) teams is challenging due to the absence of standardised performance metrics, limiting visibility of team performance and impairing management decision-making at scale. This study extends previous work that identified five key metrics within the Model for Agile Software Development Team Performance (MASTeP), by validating their applicability using historical data from a real-world enterprise. Action science research was applied to examine whether these metrics reliably reflect changes in ASD team performance and illustrate their interpretation in pursuit of operational excellence.

Validation results confirmed that velocity, throughput, defect density, cycle time and lead time are affected when ASD team performance changes. Minimal correlation was observed between throughput and defect density, and between work time and velocity. However, strong positive correlations were observed between throughput and velocity, with strong negative correlations between velocity and defect density, work hours and cycle time, throughput and cycle time, velocity and cycle time, throughput and lead time, and velocity and lead time, and these combinations should be used together. Results further indicate that consistent performance improvements are supported when the ratio between committed and

* Corresponding author

completed story points remains between 72% and 88%. Measurement reliability decreases, however, when team dynamics change significantly.

Keywords: Agile software development, team performance metrics, Agile operational excellence, team performance management

1 INTRODUCTION

With continuous innovations in the technology and software industry causing rapid growth, IT companies found that the traditional approaches to project management were not always appropriate as customer requirements often changed during a project [1]. An enterprise facing this challenge is the enterprise that initiated and sponsored this project, referred to throughout as a Large Software Development Company (LaSC). Agile approaches are found to work well in IT companies, where customer requirements change or increase rapidly, and frequent feature releases are needed [1]. One of the most popular Agile approaches often used in industry is Scrum, where developers usually work in small teams on smaller, more manageable tasks than is customary, over a shorter but fixed timeline [2]. While Agile methodologies have been successfully implemented in teams in the LaSC, its management wants to utilise these methodologies effectively to increase productivity and efficiency in teams.

The LaSC, however, lacks effective metrics and baselines to measure productivity and efficiency in Agile Software Development (ASD) teams, which makes it difficult for management to quantify performance improvements. Productivity can be defined as the ratio of the outputs of a system to its inputs, focusing on increasing the outputs [3]. Productivity differs from efficiency as efficiency focuses on reducing input while output remains constant [4]. Determining an exact metric for productivity in an ASD team is a challenging task, since little guidance is provided to companies and management, which makes it difficult to maintain consistent performance across teams [4] - [6]. Team leaders, usually called Agile masters, and management alike are thus interested in the relationship between different metrics, and how this relationship explains problem areas and inform actions needed in a team, as singular metrics give little indication of the causes of trends. The variance in the size and scope of the work of teams also makes it difficult for management to gauge the productivity and efficiency of a team, not necessarily for comparison, but to determine different teams' performances objectively. Management, therefore, wants to use metrics to manage team outputs and gauge whether interventions, such as the processes and structures that have been established, are benefiting the teams.

The problem is that the LaSC does not currently have a standard, cohesive, and validated set of metrics that can be used to quantify the performance of an ASD team, and determine whether a team is productive, efficient and continuously improving. Management currently lacks the knowledge of industry best practices and guidance from literature regarding which ASD metrics would be most practical to measure and track, which makes it difficult to reliably assess and evaluate the performance of ASD teams in the enterprise, and the enterprise as a whole. The lack of standard ASD performance metrics also makes it difficult for management and Agile masters to reliably identify problematic areas in a team, which leads to developers not being incentivised to continuously improve, since there is no reliable standard to measure performance improvements against.

This study is part of a larger research study, the first part of which indicated that the problem instance experienced at the LaSC also exists as a class-of-problems within literature. Section 2 provides additional background for our article, leading to the main research question addressed in this article. The research method applied in this paper is described in Section 3. The data collected and analysed during the demonstration of the solution are then discussed in Section 4. Section 5 gives the final conclusions, and makes recommendations for future research that could expand upon this study.

2 BACKGROUND AND RESEARCH QUESTION

This article presents the second stage of a study, with the first stage having updated a previous systematic literature review in order to understand the current maturity of the body of knowledge of Agile software development team performance metrics [7]. As part of the first stage, we identified the need for developing a Model for Agile Software Development Team Performance (MASTeP), based on the current gaps in the literature. These gaps were identified when we first investigated knowledge areas that might prove useful to develop a solution to the class-of-problems, and assessed the level of maturity of these existing knowledge areas, as discussed in the following section.

2.1 Research gap

One of the main goals in the operations of companies is to operate at optimal capacity, identifying bottlenecks to increase the productivity of development [8]. Teams can gain insight into their performance and improve their efficiency and effectiveness by using metrics [9]. While numerous studies have been done, key problems in the body of knowledge remain, such as a lack of definitions, explanations and formulas for measuring and using metrics [10]. There is also no Agile metric commonality initiative or industry standard [11]. Since there is no single universally applicable metric that can be used to determine the performance of an ASD team, various aspects of a team should be taken into consideration in order to give a holistic overview [8] , [12]. The various ASD metrics used in literature and industry were therefore investigated and classified into a hierarchical structure of five different enterprise levels with increasing order of generalisation, namely individual-, team-, program-, portfolio- and enterprise level. Through updating the previous systematic literature review (SLR), completed by Teiber [8], the 20 most frequently used metrics relating to the team-level were identified . Furthermore, these metrics were standardised, since some metrics were found to be ambiguous and could refer to the same underlying measure.

With the update of the original SLR by Teiber [8], the trends in metrics usage in literature were highlighted, indicating how teams and management have adopted and used metrics. When measuring productivity in a team, the trend in literature showed that velocity and throughput are the two most widely used metrics, with Lines Of Code (LOC) per hour recommended as a secondary productivity measure. When considering efficiency in teams, the usage of cycle time and lead time has shown an upward trend and the widest usage. The trend in metric usage for continuous improvement indicates that the metrics most often used for gauging this aspect in teams is the defect rate, or defect density, of Product Backlog Items (PBIs) in a team. The effect of qualitative aspects on continuous improvement cannot be deterministically defined but is important to gain holistic insights [13]. In order to use the metrics identified to coherently assess ASD team performance, an appropriate metric guidance artefact is needed.

While the concept of metrics and performance measurement is often seen as controversial in ASD, an important aspect of the Agile mindset is continuous improvement based on feedback, which is why measurement is pursued [14]. However, the implementation of metrics is a challenge in itself, since Agile teams are supposed to be autonomous and self-organising, and thus developers often see metrics as not exhibiting trust [15], [16]. The main challenges identified in literature include:

- *A lack of knowledge area depth.* While researchers may describe metrics in studies, details for fully understanding the metrics are often absent, along with no guidelines or standardised and agreed upon basis for constructing a measurement model required to promote trustworthiness [17]. There is also no secondary study that analyses how metrics are used, their impacts, context of usage, and their selection and classification [10].
- *Work requirement estimation difficulties.* Subjective size estimations are often based on team-specific capabilities, yet related metrics are frequently compared across teams in enterprises [18], [19]. The subjectivity of ASD metrics often gives metrics a team-specific characteristic which is not transferable, making fair comparison difficult without a set standard [20].
- *Ill-defined and unvalidated metrics.* The absence of an industry-wide standard or commonality initiative also causes inconsistencies, with researchers and practitioners appearing to add or invent new metrics as needed [11], [21]. Furthermore, few explanations or formulas are provided to aid in fully understanding the metrics, or to clarify different metrics with overlapping terms [17].
- *Data acquisition difficulties*. Poorly defined measurement attributes and fragmented data collection inhibit metric adoption in enterprises [22]. Clarification of operationalising metrics through the defining of measurement intentions, as well as tools

or data sources to use for measurement, must also be improved to reduce fragmentation and increase metric adoption in the enterprise [16], [17], [22].

