International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

Intelligent Enterprise Transformation: A Closed-Loop Reference Architecture for Automating SAP Application Lifecycle Management Through Scalable Change and Deployment Governance

Karthik Paramasivam[1]
[1]*Anna University, India*
Email id: *Karthik.sapalm@gmail.com*[1]

Abstract

In enterprise resource planning (ERP) environments, the business process logic, configuration, custom development, master data, authorisation design and financial controls are all interdependent and are tightly integrated in one operating environment. The route to production of a change is thus at the same time a delivery mechanism and a preventive control. In this research work, a platform-independent reference architecture for scalable governance of SAP changes and deployments is designed and partially evaluated, which can be realised by SAP Solution Manager Change Request Management (ChaRM) and SAP Cloud ALM Change and Deployment Management (CDM). The study adopts design science research approach. A knowledge base was synthesised from a variety of topics from continuous delivery, DevOps, ERP lifecycle risk, IT governance, access control, software configuration management and process mining; automated checks, ALM change and deployment orchestration, governance and control, and measurement and feedback. In production, one control objective was evaluated, namely the parallel-track reconciliation, for greenfield SAP S/4HANA programmes running N and N+1 maintenance landscapes. The remaining 5 per cent are due to the following 3 halting conditions: object conflicts, object locks and SAP Note content owned by the vendor. Automation is not a discharge of a control objective, but it is a partition of it: mechanical reconciliation is a repeatable logged operation, semantic reconciliation is isolated and passed to human judgement. The marginal effort per change decreases, while the correlated specification risk increases due to the fact that one bad rule can be passed on to many people. The exception ownership, process conformance, outcome monitoring and periodic policy review are therefore architectural aspects of the production change-control loop and not post hoc reporting activities.

Keywords: *SAP Solution Manager; Change Request Management; SAP Cloud ALM; Change and Deployment Management*

1. Introduction

Enterprise SAP systems are not stand-alone systems. Business process design, master data, configuration, custom code, integration logic, authorisation and control activities all exist in the same operating landscape and a seemingly local change can impact financial posting, procurement, tax determination, access control, batch processing, or downstream reporting. The research on ERP has always been related to lifecycle outcomes to the extent that organisational fit, process design, governance capability and delivery competence are all factors that influence the outcome, not just technical correctness [1] and the shift of ERP estates to cloud and hybrid deployments has introduced infrastructural volatility to an already coupled environment [2]. Thus, the production-change path should be planned as a controlled delivery path, as a preventive control, and as a long-lasting audit trail. Technical movement with SAP transport and deployment technologies is reliable, but this does not mean that the change was allowed, verified and/or suitable for the business process it impacts. This is also true for deployment using a cloud transport service or an external delivery pipeline. Technical logs provide information on what moved and when. Governance also needs to address the question of why the change was needed, who took ownership of the remaining risk, what evidence was considered, and if incompatible duties were split, and if the desired operating result

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

was met. When these answers are stored in spreadsheets or in e-mail and meetings, the level of coordination effort and evidence gaps increase with the frequency of releases and the complexity of the landscape. The continuous integration and continuous delivery design principle provides a valuable guideline: any change to production should follow a reproducible process with evidence assessed prior to promotion. These practices are linked to Systematic Reviews to earlier detection of defects, shorter feedback cycles and more repeatable release, and the lack of test automation maturity, legacy architecture, compliance obligations and organisational structure are all recurring issues [3]. Empirical characterisations of continuous integration practice also reveal that pipeline adoption is often accompanied by anti-patterns that undermine the control value, like frequent override of failing checks [4]. The principle is valid for SAP, but the implementation cannot be migrated from stateless cloud-native apps, as SAP landscapes are long-lived, configuration intensive, reliant on persistent quality systems and have limited rollback capability. The research on application lifecycle management places the challenge in integration of tools and artefacts, traceability and responsibilities instead of a single product capability [5]. There are currently two different realisations of governed change offered by SAP. Requests for change, change documents, change cycles, task lists, partner functions, status actions and transport controls are all used by ChaRM in SAP Solution Manager. SAP Cloud ALM CDM is based on the concepts of Features, Releases, Deployment Plans, system groups and deployment scheduling. SAP claims that Cloud ALM is the strategic next-generation ALM platform and that CDM is not a 1:1 cloud version of ChaRM, but rather introduces new concepts, workflows and integration patterns [D1]–[D3]. This study, therefore, does not assume product recency as a proxy for governance maturity and both platforms are included. A good ChaRM process can provide better control than a poor Cloud ALM process and vice versa. Taking governance capability out of the platform generation equation is in line with findings that differences in IT organisation performance are not solely due to the tooling used, but also to governance mechanisms [6]. One of the control objectives under consideration is reconciliation of parallel maintenance tracks, which deserves special consideration. If an organisation has a production-support track in addition to a project or implementation track, changes need to be propagated between the two on an ongoing basis. This requirement, often fulfilled by Retrofit, is not found in the greenfield continuous-delivery literature where it is assumed that artefacts can be deployed independently and there is only one path to production. Two complications in the SAP case, compared to the general case, prevent straightforward transfer of the findings of software configuration management research, which treats the general case as a version-merge problem, and has characterised the frequency, structure and resolution cost of merge conflicts in open-source settings [7]. First, it's a unit of change that contains ordering semantics, not a source branch that can be merged. Second, there is lots of configuration and repository metadata, and not so much text. Reconciliation is also one of the few SAP governance activities that has a manual cost large enough to be attributed to it and thus is a tractable first target for empirical evaluation.

