International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ENABLING ANALYTICS GOVERNANCE IN AGILE PRODUCT TEAMS: A SCALABLE TAGGING AND QA FRAMEWORK

Eshita Gupta[1]*

University of Tampa

Abstract

In modern digital product development, data analytics has become foundational for agile decision-making, experimentation, and customer-centric design. However, as organizations scale, analytics governance often becomes fragmented, leading to data quality issues, inconsistent metric definitions, and a lack of trust in insights. This paper proposes a scalable tagging and quality assurance (QA) framework tailored for agile product teams. The framework addresses challenges such as inconsistent event instrumentation, lack of metadata management, and siloed tracking strategies. It enables organizations to align business goals with measurable outcomes while maintaining the flexibility and speed inherent in agile methodologies. By combining governance, automation, and collaboration, this framework supports scalable and trustworthy analytics that empower data-informed product development.

Keywords: Analytics governance, Agile product teams, Tagging framework, Data quality assurance, Scalable instrumentation

1. Introduction

Agile methodologies have transformed how software and product teams develop, iterate, and launch digital experiences. Agile teams focus on speed, experimentation, and customer feedback, and have a tendency to work in parallel streams to release products incrementally. Within this high-speed development environment, product analytics, more specifically behavioral data tracking, is an invaluable source of information that provides context and supports assumptions, as well as quantifies the outcome. Nonetheless, this poses a major challenge to the governance of analytics due to the decentralization and iterative concept of agile delivery. Counterintuitive event names, data silos, and a blurred visibility of data between teams may, over the long term, undermine data trust and create technical debt [1][2][3]. Analytics governance is the primary structure that balances the processes of collecting, leveraging, and quality and security of analytics information. Although the classical governance system is focused on centralized control, it has a tendency to be ineffective in scaling in the agile setting, where autonomy and quickness are critical. This necessitates a hybrid approach, an approach that empowers product teams and, at the same time, allows them all to be aligned to common data and quality principles. The development of tagging frameworks and other automated QA technologies opens up new avenues to consider in dealing with these, embedding governance directly into the agile development lifecycle itself [4][5][6].

Received: August 09, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

A quality tagging and QA system has the potential to balance speed against control and allow teams to get tracking plans shipped quickly with data being accurate, complete, and trusted. Such a framework, as outlined in this paper, addresses how the framework can be scaled across more than one agile team without negatively affecting governance. The paper begins by highlighting the shortcomings of traditional analytics models when applied in an agile setting before examining the elements of a scalable governance architecture and then taking a look at the technological and organizational enablers that can scale such structures [7][8].

This guide is intended to assist organizations in integrating adequate analytics governance into these agile engagements to result in more reliable, consistent, and influential product data-based decision-making by establishing a foundation built on tangible implementation strategies and empirical evidence in recent scholarly and industry research [9][10].

2. Challenges of Analytics in Agile Environments

Moving on to more definite friction points of agile contexts, it can be seen that the standard practices of analytics are not necessarily made in such a way that they are agile in nature. In a traditional software development model, analytics instrumentation has been managed centrally by a specialized analytics or data engineering team in charge of defining event taxonomies, administering schema repositories, and controlling data flow. In agile teams where engineers, designers, and product managers work autonomously, this centralized model, however, can become a bottleneck and can slow development down, decreasing flexibility [11][12]. Consequently, most of the agile teams are decentralized and control their tracking analytics without pertaining to a global schema. This makes it more agile, but it regularly means inconsistency in the naming of events, undocumented definitions of metrics, duplication, and missing properties. This absence of metadata standardisation may cause downstream analytics to be problematic or invalid, leading to low confidence in dashboards and experimentation results. Such inconsistencies can be particularly troublesome when a firm grows and different teams are releasing to common data pipelines [13][14].