- *Misalignment between metrics and goals.* A lack of a holistic evaluation approach often causes misalignment and miscommunication of expectations and IT operations business values due to confusion over objectives, wrong goals, and disputes over metrics used [23], [24]. Most misalignments between goals and metrics indicate that goals need to be shifted from technical aspects to business value [25], [26].

As suggested by Beh et al. [27], the findings from the analysis of the class-of-problems indicate that insight into the relationship between different metrics, and how metrics relate to different aspects of ASD and goals, are needed. However, enterprises implementing Agile methodologies often lack both knowledge of which metrics to implement, as well as guidance on how to use different metrics [10].

2.2 The Model of Agile Software-development Team Performance (MASTeP)

The true effectiveness of software development metrics lies in using simple existing metrics to construct management decision-support tools that provide holistic insights into different aspects of the software development lifecycle [28]. By rephrasing the problems noted in the literature in Section 2.1, we identified the need for developing the MASTeP. The MASTeP had to: (1) be based on knowledge rigorously validated in literature; (2) address work requirement and estimation accuracy difficulties; (3) make use of validated and well-defined metrics; (4) make use of metrics where data acquisition is easy; (5) align metrics with the goals in the enterprise [7].

Through extensive deliberation, certain ASD performance metrics were excluded in the MASTeP due to various factors. A metric like estimation accuracy was excluded because it is calculated as the difference in the estimated development time and the actual time to complete an issue. The metric is unreliable when measuring development size using SP, representing size based on complexity, uncertainty and effort, and not on time [9], [29]. Since both the estimated and actual time are subjective, metrics like estimation accuracy and complexity, also frequently used at the team level, lack a set reference point per PBI. Story-point-size estimates can however be standardised within a team using relative estimations to improve the reliability and consistency of size measurements [30]. Estimation accuracy and complexity, while frequently used, are thus weak performance indicators and it is not recommended that they be used for standardisation.

Teiber [8], moreover, notes that the burndown-chart also compares the number of PBIs scoped in a sprint to the PBIs completed to indicate sprint progress, which is just a visualisation of another core metric, namely throughput. A sprint is a short cycle of two to four weeks wherein a team will commit to completing a defined number of tasks. Since defects are also logged as backlog items, time-to-fix defects reflect the time it takes to fix a defect-related PBI, which is essentially the lead time of a PBI. Within the scope of this study, code coverage was also excluded as an ASD team performance metric, since it often relates more to code quality rather than to productivity in a team. While the Net Promoter Score (NPS) is likewise widely used, it was excluded since it relies on the judgement of the customer, with data usually collected via surveys or questionnaires. Therefore, when developing the MASTeP using existing theory, five main metrics were selected: *velocity, defect count, throughput, lead time* and cycle time.

2.3 Research question

Since previous work already developed the MASTeP as a solution artefact [7], this article answers the following research question:

> *How useful and usable is the proposed model of metrics in a practical application, and what real-world evidence demonstrates its effectiveness in quantifying and identifying changes in productivity, efficiency, and continuous improvement in teams?*

3 RESEARCH METHOD

DSR is focused on creating or developing a new, novel artefact or solution that solves practical problems of general interest, based on a strong theoretical background [23], [31]. This focus on a theoretical background means DSR artefacts require high technological rigor, ensuring correct theory and principles to guide artifact creation, rather than usability or applicability to an organisational context, which is necessary in the information systems field for solutions to be successful [32]. The DSR methodology contains a design cycle of six phases: (1) Define a problem, (2) Define objectives of a solution, (3) Design and develop, (4) Demonstrate, (5) Evaluate, and (6) Communicate [32].

This study used the six phases of DSR to develop a model as an artefact, namely the MASTeP. It should be noted that the artefact was developed and shaped by the stakeholder as well as participants from the LaSC. The first stage (that is, the first three DSR phases) was completed within the scope of the larger design-based study [7]. Section 2 provides background on the work undertaken to complete the first three phases of the project. This paper focuses on the 4th phase, namely, the demonstration of the solution artifact. Table 1 provides a summary of the data-gathering methods that were used in completing the *demonstration* phase.

Table 1: Data-gathering during the demonstration phase

Design-based phase	Applied to this study	Data-gathering methods	Participants involved
Demonstrate	Demonstrating the metrics of the main artefact (MASTeP) using LaSC data	Analysing quantitative data of metrics of a team in the LaSC	ASD team in LaSC, Agile Master, primary researcher

4 DEMONSTRATION RESULTS

To evaluate the applicability of the MASTeP and the specific metrics selected, the performance of an ASD team in the LaSC was evaluated. The Agile master and senior manager of the team were able to provide insights into the performance of a specific ASD team and operations of the team. Therefore, to validate the metrics selected for the MASTeP, historical data of the team were used and analysed to validate whether the metrics for the team indicated corresponding changes during the same time fragment that the Agile master and senior manager indicated a definite change in performance. The correlation between the different MASTeP metrics, indicating true changes in team performance, was thus evaluated, as shown in Section 4.1. Since the MASTeP metrics are usually not independent of each other, and changes in one metric could correspond to changes in another metric and provide more depth of insight into the true performance of a team, the correlation and relationship of the individual metrics were evaluated, as shown in Section 4.2.

4.1 Validating correlation between MASTeP metrics with team performance changes

The historical data of a team for the LaSC metrics were obtained from the LaSC management via a report drawn from the JIRA system, which is used to manage development work. The JIRA system automatically collects data of the work done by employees to calculate and monitor various metrics of an ASD team. Since the JIRA system also records historical data on the system, data for one team from 1 January 2024 to 3 June 2025 was downloaded. This study only used and concentrated on the data and metrics of one team, and all employee data of members in the team was anonymised by removing names of the employees from the data. Instead, a unique code was used to distinguish between developers. Thus, the various developers in the team, including their working hours, could be investigated. All the data for the performance metrics in the team were also already aggregated to a team level, again using a unique identifier to maintain the anonymity of employees and objectivity in analysis. The data for the performance metrics in the team were thus analysed from 1 January 2024 to 31 May 2025, which allowed performance metrics in full months to be compared.

A clear change to a team's performance and characteristics can be seen when there is a change in the dynamics of a team, such as when team members leave or new members join [12]. The management of the LaSC keeps clear employment records, including when new people join and leave teams, as shown in Figure 1. For the purposes of this study, the researcher refers to the team as Agile Team 1 (AT1).

