

Agile Metrics in IT Projects: A Balanced and Context-Aware Framework for Selection and Application

Alem Čolaković, Bakir Karahodža, Belma Mušmulja

University of Sarajevo, Faculty of Traffic and Communications, Sarajevo, Bosnia and Herzegovina.

Abstract

Agile metrics support empirical management in modern Information Technology (IT) projects. However, their interpretation and practical use depend strongly on project context, organizational maturity, and the selected development approach. This paper provides a structured analysis of agile metrics and develops an integrated approach to their contextual selection in IT projects. Using a design-oriented synthesis of relevant literature and official framework documentation, the study classifies agile metrics by measurement domain, including product, process, quality, project, and resource metrics, and by managerial purpose. Their application is analyzed across widely used agile approaches and frameworks, including Scrum, Kanban, Lean, the Scaled Agile Framework (SAFe), and the Dynamic Systems Development Method (DSDM). Based on this analysis, the paper proposes a structured CPV (Context-Purpose-Value) matrix that supports decisions about which metric groups and key performance indicators (KPIs) to prioritize by jointly considering project context, measurement purpose, and dominant value dimensions. A scenario-based application illustrates how the matrix can be used to derive and interpret a context-specific metric set without implying universal applicability. The synthesis indicates that no single metric can adequately represent overall IT project success. Instead, project performance should be evaluated through a combination of metrics addressing project efficiency, user impact, business success, and future readiness. The proposed framework supports a more balanced and context-aware use of agile metrics but requires external empirical validation.

Keywords: agile metrics classification, contextual metric selection, balanced agile metrics, CPV matrix, agile performance measurement, IT project success.

1. Introduction

Agile approaches have become the dominant framework for managing Information Technology (IT) projects over the past two decades. This development is primarily driven by their flexibility, advances in development tools, the iterative nature of development, and the ability to adapt to changing user requirements and environmental conditions. In such a context, agile metrics play a central role in providing an objective view of project progress, team efficiency, and the quality of delivered solutions. Unlike traditional project management models, which rely heavily on planned

schedules, budgets, and predictive estimates, agile approaches are based on empirical measurement using real data, continuous delivery, and adaptive planning. Agile metrics therefore serve as key instruments for decision-making and for identifying issues during project execution. Although agile frameworks and methodologies were initially developed in software engineering, their application now extends to a broader range of IT projects, including infrastructure, DevOps, information security, Enterprise Resource Planning (ERP) systems, and digital transformation initiatives. However, research clearly shows that software development remains the primary and most extensively

Corresponding author: Alem Čolaković (alem.colakovic@fsk.unsa.ba)

Received: 30 March 2026; Revised: 1 October 2026; Accepted: 8 October 2026; Published: 21 October 2026

© 2026 The Author(s). This work is licensed under a Creative Commons Attribution 4.0 International License

studied domain for agile metrics [1], [2], [3]. For this reason, this area can be considered a representative domain for analyzing measurement principles and models that are also applicable to other types of IT projects.

The literature identifies several recurring groups of agile metrics and related measurement practices. Process and performance measures are the most prevalent and include indicators and visualization techniques such as velocity, story points, effort estimation, burndown charts, commitment reliability, and sprint speed [2], [4], [5]. Quality metrics represent the second major group and include defect density, defect rate and defect leakage, as well as various approaches for assessing qualitative requirements such as security and usability [1], [6], [7]. The third group relates to project success metrics beyond the traditional triple constraint of time, cost, and scope, where the focus shifts toward business value, user satisfaction, project visibility, and process improvement [8], [9]. Finally, there is growing research interest in large-scale agile development and scaled frameworks such as SAFe, where Flow Distribution, Flow Velocity, Flow Time, Flow Load, Flow Efficiency, and Flow Predictability are used to assess value-stream performance [10], [11], [12], [13]. However, despite their widespread use, agile metrics are often applied in practice in a fragmented manner, without clear systematization, defined selection criteria, or consistent interpretation, which can lead to misleading conclusions and limited managerial value. As a result, metrics are frequently selected on an ad hoc basis, without clear criteria, standardization, or unified interpretation. In addition, there is a high diversity of metrics, including custom metrics, which makes it difficult to compare teams and ensure consistent interpretation of results [2], [3], [4]. Some studies have shown that organizations often select metrics using a combination of literature-based practices and what teams perceive as useful, without a common framework or formally defined selection criteria [3], [14]. For these reasons, there is a need for a systematic analysis and classification of agile metrics, as well as for a clearer understanding of how they are applied across different agile frameworks.

The aim of this paper is to develop an integrated analytical framework for the classification, selection, and interpretation of agile metrics in IT projects, through

which metrics are linked to functional levels of management and the multidimensional concept of project success. The focus is not on the development of a new experimental model, but rather on the synthesis of existing knowledge through the identification, classification, and comparison of metrics used in the most widely adopted agile frameworks, including Scrum, Kanban, Lean, Scaled Agile Framework (SAFe) and Dynamic Systems Development Method (DSDM).

Although existing approaches provide useful mechanisms for defining goals, organizing performance indicators, or monitoring value flow, they do not fully integrate project-specific context, managerial purpose, and expected value within a single metric-selection structure. This limitation may lead to fragmented or overly generic metric selection, particularly when the same indicators are applied across projects with different objectives, maturity levels, and organizational conditions. The proposed CPV (Context–Purpose–Value) matrix addresses this gap by linking the project context with the purpose of measurement and the value expected from the selected metrics.

In line with this objective, the paper addresses the following research questions:

- RQ1: How can agile metrics be systematically classified according to measurement domains and managerial purposes in IT projects?
- RQ2: How do different groups of agile metrics contribute to the multidimensional assessment of IT project success?
- RQ3: How do the selection and interpretation of agile metrics vary across widely used agile approaches, including Scrum, Kanban, Lean, SAFe, and DSDM?

The remainder of the paper is organized as follows. Section 2 reviews related work on agile approaches, metric classification, selection criteria, and application risks. Section 3 describes the study design, mapping procedure, and internal validation. Section 4 presents the results, including measurement maturity, metric mappings, the Balanced Agile Metrics Set, the analytical framework, and the CPV matrix. Section 5 illustrates the framework through a scenario-based application. Sections 6 and 7 discuss the findings and limitations and present the conclusions and future research directions.

2. Related Work

Efficiency represents an important, but not sufficient, component of overall IT project success. While efficiency primarily refers to the operational ability of a team to deliver results with rational use of resources, project success encompasses a broader range of business, user, and long-term organizational outcomes. In this paper, project success is conceptualized through multiple complementary dimensions that enable an integrated connection between agile metrics and actual project outcomes.

Project management generally involves planning, organizing, executing, and monitoring activities required to achieve project objectives. Given the frequent changes in user requirements, system architectures, and business priorities, IT project management often requires an organization of work that enables flexibility, iterative development, and rapid decision-making [15]. Efficient IT project management refers to the ability of the project team and management to achieve planned objectives in a timely, economically justified, and technically sustainable manner, while making optimal use of available resources, including time, budget, human capacity, and organizational knowledge. These resources are of major importance and apply to both traditional and agile management approaches. However, their management in IT projects differs significantly because of greater uncertainty, technical complexity, and the rapid pace of change.

2.1. Agile approaches to IT project management

Traditional approaches, such as the Waterfall model, assume a predefined scope, long-term planning, and formalized processes. They emphasize control, documentation, and predictability. However, when uncertainty is common, their application may be inefficient, given their reduced capacity to adapt to changing requirements, emerging technologies, or shifts in business priorities.

Agile approaches emerged as a response to the limitations of traditional models. They enable frequent delivery, rapid feedback, and continuous adaptation to change. As a result, the agile context places emphasis on

team collaboration, iterative development, user involvement, and continuous improvement [16], [17].

Agile and traditional approaches differ primarily in dimensions such as collaboration, iterative approach, user involvement, and continuous improvement, in contrast to control, documentation, and predictability [16], [18], [19], [20]. These differences clearly indicate that the agile approach is more suitable for IT projects where requirements frequently change.

2.2. Agile metrics

Research shows that continuous monitoring of performance and risks is essential for timely corrective actions and alignment with business objectives [21], [22], [23]. In this context, metrics play a central role in planning, monitoring, process improvement, and decision-making in an agile environment [2], [3], [24].

2.2.1. Purpose

Metrics are measurable indicators that objectively describe the state or behavior of a process, product, system, or team. Agile metrics include a set of quantitative and qualitative indicators that enable continuous monitoring of progress, stability, quality, and delivery value in projects guided by agile principles. They enable monitoring of project execution and early detection of potential issues [25], systematic improvement of processes and quality through feedback across iterations [2], and enhanced visibility for management together with better alignment of delivery with business objectives [26].

Common agile measurement practices include Velocity and visualization techniques such as Sprint Burndown and Release Burnup charts, which support the monitoring and forecasting of iteration and release progress [27]. These metrics do not always capture task complexity, variations in team members' experience, or the quality of delivery, though. For this reason, it is recommended to combine them with flow, quality, and value metrics in order to achieve a more balanced evaluation of performance.

Due to the diversity and complexity of IT projects, agile metrics cannot be treated as a universal set of

indicators applicable to all projects and organizations. Their relevance depends on project objectives, the nature of the product, team maturity, and the organizational context.

Metrics themselves are not goals, but instruments for understanding and improving systems and processes. They support more realistic planning of sprints and timelines and help reduce the risk of delays [2], [9], [28]. In addition, they provide teams and stakeholders with a shared view of project status, including project visibility and delivered business value, thereby increasing trust and facilitating decision-making [2], [24], [29]. Table 1 presents some of the main goals of agile metrics in IT projects.

Table 1. Goals of agile metrics.

Goal	Primary effect	References
Planning and estimation	Improved predictability of timelines and scope	[2], [9], [28]
Progress monitoring	Increased visibility for teams and stakeholders	[2], [9], [30]
Product quality	Lower defect rates and improved reliability	[2], [28], [31]
Team productivity and capacity	Optimization of workload and workflow	[25], [30], [32]
Business value and satisfaction	Focus on value rather than output only	[9], [24], [26]

Agile metrics support an objective assessment of project performance, particularly in monitoring progress and product quality [25], [28], [33]. They enable teams to detect deviations from planned objectives, improve delivery quality, and adjust development activities accordingly.

In organizations that invest in continuous improvement, metrics become an instrument for identifying process weaknesses, detecting bottlenecks, and optimizing value flow [34]. Continuous monitoring and analysis of metrics enable teams to identify potential challenges in advance, improve delivery quality, and more objectively assess the team's ability to achieve planned objectives. In this way, metrics act as a bridge between organizational strategic goals and the operational work of teams, supporting decision-making

based on reliable data rather than subjective judgments. Therefore, agile metrics in IT projects are not merely measures, but instruments for learning, guidance, and demonstrating value. In this context, it is important to consider key performance indicators (KPIs), which represent a subset of metrics designed to capture the most critical indicators of success. They can be viewed as strategically important metrics that connect operational activities with business objectives [31], [35]. KPIs are used to monitor the most important project goals, where each KPI should have a clearly defined purpose, measurement method, data source, and assigned responsibility. To avoid terminological inconsistency and confusion, this paper uses the term metric as the general concept, while KPIs are treated as a specific subset.

In practice, the emphasis is often placed on technical aspects of work, while social and innovative dimensions remain underrepresented. Attributes such as collaboration, trust, creativity, and openness to new ideas are weakly represented in existing metrics and are difficult to measure reliably [36]. A similar issue arises when assessing the ability of teams to adapt to change and effectively solve complex problems [36], [37]. In addition, user satisfaction is frequently cited as a key measure of agile success, but in practice it is mostly evaluated through feedback and subjective assessments rather than through stable and consistent quantitative indicators [9], [29], [38].

2.2.2. Classification

The classification of metrics enables a clearer understanding of their purpose, facilitates the interpretation of results, and provides a structured framework for their consistent application across different phases of the project lifecycle. It establishes the basis for analyzing how different groups of metrics influence various dimensions of performance, quality, and delivered value.

There is no universally accepted classification of agile metrics. Different authors propose categorizations depending on the methodology, project objectives, and team maturity.

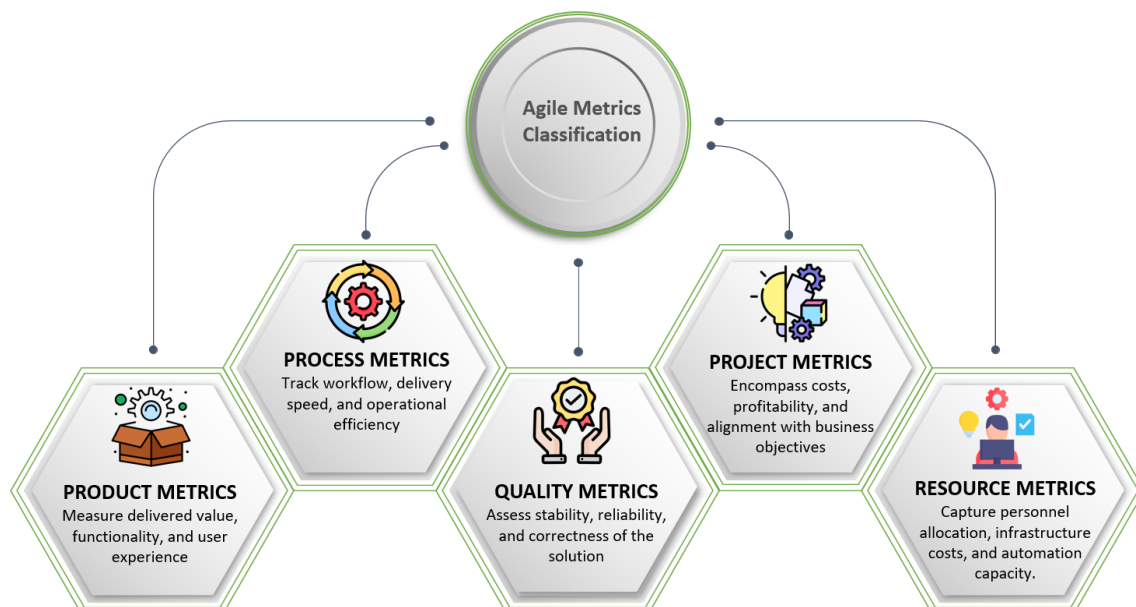


Figure 1. Classification of agile metrics.

