# A Mobile DevOps Project Management Framework

Farhan Khan
Sr. Engineering Leader, Postgraduate Researcher, Auckland University of Technology
Jing Ma
Director of Women, Department of Data Science and AI, Auckland University of Technology
Email: sgtroswell@gmail.com; jing.ma@aut.ac.nz

**Abstract**

Mobile applications have become critical infrastructure in finance, healthcare, and public services, yet their development and operations pipelines remain complex and disconnected. Although DevOps practices have matured in web and cloud systems, mobile environments continue to face fragmented monitoring, weak governance visibility, and inconsistent security assurance. This study addresses the problem of limited managerial control and transparency in mobile DevOps projects, where automation lacks connection to governance and decision making. The objective was to develop a unified model that integrates automation, monitoring, and governance to strengthen project oversight. Drawing from a Multivocal Literature Review of academic and practitioner studies and examined through a demonstration linking an iOS application with Sentry and Azure DevOps, the research produced the Mobile DevOps Project Management Framework. The demonstration illustrates how real time logging, alerting, and dashboards can provide continuous visibility for project managers. It illustrates how organisations can strengthen reliability, assurance, and transparency by embedding governance within continuous delivery cycles.

**Keywords**

Mobile DevOps, project governance, continuous assurance, monitoring and visibility, organisational maturity

## Introduction and Background

Mobile applications have evolved from being digital conveniences to essential services that sustain modern economies. They now support daily operations in banking, education, healthcare, and government. Each year more than two hundred and fifty billion applications are downloaded worldwide (Statista, 2024), showing society's increasing dependence on mobile technology. This widespread use has driven organisations to seek faster and more reliable software delivery methods, resulting in the global rise of DevOps practices that integrate development and operations through automation and collaboration (Bass, 2015; Forsgren et al., 2018).

Although DevOps has transformed how web and cloud systems are delivered, mobile environments remain less mature. Frequent operating system updates, device fragmentation, and the dependency on app store approvals create natural barriers to continuous deployment. In addition, monitoring, telemetry, and logging practices differ between mobile platforms, which prevents consistent visibility of system health or compliance (Zeng et al., 2019). These inconsistencies limit project managers' ability to make informed decisions, even when automation tools are in place.

This situation highlights a broader organisational issue. DevOps has traditionally focused on technical speed and efficiency, while governance and management visibility have developed more slowly (Kerzner, 2017). Mobile delivery pipelines therefore struggle to provide reliable evidence for managerial control and assurance (Harty et al., 2021). The lack of integration between automation and governance not only affects operational reliability but also weakens accountability, compliance, and organisational learning.

This study addresses that problem by developing a framework that connects automation, monitoring, and governance into one continuous management process. The research investigates how project managers can achieve visibility, assurance, and control in mobile DevOps projects where technical and managerial practices are often disconnected.

To guide this investigation, the following research questions were considered.

First, how can automation and monitoring be integrated to improve visibility and accountability in mobile DevOps environments?

Second, what governance mechanisms enable project managers to assure reliability and compliance during continuous delivery?

Third, how can a unified framework support organisational learning and sustain maturity in mobile DevOps?

By addressing these questions, the research aims to help organisations align project management with modern DevOps practices and to establish a governance model that improves reliability, assurance, and decision making in mobile software delivery.

## Literature Review and Current Challenges

Early DevOps literature focused on engineering speed, collaboration, and automated builds (Balalaie et al., 2016; Wettinger et al., 2016). More recent studies emphasise that governance must evolve alongside automation because effective control depends on how operational evidence is interpreted (Kerzner, 2017; Wiedemann et al., 2023). In mobile environments, this alignment is difficult to achieve due to inconsistent telemetry, unstable networks, and privacy limits on data collection (Harty et al., 2021; Kose, 2024).

Research consistently shows that fragmented toolchains weaken managerial visibility. Integration among build, test, and compliance systems is often poor, resulting in duplicated or missing data (Laso et al., 2022). Quality-of-service monitoring delivers value only when embedded within governance structures (Pontillo et al., 2024). Without interpretation and ownership, operational data becomes noise rather than insight.