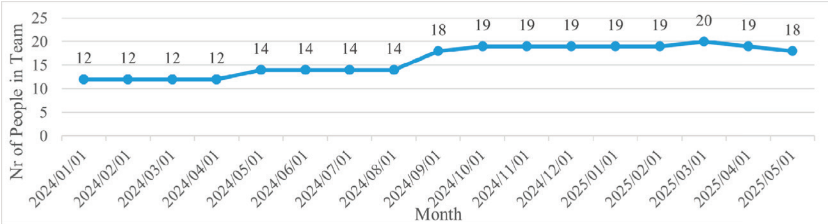

Figure 1: Number of people in the sprint

AT1 had two people who joined the team in May 2024, four people who joined the team in September 2024, one person who joined in October 2024, and one person who joined in March 2025. Furthermore, one person left the team in April 2025 and another person left the team in May 2025. To provide more clarity on the working trends and leave patterns in the team, the number of hours worked in the team is shown in Figure 2. Since the LaSC is in South Africa, employees will usually take leave over the summer holidays in December, which is evident from a sharp drop in the total hours worked by the team over December 2024. This information on the actual working time of team members adds insights into the metrics of the team and can help to inform some of the fluctuations seen in the metrics.

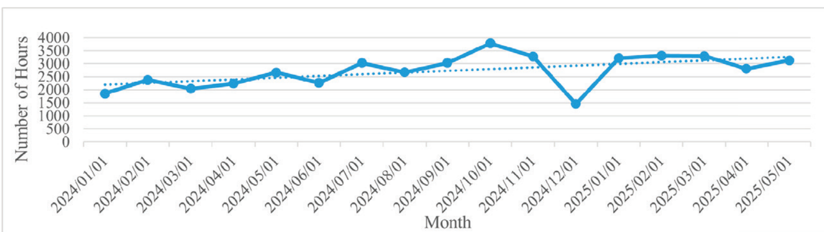

Figure 2: Total hours worked by team members in AT1 per month

Team change instances in time were of key interest when analysing the performance of the team and the data for the metrics selected for the MASTeP, as the changes in team members, either joining or leaving, should also reflect changes in the metrics being monitored. One of the metrics in the MASTeP that indicates changes in performance in AT1 is the throughput of the team over time, which measures the number of Product Backlog

Items (PBIs) completed in a sprint. The throughput of AT1 per month is shown in Figure 3, with the dotted vertical lines indicating the dates on which team members joined or left the team, thus causing a change in the dynamics of the team. In the months of February to April 2024 the Agile master indicated that the team had multiple large user stories to complete, and while the team performance was high, the Agile master believed that adding another person to the team would alleviate some pressure on the team. Two members were therefore added to the team in May 2024, with the performance of the team indicating improvement, as the team started to close more issues consistently. This can be seen in the throughput metric below, showing that the throughput (measured as PBIs completed) of AT1 increased from 29 in April 2024 to 40 in May 2024.

Figure 3: Throughput of AT1 per month

The increase in performance of AT1 represented in the throughput changes, closing over 50 PBIs per month until the end of August 2024, also aligned with the perceived improved performance noted by the Agile master over the same period after the change in number of team members. However, the Agile master noted that change in team dynamics when four new team members joined in September 2024 negatively affected the team performance initially, since many of the new members had to be onboarded and trained by the longstanding team members in the initial month. The throughput of AT1 reflected this negative change as it decreased to 38 PBIs, that is, below the previous norm of 50 PBIs closed per month. When the new team members were onboarded, the Agile Master reported that the team performance increased again in October 2024, but not significantly, because another member was added to the team in the same month.

Nevertheless, even with all team members on board, the performance of the team decreased again, in December 2024 and January 2025, which is when many team members took leave. This is evident from the few hours worked in December 2024, as shown on Figure 2. This decrease in productivity is also evident in the throughput of the team, which indicates a decrease in the number of PBIs completed in December 2024 and January 2025. Another decline in team performance, perceived by the Agile Master, was when some of the team members left the team in April 2025 and May 2025. This decline in performance is also reflected in the throughput of the team, which decreased in April and May 2025, with May, in particular, being much lower than usual, at 40 PBIs closed.

Thus, when monitoring the changes in team dynamics in the data, it was observed that throughput does reflect changes in team dynamics to a certain extent. The correlation of these metrics, however, does not imply causation, as a standard formula could not be derived to describe a direct relationship between the number of people in a team and throughput. The aim of this analysis was simply to confirm the observation that any changes in the dynamics of a team, such as a change in the number of people, affect the throughput in a team abnormally. However, when the team dynamics are stable, the changes and increases in throughput usually correspond to actual performance increases in the team. Throughput alone, however, cannot fully capture team performance as the other metrics included in the MASTeP, like velocity and defect density, also provide insights into various aspects of ASD team performance.

When monitoring the velocity of AT1, changes in team dynamics are also clearly reflected, as can be seen in Figure 4. As in Figure 3, the different vertical lines indicate changes in AT1 dynamics, such as people joining or leaving the team. The team's estimated story points of PBIs were collectively agreed upon during sprint planning meetings, using relative story-point estimation. When new members were added to the team, the standard relative estimation size was not updated, unless a new type of issue needed to be worked on in a sprint. This was done to maintain the consistency of story point estimations.

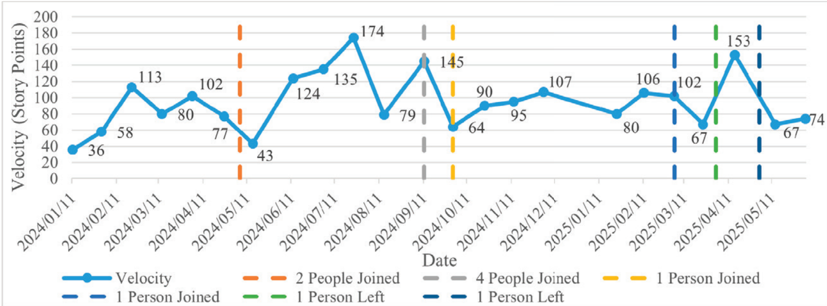

Figure 4: Sprint Velocity of AT1

The velocity of the team decreased significantly in Sprint 7, which ended on 15 May 2024. This coincides with the date on which the new team members were onboarded, namely 6 May 2024. During onboarding, some of the older team members had to take time away from their work to help the new members settle in, which thus resulted in a drop in performance and productivity. For the following sprint, however, the team members had been onboarded, taking on tasks and contributing to the team sprint, with the result that there was a significant increase in the perceived performance of the team. This is also visible with a significant increase in the velocity of the team in Sprint 8, ending on 12 June 2024. With the two new team members contributing to the work in the team, the following two sprints had better performance, with the team also having a high velocity.

The relatively lower velocity than usual in October 2024 could also have resulted from the four new team members being onboarded over that period. The lower-than-expected velocity in December and January is also likely due to fewer hours worked by the team overall, with many team members taking leave in the period. In the sprint ending 24 March 2025, another person had to be onboarded and trained, which again led to a lower team performance, with a decline in velocity also evident. The team dynamics did not change at the end of March 2025, correlating with an increase in velocity to 153 story points per sprint. When team dynamics changed again and team members left AT1 in April and May 2025, the negative impact on the performance of the team was evident in the velocity of the sprints in April and May 2025. Nevertheless, as with throughput, this evidence in the data is not implying causation, but rather correlation between velocity and change in the team dynamics. The correlation of velocity with perceived improvements or declines in team performance also validates the applicability for including velocity as a metric for ASD team performance.

