

---

## Performance metrics in scrum software engineering companies

---

Fernando Almeida\*

University of Porto, INESC TEC,  
Faculty of Engineering,  
Rua Dr. Roberto Frias, s/n,  
4200-465, Porto, Portugal  
Email: almd@fe.up.pt  
\*Corresponding author

Pedro Carneiro

Polytechnic Higher Institute of Gaya,  
School of Science and Technology,  
Av. dos Descobrimentos, 333,  
4400-103, V.N. Gaia, Portugal  
Email: pcarneiro@ispgaya.pt

**Abstract:** Finding effective ways to evaluate the process and a team in software engineering is not a trivial task. Therefore, it is pertinent to study metrics that can be used in Scrum environments to monitor and evaluate the team's progress and support the implementation of improvements. This study explores the relevance of 12 specific metrics applied in a Scrum environment and explores their importance considering multiple dimensions through the adoption of a quantitative methodology based on a survey that received 137 valid answers from Portuguese Scrum professionals. The results allowed us to conclude that the metrics related to the business value delivered and sprint goal success are the most relevant. Furthermore, factors like the number of years of experience of individuals, their role in the Scrum team, and the size of the organisation are factors that influence the perception of the importance of these metrics.

**Keywords:** scrum; agile; metrics; measurements; software development; software quality.

**Reference** to this paper should be made as follows: Almeida, F. and Carneiro, P. (2021) 'Performance metrics in scrum software engineering companies', *Int. J. Agile Systems and Management*, Vol. 14, No. 2, pp.205–223.

**Biographical notes:** Fernando Almeida has a PhD in Computer Science Engineering from the Faculty of Engineering of University of Porto (FEUP). He also holds a MSc in Innovation and Entrepreneurship and an MSc in Informatics Engineering from FEUP. He has around 10 years of teaching experience at higher education levels in the field of computer science and management. He has also worked for 15 years in several positions as a software engineer and project manager for large organisations and research centres like Critical Software, CICA/SEF, INESC TEC, and ISR Porto. During that time,

he had the chance to work in partnership with big international organisations and universities in several European projects. His current research areas include innovation policies, entrepreneurship, and decision support systems.

Pedro Carneiro has a MSc in New Technologies from the Autonomous University of Barcelona and a BSc in Computer Science from ISPGAYA. He is a Certified Scrum Master from Scrum Alliance since 2016. He has around 20 years of professional experience in computer engineering and higher education. He is a founding partner at WeMake and WeSecure. He is currently a Privacy Manager (CIPM) where he belongs to the forensic scholarship for computer experts in the field of secure software development. He is an ISO 27001 Lead Auditor CQI/IRCA certified auditor, an instructor authorised by PaloAlto Networks in cybersecurity. His current research areas include information security, and management of processes and projects using agile methodologies.

---

## 1 Introduction

In software project management engineering, there are fundamentally two approaches: the waterfall approach and the agile approach. In the waterfall approach, also known as the traditional software development approach, there is a high detail and emphasis on the gathering of requirements in the initial phase of the project, extensive documentation, and the customer did not follow the entire development process, participant only at the beginning when the project was defined (Flewelling, 2018). However, in recent decades, an agile approach emerged that sought to fill some gaps left by the previous one. This approach promotes communication with the customer and establishes that it is necessary to monitor the changes that may arise during the project life cycle (Flewelling, 2018). Almeida (2017) and Papadopoulos (2015) state that many companies are changing their waterfall software development paradigm to agile, particularly due to the high rigidity of the waterfall approach and the potential that agile methodologies offer to include feedback from customers in the early stages of the project. With this, it is intended to reduce the risks of failure in software applications releases.

Scrum is a framework used in project management and agile software development. Instead of a sequential approach, products are progressively developed and improved iteratively and incrementally (Permana, 2015). Furthermore, rather than a hierarchical communication based on written and formal communication, the teams are self-organised and informal communication in the same physical space is promoted. However, as teams grow, it is occasionally necessary to consider environments in which members of these teams may be geographically distant (Almeida et al., 2019).

Sutherland (2019) states that a key principle of the Scrum methodology is the recognition that customers can change their minds about established requirements and associated priorities. This volatility suggests that requirements cannot be easily handled in a traditional predictive or planned manner. As such, this methodology adopts an evidence-based empirical approach, in which it is accepted that the problem cannot be fully understood or defined. Dingsøy et al. (2012) advocate the central idea is to maximise the team's ability to deliver quickly. Another key principle argued by L'Ecuyer and Ahmed (2016) is the response to emerging needs and rapid response to inevitable changes.

When managing software development projects, many activities lead to the need to quantify what is being developed. Mishra et al. (2012) state that the metrics are essential for software development companies, because they help in the measurement of software quality, in the estimation of the necessary resources and costs, in the planning and control of software development progress. Furthermore, the metrics will be a quantifiable measure that companies use to track, monitor, and evaluate the success or failure of various business processes (Van Looy and Shafagatova, 2016). Kaur and Bahl (2014) complement this view of the role of software metrics by highlighting that a good metric is one that allows the construction of indicators that facilitate decision-making without compromising its reliability.

However, the use of metrics in the agile development paradigm raises new challenges. Hartmann and Dymond (2006) state that the systematic adoption of software metrics that are too complex and require the collection of a large volume of information are incompatible with agile values and principles. Additionally, Poonacha and Bhattacharya (2012) argue that many organisations adopt agile methods without understanding which factors should be measured and controlled. This issue has been fundamentally addressed by the business community through the sharing of experiences that contribute to other companies to empirically adopt these metrics. However, from the scientific point of view, the number of studies in this field is quite limited and focuses essentially on the analysis of the impact of the Scrum paradigm on the software development process. In this sense, in the first phase, this study seeks to identify the metrics that are cited in the literature and that meet the measurement needs inherent to the Scrum approach of agile software development projects. Next, and in a second phase, it is sought to perceive the receptivity of these metrics in the business environment and explore which elements should be considered relevant to define a measurement process that meets the measurement needs of those involved in projects that adopt the Scrum methodology. The manuscript is organised as follows: Initially, a literature review is performed on the Scrum development process and the application of performance measurements, metrics, and indicators. After that, the methodology of the study is presented and then the main results are analysed and discussed. In the end, the main conclusions of this study are drawn and some suggestions for future work are given.

