From Regulatory Constraints to Continuous Assurance: The Constraint-Practice-Evidence Framework for Regulated Agile-DevOps Environment

Sumaiya Tabassum Antara[1,*,†], Kari Smolander[1,†]

[1]Department of Software Engineering, LUT University, Yliopistonkatu 34, 53850 Lappeenranta, Finland

Abstract

Regulated industries like banking, healthcare and energy operate under strict regulatory requirements that mandate frequent formal approvals, end-to-end traceability, security validation and auditable compliance controls. At the same time, these organizations are increasingly applying Agile and DevOps work practices for faster delivery. While prior research suggests that regulatory standards are not inherently incompatible with iterative software development, there remains limited theoretical understanding about how regulatory constraints are operationalized within modern CI/CD environments. This research investigates the adaptive pathways through which compliance requirements can be integrated into regular Agile-DevOps workflows. The empirical foundation combines five semi-structured interviews and thirteen published industry case studies across compliance sensitive sectors, and the analysis of this combined data is performed using a unified thematic coding approach and cross-source triangulation. The findings present core regulatory constraints faced by the organizations, adaptive practices the teams are implementing and auditable evidence artifacts produced within CI/CD pipelines. The patterns from these findings reveal a consistent transformation mechanism and propose a conceptual Constraint-Practice-Evidence (C-P-E) framework, which provides a structured model for embedding compliance into automated delivery systems, contributes analytical clarity to governance-aware software engineering and outlines practical guidelines for DevOps teams to embed compliance within the Agile-DevOps pipelines without compromising iterative delivery principles.

Keywords

Agile-DevOps, Regulatory Compliance, Continuous Assurance, CI/CD, C-P-E Framework

CEUR-WS.org/Vol-4309/paper10.pdf

1. Introduction

Software-intensive regulated industries like banking, healthcare and energy sectors perform their operations under strict governance, compliance requirements and audits [1]. These organizations are rapidly adopting Agile and DevOps practices to accelerate delivery, enhance the quality of software and thus responding to the market demands. While they are increasingly adopting these modern software engineering practices, a persistent tension also arises for aligning them with regulatory demands. Regulatory frameworks and organizational governance require traceability, control mechanisms, formal approvals and document verification activities whereas Agile and DevOps practices mainly prioritize iterative software development, automation, rapid feedback and continuous deployment. Although both of their objectives aim to improve the reliability and quality of products and services, the operational logics of them might result in misalignment because governance and regulations mainly emphasize accountability and stability, whereas Agile-DevOps practices put emphasis on delivery speed and adaptability [2].

This tension can be particularly seen in banking, healthcare and critical infrastructures where software delivery processes are directly impacted by compliance requirements. In these sectors, structured control points are enforced by the organizational and regulatory requirements related to change advisory board (CAB) approvals, security verification, segregation of duties and privacy protection throughout the

TKTP 2026: Annual Doctoral Symposium of Computer Science, 8.—9.6.2026 Turku, Finland

*Corresponding author.

†These authors contributed equally.

sumaiya.antara@lut.fi (S. T. Antara); kari.smolander@lut.fi (K. Smolander)

0009-0003-1412-7766 (S. T. Antara); 0000-0002-7043-0458 (K. Smolander)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-02

software delivery lifecycles. These restrictions can slow down the continuous delivery pipelines when they are applied as external review stages instead of being applied as integrated processes [3, 4]. As a result, organizations under strict regulatory controls frequently struggle to maintain balance between iterative delivery and governance mandates.

The compatibility of Agile and DevOps with regulatory norms has been the subject of many earlier research. Those studies indicate that standards like ISO-based quality management systems or medical device regulations do not necessarily prohibit iterative development as long as traceability, verification and documentation criteria are met [5]. Similarly, previous research in compliance-driven and security-critical environments show how automated testing, pipeline gates and policy enforcement tools can help to keep things under control without going back to waterfall-style procedures [6, 7]. These studies indicate that agility and regulatory compliance are not mutually exclusive. However, there are still two significant gaps in the current literature. First, many of the studies mainly focus on single sectors or specific regulatory standards, limiting cross-contextual insights [8]. Second, the obstacles and isolated practices are well indeed widely reported, but only fewer studies systematically link the regulatory constraints to the adaptive practices teams can apply and the concrete evidence artifacts the auditors ultimately accept. Achieving compliance requires more than just aligning the policies and it requires proving that controls are functioning effectively. When the delivery settings become more automated, this proof takes the form of pipeline generated artifacts instead of static documentation. Therefore, it is consequently essential to understand how regulatory requirements are operationalized into continuous evidence within the CI/CD workflows. If compliance is not understood from this viewpoint, it could be only viewed as an after-development external documentation layer rather than as an inherent feature of the software development practices.