An anomalous sprint seen in the velocity data is Sprint 12, which ended on 14 August 2024 with 79 story points completed, indicating a sharp decrease from normal. The Agile master indicated that there was no drastic change in perceived team performance, and thus additional data on other metrics around this time were needed. The defect density for each sprint was thus analysed, which is shown in Figure 5, with the changes in the team dynamics also indicated.

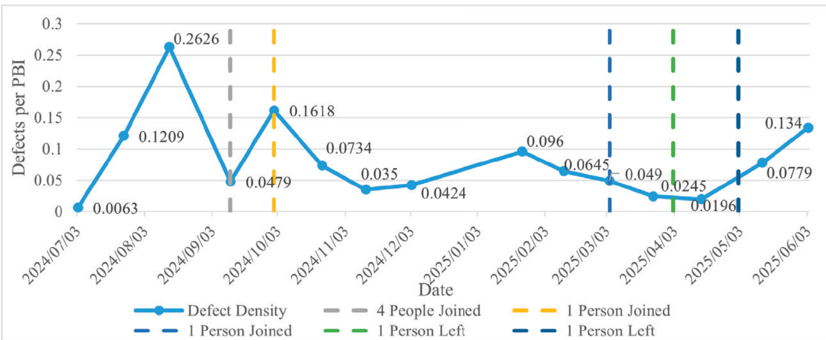

Figure 5: Defect density for AT1 per sprint

The defect density for AT1 provided clarity on the sharp decrease in velocity in Figure 4 in the sprint ending 14 August 2024, shown in Figure 5. This particular sprint had the highest defect density of all the sprints. This high defect density indicates that, during Sprint 9, the team likely had to work with new processes to fix defects and meet the original customer requirements. Having to follow the new process to log defects and needing to address more defects than normal could likely cause a decrease in velocity. While the team performance may be perceived to be high, fixing previous mistakes is not true productivity, as fixing mistakes can be seen as rework, which is often classified as waste in a production system.

The defect density also helped to monitor the performance of AT1, as the changes in the characteristics and dynamics of the team also caused changes in the defect density of the team. When four new developers were added in September 2024, they only started contributing to the codebase of the team around 1 Oct 2024, which was when the defect density initially increased. This is likely because the new team members still had to adjust and learn the new practices and code base. The defect density also stayed constant and relatively low during the period where there was little change in the team dynamics from November 2024 to March 2025, only increasing slightly in January as many team members were on leave in February, and thus it is likely that fewer people were available to check and test code. The final increase in defect density was most likely the result of team members having to take over the projects of those team members who had left in April and May 2025, and needing to familiarise themselves with unfamiliar code, leading to some errors being introduced.

To further measure how well team members were actioning and closing tasks, the cycle time and lead time of PBIs, which are also core metrics in the MASTeP, were investigated. The average cycle time per PBI is shown in Figure 6, where cycle time is the time in days since work has started on a customer PBI.

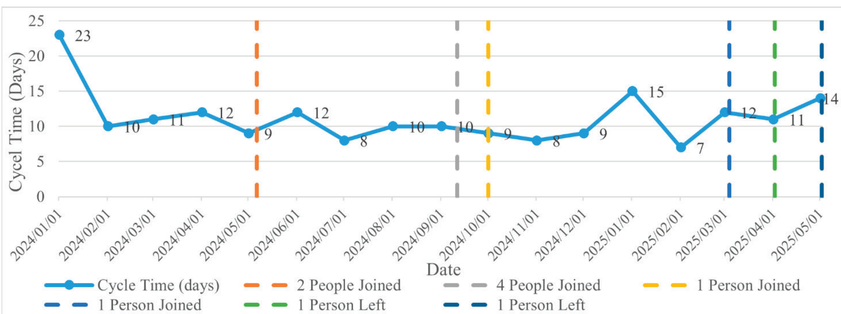

Figure 6: Average PBI cycle time per month for AT1

The cycle time also shows changes in the same period as other changes in team dynamics. The cycle time increased around the onboarding periods of the team members, in June 2024, September 2024, and March 2025, and then decreased thereafter, in July 2024, October 2024, and November 2024, indicating that the new team members made an impact on and contribution to the team. The large increase in cycle time in January 2024 and January 2025, with an unwanted increase, that is, PBIs took longer to complete, also reflected the change in team dynamics as a result of the annual leave-taking over December. It appears that the cycle times lag behind changes in team dynamics by a month.

Changes in lead time also provided insights into the impact that changes to team dynamics have, as evident from the lead time data for AT1 shown in Figure 7. The lead time reflects how long a customer would have to wait for a feature request to be completed, calculating from the date it was first added to the backlog of the team, up to the completion date. While not as pronounced, the lead time of AT1 reflected how well the team balanced new customer requests with old customer requests, with the lead time of the team tending to be higher when older PBIs were closed, and lower when newer PBIs were closed. This indicates how well a team manages the scope of work in the team.

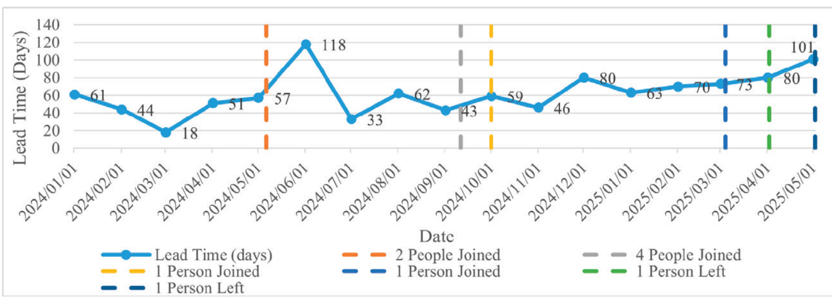

Figure 7: Average PBI lead time for AT1

In AT1, changes to the team dynamic when new members were added to the team generally benefited the team over time. During the period of new members being added to the team (two new team members joining in May 2024 and four joining in September 2024), AT1 closed more of the older PBIs, as indicated by a decrease in lead time in the following month, and thus improved performance on resolving PBIs timeously.

The increase in lead time in December 2024 was also expected as that month was classified as a feature freeze month, meaning that no new features or changes to software operations could be released, except for major problems or bugs in the system. During this period, team members who did not take leave usually worked on issues that were not prioritised throughout the year, such as older and smaller PBIs and requests. The effect of this can be seen by a significant increase in the team's lead time for December, which correctly represents the true performance and actions of the team. The increase in lead time in February did, however, correspond with the slight decrease in performance at the beginning of the year, related to the new member being onboarded in March 2025 and team members leaving in April 2025 and May 2025. Lead time thus accurately represents the performance of a team. While a distinct change in lead time could be due to a change in the sprint planning of the team, it could also be due to a change in the dynamics of the team, resulting in the team following a different structure to close customer requests and PBIs.

