POLITECNICO DI TORINO

MSc in Engineering and Management



Master's Degree Thesis

Measuring What Matters in Agile Projects

A Systematic Mapping and Taxonomy of Agile Monitoring Metrics

Supervisor

Filippo Maria Ottaviani

Candidate

Dastanbek Erniyazov

September 2026

Contents

1 Introduction **1**

1.1 Research Problem and Context 1

1.2 Motivation and Significance 2

1.3 Overview of Prior Literature 2

1.4 Research Gap 3

1.5 Research Objectives and Research Questions 3

1.6 Research Contributions 3

1.7 Thesis Structure 4

2 Literature Review **5**

2.1 Flow Metrics 5

2.2 Quality Metrics 6

2.3 Team Metrics 7

2.4 Delivery Metrics 8

2.5 Value Metrics 8

2.6 Cost Metrics 9

2.7 Synthesis and Research Gap 10

3 Research Methodology **11**

3.1 Research Design 11

3.2 Search Strategy 12

3.2.1 Data Sources 12

3.2.2 Search Strings 12

3.3 Study Selection Process 13

3.4 Eligibility Criteria 15

3.5 Data Extraction . . . 16
3.6 Data Analysis and Synthesis . . . 17
3.7 Validity and Reliability . . . 19

4 Results **20**
4.1 Literature Identified . . . 20
4.1.1 Search and Selection Outcome . . . 20
4.1.2 Composition of Included Studies . . . 21
4.2 Metric Usage by Framework and by Study Type . . . 24
4.2.1 Scrum versus Framework-Agnostic Studies . . . 25
4.2.2 Proposal versus Mixed-Method Studies . . . 25
4.3 Coverage Validity Check . . . 26
4.4 Taxonomy of Monitoring Metrics . . . 27
4.4.1 Taxonomy Structure . . . 27
4.4.2 Project Management and Software Engineering Origin . . . 28
4.4.3 Breadth versus Depth . . . 29

5 Discussion **31**
5.1 Interpretation of Main and Secondary Findings . . . 31
5.1.1 RQ1: What Metrics Are Reported, and by Whom . . . 31
5.1.2 RQ2: How the Metrics Are Structured . . . 32
5.2 Theoretical Implications . . . 33
5.3 Practical Implications . . . 33
5.4 Empirical Implications . . . 34
5.5 Methodological Contributions . . . 35

6 Conclusions **36**
6.1 Delimitations . . . 37
6.2 Limitations . . . 37
6.3 Directions for Future Research . . . 38

List of Figures

3.1 PRISMA 2020 flow diagram for the search and screening process. 14

4.1 Included records by Agile framework. 21
4.2 Included records by study type. 22
4.3 Included records by publication year. 23
4.4 The 10 most cited metric families. 24
4.5 Scrum focused studies versus studies not focused on a specific framework. . . . 25
4.6 Proposal studies versus Mixed-method studies. 26
4.7 Comparison with Kupiainen et al. (2015). 27
4.8 Taxonomy categories by Number of Families. 28
4.9 Metric Origin classification, by category and overall. 29
4.10 Classification of the 199 metric families by citation depth and naming breadth. . 30

List of Tables

3.1 The two phases of the study and the research question each phase addresses. . . 11
3.2 Search strings used for each data source. 12
3.3 Filters applied to each database search, in the order applied. 13
3.4 Reasons for exclusion at full-text assessment. 16
3.5 Data extraction form. 17

4.1 Included records by search stream. 20

Abstract

Agile software development is widely adopted, but the literature proposing metrics to monitor Agile teams and projects remains fragmented: individual studies propose or validate metrics within a single framework, a single category of concern, or a single organizational context, with no current, comprehensive structure organizing this space as a whole. This thesis addresses that gap through a two-phase systematic mapping study. Phase 1 identifies the monitoring metrics reported in the Agile literature (RQ1) through a search of Scopus, Web of Science, Google Scholar, and backward snowballing, yielding 48 included records. Phase 2 consolidates the metrics extracted from these records into a taxonomy (RQ2) through an inductive, two-stage matching and manual-review process.

The resulting taxonomy comprises 199 canonical metric families organized under six top-level categories, Flow, Quality, Team, Delivery, Value, and Cost, validated externally against Kaplan and Norton's Balanced Scorecard. A second classification axis, Metric Origin, distinguishes Project Management metrics from Software Engineering metrics and identifies Flow as the category bridging the two. Attention across the taxonomy is highly concentrated: seven metric families account for 21.4% of all recorded citations while representing only 3.5% of the 199 families. A coverage check against sixteen canonical metrics identified a decade earlier by Kupiainen et al. (2015) finds 81% representation, with three specific gaps. A breadth-and-depth classification further identifies 29 families as a consensus core with settled terminology, 36 as widely used but inconsistently named, and 129 as a long tail of niche, low-citation proposals.

These findings support the central contribution of this thesis: a taxonomy that organizes the metrics reported across all six categories into a single, externally validated structure, built from current literature and checked for completeness against an established benchmark, filling a gap no prior study among those reviewed addresses at this scope.

Chapter 1

Introduction

1.1 Research Problem and Context

Agile software development is now the dominant approach to managing software projects across a wide range of industries, having moved from a set of lightweight alternatives to traditional project management into the default operating model for many software organizations. This shift brought a corresponding need to monitor Agile teams and projects: to understand whether a team is delivering effectively, whether a process change is having its intended effect, and whether an organization's investment in Agile practice is paying off.

In response, a large and fragmented body of literature has proposed metrics for exactly this purpose. Proposals tend to cluster around a single framework (Scrum, Kanban, SAFe, Extreme Programming), a single category of concern (flow, quality, cost, value), or one particular organizational context (a single team, a scaled program, a DevOps-integrated pipeline), rarely more than one of these at a time.

The problem this thesis addresses is that no comprehensive, current structure exists for this literature. A practitioner or researcher asking what should be measured in an Agile context has to synthesize dozens of separately proposed metrics, reported under inconsistent names. These studies rarely reference one another's measurement vocabulary, and no taxonomy yet organizes this space as a whole.

1.2 Motivation and Significance

This problem matters for at least three groups. For researchers, the absence of a current, comprehensive taxonomy makes it difficult to situate a new metric proposal relative to what already exists: as Chapter 2 documents, individual studies routinely propose or validate metrics without reference to the broader inventory their contribution belongs to, in part because no such inventory has been readily available. For practitioners building dashboards, retrospectives, or reporting structures, a fragmented metric landscape means the same underlying construct is frequently reinvented under a new name each time a new study or tool addresses it, making it harder to compare measurement practice across teams or adopt an existing metric with confidence that its definition is stable. For organizations evaluating or scaling an Agile transformation, inconsistent measurement makes it difficult to state, in a way that is comparable across projects or over time, whether the transformation is succeeding.

The most recent attempt at a comprehensive synthesis of this kind, Kupiainen et al. (2015), is now a decade old and predates the DevOps-adjacent delivery metrics and scaled-Agile measurement practice that Chapter 2 shows have since become part of the field's vocabulary. A current synthesis, built from recent literature and organized into an explicit structure rather than a flat list, is therefore both overdue and, as Section 4.1.2 of Chapter 4 shows, addressing a literature that is still actively growing: nearly half of the studies included in this thesis were published from 2021 onward.

1.3 Overview of Prior Literature

Prior work on Agile measurement falls into three broad strands, discussed fully in Chapter 2. The first proposes or validates individual metrics or small metric sets within a single framework, such as Scrum-specific velocity and burndown measurement (Downey & Sutherland, 2013) or Kanban-specific flow metrics (Haidabrus et al., 2023). The second maps a broader set of metrics, typically within a single category or a single organizational scope, such as quality metrics in Scrum (Mishra & Abdalhamid, 2018) or performance measurement in scaled Agile organizations (Korpivaara et al., 2021). The third, exemplified by Kupiainen et al. (2015), synthesizes metrics across multiple frameworks into a small set of canonical constructs traceable to foundational Agile and Lean texts, but does not organize that synthesis into categories or assess its own completeness against more recent literature. No prior study combines the breadth of

the third strand with an explicit categorical structure and a validity check against an established benchmark; that combination is the specific space this thesis occupies.

1.4 Research Gap

The literature reviewed in Chapter 2 establishes three consistent patterns: measurement maturity is uneven across categories, with Flow and Quality metrics comparatively well-defined while Value and Cost metrics are acknowledged as important but measured with far less rigor; framework context shapes which category of metric a study reaches for, with Scrum's ceremonies giving rise to Team-category measurement that framework-agnostic and Kanban-rooted studies do not share to the same degree; and no included study organizes the full breadth of reported metrics, across all six categories identified in this thesis, into a single structure checked for completeness against an established benchmark. This last point is the specific gap this thesis addresses.

1.5 Research Objectives and Research Questions

The objective of this thesis is to identify the monitoring metrics reported in the Agile literature and to organize them into a taxonomy that supports practical adoption. This objective is addressed through two research questions.

RQ1. What monitoring metrics for Agile projects are identified in the academic and professional literature?

RQ2. How can these metrics be meaningfully classified into a taxonomy that supports practical adoption?

RQ1 is addressed through the systematic mapping study described in Chapter 3 and reported in Sections 4.1 through 4.3 of Chapter 4. RQ2 is addressed through the taxonomy-construction process described in Section 3.6 and reported in Section 4.4 of Chapter 4.

1.6 Research Contributions

This thesis makes the following contributions:

Theoretical contribution. A taxonomy of 199 canonical metric families organized under

six top-level categories, Flow, Quality, Team, Delivery, Value, and Cost, validated externally against Kaplan and Norton's Balanced Scorecard, together with a second classification axis, Metric Origin, distinguishing Project Management metrics from Software Engineering metrics and identifying Flow as the category that bridges the two.

Methodological contribution. An explicit, repeatable protocol for consolidating extracted metrics into canonical families, combining automated matching rules with manual review against source context, and a two-tier criterion, keyword match and manual definition match, for comparing a constructed inventory against an external benchmark where terminology has drifted over time.

Empirical contribution. A quantified account of how attention is distributed across the Agile metrics literature, showing that 3.5% of the taxonomy's families account for 21.4% of all citations, alongside an updated coverage check against Kupiainen et al. (2015)'s decade-old benchmark, finding 81% representation with three specific, named gaps.

Practical contribution. A breadth-and-depth classification identifying which metrics represent settled, consensus terminology ready for direct adoption, which are widely used but still inconsistently named and require terminology consolidation before reliable cross-team comparison, and which remain niche, single-study proposals.