## **2 Literature review**

### *2.1 The scrum development process*

Scrum is one of the most successful methodologies in agile software development (Oprins et al., 2019). It was initially proposed by Ken Schwaber and Jeff Sutherland in which the ideas of industrial control processes were applied to the development of systems in software engineering (Abrahamsson et al., 2002). According to Hidalgo (2019), Scrum argues that team members should produce a flexible system in a constantly changing environment. Furthermore, three pillars support the implementation of Scrum (Schwaber and Beedle, 2001):

- i transparency, important aspects of the process should be visible and understood by stakeholders
- ii inspection, users should regularly inspect Scrum artefacts and progress against the sprint goal to detect undesirable deviations
- iii adaptation, the inspector should adjust the process or material in production when an anomaly of a process is detected in an inspection.

To understand the life cycle of a project in Scrum it is necessary to know some terminology, as established by Schwaber and Beedle (2001). A sprint corresponds to the cycles of each project. In general, sprints are 2 to 4 weeks long and should have the same duration. The product backlog is an ordered list of the functionalities of a product. The sprint backlog contains the specific tasks that will be performed and developed in each sprint or cycle. The daily scrum is a daily meeting with a maximum of 15 min for the development team to synchronise activities and create a plan for the next day. The sprint planning meeting is a periodic meeting that takes place at the beginning of each sprint to plan and prioritise the sprint functionalities. Finally, the retrospective sprint is a meeting that takes place at the end of each sprint to analyse the results of the work done on that sprint.

In addition to the events that take place during a Scrum development process, it is also relevant to look at the functions defined by Scrum. Three roles are defined:

- i product owner
- ii scrum master
- iii scrum team.

The Product Owner is the person responsible for managing and controlling the Product Backlog and participates in the process of defining the functions to be developed and estimating the effort required; The Scrum Master is responsible for ensuring that the project follows the practices and values of the Scrum methodology and progresses according to the plan outlined; The Scrum Team corresponds to the development team and has the responsibility to organise and manage its own work. It is suggested that the team has between the three and nine elements so that the work does not have productivity constraints and there is not too much complexity in managing the process (Lacey, 2016).

According to Lee (2012), the Scrum Master assumes a crucial role in the Scrum operating process because it helps all other stakeholders maximise the added value. The Scrum Master helps the Product Owner to find techniques for the effective management of the Product Backlog and to understand long-term product planning in the empirical environment (Lee, 2012). The Scrum Master also assists the Scrum Team in developing self-management processes and fostering interdisciplinary. Additionally, it also plays the role of removing impediments to the progress of the sprint (Shankarmani et al., 2014).

## *2.2 Performance measurements, metrics, and indicators*

Some concepts must be defined and clarified to understand the role of metrics in monitoring Scrum projects, namely:

- i measurement is an assessment of an attribute according to a specific method or standard
- ii metric is a method to determine whether a system, component or process has a certain attribute, usually consisting of two or more measurements
- iii indicator is a variable that can assume a certain state based on the outcome of a process or occurrence of a certain condition, and it is generally related to the interpretation of a metric in a given context (Nudurupati et al., 2011; Star et al., 2016).

A metric can be classified according to several criteria. Yauch (2011) looks at the metric from the perspective of whether it is objective or subjective. The value of an objective metric depends only on the object being analysed, while a subjective metric depends on the individual's point of view and interpretation. One of the great advantages of objective metrics is that they can be automated (Arachchi and Perera, 2018). Lee and Xia (2010) explore the quantitative and qualitative perspectives of a metric. A quantitative metric can be quantifiable, while a qualitative metric is represented through nominal scales, symbols, or figures. Hartmann and Dymond (2006) also classifies the metrics into organisational and diagnostic. Organisational metrics are responsible for measuring the amount of business value delivered to the customer, while diagnostic metrics provide relevant information for the improvement of the process that produces business value. This categorisation is not unique since Poppendieck and Poppendieck (2003) classify them in performance evaluation metrics and informative metrics. Despite the differences in terminology, the characteristics of both are identical.

The study conducted by Hartmann and Dymond (2006) is also relevant when identifying a set of factors that a good agile metric should follow. Firstly, it is stressed that the metrics must obey and reinforce the agile principles (e.g., value delivery, collaboration with the customer), and the importance of measuring results and not outputs should also be highlighted. Therefore, it can be concluded that measuring the results obtained is more important than measuring the outputs of the process activities. Wnuk and Maddila (2017) also highlight the importance of metrics to provide frequent and regular feedback. With this, it is intended to amplify the learning and accelerate the improvement process. Another relevant aspect is that the metrics offer reliable values that meet the company's needs (Pressman and Maxim, 2014). Finally, Shawky and Abd-El-Hafiz (2014) emphasise the role of metrics in contributing to measuring the quality of the software. Given these multiple objectives of software metrics, it is relevant to explore the elements that should be part of a measurement process in a Scrum environment. In this sense, the following research question was defined:

*RQ1:* What elements should be considered in a measurement process in Scrum development teams?

One of the best-known metrics in Scrum is team velocity, which measures the number of user stories completed by the team in each sprint (Ahmed et al., 2017). From it becomes possible to estimate the speed of team development in future sprints. This is a metric widely used in Scrum development teams, but it should be employed with some caution since it is a subjective measure and does not allow a comparison of performance between teams (Akif and Majeed, 2012). Another classic metric is the sprint burndown that analyses progress within a sprint (Ahmed et al., 2017). Through its analysis can be

estimated the number of hours remaining to complete the user stories of the sprint. Another similar metric is the release burndown, in which it is explored the correlation between the amount of remained work (unfinished user stories in the product backlog) and the team's progress in reducing this work (Mahnic and Zabkar, 2012).

Other metrics related to the Scrum development process are proposed by Agarwal and Majumdar (2012), respectively:

- i standard violation
- ii defects per interaction
- iii number of stories
- iv level of automation
- v number of tests.