However, there is still a lack of theoretical understanding about the transformation from regulatory demand to automated assurance in regulated sectors. This research responds to this persistent gap by examining how organizations operationalize regulatory compliance within Agile-DevOps workflow settings and how these types of integrations reshape and enhance the software delivery practices. To be more specific, it investigates the mechanisms through which regulatory requirements are parallelly integrated into routine development activities and regular workflows by enabling the continuous production of traceable and auditable evidence. By doing this, this study shifts the abstract discussions of governance frameworks toward how compliance can be implemented into daily activities of the organizations. The empirical foundation of this research combines qualitative data from five in-depth practitioner interviews along with evidence from thirteen published industry case studies across banking, healthcare, energy and fintech sectors. These industries were chosen because they operate under continuously evolving and strict regulatory compliance and this provides a comparative understanding of how context influences implementation strategies. Then a triangulation is performed with interview narratives and secondary case studies documentation which strengthens the analytical rigor and significantly enhances the transferability of the findings. This study does not conceptualize compliance solely as a restrictive control which slows down the innovation. Also, we do not reduce the discussion only to some technical tooling adjustments. Instead, the study aims to figure out a recurring transformation pattern across various cases.

The analysis of the collected data leads to the development of a Constraint-Practice-Evidence (C-P-E) framework through iterative coding and cross-case synthesis. This framework conceptualizes compliance as a structured transformation mechanism embedded with CI/CD pipelines and explains how regulatory constraints can be translated into governance-by-design practices and how these practices can systematically generate continuous assurance artifacts, which are acceptable to auditors. This paper makes three contributions. First, it identifies the dominant cross-sector regulatory constraint categories impacting and shaping the Agile-DevOps practices. Second, it categorizes the adaptive practices that make governance operationalized within the automated workflows. And third, it proposes the C-P-E framework as a conceptual model linking the regulatory challenges, adaptive mechanisms and continuous evidence generation. These contributions provide an analytical clarity for researchers and practical guidance for organizations. The findings suggest that when compliance is systematically integrated in regular workflows, it evolves from being an external obligation and post-hoc auditing

exercise and becomes an internalized capability and built-in property of modern software engineering processes.

2. Background and Related Works

Agile and DevOps practices mainly aim to accelerate the software delivery with iterative development, automation and continuous integration. However, their adoption in regulated environments introduces structural tensions between delivery speed and controls [2, 9]. Regulated sectors like banking, healthcare and critical infrastructure must fulfil the strict governance requirements including segregation of duties, formal change approval processes, traceability mandates and security verification, which directly influence the continuous software delivery workflows [3, 4]. Previous research reflects that regulatory standards are generally process-agnostic and these do not necessarily restrict Agile practices as long as iterative methods, verification, validation and documentation requirements can demonstrably be fulfilled [5]. The empirical studies on medical devices and safety critical domains have clearly highlighted how completeness rules and risk documentation requirements can slow down the release cycles when they are implemented as manual checkpoints rather than being integrated as embedded controls [6, 3]. Similarly, the financial and industrial context studies show that segregation of duties (SoD) constraints and change advisory board (CAB) approvals often create bottlenecks unless they are directly embedded within pipeline logics [10, 11, 12]. Additional layers of security compliance introduce automated SAST, DAST and dependency scans by making them mandatory gating mechanisms within CI/CD environments [7, 13].

The GDPR and other privacy legislation have increased the requirements for compliance by mandating data masking, privacy-by-design principles and documented impact assessments throughout the lifecycle [14, 15]. Research indicates that increasing audit readiness and delivery predictability can be achieved by including such criteria directly into CI/CD pipelines rather than handling them as post development inspections [6, 7]. According to the research comparing governance frameworks like COBIT and Scrum, by systematically mapping the control objectives to workflow artifacts, the conflicts between traditional steering structures and Agile roles can be mitigated [10].

Despite these advancements, the current literature is still fragmented across sectors and standards. Although many studies have pointed out the difficulties or detail individual adjustments, very few offer a cross-sector synthesis that can connect the regulatory restrictions to concrete workflow practices and evidence-based artifacts that auditors accept as proof. Furthermore, the systematic shift from regulatory demand to continuous evidence generation within pipelines is rarely conceptualized as an integrate approach despite the frequent mention of automation and compliance-as-code in the existing studies. By synthesizing the cross-sectional patterns and outlining the systematic relationship between constraints, adaptive practices and auditable evidence artifacts, this research advances the empirical discussion and presents compliance as an integrated feature of contemporary delivery systems by shifting its role away from being an external documentation load.

3. Research Process

In this study, a qualitative and interpretive research design approach has been applied to examine the way Agile and DevOps practices are currently implemented under strict regulatory constraints. Within the scope of this research, the software development and delivery team and operating CI/CD pipelines in regulated industries are the core units of analysis. The qualitative approach in this research focuses on understanding how teams are adapting their software development and operations processes, tools and compliance mechanisms, and balancing agility with strict regulatory controls. This study is a combination of semi-structured interview data with secondary case study-based evidence. This combined approach facilitates the cross-contextual comparison and improves the robustness of research outcomes. Then, to identify the patterns across these data sources, a unified analytical approach of Constraint-Practice-Evidence (C-P-E) framework was implemented.