For example, some studies classify metrics used in Scrum implementations according to activities associated with Sprint Planning, the Daily Scrum, the Sprint Review, and the Sprint Retrospective [39]. Such classifications describe measurement practices adopted within Scrum environments rather than metrics formally prescribed by the Scrum Guide. Another common classification includes metrics related to product quality, resources and costs, as well as process performance, with a focus on the team, management, and the user [3], [24]. There are also metrics that relate to technical aspects of code, such as complexity, cohesion, and coupling, which can help distinguish agile from non-agile development approaches [40]. In addition, some studies highlight the importance of custom metrics defined by agile teams themselves in order to better track their specific needs during sprints and software development processes [2], [41]. Additionally, agile metrics are often classified according to the domain they measure, dividing them into product, process, quality, project, and resource metrics [42], [43].

The classification presented in Figure 1 is adopted in this paper because it enables teams and organizations to clearly identify the focus of measurement, link metrics to specific objectives, and avoid a fragmented or one-sided approach to performance evaluation. Each category includes specific metrics that provide insight into key indicators of progress, stability, and efficiency, thereby

supporting planning, monitoring, and optimization of agile work. Except for widely established abbreviations such as Net Promoter Score (NPS), Return on Investment (ROI) and Work in Progress (WIP), the category-specific abbreviations introduced in Tables 2–6 are internal identifiers used in this paper and should not be interpreted as standardized framework abbreviations.

Product metrics include measures of delivered functionality, scope, user-perceived value, and financial outcomes [43]. These metrics evaluate what is delivered to the end user and connect product outputs with broader user and business outcomes.

Product metrics assess whether the delivered solution provides the intended functionality and value. Their primary purpose is to determine whether the product meets user expectations, supports the intended business objectives, and justifies continued development [42].

Table 2 presents examples of product metrics and measures together with concise descriptions. The abbreviations introduced in the table are used consistently throughout the proposed CPV framework and support its practical application. Count-based measures, such as delivered capabilities or functionalities, support the monitoring of delivery scope and pace. Perception-based measures, such as user satisfaction and NPS, provide insight into perceived value and product acceptance, while ROI reflects financial outcomes.

Table 2. Examples of product metrics and measures.

Product metric or measure	Abbr.	Description
Delivered Capabilities or Functionalities	PDC	Number of completed and accepted capabilities or functionalities; indicates delivery scope and pace
Delivered Value Points	PDP	Value points assigned to delivered items; provides a relative estimate of expected value
Level of User Satisfaction	PLS	User satisfaction score for delivered functionality, value, and quality
Net Promoter Score	NPS	Percentage of promoters minus percentage of detractors; indicates user loyalty
Return on Investment	ROI	Net return relative to investment cost; indicates financial value
Sprint Goal Success	PSG	Percentage of Sprints in which the Sprint Goal was achieved; indicates consistency in delivering intended Sprint outcomes.

Product metrics also have limitations that require contextual interpretation. The number of delivered capabilities or functionalities may create an illusion of progress if low-value items are prioritized. Delivered Value Points represent assigned estimates rather than realized value, while ROI is often difficult to determine accurately during the early stages of innovative or exploratory development. Similarly, PLS and NPS depend on subjective responses, sample composition, and the method used to collect feedback. Despite these limitations, combining delivery, perception-based, and financial measures provides a more comprehensive view of product performance and supports informed decisions about future development.

Process metrics focus on how value and user outcomes are delivered [43]. While product metrics answer the question of what has been delivered, process metrics explain how delivery was achieved. They provide insight into workflow efficiency and process stability, helping teams identify constraints, bottlenecks, excessive waiting times, and uneven workload distribution. These conditions directly affect delivery predictability and efficiency [42].

Table 3 presents commonly used process-related metrics and measures together with concise descriptions. The definitions of Cycle Time, Work Item Age, Work in Progress, and Throughput follow the official Kanban

Guide [44], while the definitions of the six SAFe Flow Metrics follow current official SAFe guidance [13]. Story Points and Velocity are included because they are widely used in agile and Scrum-based implementations [2], while Story Completion Rate is included as a derived measure of plan completion. These measures represent complementary planning and forecasting techniques rather than metrics prescribed by Scrum. The Scrum Guide does not prescribe a mandatory metric set and leaves context-sensitive tactics and measurement practices outside the core framework [45].

Table 3. Examples of process metrics and measures.

Process metric or measure	Abbr.	Description
Story Points	RSP	Relative size of a backlog item based on complexity, effort, and uncertainty
Velocity	RVL	Total story points completed per iteration; supports short-term planning
Story Completion Rate	RSR	Percentage of planned stories completed within an iteration or release
Lead Time	RLT	Elapsed time from a defined request or commitment point to delivery
Cycle Time	RCT	Elapsed time from the start to completion of a work item
Work Item Age	WIA	Elapsed time from the start of an unfinished item to the current measurement point
Work in Progress	WIP	Number of work items started but not finished
Throughput	RTP	Number of work items finished per unit of time
Flow Distribution	RFDI	Proportion of each backlog-item type in the workflow
Flow Velocity	RFV	Number of backlog items completed within a defined period
Flow Time	RFT	Elapsed time from workflow entry to release
Flow Load	RFL	Total work in progress across all workflow steps
Flow Efficiency	RFE	Active work time as a percentage of total Flow Time
Flow Predictability	RFP	Planned versus actual business value
Takt Time	RTT	Available working time divided by customer demand; indicates the required delivery cadence
Deployment Frequency	RDF	Number of production deployments per unit of time

Certain process-related measures capture similar aspects of workflow from different perspectives. RLT and RCT cover different segments of delivery duration, WIA indicates how long an unfinished item has remained in progress, and WIP represents the current workflow load. RSP represents relative work size, RSR reflects completion against the planned scope, while RVL and RTP express delivery pace using different units. RTT indicates the delivery cadence required to meet demand. These differences support complementary interpretation rather than redundant use of measures.

Process metrics provide insight into the operational dynamics of agile teams. Metrics such as RLT, RCT, WIA, WIP, and RTP support flow-based analysis by enabling objective assessment of workflow and delays. However, their interpretation remains context-dependent. For example, high RFE may indicate reduced waiting time, but it does not by itself demonstrate that the team is delivering complex or high-value work.

RSP is based on team-specific estimates, while RVL is derived from those estimates. Consequently, neither measure is suitable for direct comparison across teams or organizations. An increase in RVL does not necessarily indicate an increase in delivered value, as it may result from changes in estimation practices or a greater focus on lower-complexity items.

RDF indicates operational delivery cadence, while the Continuous Delivery Pipeline represents an enabling delivery capability rather than a metric [46]. The pipeline supports automated building, testing, and deployment, but its performance should be assessed through specific flow, quality, and deployment indicators. Frequent delivery alone does not guarantee high value or quality, as teams may release low-impact changes without meaningful improvement in user experience.

Sprint Burndown Charts, Release Burnup Charts, and Cumulative Flow Diagrams support process transparency, progress monitoring, and the identification of workflow deviations. However, they are treated here as visualization techniques derived from process data rather than as separate process metrics.

Taken together, these observations indicate that process metrics provide a detailed and objective view of workflow efficiency and stability. However, they enable a balanced understanding of agile project success only

when combined with product and quality metrics. Their diagnostic value is particularly evident in identifying bottlenecks, delays, and uneven workload distribution.

To avoid terminological overlap, this paper distinguishes the metrics according to their measurement boundaries and framework-specific use. Lead Time begins at a defined request or commitment point and ends with delivery, whereas Kanban Cycle Time begins when work on an item starts and ends when it is finished. Work Item Age applies only to unfinished items and extends from the start of work to the current measurement point. Although the Kanban Guide allows Cycle Time to be referred to as Flow Time when equivalent boundaries are used [44], this paper reserves the term Flow Time for the SAFe metric covering the period from workflow entry to release [13]. Similarly, Throughput and Flow Velocity, as well as Work in Progress and Flow Load, are conceptually related but reflect framework-specific terminology whose units, scope, and measurement boundaries must be explicitly defined. The abbreviations assigned to the six SAFe Flow Metrics are used internally in this paper and are not prescribed by official SAFe guidance.

Quality metrics represent key indicators of the stability, reliability, performance, and technical conformance of delivered software solutions. Unlike process metrics, which describe how work is performed, quality metrics focus on the technical characteristics of the resulting solution and the extent to which it meets functional, technical, and user requirements.

In agile and DevOps environments, continuous quality improvement is supported by integrating development, testing, and operational activities into the value delivery process, together with close team collaboration and the early involvement of quality assurance activities [42]. This approach enables earlier identification of issues and the implementation of corrective actions before defects become systemic.

Quality metrics provide a basis for assessing the technical sustainability of delivered solutions. They enable teams to identify declining quality, increasing defect rates, growing technical debt, and performance issues that may negatively affect the user experience. Table 4 presents commonly used quality metrics and measures together with concise definitions.

Table 4. Examples of quality metrics and measures.

Quality metric or measure	Abbr.	Definition
Rework Rate	QRR	Percentage of completed items returned for correction or additional work
First-Time Pass Rate	QFP	Percentage of tested items that pass validation on the first attempt
Delivery Quality	QDQ	Frequency and severity of defects identified after delivery
Defect Count	QDC	Number of defects identified within a Sprint, release, or defined period
Test Coverage	QTC	Percentage of code, requirements, or functionalities covered by tests
Standard Violations	QSV	Number of detected violations of coding, architectural, or design standards
Number of Tests	QNT	Number of defined or successfully executed tests; indicates testing volume
Defect Density	QDD	Number of defects per defined unit of software size
Response Time	QRT	Time required for the system to respond to a request; indicates performance
Technical Debt	QTD	Estimated effort or cost required to address accumulated technical deficiencies
Quality Blockers	QNB	Number of unresolved quality issues that prevent a work item, build, or release from progressing or being accepted

The interpretation of quality metrics requires caution and contextual understanding. For example, QRR may indicate problems in requirements definition, communication with users, or implementation quality. However, a high QRR does not necessarily indicate poor team performance because rework may also result from changing business priorities or evolving requirements. Therefore, this metric should be interpreted in combination with process and product metrics.

High values of QTC and QNT do not necessarily guarantee high quality if the tests are not relevant or do not cover critical scenarios. Similarly, a high QFP may indicate a stable implementation process, but it may also result from insufficiently demanding tests or a limited validation scope. QDD depends on the selected unit of software size, which may vary across teams, technologies, and architectural approaches. QTD also

depends on the method used to identify and estimate technical deficiencies.

QRT connects technical quality with user experience by indicating whether the system provides an acceptable level of performance. In this sense, quality metrics link the internal technical characteristics of the system with the users' external perception of product quality and value. Also, QNB is classified as a quality-related measure because it includes only blockers caused by unresolved defects, failed tests, or other quality issues, rather than all impediments affecting workflow.

Project metrics provide an integrated view of project performance from the perspective of costs, resource expenditure, and expected business value. Table 5 presents commonly used project-related metrics and measures in the agile context.

Unlike product, process, and quality metrics, which focus on specific dimensions of delivery, project metrics are positioned within a broader managerial and strategic context. They support cost control, resource planning, value assessment, and management-level decision-making [42]. They are particularly important for project managers and senior management because they support the evaluation of project viability, alignment with business objectives, and sustainability of resource use.

OTC and OOC provide insight into the administrative, coordination, and supporting costs of a project or program. Their interpretation requires caution. Higher management and related costs do not necessarily indicate inefficiency but may result from increased project complexity, regulatory requirements, or the need for more intensive coordination among teams and stakeholders.

OBR is a widely used management-level metric, but it may lead to misleading conclusions when considered in isolation. A project may have a stable OBR while delivering limited business value. Conversely, increased expenditure during the early stages may be justified if it enables faster delivery of high-value functionalities or reduces long-term risks. OPA is included as an assessment basis rather than a directly calculated metric. It defines the criteria according to which project value is interpreted and therefore supports the contextual interpretation of financial and delivery indicators.

Table 5. Examples of project metrics and measures.

Project metric or measure	Abbr.	Definition
Total Cost Estimate	OEC	Estimated total cost of developing a product or service, including all relevant project activities
Total Program Management Costs	OTC	Costs associated with program governance, coordination, reporting, and other management activities
Other Program-Related Costs	OOC	Additional program costs, such as travel, regulatory compliance, or specific operational requirements
Burn Rate	OBR	Amount of project expenditure per unit of time; indicates the rate of budget consumption
Guiding Principles of Value Assessment	OPA	Agreed criteria for assessing project value, such as cost savings, user satisfaction, workload reduction, and technical footprint

Project metrics represent the strategic level of measurement because they connect technical execution with business and financial outcomes. OEC and OBR support budget control and the timely identification of potential cost overruns. However, their interpretation is limited by the fact that project scope and priorities frequently change during development, making long-term forecasting and fixed cost estimates less reliable.

Resource metrics provide insight into the use of human, technical, and infrastructural resources in IT projects. Table 6 presents commonly used resource metrics and measures in agile environments. These metrics cover personnel costs and allocation, development cost distribution, infrastructure expenditure, and automation capacity. Unlike metrics that assess workflow, quality, or delivered value, resource metrics focus on the resources used, their allocation, and the associated costs. They include measures of personnel and infrastructure costs, resource distribution, development cost allocation, and the use of automation capabilities. These metrics support capacity planning, cost optimization, and the timely identification of resource-related issues throughout the project life cycle [27].

Although some resource metrics measure related aspects of resource expenditure at different levels of granularity, their combined use provides a more detailed understanding of the project cost structure. SPC represents the aggregate infrastructure and software cost,

whereas SLC, SCC, SBC, and SSC provide its individual components. These component metrics should not be added to SPC again when calculating resource costs.

Table 6. Examples of resource metrics and measures.