Other proposed metric is the defect density, which is based on the defects per interaction metric (Farid et al., 2015). In addition to these metrics in the work published by Layton (2012), it emerges the concept of sprint goal success rate which is based on the notion of a successful sprint (i.e., a sprint that fulfils the sprint goals and is considered by the team as done). There are also metrics related to the development activity of a Scrum like team satisfaction and team member turnover (Tripp et al., 2016). The first of these metrics is a qualitative indicator that can be obtained from surveys or when performing the retrospective sprint.

The systematic review work on metrics of Scrum methodology developed by Ifra and Bajwa (2016) is a key resource for having a comprehensive understanding of the various metrics that can be used in Scrum development environments. In addition to the metrics previously identified, others such as business value delivered, time to market, total product cost, or return on investment emerge. Of the 16 metrics identified by Ifra and Bajwa (2016) it is verified that most of them are common to several Agile environments (e.g., TDD, XP, Kanban) and can also be applied in waterfall development environments.

The existence of a large set of metrics with different objectives poses the challenge of knowing their usefulness and relevance in a business context. Indeed, it is expected that the degree of applicability of each of these metrics is heterogeneous and, consequently, it is relevant to explore the business perspective on these metrics. In this sense, two research questions were established:

*RQ2*: What are the metrics that best meet the measurement needs in a Scrum development business environment?

*RQ3*: Is the profile of Scrum development companies a factor influencing the perceived relevance of these Scrum metrics?

### **3 Methodology**

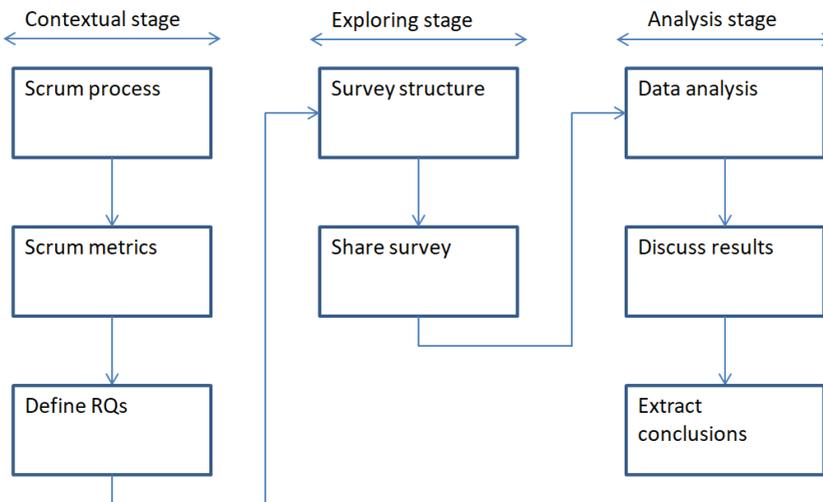
The adopted methodology can be organised in three fundamental phases as shown in Figure 1. In the contextual dimension, research is performed on the Scrum development process and the metrics that are known in the literature that allows the assessment of the development maturity of software engineering projects based on the Scrum methodology. Still in this phase, the research questions that support the realisation of this study are

defined. Next, in the exploring dimension, the structure of the survey is defined, and the survey is made available to Portuguese companies that belong to group 620 of the Portuguese Classification of Economic Activities (CAE). In this CAE, we find companies that perform their main activity in one of the following sub-areas:

- i computer programming activities
- ii computer consultancy activities
- iii management and operation of computer equipment
- iv other activities related to information technology and computing.

The survey was created on Google Drive and made available during the months from October to November 2019. Finally, in the analysis dimension, the survey data is statistically explored considering techniques in the field of descriptive and inferential statistics. Then, the obtained findings are interpreted considering the literature in the field and, finally, the main results of this study are presented.

**Figure 1** Phases of the adopted methodology (see online version for colours)



The fundamental objective of this study is to explore the performance metrics considering a Scrum software development process. The metrics previously identified in the literature review allowed us to identify that most of the metrics are not exclusive to a Scrum development environment and can be adopted in other agile methodologies and traditional waterfall models of software development. In this sense, Table 1 seeks to synthesise the various identified metrics and summarise in which areas they can be used, their customisation considering the unique features offered by a Scrum environment and the main authors who support each metric. The metrics can be grouped considering their scope:

- i measuring deliverables
- ii measuring effectiveness
- iii measuring the Scrum team.

A total of 21 metrics was identified, 12 of which are specific to the Scrum environment. The other 9 are metrics used both in Scrum and waterfall environments.

**Table 1** Synthetisation of scrum metrics

<i>Metric</i>	<i>Scope</i>	<i>Scrum specific</i>	<i>Authors</i>	<i>Description</i>
Team velocity	Measuring deliverables	Yes	Ahmed et al. (2017) Hayes et al. (2014)	Measures the amount of work a team can tackle during a single sprint
Sprint burndown	Measuring deliverables	Yes	Ahmed et al. (2017) Alegria et al. (2011) Hayes et al. (2014)	Graphic representation of the rate at which work is completed and how much work remains to be done in a sprint
Release burndown	Measuring deliverables	Yes	Alegria et al. (2011) Mahnica and Zabkar (2012)	Graphic representation of the rate at which work is completed and how much work remains to be done in a release
Standard violation	Measuring deliverables	Yes	Agarwal and Majumdar (2012)	Measures the number of standards violated by sprint track
Defects per interaction	Measuring deliverables	Yes	Agarwal and Majumdar (2012)	Calculates the defects that occurred during a sprint
Number of stories	Measuring deliverables	Yes	Agarwal and Majumdar (2012) Ifra and Bajwa (2016)	Calculates the number of stories in a release or a sprint
Level of automation	Measuring effectiveness	Yes	Agarwal and Majumdar (2012) Tyagi et al. (2018)	Indicates how many tests are automated by release or sprint.
Number of tests	Measuring deliverables	Yes	Agarwal and Majumdar (2012) Ifra and Bajwa (2016)	Calculates the number of tests that have been developed, executed, and passed to validate a story or the entire release
Defect density	Measuring deliverables	No	Farid et al. (2015)	Number of defects found divided by the size of the considered module/software
Team satisfaction	Monitoring the Scrum team	No	Tripp et al. (2016)	Degree of satisfaction of the team with the working environment and adopted methodologies
Team member turnover	Monitoring the Scrum team	No	Tripp et al. (2016)	Indicates the turnover of team members considering a full development cycle
Failed deployments	Measuring deliverables	No	Tanner and Mackinnon (2015)	Number of failed deployments considering a full development cycle
Lead time	Measuring deliverables	Yes	Hayes et al. (2014) Kupiainen et al. (2015)	Measures the elapsed time when work starts on an item (story, task, bug etc.) until it's ready for delivery