1.1.Research questions

The study aims at answering one main and two sub research questions.

- RQ1. How can a common closed-loop SAP application lifecycle management architecture deliver scalable change and deployment governance across the on-premises, cloud and hybrid landscape, via both ChaRM and Cloud ALM CDM?
- RQ2. Which of the control objectives are the same in both platforms, and what platform-specific mechanisms implement each of them, and what is the difference in the control depth in each case?
- RQ3. Once one such control objective is automated at production scale, what percentage of the control obligation is discharged mechanically, and what is the non-mechanically discharged part, and

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

what is the governance obligation of the non-mechanically discharged part?

1.2.Contributions

The study has five contributions. It introduces (i) a five-layer closed-loop reference model for SAP change and deployment governance; (ii) a mapping of invariant control objectives and SAP change and deployment governance mechanisms (particularly CDM in the Cloud ALM); (iii) a comparison of three operating models describing the level of governance maturity without ranking vendor products; (iv) a path for the phased adoption of SAP change and deployment governance with a stable set of measures; and (v) an initial production evaluation of automated parallel-track reconciliation, including a structured interpretation of the non-automated residual in control terms. The remainder of the paper is organised as follows. The literature review and the statement of the research gap can be found in the second section. The research design is described in Section 3. The reference architecture is discussed in Section 4. The platform realisation and co-existence is covered in Section 5 and the control design throughout the lifecycle is covered in Section 6. The governance mapping is presented in Section 7, operating models and residual risk in Section 8, and the adoption path and measurement framework in Section 9. The empirical evaluation is presented in Section 10. Implications and conclusions discussed in sections 11 & 12.

2. Related Work and Research Gap

This unit covers the risks of change throughout the ERP implementation lifecycle and after the software is already in place. The ERP risk is not over once you go live! Requirements, development, validation, release and operation are intertwined along the lifecycle, and the outcome factors are mostly organisational [1]. The introduction of cloud and hybrid topologies in ERP workloads provides another level of risk due to the difference between the desired infrastructure and real-time changing configuration of the deployment, known as infrastructural drift [2]. For SAP, lifecycle integration needs to maintain the business intent relationship with technical objects, tests, approvals and operational results. Import return codes aren't enough to prove quality because a transport may be able to import and provide an incorrect process, configuration or authorisation outcome. Introducing the concept of continuous delivery and DevOps. An artefact is delivered when certain checks are met. Evidence for systematic review shows that continuous integration is correlated with early detection and repeatability of defects, and that there has been a consistent challenge with environment availability, legacy architecture, data management and organisational adoption [3]. Empirical studies of practice have confirmed that these benefits are often deliberately ignored by practitioners, either by routinely not following the failing checks, or by failing to take certain precautions in the builds [4], and that the benefits that practitioners report getting are strongly dependent on the context and the maturity of supporting practices before tool adoption [8]. Multivocal review of DevOps capabilities also reveals that the results of DevOps depend on complementary, technical and organizational capability, not just on automation [9]; design-science research of DevOps metrics demonstrates that capability and measurement need to be specified together for improvement claims to be measurable [10]. One such dimension of DevOps adoption which is found to be different and often neglected is security capability [11]. In SAP, the artefact can be a Change and Transport System (CTS) transport, a cloud transport, a configuration collection or a group of related deployment objects. While the principle of the gate still applies, the checks need to take account of persistent systems, transport dependencies, the configuration content, cross-system references, and regression of business processes. Application lifecycle management and IT service management is one of the two part types. Application lifecycle management in industrial evaluations is not a product feature, but rather integration between tools, artefacts, traceability, and responsibilities [5]. Similar results can be found for change and release processes in IT service management research reports, but the implementation of the process has not gained the same traction as the documented process, and it is not the process itself that provides the greatest

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

benefits but rather the process with some integration of process, evidence and tooling. But there's another warning for enterprise architecture research; architectural artefacts can only provide value if they are used in decision making processes, and artefacts created for documentation purposes - without being used in decision making - have a weak influence on the outcome [13]. This is the same logic used in governance configuration, for an ALM platform, compliance is recorded as a status check without any control. This course covers the principles of good IT governance, segregation of duties and access control. IT governance is one that assigns accountability and decision rights while change management determines who can request, build, test, approve, schedule and deploy. Research on contemporary governance indicates that the governance configuration is not only a documentation issue but also a designed artefact that has an impact on firm performance through its mechanism design, including the formal/informal mechanism design matrix [6]. In the case of assignments that are only specified in procedure documents, the effectiveness of control relies on manual memory and periodic review, while in the case of assignments specified in partner functions, project roles, authorisations, workflow conditions and status rules, the effectiveness of control is to be assessed consistently and at population level. Mutually exclusive role assignment is the formal foundation of access control research and controlled, time-bounded emergency access has been documented in recent surveys [14] of how role based models have been extended to heterogeneous and distributed estates. In parallel, audit research indicates that some of the control activities that were previously manually performed can be carried out by software agents that can record the results of the control, but that the design, ownership and review of the automated rule is the key remaining control concern [15].