Additionally, the focus on shipping features rapidly in the agile environment can lead to prioritizing at the expense of analytics planning, such that instrumentation is an afterthought. This reactive model brings in lags in both data availability and in the alignment of the business questions and data gathered, as well as repeated problems in the tracking of behavior. Otherwise, without going through the critical QA procedures, teams might only detect tracking problems after campaigns or experiments are complete, at which point the data may not be salvaged [15][16]. To tackle these expanding needs, progressive organizations are starting to re-imagine analytics as a top-quality element of the agile development effort. This transition requires the implementation of models that integrate data quality assurance and collaborative documentation that culminate in the establishment and execution of a scalable analytics governance architecture.

Received: August 09, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

While the previous section explored the structural causes of analytics challenges in agile environments, the following table categorizes some of the most frequent instrumentation errors encountered by agile teams and their corresponding root causes.

Table 1: Common Instrumentation Errors in Agile Teams and Their Root Causes

Instrumentation Error	Root Cause	Impact
Missing tracking events	Tracking requirements not defined in user stories	Leads to incomplete data and poor analysis
Duplicate events	Multiple developers tracking the same action without coordination	Skews metrics and introduces data noise
Inconsistent event naming	Lack of standardized taxonomy or naming conventions	Hinders cross-team data aggregation
Incorrect property data types	Improper implementation or lack of validation checks	Causes downstream ETL and dashboard errors
Tracking in the wrong user flow	Misunderstanding of feature behavior or business logic	Misinforms analysis and misleads decision-making
Delayed event delivery	Network issues or client-side SDK misconfiguration	Affects real-time dashboards and experimentation

3. The Case for a Scalable Tagging and QA Framework

As a solution to the existing gap between agile development practices and analytics reliability, the scalable tagging and QA framework becomes the first-layer solution. The framework can be thought of as the glue between the desire of the product teams to have autonomy and the desire of an organization to have consistency, reliability, and regulatory compliance in analytics data. Unlike a model that centralizes all decision-making, the framework gives guardrails, templates, and automated validation functions that enable groups to maintain data tracking properly and reliably [17][18].

The most central element of the framework is the tagging plan, which is a well-organized depiction of user actions, business data, and product-related occasions. An extensible tagging model is based on pre-defined taxonomies, naming schemas, and metadata schemas, and is deployed to make sure that all events are defined in a similar way. Notably, it encourages modularization and, in this way, the groups establish local sets of events but provide these sets with connections to global standards. Such modularity is a necessity to scale many teams and avoid creating too stiff constraints [19][20].

Received: August 09, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The second pillar of the framework relates to quality assurance. Automated quality assurance (QA) tools sweep through event payloads in both staging and production deployments, flagging exceptions like missing properties, incorrect types, or undocumented events. Such tools may be combined with continuous integration/continuous deployment (CI/CD) pipelines, so that there is no opportunity to publish defective tracking to the production environment. This type of real-time QA is essential to the maintainability of data in an agile environment where a high amount of shipping is made on the code [21][22]. In addition, the framework encourages a common source of truth (commonly adopted formulas that are usually a version-controlled tracking specification (e.g., in YAML or JSON) and saved in an equivalent wording contention re-depositories). This guarantees openness and cross-functional collaboration that allows co-owning of the analytics life cycle by the developers, analysts, and product managers. Within more implementation, we discuss the movement of the principles of such frameworks into practical architecture in the world of agile teams.

4. Implementing the Framework in Agile Teams

With the need and some fundamentals of what makes a tagging and QA framework scale defined, we now shift gears and discuss its operational application in a day-to-day agile product team, presented in Figure 1. To make such a framework a part of agile routines, it must correspond to the main agile principles, including the possibility of iterative delivery, team self-organization, frequent feedback, and cross-functional teamwork. Instead of imposing governance as top-down mandates, it should become part of the rhythm of agile by tooling, process integration, and ownership models that can be shaped to fit the team structure and product lifecycle [23][24].