When investigating two prominent points in the lead time of AT1, further discussion is needed. In March 2024, the lead time of the team was quite low, which would usually be desirable behaviour if it did not correspond with a throughput and velocity decrease in the team. This could indicate that the team may have been struggling to complete all issues in the sprint, and that,therefore, more help was needed in the team. The dramatic increase in lead time in June 2024 also indicates that many issues that had been requested by customers a long time before had been closed. This change indicates that the increase in team performance, with two new members added in May 2024, led to an increase in older issues being closed, with the velocity and throughput of the team also increasing significantly during this time. Since the comparison of these metrics adds depth to the discussion, the correlation between different metrics is investigated further in the next section.

4.2 Validating correlation between individual MASTeP metrics and interpretation guidance

The MASTeP metrics were used to indicate whether an apparent improvement in team performance was a true reflection of a team's prowess, and, furthermore, to show the effects of changes in team dynamics on team performance. In addition, the possible existence of relationships between metrics needed to be taken into consideration to enhance the insights gained from the metrics. To reliably validate the existence of relationships between metrics, consistent and isolate time slots needed to be identified, during which team dynamics remained unchanged for a substantial period of time. This would ensure valid comparison and evaluation of the metrics, because changes in team dynamics can affect the performance of a team and thus also affect the metrics of a team, as indicated by Katikireddy and Veereddy [12]. To ensure a valid comparison, the performance of the team, assessed according to the relationship between metrics, was only evaluated during times when no external changes were reported, as external changes in the team dynamics could have had undue influence on the metrics being compared, skewing the results.

The time slot into which these metrics were divided was January 2024 to April 2024, when the team consistently had 12 people in the team, referred to as Period 1. The second period selected was from May 2024 to August 2024, which is directly after two additional team members joined and the team consistently had 14 people, referred to as Period 2. The third period selected because of limited change in team dynamics is referred to as Period 3, and is the period fromOctober 2024 to February 2025, when, despite a number of team members being on leave, there were, nevertheless, limited changes in the team dynamics. The relationship between the MASTeP metrics monitored during these times and the work hours in the team as well as other metrics are further discussed in the sections that follow. The aspects of correlation and the nature of the relationships between metrics is the main focus in the evaluation. The intention, therefore, is not to show causation, but rather to validate the relationships.

4.2.1 Velocity in practice

The evidence to indicate the existence of a possible relationship between velocity and hours worked was investigated for Period 1, shown in Figure 8, with a large positive correlation evident between the number of hours worked in the team and the velocity of the team. The trend indicates that as work hours increase, the velocity of the team also increases. The correlation coefficient is a statistical measure used to indicate the degree of association of two variables, with 1 being the maximum positive correlation, -1 being the maximum negative correlation, and 0 indicating no association between the two variables [33]. The correlation coefficient r between velocity and *hours worked* for Period 1 is r = 0.852, which indicates a strong positive relationship, suggesting that the more worked hours in a sprint, the greater the increase in the number of story points completed.

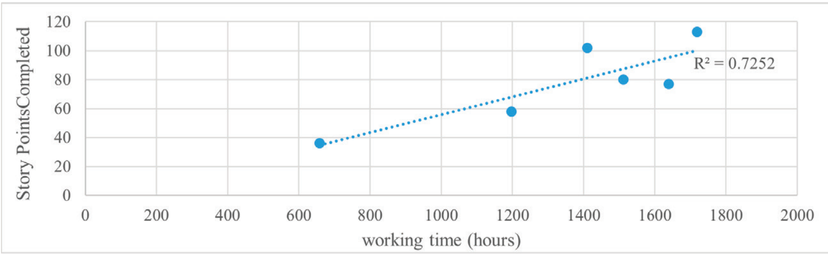

Figure 8: Velocity vs work hours for AT1 from January to April 2024

The regression test is another statistical measure that analyses the relationship between metrics and can be used to indicate the predictability of the variables being measured [34]. This value is represented by in Figure 8 and is the square of the correlation coefficient, with = 0.725 for Period 1. A regression coefficient of = 0.725 is, however, not seen as reliable enough for predictability as often only values greater than 0.9 are seen as reliable enough for predictability [34]. Therefore, while velocity and working hours have a strong correlation, story points estimated for a team are not reliable enough to be used to predict the working hours for a team. As suggested in the MASTeP, story points should thus be decided upon by a team in the sprint planning for predictability and planning, rather than used as a time estimate derived for the team. When evaluating team performance, the plan should be measured against the actual output of the team. This can be done by comparing the number of story point (SP) commitments by the team, that is, SPs planned for completion, against the number of SPs completed by the team. The comparison is shown in Figure 9, demonstrating team performance according to the available data.

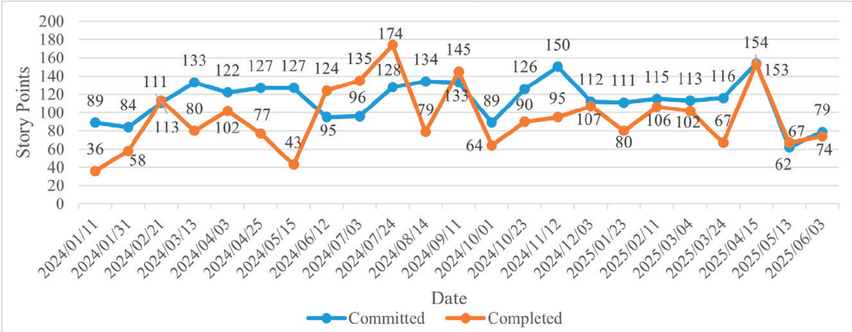

Figure 9: AT1 committed story points and completed story points over time

When comparing the completed and committed SPs of AT1, the goals of the team remained relatively consistent. While the team struggled to meet the sprint goals at the beginning of the year, once the new members were added to the team, AT1 began to meet its goals more consistently. The team also maintained a balance between performance improvements, such as taking risks and striving to complete more tasks, and being cautious and not overcommitting to story points. This behaviour is evident when looking at the target velocity, or goal, of the team and actual velocity of the team, as the team had an average sprint commitment reliability of 83%. This means that 83% of the SPs committed to in a sprint were completed in the sprint, which aligns well with the guidelines of Downey and Sutherland [35], which recommend that estimation accuracy, or commitment reliability in this case, should be maintained between 72% to 88%. Commitment reliability refers to the success rate of completion of issues scoped for a sprint. An accuracy exceeding 88% suggests risk aversions in a team, with accuracy below 72% indicating poor understanding of requirements. To further analyse the reliability of the team's task completion, the throughput of a team should be monitored in relation to the other metrics.

4.2.2 Throughput in practice

The throughput of a team helps to monitor the effectiveness of the team's actions in a sprint, with throughput only increasing if a PBI is completed and closed, no matter the size. When comparing the velocity of a team and

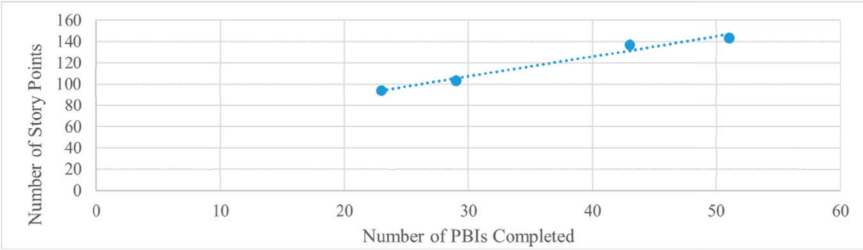

Figure 10: AT1 velocity over throughput per month in Period 1

When analysing the correlation between throughput and velocity, the correlation coefficient is shown to be $r = 0.987$, which is a very strong positive correlation. This means that there is a strong relationship and linking between these two metrics, which is to be expected since each PBI closed has a number of story points assigned to the issue, and thus if more PBIs are closed, more story points will be completed in the sprint. The average size of PBIs for AT1 is 2.67 story points per issue. This indicates that the team is more comfortable with addressing many smaller issues that are clearly defined, rather than a larger issue with many unknowns. This helps to inform how to best help the team address other problems in ASD, such as defects that occur in operations.