2.1.Quality evidence, static analysis and change-level risk

The depth of the verification should be commensurate with the risk. Literally, it is an accepted fact that change-level properties like size, dispersion across files, and author history are evidence-based and predictive of defect introduction, thus justifying the use of different depths of gates for verification, not a single gate [16]. Studies of adoption of static analysis indicate that when results are not actionable, not attributed to the current change, or not part of the developer's workflow, then the findings are ignored [17] while large scale usability testing of static analysis tools reveals that the presentation, explanation and configurability of the tool determines whether findings are acted upon [18]. These results validate a phased gating approach - rather than applying all rules to an inherited finding baseline - that targets changed objects and high severity rules. The introduction of parallel development tracks and reconciliation. The comparative study of software configuration management research describes concurrent development as a merge problem and measures the occurrence of these conflicts and the cost of solving them in open-source software projects [7] then follows several studies on predictive factors that enable the prediction of the likelihood of conflict before attempting to merge [19]. The case of parallel-maintenance is structurally different from the SAP case. The reconciliation unit is a transport request with ordering semantics, the content does not contain text but configuration and repository metadata as well, and both tracks are landscapes, with their own quality and pre-production tiers. As a result, the conflict and resolution patterns reported for source-code merging don't necessarily map over and the SAP case has not been empirically described in the peer-review literature. The final stage involves measuring, process mining and analytical risk, which will be addressed in the following section.

Using process mining it is possible to re-create actual lifecycle paths from event data, and uncover rework loops, approval bottlenecks and non-conforming paths. It is reported that the maturing method support it provides in business management is complemented by the continued data-quality requirements [20] and a Delphi study of organisational adoption reveals that data available, event-log quality and embedding it in the organisation are the most prevalent constraints, not algorithms [21]. Field research also indicates that

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

value realization requires embedding the process mining into an existing management practice, instead of as an independent analysis [22]. If the use of analytic modelling is to guide routing, then the deployment literature is relevant. Data dependency failures are not uncommon, as captured in survey data on how machine learning is deployed [23] and in the definition of the lifecycle and infrastructure needed to maintain accountability of models in production [24] [25]. In the context of change governance, it means that deterministic controls must continue to be authoritative in case of high-risk changes and that any model that affects a control decision must be explainable, calibrated, monitored for drift, and bound by policy.

2.2.Research gap

The gap is summarised in Table 1. None of the literature streams deals with the composite problem, a control model that is not necessarily modifiable whilst in operation, but is realised by two distinct vendor platforms, with landscapes that carry a continuous reconciliation requirement between parallel maintenance tracks. Table 1 Gap analysis across contributing literature streams.

Table 1 Literature Contributions and Limitations in SAP Governance

Literature Stream	Contribution	Limitation
ERP lifecycle [1], [2]	Risk management	Control gaps
CI/CD [3], [4], [8]	Automation	Deployment complexity
DevOps [9]–[11]	Performance	ERP specificity
ALM/ITSM [5], [12], [13]	Traceability	Platform integration
IT governance [6], [14], [15]	Compliance	Transport control
Risk analysis [16]–[18]	Verification	Configuration gaps
Configuration management [7], [19]	Conflict detection	Transport sequencing
Process mining [20]–[25]	Conformance	Data quality

3. Research Design and Method

3.1.Design science framing

The study is based on design science research [26] which involves designing an artefact that can serve a class of emerging problems in the field and testing the artefact against criteria developed from the problems in this class. The artefact is the five-layer reference architecture that is provided in Section 4, along with its realisation mapping in Sections 5 to 7. The problem class is scalable change and deployment governance for SAP estates across on-premises, cloud and hybrid deployments and subject to audit obligation. Three design cycles were completed. The field requirements were set up by examining the field needs through the relevance cycle, which involved practitioner observation of the failure modes in the SAP change governance. The rigour cycle was derived from the literature summarized in Section 2 and from the technical documentation from the current vendors (D1–D6). The design cycle built and developed the architecture based on the following, for each transition in the lifecycle: What evidence should exist prior to the transition to the next phase, and what ALM or deployment process can evaluate and record that evidence? In this section, the students will create the architecture. Requirements were

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

retrieved from the Knowledge Base and merged together to group requirements with only minor differences in language. The consolidated set was then broken down into control objectives (platform-independent) and realisation mechanisms (platform-dependent). This separation is an essential design decision of artefact and what makes it possible to have two platforms instantiating the same governance model with different object models. Product-specific concepts were not validated against practitioner commentary, but against the documentation of the current SAP Solution Manager and SAP Cloud ALM [D1]–[D6]. Capability claims are referred to the SAP Solution Manager or SAP Cloud ALM documentation at the point of use. Architectural experience with SAP change processes and transport landscape, quality gates, parallel development tracks and audit controls provides further guidance on the design. This experience is not referred to as evidence; claims based on this experience are not reported as results; it is more of a framing of problems and mechanisms of candidates.