To implement, the process starts with an established standard tagging specification template. This specification uses this as a canonical form to specify user interactions, with event names, triggers, property keys and values, and metadata thereof (e.g., feature flags, A/B test identifiers, privacy classifications). Teams construct tracking plans on a per-sprint or epic basis with this template and save these in version-controlled repositories along with application code. This facilitates traceability and change management, two of the fundamentals of auditability and collaborative development [25][26]. In order to enhance agile delivery, the tagging process should always be a part of sprint planning and backlog grooming. The product managers formulate the measures of success and user behaviors that they want to monitor, whereas the developers and analysts discuss implementation-related subjects. These tracking requirements are spelled out in user stories or sub-tasks with acceptance criteria that consist of functional validation, as well as tracking validation. The practice in question is to establish the analytics instrumentation into the definition of done of all new features [27][28].

The process of QA is implemented by using automated validators and manual checks of instruments during a sprint cycle. These validations examine compatibility with the names standards, the inheritance of mandatory qualities, and compatibility with previous schemes of data. In staging environments, validation tools may also be used to simulate events and to test delivery to analytics services, e.g., Segment, Amplitude, or Google Analytics. In companies whose DevOps is more advanced, such validations are incorporated into CI / CD pipelines as

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

a gate to give the assurance of a deployment [29][30]. Importantly, the implementation should be rolled out in a flexible way with a view to taking account of the varying cadences and maturity levels of agile teams. Teams are going to want to deploy code on a daily basis, and others do two-week sprints, and the governance structure should allow both to occur, but not be cumbersome. Scalability is aided with modular tagging plans, reusable templates, and self-service documentation portals, all of which protect agility.

This implementation is not purely technical; it also involves organizational behavior change. Teams must embrace co-ownership of analytics, requiring a culture that values data literacy, continuous learning, and shared accountability. To support this culture, the next section explores the role of automation and tooling in reinforcing governance without slowing down agile teams.