Resource metric or measure	Abbr.	Definition
Agile Team Cost	STC	Personnel cost of agile development teams over a defined period, determined by team size, composition, and compensation
Total Hardware, Software, Cloud, and Licensing Costs	SPC	Aggregate estimated or actual cost of IT infrastructure, cloud services, software, and required licenses
Allocation of Development Costs	SAC	Distribution of development costs among activities such as defect correction, feature implementation, and refactoring
Percentage of Resources by Function	SPF	Percentage distribution of personnel across functional roles, such as developers, designers, testers, and engineers
Software Licensing Fees	SLC	Cost of software licenses, subscriptions, and cloud-based applications
Computing Costs	SCC	Cost of using computing infrastructure, including on-premises and cloud computing resources
Bandwidth Costs	SBC	Cost of network bandwidth and data transfer
Storage Costs	SSC	Cost of data storage in on-premises or cloud environments
Share of Automated Activities	SAA	Percentage of eligible testing or delivery activities performed using automated tools

The application of resource metrics requires careful and context-aware interpretation. For example, STC may vary considerably depending on labor market conditions, team experience, and task complexity. High personnel costs do not necessarily indicate low efficiency but may reflect the need for specialized expertise in complex projects. Similarly, SPC depends on procurement models, contractual arrangements, and architectural decisions. Consequently, changes in SPC do not necessarily reflect changes in team performance.

SAC provides deeper insight into the distribution of development costs across different activities but may

produce misleading conclusions when interpreted in isolation. For example, a high proportion of costs associated with defect correction does not necessarily indicate low productivity but may result from accumulated technical debt, legacy-system constraints, or changing requirements.

Infrastructure-related costs, including SCC, SBC, and SSC, largely depend on system architecture, scaling strategy, workload, and testing intensity. An increase in these costs is not necessarily a negative signal, as it may accompany growth in the number of users, increased business demand, or stricter performance and security requirements.

SAA represents the proportion of eligible testing and delivery activities performed using automated tools. However, a high level of automation does not guarantee quality if automated tests and procedures are irrelevant, outdated, or do not adequately cover critical functionalities. Therefore, SAA should be interpreted together with process and quality metrics to provide a balanced assessment of automation effectiveness.

2.3. Selection and application of agile metrics

When selecting metrics, teams should ensure that they are simple, clearly defined, based on objective and accessible data, and aligned with business objectives [42]. Metric selection should also reflect project goals and adherence to agile principles [47]. Because teams differ in their priorities and working practices, the

relevance of individual metrics depends on the specific project context. Metrics should therefore be adopted because they provide valid and useful insights into processes and outcomes, not merely because they are widely used.

2.3.1. Selection criteria

Incorrectly selected metrics can produce undesirable effects, such as prioritizing speed over quality or focusing on measurability rather than actual value. For example, treating output or planning measures, such as team velocity or the number of completed user stories, as goals in themselves rather than as indicators of process state can encourage dysfunctional behavior. Such use of metrics may promote faster delivery at the expense of quality, potentially increasing defect density and technical debt. This may produce short-term performance gains while undermining the long-term reliability and sustainability of the solution [7], [28]. Teams may also be diverted from their core objectives by optimizing easily measurable elements rather than outcomes that provide meaningful value [48], [49].

Table 7 presents key criteria for the effective selection of agile metrics. Effective selection requires a balance between precision and practicality, together with clear prioritization of what should be measured. The relative importance of these criteria may change as projects evolve and organizations mature. Nevertheless, they provide a foundation for aligning metric selection with project goals and context.

Table 7. Criteria for effective selection of agile metrics.

Criterion	Description	Consequences of neglect
Relevance to objectives	The metric must align with key project objectives (e.g., value, quality, stability, etc.)	Measurement loses its purpose and becomes administrative reporting
Contextual suitability	Metric selection must consider project type, team size, domain, and culture	Metrics become inapplicable or counterproductive and may undermine agile principles
Clarity and understandability	Metrics should be easily understood by all team members and stakeholders	Misinterpretation and loss of trust in measurement
Measurability and data availability	Data required for the metric must be reliable, consistent, and easily accessible	Increased team burden and reduced accuracy of measurement
Actionability	Measurement results should support concrete decisions and corrective actions	Metrics do not influence behavior and do not contribute to improvement
Resistance to manipulation	The metric should minimize the possibility of gaming or artificial improvement of results	Optimization of numbers instead of real value and quality
Balanced set of metrics	Metrics should be used in combination to cover multiple dimensions of success	One-sided optimization and distorted view of project performance

The process should begin with a clear definition of project objectives and priorities. In some contexts, delivery speed and flexibility are the primary concerns, whereas in others the focus is on quality, security, regulatory compliance, or long-term architectural sustainability.

In addition to objectives, the implementation must consider the broader project context, including the nature and complexity of the product, as well as the maturity of the team and organization. Complex, distributed, or safety-critical systems require different metrics from smaller, incremental projects. Similarly, teams at lower levels of maturity typically begin with basic flow and delivery metrics, while more mature organizations adopt advanced metrics related to value, stability, and predictability. Agile teams commonly use metrics such as RVL, RCT, QDD, and QTC to improve workflow and product quality, while ensuring alignment with client and organizational goals [32]. Research indicates that metric systems tailored to the specific context of a project and organization contribute to improved transparency, team motivation, and continuous process improvement [2], [3], [4]. This implies that without proper alignment with objectives and context, there is a risk that metrics will not reflect actual project priorities, which may lead to inefficiencies or incorrect decisions [24], [50].

2.3.2. *Limitations and risks of application*

Risks in applying agile metrics are often associated with limited transparency, lack of standardization, and constrained precision, which may lead to misinterpretations and inadequate decision-making within teams [28]. Challenges include difficulties in data collection, limited relevance of metrics for specific contexts, and issues in their calculation, all of which can reduce their practical value and applicability [14]. Incorrect selection or inappropriate use of metrics may encourage dysfunctional behaviors, undermine agile culture, and shift team focus toward short-term targets instead of real value and continuous improvement [51]. Furthermore, relying on quantitative metrics without considering qualitative factors and context may result in superficial conclusions and increase the risk of poor planning and project management decisions [2], [52].

A common issue is metric fixation, where focus on a single metric leads to optimization at the expense of overall value and quality. Related risks include local optimization and metric gaming, where improvements in numbers do not reflect real performance gains. Also, many agile metrics are context-dependent and unsuitable for cross-team comparison. Misuse may reduce complex aspects such as user satisfaction or team dynamics to simplified indicators. Therefore, metrics should guide learning and improvement rather than control.

Ethically, metrics must support trust and psychological safety. Using them for individual ranking undermines agile principles. Therefore, agile metrics should primarily be used as tools for learning, reflection, and improvement rather than as instruments of surveillance.

Table 8 systematizes the key limitations, risks, and ethical aspects of applying agile metrics. The table links common threats in the application of agile metrics with their underlying causes, affected success dimensions, and recommended mitigation approaches. These risks demonstrate that measurement may produce effects opposite to those intended if the focus is placed on numerical targets rather than on real value. The risks associated with the application of metrics reinforce the importance of the analytical interpretation framework and the balanced set of indicators.

Existing approaches address metric selection from different perspectives. The Goal–Question–Metric (GQM) approach links measurement to explicitly defined goals and questions [53], while Balanced Scorecard-based models organize performance indicators across complementary strategic dimensions [54]. SAFe measurement practices primarily support the assessment of business outcomes and value flow in scaled agile environments [55]. In contrast, the CPV matrix proposed in this paper integrates project context, managerial purpose, and expected value within a single selection structure, thereby providing a complementary mechanism for choosing and combining agile metrics across different project conditions.

Table 9 summarizes the main differences between the CPV matrix and the related measurement approaches discussed above.

Table 8. Limitations, risks, and ethical aspects of applying agile metrics.

Risk / threat	Mechanism of occurrence	Affected success dimension	Potential consequence	Recommended approach
Focus on a single metric (metric fixation)	Focusing on one metric without relational analysis	Project efficiency	Local optimization, decline in quality	Use a balanced set of metrics
Local optimization	Optimizing processes without considering the overall system	Future readiness	Accumulation of technical debt	Combine process and quality metrics
Metric gaming	Adjusting behavior to improve numerical results	Business success	Illusion of progress, loss of real value	Focus on trends and relational analysis
Comparison of non-comparable teams	Using contextual metrics for external evaluation	Efficiency and team dynamics	Unfair assessments, demotivation	Limit metrics to team-specific context
Neglect of qualitative aspects	Excessive reliance on quantification	User impact	Superficial assessment of satisfaction and quality	Include retrospectives and qualitative insights
Misinterpretation of trends	Analysis of absolute values without context	All dimensions	Incorrect management decisions	Apply temporal dimension in interpretation
Excessive measurement	Tracking a large number of KPIs without clear purpose	Organizational sustainability	Administrative burden	Clearly define the purpose of each metric
Individualization of measurement	Using metrics for ranking individuals	Psychological safety	Decline in trust and team cohesion	Focus on team rather than individual performance
Organizational immaturity	Mechanical application without analytical framework	Strategic dimension	Poor high-level decisions	Provide education and gradual metric adoption

Table 9. Comparison of related measurement approaches and the proposed CPV matrix.

Approach	Primary input	Selection logic	Decision focus	Main strength	Main limitation
Goal–Question–Metric (GQM)	Organizational or project goals and related questions	Metrics are derived from explicitly defined goals and questions	Goal-oriented measurement and evaluation	Provides a clear link between goals and measurable indicators	Does not explicitly integrate organizational maturity, project context, and multiple value dimensions
Balanced Scorecard	Strategic objectives across complementary performance perspectives	Indicators are organized according to strategic performance dimensions	Strategic and organizational performance	Provides a balanced view of financial and non-financial performance	Offers limited guidance for selecting specific agile metrics at operational and tactical levels
SAFe measurement practices	Business outcomes, value streams, flow, and competency assessment	Metrics are selected according to the structure and measurement guidance of SAFe	Scaled agile delivery and organizational agility	Provides framework-specific guidance for measuring flow and business agility	Primarily designed for organizations applying SAFe and may require adaptation in other agile contexts
Proposed CPV matrix	Project context, measurement purpose, and expected value	Metric groups are selected by jointly considering context, managerial purpose, and value dimensions	Operational, diagnostic, tactical, strategic, and learning decisions	Supports context-aware selection and combination of existing agile metrics	Requires contextual interpretation and expert judgement and has not yet been externally validated

The novelty of the CPV approach lies not in introducing new individual metrics, but in providing a context-aware decision mechanism for selecting and combining existing metrics. More specifically, the

framework supports decisions about which metric groups should be prioritized in a particular project context, which managerial purposes they should serve, which value dimensions must be balanced, and how the metric set should be adapted when organizational conditions change. GQM derives metrics from predefined goals and questions but does not explicitly incorporate project maturity, development stage, or competing value dimensions into the selection process. The Balanced Scorecard provides a strategic performance structure but offers limited guidance for selecting operational and tactical agile metrics, while framework-specific guidance, such as SAFe, is primarily applicable within its own organizational setting. As shown in Table 9, the CPV matrix complements these approaches by enabling context-dependent metric-selection decisions across different agile frameworks and management levels.

3. Research Method

3.1. Study design

This study follows a design-oriented synthesis that combines a structured analysis of the literature with conceptual mapping and framework development. The literature analysis was intended to support conceptual framework development rather than to constitute an exhaustive systematic review. Therefore, the objective

was to produce a context-aware, managerially useful method for selecting and interpreting agile metrics rather than to collect new primary empirical data. The workflow follows the steps illustrated in Figure 2.

The analysis draws on peer-reviewed articles, conference proceedings, practitioner guides, and official documentation for dominant agile frameworks (Scrum, Kanban, Lean, SAFe, DSDM), with additional reference to DevOps and continuous-delivery literature where relevant. Documents were included when they explicitly defined or operationalized agile metrics, discussed their purpose, interpretation, or managerial use, or reported empirical evidence concerning their behavior or effects.

The previously described agile metrics classification links measurement domains with management levels or functional purposes, providing the analytical basis for applying agile metrics. It emphasizes a deliberate, goal-oriented approach in which metrics are selected according to specific objectives, reducing the risk of fragmented and misleading use. Metrics were grouped into the five measurement domains used throughout the paper: Product, Process, Quality, Project, and Resource. Each metric was further tagged by primary managerial purpose (operational, diagnostic, tactical, strategic, or learning) to support mapping to management levels and to the CPV dimensions.

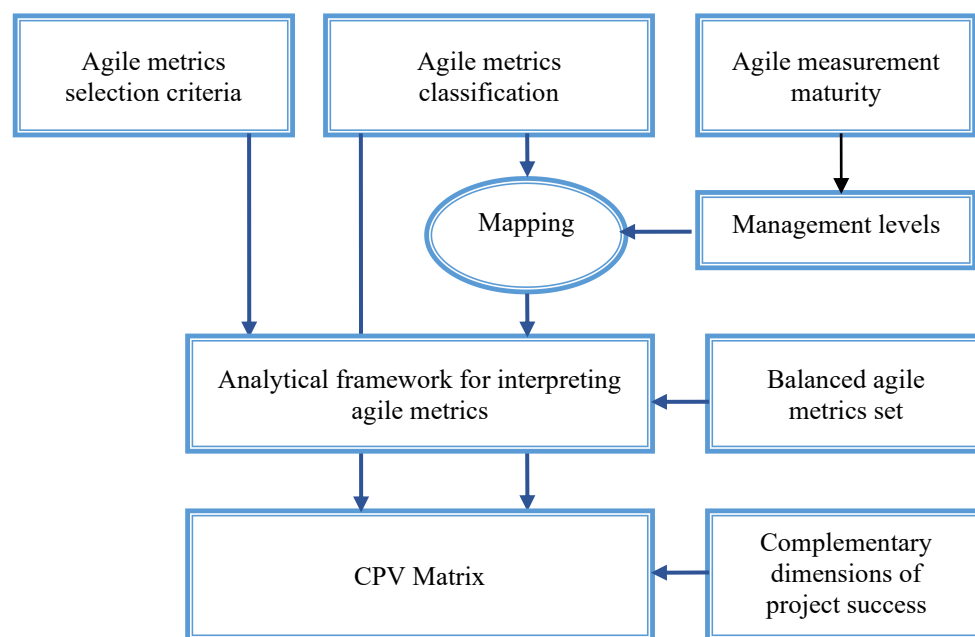


Figure 2. Research methodology.