3.1. Data Collection

The main data source for this research was five semi-structured interviews which were conducted with senior practitioners from software development, delivery and assurance teams of highly regulated companies. The interview participants were selected by subjective sampling which ensured that the participants had direct experience with Agile-DevOps practices under strict regulatory constraints. The roles of the participants mainly included Lead DevOps Engineers, Software Development Team Leads and IT auditor. The primary focusing sector for the interviews was banking and financial institutions which was additionally supported with interviews from energy sector for supporting the cross-sector comparison. The participants represented organizations from Finland, France, United States of America and Bangladesh, providing perspectives from both European and non-European regulatory settings. The interview questionnaire covered the core areas of the research focusing on these three research questions: **RQ1:**What specific constraints limit the implementation of Agile and DevOps practices in compliance sensitive industries?, **RQ2:**How do teams adapt their processes, tools and culture to meet both agility goals and regulatory demands?, **RQ3:**Which practical pathways successfully link regulatory constraints to the right practices and acceptable evidence? Every interview lasted between 40 minutes to 50 minutes and were recorded with the consent of the participants. The interview transcripts were anonymized which were extensively utilized for the data analysis.

To make the empirical grounding stronger, a triangulation process was applied to this research where the interview data were triangulated with thirteen published case studies from peer-reviewed journals and well-documented industry reports [16, 5, 17, 18, 19, 4, 20, 21, 11, 14, 22, 23, 24]. The cases include several scenarios or situations from the banking, energy, healthcare and medical devices, industry systems etc. areas which explicitly describe the adoption of Agile and DevOps under strict regulatory frameworks like ISO 9001, IEC 62304, IEC 62443, COBIT and GDPR. The cases that had sufficient methodological clarity and contextual description, were considered for including in scope. And these cases were selected based on their ability to describe the regulatory constraints, report effective process adaptations against those constraints and reference the observable evidence artifacts. The complete view of the research design has been shown in Figure 1 where it reflects the flow from research questions to the final integrated results.

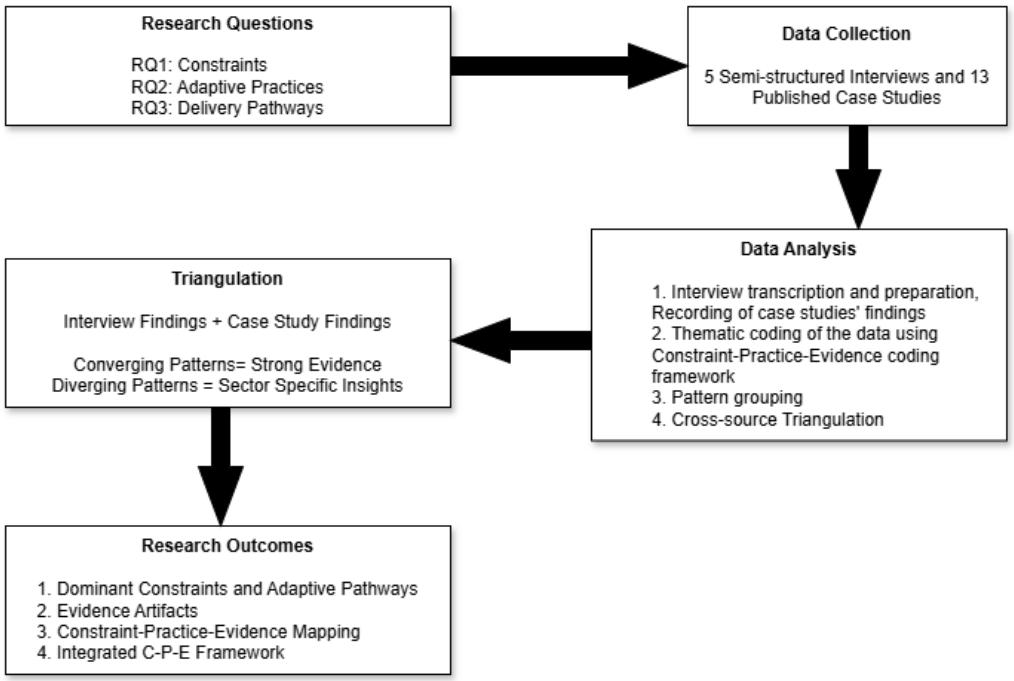


Figure 1: Overall Research Design and Data Flow

3.2. Data Analysis

All interviews and case studies were analysed with a thematic coding approach following constraint, practice and evidence structure. This structure emerged during the thorough review of interview scripts, and this clear pattern was also observed in every case study as well, which later created the idea of developing the Constraint-Practice-Evidence (C-P-E) framework. The term Constraints refer to governance and regulatory requirements that limit or impact the Agile-DevOps workflows. Practices refer to the adaptive mechanisms that are implemented by teams to operate within these constraints. Evidence denotes the metrics or artifacts generated by these adaptive practices which can be accepted during audits and inspections. The coding of data was performed in four stages. Firstly, an initial open coding was done for interview and case study data. Secondly, the similar codes were grouped into higher level themes. Thirdly, a cross-source comparison was done between interviews and case studies. Finally, the patterns were mapped with research questions to find out the C-P-E pathways across sectors.