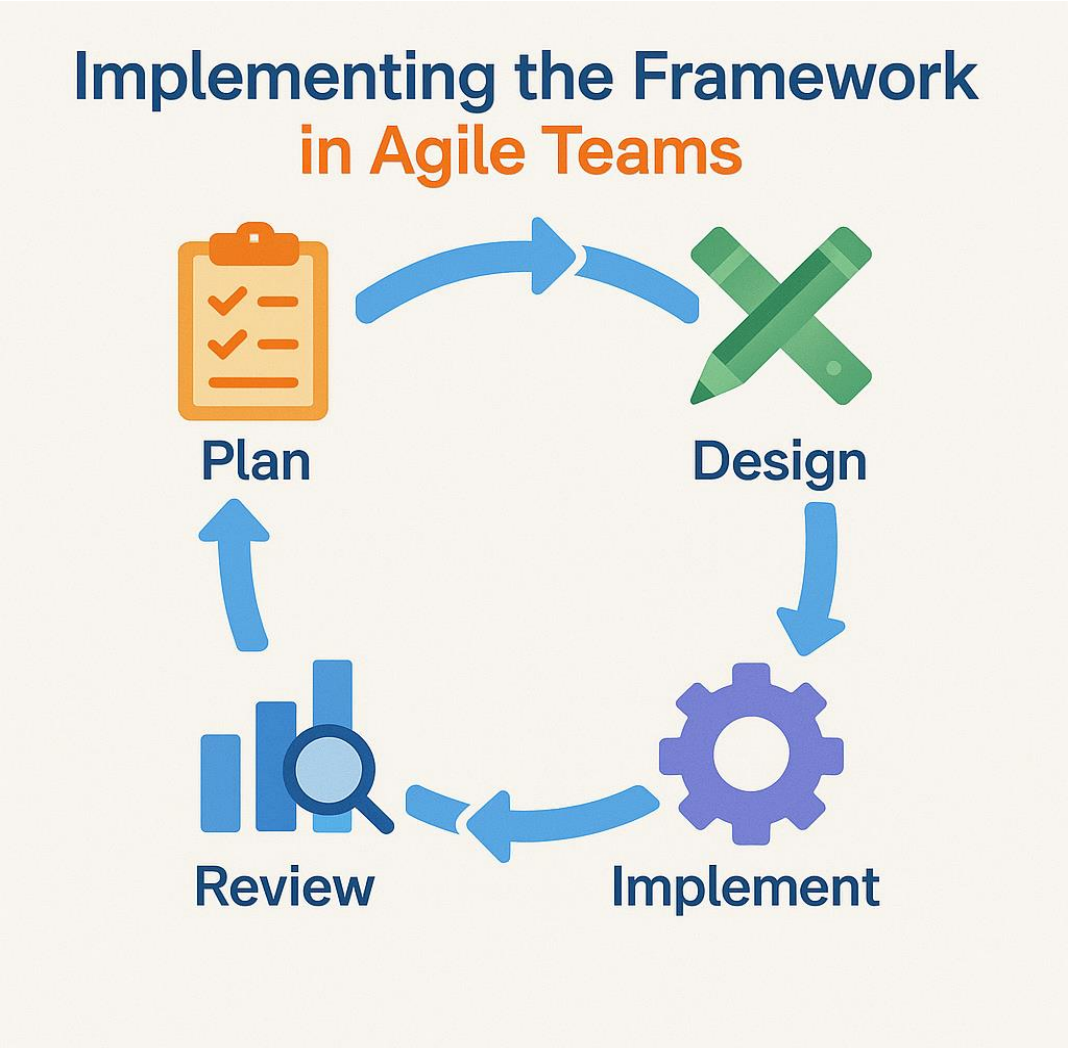

Figure 1: Diagram illustrating the cyclical process of implementing a framework in Agile teams, featuring four key stages: Plan, Design, Implement, and Review; connected in a continuous feedback loop that promotes iterative development and continuous improvement

Received: August 09, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

5. Tooling and Automation for Scalable Governance

Entering the depth of how to operationalize analytics governance, automation stands out as the key contributor to balancing agility and control. Agile product teams do not have the time to manage data instrumentation as they update features with manual processes, with errors. Automation provides consistent, low-weight enforcing of tracking standards to release time to innovate and guarantee the integrity and stability of analytics data [1][4][10].

The schema validation is one of the central components of automation. Event comparison tools can be used to compare incoming event payload against pre-defined schema, like Snowplow, The Tracking Plan API offered by Amplitude, and open-source event comparison tools. These schemas state the form, type of expected data, and required fields in each event. Real-time outliers are flagged in automation, so risk analysis is lower to receive the broken or malformed data in the analytics warehouses [12][15]. Automated tagging assistants built as an IDE plug-in or a browser extension, these tools guide developers to write the proper event code according to the tagging specification as agreed upon by the team. These tools will make suggestions to the developers in auto-complete terms about event names, display required properties, and even insert snippets of instrumentation code. This eliminates human error, provides consistency, and a speedier implementation, particularly in complicated front-ending environments [13][16]. Another very important feature is governance dashboards. These dashboards compile tracking health metrics of event volume patterns, schema compliance levels, error rates, and time in delivering events. By bringing these KPIs out in a convenient display, they enable teams to track/address problems with greater ease. This is achieved through integration with alerting tools to be notified of data breaks, schema drift, or missing events in time [18][22].

documentation on all instrumented events and their definitions, owners, and contexts of use is stored in at least metadata registries and analytics catalogs at the organizational level. Presumably, such catalogs are frequently version-controlled and searchable, allowing self-discovery and minimizing duplicate version tracking. Lineage tracking, the ability to display how raw event data is related to dashboards, KPIs, and experiments, has also been provided on some platforms to support compliance, debugging, and reproducibility.