Automation is frequently mistaken for managerial control. Continuous delivery pipelines can generate excessive alerts, and security scans expose vulnerabilities without assigning responsibility for remediation (Feio et al., 2024; Lombardi & Fanton, 2023). Such conditions create fast but fragile systems where speed undermines reliability.

Cultural maturity remains essential. (Erich et al., 2017) and (Kose, 2024) found that DevOps success relies on leadership commitment and shared understanding of measurement. Without reflective learning, even advanced automation fails to create improvement.

To explore this problem, the study conducted a Multivocal Literature Review combining academic and grey sources following (Garousi et al., 2019). Sixty-nine studies published between 2010 and 2025 were assessed and synthesised to identify recurring challenges in visibility, assurance, and governance. The findings informed the conceptual foundations of the Mobile DevOps Project Management Framework, later examined though a demonstration that integrated Sentry and Azure DevOps to connect automated monitoring with project governance.

The objectives of this study were to identify gaps limiting managerial visibility in mobile DevOps, to design a unified framework integrating automation, monitoring, and governance, and to examine it through a demonstration using Sentry and Azure DevOps. These aims ensured the research connected theoretical insight with practical application.

The review of prior work revealed a clear gap, as existing DevOps and observability models focus on technical integration but overlook the managerial processes that transform operational data into governance insight. This gap prompted the development of the Mobile DevOps Project Management Framework.

## Theoretical Foundations

The research problem addressed in this study concerns the lack of governance visibility and managerial oversight in mobile DevOps environments. This challenge sits within the broader area of project governance and management control theory. Governance in projects is not limited to structural approval gates. It is a system of controls that links operational behaviour to organisational objectives through measurement, feedback, and accountability as described by (Kerzner, 2017) and (PMI, 2021). Recent work on DevOps teams reinforces this understanding by framing DevOps as a control environment that combines process controls, information controls, and cultural controls rather than a practice focused only on technical execution. This study adopts that perspective and treats mobile DevOps pipelines as governance mechanisms in which telemetry, automation, and workflows operate as instruments of control.

Within this control-based lens, the four core constructs in the Mobile DevOps Project Management Framework are interpreted as governance controls rather than engineering capabilities. Integration reliability functions as an information control. It determines whether the data feeding dashboards and audit artefacts can be trusted as a sound basis for managerial decisions. Automation and visibility function as diagnostic and interactive controls. They provide continuous feedback that allows managers to observe performance and intervene when required. Continuous assurance represents boundary and risk controls by embedding security and compliance rules directly into the delivery pipeline so that violations are detected early. Organisational learning and maturity align with cultural and learning controls. They ensure that incidents and telemetry are converted into shared norms, improved practices, and higher capability across teams. Taken together, these constructs position the framework directly within project governance and management control theory. They show how technical automation can be governed through structured information flow, timely feedback, and repeated learning cycles.

Mobile DevOps functions as an interdependent system of tools, workflows, and human decision making. Studies such as (Erich et al., 2017) and (Kose, 2024) show that technical failures often arise from organisational issues such as unclear ownership, limited analytical capability, or gaps in shared understanding. This perspective aligns with contemporary socio-technical thinking, which emphasises that technical automation and organisational practices must evolve together. Interpreting mobile DevOps in this way helps explain why integration reliability, visibility, assurance, and learning must be treated as governance controls rather than purely engineering activities.

This theoretical perspective supports the study in two ways. First, it guides the analysis of evidence in the literature review by interpreting findings about logging, monitoring, and governance as findings about how control is created or weakened in mobile DevOps environments. Second, it provides the foundation for the core contribution of the study. The Mobile DevOps Project Management Framework does not offer another tooling pattern. It offers a governance-oriented control model for mobile DevOps that brings information controls, diagnostic feedback, boundary controls, and learning mechanisms together within a single repeatable cycle.

## Methodology

The study used a Multivocal Literature Review to combine academic and practitioner evidence on mobile DevOps governance. This method was selected because mobile engineering practices evolve quickly and rely heavily on community platforms, technical blogs, and industry documentation. The review followed the process outlined by (Garousi et al., 2019) and was adapted to the specific needs of this project. The search covered the period from 2010 to 2025. Academic studies were identified through Google Scholar, Scopus, IEEE Xplore, ACM Digital Library, SpringerLink and ScienceDirect, while grey literature was sourced from engineering blogs, industry reports, cloud provider documentation, and mobile analytics platforms. Search terms related to mobile DevOps, monitoring, logging, observability, continuous assurance, governance, cultural maturity, security integration, telemetry, and toolchain reliability, and these terms were applied in different combinations to ensure broad coverage.