3.2.Evaluation design

A comprehensive assessment of the architecture would necessitate instrumented measurement of each of the control objects listed in Table 3 at a variety of organizations, but such a study is not possible within the scope and scale of this study. The evaluation is therefore a naturalistic, ex post evaluation of one control objective, which was selected because it had two characteristics: it was the one control objective that was not very expensive and could be measured without a wide instrumentation programme and the one control objective that was least covered by the continuous-delivery literature.

Three hypotheses were tested, based on the architecture.

- P1. A significant portion of the parallel-track reconciliation obligation can be automatically generated and completed without any human intervention.
- P2. The non-automated portion is not a selection of inevitable automation failures, but a finite number of logged failures in which there is no guaranteed solution.
- P3. It's the same with the objective – it now takes the place of coordination effort while the governance responsibility for the rest remains in the hands of ownership – if an organisation automates without assigning ownership it takes the place of a manual baseline but it's in a worse control position.

The only ones that can be directly tested against the platform log are P1 and P2. In Section 10.6, P3 is identified as a design claim that has been evaluated in an argumentative manner when compared to the outcome distribution that was observed.

3.3.Data sources and measurement procedure

Evaluated population: retrofit eligible transport requests resulting from greenfield SAP S/4HANA transport projects in a multiregional MIE with N and N+1 parallel maintenance. Outcome data were measured at platform export not manually or by status reports or recollection. A transport identifier and a terminal outcome are present for each log record, and this allows computation of the outcome distribution across the entire eligible population instead of a sample. The effort baseline is of lesser evidential quality and reported as such. It is reconstructed based on a documented staffing pattern and is therefore not based on instrumented time records; its derivation is published parameter by parameter in Table 5 so that readers may use other values as they replace the published ones. Throughout the document, details of organisation identification, system identification and breakdowns of business units are omitted.

3.4.Scope of claims

Production data are only used to assess the reconciliation objective. The other architecture is seen as a structured reference model that needs more case studies, longitudinal measures and platform assessments that are similar. The comparative analysis of the operating models in Section 8 is analytical. When the text describes expected but not actual behaviour, this is indicated at the point when the behaviour is described.

4. Closed-Loop Reference Architecture

The artefact is shown in figure 1. Policy is cascaded down through roles, exceptions, thresholds,

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

quality, and deployment windows, and risk classification. Approval, test results, transport and deployment logs, lifecycle and incident and operational outcomes are all evidence that rise. The loop is closed only when a descending policy is updated based on ascending evidence. A dashboard reporting activity, but not changing ineffective rules is monitoring, not closed-loop governance. As shown in Figure 1 Closed-loop reference architecture for scalable SAP change and deployment governance.

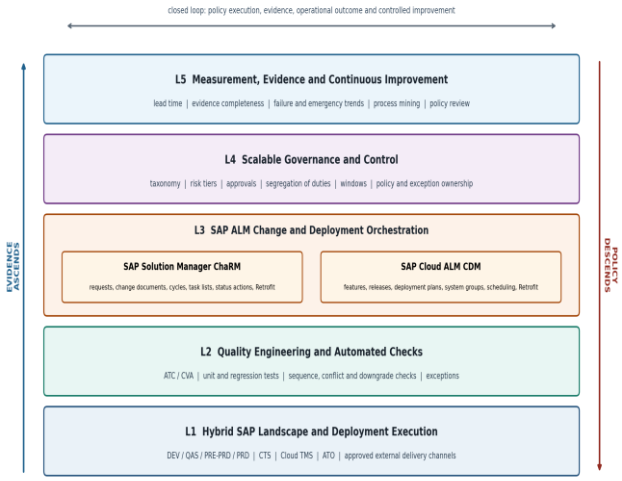


Figure 1 Closed-loop reference architecture for scalable SAP change and deployment governance

4.1.L1: Hybrid landscape and deployment execution

The first layer is the managed systems and moving software/configuration. It could consist of development, quality, pre-production and production systems, single or parallel transport tracks, CTS, Cloud Transport Management service, deployment paths to adapt to public-cloud scenarios, and approved external delivery channels. There is a number of different ways to do this, but all deployments of a production should be tied to an authoritative change object and a unique identifier that correlates business scope, transport or deployment artefacts, tests and incidents.

A fundamental design question at this layer is not whether a transport tool can move an object, but whether it can still move an object in an uncontrolled manner. In rare cases, direct imports, emergency credentials or manual cloud deployment and pipeline shortcuts may be necessary, but need to be limited, documented, and monitored. An optional control path is not a control when teams can easily and frequently circumvent the control [4] (as in CI practice where checks are failed and then ignored). If there are parallel maintenance tracks this layer should also document the relationship between parallel tracks. The architecture considers the second track not as another landscape, but as a managed dependency – each object released within a first track introduces a potential obligation in a second track, and the mechanism that "discharges" the obligation is not part of the first track, but part of the second track.