Enforcement is automated as well. Policies like Open Policy Agent (OPA) can be used to specify and govern policies of where and how the tracking can be implemented. To give some examples, a rule may not allow sending personal identifiers in a particular area, or certain names of events that do not match the organizational standards. One can perform evaluations of such policies in pre-commit hooks or during reviews of pull requests, limiting governance drift without affecting the speed of development. In combination, these tools form a technological safety net whereby the agile teams can operate fast and always trust the data. Nevertheless, no matter how and by what such tools can substitute for it, the effective human collaboration remains important. In the transition, we will proceed to look at the organizational structures and the team collaboration models that make the adoption of the framework across multiple agile teams successful.

Received: August 09, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

To supplement our discussion on automation, the table below outlines the key categories of tools that enable scalable analytics governance in agile teams, along with their primary use cases and typical examples.

Table 2: Tooling Categories for Scalable Analytics Governance

Tool Category	Primary Use Case	Example Tools
Schema Validation Engines	Enforce event structure and prevent malformed payloads	Snowplow Inspector, Amplitude Schema Validator
Tagging Specification Editors	Maintain tracking plans using structured templates	Iteratively, Segment Protocols
Automated QA Integrations	Catch tracking bugs in CI/CD pipelines	Custom scripts, GitHub Actions, CI hooks
Real-Time Observability Dashboards	Monitor data quality metrics (e.g., dropouts, delays)	Monte Carlo, Datadog, custom dashboarding via Grafana
Event Metadata Catalogs	Document and search events, owners, and schema evolution	Atlan, DataHub, Amundsen
Policy Enforcement Gatekeepers	Prevent violations of naming conventions or PII tracking	Open Policy Agent (OPA), pre-commit hooks

6. Cross-Team Collaboration and Ownership Models

Although the backbone of scalable governance of analytics is tooling and automation, collaboration and ownership remain essential, with clear goals in successful deployment within the agile teams. Data governance in a decentralized agile organization cannot be centralized on a centralized data or analytics team. This should be replaced by a distributed model of ownership, in which every product team is responsible for the quality and consistency of its analytics data [3][9][14].

One of the usual trends is the creation of cross-functional analytics champions in every portion. Such people (typically product analysts, technical product managers, or developers with a high level of data orientation) are intermediaries between the core analytics governance team and their product organizations. They facilitate the admission of tagging plans, see the QA practices followed, and elevate the problems when needed. In this model, governance becomes decentralized, and the model is anchored to international standards [7][17]. There are a lot of organizations that develop a virtual data council or analytics guild to support such a decentralized structure. This group meets on a regular basis to check tagging patterns, address schema conflicts, update the joint documentation, and share best practices. It turns into a platform on which the standards are discussed, as opposed to imposed. The method used in this

Received: August 09, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

collaboration will help develop the semblance, but at the same time, consider the team autonomy and the rhythm of the development [11][20].

The collaboration also gets enhanced through shared platforms and documentation centers. The name conventions, taxonomy definitions, QA processes, and onboarding guides are the sources of truth most commonly found in Wikis or Confluence pages, or even custom documentation portals. Having this knowledge easily available means that teams know how to apply governance properly and do not depend on consistent central oversight to do so [6][24]. Furthermore, analytics governance has to be formally present during team ceremonies. The tracking requirement discussions should be part of the sprint planning, the backlog grooming should be part of tracking plan updating, and the analytics quality should be the reflections used in the retrospectives. Building governance into everyday rituals means governance is not something with which agile is compliant, but instead an intrinsic part of agile practice.

Collaboration is further enhanced by incentive alignment. Teams should be measured not just on product delivery, but on the quality and completeness of their analytics implementation. Metrics such as tracking coverage, QA pass rates, and event documentation completeness can be incorporated into OKRs or team scorecards. This reinforces accountability while fostering a sense of pride in delivering high-quality data. By building a culture of shared ownership and cross-functional collaboration, organizations can ensure that their analytics governance framework scales sustainably. But even with collaboration in place, ongoing success depends on the ability to monitor effectiveness, identify gaps, and evolve practices continuously, an idea we now explore in the context of monitoring and feedback loops.