**Table 1** Synthetisation of scrum metrics (continued)

<i>Metric</i>	<i>Scope</i>	<i>Scrum specific</i>	<i>Authors</i>	<i>Description</i>
Work in progress	Measuring deliverables	Yes	Hayes et al. (2014) Kupiainen et al. (2015)	Indicates the stories that the team is currently working on
Business value delivered	Measuring effectiveness	Yes	Ifra and Bajwa (2016) Sulaiman et al. (2006)	Indicates how much value the company attached to a story
Sprint goal success	Measuring deliverables	Yes	Ifra and Bajwa (2016) Layton (2012)	Tracks how frequently it succeeds in reaching the sprint goals
Total project duration	Measuring effectiveness	No	Ifra and Bajwa (2016)	Total project duration typically estimated in weeks or days
Time to market	Measuring effectiveness	No	Ifra and Bajwa (2016) Sulaiman et al. (2006)	Measures the time it takes for a product to be conceived until it is available for sale
Total product cost	Measuring effectiveness	No	Ifra and Bajwa (2016) Sulaiman et al. (2006)	Total cost of the product considering its various components and phases
Return on investment	Measuring effectiveness	No	Ifra and Bajwa (2016) Sulaiman et al. (2006)	Relationship between the result obtained and the investment made in the software
Lack of cohesion in method	Measuring effectiveness	No	Ifra and Bajwa (2016)	Measures how the methods of a class are related to each other

The questionnaire consists of 15 groups of questions in two sections. In the 1st section, three control data questions are posed regarding the size of the company, role in the agile development process, and the number of years of experience in Scrum. In the categorisation of the company's size, the OECD reference regarding the number of employees of each company was adopted (OECD, 2019). In the 2nd section, eight evaluation questions are asked for each Scrum metric, such as:

- i reinforce the agile practices
- ii improve the communication
- iii improve the customer involvement
- iv improve the training of team members
- v increase the support from top management
- vi enhance the delivery strategy
- vii offer reliable values
- viii contribute to measuring the quality of software.

This group of questions adopts the Likert scale of five levels (i.e., strongly disagree; disagree; neutral; agree; strongly agree. According to Yin (2017), the use of the Likert scale enables us to measure attitudes and know the interviewee's degree of compliance

with the questions raised. In this sense, this scale is useful to measure and capture the intensity of the respondent's opinion.

The questionnaire data were statistically analysed using SPSS v.21. Cronbach's alpha coefficient was calculated to determine the internal consistency of the questionnaire. This coefficient offers the advantage of being easily calculated and can be applied when the questionnaire is applied only once. Taber (2018) states the Cronbach's alpha coefficient should be at least higher than 0.6 to ensure substantial internal consistency of the questionnaire. In this study, the Cronbach's alpha coefficient value calculated for the control data is 0.682 and for the Scrum metrics evaluation section is 0.956.

A total of 158 responses were received, 21 of which were excluded due to lack of total or partial completion of control data variables. In this sense, 137 valid answers from Scrum IT professionals were considered. Table 2 presents a brief statistical analysis of the control variables. It was found that more than 50% of the individuals are part of the development team in Scrum and most of them also have more than 3 years of experience in Scrum. This factor indicates a high level of respondents' maturity in knowledge of Scrum practices. Finally, it should be noted that most individuals work in small or medium-sized companies.

**Table 2** Sample characteristics

<i>Variable</i>	<i>Absolute frequency</i>	<i>Relative frequency</i>
<i>What is your role?</i>		
Product Owner	23	0.168
Scrum Master	42	0.307
Development Team	72	0.526
<i>How many years of experience in Scrum?</i>		
Less than 1 year	17	0.124
Between 1 and 3 years	26	0.190
Between 3 and 5 years	46	0.336
More than 5 years	48	0.350
<i>What is the size of your company?</i>		
Micro company (<10 employees)	18	0.131
Small company (<50 employees)	53	0.387
Medium-sized company (<250 employees)	45	0.328
Big company ( $\geq$ 250 employees)	21	0.153

## 4 Results and discussion

### 4.1 Relevance of metrics in a Scrum environment

Table 3 presents a statistical analysis of each metric considering the following eight characteristics:

- i RAP, reinforce the agile practices
- ii IC, improve the communication

- iii ICI, improve the customer involvement
- iv ITT, improve the training of team members
- v IST, increase the support from top management
- vi EDS, enhance the delivery strategy
- vii ORV, offer reliable values
- viii MQS, contribute to measuring the quality of software.

For each metric, the mean, and standard deviation (std. dev.) are calculated.

The reinforcement of agile practices is essentially fostered in three metrics:

- i business value delivered
- ii sprint goal success
- iii work in progress.

These results are aligned with the fundamental priority of agile methodologies, which is to satisfy the consumer through the continuous delivery of valuable software. According to Rigby et al. (2016), delivering value as early as possible helps the customer to understand what will bring more value in the near future. It becomes pertinent to recognise that the customer is easier to understand how a system works when he has an incremental version of it. In this way, the customer can still find details that are left out and are very important to the user.

The metrics for ‘business value delivered’ and ‘sprint goal success’ are those that most contribute to improving the communication and customer involvement. These two characteristics seem to be correlated. These two indicators are essential elements for companies to measure the outcome of Scrum projects. A pertinent issue is to look at how both metrics can be applied in practice. The business value delivered can be estimated by looking at the absolute value of the overdue item or by making a relative estimate, in which the value of the overdue item is considered in comparison to other delayed items. The sprint goal success can be used to track how often the sprint goals are achieved. This information is useful to understand the level of maturity of the team. Moreover, Layton (2012) highlights the role of sprint goal success for the development team to inspect the progress of a user story.