4.2.The class consists of 30 students

The second layer specifies the conditions that must be met by the quality evidence in order to be considered valid. Common checks consist of static analysis, security scanning, unit tests, regression tests, object dependency checks, transport sequence validation, downgrade analysis and cross-reference analysis. Results should be machine-readable, and should be added to the change context and not as optional reports. The check ordering is aimed at the continuous delivery goal of obtaining cheap feedback early and costly end-to-end feedback only after potentially passing through basic controls [3]. As early as possible, ideally at or before transport release, fast and inexpensive checks like syntax validation, static analysis, object-level dependency checks and unit tests should be run. Other functional or regression tests will be performed later in the quality "landscape", and should not be performed until production readiness is confirmed. If the adoption of an inherited finding baseline is not practical, then a finding baseline should first concentrate on changed objects, critical business processes and high severity findings. This staging is backed up by evidence of the lack of action on findings from static analysis when they either are not actionable or they cannot be linked to the change being considered [17], [18]. Each gate must have an owner, a severity model, a remediation pathway

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

and an exception pathway – an exception should specify the reason, risk owner, compensating control and expiry date. Defects that pass through the gate and false blocks should be used to measure the effectiveness, not the number of times the gate is executed; if a gate is executed every time it is expected to be executed, and it produces noise or is frequently thrown, it will not protect.

4.3.Orchestrate the change and deployment of SAP ALM

When we talk about orchestration, the layer where business intent, technical work and deployment evidence collide. In ChaRM, the main objects are requests for change, change documents, change cycles, task lists, partner functions and status dependent action, the process is able to distinguish Normal, Urgent, Administrative and General Change, and can synchronize transport execution and status and approval rules [D1]. The main object in Cloud ALM CDM is the Feature and Requirements, which are supported by the system groups and deployment scheduling, and a Feature marked as a Correction or Fix which has a streamlined path for time-sensitive change [D4] or [D5]. The platforms are not a two layer mandatory approval on top of each other on the same change, but rather alternative realisations of one enterprise control model. Each landscape scope must have exactly one landscape scope to provide authoritative status, approval and deployment evidence. An organisation can use ChaRM for existing ABAP landscapes and Cloud ALM for new cloud or hybrid scope, as long as the scope is clearly defined with regards to ownership and reporting. Having both tools show the same status and approval creates confusion instead of controlling.

4.4.L4: Scalable governance and control

The fourth layer is governance in the form of configuration that can be executed and reviewed. A policy table defines the different types of changes, their business criticality, financial control impact, technical scope, integration impact, emergency status and recent failure history to the necessary approvers, tests, quality gates and deployment conditions. The policy artefact should be versioned, tested, approved and auditable as there can be a large population of changes affected by a defect in the policy. Research on governance points to considering mechanism design as a planned artefact and its performance consequences as measurable [6] while enterprise architecture research shows that artefacts affect the outcomes when they are actually used in making actual decisions, and not just documented.

Requester, developer, tester, approver and production deployer should be segregated wherever possible, based on risk, according to organisational policy, and assessed prior to promotion. Emergency access should only be for a limited period of time and should be reviewed post use. The same should go for deployment windows, high volume import rules, and freeze windows.

4.5.L5: Measurement, evidence and continuous improvement

The fifth layer is about using lifecycle evidence to enhance the process. Process mining can reconstruct actual status paths, uncover repeated reversion to development, detect bottlenecks in approving the path and detect non-conforming paths [20] as long as the event-log quality requirements that are identified by adoption studies are met [21], [22] which is the most common practical constraint. Results can be related to change type, component, test coverage, exception use and deployment time. To begin with, the initial intent should not be to "relax" controls automatically, but rather to review the policies and make improvements in the process.

Deterministic controls will be authoritative for high-risk changes, and may also help with the routing at a later time using analytic risk scoring. All models affecting the decision of approval of testing requirements should be explainable to the accountable decision-maker [25], calibrated, monitored for drift, and bounded by policy, and operated through a lifecycle that provides for retraining, monitoring and rollback [23, 24]. In actual use, the value of a reliable identifier and complete event history outweighs that of a sophisticated prediction model built with incomplete event history. As shown in Table 2 Comparison of SAP Solution Manager ChaRM and SAP Cloud ALM CDM.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

Table 2 Comparison of SAP Solution Manager ChaRM and SAP Cloud ALM CDM

Governance Concept	SAP ChaRM	SAP Cloud ALM	Architecture Consideration
Central object	Change document	Feature	Traceability
Release grouping	Change cycle	Deployment plan	Separation
Urgent change	Urgent Change	Correction/Fix	Authorisation
Workflow	Customisable	Standardised	Control mapping
Deployment	CTS/CTS+/TMS	CTS/TMS/ATO	Compatibility
Synchronisation	Retrofit	Retrofit	Conflict management
Conflict prevention	CSOL	Downgrade checks	Risk mitigation
ITSM integration	Native	API-based	Linkage
Operations	Customer-managed	SAP-managed	Monitoring

Table 2 illustrates the common architecture in terms of the realisation in ChaRM and Cloud ALM CDM. The mapping is not intended to necessarily be identical but to describe the similarity in how that feature operates and the consideration that the architect takes into account. It is clear in SAP that the changeover from ChaRM to CDM has new concepts and process design, and not configuration transfer [D2]. There are three possible ways to achieve co-existence. Cloud ALM is the authority for new cloud scopes and ChaRM is for mature on-premises or private cloud scopes, in a platform-per-landscape model. With a staged transition model, new work starts in Cloud ALM after the old work is done in ChaRM. In federated model, both platforms are used; common definitions for risk, evidence completeness, lead time, failure and exception rates are applied to enterprise reporting. The governance layer needs to specify the ownership of the system-group and change-type levels for all 3 patterns.