For triangulation, the interview findings were systematically compared with case study evidence. Some patterns were identified in both datasets which were considered as convergent, and the context specific patterns were considered as sector nuances. So, this triangulation approach strengthened the analytical credibility and reduced reliance on single source interpretation.

3.3. Trustworthiness and Limitations

Several approaches were adopted in this study to ensure the trustworthiness, including a transparent coding process, a consistent application of the analytical framework and cross-source validation. Although the interview sample is limited in size, this limitation has been mitigated with the integration of multiple published case studies. This integration supports analytical transferability by not pushing towards statistical generalization. In addition, this research does not aim to provide any legal interpretation of regulatory standards, and it also does not perform evaluation of any specific tools. The findings from the analyzed data are limited to delivery pathways and evidence generation mechanisms of Agile-DevOps pipelines.

4. Findings

This section presents the synthesized core findings derived from the cross-source analysis of semi-structured interviews and industry case studies across highly regulated industries like banking, medical, fintech and energy sectors. An iterative coding procedure combining cross-case comparison and within-case analysis, was applied to analyze the collected data. The analysis revealed three analytically distinct yet interconnected dimensions. Accordingly, the results are structured in 1) dominant regulatory constraints, 2) adaptive practices adopted by teams and 3) continuous evidence mechanisms that operationalize compliance within the CI/CD pipelines. Collectively, these dimensions form the foundation of the proposed Constraint-Practice-Evidence (C-P-E) framework, which demonstrates that regulatory requirements systematically influence the pipeline architecture and workflow design rather than considering compliance as an external and static organizational layer.

4.1. Dominant Regulatory Constraint Categories

Four recurring categories of regulatory constraints were identified as dominant across all sectors and data sources. These constraints function as structural boundaries which mainly affects the application of Agile-DevOps practices.

We found **restricted development environments** to be one of the most dominant categories of constraints. This category reflects the restrictions of external and cloud by data residency policies and mandatory compliance reviews before release. As a result, the development activities are limited to pre-approved environments and tools, and testing to synthetic or non-production datasets. In addition,

deployment activities remain limited to approved business-hour windows and there are multilevel formal code approval requirements. This constraint was consistently reflected in the interview data. One participant described how organizational cloud policies directly affected the selection of development tools and access management: Interviewee X- *"Postman used to be very commonly used application, but then they changed the policy and they wanted to store the scripts into their own cloud space in the US. That wasn't okay with the company. So we had to stop using it. Also, we review access to different systems every couple of months. I have to testify why I need certain accounts periodically. If I don't act on it, I will lose the access."*. This quotation exemplifies how regulatory requirements extend beyond technical implementation to influence routined development activities, tool selection and periodic access governance.

Then we identified **access control and segregation of duties (SoD) category** as another dominant category which captures strict role separation and manual handover requirements, complex authentication and multi-level approval for external integrations, developer exclusion from deployment operations, OTP-secured remote sessions affecting pipeline flow and periodic access reviews disrupting the CI/CD workflows.

The analysis also revealed **delay in workflow** as another dominant constraint category which highlights the factors that extends and slow down the release cycle. The factors include mid-sprint compliance reviews, mandatory CAB approval before production deployment, pre-integration unit testing enforcement, mandatory DPIA and privacy-by-design validation steps, manual pre-deployment security validation, lengthy compliance verification and multi-step authentication and approval for external collaboration.

Finally, we identified **organizational regulatory burden** as another dominant category affecting Agile-DevOps implementation. This category reveals the structural and procedurals factors like multi-level bureaucratic approval chains slowing down the decision-making process, mandatory 100% test coverage requirements, developer's responsibility for compliance documentation tasks, legacy systems and limited DevOps expertise constraints, steering governance vs product owner authority tension, independent QA requirements under COBIT, regulatory misalignment within quality management systems, safety critical V&V documentation burden, release completeness requirement before approval, ISO Clause 7 documentation and record-keeping mandates etc.

4.2. Adaptive Practices within Workflows

In response to these key constraint categories, several adaptive practice categories were identified through the thematic analysis of the interview and case study data. The interview data demonstrated how automated security controls have become embedded within CI/CD pipelines rather than remaining as external review activities. Interviewee Y explained: *"We use SonarQube for code smell and testing coverage, integrated with our pipeline. The other tool is MEND. It is a vulnerability detection tool that scans the whole Docker image for commonly reported vulnerabilities. Anytime they detect a critical vulnerability, the pipeline aborts, irrespective of any executive sign-off. The pipeline totally blocks the code from being released"*. These practices indicate that the organizations do not replace Agile-DevOps principles to address and mitigate the core constraints. Instead, they systematically redesign their workflows to internalize the governance mechanisms.