7. Monitoring, Feedback Loops, and Continuous Improvement

Once the collaboration and ownership layers are in place, the second pillar in scalable analytics governance is continuous monitoring and improvement of tagging, as well as the QA processes. Agile approaches are about iteration and about a continuous learning process, and this should be reflected in the evolution of analytics governance frameworks over time. A tagging and QA framework must be designed and not simply left at that because organizations need to measure its success, adjust it to the comments of the teams, and ensure that it evolves with products and organizational growth [1][4][8].

Monitoring is also technical and organizational. Technically, the analytics observability systems should monitor a variety of indicators: schema conformance, event dropout rates, instrumentation latency, QA failure rate, and coverage gaps. These measures constitute the main body of instrumentation health dashboards that give a glimpse of how accurately data gathering is working toward meeting organizational expectations. These metrics should be reviewed consistently, and any quality problems in the data should be fixed before they affect the decision. In addition to technical health, the teams must also collect qualitative feedback on the framework itself. This is possible by periodic surveys or interviews or retrospective, in which the product managers, developers, and analysts can ponder on the ease of use, clarity, and friction points in the tagging processes. Do tagging templates have an intuitive feel? Do you see an unhealthy number of false positives in your QA tools? Is the documentation of

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

processes available and current? These people-focused assessments give valuable feedback in the areas where governance practice should be improved [13][15].

The other tip is the employment of feedback loops allied with production results. As an example, should an experiment fail because of missing or broken tracking, post-mortems must examine how the governance structure either assisted or failed to anticipate the problem. These post-incident reviews can then be used to make changes to tooling, the process, or training. An experienced structure will use learnings to update playbooks or documentation to ensure that the same issues do not recur on the different teams [16][18]. To encourage nimbleness, companies can test their governance modification within a smaller group of teams before it can be implemented among others in an organization. Likewise, a potentially improved automated QA capability could be implemented and evaluated in two speedy product groups and modified according to their response. The given iterative, experimental method frames the evolution of governance as adhering to agile principles and exposes less of a risk of implementing changes either too rigid or too disruptive.

Continuous improvement also applies to taxonomy evolution. As products mature and user behaviors change, tracking requirements will shift. Governance teams must proactively revisit metric definitions, update tracking schemas, and sunset outdated events. Using version control and changelogs ensures that changes to analytics structures are well-documented, traceable, and backward-compatible where necessary.

Ultimately, the success of monitoring and continuous improvement hinges on leadership buy-in. Executive sponsorship can help prioritize governance as a strategic enabler of data-driven product development. Leaders must champion the idea that analytics quality is not just a technical concern but a core requirement for achieving business goals. This strategic alignment sets the stage for long-term success and enables organizations to move toward a state of data maturity where agile decision-making is powered by reliable, well-governed analytics. With this vision in mind, we conclude by summarizing key learnings and exploring how future developments in technology and organizational behavior may further support scalable, agile analytics governance.

8. Conclusion and Future Directions

The agile product development, together with the analytics governance interface, is a challenge and an opportunity. With digital products increasing in their dynamism, their personalization, and data intensity, the reliability and consistency of analytics take primary importance. The nature of agile delivery is decentralized and fast, and therefore can overload traditional governance models, resulting in broken tracking, unreliable insights, and non-standard definitions of metrics. To alleviate this, organizations should adopt governance tactics that are as nimble as the teams they are supporting.