All records were screened in three stages. The first stage removed duplicates and sources unrelated to mobile delivery. The second stage applied inclusion criteria that required each study to provide evidence about logging practices, monitoring reliability, security validation, cultural maturity, or governance visibility. The third stage excluded sources that did not contain empirical findings or that focused only on general DevOps topics without mobile applicability. This process resulted in sixty-nine studies that were suitable for analysis. Each study was assessed using criteria adapted from standard software engineering review guidelines. The criteria considered clarity of method, relevance to mobile DevOps, depth of evidence, and reliability of the findings. Studies that did not meet a basic threshold of clarity or relevance were removed, and the remaining studies provided a balanced mix of academic and practitioner insights.

Data was extracted into structured tables that captured author information, publication year, context, problem findings, and DevOps practice details. The extracted material was coded through an iterative process. Initial codes described issues such as fragmented telemetry, weak correlation of logs, gaps in security checks, excessive manual intervention, and unclear ownership of monitoring practices. These codes were repeatedly reviewed and refined to identify patterns across the studies. The refined codes were clustered into broader themes that represented recurring challenges in mobile DevOps. These themes included integration reliability, visibility and governance, continuous assurance, and cultural maturity. Each theme was linked to direct evidence from the studies. The themes were then used to define the foundational elements of the Mobile DevOps Project Management Framework. The complete protocol is available in the extended research report.

### *Mapping Literature Themes to Framework Components*

**Table 4.1**

*Mapping of Literature Theme to Framework Components*

| Framework Component | Supporting MLR Themes | Evidence from Reviewed Studies |
|---|---|---|
| Integration Reliability | Fragmented telemetry, inconsistent logs, weak cross tool correlation | (Harty et al., 2021) reported inconsistent logging in mobile analytics. (Zeng et al., 2019) showed weak correlation across SDKs. (Batoun et al., 2024) identified schema reliability issues. (Wettinger et al., 2016) highlighted toolchain misalignment. |
| Automation and Visibility | Limited visibility, weak monitoring integration, missing contextual data | (Laso et al., 2022) observed gaps in connected monitoring systems. (Pontillo et al., 2024) identified limited insight into performance behaviour. (Feio et al., 2024) noted excessive manual review of alerts. |
| Continuous Assurance | Late security checks, manual validation, lack of compliance alignment | (Lombardi & Fanton, 2023) described security scanning delays. (Feio et al., 2024) found limited automation in security testing. (Madhushanie et al., 2025) identified manual privacy checks. |
| Governance Alignment | Poor decision visibility, unclear accountability, inconsistent reporting | (Kerzner, 2017) emphasised the need for measurable governance. (Wiedemann et al., 2023) presented DevOps as a control environment. (Laso et al., 2022) showed weak integration between metrics and governance. |

| | | |
|---|---|---|
| Organisational Learning and Maturity | Cultural gaps, limited capability building, fragmented ownership | (Erich et al., 2017) highlighted cultural maturity challenges. (Kose, 2024) observed leadership and measurement gaps. Studies across the review linked adoption success to reflective learning. |

The following examples illustrate how findings from the reviewed studies informed each component of the framework. (Harty et al., 2021) observed that mobile analytics logs often differ in structure, completeness, and reliability, which directly supports the need for integration reliability within the framework. Similarly, (Zeng et al., 2019) reported that log messages generated by different mobile SDK implementations are inconsistent, reinforcing the importance of correlated telemetry across tools. (Lombardi & Fanton, 2023) noted that security validation in many delivery pipelines tends to occur late in the process, which aligns with the framework's emphasis on continuous assurance. (Erich et al., 2017) found that successful DevOps adoption depends strongly on shared ownership, collaboration, and organisational learning, supporting the inclusion of cultural maturity and learning capability as core elements of the model. These examples demonstrate how empirical evidence from the literature shaped each component of the proposed framework.