We identified **automated access governance** as one of the key adaptive practice categories adopted by the organizations to address the regulatory constraints. This category emphasizes the discontinuation of non-compliant cloud tools under automated access control, periodic access reviews with approved tool enforcement, business-hour deployment restrictions with expert override control, automated deployment via Kubernetes service accounts and RBAC, automated governance of approved IDEs, toolsets and license quotas and continuous user access auditing in development platforms.

Another key adaptive practice category is **automated governance pipeline** which highlights the use of SonarQube and MEND-integrated vulnerability gating in CI/CD, automated enforcement of secure coding standards in APIs, risk-based CAB gating embedded within CI/CD pipelines, automated generation of compliance evidence per pipeline run, real-time logging (ELK) and automated risk scoring

(Jira), automated compliance reporting for audit efficiency, SAST pre-merge gates and pipeline embedded CAB checkpoints, DORA and audit controls integrated into GitLab CI and Terraform, Standardized CI/CD templates with protected branches and service accounts, automated unit testing as pre-merge compliance gate, DPIA validation and DAST integration across DevOps stages, automated cross-team tracking and workflow synchronization, automated testing integration within SAFe pipelines etc.

In addition, **customized workflow** was observed to be another key adaptive practice category adopted by the organizations. It highlights how organizations adjust their development process to incorporate compliance requirements without disrupting any Agile and DevOps practices. The specific practices include implementing data encoding and masking aligned with predefined PII standards, compliance workload integrated into Agile estimation, compliance milestones embedded strategy (80% meaningful and full for critical systems), QMS updates and staff training for traceable Agile change management, lightweight Q-interface aligning DevOps with QMS functions etc.

4.3. Continuous Evidence Mechanisms

The third analytical dimension concerns how compliance was ultimately demonstrated through system generated and automated assurance mechanisms. The shift toward pipeline generated evidence was directly reflected in practitioner accounts. As Interviewee Z explained when describing about the evidence they show during audits, *"Pipeline run report shows scan passed, policy passed, deploy approved"*. These types of evidence mechanisms across the studied organizations are reflected in three primary categories of evidence artifacts within CI/CD pipelines to support the automated assurance. The first category includes authorization related evidence such as **Approval Logs, IAM Access Reports and CAB Approval Records** capture every access control and approval level activity throughout the delivery workflow. The second category consists of system level evidence such as **Immutable Systems Logs, Security and Vulnerability Records, CAB Ticket Histories and DPIA Validation Records** provides immutable and timestamped evidence of system security testing, integrity and privacy by design enforcement. Finally, the third category includes process and audit-oriented artifacts such as **Digital Sprint Artifacts, Compliance Checklists, Audit Dashboards, Auto-generated Compliance Reports and Risk Scoring Summaries**, which serve as audit ready evidence, conform policy adherence as well as align with audit and risk management standards. These outputs indicate the transformation of compliance from static documentation and external verification layer to pipeline generated evidence artifacts.

4.4. Cross-Case Synthesis

Table 1: Constraint-Practice-Evidence Mapping

Sector	Main Regulatory Constraint (RQ1)	Adaptive Practice (RQ2)	Evidence Artifact (RQ3)
Banking and Finance	Rigid change management and segregation of duties (no developer access to production)	Introduced gated CI/CD pipelines and policy-as-code to automate the approvals and access reviews	CAB approval tickets, immutable log exports and ELK dashboards showing change history
Energy	External partner integrations required multiple logins and GDPR/MiCA compliance checks	Created continuous documentation and periodic access reviews, added policy enforcement checks for every new tool	Access review reports, policy audit logs, documented proof-of-concepts for third party tools
Healthcare	Heavy safety and verification documentation required by ISO 13485/IEC 62304 standards	Embedded a quality interface layer within DevOps pipelines to connect automated testing with compliance review	Verification and validation reports, signed test artifacts, automated traceability reports
Telecom (Large scale agile)	Distributed global teams struggled with inconsistent GDPR interpretation and late compliance reviews	Implemented shared compliance wikis and automated tracking tools to ensure consistent understanding and synchronized timelines across all teams	Wiki based change records, compliance dashboards and audit ready reports generated automatically
Industrial and Manufacturing (Security focused)	Security standards like IEC 624434-1 slowed down continuous delivery due to manual reviews	Used automated SAST/DAST scanning and security champions within teams to ensure compliance without delay	Pipeline security reports, automated vulnerability dashboards and audit snapshots of each release
Cross-sector Observation	Common challenge is balancing audit traceability with delivery speed	Core solution is implementing compliance as code which will embed all checks and policies directly into CI/CD pipelines	Automated logs and approval reports accepted by auditors as official compliance evidence

The cross-case comparison resulted in constant linkages between regulatory constraints, adaptive practices and resulting evidence artifacts across sectors. Table 4.4 shows the synthesized mapping and key patterns and highlights that adaptive practices do not only mitigate the core constraints for maintaining Agile-DevOps practices. In fact, they also transform the burden of regulatory requirements into systematic evidence generation methods. And this recurring translation dynamic established the analytical basis for the development of C-P-E framework.