Cloud ALM should not be characterised as ChaRM in the cloud. The standardised Feature status model, the external workflow approach, the deployment integrations and the lack of direct CSOL equivalent result in materially different design choices [D2]. On the other hand, ChaRM shouldn't be written off as legacy workflow where it still serves as the one and only transport-control solution for all those complex or regulated scopes. Placing the question in each direction is whether the platform generates complete, effective and reviewable control evidence.

5. Managing the Lifecycle of a Control Design

5.1.Risk-based classification and approval

Approval should not be levelled but differentiated according to risk. Practical risk model has a set of just a few stable dimensions: business criticality, financial-control impact, object scope and/or configuration scope, reach of integration, change type, and deployment urgency and failure frequency in the last few years. The empirical study

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

of just-in-time defect prediction allows to use change properties like change size, dispersion and author history as risk signals [16] and gives a clear indication to differentiate the levels of verification depth instead of verifying all changes with a single gate. The result shall be a risk tier of approvers, test depth, evidence and deployment restrictions. A low risk repeatable change can be made using pre approved conditions; a normal change may be approved by the affected service and scope of control; a high risk or financially significant change should be reviewed by more people. Logic for classification needs to be owned by the name. Criticality and impact on control should be defined by business process owners, technical and access risk by platform and security owners and deployment constraints by release management. Rules without domain participation will be internally consistent but operationally incorrect, e.g. by central ownership. The policy needs to be tested against the historical and representative change scenarios prior to activation.

5.2.Test, Static-analysis and Quality evidence

Static analysis and unit tests should run prior to a development release, if possible. Blocking findings should result in an obvious remediation condition and block the appropriate status transition. The change and the business-process scope should be related to functional and regression evidence, not just a change schedule. Retained evidence should include the version of the tool, test set, time of execution, result of the test and approval of exception if any, so that a reviewer can replicate the decision. In the majority of large SAP estates, findings are left due to the nature of the finding and incomplete automated regression coverage, making an all-or-nothing gate impracticable. A defensible rollout starts with changed objects, high severity rules and critical processes and expands the rollout as the baseline gets better. Exceptions must be temporary and observable. A repeated exception should signal a change in the rule, the test asset or the design; not a normalisation of the bypass, as evidenced by successfully unacting findings which are systematically ignored [17, 18].

5.3.Sequence, conflict, downgrade and reconciliation controls

Errors in transport sequences may cause overwriting of newer versions of objects or reordering of dependent changes. For pre-deployment analysis, versions of objects, the sequence to be followed, cross-references, and the state of the target-system should be compared. The conflict must be resolved or the exception for it be documented and approved before allowing deployment. It is not the number of scans done, but how many incidents have been reduced, emergency manual intervention and failed deployment recovery, because of the effectiveness of the measure. Divergence risk occurs with parallel maintenance and project tracks. You should regard reconciliation as a governed process with an owner, status, age and volume, and separate automatic, semi-automatic and manual handling where supported by the platform as evidence and review effort varies between these 3 types. The functions of Retrofit and conflict prevention are provided by ChaRM with mature functions available in appropriate scenarios [D1] and Focused Build Standalone Extensions include bulk Retrofit Automation with per-transport outcome recording [D6]. For supported scenarios Cloud ALM offers automated or manual scenario categorisation as well as downgrade and cross-reference checks, but does not offer a direct CSOL equivalent [D2]. This difference has to be reflected in the migration risk assessment as such, not as part of a generic platform transition plan. In automated reconciliation, the control is governed by a design property that dictates the control's stopping condition: If there is not a deterministic answer to the condition, the control will not continue to be automated. The architectural implication is that the non-automated population needs an owner named, a service expectation and an age threshold. In Section 10, this population is quantified and the effect in a production context is studied.

5.4. Urgent and emergency changes

An urgent change is a compressed control path, not an ungoverned control path. It should include minimum authorisation, complete linkage between change and deployment artefacts, limited production rights and a time-limited review of the

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

past. The record should record the reasons why it was not possible to take the normal path and what evidence was deferred. If the emergency activity is repeated in one component, that may be a sign of unstable design, limited test coverage, incomplete release planning or technical debt is not addressed, so it should be monitored as a trend rather than addressed on a per Case basis.

Discipline must be maintained for the freeze and focus periods as well. While technically allowed, production deployment should be denied or escalated if an exception has not been approved by the governance layer. This is why policy should not be stored in calendars and e-mail: the rule must be made available when the decision to deploy and/or activate it is made.

5.5.The principles of segregation of duties

There should be a separation of conflicting activities in role design; and workflow assignment and technical authorisation should be taken into account when designing the architecture. The record of the change could have different requester/approver names when the same person has production import authorisation. It is therefore important to evaluate the functions of partners or project roles together with access to the back-end and deployment tools, as is done in access control research [14] with the principles of mutual exclusion.