The paper has described one such scalable tagging and QA framework. Teams can achieve high levels of data quality and speed, as well as autonomy, by integrating standardized tagging plans, automated QA toolkits, metadata registries, and feedback loops into the agile lifecycle. Domains such as cross-functional teams, lack of centralized ownership, and sponsorship of the

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

top leadership further contribute to the framework's success and scalability. Into the future, there are a number of emerging trends that are going to dominate the future of agile analytics governance. First, the emergence of declarative analytics infrastructure, in which tagging specifications are encoded as programs and optimized to platform-specific implementations, provides new means of overcoming tracking inconsistencies. Second, the development of AI-powered QA tools can automate the verification of events, anomaly detection, and schema drift detection at scale. Not only will these tools alleviate the overhead associated with manual processes but it will also be able to find the root cause of instrumentation problems more precisely.

Third, ethical data practices and privacy rules will become more significant in governance. Tagging frameworks should have in-built capabilities of user consent, data minimization, and regional compliance. This brings complexity as well as the assurance that organizations are future-proofing their analytics infrastructure with the shifting standards of law. Lastly, with an increasing more organizations focusing on product-led growth, analytics will also gain further importance in the experiment parameters, onboarding optimization, and user retention. Here, analytics governance is not simply an operational requirement; it is an important strategic competency that allows teams to move and learn quickly and create superior products. To sum up, analytics governance on agile product teams cannot be a luxury anymore or an afterthought. It is one of the fundamental conditions of scaling data-informed decision-making in dynamic and changing environments. The scalable tagging and QA framework described in this paper provides a realistic, flexible way forward to organizations that want to reconcile the pace of agile with the quality of reliable analytics.

References

[1] Pinto, G., Castor, F., & Liu, Y. D. (2014, May). Mining questions about software energy consumption. In Proceedings of the 11th working conference on mining software repositories (pp. 22-31).

[2] Qi, H., Guo, Y., Hou, D., Xing, Z., Ren, W., Cong, L., & Di, X. (2022). SDN-based dynamic multi-path routing strategy for satellite networks. Future Generation Computer Systems, 133, 254-265.

[3] Siddik, M. A. B., Shehabi, A., & Marston, L. (2021). The environmental footprint of data centers in the United States. Environmental Research Letters, 16(6), 064017.

[4] Chukwurah, N., Ige, A. B., Idemudia, C., & Eyieyien, O. G. (2024). Integrating agile methodologies into data governance: Achieving flexibility and control simultaneously. Open Access Research Journal of Multidisciplinary Studies, 8(01), 045-056.

[5] Swarup, S., Shakshuki, E. M., & Yasar, A. (2021). Task scheduling in the cloud using deep reinforcement learning. Procedia Computer Science, 184, 42-51.

Received: August 09, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[6] Mittal, S., Verma, G., Kaushik, B., & Khanday, F. A. (2021). A survey of SRAM-based in-memory computing techniques and applications. Journal of Systems Architecture, 119, 102276.

[7] Hosseini, S. M., Soltanpour, Y., & Paydar, M. M. (2022). Applying the Delphi and fuzzy DEMATEL methods for identification and prioritization of the variables affecting Iranian citrus exports to Russia. Soft Computing, 26(18), 9543-9556.

[8] Ng, B. Y., & Kankanhalli, A. (2012). Information systems for large-scale event management: a case study. Pacific Asia Journal of the Association for Information Systems, 4(3), 3.

[9] Alhumam, A., & Ahmed, S. (2024). Software requirement engineering over the federated environment in distributed software development process. Journal of King Saud University-Computer and Information Sciences, 36(9), 102201.

[10] Gade, K. R. (2023). Event-Driven Data Modeling in Fintech: A Real-Time Approach. Journal of Computational Innovation, 3(1).

[11] Irshad, M., Börstler, J., & Petersen, K. (2022). Supporting refactoring of BDD specifications—An empirical study. Information and Software Technology, 141, 106717.

[12] Budacu, E. N., & Pocatilu, P. (2018). Real Time Agile Metrics for Measuring Team Performance. Informatica economica, 22(4).

[13] Shahin, M., Babar, M. A., & Zhu, L. (2017). Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices. IEEE access, 5, 3909-3943.