4.5. Constraint-Practice-Evidence (C-P-E) Framework

The thematic integration of the identified constraints, practices and evidence relationships is represented in the conceptual C-P-E framework in Figure 2, which captures the observed translation process where regulatory controls are operationalized through workflow redesign and production of verifiable evidence.

Each pathway within the framework illustrates how automation, procedural alignment and integrated reporting mechanisms convert the regulatory constraints into auditable and repeatable processes. This integration shows that compliance can be an embedded function within software development and it is no longer an external factor to development. Thus, this resulting framework represents a systematic pipeline shift from externally enforced compliance to embedded continuous assurance, which did not occur abruptly. This transformation has emerged incrementally as the organizations have been translating the constraints into executable pipeline logics in their daily workflows and the C-P-E framework captures this translation at the analytical level.

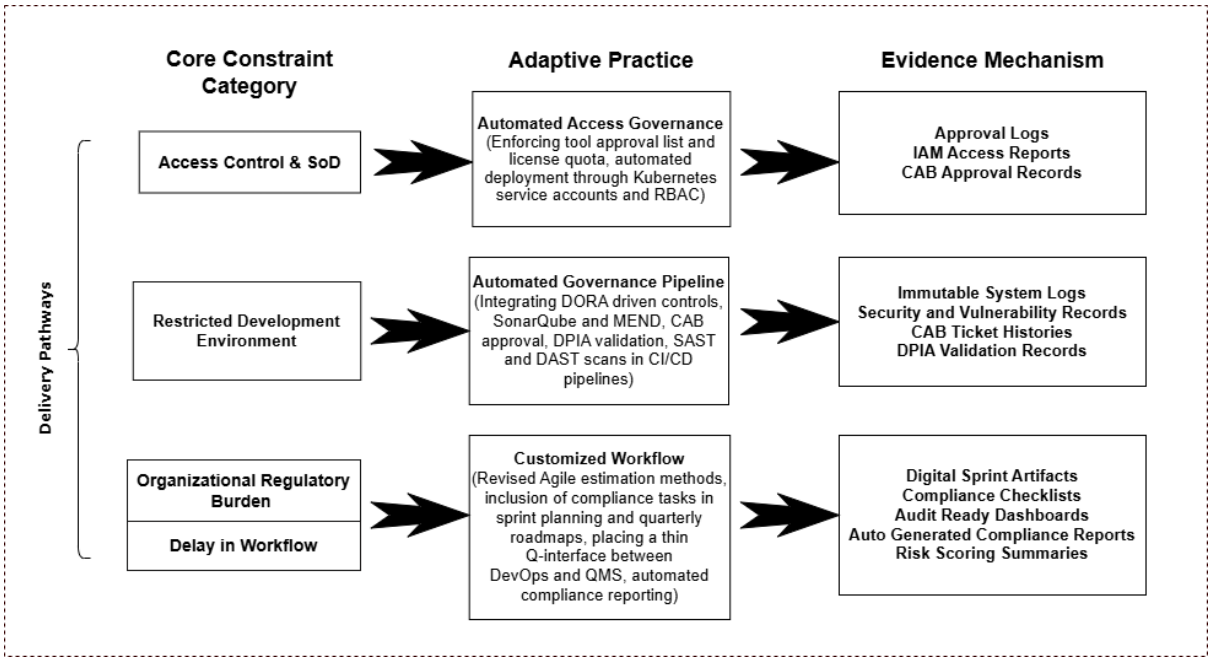


Figure 2: Constraint-Practice-Evidence (C-P-E) Framework

5. Discussion

This research aimed to figure out how regulated organizations like healthcare, banking, energy etc., are integrating Agile-DevOps practices in ways that ensure the delivery speed along with generating auditable assurance artifacts. The findings are consistent with prior research highlighting the challenges of aligning Agile-DevOps practices with regulatory and governance frameworks. For instance, existing studies have shown that compliance requirements such as segregation of duties, formal approvals and auditability often introduce structural tensions within iterative development environments [3, 6]. Similarly, research on governance frameworks like COBIT emphasizes the difficulty of reconciling flexible development practices with rigid control mechanisms [10].