The evidence of an audit should be population and reproducible. A ChangeReviewer should be able to follow up a business request, classification, approvals, tests, exceptions, transport or deployment artefacts, status history and outcome for each production change without needing to seek personal explanations. Completeness and effectiveness are two different things: a complete record, even those that are automatically approved, but irrelevant for the testing is traceable but not controlled. If the control execution is automated, audit research shows that the design, ownership and periodic review of the automated rule is the major remaining issue [15].

6. Mapping of control objectives (Governance mapping)

Table 3 couples invariant control objectives with the mechanisms available in ChaRM and Cloud ALM CDM and the metric used to ensure each objective is carried out. It is not intended to enforce a standard configuration, but to keep the same control outcome and evidence definition with the realisation of the platform changing. It is the verification metric which is the active part - it allows the two realisations to be compared on control effectiveness, not features. As shown in Table 3 Control objectives, platform mechanisms and verification metrics.

Table 3 Control objectives, platform mechanisms and verification metrics

Control Objective	Common Rule	ChaRM	Cloud ALM CDM	Verification Metric
Authorisation	Risk-based	Approvals	Feature workflow	Compliance
Segregation	Role separation	Partner functions	Project roles	Conflicts
Quality	Validation	Status gates	Test evidence	Pass rate
Sequencing	Dependency checks	CSOL	Downgrade checks	Incidents
Urgent changes	Fast-track	Urgent Change	Correction/Fix	Review rate
Synchronisation	Reconciliation	Retrofit	Retrofit	Backlog
Auditability	Traceability	Change logs	Deployment history	Completeness
Performance	Monitoring	Lifecycle metrics	Deployment metrics	Failure rate

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

7. Operating Models and Residual Risk

Three operating models describe governance maturity without making either SAP product a mandatory stage above the other. Model A relies on transport tools and manual coordination. Model B uses ChaRM or Cloud ALM CDM as the authoritative lifecycle and evidence platform while many decisions remain manual. Model C integrates policy-driven routing, automated quality checks, continuous role evaluation and population-level evidence with the selected ALM platform. Except where Section 10 reports observed data, the comparison in Table 4 is analytical and describes expected capability rather than measured production outcomes. As shown in Table 4. Comparison of SAP change and deployment operating models

Table 4 Comparison of SAP change and deployment operating models

Evaluation Dimension	Model A: Manual	Model B: ALM	Model C: Policy-driven
Authorisation	Manual	Recorded	Risk-based
Sequencing	Manual	Structured	Automated
Defect detection	Reactive	Testing	Quality-gated
Duty segregation	Conventional	Role-based	Rule-based
Audit evidence	Fragmented	Traceable	Continuous
Parallel-track control	Manual	Tracked	Automated
Effort	High	Moderate	Optimised
Residual risk	Noncompliance	Control gaps	Policy errors

There are three governance maturity models that do not require that the specific SAP product must come before it in the succession. The Model A is based on the use of transportation tools and manual coordination. Many decisions are still manual and Model B is an authoritative CDM for the lifecycle and evidence. Model C seamlessly combines policy-driven routing, automated quality checks, continuous role evaluation and population-level evidence with the chosen ALM platform. The comparison for Table 4 is analytical in nature and reflects anticipated capability, as opposed to actual production results, except where data is reported in Section 10. The greatest advantage to Model C is the abandonment of control work. The expertise in governance transitions from coordinating elements across change to designing and managing taxonomy, roles, gates, evidence and exceptions. This can make effort less sensitive to change volume after investments in platform integration, test coverage, policy ownership. According to DevOps capability research, automation can only produce results with technical and organisational capability [9],[10]. sAlso, the residual risk profile alters, rather than simply in degree. Many errors of omission and execution occur in the manual operation. Automated governance minimizes those errors, and at the same time brings into existence correlated specification risk, where one bad rule impacts numerous changes at speed. Monitoring should therefore detect sudden changes in pass rates, unusual use of exception rules, usage patterns of approvals, failures to deploy or operational incidents after the policy change and view these

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

anomalies as candidate policy defects, not execution noise. In the case where analytic models play a role in routing decisions, the requirement of drift and explainability in Section 4.5 becomes even more pertinent, as such a model defect is correlated and opaque [23] – [25].