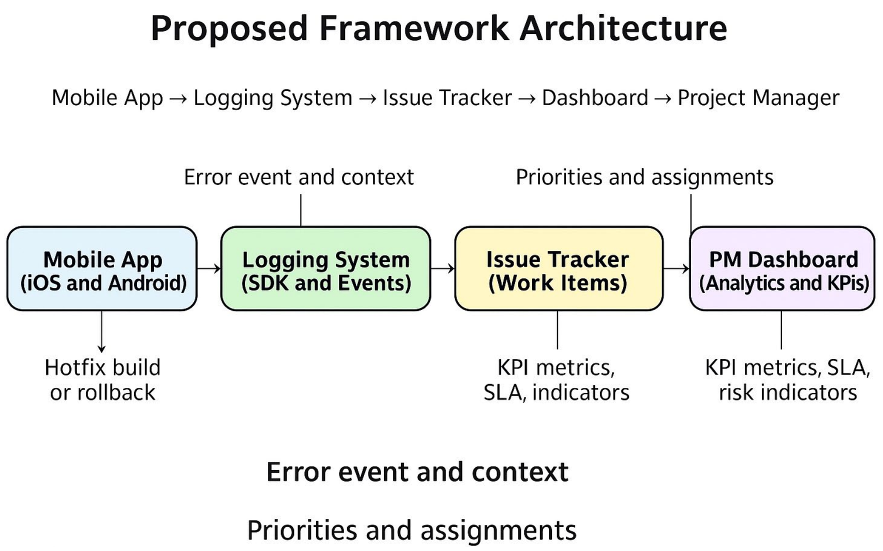
## Mobile DevOps Project Management Framework

The framework was developed through a structured design process informed by both academic research and practitioner insights. A Multivocal Literature Review following (Garousi et al., 2019) was conducted to identify recurring challenges in mobile DevOps projects, focusing on visibility, assurance, and governance. Findings from sixty-nine studies were coded and synthesised into thematic dimensions that shaped the five foundational elements of the framework. These conceptual findings were then refined through an iterative design science approach, aligning framework components with established project management standards such as those proposed by the (PMI, 2021) and (Kerzner, 2017).

Figure 5.1 presents the framework's architecture. It is designed to translate operational events into management insight, ensuring that reliability, assurance, and learning are embedded throughout the delivery process. The model is built on five foundational principles. Integration reliability maintains consistency of telemetry across all connected tools. Automation and visibility provide a shared data environment that links technical and managerial decision-making. Continuous assurance embeds security and compliance validation directly within development workflows. Governance alignment connects operational metrics with organisational objectives, enabling measurable accountability. Organisational learning ensures that each iteration generates knowledge and capability growth.

**Figure 5.1**

*Proposed Framework Architecture*



The Mobile DevOps Project Management Framework differs from traditional DevOps, DevSecOps, and observability approaches because it treats mobile delivery as a governance system rather than a technical automation pipeline. DevOps practices focus on speed and integration, while DevSecOps extends this with security checks. Observability frameworks concentrate on system insight through telemetry, tracing, and metrics. These models provide technical capability but do not explain how operational evidence becomes part of project governance or how managers interpret that evidence for decision making. The proposed framework addresses this gap by linking information controls, diagnostic feedback, boundary controls, and learning mechanisms within a single management cycle. It also differs from web or server-based DevOps because mobile environments require distributed telemetry, platform specific constraints, and delayed release cycles, which limit visibility and complicate assurance. The framework therefore offers a governance-oriented control model that integrates managerial oversight with mobile specific automation and monitoring constraints. Its originality lies in positioning mobile DevOps as a socio technical control system that supports organisational decision making rather than a sequence of engineering tasks.