1.7 Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 reviews the relevant literature by metric category and develops the research gap introduced above. Chapter 3 describes the systematic mapping and taxonomy-construction methodology, including the search strategy, eligibility criteria, and taxonomy-consolidation protocol. Chapter 4 presents the results in two parts, corresponding to RQ1 and RQ2. Chapter 5 interprets these results, relates them to the literature reviewed in Chapter 2, and discusses their theoretical, practical, empirical, and methodological implications. Chapter 6 concludes the thesis, stating its delimitations and limitations and outlining directions for future research.

Chapter 2

Literature Review

This chapter synthesizes what the literature identified in Chapter 3 reports about Agile monitoring metrics, organized around the six categories that Chapter 4 later formalizes into a taxonomy: Flow, Quality, Team, Delivery, Value, and Cost. Each section traces how a category's metrics are framed across the included studies, where the literature agrees, and where it does not. The chapter closes by drawing these threads together into the research gap this thesis addresses.

2.1 Flow Metrics

Flow metrics, tracking how work moves through a process rather than what the work produces, are the most heavily represented category in the literature reviewed, and also its most contested. Their conceptual root is not Scrum but Kanban and Lean: Sjøberg et al. (2012) report an empirical comparison of Kanban and Scrum teams built almost entirely on flow measures, cycle time and lead time chief among them, and find that teams switching from Scrum to Kanban saw lead time roughly halve, attributing the change to Kanban's explicit work-in-progress limits rather than to any change in team composition. This finding recurs, in weaker form, across studies that were not designed as Kanban-versus-Scrum comparisons at all: Haidabrus et al. (2023) build an entire measurement approach for Kanban teams around the same small set of flow constructs (work-in-progress, cumulative flow, cycle time), treating them as sufficient on their own to diagnose team effectiveness without reference to Scrum-style estimation.

Velocity is the exception to this Kanban-rooted story, and the single most contested metric in the reviewed literature. Downey & Sutherland (2013) present velocity, alongside story points and the burndown chart, as foundational to what they call Scrum "hyperproductivity," arguing

that a stable, rising velocity signals genuine throughput improvement. Other studies are more guarded. Kurnia et al. (2018) classify velocity as a product metric useful for sprint planning but caution that it is meaningful only within a single team's own estimation baseline, not across teams. Almeida & Carneiro (2023), surveying practitioner perceptions directly, find velocity rated among the most *used* metrics but not among the most trusted, with several respondents describing it as easily gamed through estimation inflation. This tension, velocity as the field's most-cited metric yet also its most openly distrusted, is a recurring theme rather than an isolated finding.

Large-scale and multivocal reviews complicate the picture further. Menezes et al. (2024) find that flow metrics translate awkwardly once Agile teams are scaled to program or portfolio level: cycle time and throughput remain meaningful within a single team, but aggregating them across teams working on interdependent components requires assumptions the underlying literature rarely states explicitly. Leal et al. (2022) reach a related conclusion from a broader systematic mapping: flow metrics dominate industrial usage reports, but the studies proposing them rarely validate whether a given flow metric's interpretation holds outside the single-team, single-board context in which it was originally defined.

2.2 Quality Metrics

Quality metrics are the second-largest category and, unlike Flow, show comparatively little definitional disagreement. Mishra & Abdalhamid (2018) conduct a systematic mapping specifically of quality issues in Scrum and find defect density, test coverage, and code review measures reported consistently across the studies they examine, with authors largely agreeing on what each metric means even when they differ on how often it should be measured. Maddox & Walker (2021) similarly treat quality metrics as the most mature category in their inventory of Agile software quality metrics, noting that most originate directly from pre-Agile software engineering practice (static analysis, defect tracking) rather than being purpose-built for Agile teams.

Where disagreement does appear, it concerns applicability rather than definition. Filho et al. (2024) examine quality metrics specifically in low-code development, a context increasingly layered onto Agile delivery, and find that conventional metrics such as code coverage lose meaning when large portions of an application are generated through visual tooling rather

than hand-written code; they propose low-code-specific substitutes rather than disputing the underlying quality construct itself. Bayona-Oré & Hostos (2022) situate Agile quality metrics alongside ITIL and CMMI practice and find that organizations blending these frameworks tend to keep their quality metrics essentially unchanged from a pre-Agile baseline, in contrast to the flow and delivery metrics they adopt, which are reported as more visibly reshaped by the move to Agile. Taken together, the literature treats Quality as the category where Agile adoption changed *how often* something is measured far more than *what* is measured.

2.3 Team Metrics

Team-category metrics, covering team satisfaction, capacity, and team composition, are reported with substantially more variation in what each study means by the term than either Flow or Quality metrics. Sadiq & Rana (2024), focusing specifically on self-organizing team characteristics, treat team satisfaction as a multidimensional construct spanning autonomy, psychological safety, and role clarity, and criticize the wider literature for frequently collapsing all three into a single satisfaction score without stating which dimension is actually being captured. Boon et al. (2023) report a related operational problem: several of the industrial teams in their study track a team satisfaction metric but do so through informal retrospective sentiment rather than a defined instrument, making the resulting numbers difficult to compare between teams or over time even within the same organization.

Capacity, the second most-cited Team metric identified in this review, is treated more consistently, largely because it maps onto a concrete planning artifact (available team hours or story points per sprint) rather than a subjective construct. Kurnia et al. (2018) and Almeida & Carneiro (2023) both report capacity as primarily a Scrum-context metric, tied to sprint planning ceremonies, with comparatively little use reported outside Scrum-structured teams. This is consistent with the framework-emphasis pattern later confirmed empirically in Section 4.2, where Scrum-focused studies show a materially higher share of Team-category citations than framework-agnostic studies.

2.4 Delivery Metrics

Delivery metrics, covering deployment frequency, time to market, and delivery predictability, are the category most clearly reshaped by DevOps practice layered on top of Agile development. Visser & Hulstijn (2023) and Panda (2025), both examining combined Agile/DevOps team performance rather than Agile alone, treat delivery frequency and time to market as the primary indicators of whether an Agile team's process improvements are actually reaching production, arguing that Flow-category metrics such as velocity describe internal team activity but say nothing about whether that activity results in delivered value. Panda (2025) report mean time to recovery alongside deployment frequency as a paired indicator, following DORA-style DevOps measurement practice, a pairing not present in any of the Scrum-only studies reviewed.

Delivery predictability, rather than delivery speed, is the more contested sub-theme. Korpivaara et al. (2021), studying performance measurement in scaled Agile organizations, find that delivery-on-time metrics become substantially harder to interpret at scale, since a single late component can cascade into apparent delivery failure across an entire program even when every individual team met its own commitments; they argue this is an underexamined weakness of applying single-team delivery metrics unmodified to a SAFe-scaled context. Ilieva et al. (2004), in one of the earliest studies in this review, report actual development time as a delivery-adjacent metric specific to Extreme Programming's planning-game practice, illustrating that delivery-timing metrics have been part of the Agile measurement vocabulary since before Scrum's dominance of the field, not a later DevOps-era addition.

2.5 Value Metrics

Value metrics, covering business value and customer satisfaction, are reported with the least methodological consistency of any category in this review, largely because "value" itself is rarely operationalized the same way twice. Sambinelli & Borges (2019) and Sambinelli & Borges (2022), in the most extensive treatment of this category among the included studies, survey 378 Brazilian practitioners and find fourteen distinct strategies reported for increasing customer value, with no single dominant metric; customer satisfaction scores, feature adoption rates, and direct customer feedback loops are all reported as roughly equally common, a fragmentation the authors attribute to the inherently qualitative nature of value as a construct compared to flow or delivery metrics.

Mahnič & Vrana (2007), by contrast, propose a more formal stakeholder-driven approach, deriving a small set of process performance indicators directly from stakeholder priorities established at project initiation, and argue this is preferable to post-hoc satisfaction surveys precisely because it fixes the definition of value before measurement begins rather than after. Almeida & Carneiro (2023) find business value rated by practitioners as important but rarely tracked with the same rigor as flow or quality metrics, describing it as "aspirational" in most of the organizations surveyed: acknowledged as something that should be measured, but frequently reduced in practice to a proxy such as revenue or customer count that only loosely reflects the construct it stands in for. This gap between stated importance and actual measurement rigor is the most consistent finding across the Value-category literature.

2.6 Cost Metrics

Cost is the smallest category identified in this review, both in the number of distinct metric families and in how much attention individual studies give it. Shirokova et al. (2020), comparing SAFe, LeSS, and Nexus as scaling approaches, is among the few studies to treat cost as a primary outcome measure rather than a secondary consideration, reporting budget adherence as a key criterion in their recommendation of SAFe for enterprise-scale adoption. Torrecilla-Salinas et al. (2015) similarly foreground cost, but at the project level rather than the organizational level, proposing a value-based estimation approach for Agile web development projects that ties cost tracking directly to the business value the work is expected to deliver, rather than treating cost as a constraint tracked independently of value.

Most other studies in this review mention cost metrics, where they appear at all, only as one line item within a broader dashboard or framework rather than as a subject of dedicated analysis. Storti & Clear (2020) note this directly: in their systematic mapping of what Agile measures actually focus on, cost- and budget-related metrics appear far less frequently than process metrics, despite cost being one of the three constraints, alongside scope and schedule, that Agile methods explicitly claim to manage. They interpret this imbalance as evidence that Agile measurement practice has drifted toward what is easy to instrument (flow, activity) rather than what governance stakeholders actually need to see (cost, budget, return on investment).

2.7 Synthesis and Research Gap

Three patterns recur across the six categories reviewed above. First, measurement maturity is uneven: Flow and Quality metrics are both heavily used and comparatively well-defined, while Value and Cost metrics are acknowledged as important but are measured, where they are measured at all, with far less methodological rigor. Second, framework context matters more than most individual studies acknowledge: Scrum-specific ceremonies give Team metrics a vocabulary and cadence that framework-agnostic and Kanban-rooted studies do not share, while DevOps-adjacent delivery metrics form a distinct vocabulary again, layered on top of rather than replacing Agile-native flow and quality metrics. Third, no single study among those reviewed here organizes the full breadth of reported metrics, across all six categories, into one structure; each study instead surveys, proposes, or empirically validates metrics within a narrower scope, whether a single framework, a single category, or a single organizational context.

Kupiainen et al. (2015) comes closest to this kind of cross-category synthesis, and their sixteen canonical metrics, traced back to the Scrum Guide, *Extreme Programming Explained*, the Kanban method, and *Lean Software Development*, remain the most widely cited benchmark for what the Agile and Lean literature treats as foundational. However, that review is now a decade old, predates the DevOps-adjacent delivery metrics reported by Visser & Hulstijn (2023) and Panda (2025), and, as a synthesis rather than a taxonomy, does not attempt to organize its sixteen metrics into categories or assess how completely they are covered by more recent literature. Section 4.3 returns to this benchmark directly, using it as an external check on the taxonomy constructed in this thesis.