However, this article extends prior works by demonstrating compliance integration as an architectural transformation within CI/CD pipelines. This perspective is captured through the proposed constraint-practice-evidence (C-P-E) framework, which provides a structured and operational understanding of governance integration in Agile-DevOps environments. Through the cross-case study and analysis of banking, healthcare, energy and fintech organizations, the findings of this study reveal that compliance assurance integration is not merely a process of aligning with regulatory standards. In fact, it is fundamentally an architectural phenomenon. The analysis of the interview and case studies data show that the high-performing organizations are internalizing governance within their CI/CD architectures, rather than treating the regulatory constraints just as external layer of checking. Regulatory constraints like structured change controls, segregation of duties, security enforcement and traceability mandates act as boundary conditions which transform the workflow design of these organizations. The resulting Constraint-Practice-Evidence (C-P-E) framework formalizes this translation mechanism by showing how each constraints push the teams to adapt their workflows that, in turn, systematically generate evidence artifacts.

Therefore, the conceptual framework reframes regulatory compliance from being a post development process to an evidence-centric delivery capability. In this way, the evidence for ensuring compliance is not produced retrospectively for auditors and it is automatically generated within the program execution logic of the continuous delivery pipelines. This artifact-centric transformation makes significant contributions to the existing Agile and DevOps literature by clarifying how regulatory demands can

be operationalized without compromising the iterative delivery goals. Besides, the results extend the theoretical understanding of governance in software development by showing compliance as an emergent property of socio-technical system design. Practically, the C-P-E framework provides a structural lens for the software development teams and auditors to reasonably think about compliance integration as a part of normal execution which reduces the cycle time and audit burden as well.

In summary, this paper presented how compliance can be effectively integrated into Agile-DevOps settings through architectural embedding within CI/CD pipelines, rather than relying on external procedural controls. The findings show that regulatory requirements can be operationalized without compromising the iterative delivery practices. The proposed C-P-E framework can support the development teams and auditors in treating compliance as a part of normal workflow execution, thereby reducing cycle time and audit overhead. The future works will focus on empirically validating the C-P-E framework through industrial case studies and real-world implementations across diverse regulated industries and assessing its effectiveness in supporting governance-aware Agile-DevOps practices.

6. Conclusions

The increasing adoption of Agile and DevOps mechanisms in compliance sensitive industries has intensified the requirement of balancing between regulatory control and delivery speed. This paper demonstrated this architectural tension between governance and agility. The findings described how the regulatory controls and continuous delivery can coexist in a stable and repeatable way when compliance requirements are directly integrated into delivery workflows. It revealed a consistent transformation pattern through cross-source analysis of practitioner interviews and industry cases, where the identified regulatory constraints shape adaptive practices in teams and these practices generate structured and auditable evidence within CI/CD pipelines. By connecting external governance requirements to workflow redesign and continuous assurance artifacts, the proposed Constraint-Practice-Evidence (C-P-E) framework captures this transformation logic and reframes compliance from being portrayed as documentation burden into an embedded property or feature of software development life cycle. This work contributes both analytical clarity and practical guidance for organizations operating under strict regulatory constraints. By embedding governance within delivery workflows, these organizations can reduce audit overhead and improve traceability along with maintaining the delivery efficiency. As regulatory environments are continuously evolving, especially in areas like AI governance and digital resilience, integrating compliance directly into the delivery pipelines is also becoming more critical. This is because the emerging regulations require continuous monitoring, traceability and real-time assurance, which cannot be effectively achieved through traditional and manual compliance processes.

Acknowledgments

We sincerely thank all interview participants for their time, openness, and valuable insights. Their contributions played a crucial role in shaping the findings and strengthening the practical relevance of this study.

Declaration on Generative AI

During the preparation of this work, the authors used generative AI tools (Grammarly, ChatGPT Go) to check grammar and spelling, refine language and explore ideas for figure design. The authors reviewed and edited the content after using these tools and take full responsibility for the publication's content.

References

[1] J. W. Rittinghouse, J. F. Ransome, Cloud Computing: Implementation, Management, and Security, CRC Press, 2017. doi:10.1201/9781439806814.

[2] B. Fitzgerald, K.-J. Stol, Continuous software engineering: A roadmap and agenda, Journal of Systems and Software 123 (2017) 176–189. doi:10.1016/j.jss.2015.06.063.

[3] T. Laukkarinen, K. Kuusinen, T. Mikkonen, Devops in regulated software development: Case medical devices, in: Proceedings of the IEEE/ACM International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), 2017, pp. 15–18. doi:10.1109/ICSE-NIER.2017.20.

[4] F. Moyón, D. M. Fernández, K. Beckers, S. Klepper, How to integrate security compliance requirements with agile software engineering at scale?, in: Proceedings of the International Conference on Product-Focused Software Process Improvement (PROFES), Springer, 2020, pp. 69–87. doi:10.1007/978-3-030-64148-1_5.

[5] N. Hrgarek, Certification and regulatory challenges in medical device software development, in: Proceedings of the International Workshop on Software Engineering in Health Care (SEHC), 2012, pp. 40–43. doi:10.1109/SEHC.2012.6227011.