**Table 5.1**

*Comparison of Related Approaches and the Proposed Framework*

| Approach | Primary Focus | Limitation for Mobile | Distinction from Proposed Framework |
|---|---|---|---|
| DevOps | Speed and continuous integration | Relies on uniform environments absent in mobile | Framework introduces governance controls and decision visibility |
| DevSecOps | Integrated security checks | Emphasises scanning but not managerial assurance | Framework connects security events with governance and accountability |
| Observability Models | Telemetry and insight | Focus on signals but not ownership or interpretation | Framework embeds telemetry inside governance cycles |
| Web and Server DevOps Models | Server-side pipelines | Do not address platform diversity and delayed releases | Framework is tailored to mobile complexity and distributed telemetry |

The Mobile DevOps Project Management Framework operates through a seven-stage cycle that transforms mobile delivery into a continuous governance process. It begins with event planning, where measurable indicators and risks are defined. Instrumentation and logging follow, ensuring telemetry is auditable and well-structured. Continuous integration and deployment automate builds while retaining traceability. Alerts and tickets are generated automatically, making every issue visible to management. Dashboards then consolidate technical and governance information for real-time monitoring. Controlled release and validation confirm stability before deployment, while review and learning close the loop by capturing insights for future improvement.

Performance indicators summarised in Table 5.2 demonstrate how the framework enhances maturity. Integration reliability improves data correlation and traceability (Batoun et al., 2024; Wettinger et al., 2016). Visibility and governance shorten managerial decision cycles (Feio et al., 2024; Pontillo et al., 2024). Continuous assurance reduces audit preparation time (Lombardi & Fanton, 2023). Cultural maturity grows through structured review and training (Erich et al., 2017; Kose, 2024).

**Table 5.2**
*Framework Performance Indicators*

| Dimension | Maturity Contribution | Observed Improvement | Supporting Studies |
|---|---|---|---|
| Integration Reliability | Stable feedback pipeline | Demonstrated high consistency in data correlation and error traceability across tools | (Batoun et al., 2024; Wettinger et al., 2016) |
| Visibility Governance | Actionable dashboards and reviews | Faster managerial decision cycles due to continuous data refresh and shared dashboards | (Feio et al., 2024; Pontillo et al., 2024) |
| Continuous Assurance | Automated security validation and evidence | Significant reduction in audit preparation time through integrated compliance checks | (Lombardi & Fanton, 2023) |
| Cultural Maturity | Learning and training programmes | Noticeable increase in DevOps adoption supported by structured learning and leadership involvement | (Erich et al., 2017; Kose, 2024) |

The framework establishes a balanced ecosystem in which automation supports management oversight rather than replacing it. To validate its effectiveness in practice, a real-world demonstration was conducted using Sentry and Azure DevOps.

## Framework Demonstration

This demonstration explores the practical behaviour of the framework's principles rather than validating the conceptual model. It examines how telemetry moves through the workflow and how automation supports governance visibility. Full validation of the framework requires organisational testing and expert evaluation, which is noted as future research.

To examine the practical behaviour of the framework's principles, a practical demonstration was conducted to test its core principles of integration reliability, visibility, and continuous assurance. A prototype iOS application was instrumented to generate operational telemetry that flowed through Sentry and Azure DevOps pipelines. This environment was designed to simulate a real delivery context where automation directly supports project governance. The demonstration assessed response time, data traceability, and auditability, verifying that each event could be captured, logged, and converted into a governance artefact visible to project managers.

The demonstration aimed to test the framework's principles of integration reliability, governance visibility, and continuous assurance. A prototype iOS application related to Sentry for monitoring and Azure DevOps for issue tracking and dashboards. The setup replicated a realistic delivery pipeline in which user actions, such as login, data entry, and simulated crashes, automatically generated telemetry events that were automatically logged in Sentry and converted into work items in Azure DevOps.

**Table 6.1**
*Timestamp Analysis of Event Flow*

| Scenario | User Action (iOS App) | Sentry Log Time (UTC) | Azure DevOps Record Time (UTC) | Observed Delay (seconds) |
|---|---|---|---|---|
| Login: Invalid Credentials | 4:00:04 PM | 4:00:06 PM | 4:01:08 PM | 64 |
| IPO: Incorrect Submission | 4:20:19 PM | 4:20:23 PM | 4:22:13 PM | 114 |
| IPO: Correct Submission | 4:20:55 PM | N/A | N/A | N/A |
| Simulated Crash | 4:24:40 PM | 4:24:49 PM | 4:25:26 PM | 46 |

Timing analysis confirmed near-real-time synchronisation, with differences between systems of less than 2 minutes. Efficiency improved significantly compared with manual log handling reported in earlier studies (Batoun et al., 2024; Madhushanie et al., 2025). Table 6.2 summarises these gains, showing a marked reduction in manual correlation and validation work. Figures 6.1 also illustrates how the dashboards provided live visibility into incidents and their corresponding tasks, creating an unbroken chain of accountability from event to resolution.