3.2. Mapping procedure and CPV matrix development

Metric assignment followed a rule-based mapping procedure. First, each metric was assigned to one primary measurement domain according to the construct it directly measures. Product Metrics capture delivered functionality and user or business value; Process Metrics describe workflow, delivery speed, and process stability; Quality Metrics assess technical correctness, reliability, and testing outcomes; Project Metrics address project costs, expenditure patterns, value-assessment criteria, and overall project viability; and Resource Metrics capture human, technical, and infrastructural capacity or consumption. When a metric was relevant to more than one domain, its primary classification was determined by its direct measurement object, while secondary contributions were represented through subsequent mappings rather than duplicate classification.

Managerial purpose was assigned according to the decision most directly supported by the metric. Operational metrics describe current workflow, resource use, or delivery status; diagnostic metrics help identify causes of deviations and performance problems; tactical metrics support sprint, release, capacity, and medium-term planning; strategic metrics inform business value, investment, scalability, and long-term viability; and learning and adaptation metrics support trend evaluation, experimentation, and continuous improvement. A metric could contribute to multiple managerial purposes because its function may change with the decision level and application context. The strength of each relationship was therefore determined by the directness of the decision support provided.

Mapping to value dimensions followed a corresponding rule. Project Efficiency was assigned where metrics primarily reflected time, cost, throughput, resource use, or delivery predictability. User Impact included metrics related to satisfaction, usability, reliability, and perceived value. Business Success covered economic return, project viability, and contribution to strategic objectives. Future Readiness included technical sustainability, adaptability, automation, organizational learning, and long-term scalability. Direct contributions were classified as strong,

mediated or context-dependent contributions as moderate, and marginal contributions as limited.

Initial assignments were based on explicit metric definitions and documented applications in the reviewed sources. Wherever sources provided different classifications, priority was given to the construct directly measured by the metric, followed by evidence from empirical studies and official framework documentation. Ambiguous assignments were re-examined against the domain, managerial-purpose, and value-dimension rules defined above. The resulting mappings were then cross-checked for consistency between metric definitions, management levels, success dimensions, and the recommendations subsequently incorporated into the CPV matrix.

As an illustration of the mapping procedure, velocity (RVL) represents a potentially ambiguous metric because it can be used for operational monitoring and tactical planning, while it may also be misinterpreted as a direct indicator of productivity. Following the predefined mapping rules, RVL was assigned primarily to the Process domain because its direct measurement object is the amount of estimated work completed within an iteration, rather than product quality, user value, project cost, or resource consumption. Its principal managerial contribution was likewise mapped to operational and tactical purposes, as it supports monitoring of iteration-level delivery pace and short-term planning.

With respect to value dimensions, RVL was mapped most directly to Project Efficiency; however, this relationship was treated as context-dependent rather than as evidence of Business Success or User Impact. This distinction reflects the fact that an increase in velocity may result from changes in estimation practices, team composition, task decomposition, or the completion of lower-complexity work and therefore does not, by itself, demonstrate an increase in delivered value.

The metric definitions, classifications, and documented areas of application presented in the preceding sections were derived from the reviewed literature and agile framework documentation. In contrast, their integration into the CPV matrix and the proposed combinations of contexts, purposes, value dimensions, metric groups, and KPIs represent a

conceptual synthesis. These combinations should therefore be interpreted as context-dependent recommendations rather than empirically confirmed prescriptions.

The CPV matrix was constructed by integrating three mapped dimensions: project and organizational context, managerial purpose of measurement, and the value dimension to be prioritized. Contextual configurations were differentiated according to product-development stage, organizational maturity, regulatory and risk conditions, and the scale of agile implementation. For each configuration, the dominant objectives were identified first, followed by the primary measurement purpose and value dimension. Relevant metric groups and illustrative KPIs were selected only after these elements had been specified. The resulting recommendations were iteratively reviewed against the metric definitions, mapping rules, selection criteria, and identified application risks to ensure internal coherence and practical relevance.

3.3. Analytical techniques and internal validation

The study applied thematic synthesis to qualitative descriptions of agile metrics and relational analysis to their definitions, documented applications, managerial purposes, and contributions to project value. The analysis emphasized three complementary interpretive perspectives: contextual, relational, and temporal. The rule-based procedure described in the preceding subsection was used to structure the assignment of metrics to measurement domains, managerial purposes, and value dimensions.

Internal conceptual validation was conducted by the authors through successive consistency checks. Each assignment was reviewed against the metric definition, its direct measurement object, its documented application in the literature or official framework guidance, and the decision or value dimension it most directly supports. The resulting mappings were also cross-checked against the metric-selection criteria and application risks identified in Tables 7 and 8. Wherever classifications remained ambiguous, the assignment was re-evaluated using the predefined mapping rules, with priority given to direct measurement meaning and independently documented use.

This procedure supports the internal coherence and transparency of the proposed mappings but does not constitute independent external validation. No external expert panel, independent inter-rater assessment, or empirical evaluation of the CPV matrix was conducted. Consequently, the proposed mappings and metric combinations should be interpreted as conceptually grounded and transparent recommendations whose practical validity requires further assessment across different IT project contexts.

4. Results of the Conceptual Synthesis

The results presented in this section are outputs of the design-oriented synthesis and rule-based conceptual mapping described in Section 3. They should therefore be interpreted as conceptually derived classifications and framework propositions rather than as findings based on primary empirical data or external validation. The mappings and relative relationships reported below reflect the application of predefined domain, managerial-purpose, and value-dimension rules to the reviewed literature and framework documentation. Their purpose is to establish an internally coherent structure that is subsequently operationalized through the CPV matrix, while its empirical validity remains to be established through external evaluation.

4.1. Agile measurement maturity

The agile measurement system evolves with organizational maturity. Accordingly, this paper conceptualizes agile measurement maturity through four hierarchical levels, as illustrated in Figure 3.

At the operational, or descriptive level, measurement focuses on tracking activities and stabilizing workflow, with process metrics dominating and answering the question of what is happening within a sprint or delivery process. At the diagnostic, or analytical level, metrics are combined and analyzed through trends to understand the causes of deviations and systemic weaknesses. Value alignment represents a transition toward a transition toward tactical and strategic measurement, where product, project, and resource metrics are integrated to assess whether delivered activities generate actual business and user value.

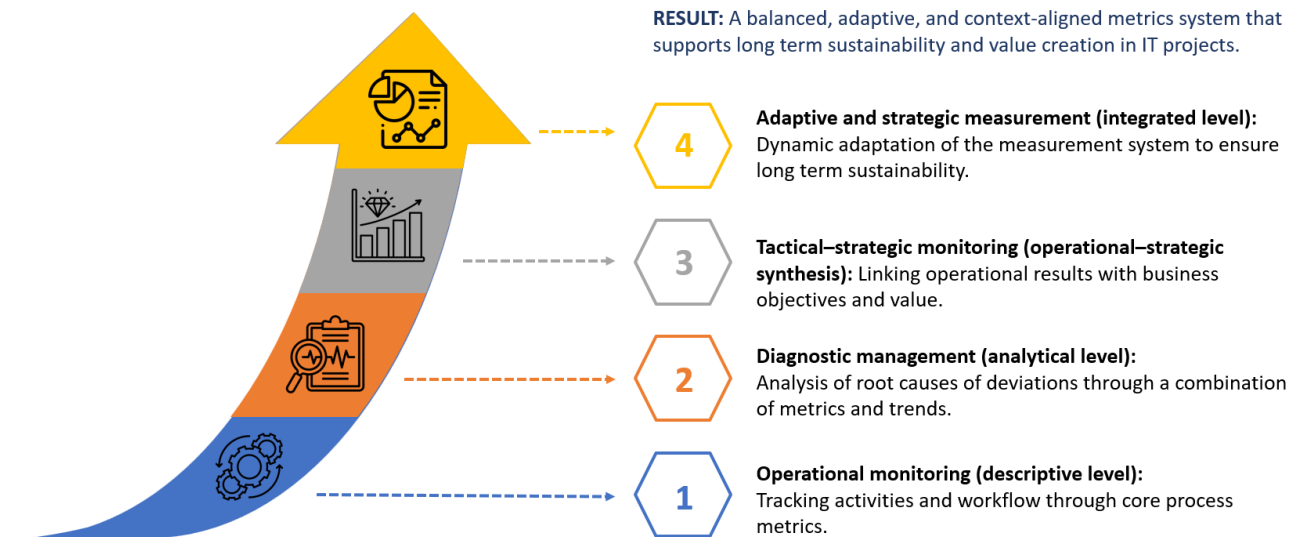


Figure 3. Hierarchy of agile measurement maturity.

At the highest, adaptive and strategic level, measurement becomes a dynamic instrument of organizational learning and long-term sustainability, with continuous adjustment of the metric set in response to changes in context and strategic priorities.

The proposed hierarchy suggests that an effective metric system is not a static set of indicators but an evolutionary framework that develops alongside the organization. As organizational maturity increases, the logic of selecting, interpreting, and combining metrics also changes. The four-level maturity hierarchy represents a conceptual synthesis proposed in this study and should be interpreted as a heuristic structure rather than an empirically validated maturity model.

Based on these maturity levels, we propose an additional classification of metrics. This perspective aligns metrics with project goals and development stages through the following management levels:

- **Operational:** Monitor task flow, process stability, and short-term progress, supporting transparency and rapid response.
- **Diagnostic:** Identify bottlenecks, deviations, and risks, supporting root cause analysis.
- **Tactical:** Support planning, coordination, and capacity balancing, linking operations with medium-term goals.

- **Strategic:** Assess business value, profitability, and long-term sustainability, connecting technical work with business outcomes.
- **Learning and adaptation:** Track trends over time, enabling retrospectives, experimentation, and continuous improvement.

This classification does not replace the previously presented classification, but rather complements it and provides an additional perspective on their application. Its key value lies in helping teams and organizations to select metrics in accordance with specific objectives, whether the focus is on operational efficiency, improved coordination, strategic alignment, or learning

4.2. Mapping metric groups to management levels

To further clarify the functional role of different groups of agile metrics, this section examines the strength of their association with operational, diagnostic, tactical, strategic, and learning and adaptation levels. This mapping shows how the same metric group can support different types of decisions and management purposes.

Table 10 presents the relationship between these two complementary perspectives. No metric group is exclusively tied to a single management level; rather, its relative influence and functional role vary depending on the measurement purpose, decision context, and manner of interpretation.

Table 10. Relationship between metric groups and management levels.

Metric group	Operational	Diagnostic	Tactical	Strategic	Learning and adaptation			
Product	Monitoring delivered functionalities and short-term delivery progress	Identification of insufficient value or misalignment with user needs	Evaluation of sprint and release goal achievement	Assessment of business value, ROI, and user satisfaction	Learning about user preferences and adapting future delivery			
Process	Monitoring workflow and process stability in real time	Detection of bottlenecks, delays, and inefficiencies	Capacity planning, flow optimization, and work distribution	Assessment of process maturity and organizational agility	Process improvement through retrospectives and experimentation			
Quality	Monitoring defects and delivery stability per iteration	Identification of root causes of defects and quality degradation	Balancing delivery speed and quality	Ensuring long-term sustainability and reducing technical debt	Learning about the impact of testing practices and engineering decisions			
Project	Monitoring expenditure and budget status	Identification of cost deviations and unusual spending patterns	Management of cost estimates, expenditure, and value-assessment priorities	Assessment of project viability and strategic justification	Improvement of estimation, budgeting, and decision-making practices			
Resource	Monitoring personnel allocation and infrastructure expenditure	Identification of excessive consumption or resource misalignment	Optimization of resource allocation and capacity	Long-term planning of budget and infrastructure	Learning about organizational efficiency and maturity			
				Influence (shaded)	Low	Middle	Strong	

Product metrics show the strongest influence at the strategic level of management, as they directly reflect delivered business value, user satisfaction, and the overall viability of development efforts. Their application enables management to assess whether the developed solution justifies the investment and contributes to long-term organizational goals. At the same time, these metrics have a moderate influence on operational and tactical purposes, as they provide feedback to teams and project managers regarding the achievement of release objectives and the focus on value. Their diagnostic and learning roles are more limited, as they primarily describe outcomes rather than the underlying causes of process issues.

Process metrics have a strong influence on operational, tactical, and diagnostic purposes, making them a central instrument of day-to-day agile management. At the operational level, they enable teams to monitor workflow, process stability, and respond quickly to deviations. At the tactical level, these metrics support sprint and release planning, capacity balancing, and workload management. Their diagnostic value is particularly evident in identifying bottlenecks, delays,

and variations in value flow. Process metrics also play a significant role in learning and adaptation, as they enable the monitoring of trends over time and support retrospectives and continuous improvement. Their strategic influence is present, although secondary, since process metrics do not directly measure business value.

Quality metrics have a multilayer functional role that varies across decision-making levels. At the operational level, metrics such as QDC, QFP, QSV, and QTC enable teams to monitor technical correctness and delivery stability on a daily basis within a sprint. From a diagnostic perspective, metrics such as QDD, QRR, and QNB support the identification of technical debt, recurring defects, and structural weaknesses of the system. At the tactical level, aggregated quality indicators enable the assessment of delivery stability at the release level and support capacity planning and prioritization between speed and quality. At the strategic level, quality metrics have an indirect but significant role, as long-term trends in defects, stability, and performance influence architectural sustainability, maintenance costs, and product reputation. Finally, in the context of learning and adaptation, trends in quality metrics provide a foundation for retrospectives and

continuous improvement of development practices, with a focus on changes over time rather than on individual values.

Project metrics primarily operate at the tactical and strategic levels, where they support cost monitoring, expenditure analysis, value assessment, and evaluation of project viability. At the tactical level, they support decisions concerning priorities, budget use, and plan adjustments. At the strategic level, they connect technical execution with business objectives and investment decisions. Cost deviations or unusual spending patterns may also indicate issues in planning, coordination, or project scope. Trend analysis supports improved estimation, budget management, and decision-making in future agile initiatives.