[6] M. F. Lie, M. Sánchez-Gordón, R. Colomo-Palacios, Devops in an ISO 13485 regulated environment: A multivocal literature review, in: Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3382494.3410679.

[7] T. J. Wagner, T. C. Ford, Metrics to meet security and privacy requirements with agile software development methods in a regulated environment, in: Proceedings of the International Conference on Computing, Networking and Communications (ICNC), 2020, pp. 17–23. doi:10.1109/ICNC47757.2020.9049681.

[8] E. Klotins, M. Unterkalmsteiner, P. Chatzipetrou, T. Gorschek, R. Prikladnicki, N. Tripathi, L. B. Pompermaier, A progression model of software engineering goals, challenges, and practices in start-ups, IEEE Transactions on Software Engineering 47 (2021) 498–521. doi:10.1109/TSE.2019.2900213.

[9] L. Bass, I. Weber, L. Zhu, DevOps: A Software Architect's Perspective, Addison-Wesley, 2015.

[10] N. Ozkan, A. K. Tarhan, B. Goren, I. Filiz, E. Ozer, Harmonizing it frameworks and agile methods: Challenges and solutions for the case of cobit and scrum, in: Proceedings of the 15th Conference on Computer Science and Information Systems (FedCSIS), 2020, pp. 709–719. doi:10.15439/2020F47.

[11] L. Su, T. Storer, A case study of devops adoption within a large financial organisation, in: Proceedings of the IEEE International Conference on Software Maintenance and Evolution (ICSME), 2023, pp. 403–413. doi:10.1109/ICSME58846.2023.00053.

[12] A. Rahman, L. Williams, Software security in devops: Synthesizing practitioners' perceptions and practices, in: Proceedings of the International Workshop on Continuous Software Evolution and Delivery (CSED), 2016, pp. 70–76. doi:10.1145/2896941.2896946.

[13] F. Moyón, R. Soares, M. Albuquerque, D. M. Fernández, K. Beckers, Integration of security standards in devops pipelines: An industry case study, https://arxiv.org/abs/2105.13024, 2021. doi:10.48550/arXiv.2105.13024.

[14] L. Leite, D. R. dos Santos, F. Almeida, The impact of general data protection regulation on software engineering practices, Information and Computer Security 30 (2021) 79–96. doi:10.1108/ICS-03-2020-0043.

[15] P. Voigt, A. von dem Bussche, The EU General Data Protection Regulation (GDPR): A Practical Guide, Springer, 2017. doi:10.1007/978-3-319-57959-7.

[16] R. K. Gupta, M. Venkatachalapathy, F. K. Jeberla, Challenges in adopting continuous delivery and devops in a globally distributed product team: A case study of a healthcare organization, in: Proceedings of the ACM/IEEE International Conference on Global Software Engineering (ICGSE), 2019, pp. 30–34. doi:10.1109/ICGSE.2019.00020.

[17] B. Gallina, F. U. Muram, J. P. C. Ardila, Compliance of agilized (software) development processes with safety standards: A vision, in: Proceedings of the International Conference on Agile Software

Development Companion (XP), 2018, p. 14. doi:10.1145/3234152.3234175.

[18] M. Zeller, Devcertops: Strategies to realize continuous delivery of safe software in regulated domains, in: Proceedings of the IEEE/ACM International Conference on Software Engineering Companion (ICSE-Companion), 2023, pp. 334–335. doi:10.1109/ICSE-Companion58688.2023.00094.

[19] M. Usman, M. Felderer, M. Unterkalmsteiner, E. Klotins, D. M. Fernández, E. Alégroth, Compliance requirements in large-scale software development: An industrial case study, in: Proceedings of the International Conference on Product-Focused Software Process Improvement (PROFES), Springer, 2020, pp. 385–401. doi:10.1007/978-3-030-64148-1_24.

[20] A. Boer, T. van Engers, Agile: A problem-based model of regulatory policy making, Artificial Intelligence and Law 21 (2013) 399–423. doi:10.1007/s10506-013-9144-0.

[21] Z. Bakalova, M. Daneva, T. Nguyen, Standards compliance helps value creation in agile projects, in: Proceedings of the IEEE International Conference on Research Challenges in Information Science (RCIS), 2014, pp. 1–6. doi:10.1109/RCIS.2014.6861054.

[22] Özden Özcan Top, F. McCaffery, To what extent the medical device software regulations can be achieved with agile software development methods? xp, dsdm, scrum, The Journal of Supercomputing 75 (2019) 5227–5260. doi:10.1007/s11227-019-02793-x.

[23] A. Mathrani, S. Wickramasinghe, N. P. Jayamaha, An evaluation of documentation requirements for iso 9001 compliance in scrum projects, The TQM Journal 34 (2021) 901–921. doi:10.1108/TQM-08-2020-0177.

[24] U. Mukherjee, Paytm's crisis management: Regulatory compliance and market impact, Journal of International Economics 15 (2024) 82–87.