**Table 6.2**

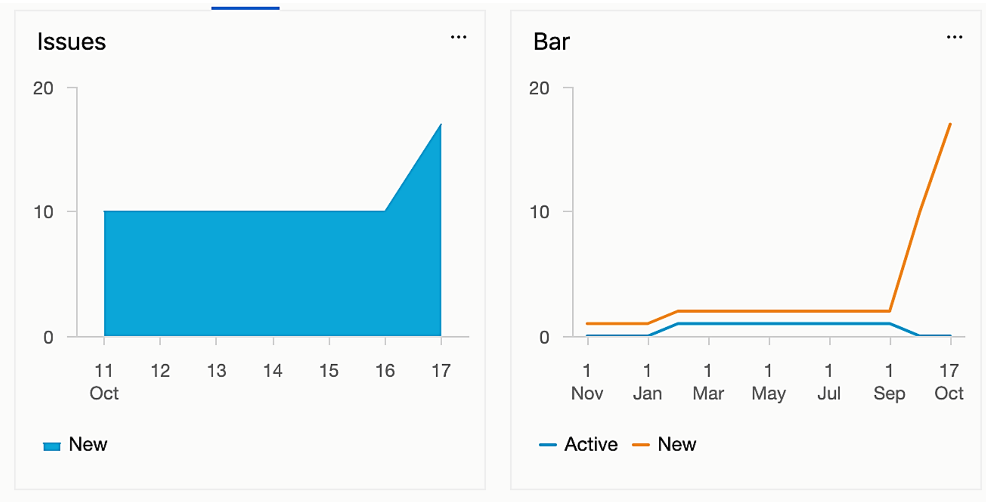
*Comparison of Manual Baseline and Automated Demonstration Results*

| Process Area | Baseline Manual Effort | Key Academic Source | Automated Outcome | Relative Improvement Observed |
|---|---|---|---|---|
| Alert triage and classification | Manual review and categorisation of large volumes of alerts | (Madhushanie et al., 2025) | Automated ticket creation in Azure DevOps directly from Sentry events | Manual triage replaced by automated correlation |
| Log correlation across tools | Reconciliation of logs between build, monitoring, and deployment systems | (Wettinger et al., 2016) | Automatic linking of logs and work items across systems | Human correlation eliminated |
| Data verification and schema validation | Repeated manual checks of configuration and schema consistency | (Batoun et al., 2024) | Validation checkpoints enforced automatically within the pipeline | Human verification replaced by automated validation |

The demonstration showed that automation and governance can operate seamlessly together. Logs became auditable records, dashboards delivered real time visibility, and the results confirmed the framework's practical value in turning governance principles into measurable outcomes.

**Figure 6.1**

*Sample Azure DevOps Dashboard for Project Managers*




### *Limitations of the Study*

The study has several limitations that must be acknowledged to provide transparency regarding the scope of the findings. The demonstration examined the framework in a controlled setting using a single prototype iOS application connected with Sentry and Azure DevOps. This environment allowed close observation of telemetry flow and governance visibility but did not represent the diversity of tools, organisational structures, or delivery constraints found in real projects. The behaviour of the framework therefore reflects an illustrative example rather than evidence gathered from field deployment.

The Multivocal Literature Review synthesised a wide range of academic and practitioner sources, yet the evidence relied on published studies and available documentation rather than primary data from organisations. Grey literature can be uneven in methodological depth and may reflect specific tooling contexts that are not generalisable. Although quality criteria were applied, variability in evidence strength remains a constraint.

The framework was conceptualised and examined through design principles and demonstration rather than through organisational case studies, expert evaluations, or longitudinal assessment. As a result, the study does not claim empirical validation of the framework's outcomes. The demonstration confirmed that telemetry can move reliably through connected systems, but it did not test leadership behaviour, organisational maturity, governance processes, or cultural adoption. Future studies are needed to evaluate how the framework performs across different organisations, mobile platforms, compliance environments, and levels of DevOps maturity.