A development team in Scrum is naturally multidisciplinary and therefore it is important that agile metrics can promote the development of skills in various dimensions, both hard and soft skills. It is pertinent to consider that there is a diverse set of metrics that promote the training of team members, such as working in progress, lead time, business value delivered, among others.

The increase of support from top management is mainly achieved through the ‘business value delivered’ metric. On the other hand, the lead time is a metric that contributes to enhancing the delivery strategy. Apparently, in Scrum methodology, the lead time is fixed and defined by Sprint duration. However, this view is necessarily simplistic because it only considers the user stories in production. In this sense, and according to Minick (2019) advocates, lead time should consider the entire release pipeline, including the user stories that are still in the backlog. Through this, the teams can perceive their ability to meet customer requirements.

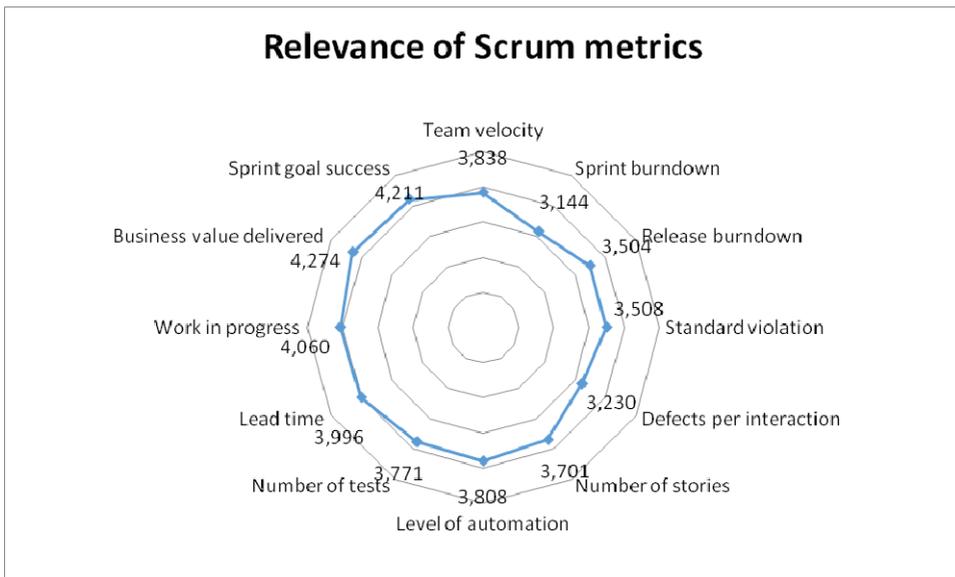
**Table 3** Relevance of scrum metrics

<i>Metric</i>	<i>RAP</i>	<i>IC</i>	<i>ICI</i>	<i>ITT</i>	<i>IST</i>	<i>EDS</i>	<i>ORV</i>	<i>MQS</i>
<i>Team velocity</i>								
Mean	3.927	3.869	3.891	3.854	3.942	3.964	3.737	3.518
Std. dev.	0.376	0.794	0.773	0.354	0.359	0.410	0.807	0.502
<i>Sprint burndown</i>								
Mean	3.007	3.526	3.175	3.204	3.007	3.109	2.912	3.212
Std. dev.	0.903	1.207	0.593	0.876	0.781	0.837	0.836	0.878
<i>Release burndown</i>								
Mean	3.584	3.445	3.613	3.511	3.175	3.547	3.613	3.543
Std. dev.	0.703	0.568	0.585	0.925	0.452	0.568	1.024	0.982
<i>Standard violation</i>								
Mean	3.540	3.526	3.577	3.591	3.438	3.540	3.204	3.650
Std. dev.	0.556	0.530	0.565	0.601	0.540	0.556	0.596	0.564
<i>Defects per interaction</i>								
Mean	3.307	3.620	3.314	3.730	2.803	3.139	3.226	2.701
Std. dev.	0.648	0.917	1.217	0.845	0.882	0.842	0.970	1.140
<i>Number of stories</i>								
Mean	3.810	3.785	4.015	4.007	3.409	3.912	3.459	3.212
Std. dev.	0.753	0.737	0.776	0.636	0.800	0.702	0.805	0.752
<i>Level of automation</i>								
Mean	3.730	3.657	3.285	3.715	3.723	4.146	4.007	4.204
Std. dev.	0.522	0.803	0.499	0.499	0.539	0.478	0.636	0.749
<i>Number of tests</i>								
Mean	4.007	3.810	3.701	3.803	3.321	3.927	3.810	3.790
Std. dev.	0.636	0.753	0.586	0.765	0.581	0.564	0.753	0.746
<i>Lead time</i>								
Mean	3.730	3.993	4.131	4.117	3.599	4.401	4.000	4.000
Std. dev.	0.575	0.636	0.497	0.515	1.018	0.492	0.630	0.343
<i>Work in progress</i>								
Mean	4.204	4.007	3.599	4.204	4.000	4.241	4.007	4.215
Std. dev.	0.749	0.895	0.799	0.749	0.630	0.636	0.895	0.749
<i>Business value delivered</i>								
Mean	4.511	4.204	4.482	4.153	4.299	4.255	4.109	4.182
Std. dev.	0.583	0.749	0.608	0.737	0.657	0.543	0.811	0.677
<i>Sprint goal success</i>								
Mean	4.285	4.161	4.321	4.146	4.058	4.292	4.234	4.187
Std. dev.	0.696	0.597	0.840	0.772	0.802	0.596	0.546	0.538

Finally, it is important to note that respondents considered that not all metrics offer reliable values. In particular, the Sprint burndown is considered the least reliable metric. The idea of this metric is to give visibility to the remaining work in the context of a Sprint (Hayes et al., 2014). However, this metric alone may give unreliable results, as the complexity of each user story may necessarily be different and lead to quite different runtimes as well. Exploring the contribution of metrics to measure software quality shows that the ‘defects per interaction’ metric is the one with the lowest score. However, it is also this metric that has received the highest standard deviation, meaning that there is a large dispersion of opinion among respondents. Apparently, this result is surprising since one would expect that having a perception of the number of defects per interaction helps to understand the contribution made by the development process to the increase of software quality. However, the excessive quantitative weight of this indicator does not enable exploring the qualitative dimension of the error and its causes. Singh (2019) advocates that it becomes more pertinent to analyse the causes of the occurrence of each bug that may not be focused only on the technical component, but on a poor specification of the client’s need.