Resource metrics primarily support tactical and strategic purposes by enabling the management of capacity, costs, and long-term sustainability. At the tactical level, they support resource allocation, capacity planning, and team optimization, while at the strategic level, they contribute to the assessment of scalability and economic viability. Their operational role includes monitoring team workload and infrastructure use, whereas their diagnostic role involves identifying excessive consumption and resource misalignment. In the context of learning and adaptation, trends in resource metrics help organizations improve automation, optimize infrastructure and system architecture, and strengthen organizational maturity.

This mapping suggests that agile metrics do not belong exclusively to a single functional purpose, as their value depends on the context and level of decision-making. The relationship between metric groups and management levels further supports the need for clearly defined principles and guidelines for metric selection and application.

4.3. Balanced agile metrics set

Agile IT projects are characterized by a high degree of complexity, interdependence, and variability, which makes it difficult to adequately assess their success by relying on a single group of metrics. For this reason, we adopt the concept of a Balanced Agile Metrics Set, which involves the simultaneous monitoring of multiple

complementary dimensions of success. The concept is based on the integration of all previously discussed metric groups. The set combines quantitative indicators, such as Velocity, Cycle Time, and Throughput, with qualitative aspects, such as team satisfaction and business value, thereby improving transparency and accountability.

Balancing metrics prevents overemphasis on delivery speed or quality alone, which may otherwise lead to undesirable outcomes such as reduced system reliability or team demotivation [51], [56]. Instead of monitoring isolated symptoms, such as a decrease in velocity, a balanced set of metrics enables the identification of potential relationships and trade-offs, including the impact of technical debt on workflow, trade-offs between speed and quality, and the relationship between investment in automation and long-term delivery efficiency.

The additional value lies in its applicability across different levels of decision-making. At the operational level, it helps teams optimize workflow and delivery quality. At the tactical level, it supports more effective planning of sprints and releases. At the strategic level, it provides management with a reliable basis for assessing project viability and long-term sustainability. In this way, metrics are positioned as tools for learning and guidance rather than mechanisms of control. Consequently, the use of a balanced set of metrics supports continuous process improvement, team motivation, and the maintenance of balance between development speed and product quality [2], [57].

A balanced agile metrics set is not a fixed or universal model. Its composition must be adapted to the project context, organizational objectives, and team maturity. Balance is not achieved through the number of metrics, but through their thoughtful selection and integrated interpretation. In this sense, the Balanced Agile Metrics Set represents a conceptual framework that helps organizations build a sustainable, meaningful, and context-aware measurement system in agile IT projects.

4.4. Framework for interpreting agile metrics

The application of metrics without a clearly defined analytical framework may lead to superficial or incorrect conclusions and reduce the value of decision-making

[58]. In addition, inadequate validation and the selection of metrics that do not reflect domain-specific objectives may result in inaccurate performance assessments and limit the practical applicability of the results [59]. Therefore, in addition to classifying metrics by measurement domain and functional purpose, a clear framework for their interpretation is required. Such a framework can guide metric selection, application, and evaluation, helping to identify potential pitfalls and support deeper and more reliable analysis [59], [60].

Because metrics have no unambiguous meaning outside their context, apparent improvements may be misleading. Higher velocity may reflect changed estimation practices, while a lower defect count may result from insufficient testing. Distinguishing genuine improvement from contextual or measurement-related artifacts therefore requires analysis across three complementary dimensions:

- **Contextual dimension:** interpretation of metrics in relation to project objectives, product characteristics, team maturity, and the organizational environment.

- **Relational dimension:** analysis of relationships among metrics, including the combined interpretation of process, quality, and value metrics to avoid local optimization.
- **Temporal dimension:** analysis of trends and patterns across multiple iterations rather than reliance on isolated observations.

This approach shifts the focus from control to learning and systemic improvement. Metrics serve as signals that help identify systemic issues, guide experiments, and improve team interactions rather than evaluate individuals or compare teams. They may be used descriptively to characterize the current state, diagnostically to examine potential causes, and prescriptively to guide subsequent actions. The framework thus links measurement with management by transforming data into actionable knowledge.

To reduce misinterpretation, Table 11 presents an operational framework linking metric selection, context, interpretation, and action. It emphasizes relational and temporal analysis to support systemic learning and continuous improvement rather than individual control.

Table 11. Operational analytical framework for the selection and interpretation of agile metrics.

Process	Purpose	Potential risk	Expected outcome
1. Definition of measurement objective	Identification of the priority dimension such as time, quality, value, or cost	Unclear objective leads to incorrect metric selection	Clearly defined measurement focus
2. Selection of metric groups	Combination of complementary groups such as process and quality	Reliance on a single group leads to local optimization	Balanced set of metrics
3. Mapping to decision level	Alignment of metrics with operational, tactical, or strategic objectives	Misalignment between metrics and management levels	Context relevant set of indicators
4. Contextual verification	Verification of changes in team, estimates, scope, and architecture	Misinterpretation due to contextual artifacts	Contextually informed interpretation
5. Trend analysis	Observation of metrics across multiple iterations	Conclusions based on isolated values	Identification of patterns and trends
6. Relational assessment	Comparative analysis of different metric groups	Illusion of progress such as increased velocity with declining quality	Identification of systemic relationships
7. Interpretation classification	Distinction between descriptive, diagnostic, and prescriptive use	Premature prescriptive conclusions	Clearly defined type of conclusion
8. Action definition	Selection of corrective action or experiment	Reactive or excessive interventions	Controlled experiment or adjustment
9. Feedback loop	Evaluation of effects in subsequent iterations	Lack of learning and adaptation	Organizational learning and adaptation

4.5. Mapping agile metrics to IT project success

One of the key questions is how measurement results should be interpreted in relation to actual project success. Measurement gains meaning only when linked to clearly defined performance dimensions. Therefore, this section connects agile metrics with the multidimensional concept of IT project success.

Some authors argue that traditional criteria are insufficient for evaluating project success and propose extended frameworks that include benefits for users and organizations, together with continuous evaluation throughout the project life cycle [61], [62]. Agile projects place additional emphasis on frequent evaluation and user satisfaction as important indicators of success [9], [38].

Figure 4 presents four complementary dimensions of IT project success as interrelated components of an integrated model. No dimension has inherent priority; their combined consideration supports a comprehensive and sustainable evaluation of project outcomes.

Project efficiency refers to the ability of a team to deliver functionality within acceptable time and budget constraints. This dimension is primarily supported by process and project metrics such as RLT, RCT, RTP, RVL, and OBR.

User impact emphasizes satisfaction, usability, and the perceived value of the delivered solution. Product and quality metrics, including PLS, NPS, QDQ, and

QRT, help assess the extent to which the product addresses user needs. However, their interpretation requires sufficient context and continuous feedback.

Business success refers to the economic viability of the project and its contribution to organizational strategic objectives. Product, project, and resource metrics and measures, such as ROI, OEC, SAC, and OPA, support evaluation of the relationship between investment and achieved benefits. In agile projects, business outcomes may be evaluated iteratively, although their precise quantification remains difficult because of evolving scope and incremental delivery.

Future readiness includes long-term sustainability, technical stability, organizational learning, and adaptability to change. It is supported by quality and resource metrics such as SAA, QTC, QDD, and QTD, but may be overlooked during the early stages of development.

No single metric group captures all success dimensions. Product and quality metrics drive user value, process and resource metrics support efficiency and sustainability, while project metrics link operations with business goals but require support from other metrics. Table 12 presents the mapping of agile metric groups to project success dimensions. This mapping further justifies the need for a balanced set of agile metrics, as IT project success is not a single measure but the result of the interaction of multiple dimensions.

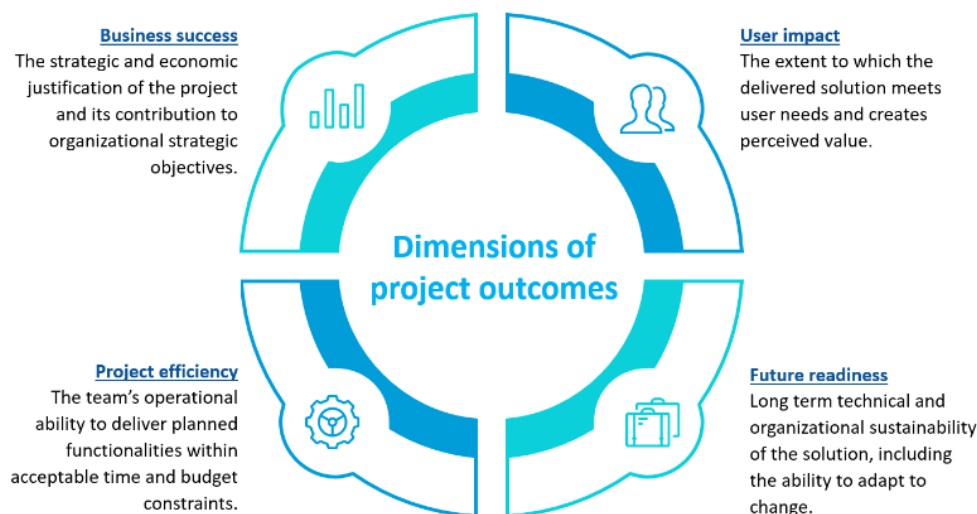


Figure 4. Complementary dimensions of IT project success.

Table 12. Agile metric groups across IT project success dimensions.

Metric group	Project efficiency	User impact	Business success	Future readiness
Product	●	●	●	●
Process	●	●	●	●
Quality	●	●	●	●
Project	●	●	●	●
Resource	●	○	●	●

- direct and strong contribution
- indirect or context dependent contribution
- limited or mediated contribution

Each success dimension depends on a combination of different metric groups, yet no single group can represent overall project success on its own. For example, when applying process and project metrics in the context of project efficiency, high process efficiency, such as low lead time or stable velocity, may conceal the delivery of low business value functionalities or a decline in quality. A strong focus on speed may also lead to the accumulation of technical debt. In the context of user impact, user satisfaction may be influenced by subjective factors or short-term effects, while a low number of defects does not necessarily imply high usability or relevance of the solution.

4.6. Metrics across agile approaches

Building on the classification of agile metrics and the proposed analytical framework, this section examines how metrics are applied across widely used agile approaches, including Scrum, Kanban, Lean, SAFe, and DSDM. It identifies the metrics and metric groups commonly associated with these approaches and analyzes their relationships with managerial purposes and dimensions of IT project success. Although these approaches share common agile principles, they differ in development focus and measurement practices, limiting the universal applicability of the same metrics [6], [63]. In large-scale agile environments, coordination and monitoring requirements increase, creating a need for metrics that may not be relevant to smaller teams. Agile teams therefore combine established metrics with context-specific measures reflecting their particular objectives and working practices [2], [3]. Flexibility in metric selection and application is consequently essential

for effective performance measurement in agile IT projects [28], [64].

Although the analyzed approaches share common agile principles, their measurement practices differ in focus and interpretation of project success. Scrum, Kanban, Lean, SAFe, and DSDM vary in structure, scope, and organizational application. Scrum primarily supports iterative product delivery at the team level, while Kanban focuses on workflow management and can be applied at different organizational levels. Lean emphasizes value optimization and waste reduction, SAFe explicitly addresses coordination across multiple teams and portfolios, and DSDM provides a structured approach to agile project delivery. The adoption or combination of these approaches depends on project complexity, organizational scale, governance requirements, and specific delivery needs.

Table 13 shows that agile approaches differ not only in procedures but also in measurement logic and value priorities across project efficiency, user impact, business success, and future readiness.

Table 13. Integration of agile frameworks, metric groups, and success dimensions.

Agile framework	Dominant metric groups	Primary functional purposes	Primary success dimensions
Scrum	Process Product	Operational Tactical	● Project efficiency ● Business success
Kanban	Process	Operational Diagnostic	● Project efficiency ● Future readiness
Lean	Process Resource Product	Diagnostic Strategic Learning and adaptation	● Project efficiency ● Business success ● Future readiness
SAFe	Process Project Resource	Strategic Tactical	● Project efficiency ● Business success ● Future readiness
DSDM	Quality Product	Operational Learning and adaptation	● User impact ● Future readiness

Scrum operationalizes agility through iterative planning, inspection, and adaptation around the Product Goal and Sprint Goal. The Scrum Guide does not prescribe a specific metric set, leaving measurement and forecasting techniques to the context of individual Scrum implementations [45]. In practice, teams frequently use indicators such as Velocity and Burndown Charts to support sprint planning, progress monitoring, and tactical decision-making [2], [39]. These indicators should therefore be understood as complementary practices used within Scrum environments rather than mandatory elements of the framework. Although they may support Project Efficiency, excessive focus on stable Velocity or task completion can conceal declining quality or limited business relevance. Additional value-oriented indicators are therefore needed to assess broader business outcomes.

Kanban conceptualizes agility as the optimization of workflow systems. Its core flow metrics include Work in Progress (WIP), Throughput, Work Item Age, and Cycle Time [44]. These metrics reflect a systemic perspective on workflow stability, predictability, and flow efficiency. Unlike Scrum, Kanban places less emphasis on capacity planning and more on continuous load balancing and bottleneck elimination. This approach strongly supports project efficiency and long-term process sustainability, but introduces the risk of local optimization, where improvements in workflow do not necessarily translate into increased business or user value. Kanban thus illustrates how operational and diagnostic purposes of measurement may dominate over strategic success dimensions.

Lean emphasizes value-stream optimization, waste reduction, and continuous improvement. In Lean software development, measurement practices primarily support process monitoring and value delivery [2]. The use of process and resource metrics, particularly lead time and throughput, reflects a system-level perspective focused on the entire delivery chain rather than individual iterations. These practices primarily support project efficiency, business success, and future readiness. However, if measurement focuses too narrowly on efficiency and waste reduction, innovation and user-perceived value may be underrepresented, reinforcing the need for a balanced and context-aware approach.

SAFe extends agile measurement logic to the organizational and value-stream levels. Current SAFe guidance identifies six Flow Metrics: Flow Distribution, Flow Velocity, Flow Time, Flow Load, Flow Efficiency, and Flow Predictability [13]. These metrics support the identification of delays, excessive work in progress, imbalanced capacity allocation, and differences between planned and delivered business value. The Continuous Delivery Pipeline is an enabling delivery capability rather than a metric [46]. Its performance can therefore be assessed through specific flow, quality, and deployment indicators. SAFe measurement supports coordination across multiple teams and strategic alignment, although its breadth may increase the risk of overmeasurement and formalization if indicators are used primarily for control rather than learning.