This is the gap the present study addresses: no existing work organizes the metrics reported across Flow, Quality, Team, Delivery, Value, and Cost into a single taxonomy, built from current literature, and checked for completeness against an established benchmark. RQ1 and RQ2, introduced in Chapter 3, are formulated directly against this gap.

Chapter 3

Research Methodology

3.1 Research Design

This thesis uses a systematic mapping study to answer both research questions. A systematic mapping study identifies, screens, and classifies existing literature on a topic through a structured and repeatable process, which fits RQ1 and RQ2 better than a narrow, statistically pooled review would, since both questions require broad coverage of a fragmented literature rather than a small set of directly comparable studies. The study follows the guidelines of Petersen et al. (2015) and reports its process according to the PRISMA 2020 statement (Page et al., 2021).

The study has two phases. Phase one identifies and screens the primary studies that report monitoring metrics for Agile projects; it answers RQ1. Phase two consolidates the metrics extracted in phase one into a structured taxonomy; it answers RQ2. Table 3.1 summarizes this mapping.

Phase	Activity	Research question
1	Search, screening, and data extraction	RQ1
2	Taxonomy construction	RQ2

Table 3.1: The two phases of the study and the research question each phase addresses.

3.2 Search Strategy

3.2.1 Data Sources

Two academic databases were searched: Scopus and Web of Science (WoS). Scopus and WoS were selected because they are among the most widely used databases in software engineering systematic reviews, and together they index the two most relevant venues for this topic, computer science and engineering conferences and journals. Searching both reduces the risk of coverage bias that comes from relying on a single database. A supplementary grey literature search was run on Google Scholar to reduce the risk of missing practitioner-reported metrics that never reach a peer-reviewed venue. Backward snowballing on the reference lists of the included studies was run as a fourth stream to catch studies missed by the first three.

3.2.2 Search Strings

The search string combines three concepts: an Agile-method term, a measurement term, and an outcome term. Each database required a different field syntax, shown in Table 3.2. In both Scopus and WoS, the string was applied to the title, abstract, and keyword fields, with the Agile-method term additionally restricted to the title field to keep precision high given the size of both databases. The Google Scholar string uses a simplified syntax because Google Scholar does not support field-restricted search or a wildcard operator.

Database	Search string
Scopus	`TITLE ( agile OR scrum OR "extreme programming" ) AND TITLE-ABS-KEY ( ( measure OR monitor OR metric* ) AND ( productivity OR performance OR value ) )`
WoS	`TI=(agile OR scrum OR "extreme programming") AND TS=((measure OR monitor OR metric) AND (performance OR productivity OR value))`
Google Scholar	`("agile" OR "scrum") AND ("measure" OR "metric") AND ("productivity" OR "performance")`

Table 3.2: Search strings used for each data source.

Table 3.3 reports the filters applied on top of each string and the number of records remaining after each filter. In Scopus, the string above returned 661 records before filtering; in WoS, 656. Both databases were then restricted to the same publication period, 2001 to 2026. Scopus was further restricted to the Computer Science and Business, Management and Accounting subject categories, to conference papers and journal articles, and to English-language publications. WoS was further restricted to articles, review articles, proceedings papers, and early access records; to six WoS categories (Computer Science–Software Engineering; Computer Science–Information Systems; Management; Business; Operations Research and Management Science; Information Science and Library Science); and to English-language publications.

Database	Filter	Records remaining
Scopus	Unfiltered string	661
Scopus	Publication period (2001–2026)	648
Scopus	Subject categories	497
Scopus	Document type	451
Scopus	Language (English)	441
WoS	Unfiltered string	656
WoS	Publication period (2001–2026)	637
WoS	Document type	635
WoS	WoS categories	309
WoS	Language (English)	308

Table 3.3: Filters applied to each database search, in the order applied.

3.3 Study Selection Process

The Scopus and WoS searches were combined, deduplicated, and screened in three further stages: title and abstract screening, retrieval, and full-text assessment. Figure 3.1 reports the number of records at each stage. Scopus contributed 441 records and WoS 308, for 749 records combined. Deduplication removed 126 records, leaving 623 unique records for title and abstract screening.

Title and abstract screening against the eligibility criteria in Section 3.4 reduced the 623 records to 163 records sought for full-text retrieval. Of these, 23 could not be retrieved, leaving

140 assessed in full text. Full-text assessment resulted in 43 included studies and 97 exclusions. Table 3.4 reports the exclusion reasons for these 97 records.

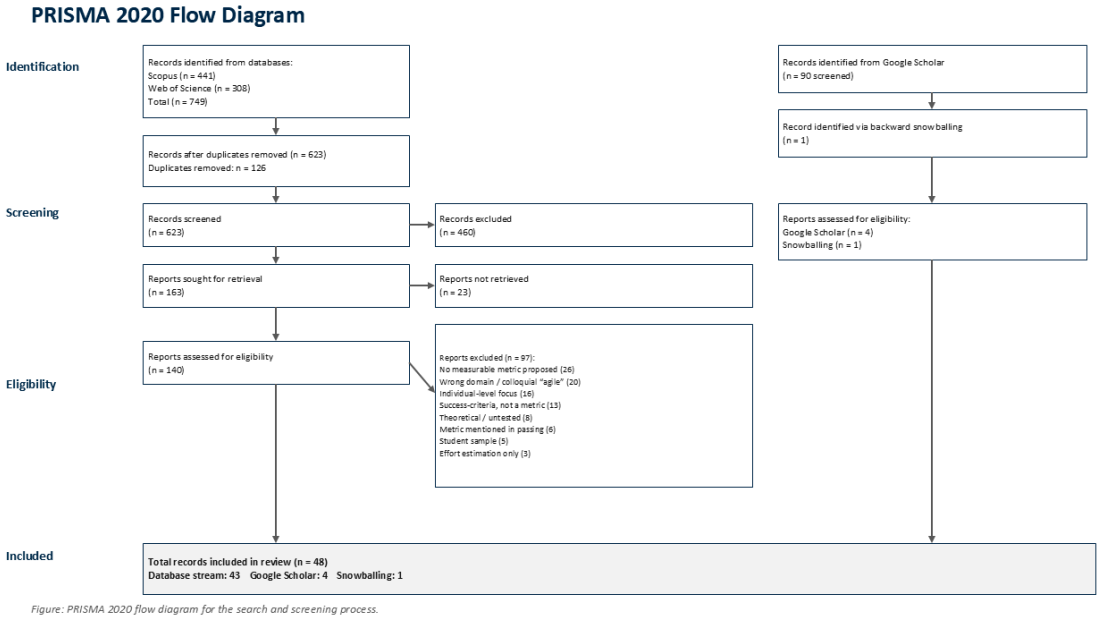


Figure 3.1: PRISMA 2020 flow diagram for the search and screening process.

The Google Scholar search was screened separately, in order of relevance, until reaching saturation, defined as two consecutive pages (20 results) with no new eligible record. Saturation was reached after 90 results had been screened, of which 4 met the eligibility criteria and were retrieved for full-text assessment.

Backward snowballing on the reference lists of the included studies identified one further candidate record. Full-text comparison showed that this record and a study already included through the Google Scholar search reported the same underlying work: Kupiainen's 2014 master's thesis, *Reasons and Effects of Metric Use in Industrial Agile Teams* (Kupiainen, 2014), and the 2015 journal article that later reported the same systematic literature review, *Using Metrics in Agile and Lean Software Development* (Kupiainen et al., 2015). The database-search stream also included two related studies by Sambinelli and Borges: a 2019 survey study (Sambinelli & Borges, 2019) and a 2022 mixed-methods study (Sambinelli & Borges, 2022). The 2022 study reuses the 2019 survey sample (378 respondents, 123 organizations) as its first phase, but adds a second phase of new focus-group data collected from five organizations, not present in the 2019 study. Because the 2022 study contributes empirical findings beyond the 2019 survey, the two are treated as separate studies rather than a duplicate pair, and both are counted

independently throughout this thesis, including in the metric-citation analyses described below.

Combining all streams gives 48 included records. Chapter 4 reports all composition-level figures and tables (Agile framework, study type, publication year) at the full 48 record count. For analyses that count metric citations across studies (concentration/Pareto analysis, category emphasis by framework and study type, top-cited metric families, and the gap analysis against Kupiainen et al. (2015)), the Kupiainen pair, whose two records report an identical set of results, is counted once; one of its two records was excluded from metric extraction for this reason. All other included records, including both Sambinelli studies, are counted independently in these analyses. Metric-citation analyses in Chapter 4 therefore use a base of 47 unique studies. Every figure and table in Chapter 4 states which of the two counting bases it applies.

3.4 Eligibility Criteria

A record was eligible for inclusion if it satisfied one or more of the following three conditions.

1. It proposes, identifies, or discusses a specific metric for monitoring Agile project or team performance.
2. It empirically measures Agile project or team performance, including through practitioner surveys of metric usage.
3. It reviews or classifies existing Agile measurement frameworks.

Records that did not meet any condition above were excluded at full-text assessment. Table 3.4 reports the exclusion reason for each of the 97 excluded records, grouped into eight categories. No measurable metric proposed was the most common reason, followed by records set in a domain unrelated to software or using “agile” in a colloquial, non-methodological sense; together these two reasons account for nearly half of all exclusions.

Exclusion reason	Records
No measurable metric proposed	26
Wrong domain, or colloquial use of “agile”	20
Individual-level or personal-trait focus	16
Success-criteria or fulfillment-rating study, not a metrics study	13
Theoretical or untested proposal	8
Metric mentioned only in passing	6
Student sample, not practitioners	5
Effort estimation only	3
Total	97

Table 3.4: Reasons for exclusion at full-text assessment.

Two borderline rules kept these categories from overlapping during screening. A metric computed at the level of an individual developer was excluded even when the surrounding study was otherwise clearly Agile in context, since RQ1 is scoped to team and project level measurement. Studies that scored constructs on a uniform Likert-style fulfillment scale, without a defined per-item formula, were treated as critical success factor studies rather than metric studies and excluded on that basis, even where the source paper used the word “metric” loosely.

3.5 Data Extraction

One reviewer extracted data from the full text of each included study. Table 3.5 lists the fields recorded for every study; the same form was used throughout to keep extraction consistent.