For each dimension, the average of the eight sub-dimensions was calculated. Figure 2 shows the relevance of each metric in a radial graph. The metrics related to ‘business value delivered’ and ‘sprint goal success’ are those that stand out as being the most relevant in a Scrum environment. On the opposite side, ‘sprint burndown’ and ‘defects per interaction’ metrics have less receptivity in a Scrum business environment.

**Figure 2** Relevance of scrum metrics (see online version for colours)



#### 4.2 Exploring the impact of control variables

Analysis of variance (ANOVA) is a statistical procedure that can be used to compare the distribution of three or more groups in independent samples. According to Spiegelhalter (2019), analysis of variance can be used to summarise a linear regression model by

decomposing the sum of the squares for each source of variation in the model and, using the F-test, test the hypothesis that any source of variation in the model is equal to zero. In this study, we employed the one-way ANOVA for each factor (i.e., role, years of experience, and size of the company) because all groups have a normal distribution of equal variance. In this sense, the mean between the groups was compared and the F and significance values were determined as shown in Table 4. A significance level of 5% was adopted ( $\sigma = 0.05$ ).

**Table 4** Relevance of control variables

<i>Metric</i>	<i>Mean</i>	<i>Std. dev.</i>	<i>Role</i>		<i>Years of experience</i>		<i>Size of the company</i>	
			<i>F value</i>	<i>Sig.</i>	<i>F value</i>	<i>Sig.</i>	<i>F value</i>	<i>Sig.</i>
<i>Team velocity</i>	3.838	0.547	1.019	0.364	5.590	0.001	1.244	0.296
<i>Sprint burndown</i>	3.144	0.864	2.998	0.053	11.527	1*10 <sup>-3</sup>	2.066	0.108
<i>Release burndown</i>	3.504	0.726	5.714	0.004	17.634	1*10 <sup>-3</sup>	5.291	0.002
<i>Standard violation</i>	3.508	0.564	1.816	0.167	10.980	1*10 <sup>-3</sup>	3.628	0.015
<i>Defects per interaction</i>	3.230	0.933	1.962	0.145	15.113	1*10 <sup>-3</sup>	2.097	0.104
<i>Number of stories</i>	3.701	0.745	5.551	0.005	22.116	1*10 <sup>-3</sup>	9.283	1*10 <sup>-3</sup>
<i>Level of automation</i>	3.808	0.591	1.300	0.276	7.504	1*10 <sup>-3</sup>	1.168	0.324
<i>Number of tests</i>	3.771	0.673	1.688	0.189	14.479	1*10 <sup>-3</sup>	1.710	0.168
<i>Lead time</i>	3.996	0.588	1.115	0.331	14.242	1*10 <sup>-3</sup>	1.576	0.198
<i>Work in progress</i>	4.060	0.763	3.769	0.026	4.655	0.004	6.984	1*10 <sup>-3</sup>
<i>Business value delivered</i>	4.274	0.671	5.158	0.007	3.898	0.010	5.696	0.001
<i>Sprint goal success</i>	4.211	0.673	6.907	0.001	4.473	0.005	3.782	0.012

The findings indicate that years of experience is a determining factor in the evaluation of the relevance of each of the metrics by respondents. The individuals with the most years of experience are those who attach the greatest importance to the application of metrics to measure team performance in a Scrum environment. These findings are aligned with the results obtained by both Mahnic and Zabkar (2012) and Bancroft-Connors (2017) that highlight the relevance of experience in the process of using Scrum metrics. Furthermore, Bancroft-Connors (2017) highlights that the experience of the individual is a relevant factor for teams to understand the relative importance of each metric and, above all, to select the metrics that can give actionable value. Moreover, individuals who have had previous experience in the three roles in Scrum environment are also those who best know the needs of the teams and consequently offer greater conditions to increase team collaboration and commitment (Khosravi et al., 2017).

The role of employees in the Scrum team also assumes some relevance in the perception of the utility of each Scrum metric. Nevertheless, its relevance is clearly less when compared to the years of experience of the respondents. Individuals with a product owner role tend to value metrics related to business value delivered and sprint goal success, while scrum masters value release burndown, number of stories, and work in progress more strongly. The importance of metrics for both the product owner and the scrum master becomes clear. Sverrisdottir et al. (2014) point out that the work of the product owner is essentially composed of planning and communication. Consequently, it becomes essential to ensure that stakeholders need to have a clear vision of the work to be done in each sprint and all stakeholders need to have a defined vision for the product. Similarly, the scrum master needs to have a clear vision of what to do in the home sprint. This way, it becomes clear the role of metrics in increasing visibility about the development process and its outcomes.

The same applies to the factor relating to the size of the company where the employee works. In this case, some metrics are more important for larger companies, namely the following six metrics:

- i release burnout
- ii standard violation
- iii number of stories
- iv work in progress
- v business value delivered
- vi sprint goal success.

The application of agile methodologies such as Scrum is often questioned in large companies that involve managing interdependencies between multiple teams and dealing with non-agile hierarchical pyramids. However, the use of agile practices such as Scrum on a large scale can be successfully achieved as Almeida et al. (2019) and Conboy and Carroll (2019) highlight and, for that purpose, it becomes fundamental to promote communication and collaboration among teams using knowledge management tools. Furthermore, it is important to recognise that the adoption of Scrum in larger teams requires a change in mentality in people and organisations. In this sense, agile metrics can make an important contribution to increasing the visibility and transparency of the work developed by multiple teams.