Within the conceptual mapping proposed in this study, DSDM is associated primarily with Quality and Product Metrics and with the User Impact and Future Readiness dimensions. This association is consistent with DSDM's emphasis on quality and user involvement. However, the specific indicators considered in this study should be interpreted as illustrative metrics selected within the proposed framework rather than as metrics formally prescribed by DSDM.

This comparative synthesis suggests that differences among agile frameworks extend beyond process structures to the logic of measurement and performance interpretation. While certain metrics are shared across approaches, such as lead time and velocity, their functional purpose and interpretative scope vary depending on organizational context and strategic objectives. This further supports the view that effective measurement in agile IT projects requires a balanced, contextualized, and integrated measurement system.

4.7. Context-Purpose-Value matrix

In order to overcome fragmented and intuitive selection of agile metrics, this study proposes the Context-Purpose-Value (CPV) matrix as a structured tool for their systematic and context-grounded application. The matrix supports the selection of relevant metrics aligned with agile development objectives, with a focus on delivering business value and enabling

continuous process improvement [51]. In agile project environments, the CPV matrix helps balance quantitative indicators, such as team velocity or cycle time, with qualitative aspects that are critical for users and business objectives [2], [48]. In this way, metrics are not used solely for tracking progress, but also for motivating teams and supporting decisions that lead to the delivery of valuable software.

The CPV matrix operationalizes the analytical framework by linking project and organizational context, measurement purpose, and value dimensions. This structure aligns metric selection with project objectives and decision-making levels while reducing the risk of misinterpretation and one-sided optimization. Metric selection cannot be universal or context-independent but must reflect the intersection of context, measurement purpose, and the value dimensions to be prioritized. Context includes the stage of product development, regulatory requirements, organizational maturity, and strategic priorities. Because the meaning of a metric may vary under different conditions, the selection process must begin with a clear definition of context.

As illustrated in Figure 5, the matrix follows three steps. First, the project and organizational context is defined. Second, the measurement purpose is specified, such as operational monitoring, diagnosis, tactical planning, strategic alignment, or organizational learning. Third, the relevant value dimensions are identified,

including project efficiency, user impact, business success, and future readiness. These steps guide the selection of appropriate metric groups and KPIs, providing a structured and context-aware alternative to ad hoc metric selection.

Within the operational sequence presented in Table 11, the CPV matrix primarily supports the first three steps: defining the measurement objective, selecting complementary metric groups, and aligning them with the relevant decision level. Once the metric set has been selected, the remaining steps of the analytical framework guide its contextual verification, temporal and relational interpretation, translation into action, and subsequent reassessment.

The practical value of the CPV matrix lies in translating broad agile principles into operational criteria for metric selection. It helps organizations align measurement practices with project objectives, organizational maturity, and strategic priorities instead of uncritically adopting metrics from individual agile approaches. An additional contribution is the integration of three perspectives: measurement domains, measurement purposes, and project success dimensions. This unified structure provides a basis for future empirical validation and quantitative assessment of metric configurations across different organizational contexts.

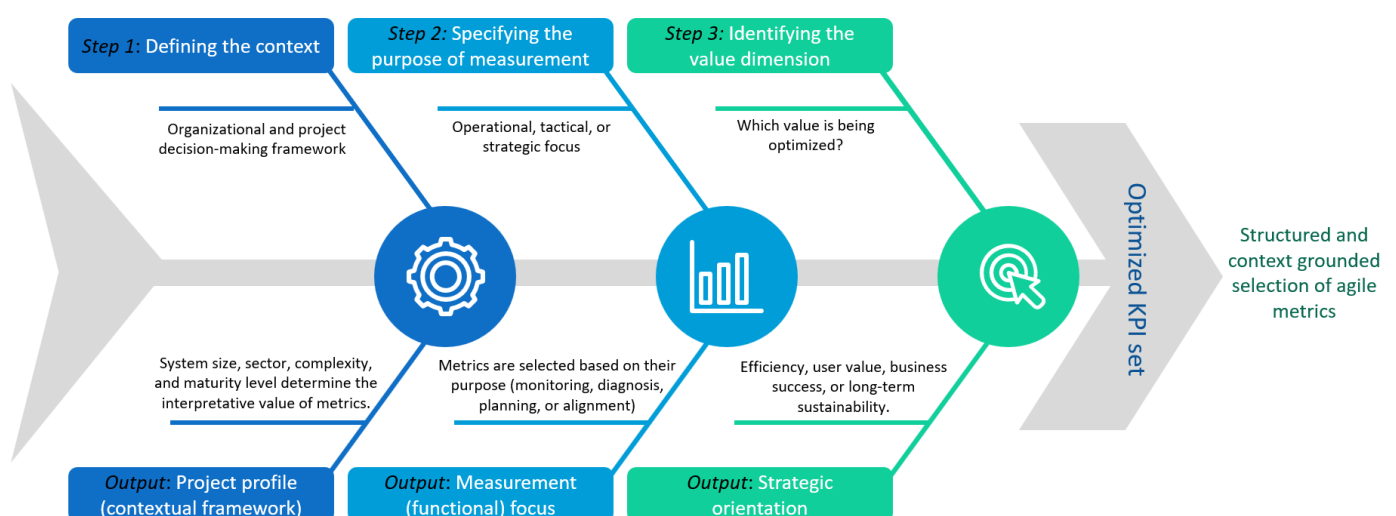


Figure 5. Process of contextual selection of agile metrics.

5. Scenario-based application of the CPV matrix

This section provides an illustrative, scenario-based application of the CPV matrix. The findings should not be interpreted as an empirical case study, a validation study, or an assessment of the actual development organization. Its purpose is solely to demonstrate how the conceptually derived framework can be operationalized when a project context, measurement purpose, and dominant value dimensions are specified. No empirical project-performance data or internal company data are analyzed, and no conclusions regarding the effectiveness of the CPV framework are drawn from this scenario.

The scenario is based on an ongoing project involving the enhancement of Sympoq's Artificial Intelligence (AI) features within an established customer-support Software-as-a-Service (SaaS) platform. The platform has been available on the SaaS market for the past eight years, and the functionalities presented publicly are real; however, the development context used in this scenario is illustrative. The CPV matrix is first applied to select context-specific metrics, after which the operational framework presented in Table 11 guides their contextual assessment, trend and relational analysis, interpretation, and translation into potential improvement actions. The scenario therefore demonstrates the operationalization of the framework rather than its empirical validation or an assessment of the actual Sympoq development team.

5.1. Scenario context and measurement objectives

The scenario focuses on the continued development and enhancement of Sympoq's AI features, which support functions such as automated ticket classification and routing, response suggestions, and chatbot assistance [65]. For illustrative purposes, the development setting is represented by a cross-functional Scrum team of five members working in two-week sprints. This team configuration is an analytical assumption introduced to demonstrate the application of the CPV matrix and does

not describe the actual internal organization of Sympoq's development activities.

Because the AI features are integrated into an operational SaaS platform with an existing user base, the scenario concerns the enhancement and scaling of established functionality rather than the initial validation of a new product. The team must deliver incremental improvements without disrupting the stability of the live platform. At the same time, AI-based features introduce specific measurement challenges, because a technically successful release may still produce outputs that are inaccurate, inconsistent, insufficiently useful, or poorly accepted by users. Delivered performance must therefore be considered together with technical reliability, output quality, and user trust.

To identify the applicable CPV context, the scenario was compared with the five contextual patterns defined in Table 14. As shown in Table 15, the growth and scaling phase provides the closest match because the scenario combines continued product enhancement, quality stabilization, and increased use of existing AI functionality. The remaining contexts are less applicable, because the scenario does not concern early product validation and provides no evidence of a stable organizational portfolio, dominant regulatory constraints, or a multi-team scaled agile structure. The growth and scaling classification refers specifically to the development stage of the AI features and should not be interpreted as evidence of organization-wide maturity or scaling.

Within this context, the dominant objectives are to improve the quality and reliability of the AI features, support continued user adoption, and maintain an efficient and predictable development flow. Measurement therefore primarily serves tactical decision-making and diagnostic analysis. At the tactical level, it should support backlog prioritization and help balance delivery pace with software quality, AI output quality, and user value. At the diagnostic level, it should help identify workflow bottlenecks, quality degradation, unreliable AI outputs, inadequate test coverage, and limited user acceptance.

Table 14. Integrated CPV matrix for contextual selection of agile metrics.

Context	Dominant objectives	Primary measurement purpose	Dominant value dimension	Recommended metric groups	Example KPIs
Early stage of product development	Idea validation, iteration speed	- Operational - Diagnostic	Project efficiency and user impact	- Process - Quality - Product	RLT, WIP, QDC, PLS
Growth and scaling phase	Quality stabilization, user growth	- Tactical - Diagnostic	User impact and future readiness	- Quality - Product - Process	QTC, NPS, RCT
Mature organization (stable portfolio)	Profitability, resource optimization	- Strategic - Tactical	Business success and future readiness	- Project - Resource - Product	ROI, OBR, SAC
Regulated or high-risk environment	Compliance, reliability, security	- Diagnostic - Strategic	Future readiness and business success	- Quality - Resource	QDD, QSV, SAA
Scaled agile system (multiple teams)	Coordination and synchronization	- Strategic - Tactical	Project efficiency and business success	- Process - Project - Resource	RFE, RFT, RFP

Table 15. Context identification for the Sympoq AI features scenario.

CPV context	Applicability	Scenario-based rationale
Early stage of product development	Not primary	The AI features are being enhanced within an operational SaaS platform with an existing user base rather than being developed as an unvalidated new product.
Growth and scaling phase	Primary	The scenario focuses on improving the quality, reliability, and continued use of existing AI functionalities while maintaining stable delivery within a live product.
Mature organization (stable portfolio)	Not established	The scenario does not provide sufficient information about portfolio stability, profitability, or organization-wide resource optimization. An established product cannot, by itself, be treated as evidence of a mature organizational portfolio.
Regulated or high-risk environment	Not established	No scenario assumption identifies regulatory compliance, safety-critical operation, or elevated project risk as a dominant development constraint.
Scaled agile system (multiple teams)	Not applicable	The illustrative setting involves one five-member Scrum team rather than coordination across multiple teams or an organization-wide scaled agile structure.

The overall measurement objective is to determine whether successive improvements to the AI module are delivered efficiently while maintaining technical stability and producing reliable and meaningful user outcomes. This objective requires a balanced interpretation of delivery, quality, and user-oriented indicators rather than optimization of any individual metric. The defined context, objectives, and measurement purposes provide the basis for selecting the relevant value dimensions, metric groups, and KPIs in the following subsection.

5.2. CPV-based metric selection and interpretation

The metric-selection process begins with the integrated CPV matrix presented in Table 14. Based on

the scenario defined in Section 5.1, the Sympoq AI features are positioned within the growth and scaling phase because it is continuously enhanced as part of an operational product with an existing user base. The dominant objectives are therefore quality stabilization and increased use of the AI features. The primary measurement purposes are tactical decision support and diagnostic analysis, while the dominant value dimensions are User Impact and Future Readiness. Their intersection indicates that Quality, Product, and Process Metrics should form the balanced measurement set.

Table 16 presents the scenario-specific application using the same analytical structure as the integrated CPV matrix.

Table 16. Scenario-based application of the integrated CPV matrix to the Sympoq AI features.

Context	Dominant objectives	Primary measurement purpose	Dominant value dimension	Recommended metric groups	Selected KPIs
Growth and scaling of the AI module within an operational SaaS platform	Quality stabilization and increased use of reliable and useful AI functionalities	Tactical and diagnostic	User Impact and Future Readiness	Quality, Product, and Process	Quality: Delivery Quality (QDQ), Test Coverage (QTC); Product: Level of User Satisfaction (PLS), Net Promoter Score (NPS); Process: Cycle Time (RCT)

The selected KPIs represent a focused scenario-specific subset rather than a universal metric package. Cycle Time (RCT) indicates whether AI-related backlog items move efficiently through the development process.

Delivery Quality (QDQ) captures the frequency and severity of defects in released AI functionalities, including function-specific outputs that fail predefined acceptance criteria. Test Coverage (QTC) indicates whether relevant code, integration paths, and critical AI-features' behavior are adequately covered by tests. Level of User Satisfaction (PLS) reflects users' perceived usefulness and quality of the AI functionalities, while Net Promoter Score (NPS) provides a broader indication of user perception and loyalty. Together, these metrics cover delivery flow, technical and AI-related quality, and perceived user value without introducing indicators that are unrelated to the defined scenario.

Before their use, the selected metrics should also be assessed against the criteria presented in Table 7, particularly relevance to objectives, contextual suitability, data availability, actionability, resistance to manipulation, and balance. For example, QTC should be retained only if test scope and coverage rules are defined consistently, while PLS and NPS require a stable feedback method and a clearly identified target user group. The final metric set may therefore be adjusted if reliable data cannot be obtained or if a metric does not support a concrete team or product decision.

Metric selection alone does not ensure valid conclusions. The selected KPIs must subsequently be applied through the operational analytical framework presented in Table 11. Table 17 preserves its nine-step

structure and demonstrates how the selected metrics can be validated, interpreted, and translated into action within the Sympoq AI features scenario. Because no empirical project data are available, the table defines the analytical procedure and conditional decision logic rather than reporting observed results.

The relational interpretation can be illustrated through a conditional pattern. A reduction in RCT would represent a positive development signal only if QDQ remains stable or improves, QTC does not decline, and PLS and NPS show no deterioration. If RCT decreases while QDQ worsens, the apparent improvement in delivery flow would indicate local optimization rather than overall progress. Conversely, stable technical quality combined with declining PLS or NPS would suggest that technically correct improvements are not producing sufficient user value. In either case, the conclusion should remain diagnostic until contextual factors have been examined and the effect of a controlled intervention has been evaluated through the feedback loop.

This scenario illustrates the general application logic of the proposed framework. Although the context, objectives, value dimensions, metric groups, and KPIs may differ across projects, the analytical sequence remains the same. Table 14 supports the selection of a context-appropriate and balanced metric set, while Table 11 guides its contextual verification, temporal and relational analysis, interpretation, translation into action, and subsequent learning. The framework is therefore intended to support application in other IT project contexts without treating the metric set selected for the Sympoq AI features as universally applicable.