Field	Description
Name	Title of the study
Publication year	Year of publication
Author(s)	Author name(s)
Document type	Conference paper, journal article, or thesis
Study type	Empirical, survey, case study, SLR, or conceptual
Agile framework	Scrum, XP, Kanban, SAFe, or other
Industry domain	Reported industry or application domain
Metrics identified	Metric names reported in the study
Metrics category	Provisional category assigned during extraction
Key finding	Main result or contribution of the study
Limitation(s)	Limitations reported by the study's authors
Exclusion reason	Recorded for excluded records only

Table 3.5: Data extraction form.

Most included studies name a metric without giving a formula or an operational definition. In these cases the metric was still recorded, and its meaning was inferred from the surrounding text describing what it was used to track or decide. Ambiguous cases were resolved by the same reviewer through closer reading of the source, rather than through a separate disambiguation protocol; Section 3.7 discusses the reliability risk this single-reviewer process creates. The extraction form was not piloted on a trial batch before full extraction: each of the 48 included studies was read in full and its data pulled directly, and the six top-level categories used in the taxonomy (Section 3.6) emerged from this same reading process rather than from a form designed and tested in advance. This is acknowledged as a limitation in Chapter 6.

3.6 Data Analysis and Synthesis

The metrics extracted in Phase 1 were consolidated into a taxonomy through an inductive process. Metrics were first extracted without reference to a predefined classification scheme. The six top-level categories reported in Chapter 4 emerged from repeated grouping and regrouping of the extracted metrics during extraction itself, rather than being fixed in advance and applied top-down.

Consolidation from extracted metrics to canonical families used two stages. The first stage grouped metrics automatically using three matching rules, applied in sequence. Exact-text matching grouped metrics with identical wording after normalizing case and punctuation. Descriptor-stripped matching grouped metrics with identical wording after removing generic qualifier terms such as "chart," "rate," "index," or "metric," so that, for example, "Burndown" and "Burndown Chart" were recognized as the same construct. Abbreviation matching grouped a short, capitalized term with the initials of a longer candidate phrase in the same category, such as "CPI" with "Cost Performance Index."

The second stage manually reviewed each automatically formed group against the metric's original source context, since the automated pass could only detect lexical similarity and several metrics measuring the same construct shared no vocabulary at all. For example, "No. of Active Customers" was merged with "Customer Satisfaction," because its source description identified it as a satisfaction proxy despite sharing no words with the family name. Conversely, "MTTP," which measures the mean time to provision a virtual machine, was kept separate from the MTTR reliability family despite the similar abbreviation, because its source description confirmed an unrelated meaning.

This process yielded 199 canonical metric families, organized under six top-level categories: Flow (52), Quality (50), Team (31), Delivery (28), Value (22), and Cost (16).

The six top-level categories were validated after construction against Kaplan and Norton's Balanced Scorecard (Kaplan, Norton, et al., 2005). Cost corresponds to the Financial perspective, Value to the Customer perspective, and Team to the Learning and Growth perspective. Flow, Quality, and Delivery jointly decompose the Internal Business Process perspective, reflecting the finer process granularity needed in a process-intensive domain such as Agile software development. This mapping gives the category structure external validation beyond the consolidation process alone.

A second classification, independent of the six categories, was applied to every family: Metric Origin, with three possible values, Project Management, Software Engineering, or Both. Two categories were classified uniformly at the category level. Quality was classified entirely as Software Engineering, because defect density, code coverage, and build stability have no meaning outside a software codebase. Cost, Value, and Team were classified entirely as Project Management, because budget adherence, return on investment, and team satisfaction are general management constructs that apply to any project type.

Flow and Delivery required a per-family rule rather than a single category-level label. Flow families defaulted to Both, since Kanban and Lean flow concepts are general process-management techniques applied here to software work items; 48 of the 52 Flow families were classified this way, with the remaining 4 overridden to Software Engineering because they name a software-specific activity, such as commit frequency or code review latency. Delivery families defaulted to Project Management, since schedule, scope, and risk are general project-management concepts; 25 of the 28 Delivery families were classified this way, with the remaining 3 overridden to Both because they are tied specifically to a software delivery pipeline, such as deployment frequency. Chapter 4 reports the resulting distribution of families across the three Metric Origin values.

3.7 Validity and Reliability

Screening was conducted by a single reviewer. Decisions that were not immediately clear against the eligibility criteria were flagged during the first pass rather than resolved on the spot, and were revisited and resolved in a dedicated second pass once the full screening was complete. No second reviewer was available to conduct independent screening or calculate inter-rater agreement (e.g., Cohen's kappa); this is acknowledged as a limitation in Chapter 6, and independent dual-reviewer screening is recommended for any replication.

Taxonomy construction was conducted by the same single reviewer, without independent cross-checking. Several specific consolidation decisions were reviewed a second time before the taxonomy was finalized and were consciously kept rather than revised. No practitioner or second researcher validated the resulting category structure; Chapter 6 discusses this as the study's main limitation and recommends independent or practitioner validation as a direction for future work.

Chapter 4

Results

4.1 Literature Identified

This section reports the outcome of Phase 1 (Section 3.1) and characterizes the literature it identified, answering RQ1: what monitoring metrics for Agile projects are reported in the literature, and by what kind of study.

4.1.1 Search and Selection Outcome

The search and screening process described in Section 3.3 yielded 48 included records across four streams: 43 from the combined Scopus and WoS search, 4 from Google Scholar, and 1 from backward snowballing. One pair of these 48 records, both by Kupiainen, reports the same underlying study (a 2014 master's thesis and its 2015 journal version); both records were retained in the extraction table. Table 4.1 restates the funnel outcome for reference.

Stream	Records included
Scopus and WoS (combined, after full-text assessment)	43
Google Scholar	4
Backward snowballing	1
Total included records	48

Table 4.1: Included records by search stream.

All figures and tables in this chapter that describe the composition of the literature (Agile framework, study type, publication year) report the full 48 record count. Figures and tables

that count metric citations across studies (Sections 4.2 onward) instead use a base of 47 unique studies, since the two records in the Kupiainen pair report an identical set of results and one was excluded from metric extraction for this reason; all other included records, including both of Sambinelli and Borges' studies, are counted independently in these analyses, as stated in Section 3.3.

4.1.2 Composition of Included Studies

By Agile Framework

Figure 4.1 reports the Agile framework studied by each of the 48 included records, following manual verification of this field against each source paper. Scrum is the framework studied by the largest share of the literature (22 of 48 records, 46%), followed by studies that do not focus on a specific framework (20 records, 42%). The remaining framework categories, SAFe, DevOps, Kanban, and XP, together account for the remaining 6 records (12.5%).

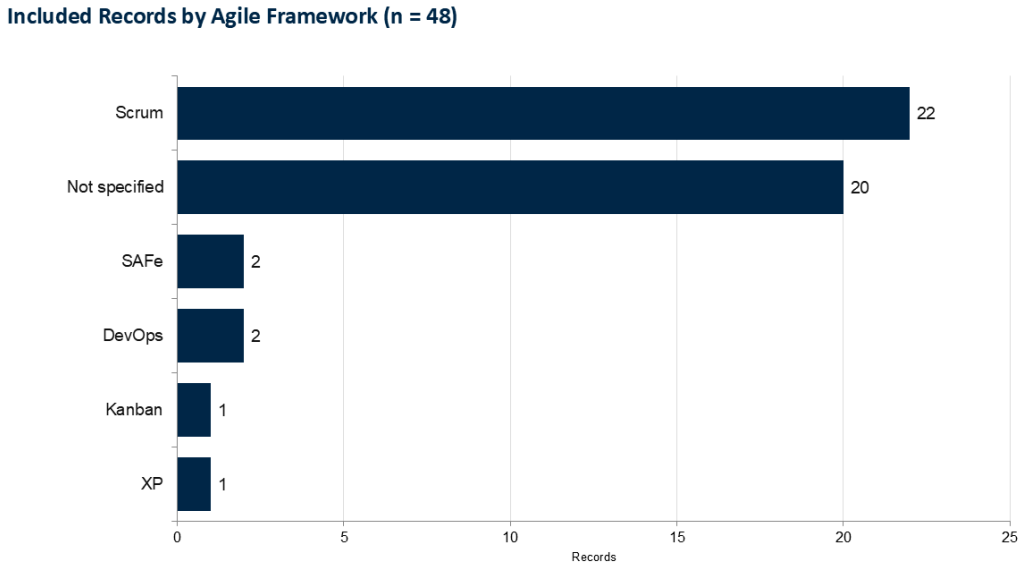


Figure 4.1: Included records by Agile framework.

Because the included literature is concentrated in Scrum and framework-agnostic studies, the metric inventory reported later in this chapter should be read as most directly representative of these two contexts; Section 4.2 returns to this point when comparing what each framework group measures.

By Study Type

Fig 4.2 reports the study type of each included record. Proposal studies, which introduce a new metric, model, or measurement approach without a full empirical validation, are the largest group (14 records, 29%), followed by studies using a mix of methods (12 records, 25%).

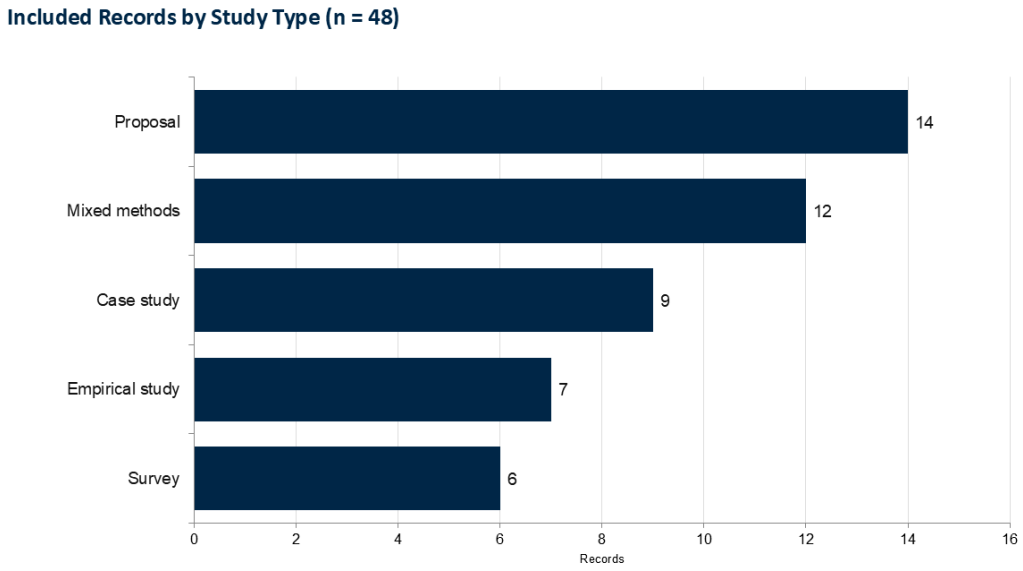


Figure 4.2: Included records by study type.