4.2.3 Defect density in operations

When monitoring the defect density of a team, insights can be gained as to how effectively a team develops new code without sacrificing quality of effort. Since defect data is only recorded for AT1 from July 2025, Period 3 was analysed to compare the relationship between velocity and defect density, with the data plotted in Figure 11.

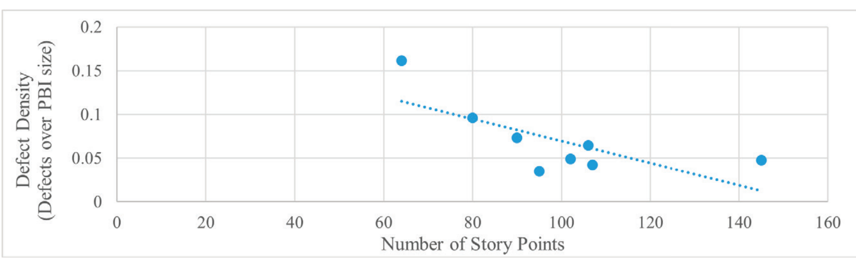

Figure 11: AT1 defect density over velocity for Period 3

The data indicates a strong negative correlation between the defect density and the velocity of the team, with the correlation coefficient r = - 0.718. This negative correlation indicates that as the velocity of a team increases, the defect density of the team will also decrease. This is understandable, as defect density is calculated by the number of defects over the size of the PBI measured in story points, and thus if an increasing number of issues are being developed in a sprint with the sprint backlog thus increasing, the defect density denominator will greatly decrease in size.

When considering the data for Period 3, the relationship between throughput and defect density, the opposite association can be seen by the trendline in Figure 12. As with velocity, there appears to be a negative correlation between throughput and defect density in operations, with the correlation coefficient being weak and negative at r = -0.22. This then shows that there is a weak relationship between throughput and defect density and that if one metric increases, such as throughput, it does not necessarily mean that more defects will be produced during development.

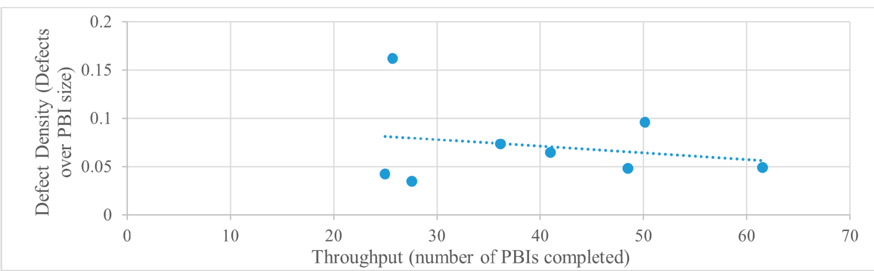

Figure 12: AT1 defect density vs throughput for Period 3

Since there is a slight negative correlation, it will be fair to say that when driving a performance increase in a team and focusing on improving productivity in code, it does not mean that more defects will naturally be produced. As the work requirements in the team increase, along with the backlog size, defect density helps to show how effectively the team is addressing issues and closing PBIs without causing more defects, even when work requirements increase. Therefore, a decrease in the defect density shows continuous improvement in the team, as, with fewer defects being produced in relation to the number of the other PBIs needing to be addressed in development, the team usually closes more issues with fewer defects. This continuous improvement, consequently, leads to an increase in the efficiency of the team.

4.2.4 Cycle time and lead time in operations

When monitoring the relationship between cycle time and other performance aspects in a team, an improvement in the cycle time of a team (with improvements being a decrease in the cycle time of issues) should generally indicate improved performance as the improvements come as a result of issues assigned to the team being addressed faster. Therefore if a team puts in more effort and works harder to improve the performance of the team, it is expected that the cycle time should also decrease. This relationship is evident when comparing the impact of the number of hours worked by the team on the cycle time of AT1, there is a clear positive trend in Period 2, as can be seen in Figure 13.

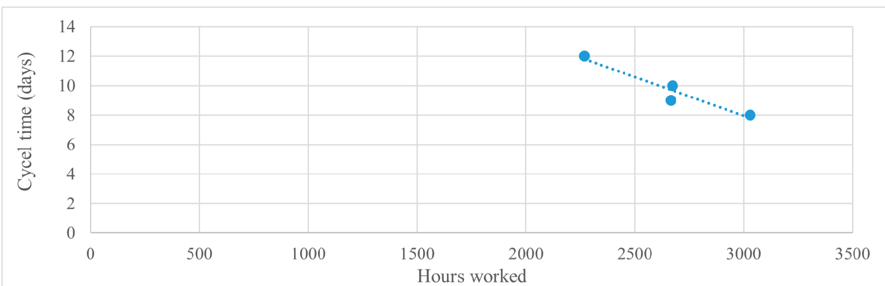

Figure 13: AT1 Cycle time and work hours per month for Period 2

The correlation coefficient for Period 1 and Period 2 indicates a strong negative correlation between the number of hours worked in the team and the decrease in cycle time, with *r1* = -0.826 and *r2* = -0.959 respectively. This indicates that when the team works more hours, the performance in the team improves. This is in line with the assertion by the Agile master that when the perceived performance of the team improves, the efficiency of the team also improves, and thus PBIs will be closed faster.

While AT1 improved in efficiency in closing PBIs faster, a true improvement in performance can only be claimed when considering the productivity increase in the team during the same time. When considering

Period 1, the velocity of the team is compared with the cycle time of the team, as shown in Figure 14. The correlation between velocity improvement and a corresponding decrease in cycle time gives validity to the observation that there was indeed a performance increase in Period 1.

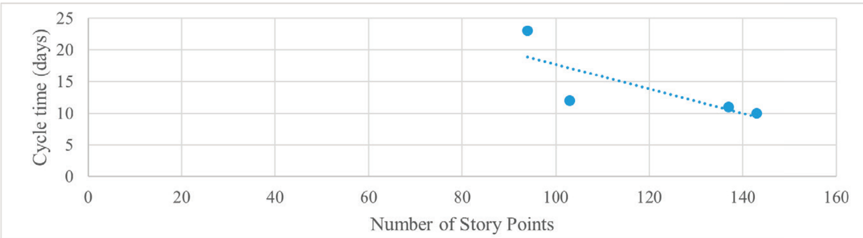

Figure 14: AT1 story points and cycle time per month for Period 1

Period 2 and Period 3 also had negative correlations between velocity and cycle time, with $r2 = -0.179$, and $r3 = -0.572$. Thus, while a weak correlation is visible, an increase in velocity should lead to a decrease in *cycle time* in a team if its performance has improved. As shown in Figure 15, this relationship is true for throughput and cycle time for *Period 1*.