Table 17. Scenario-based application of the operational analytical framework to the Sympoq AI features.

Process	Purpose	Potential risk	Expected outcome
1. Definition of measurement objective	Determine whether AI features' improvements are delivered efficiently while maintaining technical reliability and producing meaningful user outcomes.	An exclusive focus on delivery speed, technical quality, or user perception may produce a one-sided assessment.	A clearly defined measurement focus that balances delivery flow, quality, and user value.
2. Selection of metric groups	Combine Quality Metrics (QDQ and QTC), Product Metrics (PLS and NPS), and a Process Metric (RCT).	Reliance on a single metric group may create local optimization, such as faster delivery without corresponding quality or user value.	A balanced set of complementary metrics aligned with the scenario-specific CPV determination in Table 16.
3. Mapping to decision level	Use RCT, QDQ, and QTC for tactical prioritization and use their relationships with PLS and NPS for diagnostic assessment of released improvements.	Metrics may be applied at a decision level for which they provide insufficient evidence.	A context-relevant set of indicators linked to tactical and diagnostic purposes.
4. Contextual verification	Verify whether changes in backlog composition, task complexity, acceptance criteria, test scope, release frequency, or feedback participation affect the observed values.	Changes in metrics may be incorrectly attributed to team or product performance when they result from contextual artifacts.	An interpretation validated against relevant changes in the development and measurement context.
5. Trend analysis	Examine RCT, QDQ, QTC, PLS, and NPS across several sprints and releases rather than using isolated observations.	A single unusual sprint or a small feedback sample may lead to premature conclusions.	Identification of persistent patterns, deviations, and directional changes over time.
6. Relational assessment	Interpret changes in RCT together with QDQ and QTC, and relate technical results to PLS and NPS.	A shorter RCT may create an illusion of improvement if delivery quality, test coverage, or user perception declines.	Identification of trade-offs and systemic relationships among flow, quality, and user value.
7. Interpretation classification	First describe the observed pattern, then diagnose plausible causes, and only afterward determine whether an intervention is justified.	Descriptive evidence may be treated as proof of causality, leading to unsupported prescriptive conclusions.	A clearly distinguished descriptive, diagnostic, or prescriptive conclusion.
8. Action definition	Select a controlled response, such as reprioritizing the backlog, refining AI-output acceptance criteria, strengthening tests, or conducting targeted user-feedback analysis.	Reactive changes may address an isolated value rather than the underlying systemic cause.	A defined and proportionate adjustment or improvement experiment.
9. Feedback loop	Reassess the same metric relationships in subsequent sprints and releases to determine whether the intervention produced the intended effect.	An action may be implemented without evaluating its consequences or capturing the resulting learning.	Evidence-based adaptation of development practices, backlog priorities, or the metric set itself.

6. Discussion

The synthesis indicates that agile metrics are most valuable when used as a balanced set rather than in isolation. Mapping metrics to the five measurement domains and to their managerial purposes revealed complementary roles: Process Metrics diagnose flow and bottlenecks, Product Metrics connect delivery to user and business outcomes, Quality Metrics safeguard technical sustainability, Project Metrics align delivery with broader project constraints and strategic objectives, and Resource Metrics inform capacity and long-term

sustainability. The CPV matrix operationalizes these relationships by linking project context, measurement purpose, and dominant value dimensions, enabling practitioners to select metric combinations that reflect both immediate management needs and longer-term organizational priorities. The Sympoq AI features scenario further illustrates how the CPV matrix can be embedded within the operational analytical framework to connect metric selection with contextual verification, relational interpretation, improvement actions, and subsequent reassessment.

The scenario-based application demonstrates the complementary functions of the two proposed instruments. The operational analytical framework defines the overall sequence from the measurement objective to the feedback loop, while the CPV matrix supports the initial selection of metric groups and KPIs by combining Context, Purpose, and Value. In the illustrative growth and scaling context, this procedure resulted in a combination of Quality, Product, and Process Metrics rather than the adoption of a universal metric package. The example also shows that a favorable change in an individual indicator, such as shorter Cycle Time, cannot be interpreted as improvement unless it is examined together with technical quality and user-oriented indicators. Although the selected KPIs are specific to the Sympoq AI-module scenario, the procedure is transferable because another organization can retain the same analytical sequence while redefining the context, objectives, value dimensions, metric groups, and individual KPIs.

The analysis also provides explicit answers to the three research questions. Regarding RQ1, agile metrics can be systematically classified into Product, Process, Quality, Project, and Resource Metrics, while their managerial purposes can be distinguished as operational, diagnostic, tactical, strategic, and learning and adaptation. Regarding RQ2, the proposed mapping suggests that these metric groups contribute differently but complementarily to Project Efficiency, User Impact, Business Success, and Future Readiness, supporting the conclusion that no single group can represent overall IT project success. Regarding RQ3, metric selection and interpretation vary according to framework logic and organizational scale: Scrum emphasizes sprint predictability and tactical process control, Kanban prioritizes flow and bottleneck diagnosis, Lean focuses on value-stream efficiency and waste reduction, SAFe supports cross-team coordination and strategic alignment, and DSDM gives greater emphasis to quality and user validation. These mappings provide the analytical basis for the context-dependent selection logic formalized through the CPV matrix.

Compared with GQM, Balanced Scorecard-based approaches, and SAFe measurement practices, the CPV matrix offers a complementary rather than substitutive perspective by linking project context, managerial

purpose, and expected value within a single selection structure. The CPV matrix does not replace the goal-decomposition logic of GQM, the strategic performance perspective of the Balanced Scorecard, or the framework-specific measurement guidance provided by SAFe; rather, it supports the contextual selection and combination of metrics derived from such approaches. The distinctive decision support provided by the CPV matrix therefore concerns the context-dependent prioritization and adaptation of metric groups across managerial purposes and value dimensions, rather than goal decomposition, strategic performance structuring, or framework-specific measurement. However, the framework may produce ambiguous recommendations when several metrics appear equally relevant, when project objectives conflict, or when the organizational context is insufficiently defined. In such cases, expert-judgement and stakeholder discussion are required to prioritize metrics and adapt the proposed combinations to the specific project environment.

The practical implications for managers and teams concern contextual adaptation, alignment with decision levels, and the combined interpretation of complementary metrics. Metric meaning and usefulness change with team maturity, scale, and organizational incentives. Managers should avoid transferring metric sets from other teams or frameworks without adapting them to the local context. Operational decisions benefit from flow and cycle metrics, whereas strategic decisions require product and project KPIs such as ROI and customer satisfaction. Using both short-term flow indicators and longer-term value indicators reduces the risk of optimizing throughput at the expense of impact. Several metrics, such as RSP and RVL, are sensitive to local conventions and may be misinterpreted. The framework therefore recommends interpreting them together with more objective measures, such as RLT and QDD, and qualitative reviews to preserve learning and trust.

Several limitations should be considered when interpreting the proposed framework. The selection of literature and practitioner-oriented framework documentation may have introduced source-selection bias, while the mapping of metrics to measurement domains, managerial purposes, and value dimensions involved a degree of conceptual judgement. Although

ambiguous classifications were re-evaluated using the predefined mapping rules, no independent inter-rater assessment or external expert validation was conducted. It therefore demonstrates the operational feasibility and internal logic of the proposed approach, but does not provide empirical evidence of its effectiveness. Consequently, the CPV matrix and the operational analytical framework should be regarded as conceptually grounded, practically illustrated instruments that require empirical evaluation across diverse IT project contexts.

7. Conclusion

This study examined the role of agile metrics in monitoring the progress, efficiency, and quality of IT projects through their classification, functional analysis, and comparison across major agile frameworks. The synthesis indicates that agile metrics are not universal indicators but context-dependent management tools whose value depends on how they are interpreted and applied. Process and quality metrics form the operational core, while product, project, and resource metrics connect delivery performance with broader business outcomes and sustainability.

In direct response to the research questions, the study proposes a dual classification of agile metrics by measurement domain and managerial purpose, maps their complementary contributions to four dimensions of IT project success, and identifies framework-specific differences in their selection and interpretation across Scrum, Kanban, Lean, SAFe, and DSDM. The findings suggest that effective agile measurement requires a balanced combination of metrics adapted to the project context rather than a universal metric set.

A key contribution is the integration of metric classification, managerial purposes, and value dimensions into a unified analytical framework. The CPV matrix operationalizes this structure by supporting systematic, context-aware selection of metrics and KPIs aligned with project objectives. It is intended to reduce risks such as metric fixation, local optimization, and misinterpretation, while promoting the use of metrics as instruments for learning and informed decision-making rather than control.

The research method is interpretive and depends on the scope and quality of the available literature and practitioner documentation. Because metric definitions and measurement practices vary across organizations and tools, the framework should be regarded as a structured, context-sensitive guide rather than a prescriptive checklist. Empirical evaluation of the CPV matrix and recommended metric bundles across diverse organizational contexts therefore remains an important direction for future research.

The comparative analysis indicates that the relevance of agile metrics depends on framework logic, project context, and organizational maturity. Effective application therefore requires a balanced and strategically aligned measurement approach integrating operational, tactical, and strategic perspectives. Theoretically, this study synthesizes and extends prior classifications by explicitly linking metric domains with managerial purposes and value dimensions. In contrast to measurement approaches that emphasize a limited metric family, such as velocity in some Scrum implementations or flow-oriented indicators in Kanban and scaled agile environments, the CPV framework supports multidimensional evaluation and provides a structured rationale for context-sensitive metric selection. This synthesis is consistent with literature cautioning against one-dimensional measurement and advocating context-aware measurement systems.

Future research should focus on empirical evaluation of the proposed model across diverse IT project contexts, particularly in complex and scaled organizational environments. This should include independent expert assessment and inter-rater agreement analysis of the proposed mappings and four-level maturity hierarchy, followed by application of the CPV matrix to projects differing in development stage, organizational maturity, risk exposure, and scale. Further research should also examine the balance between metric groups and its effect on long-term project value, refine instruments for measuring success dimensions, and investigate the relationship between metric configurations and organizational learning. Such studies would strengthen the empirical robustness and practical applicability of the framework and provide deeper insights into how agile metrics contribute to sustainable performance and value creation.

Competing Interest Statement

Alem Čolaković serves as Editor-in-Chief of *Science, Engineering and Technology*. To ensure an independent and unbiased editorial process, he had no involvement in the editorial handling, peer review, or decision-making process for this manuscript. The manuscript was handled independently by another editor. The authors declare no other competing interests.

Data Availability Statement

No new datasets were generated or analyzed as part of this study. The study is based on the cited scientific literature and publicly available framework documentation. No internal company data or empirical project-performance data were used in the scenario-based application.

References

- [1] L. López *et al.*, “Quality measurement in agile and rapid software development: A systematic mapping,” *Journal of Systems and Software*, vol. 186, Art. no. 111187, Apr. 2022, doi: 10.1016/j.jss.2021.111187.
- [2] 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, Jun. 2015, doi: 10.1016/j.infsof.2015.02.005.
- [3] S. Leal, J. Hauck, M. Bertan, and G. Vieira, “How Agile Organizations Use Metrics: A Systematic Literature Mapping,” in *Proceedings of the XXI Brazilian Symposium on Software Quality*, in SBQS '22. New York, NY, USA: Association for Computing Machinery, Jan. 2023, pp. 1–11. doi: 10.1145/3571473.3571479.
- [4] M. Choraś *et al.*, “Measuring and Improving Agile Processes in a Small-Size Software Development Company,” *IEEE Access*, vol. 8, pp. 78452–78466, 2020, doi: 10.1109/ACCESS.2020.2990117.
- [5] M. Huss, D. R. Herber, and J. M. Borky, “Comparing Measured Agile Software Development Metrics Using an Agile Model-Based Software Engineering Approach versus Scrum Only,” *Software*, vol. 2, no. 3, pp. 310–331, Sep. 2023, doi: 10.3390/software2030015.
- [6] M. Maddox and S. Walker, “Agile Software Quality Metrics,” in *2021 IEEE MetroCon*, Nov. 2021, pp. 1–3. doi: 10.1109/MetroCon54219.2021.9666049.
- [7] K. Chakravarty and J. Singh, “A Study of Quality Metrics in Agile Software Development,” in *Machine Learning and Information Processing*, D. Swain, P. K. Pattnaik, and T. Athawale, Eds., Singapore: Springer, 2021, pp. 255–266. doi: 10.1007/978-981-33-4859-2_26.
- [8] S. Sharma, “Impact of Agile Methodologies on Project Success in Software Engineering,” *International Journal of Engineering and Computer Science*, vol. 14, no. 07, pp. 27492–27498, Jul. 2025, doi: 10.18535/ijecs.v14i07.5179.
- [9] E. Kandengwa and L. T. Khoza, “Measuring Agile software project success beyond the triple constraint,” *South African Journal of Information Management*, vol. 23, no. 1, pp. 1–8, Aug. 2021, doi: 10.4102/sajim.v23i1.1375.
- [10] R. Menezes, M. Marinho, and S. Sampaio, “Metrics for Large-Scale Agile Development: A Survey of the Brazilian Software Industry,” in *Proceedings of the XXIII Brazilian Symposium on Software Quality*, in SBQS '24. New York, NY, USA: Association for Computing Machinery, Dec. 2024, pp. 210–219. doi: 10.1145/3701625.3701660.
- [11] A. F. Akbar, E. Indrajit, A. Makmur, and H. Santoso, “Optimizing Software Development Through Flow Metrics Analysis in the Scaled Agile Framework (SAFe),” *Sinkron*, vol. 9, no. 2, pp. 663–678, Apr. 2025, doi: 10.33395/sinkron.v9i2.14545.
- [12] R. Menezes, M. Marinho, and S. Sampaio, “Metrics in Large-Scale Agile Software Development: A Multivocal Literature Review,” in *Congresso Ibero-Americano em Engenharia de Software (CIBSE)*, SBC, May 2024, pp. 106–120. doi: 10.5753/cibse.2024.28442.
- [13] Scaled Agile, Inc., “Accelerating Flow with SAFe,” Scaled Agile Framework. Accessed: Aug. 17, 2026. [Online]. Available: <https://framework.scaledagile.com/accelerating-flow-with-safe>
- [14] P. Ram, P. Rodriguez, and M. Oivo, “Software Process Measurement and Related Challenges in Agile Software Development: A Multiple Case Study,” in *Product-Focused Software Process Improvement*, M. Kuhrmann, K. Schneider, D. Pfahl, S. Amasaki, M. Ciolkowski, R. Hebig, P. Tell, J. Klünder, and S. Küpper, Eds., Cham: Springer International Publishing, 2018, pp. 272–287. doi: 10.1007/978-3-030-03673-7_20.
- [15] Project Management Institute, “What Is Project Management.” Accessed: Jan. 31, 2026. [Online]. Available: <https://www.pmi.org/about/what-is-project-management>
- [16] A. K. M. Z. Islam and A. Ferworn, “A Comparison between Agile and Traditional Software Development Methodologies,” *GJCST*, pp. 7–42, Dec. 2020, doi: 10.34257/GJCSTCVOL20IS2PG7.
- [17] S. Rathor, W. Xia, and D. Batra, “Achieving software development agility: different roles of team, methodological and process factors,” *Information Technology & People*, vol. 37, no. 2, pp. 835–873, Apr. 2023, doi: 10.1108/ITP-10-2021-0832.
- [18] G. S. Matharu, A. Mishra, H. Singh, and P. Upadhyay, “Empirical Study of Agile Software Development Methodologies: A Comparative Analysis,” *SIGSOFT Softw. Eng. Notes*, vol. 40, no. 1, pp. 1–6, Feb. 2015, doi: 10.1145/2693208.2693233.