Publication Trend

Figure 4.3 reports the publication year of each included record. Publication activity is concentrated in the second half of the period covered by the search: 42 of 48 records (87.5%) were published from 2015 onward, and 23 of 48 (48%) from 2021 onward. This trend supports treating Agile metrics monitoring as an active and still-growing research area, and situates the present study's contribution relative to earlier surveys of the same topic, including Kupiainen et al. (2015).

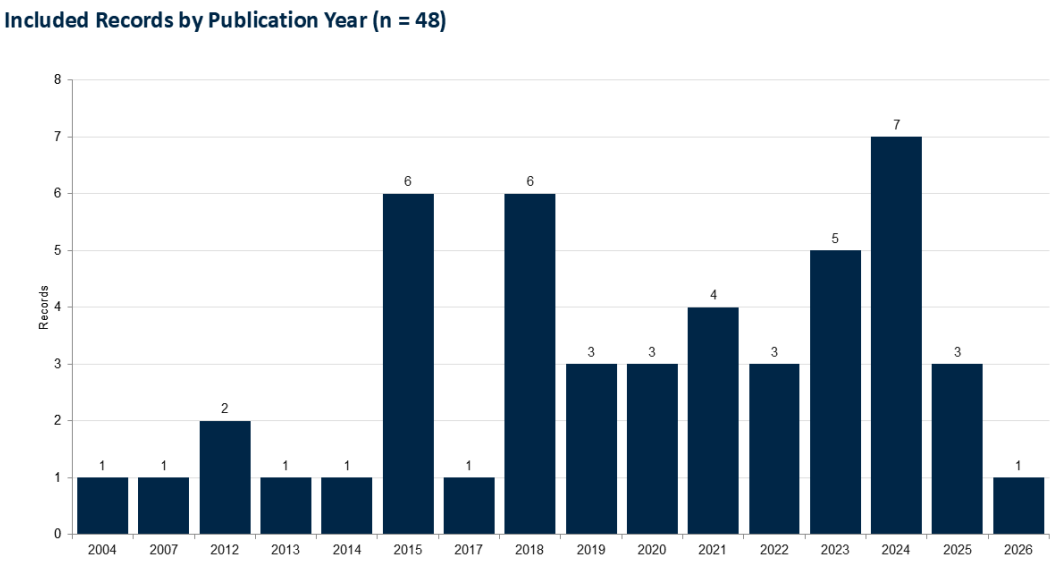


Figure 4.3: Included records by publication year.

The taxonomy constructed in Phase 2 (Chapter 3, Section 3.6) organizes 199 canonical metric families under six top-level categories. This section examines how citations to these families are distributed across the 48 study basis defined in Section 4.1, rather than how many families exist; Section 4.4 in the taxonomy results reports the family and category counts themselves.

Across all 199 families, 627 metric-family citations were recorded. This attention is highly concentrated rather than evenly spread. Fig 4.4 lists the 10 most-cited families, together accounting for 172 of the 627 citations (27.4%). Velocity is the single most-cited family, cited by 29 studies, followed by Bugs/Errors/Defect (27 studies) and Cycle Time (18 studies).

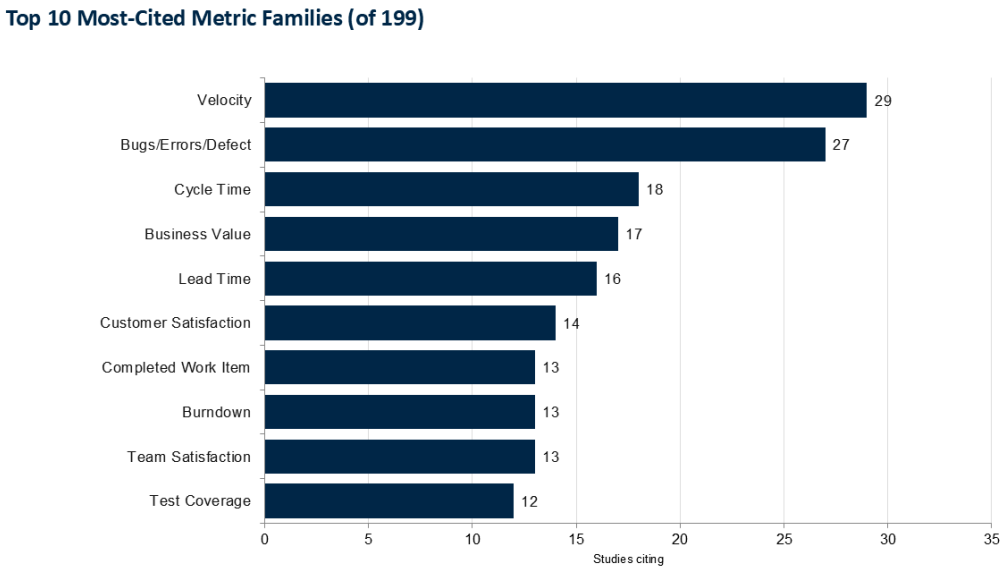


Figure 4.4: The 10 most cited metric families.

Nine of the ten most cited families belong to the Flow, Quality, or Value categories; only one, Team Satisfaction, represents the Team category among the top 10. The remainder of the 199-family taxonomy forms a long tail of families cited by few studies each, a pattern this chapter returns to when relating metric-family breadth to naming consistency.

4.2 Metric Usage by Framework and by Study Type

This section compares which of the six metric categories the literature emphasizes, across the two largest groups within Agile framework and within study type. The remaining framework groups (SAFe, DevOps, Kanban, XP) and study-type groups are each small enough ($n \leq 2$ for the framework groups; see Fig 4.1) that a category-emphasis comparison built on them would not support a reliable claim, and are excluded from this comparison for that reason. The comparisons below are descriptive: no significance test was run on the underlying category-count tables, and the reported percentage differences should be read as patterns in this specific sample rather than as generalizable claims.

4.2.1 Scrum versus Framework-Agnostic Studies

Fig 4.5 compares category emphasis between Scrum-focused studies (n = 22) and studies that do not focus on a specific framework (n = 20), with each category's share expressed as a percentage of that group's total metric-category citations.

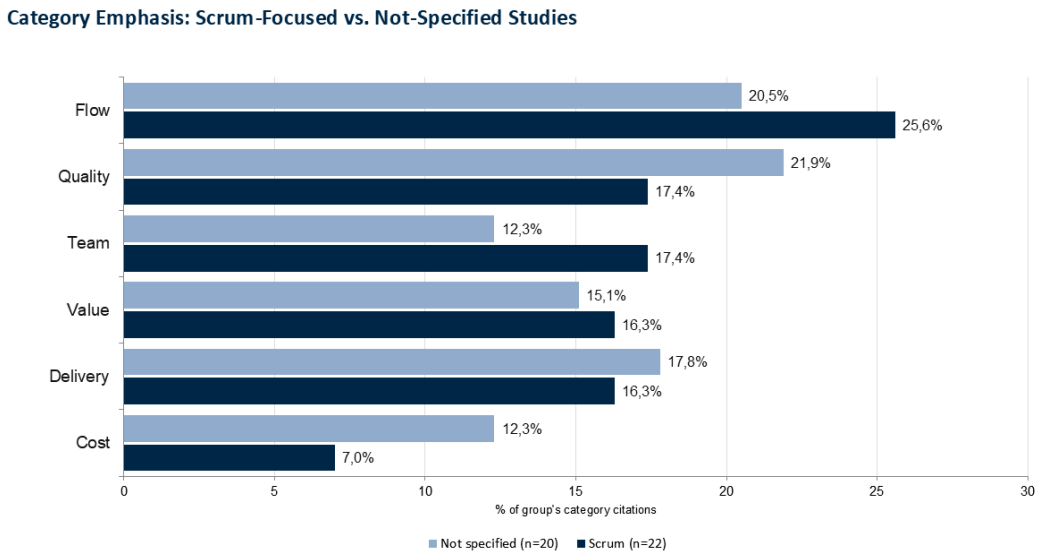


Figure 4.5: Scrum focused studies versus studies not focused on a specific framework.

Both groups emphasize Flow and Quality most heavily, together accounting for over 40% of category citations in each group. Two differences stand out. Scrum studies show a higher share of Team-category citations than framework-agnostic studies (17.4% versus 12.3%), consistent with Scrum's built-in ceremonies and roles (retrospectives, the Scrum Master role) giving these studies a vocabulary for team-level measurement that a generic Agile study may not draw on. Conversely, framework-agnostic studies show a higher share of Cost-category citations (12.3% versus 7.0%) and of Quality-category citations (21.9% versus 17.4%).

4.2.2 Proposal versus Mixed-Method Studies

Fig 4.6 compares category emphasis between Proposal studies (n = 14) and Mixed-method studies (n = 12), the two largest study-type groups.

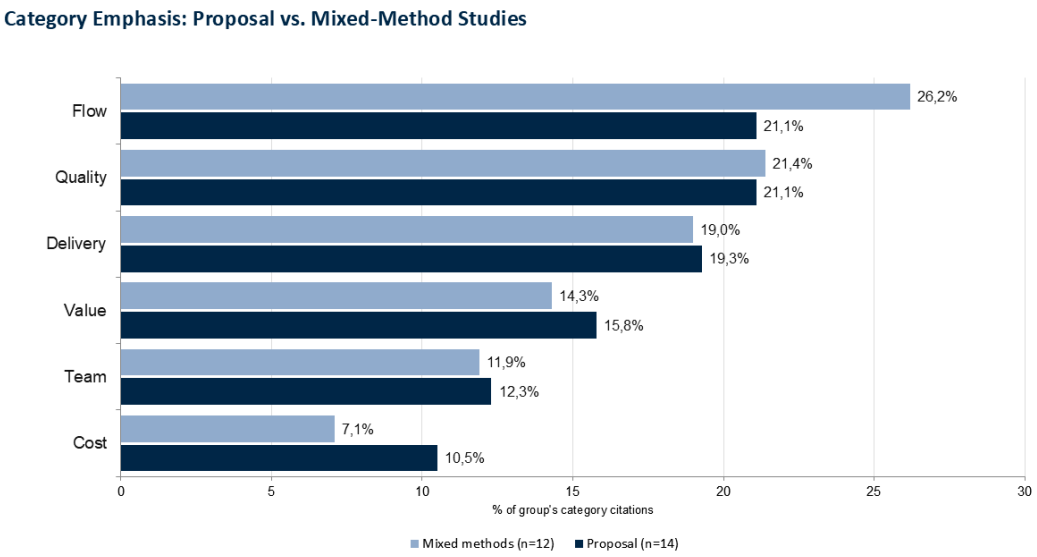


Figure 4.6: Proposal studies versus Mixed-method studies.