### *Future Research Directions*

Future research should extend the framework beyond controlled demonstrations and examine its effectiveness within real organisational settings. Field based case studies can explore how the model performs when teams operate under varied constraints such as strict compliance rules, diverse toolchains, and complex governance structures. These studies would provide evidence about adoption behaviour, leadership influence, and the quality of managerial decisions produced through the framework. A Delphi study involving experts in mobile DevOps, governance, and software project management would offer systematic evaluation of the framework and help establish consensus around its conceptual validity. Longitudinal studies can also assess how the framework influences organisational maturity, capability progression, and the development of data driven governance habits. Further work may investigate the application of the model across different mobile ecosystems such as Android, Flutter, and cross platform architectures, as well as expanding the concept of continuous assurance with predictive monitoring and artificial intelligence to support proactive project decision making. These directions would establish the empirical grounding required to confirm the value of the framework in practical project management environments.

## Project Management Implications

The framework and its demonstration together illustrate that mobile DevOps can be managed as an integrated system of technical automation and organisational governance. This section interprets these findings in the context of project management practice, focusing on how the framework redefines managerial roles, enhances decision visibility, and promotes continuous assurance.

The Mobile DevOps Project Management Framework positions project managers not merely as task coordinators but as custodians of data reliability and governance. When governance principles guide automation, it delivers transparency and predictability. Managers can observe incidents as they occur and make informed decisions before issues escalate. This aligns with (Kerzner, 2017) argument that performance control depends on measurable accountability and with the (PMI, 2021) standard that defines governance as integral to organisational success.

Dashboards play a transformative role in this process. They are no longer technical artefacts but instruments of governance. By allowing engineers and managers to share the same visual data, the framework encourages mutual understanding and trust. Performance discussions shift from a retrospective to a continuous approach, fostering agile leadership and collective ownership.

Continuous assurance emerges as a critical managerial capability. Automated policy validation brings compliance into everyday operations, reducing risk while accelerating delivery. This is particularly valuable in sectors such as finance and healthcare, where real-time governance is essential for compliance (Lombardi & Fanton, 2023).

Finally, learning completes the governance cycle. Data becomes valuable only when it drives behavioural change. Regular review sessions and capability building ensure that each cycle strengthens analytical maturity and leadership insight. This supports (Kose, 2024) conclusion that cultural evolution, rather than tool adoption, determines the long-term sustainability of DevOps transformation.

## Conclusion

This research developed and examined a comprehensive framework that unites automation, monitoring, and governance within mobile DevOps environments. The Mobile DevOps Project Management Framework responds to the long-standing absence of a structured model that connects technical automation with managerial oversight. The framework was informed by a Multivocal Literature Review that synthesised academic and industry sources, and its principles were examined through a live demonstration using Sentry and Azure DevOps.

The findings show that governance and assurance can be embedded directly into automation processes. When automation is properly aligned with project management objectives, it enhances visibility, accountability, and reliability across the delivery pipeline. The results illustrate that automated monitoring not only reduces manual workload but also transforms operational data into governance evidence that managers can use for real time decisions. This demonstrates that automation can serve as both a technical enabler and a managerial control mechanism.

The study contributes to the project management discipline by extending traditional governance concepts into continuous delivery environments. It provides a practical model for integrating performance measurement, compliance, and organisational learning within mobile software projects. The framework also contributes a replicable approach for combining DevOps toolchains with management standards such as those outlined by the (PMI, 2021) and (Kerzner, 2017).