- [19] G. Y. Koi-Akrofi, J. K. Akrofi, and H. Akwetey Matey, "Understanding the Characteristics, Benefits and Challenges of Agile it Project Management: A Literature Based Perspective," *IJSEA*, vol. 10, no. 5, pp. 25–44, Sep. 2019, doi: 10.5121/ijsea.2019.10502.
- [20] S. Alsaqqa, S. Sawalha, and H. Abdel-Nabi, "Agile Software Development: Methodologies and Trends," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 11, pp. 246–270, Jul. 2020, doi: 10.3991/ijim.v14i11.13269.
- [21] C. Muriana and G. Vizzini, "Project risk management: A deterministic quantitative technique for assessment and mitigation," *International Journal of Project Management*, vol. 35, no. 3, pp. 320–340, Apr. 2017, doi: 10.1016/j.ijproman.2017.01.010.
- [22] Z. T. Kosztyán, R. Jakab, G. Novák, and C. Hegedűs, "Survive IT! Survival analysis of IT project planning approaches," *Operations Research Perspectives*, vol. 7, p. 100170, Jan. 2020, doi: 10.1016/j.orp.2020.100170.
- [23] M. Vanhoucke, "On the dynamic use of project performance and schedule risk information during project tracking," *Omega*, vol. 39, no. 4, pp. 416–426, Aug. 2011, doi: 10.1016/j.omega.2010.09.006.
- [24] D. Chloros, V. C. Gerogiannis, and G. Kakarontzas, "Use of software and project management metrics in agile software development methodologies: A systematic mapping study," in *Proceedings of the 2022 European Symposium on Software Engineering*, in ESSE '22. New York, NY, USA: Association for Computing Machinery, Feb. 2023, pp. 25–32. doi: 10.1145/3571697.3571701.
- [25] R. L. Ulloa and R. D. J. G. Herrera, "A Cloud-KPIs Dashboard to Evaluate Agile Development Teams' Performance: A Mexican Case Study," *Journal of Cases on Information Technology*, vol. 26, no. 1, pp. 1–30, Oct. 2024, doi: 10.4018/JCIT.356506.
- [26] C. Fagarasan, C. Cristea, M. Cristea, O. Popa, and A. Pisla, "Integrating Sustainability Metrics into Project and Portfolio Performance Assessment in Agile Software Development: A Data-Driven Scoring Model," *Sustainability*, vol. 15, no. 17, Art. no. 13139, Jan. 2023, doi: 10.3390/su151713139.
- [27] S. Madan, "13 Agile Metrics and KPIs You Must Track in 2026." Accessed: Jan. 31, 2026. [Online]. Available: <https://agilemania.com/agile-metrics-and-kpi-you-must-track>
- [28] K. P. Pham and M. Neumann, "How to Measure Performance in Agile Software Development? A Mixed-Method Study," in *2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Aug. 2024, pp. 443–450. doi: 10.1109/SEAA64295.2024.00074.
- [29] L. Siddique and B. A. Hussein, "A qualitative study of success criteria in Norwegian agile software projects from suppliers' perspective," *International Journal of Information Systems and Project Management*, vol. 4, no. 2, pp. 63–79, 2016, doi: 10.12821/ijispm040204.
- [30] E. Budacu and P. Pocatilu, "Real Time Agile Metrics for Measuring Team Performance," *IE*, vol. 22, no. 4/2018, pp. 70–79, Dec. 2018, doi: 10.12948/issn14531305/22.4.2018.06.
- [31] S. Kognole, "Agile Project Success through Software Quality KPIs," *IJSR*, vol. 13, no. 7, pp. 287–289, Jul. 2024, doi: 10.21275/SR24705005648.
- [32] S. Arif, M. Faisal, M. S. K. Lodhi, P. S. Z. Ahmad, and S. Arif, "Driving agile success: How software process metrics transform project outcomes (2025)," *Contemporary Journal of Social Science Review*, vol. 3, no. 1, pp. 1008–1014, Feb. 2025, doi: 10.63878/cjssr.v3i1.401.
- [33] F. Almeida and P. Carneiro, "Perceived Importance of Metrics for Agile Scrum Environments," *Information*, vol. 14, no. 6, Art. no. 327, Jun. 2023, doi: 10.3390/info14060327.
- [34] F. Almeida and P. Carneiro, "Performance metrics in scrum software engineering companies," *IJASM*, vol. 14, no. 2, p. 205, 2021, doi: 10.1504/IJASM.2021.118061.
- [35] M. Nejatian, M. H. Zarei, M. Nejati, and S. M. Zanjirchi, "A hybrid approach to achieve organizational agility: An empirical study of a food company," *BIJ*, vol. 25, no. 1, pp. 201–234, Feb. 2018, doi: 10.1108/BIJ-09-2016-0147.
- [36] F. Cunha *et al.*, "An Insight into the Capabilities of Professionals and Teams in Agile Software Development: An Update of the Systematic Literature Review," *Journal of Communications Software and Systems*, vol. 20, no. 1, pp. 99–112, Feb. 2024, doi: 10.24138/jcomss-2023-0172.
- [37] C. Tam, E. J. da C. Moura, T. Oliveira, and J. Varajão, "The factors influencing the success of on-going agile software development projects," *International Journal of Project Management*, vol. 38, no. 3, pp. 165–176, Apr. 2020, doi: 10.1016/j.ijproman.2020.02.001.
- [38] C. Bilir, "Project Success Criteria, Critical Success Factors (CSF), and Agile Projects," in *Contemporary Challenges for Agile Project Management*, V. Naidoo and R. Verma, Eds. Hershey, PA, USA: IGI Global, 2022, pp. 52–72. doi: 10.4018/978-1-7998-7872-8.ch004.
- [39] R. Kurnia, R. Ferdiana, and S. Wibirama, "Software Metrics Classification for Agile Scrum Process: A Literature Review," in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Nov. 2018, pp. 174–179. doi: 10.1109/ISRITI.2018.8864244.
- [40] D. M. Shawky and S. K. Abd-El-Hafiz, "Characterizing software development method using metrics," *Journal of Software: Evolution and Process*, vol. 28, no. 2, pp. 82–96, 2016, doi: 10.1002/smr.1766.
- [41] H. Huijgens, R. Lamping, D. Stevens, H. Rothengatter, G. Gousios, and D. Romano, "Strong agile metrics: mining log data to determine predictive power of software metrics for continuous delivery teams," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, in ESEC/FSE 2017. New York, NY, USA: Association for Computing Machinery, Aug. 2017, pp. 866–871. doi: 10.1145/3106237.3117779.
- [42] S. Misra and M. Omorodion, "Survey on agile metrics and their inter-relationship with other traditional

- development metrics,” *SIGSOFT Softw. Eng. Notes*, vol. 36, no. 6, pp. 1–3, Nov. 2011, doi: 10.1145/2047414.2047430.
- [43] Department of Defense, “Agile Metrics Guide: Strategy Considerations and Sample Metrics for Agile Development Solutions.” OUSD(A&S), 2020. [Online]. Available: <https://aaf.dau.edu/wp-content/uploads/2022/08/Agile-Metrics-Guide.pdf>
- [44] J. Coleman and D. Vacanti, “The Kanban Guide.” Orderly Disruption Limited, May 2025. Accessed: Aug. 17, 2026. [Online]. Available: <https://kanbanguides.org/the-kanban-guide/2025.5/pdf/kanban-guide.v2025.5.en.pdf>
- [45] K. Schwaber and J. Sutherland, “The Scrum Guide.” Nov. 2020. Accessed: Aug. 17, 2026. [Online]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- [46] Scaled Agile, Inc., “Continuously Delivering Value Competency,” Scaled Agile Framework. Accessed: Aug. 17, 2026. [Online]. Available: <https://framework.scaledagile.com/team-and-technical-agility-discipline/continuously-delivering-value-competency>
- [47] 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)*, Serdang, Malaysia: IEEE, May 2022, pp. 83–87. doi: 10.1109/AiIC54368.2022.9914025.
- [48] H. Alahyari, R. Berntsson Svensson, and T. Gorschek, “A study of value in agile software development organizations,” *Journal of Systems and Software*, vol. 125, pp. 271–288, Mar. 2017, doi: 10.1016/j.jss.2016.12.007.
- [49] M. Franco-Santos and D. Otley, “Reviewing and Theorizing the Unintended Consequences of Performance Management Systems,” *International Journal of Management Reviews*, vol. 20, no. 3, pp. 696–730, 2018, doi: 10.1111/ijmr.12183.
- [50] K. Imgharene, K. Doumi, and S. Baina, “Agility metrics model for Business IT Alignment,” in *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, Jun. 2020, pp. 662–667. doi: 10.1109/CoDIT49905.2020.9263839.
- [51] D. Hartmann and R. Dymond, “Appropriate agile measurement: using metrics and diagnostics to deliver business value,” in *AGILE 2006 (AGILE’06)*, Minneapolis, MN, USA, Jul. 2006. doi: 10.1109/AGILE.2006.17.
- [52] K. Ghane, “Quantitative planning and risk management of Agile Software Development,” in *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*, Jun. 2017, pp. 109–112. doi: 10.1109/TEMSCON.2017.7998362.
- [53] V. R. Basili, G. Caldiera, and H. D. Rombach, “The Goal Question Metric Approach,” in *Encyclopedia of Software Engineering*, 1st ed., vol. 1, New York, NY, USA: John Wiley & Sons, 1994, pp. 528–532.
- [54] R. S. Kaplan and D. P. Norton, “The Balanced Scorecard—Measures That Drive Performance,” *Harvard Business Review*, vol. 70, no. 1, pp. 71–79, 1992.
- [55] Scaled Agile, Inc., “Measure and Grow,” Scaled Agile Framework. Accessed: Jul. 28, 2026. [Online]. Available: <https://framework.scaledagile.com/measure-and-grow>
- [56] M. Selvam, K. Kanmani, J. Vasavi, J. Mayilsamy, K. Dhivakar, and Moorthy, “Evaluating the Impact of Agile Methodologies on Speed and Reliability in DevOps for Large-Scale Systems,” in *2025 2nd International Conference on Computing and Data Science (ICCDs)*, Chennai, India, Jul. 2025, pp. 1–6. doi: 10.1109/ICCDs64403.2025.11209470.
- [57] I. Korpivaara, T. Tuunanen, and V. Seppänen, “Performance Measurement in Scaled Agile Organizations,” Jan. 2021. doi: 10.24251/HICSS.2021.830.
- [58] R. L. Thomas and D. Uminsky, “Reliance on metrics is a fundamental challenge for AI,” *Patterns*, vol. 3, no. 5, Art. no. 100476, May 2022, doi: 10.1016/j.patter.2022.100476.
- [59] L. Maier-Hein *et al.*, “Metrics reloaded: recommendations for image analysis validation,” *Nat Methods*, vol. 21, no. 2, pp. 195–212, Feb. 2024, doi: 10.1038/s41592-023-02151-z.
- [60] D. Manheim, “Building less-flawed metrics: Understanding and creating better measurement and incentive systems,” *Patterns*, vol. 4, no. 10, Art. no. 100842, Oct. 2023, doi: 10.1016/j.patter.2023.100842.
- [61] G. Y. Koi-Akrofi, D. Aboagye-Darko, E. Gaisie, and F. Banaseka, “IT project success in perspective: systematic literature review analysis founded on the ADO, TCM and the PSALAR frameworks,” *Manag Rev Q*, vol. 74, no. 4, pp. 2401–2441, Dec. 2024, doi: 10.1007/s11301-023-00362-4.
- [62] R. Atkinson, “Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria,” *International Journal of Project Management*, vol. 17, no. 6, pp. 337–342, Dec. 1999, doi: 10.1016/S0263-7863(98)00069-6.
- [63] W. D. Piotrowicz, U. Ryciuk, and M. Szymczak, “Lean and agile metrics. Literature review and framework for measuring leagile supply chain,” *International Journal of Productivity and Performance Management*, vol. 72, no. 6, pp. 1560–1583, Dec. 2021, doi: 10.1108/IJPPM-10-2020-0560.
- [64] C. M. Martínez, J. G. Carracedo, and J. S. Gallego, “Characterizing Agile Software Development: Insights from a Data-Driven Approach Using Large-Scale Public Repositories,” *Software*, vol. 4, no. 2, Art. no. 13, May 2025, doi: 10.3390/software4020013.
- [65] Sympoq, “AI-Driven Help Desk Software for Growing Teams.” Accessed: Aug. 17, 2026. [Online]. Available: <https://www.sympoq.com/>