These two profiles are considerably more similar to each other than the framework comparison above. The largest single gap is in Flow (26.2% for Mixed-method studies versus 21.1% for Proposal studies), suggesting Mixed-method studies lean slightly more on flow-related metrics; Proposal studies show a correspondingly larger share of Cost-category citations (10.5% versus 7.1%). Unlike the framework comparison, which showed a clear split in what is measured, study type appears to have comparatively little relationship with which categories of metric a study reports.

4.3 Coverage Validity Check

As a check on whether the systematic mapping in Phase 1 captured the metrics that the Agile and Lean literature already treats as canonical, the taxonomy was compared against the 16 metrics listed in Table 8 of Kupiainen et al. (2015), a benchmark traceable to four foundational Agile/Lean texts (the Scrum Guide, *Extreme Programming Explained*, the Kanban method, and *Lean Software Development*). Fig 4.7 reports the result of this comparison.

Coverage Validity Check: Comparison with Kupiainen et al. (2015)

Canonical Metric	Suggested By	Match in Taxonomy
Effort estimate	Scrum, XP, LeanSD	Gap
Velocity	Scrum, XP, LeanSD	Flow-49: Velocity (29)
Written & passed unit tests	XP, LeanSD	Quality-43/46/47
Actual development time	XP	Delivery-20: Time to Market (11)
Load factor	XP	Flow-25: Focus Factor (7)
Work in progress	Kanban	Flow-51: WIP (9)
Lead time	Kanban	Flow-27: Lead Time (16)
Due date performance	Kanban	Delivery-04: Delivery on Time (8)
Throughput	Kanban	Flow-47: Throughput (10)
Issues & blocked work items	Kanban	Flow-05 / Quality-04 (4)
Flow efficiency	Kanban	Flow-21: Flow Efficiency (3)
Initial quality	Kanban, LeanSD	Gap
Failure load	Kanban	Gap
Cycle time	LeanSD	Flow-16: Cycle Time (18)
Value Stream Maps	LeanSD	Flow-39/40: Processing + Queue Time (7)
Passed acceptance tests / iteration	LeanSD	Quality-46: Test Pass Rate (7)

Figure 4.7: Comparison with Kupiainen et al. (2015).

Thirteen of the sixteen canonical metrics are represented in the taxonomy, 11 by direct keyword match and 2 by manual comparison of definitions (Load Factor with Focus Factor; Value Stream Maps with the combination of Processing Time and Queue Time). Three genuine gaps remain: Effort estimate, Initial quality, and Failure load are not represented by any family in the taxonomy. An 81% hit rate against an independent, traceable benchmark supports treating the taxonomy's coverage as reasonably complete, while the three remaining gaps mark an explicit boundary of that coverage rather than an oversight to be hidden.

4.4 Taxonomy of Monitoring Metrics

This section reports the structure of the taxonomy constructed in Phase 2 (Section 3.6), answering RQ2: how the metrics identified under RQ1 can be organized into a classification that supports practical adoption.

4.4.1 Taxonomy Structure

Consolidation of the metrics extracted from the 48 included records produced 199 canonical metric families, organized under six top-level categories. Figure 4.8 reports the size of each category.

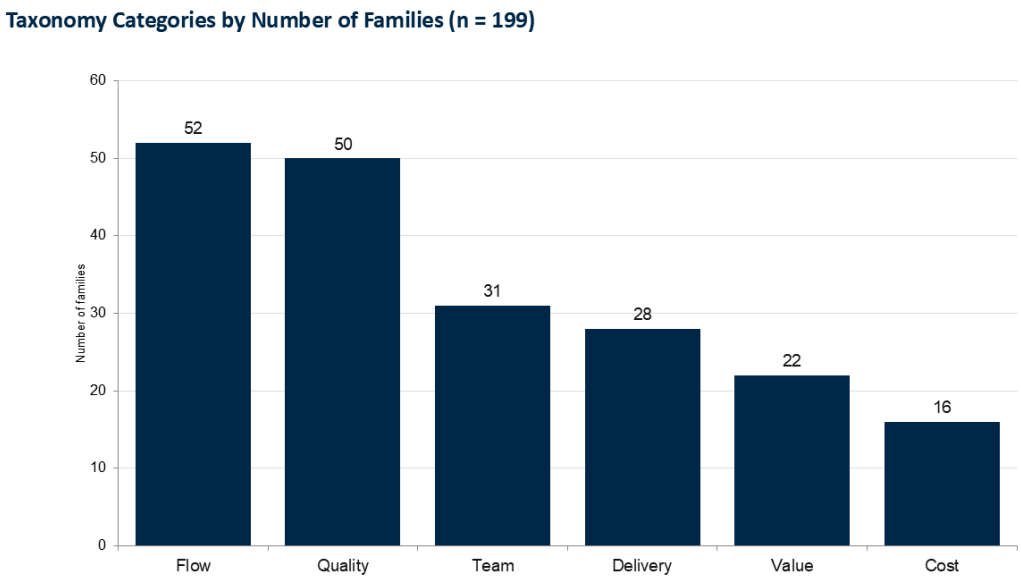


Figure 4.8: Taxonomy categories by Number of Families.

Flow and Quality are the two largest categories, together accounting for just over half of all 199 families. Section 3.6 describes how these categories were derived and validated, including their correspondence to Kaplan and Norton's Balanced Scorecard perspectives (Kaplan, Norton, et al., 2005).

4.4.2 Project Management and Software Engineering Origin

Each of the 199 families was additionally classified by Metric Origin: Project Management, Software Engineering, or Both, following the rule described in Section 3.6. Figure 4.9 reports the resulting distribution, both overall and broken down by category.

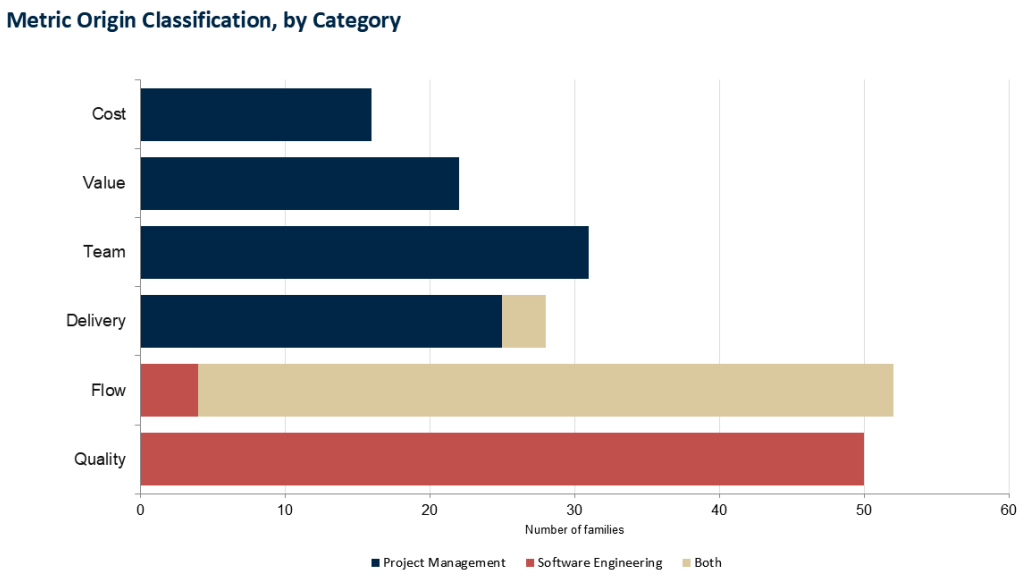


Figure 4.9: Metric Origin classification, by category and overall.

Nearly half of all identified families (94, 47.2%) are Project Management metrics with no software-specific meaning; Cost, Value, and Team are entirely Project Management by category. Quality is entirely Software Engineering: its constituent families (defect density, code coverage, build stability, and similar) have no meaning outside a software codebase. Flow is the only category classified predominantly as Both (48 of 52 families), reflecting that Kanban and Lean flow concepts are general process-management techniques applied here specifically to software work items; the remaining 4 Flow families were overridden to Software Engineering because they name a software-specific activity, such as commit frequency or code review latency. Delivery follows the opposite pattern: it defaults to Project Management (25 of 28 families), with 3 families overridden to Both because they are tied specifically to a software delivery pipeline, such as deployment frequency. This split supports treating the taxonomy's six categories as reflecting a genuine division in the literature between project-level and engineering-level measurement, with Flow acting as the category that bridges the two, rather than as an arbitrary grouping.

4.4.3 Breadth versus Depth

To assess which families in the taxonomy represent settled, consensus terminology and which remain inconsistently named, each of the 199 families was classified along two dimensions:

depth (the number of studies citing the family) and breadth (the number of distinct names or variants recorded for the family across those studies). A threshold of 3 was used for both dimensions to separate a meaningful core from single- or double-mention entries; this threshold is a judgment call rather than a statistically derived cutoff, since the median value for both depth and breadth across the 199 families is 1, leaving no natural break point in the data.

Figure 4.10 reports the resulting four-way classification.

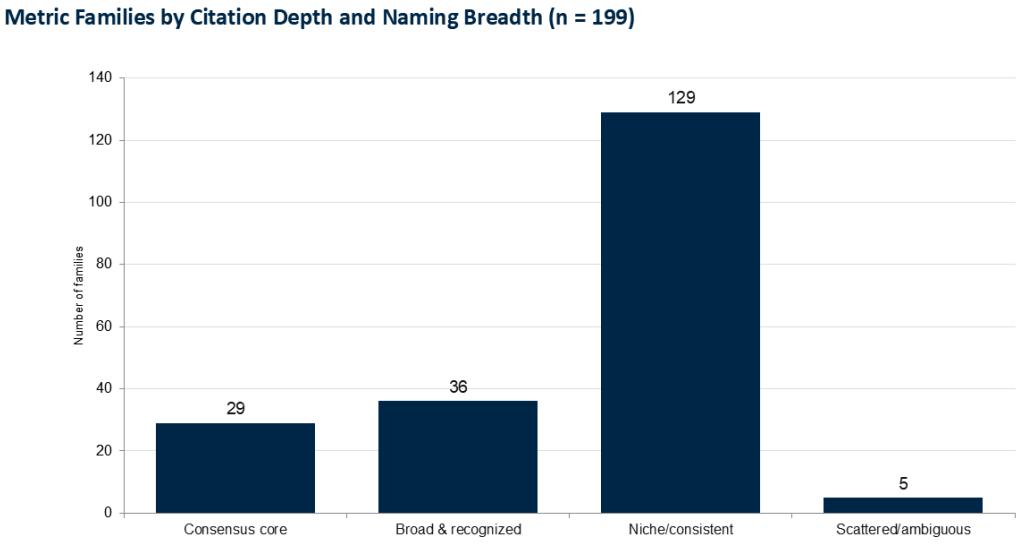


Figure 4.10: Classification of the 199 metric families by citation depth and naming breadth.