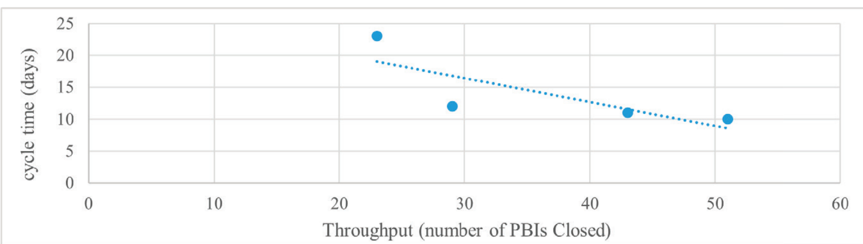

Figure 15: AT1 throughput and cycle time per month for Period 1

A strong negative correlation exists between throughput and cycle time, with $r = -0.792$, which indicates that, like velocity, as the number of PBIs closed in the month increases, there is a corresponding decrease in cycle time for the issues. This would seem to represent reality well, as to increase the number of PBIs closed in a month, PBIs will need to be worked on and closed faster. Likewise, this negative correlation is present in AT1 in Period 2 and Period 3, with $r2 = -0.356$, and $r3 = -0.656$, which indicates that an improvement in performance in the other sprints, which results in more PBIs being closed, should lead to a decrease in the cycle time.

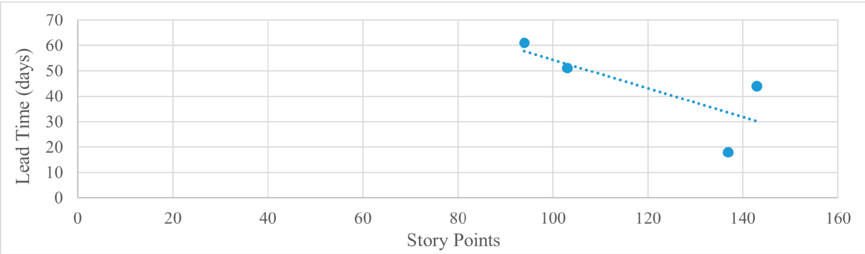

Figure 16: AT1 lead time and velocity for Period 1

The same negative correlation between cycle time and velocity is also true for lead time and velocity, as shown in Figure 16. As can be seen, there is a strong negative relationship between the two metrics, with $r1 = -0.748$, which indicates that improvement in velocity should result in a decrease in the lead time of PBIs, and that customer requests should also decrease. If the team is, therefore, improving in efficiency and performance, customer requests should be addressed more promptly, with an increase in the number of story points completed, and also with a corresponding decrease in the lead time on PBIs.

5 CONCLUSION, LIMITATIONS AND RECOMMENDATIONS

In the initial stage of this study five key metrics were identified from literature to construct the Model for Agile Software Development Team Performance (MASTeP), namely velocity, throughput, defect density, cycle time, and lead time. Using historical operational data from an active team, this study validated that each metric represents a distinct aspect of ASD team performance and therefore justified their inclusion in the model. By analysing the changes in these five metrics over periods with significant changes in team dynamics which would have a clear impact on the performance of the team, such as people joining or leaving the team, each of the metrics correctly showed significant corresponding shifts.

This demonstrated that the metrics included in the MASTeP effectively monitors ASD team performance and can indicate whether a change in a team had a positive or negative effect on the team performance.

The validation of relationships and correlation between these metrics in this study also helped to provide clarification of the dynamics of the performance of an ASD team. Therefore, by standardising velocity, throughput, defect density, cycle time and lead time as performance metrics in an ASD team, the MASTeP provides insight into all aspects of team performance, and the different characteristics and facets that could affect team performance. The findings of this study are in line with Downey and Sutherland [35], namely that the success rate of completing issues committed to by the team in a sprint, or commitment reliability in this case, should be between 72% and 88%.

One of the limitations of this study was that the leave patterns and data of individual team members were not available. While it is normal for most South African employees to take a break over December, the exact data was not available in this instance. Further investigation into the period of leave of certain team members would help to explain or attribute specific changes in team performance. To keep the participants anonymous, specific investigation into the skill level and years of experience of each team member was not done. Further investigation into this aspect would benefit the explainability of performance changes in the team, but this data were not available at the time. The estimation of story points was kept consistent by using relative estimation, and referencing a set standard for the size and type of user story set by the team. While this improves the consistency of estimations, adjustment of user story points during a sprint was not allowed, in order to keep story points consistent with the set standard. Further investigation into the accuracy of this set reference standard, while not undertaken in this study, would improve the accuracy of claimed performance improvements and prevent distorted performance data due to scope creep. Future investigation into such a standard would also benefit the model by suggesting when adjustments to references need to be made. In addition, greater validity of findings could be gained if the study is extended to multiple teams, across different software development areas, over several years, and within different companies in the future. This was, however, not possible within the scope and time constraints of this study, but is strongly recommended for future work.

The real-world data gathered in the study validate the inclusion of these metrics for the MASTeP, which demonstrated how different metrics should be interpreted, and highlighted the effects of changing team dynamics. This study, however, only used data collected from one team in one company, with input provided by LaSC members. Additional real-world cases, where the five key metrics are used, would increase the generalisability of the artefact. A second data set, from a different enterprise, would also increase the credibility of the interpretation guidance and findings in the study. The LaSC requested a set of metrics that could be used to compare the performances of teams. This study validated the metrics related to a single team. In a further study, the data of two teams could be extracted, and the validity of the metrics and their interpretation evaluated when comparing the performance of the teams.

6 REFERENCES

[1] A. M. Dima and M. A. Maassen, "From Waterfall to Agile software: Development models in the IT sector, 2006 to 2018. Impacts on company management," *Journal of International Studies* , vol. 11, no. 2, 2018, doi: 10.14254/2071-8330.2018/11-2/21.

[2] T. Dybå and T. Dingsøyr, "Empirical studies of Agile software development: A systematic review," *Information and Software Technology,* vol. 50, no. 9-10, pp. 833-859, 2008, doi: 10.1016/j.infsof.2008.01.006.

[3] E. Koss and D. A. Lewis, "Productivity or efficiency—Measuring what we really want," *National Productivity Review,* vol. 12, no. 2, pp. 273-284, 1993, doi: 10.1002/npr.4040120212.

[4] D. Smit, H. Smuts, P. Louw, J. Pielmeier, and C. Eidelloth, "The impact of GitHub Copilot on developer productivity from a software engineering body of knowledge perspective," 2024. [Online]. Available: https://aisel.aisnet.org/amcis2024/ai_aa/ai_aa/10. [Accessed: 25 Aug 2025].

[5] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale Agile transformations: A systematic literature review," *Journal of Systems and Software,* vol. 119, pp. 87-108, 2016, doi: 10.1016/j.jss.2016.06.013.

[6] H. Edison, X. Wang, and K. Conboy, "Comparing methods for large-scale Agile software development: A systematic literature review," *IEEE Transactions on Software Engineering,* vol. 48, no. 8, pp. 2709-2731, Aug 2022, doi: 10.1109/TSE.2021.3069039.