[14] Welten, S., Neumann, L., Yediel, Y. U., da Silva Santos, L. O. B., Decker, S., & Beyan, O. (2021). DAMS: a distributed analytics metadata schema. Data Intelligence, 3(4), 528-547.

[15] Chiñas-Palacios, C., Aguila-Leon, J., Vargas-Salgado, C., Garcia, E. X., Sotelo-Castañon, J., & Hurtado-Perez, E. (2021). A smart residential security assisted load management system using hybrid cryptography. Sustainable computing: Informatics and systems, 32, 100611.

[16] Xu, S., Koren, I., & Krishna, C. M. (2021). Adaptive workload adjustment for cyber-physical systems using deep reinforcement learning. Sustainable Computing: Informatics and Systems, 30, 100525.

[17] Grady, N. W., Payne, J. A., & Parker, H. (2017, December). Agile big data analytics: AnalyticsOps for data science. In 2017 IEEE international conference on big data (big data) (pp. 2331-2339). IEEE.

[18] He, Z., Ning, D., Gou, Y., & Zhou, Z. (2022). Wave energy converter optimization based on differential evolution algorithm. Energy, 246, 123433.

Received: August 09, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[19] Akerele, J. I., Uzoka, A., Ojukwu, P. U., & Olamijuwon, O. J. (2024). Increasing software deployment speed in agile environments through automated configuration management. International Journal of Engineering Research Updates, 7(02), 028-035.

[20] Read, N., & Cosgrove, P. (2021). Comments on:"Carbon emissions and costs associated with subsidizing New York nuclear instead of replacing it with renewables". Journal of Cleaner Production, 290, 125835.

[21] Bhanushali, A. (2023). Impact of automation on quality assurance testing: A comparative analysis of manual vs. automated qa processes. International Journal of Advances in Engineering Research, 4, 26.

[22] Bicer, Y., Sajid, M. U., & Al-Breiki, M. (2022). Optimal spectra management for self-power producing greenhouses for hot arid climates. Renewable and Sustainable Energy Reviews, 159, 112194.

[23] Kabel, C. M., Cruz, A., Rosga, A., Esparrago Lieu, T., & Blackmur, N. (2020). Collaborating Within to Support Systems Change: The Need For—and Limits of—Cross-Team Grantmaking. The Foundation Review, 12(1), 11.

[24] Zhou, L., Nandal, A., Ansari, I. S., Polat, K., Polak, L., Dhaka, A., ... & Alenezi, F. (2023). Secrecy outage probability and limiting threshold criteria in an optimal SNR regime by maximizing desired transfer rate. Internet of Things, 22, 100789.

[25] Peters, R. J., & Özsu, M. T. (1997). An axiomatic model of dynamic schema evolution in objectbase systems. ACM Transactions on Database Systems (TODS), 22(1), 75-114.

[26] Tran, H., Zdun, U., Holmes, T. I., Oberortner, E., Mulo, E., & Dustdar, S. (2012). Compliance in service-oriented architectures: A model-driven and view-based approach. Information and Software Technology, 54(6), 531-552.

[27] Hu, M., Luo, X., Chen, J., Lee, Y. C., Zhou, Y., & Wu, D. (2021). Virtual reality: A survey of enabling technologies and its applications in IoT. Journal of Network and Computer Applications, 178, 102970.

[28] Schneeweiss, S., & Glynn, R. J. (2018). Real-world data analytics fit for regulatory decision-making. American journal of law & medicine, 44(2-3), 197-217.

[29] Ibrahim, M. M. A., Syed-Mohamad, S. M., & Husin, M. H. (2019, February). Managing quality assurance challenges of DevOps through analytics. In Proceedings of the 2019 8th International Conference on Software and Computer Applications (pp. 194-198).

[30] Hildebrandt, T., van Dongen, B. F., Röglinger, M., & Mendling, J. (2022). Selected papers of BPM 2019-editorial to the special issue. Information Systems, 104, 101902.

Received: August 09, 2025