The majority of the taxonomy (129 families, 64.8%) falls into the Niche/consistent quadrant: metrics that appear in the literature only occasionally, but under a single, stable name where they do appear. The 29 Consensus core families are the strongest candidates for standard adoption without further terminology work, since the literature already agrees both on their relevance and on what to call them. The 36 Broad & recognized families are widely cited but still inconsistently named, marking them as the entries most in need of terminology consolidation before they can be cited reliably as a single construct. The 5 Scattered/ambiguous families, cited rarely but already recorded under several different names, are the weakest candidates for a clean taxonomy entry as currently defined.

Chapter 5

Discussion

This chapter interprets the findings reported in Chapter 4, relates them to the literature reviewed in Chapter 2, and considers what they contribute theoretically, practically, empirically, and methodologically. As stated throughout Chapters 3 and 4, the findings discussed here are descriptive: no statistical significance testing was performed, and the interpretations offered below should be read as patterns identified in this specific sample of 48 studies rather than as generalizable claims about the Agile literature as a whole.

5.1 Interpretation of Main and Secondary Findings

5.1.1 RQ1: What Metrics Are Reported, and by Whom

The 48 included studies confirm a literature that is active and comparatively recent, with 87.5% of records published from 2015 onward and nearly half from 2021 onward. This recency matters for how the findings below should be read: the taxonomy constructed in this thesis reflects a field still in the process of consolidating its own vocabulary, not a settled body of established practice.

The clearest secondary finding is how sharply attention is concentrated among a small number of metric families. The seven most-cited families, Velocity, Bugs/Errors/Defect, Cycle Time, Business Value, Lead Time, Customer Satisfaction, and Completed Work Item, account for 134 of 627 recorded citations, 21.4%, while representing only 7 of the taxonomy's 199 families, 3.5%. A family drawn at random from the taxonomy would be expected to account for roughly 0.5% of citations if attention were spread evenly; the top seven families average

closer to 3.1% each, a six-fold overrepresentation relative to that baseline. This concentration is consistent with, and gives a precise figure to, what Chapter 2 described qualitatively: Velocity in particular is both the field's most-used and most-distrusted metric, cited far more often than it is trusted, per Almeida & Carneiro (2023)'s practitioner survey. The finding also has a counterpart at the other end of the distribution: Section 4.4.3 found that 134 of 199 families, 67.3%, are cited by fewer than three studies each. Read together, these two results describe a field where a handful of metrics dominate attention while the majority of the taxonomy consists of proposals that have not yet been picked up beyond the study that introduced them.

The framework composition findings extend a pattern Chapter 2 raised but could not quantify. Scrum-focused studies show a materially higher share of Team-category citations than framework-agnostic studies, 17.4% against 12.3%, consistent with Scrum's built-in ceremonies and roles giving these studies a vocabulary for team-level measurement that a generic Agile study does not share. This is the empirical counterpart to the theoretical point raised in Section 4.2 of Chapter 2: framework context shapes not just which metrics are used, but which category of metric a study reaches for in the first place.

Finally, the coverage validity check against Kupiainen et al. (2015) situates this thesis's inventory relative to a decade-old benchmark. Thirteen of sixteen canonical metrics are represented, 81%, with the three gaps, Effort estimate, Initial quality, and Failure load, marking metrics that foundational Agile and Lean texts treat as canonical but that the more recent literature reviewed here does not report using, at least not under a directly comparable name. This is discussed further as a methodological contribution in Section 5.5.

5.1.2 RQ2: How the Metrics Are Structured

The taxonomy's six categories are not evenly sized: Flow (52 families) and Quality (50) together account for just over half of the 199 families, while Cost (16) is the smallest. This asymmetry mirrors Chapter 2's qualitative finding that Quality metrics inherit directly from pre-Agile software engineering practice and are comparatively well-defined, while Cost metrics are acknowledged as important but rarely measured with the same rigor, appearing, per Storti & Clear (2020), as a secondary line item rather than a primary subject of study.

The Metric Origin classification (Section 4.1.2, second table) adds a dimension that no single included study proposes on its own: 47.2% of families are Project Management metrics with no software-specific meaning, 27.1% are Software Engineering metrics with no meaning

outside a codebase, and 25.6% (concentrated almost entirely in Flow) apply to both. Flow's position as the category that bridges project-management and software-engineering measurement is itself a finding, not an assumption built into the taxonomy: it emerges from the fact that 48 of Flow's 52 families default to the Both classification, against a Delivery category that defaults overwhelmingly to Project Management instead. This split gives empirical support to a claim Chapter 2 could only make qualitatively, that Flow metrics sit at the intersection of general process management and software-specific practice in a way Quality, Team, Value, and Cost metrics individually do not.

5.2 Theoretical Implications

Kupiainen et al. (2015) identified sixteen individual metrics as canonical but did not attempt to organize the broader literature into categories. No other study included in this review does either, at the scope attempted here: each proposes, surveys, or validates metrics within a single framework, a single category, or a single organizational context, as documented across Sections 4.1.2 through 4.3 of Chapter 4. The taxonomy constructed in this thesis is therefore a theoretical contribution in its own right: a structure that did not previously exist, built inductively from 48 studies rather than imposed on them, and validated externally against Kaplan and Norton's Balanced Scorecard (Section 3.6).

The Metric Origin axis is a second theoretical contribution. By classifying every family as Project Management, Software Engineering, or Both, this thesis makes explicit a distinction the reviewed literature treats implicitly at best. Studies such as Visser & Hulstijn (2023) and Panda (2025) discuss Agile/DevOps metrics as a blend of process and engineering concerns without naming the blend as a structural property of the metrics themselves; this thesis's finding that Flow, specifically, is the category bridging the two gives that intuition a concrete, falsifiable basis.

5.3 Practical Implications

For practitioners choosing what to measure, the Breadth-versus-Depth classification (Section 4.4.3) offers a direct decision aid. The 29 Consensus Core families, cited by three or more studies and named consistently, are the strongest candidates for adoption without further terminology

work: the literature already agrees both that they matter and what to call them. The 36 Broad and Recognized families are widely used but still inconsistently named; an organization adopting one of these should expect to first agree on a single name and definition internally before comparing the metric across teams, since the source literature itself has not yet converged on one.

The framework-emphasis findings in Section 4.2 carry a specific, actionable implication: a Scrum team should not assume its natural emphasis on Team-category metrics reflects universal best practice, since framework-agnostic studies show a correspondingly higher emphasis on Cost and Quality metrics instead. An organization running Scrum but concerned specifically about budget visibility may need to deliberately import Cost-category measurement practice from outside the Scrum-specific literature, rather than assuming its framework's own literature already covers this. Similarly, the Value category's documented fragmentation, fourteen distinct customer-value strategies reported by Sambinelli & Borges (2019) and Sambinelli & Borges (2022) with no dominant metric among them, means an organization asking "how do we measure whether we are delivering value" should not expect the literature to hand it a single validated answer; some local definition work, of the kind Mahnič & Vrana (2007) recommend through stakeholder-driven indicator design, is likely still necessary.

5.4 Empirical Implications

This thesis contributes new empirical evidence in three respects. First, it quantifies a concentration pattern in Agile metric usage that prior literature described only qualitatively: 3.5% of the taxonomy's metric families account for 21.4% of citations, a six-fold overrepresentation relative to a uniform baseline. Second, the publication-trend data (Section 4.1.2) provides updated evidence that Agile metrics research remains an active and growing area a decade after Kupiainen et al. (2015), rather than a settled topic; nearly half of the included studies were published from 2021 onward. Third, the coverage validity check itself is new empirical evidence: it is the first quantified check, among the studies reviewed here, of how completely a broad, recent literature sample covers a decade-old, foundational benchmark, and it establishes that coverage at 81%, with three specific, named gaps rather than a vague sense of incompleteness.

5.5 Methodological Contributions

Two methodological choices made in this thesis are offered as transferable to future taxonomy-construction work in this and adjacent areas. The first is the three-stage matching protocol used to consolidate extracted metrics into canonical families (Section 3.6): exact-text matching, descriptor-stripped matching, and abbreviation matching, applied in sequence and followed by manual review against source context. This protocol is explicit and repeatable in a way that the consolidation processes described by the included systematic reviews and mappings, such as Menezes et al. (2024) and Leal et al. (2022), generally are not; those studies report a final metric count without documenting the merge and split rules that produced it.

The second is the two-tier match criterion used in the coverage validity check (Section 4.3): a direct keyword match, and a manual definition match applied where two metrics measure the same underlying construct through essentially the same calculation despite different terminology, such as Load Factor and Focus Factor. This distinction, and the explicit criterion behind it, allowed two additional canonical metrics to be credited as covered rather than misclassified as gaps, and offers a reusable template for future comparisons between a new metric inventory and an established benchmark where terminology has drifted over time.

Chapter 6

Conclusions

This thesis set out to answer two research questions: what monitoring metrics for Agile projects are reported in the literature (RQ1), and how can these metrics be organized into a taxonomy that supports practical adoption (RQ2).

RQ1 was answered through a systematic mapping of 48 included records, spanning Scopus, Web of Science, Google Scholar, and backward snowballing, and screened against three eligibility conditions applied with an OR relationship (Section 3.4). The resulting literature is concentrated in Scrum-focused and framework-agnostic studies, active and comparatively recent, with 87.5% of records published from 2015 onward, and shows highly uneven attention across the metrics it reports: seven metric families account for 21.4% of all citations while representing 3.5% of the taxonomy eventually constructed. A coverage check against the sixteen canonical metrics identified a decade earlier by Kupiainen et al. (2015) found 81% representation, with three specific, named gaps rather than a general sense of incompleteness.

RQ2 was answered by consolidating the metrics identified under RQ1 into 199 canonical families under six top-level categories, Flow, Quality, Team, Delivery, Value, and Cost, through an inductive, two-stage matching and manual-review process, and validated externally against Kaplan and Norton's Balanced Scorecard. A second classification axis, Metric Origin, showed that Flow is the category bridging project-management and software-engineering measurement, while Quality, Cost, Value, and Team sit almost entirely on one side of that divide or the other. A breadth-and-depth classification of the same 199 families identified 29 as a consensus core, cited by multiple studies and named consistently, and 129 as a long tail of niche, low-citation entries.