## **5 Conclusions**

In the current competitive business environment, where critical business projects are crucial to an organisation's sustainable progress, metrics are key to measuring and tracking project progress. Without the application of metrics, a team can implement Scrum in its most basic form but will not be continuously improving which is a serious risk to the sustainability of the organisation. Consequently, and to remain agile an organisation must always be seeking continuous improvement, and metrics play a key role in the process of assertive and sustainable improvement.

This study offers relevant theoretical and practical contributions. From a theoretical point of view, a total of 12 specific metrics applied in a Scrum environment were found and it was possible to compare the relative importance of them. For this purpose, it was identified that the metrics related to the business value delivered and sprint goal success measurement process are the most relevant, considering multiple dimensions such as improving communication, enhancing the delivery strategy, contribution to the process of measuring software quality, among others. The study also revealed that the perception of the importance of these metrics is mainly affected by the number of years of experience of respondents, although other factors like the role of employees and size of the organisation also have significant relevance for several metrics such as work in progress, sprint goal success, business value delivered, among others. From a practical perspective, this information is relevant for organisations that adopt the Scrum methodology and thus can apply these metrics in the development of their business activity. These Scrum specific metrics will contribute to companies assess their software development processes, ensuring that Scrum teams deliver maximum value to customers at each interaction. These metrics are not only relevant to measure the success of deliverables and effectiveness of a Scrum team but can also be applied as a benchmarking within the company (for past, present, and future software development process). If these metrics are agreed among companies, it can be an industry benchmarking standard and becomes targets for future improvement.

As future work, it might be pertinent to explore and monitor the process of applying these metrics over time by conducting a longitudinal study. Moreover, it would also be pertinent to carry out a comprehensive study of the challenges faced in the application of these metrics in large-scale agile teams.

## References

- Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J. (2002) *Agile Software Development Methods Review and Analysis*, VTT Publication, p.478, Retrieved from <https://www.vtt.fi/inf/pdf/publications/2002/P.478.pdf>
- Agarwal, M. and Majumdar, R. (2012) 'Tracking scrum projects tools, metrics and myths about agile', *International Journal of Emerging Technology and Advanced Engineering*, Vol. 2, No. 3, pp.97–104.
- Ahmed, A., Tayyab, M., Bhatti, S., Alzahrani, A. and Babar, M. (2017) 'Impact of story point estimation on product using metrics in scrum development process', *International Journal of Advanced Computer Science and Applications*, Vol. 8, No. 4, pp.385–391.
- Akif, R. and Majeed, H. (2012) 'Issues and challenges in scrum implementation', *International Journal of Scientific and Engineering Research*, Vol. 3, No. 8, pp.1–4.
- Alegria, J., Bastarrica, M.C. and Bergel, A. (2011) 'Is it safe to adopt the scrum process model?', *CLEI Electronic Journal*, Vol. 14, No. 3, pp.1–11.
- Almeida, F. (2017) 'Challenges in migration from waterfall to agile environment', *World Journal of Computer Application and Technology*, Vol. 5, No. 3, pp.39–49.
- Almeida, F., Miranda, E. and Falcão, J. (2019) 'Challenges and facilitators practices for knowledge management in large-scale scrum teams', *Journal of Information Technology Case and Application Research*, Vol. 21, No. 2, pp.90–102.
- Arachchi, S. and Perera, I. (2018) 'Continuous integration and continuous delivery pipeline automation for agile software project management', *Proceedings of the MERCon 2018*, Sri Lanka, pp.1–7.

- Bancroft-Connors, J. (2017) *4 Balanced Metrics for Tracking Agile Teams*, Retrieved from <https://www.agileconnection.com/article/4-balanced-metrics-tracking-agile-teams>
- Conboy, K. and Carroll, N. (2019) 'Implementing large-scale agile frameworks: challenges and recommendations', *IEEE Software*, Vol. 36, No. 2, pp.44–50.
- Dingsøyr, T., Nerur, S., Balijepally, V. and Moe, N. (2012) 'A decade of agile methodologies: towards explaining agile software development', *Journal of Systems and Software*, Vol. 85, No. 6, pp.1213–1221.
- Farid, A., Fathy, E. and Ellatif, M. (2015) 'Towards agile implementation of test maturity model integration (TMMI) level 2 using scrum practices', *International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 9, pp.230–238.
- Flewelling, P. (2018) *The Agile Developer's Handbook: Get More Value From your Software Development: Get the Best Out of the Agile Methodology*, Packt Publishing, Birmingham, UK.
- Hartmann, D. and Dymond, R. (2006) 'Appropriate agile measurement: using metrics and diagnostics to deliver business value', *Proceedings of the 2006 Agile Conference*, Washington, DC, pp.126–134.
- Hayes, W., Garcia-Miller, S., Lapham, M., Wrubel, E. and Chick, T. (2014) 'Agile metrics: progress monitoring of agile contractors', *Software Engineering Institute*, Retrieved from <http://repository.cmu.edu/sei/775>
- Hidalgo, E. (2019) 'Adapting the scrum framework for agile project management in science: case study of a distributed research initiative', *Heliyon*, Vol. 5, No. 3, pp.1–32.
- Ifra, I. and Bajwa, J. (2016) 'Metrics of scrum methodology', *International Journal of Computer Applications*, Vol. 149, No. 2, pp.24–27.
- Kaur, G. and Bahl, K. (2014) 'Software reliability, metrics, reliability improvement using agile process', *International Journal of Innovative Science, Engineering and Technology*, Vol. 1, No. 3, pp.143–147.
- Khosravi, A., Gandomani, T. and Fahimian, H. (2017) 'Introduction of scrum in an elite team: a case study', *Journal of Software*, Vol. 12, No. 4, pp.173–179.
- Kupiainen, E., Mäntylä, M., Mika, V. and Itkonen, J. (2015) 'Using metrics in agile and lean software development – A systematic literature review of industrial studies', *Information and Software Technology*, Vol. 62, pp.143–163.
- L'Ecuyer, A. and Ahmed, S. (2016) 'Controlling change on agile software development projects', *Universal Journal of Management*, Vol. 4, No. 1, pp.42–49.
- Lacey, M. (2016) *Scrum Field Guide, The Agile Advice for Your First Year and Beyond*, Addison-Wesley, Boston, Massachusetts.
- Layton, M. (2012) *Agile Project Management for Dummies*, O'Reilly Media, Newton, Massachusetts.
- Lee, G. and Xia, W. (2010) 'Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility', *MIS Quarterly*, Vol. 34, No. 1, pp.87–114.
- Lee, R. (2012) 'The success factors of running scrum: a qualitative perspective', *Journal of Software Engineering and Applications*, Vol. 5, pp.367–374.
- Mahnic, V. and Zabkar, N. (2012) 'Measuring progress of scrum-based software projects', *Elektronika Ir Elektrotehnika*, Vol. 18, No. 8, pp.73–76.
- Minick, E. (2019) Measure Lead Time for the business. Retrieved from <https://dzone.com/articles/measure-lead-time-for-the-business>
- Mishra, D., Balcioglu, E. and Mishra, A. (2012) 'Measuring project and quality aspects in agile software development', *Technics Technologies Education Management*, Vol. 7, No. 1, pp.122–127.
- Nudurupati, S., Bititci, U., Kumar, V. and Chan, F. (2011) 'State of the art literature review on performance measurement', *Computers and Industrial Engineering*, Vol. 60, No. 2, pp.279–290.