[7] W. Smit and M. de Vries, "Standardising metrics to construct a Model for Agile Software development Team-performance (MASTeP)," *Authorea,* Aug. 2025, doi: 10.22541/au.175647873.33322547/v1.

[8] N. Teiber, "Goals and KPIs in large-scale Agile development: A systematic literature review," M.A. thesis, Technical University of Munich, Munich, 2021. [Online]. Available: https://www.cs.cit.tum.de/sebis/student-theses-guided-research/completed-theses-guided-research/2021/masterarbeit-nadine-teiber/ [Accessed: 31 July 2025].

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

[10] S. Leal, J. C. R. Hauck, G. Vieira, and M. Bertan, "How Agile organizations use metrics: A systematic literature mapping," in *Proceedings of the XXI Brazilian Symposium on Software Quality (SBQS '22)*, 2022, pp. 1-11, doi: 10.1145/3571473.3571479.

[11] M. Maddox and S. Walker, "Agile software quality metrics," in *2021 IEEE MetroCon*, Hurst, TX, 2021, pp. 1-3, doi: 10.1109/MetroCon54219.2021.9666049.

[12] N. D. L. P. Katikireddy and N. Veereddy, "LeAgile measurement and metrics: A systematic literature review and case study," M.A. thesis, Blekinge Institute of Technology, Karlskrona, Sweden, 2017. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1153569&dswid=-6908 [Accessed: 13 Aug 2025]

[13] 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, Sept. 2021, doi: 10.1504/IJASM.2021.118061.

[14] H. Looks, J. Fangmann, J. Thomaschewski, M.-J. Escalona, and E.-M. Schön, "Towards a standardized questionnaire for measuring agility at team level," in *International Conference on Agile Software Development*, 2021, pp. 71-85. [Online]. Available: https://library.oapen.org/bitstream/handle/20.500.12657/49499/1/9783030780982.pdf#page=80 [Accessed: 28 Aug 2025].

[15] T. Visser and J. Hulstijn, "Measuring Agile/DevOps team performance," in *Agil-ISE 23: 2nd Intl. Workshop on Agile Methods for Information Systems Engineering*, 2023, pp. 17-23. [Online]. Available: https://ceur-ws.org/Vol-3414/paper-3.pdf [Accessed: 14 Aug 2026].

[16] P. Philipp, F. Tobisch, and F. Matthes, "Investigating the adoption of metrics in large-scale agile software development," in *Pacific Asia Conference on Information Systems*, 2022. [Online]. Available: https://www.cs.cit.tum.de/sebis/publications/2022/ph22d/ [Accessed: 14 Aug 2025].

[17] L. López *et al.*, "Quality measurement in Agile and rapid software

development: A systematic mapping," *Journal of Systems and Software,* vol. 186, 2022, doi: 10.1016/j.jss.2021.111187.

[18] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in Agile software development: a systematic literature review," in *Proceedings of the 10th International Conference on Predictive Models in Software Engineering,* 2014, pp. 82-91, doi: 10.1145/2639490.2639503.

[19] J. Pinter, A. Lee, and J. Hall, "New metrics for evaluating project progress and team contribution using Agile methods," in *Proceedings of the International Annual Conference of the American Society for Engineering Management.,* 2023, pp. 1-10. [Online]. Available: https://www.proquest.com/conference-papers-proceedings/new-metrics-evaluating-project-progress-team/docview/3082336007/se-2?accountid=14717 [Accessed: 25 Aug 2025].

[20] S. Trudel and L. Buglione, "Guideline for sizing Agile projects with COSMIC," *Proceedings of the IWSM-MetriKon-Mensura, Stuttgart, Germany,* 2010.

[21] E. Kupiainen, M. V. Mäntylä, and J. Itkonen, "Using metrics in Agile and Lean Software Development–A systematic literature review of industrial studies," *Information and software technology,* vol. 62, pp. 143-163, 2015, doi: 10.1016/j.infsof.2015.02.005.

[22] P. Philipp, F. Tobisch, and F. Matthes, "Challenges and success factors for metrics in large-scale Agile development," in *AMCIS 2022 Proceedings,* 2022. [Online]. Available: https://www.cs.cit.tum.de/sebis/publications/2022/ph22b/ [Accessed: 25 Aug 2025].

[23] T. Held and M. Westner, "IT business value measurement and communication among German CIOs: a conceptual framework," in *2022 IEEE 24th Conference on Business Informatics (CBI),* 2022, pp. 146-155, doi: 10.1109/CBI54897.2022.00023.

[24] N. B. Lindström, D. Koutsikouri, M. Staron, W. Meding, and O. Söder, "Understanding metrics team-stakeholder communication in Agile metrics service delivery," in *2021 28th Asia-Pacific Software Engineering Conference (APSEC),* 2021, pp. 401-409, doi: 10.1109/APSEC53868.2021.00047.

[25] I. Korpivaara, T. Tuunanen, and V. Seppänen, "Performance measurement in scaled Agile organizations," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, University of Hawai'i at Manoa, 2021, doi: 10.24251/HICSS.2021.830.
[26] Z. Zhao, S. X. Deng, Y. Ma, Q. Gao, and N. Zhao, "Software project measurement based on the 5P model," *Soft Computing*, vol. 28, pp. 2083-2105, 2024, doi: 10.1007/s00500-023-09175-9.
[27] H. C. Beh, Y. Y. Jusoh, R. Abdullah, and S. Hassan, "Dimensions in measuring performance of Agile software development projects: A literature review," in *2022 Applied Informatics International Conference (AiIC)*, 2022, pp. 83-87, doi: 10.1109/AiIC54368.2022.9914025.
[28] N. E. Fenton and M. Neil, "Software metrics: roadmap," in *Proceedings of the Conference on the Future of Software Engineering*, 2000, pp. 357-370, doi: 10.1145/336512.336588.
[29] A. Trendowicz and R. Jeffery, Software Project Effort Estimation: *Foundations and Best Practice Guidelines for Success, Constructive Cost Model* Cham: Springer, 2014, doi: 10.1007/978-3-319-03629-8.
[30] M. Cohn, *Agile Estimating and Planning*. Hoboken, NJ: Pearson Education, 2005.
[31] J. A. Hoogervorst, *Practicing Enterprise Governance and Enterprise Engineering*. Cham: Springer, 2018, doi: 10.1007/978-3-319-73658-7.
[32] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77, 2007, doi: 10.2753/MIS0742-1222240302.
[33] R. Taylor, "Interpretation of the correlation coefficient: a basic review," *Journal of Diagnostic Medical Sonography*, vol. 6, no. 1, pp. 35-39, 1990, doi: 10.1177/875647939000600106.
[34] A. G. Asuero, A. Sayago, and A. G. González, "The correlation coefficient: An overview," *Critical Reviews in Analytical Chemistry*, vol. 36, no. 1, pp. 41-59, 2006, doi: 10.1080/10408340500526766.
[35] S. Downey and J. Sutherland, "Scrum metrics for hyperproductive teams: how they fly like fighter aircraft," in *2013 46th Hawaii International Conference on System Sciences*, 2013, pp. 4870-4878, doi: 10.1109/HICSS.2013.471.