Together, these findings support the central contribution of this thesis: no prior study among

those reviewed organizes the metrics reported across all six categories into a single structure, built from current literature and checked for completeness against an established benchmark. The taxonomy constructed here fills that gap, while making its own limitations, discussed below, explicit rather than implicit.

6.1 Delimitations

The scope of this study was bounded by several deliberate design choices, distinct from the limitations discussed in Section 6.2.

The search was restricted to Scopus, Web of Science, and a supplementary Google Scholar and snowballing pass; other databases and grey-literature sources (industry blogs, conference talk recordings, unpublished practitioner reports) were not systematically searched. Records were further restricted to the 2001–2026 publication period, to English-language publications, and, within Scopus and WoS, to a fixed set of subject and document-type categories (Section 3.1). These restrictions were chosen to keep the search precise and reproducible given the size of both databases, not because relevant work is assumed to be absent outside them.

The eligibility criteria scoped this study to metrics computed at the team or project level; a metric computed at the level of an individual developer was excluded even where the surrounding study was otherwise clearly Agile in context (Section 3.4). This delimitation follows directly from RQ1's own framing, which asks about project-level monitoring, and was not a limitation encountered during screening but a boundary set before it began.

6.2 Limitations

Several limitations affect how the findings of this thesis should be interpreted.

Single-reviewer process. Screening, data extraction, and taxonomy construction were each conducted by a single reviewer, the author (Sections 3.7, 3.6). Decisions that were not immediately clear were flagged and revisited in a dedicated second pass rather than resolved on first encounter, providing a measure of internal consistency-checking, but no second reviewer was available to calculate inter-rater agreement at the screening stage or to independently verify the taxonomy's category structure. Both are acknowledged risks to reliability rather than resolved ones; independent replication of either stage could yield a different set of included studies or a

different consolidation of metric families.

No piloting of the extraction form. The data extraction form (Table 3.5) was not tested on a trial batch of studies before being applied to all 48 included studies; fields and their interpretation were established through the extraction process itself rather than validated in advance (Section 3.6).

Small comparison groups. The category-emphasis comparisons in Section 4.2 were restricted to the two largest groups within Agile framework and within study type, because the remaining groups, SAFe, DevOps, Kanban, and XP, each contain two or fewer records. Findings for these smaller groups are not reported as comparisons for this reason, and the taxonomy's coverage of framework-specific metrics used by SAFe, DevOps, Kanban, or XP teams specifically should be read as based on a correspondingly thin evidence base within this sample.

Descriptive rather than statistical findings. No statistical significance testing was performed anywhere in this thesis. The breadth-and-depth threshold used in Section 4.4.3 (citation by three or more studies, naming in fewer than three variants) is a judgment call rather than a statistically derived cutoff, chosen because the median value for both depth and breadth across the 199 families is 1, leaving no natural break point in the underlying data. All percentage differences and comparisons reported in Chapter 4 and interpreted in Chapter 5 should be read as descriptive patterns in this specific sample, not as generalizable or statistically validated claims.

Single work for the coverage validity check. The gap analysis in Section 4.3 compares the constructed taxonomy against one benchmark, Kupiainen et al. (2015). This benchmark was chosen because it is traceable to four foundational Agile and Lean texts and was independently corroborated against candidate alternatives from within this thesis's own included studies (Section 3.6), but a taxonomy's coverage relative to a single benchmark is not the same claim as coverage relative to the full space of metrics practitioners might consider canonical.

6.3 Directions for Future Research

Several directions follow directly from the limitations above.

Independent or practitioner validation of the taxonomy's category structure, and of the metric families within it, would address the single-reviewer limitation directly and is the most immediately actionable direction for future work. A structured validation exercise, for instance asking practitioners to sort a sample of the 199 families into the six categories independently of

this thesis's own classification, would test whether the categories are recognizable to the field they claim to describe, not only internally consistent.

A practitioner survey assessing how relevant and how frequently used the taxonomy's metrics are in practice was considered during the design of this study but was not conducted; it remains a natural next phase, since Chapter 4 establishes what the literature reports but not how closely literature and practice align. Such a survey could also test the three gaps identified in Section 4.3, Effort estimate, Initial quality, and Failure load, directly: it is not established by this thesis whether these metrics have genuinely fallen out of use, or whether they are still used in practice but no longer reported under a name this taxonomy's search and extraction process could recognize.

The Broad and Recognized quadrant identified in Section 4.4.3, 36 families that are widely cited but inconsistently named, is a specific, bounded target for terminology-consolidation work, for instance a Delphi study among Agile measurement researchers aimed at converging on single names for these families before they are proposed as a stable reference vocabulary.

Finally, replicating the coverage validity check against additional benchmarks beyond Kupiainen et al. (2015), and repeating the screening and taxonomy-construction stages with a second, independent reviewer, would directly test the two limitations most central to this thesis's own reliability: its single comparator and its single reviewer.

References

Almeida, F. & Carneiro, P. (2023). “Perceived Importance of Metrics for Agile Scrum Environments”. *Information (Switzerland)*, 14:6. 10.3390/info14060327.

Bayona-Oré, S. & Hostos, M. (2022). “Metrics for Performance Improvement in Organisations Using Scrum, ITIL and CMMI”. *WSEAS Transactions on Electronics*, 13, 89–99. 10.37394/232017.2022.13.12.

Boon, G. et al. (2023). “Beyond Dashboards: Operationalising a Measurement Framework for Agile Teams”. *Commun. Comput. Info. Sci.* Ed. by F. J.M. et al. Vol. 1871 CCIS, 130–146. 10.1007/978-3-031-43703-8_10.

Downey, S. & Sutherland, J. (2013). “Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft”. *Proc. Annu. Hawaii Int. Conf. Syst. Sci.* 4870–4878. 10.1109/HICSS.2013.471.

Filho, R. et al. (2024). “Low-Code Quality Metrics for Agile Software Development”. *ACM Int. Conf. Proc. Ser.* 417–425. 10.1145/3701625.3701626.

Haidabrus, B. et al. (2023). “Improving Agile Teams Effectiveness Through the Metrics”. *IEEE Int. Sci. Conf. Inf. Technol. Manag. Sci. Riga Tech. Univ., ITMS - Proc.* Ed. by G. J. et al. 10.1109/ITMS59786.2023.10317789.

Ilieva, S. et al. (2004). “Analyses of an agile methodology implementation”. *Proceedings. 30th Euromicro Conference, 2004.* IEEE, 326–333.

Kaplan, R. S., Norton, D. P., et al. (2005). *The balanced scorecard: measures that drive performance*. Vol. 70. Harvard business review Boston, MA, USA.

Korpivaara, I. et al. (2021). “Performance Measurement in Scaled Agile Organizations”. *Proc. Annu. Hawaii Int. Conf. Syst. Sci.* Ed. by B. T.X. Vol. 2020-January, 6912–6921.

Kupiainen, E. (2014). “Reasons and effects of metric use in industrial agile teams-a systematic literature review”.

Kupiainen, E. et al. (2015). “Using metrics in Agile and Lean Software Development–A systematic literature review of industrial studies”. *Information and software technology*, 62, 143–163.

Kurnia, R. et al. (2018). “Software Metrics Classification for Agile Scrum Process: A Literature Review”. *Int. Semin. Res. Inf. Technol. Intell. Syst., ISRITI*, 174–179. 10.1109/ISRITI.2018.8864244.

Leal, S. et al. (2022). “How Agile Organizations Use Metrics: A Systematic Literature Mapping”. *ACM Int. Conf. Proc. Ser.* 10.1145/3571473.3571479.

Maddox, M. & Walker, S. (2021). “Agile software quality metrics”. *2021 IEEE MetroCon*. IEEE, 1–3.

Mahnič, V. & Vrana, I. (2007). “Using Stakeholder-Driven Process Performance Measurement for Monitoring the Performance of a Scrum-based Software Development Process”. *Elektrotehniski Vestnik/Electrotechnical Review*, 74:5, 241–247.

Menezes, R. et al. (2024). “Metrics in Large-Scale Agile Software Development: A Multivocal Literature Review”. *Ibero-Am. Conf. Softw. Eng., CIbSE*, 106–120.

Mishra, D. & Abdalhamid, S. (2018). “Software Quality Issues in SCRUM: A Systematic Mapping”. *Journal of Universal Computer Science*, 24:12, 1690–1716.

Page, M. J. et al. (2021). “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews”. *bmj*, 372.

Panda, S. (2025). “Optimizing Performance in Agile and DevOps Teams”. *IC2IE - Int. Conf. Comput. Informatics Eng.: Human-Mach. Synerg. Brings Together Phys. Digit. Worlds*. Ed. by R. M. et al. 10.1109/IC2IE67206.2025.11283346.

Petersen, K. et al. (2015). “Guidelines for conducting systematic mapping studies in software engineering: An update”. *Information and software technology*, 64, 1–18.

Sadiq, J. & Rana, Z. (2024). “The Study of Characteristics of Self Organizing Teams in Agile”. *Proc. Int. Bhurban Conf. Appl. Sci. Technol., IBCAST*, 203–208. 10.1109/IBCAST61650.2024.10877114.

Sambinelli, F. & Borges, M. (2019). “The Strategies to Increase Customer Value in Agile: A Survey of Brazilian Software Industry”. *Journal of Information Systems Engineering and Management*, 4:2. 10.29333/jisem/5889.

— (2022). “What Software Agile Teams Do To Create Customer Value: A Mixed-Methods Analysis In Brazil”. *International Journal of Computer Information Systems and Industrial Management Applications*, 14, 68–91.

Shirokova, S. et al. (2020). “Company Efficiency Improvement Using Agile Methodologies for Managing IT Projects”. *ACM Int. Conf. Proc. Ser.* 10.1145/3446434.3446465.

Sjøberg, D. et al. (2012). “Quantifying the Effect of Using Kanban versus Scrum: A Case Study”. *IEEE Software*, 29:5, 47–53. 10.1109/MS.2012.110.

Storti, A. & Clear, T. (2020). “The Contradiction of Agile Measures: Customer as Focus, but Process as Measured?” *ACIS Proc.- Aus. Conf. Inf. Syst.*

Torrecilla-Salinas, C. J. et al. (2015). “Estimating, planning and managing Agile Web development projects under a value-based perspective”. *Information and Software Technology*, 61, 124–144.

Visser, T. & Hulstijn, J. (2023). “Measuring Agile/DevOps Team Performance”. *CEUR Workshop Proc.* Vol. 3414, 17–23.