## References

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables DevOps: Migration to a cloud-native architecture. *IEEE Software*, *33*, 42-52. https://doi.org/10.1109/MS.2016.64

Bass, L. E. N. (2015). *DevOps: A software architect's perspective*. Addison-Wesley. https://research.ebsco.com/linkprocessor/plink?id=8ddc48bb-9282-3423-9c66-1d038728c63c

Batoun, M. A., Sayagh, M., Aghili, R., Ouni, A., & Li, H. (2024). A literature review and existing challenges on software logging practices. *Empirical Software Engineering*, *29*(4), 103. https://doi.org/10.1007/s10664-024-10452-w

Erich, F. M. A., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. *Journal of Software: Evolution and Process*, *29*(6), e1885. https://doi.org/10.1002/smr.1885

Feio, C., Santos, N., Escravana, N., & Pacheco, B. (2024). An empirical study of DevSecOps focused on continuous security testing. *2024 IEEE European Symposium on Security and Privacy Workshops* (EuroS&PW), https://research.ebsco.com/linkprocessor/plink?id=017de6be-ac91-3048-a16c-58d92af30cf1

Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The science of lean software and DevOps: Building and scaling high performing technology organizations* (First edition ed.). IT Revolution Press. https://research.ebsco.com/linkprocessor/plink?id=fa410a9c-f446-36d8-b237-29846c3dc1c6

Garousi, V., Felderer, M., & Mäntylä, M. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. https://doi.org/https://doi.org/10.1016/j.infsof.2018.09.006

Harty, J., Zhang, H., Wei, L., Pascarella, L., Aniche, M., & Shang, W. (2021). Logging practices with mobile analytics: An empirical study on Firebase *2021 IEEE/ACM 8th International Conference on Mobile Software Engineering and Systems* (MobileSoft). https://research.ebsco.com/linkprocessor/plink?id=f1191c35-9dc6-3a29-99ef-8700e57c1a64

Kerzner, H. (2017). *Project management: A systems approach to planning, scheduling, and controlling* (Twelfth edition. ed.). Wiley. https://research.ebsco.com/linkprocessor/plink?id=bee68b01-87e2-3bd5-ad75-6f69a1939e16

Kose, B. O. (2024). Mobilizing DevOps: Exploration of DevOps adoption in mobile software development. *Kybernetes*. https://doi.org/10.1108/K-04-2024-0989

Laso, S., Berrocal, J., Fernández, P., Ruiz-Cortés, A., & Murillo, J. M. (2022). Perses: A framework for the continuous evaluation of the QoS of distributed mobile applications. *Pervasive and Mobile Computing*, 101627. https://doi.org/10.1016/j.pmcj.2022.101627

Lombardi, F., & Fanton, A. (2023). From DevOps to DevSecOps is not enough: CyberDevOps: An extreme shifting-left architecture to bring cybersecurity within software security lifecycle pipeline. *Software Quality Journal*. https://doi.org/10.1007/s11219-023-09619-3

Madhushanie, N., Vidanagamachchi, S., & Arachchilage, N. (2025). Real-time privacy vulnerability detection techniques in software development: A systematic literature review. *Computers & Security*, *159*, 104659. https://doi.org/10.1016/j.cose.2025.104659

PMI. (2021). *The standard for project management and a guide to the project management body of knowledge (PMBOK® guide)* (Seventh edition. ed.). Project Management Institute, Inc. https://research.ebsco.com/linkprocessor/plink?id=376ee077-bd9d-3fbe-8cfa-e241e56f6129

Pontillo, V., Palomba, F., & Ferrucci, F. (2024). Test code flakiness in mobile apps: The developer's perspective. *Information and Software Technology*, *168*. https://doi.org/10.1016/j.infsof.2023.107394

Statista. (2024). Number of mobile app downloads worldwide from 2016 to 2024 (in billions). https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads

Wettinger, J., Breitenbücher, U., Kopp, O., & Leymann, F. (2016). Streamlining DevOps automation for Cloud applications using TOSCA as standardized metamodel. *Future Generation Computer Systems*, *56*, 317-332. https://doi.org/10.1016/j.future.2015.07.017

Wiedemann, A., Wiesche, M., Gewald, H., & Krcmar, H. (2023). Integrating development and operations teams: A control approach for DevOps. *Information and Organization*, *33*(3). https://doi.org/10.1016/j.infoandorg.2023.100474

Zeng, Y., Chen, J., Shang, W., & Chen, T. H. P. (2019). Studying the characteristics of logging practices in mobile apps: A case study on F-Droid. *Empirical Software Engineering*, *24*(6), 3394-3434. https://doi.org/10.1007/s10664-019-09687-9