- OECD (2019) *Enterprises by Business Size*, Retrieved from <https://data.oecd.org/entrepreneur/enterprises-by-business-size.htm>
- Oprins, R., Frijns, H. and Stettina, C. (2019) 'Evolution of scrum transcending business domains and the future of agile project management', *Proceedings of the International Conference on Agile Software Development (XP.2019)*, Montreal, Canada, pp.244–259.
- Papadopoulos, G. (2015) 'Moving from traditional to agile software development methodologies also on large, distributed projects', *Procedia – Social and Behavioral Sciences*, Vol. 175, pp.455–463.
- Permana, P. (2015) 'Scrum method implementation in a software development project management', *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 6, No. 9, pp.198–204.
- Poonacha, K. and Bhattacharya, S. (2012) 'Towards a framework for assessing agility', *Proceedings of the 45th Hawaii International Conference*, Hawaii, USA, pp.5329–5338.
- Poppendieck, M. and Poppendieck, T. (2003) *Lean Software Development: An Agile Toolkit*, Addison-Wesley Professional, Boston, Massachusetts.
- Pressman, R. and Maxim, B. (2014) *Software Engineering: A Practitioner's Approach*, McGraw-Hill Education, New York.
- Rigby, D.K., Sutherland, J. and Takeuchi, H. (2016) 'Embracing agile', *Harvard Business Review*, Retrieved from <https://hbr.org/2016/05/embracing-agile>
- Schwaber, K. and Beedle, M. (2001) *Agile Software Development with Scrum*, Pearson, London, UK.
- Shankarmani, R., Mantha, S. and Babu, V. (2014) 'A framework for reducing impediments in agile projects', *International Journal of Computer Applications*, Vol. 102, No. 11, pp.7–14.
- Shawky, D. and Abd-El-Hafiz, S. (2014) 'The impact of agile approaches on software quality attributes an empirical study', *Proceedings of the 9th International Conference on Software Paradigm Trends*, Vienna, Austria, pp.49–57.
- Singh, S. (2019) *Good Tests and Bad Metrics: Choosing Quality Over Quantity in Software Testing*, Retrieved from <https://dzone.com/articles/why-to-choose-quality-over-quantity-in-software-te>
- Spiegelhalter, D. (2019) *The Art of Statistics: How to Learn From Data*, Basic Books, New York.
- Star, S., Russ-Eft, D., Braverman, M. and Levine, R. (2016) 'Performance measurement and performance indicators: a literature review and a proposed model for practical adoption', *Human Resource Development Review*, Vol. 15, No. 2, pp.151–181.
- Sulaiman, T., Barton, B. and Blackburn, T. (2006) 'AgileEVM – earned value management in scrum projects', *Proceedings of the Agile 2006 Conference*, Minneapolis, MN, pp.1–16.
- Sutherland, J. (2019) *The Scrum Fieldbook: A Master Class on Accelerating Performance, Getting Results, and Defining the Future*, Currency Press, Danvers, MA.
- Sverrisdottir, H., Ingason, H. and Jonasson, H. (2014) 'The role of the product owner in scrum-comparison between theory and practices', *Procedia – Social and Behavioral Sciences*, Vol. 119, pp.257–267.
- Taber, K. (2018) 'The use of Cronbach's alpha when developing and reporting research instruments in science education', *Research in Science Education*, Vol. 48, No. 6, pp.1273–1296.
- Tanner, M. and Mackinnon, A. (2015) 'Sources of interruptions experienced during a scrum sprint', *The Electronic Journal of Information Systems Evaluation*, Vol. 18, No. 1, pp.3–18.
- Tripp, J., Riemenschneider, C. and Thatcher, J. (2016) 'Job satisfaction in agile development teams: agile development as work redesign', *Journal of the Association for Information Systems*, Vol. 17, No. 4, pp.267–307.
- Tyagi, S., Sibal, R., Suri, B., Wadhwa, B. and Shekhar, S. (2018) 'Development of reusable hybrid test automation framework for web based scrum projects', *Journal of Applied Science and Engineering*, Vol. 21, No. 3, pp.455–462.

- Van Looy, A. and Shafagatova, A. (2016) 'Business process performance measurement: a structured literature review of indicators, measures and metrics', *SpringerPlus*, Vol. 5, pp.1–24.
- Wnuk, K. and Maddila, K. (2017) 'Agile and lean metrics associated with requirements engineering', *Proceedings of the 27th International Workshop on Software Measurement*, Haarlem, Netherlands, pp.33–40.
- Yauch, C. (2011) 'Measuring agility as a performance outcome', *Journal of Manufacturing Technology Management*, Vol. 22, No. 3, pp.384–404.
- Yin, R. (2017) *Case Study Research and Applications: Design and Methods*, SAGE Publications, Thousand Oaks, California.