8. Open House for discussion of the proposal

The first step in adopting governance is to start with scope and not tool migration. First, determine the authoritative ALM platform in each system group and change type – existing landscapes can proceed with ChaRM, new cloud or hybrid landscapes will follow Cloud ALM CDM. The second step is to establish a common change taxonomy, risk levels, approval roles, evidence and exception process. Automation should be used for lifecycle transitions only when these building blocks are established. A practical sequence is used in ChaRM for the main purpose of defining the governed production path; it cleans up the change taxonomy/role model, links static analysis and testing with transport release, enhances conflict and reconciliation controls and automates evidence extraction. The sequence in the Cloud ALM – projects, system groups, Deployment Plans and Releases – Feature ownership and tags – Integrate deployment tools – Connect test and workflow evidence – Schedule and deploy based on conditions – Policy driven deployment conditions When moving to a new release of ChaRM, open questions should either be finished or clearly marked as such, and not duplicated silently on both releases [D2]. The measurement set should be brief and stable, and should contain the following metrics: change failure rate, approval-to-production lead time, active versus waiting time, proportion of production deployments with complete evidence, gate failure rate, false-block rate, exception rate, emergency-change frequency, retrospective-review completion, automatic reconciliation rate and reconciliation backlog age. Definitions should be consistent across platforms and should be posted with results, since the use of inconsistent definitions is one of the main challenges to comparing results across studies in this domain [10]. For instance, production incidents, rollback and corrective deployment or material business-process disruption should be consistently all parts of a change failure, as per the agreed definition by the Organisation. When the quality of the events is reliable, the process mining can be useful to find out how much lead time is used, if the lifecycle defined is followed in practice, and where lead time is lost due to bypass or rework [20]-[22]. Predictive risk scoring is only recommended once the deterministic controls and historical evidence are stable. The first goal is trustworthy governance data, not sophisticated analyses.

9. Empirical Evaluation: Automated Parallel-Track Reconciliation

9.1.Setting and population

The assessment is based on greenfield implementation programmes for the SAP S/4HANA in a multinational enterprise environment. The programmes were of 18-24 months' duration from the design phase to the production go-live phase. During that time, the organisation maintained two tracks of development (production and implementation) each with four levels of development, quality, pre-production and production, and hence had eight systems under concurrent maintenance. Change orchestration was done via ChaRM in SAP Solution Manager and Reconciliation was done via Retrofit between tracks and Bulk Retrofit Automation was done via Focused Build Standalone Extensions [D1, D6]. During the observation period, there were about 2,000 transport requests evaluated that can be retrofitted. All of the organisation-identifying details, system identifiers and business-unit breakdowns are omitted. The three definitions are explicitly given because similar terms often are omitted from practitioner literature, making it impossible to replicate results and making it difficult to obtain accurate estimates of automation. Retrofit-eligible transport. A transport request that has at least one object specified on it that is also specified on the parallel track and therefore has a reconciliation obligation. Automatically synchronised. A transport that was automated and no manual work was done at the object level and was recorded as such in the exported transport log. Paused and given to manual handling. A transport that was recalled for a reason, and it was

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

determined that the transport was automation stop/start, and was considered retrofit. The denominator is the number of people who can be reached by a retrofit, not all transports in the landscape. Transports that do not have an object in the parallel track do not have a reconciliation obligation, as to include them would only raise the level of automation reported without describing further control. In pre-automation days, reconciliation between the two tracks was done as a separate coordination function. A project management role was established to coordinate changes between project teams, functional consultants, developers and cross functional stakeholders. It was not operated per release, as divergence would accrue day-by-day, and fixing it would be extremely costly when it was uncovered late. The experience of the organisation was that it needed at least six months to synchronise two tracks within a project cycle. This staffing pattern is used to reconstruct the baseline, instead of using an instrumented time record. It is thus stated as an "estimate" that is documented and, parameter by parameter in Table 5, is derived on the basis of which readers can change the values and recalculate the estimate.

Table 5 Derivation of the manual reconciliation effort baseline

Parameter	Value	Basis
Dedicated roles	1 PMO	Coordination
Project cycles	2	Consecutive
Cycle duration	6–8 months	Reported
Working days	260–350	Estimated
Daily effort	3–4 hours	Reported
Total effort	800–1,400 hours	Calculated
Person-days	100–175	8-hour day

9.2.Measurement procedure

An automated reconciliation was done in bulk by using the Retrofit Automation capability of Focused Build Standalone Extensions for Change Request Management that processes eligible transports, and the result for each [D6] is recorded. The outcome data were not based on manual counts or on the reconstructed outcomes, but were the outcome data that were exported from the Retrofit log; the outcome distribution was calculated for the entire population, not for a sample. On each halted transport the reason for halting is recorded allowing the residual population to be divided into its constituent parts and not simply reported as residual. The distribution of the outcome is reported in Table 6. For 95% of transport for which a retrofit was possible, we were able to conciliate transport without having to do any manual work at the object level. As shown in Table 6. Observed reconciliation outcomes across the evaluated population.

Table 6 Observed reconciliation outcomes across the evaluated population

Measure	Observed Value	Note
Landscape	8 systems (N/N+1)	Parallel tracks
Observation period	18–24 months	Programme duration
Eligible transports	~2,000	Total
Automated	~95%	Synchronised
Manual handling	~5% (100)	Deferred
Coordination effort	800–1,400 hours	Estimated

9.3.Interpretation of the residual population

The five per cent which was not reconciled automatically should not be interpreted as an automation failure rate and reporting it as such would be a false representation of the control. The automation stops on three reported conditions, none of which are a fault, but are platform

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

behaviour. An object conflict occurs when there is no unambiguous merge in two tracks for the same object that has been modified independently in each track. An object lock occurs if the object is being modified concurrently somewhere else, and reconciliation would race the concurrent change. An SAP Note object is created when the content is of external origin that is not associated with a version relationship owned by the customer but by the vendor. Each of these conditions is where there is a deterministic rule that cannot not determine the right answer, thus helping P2. In control mode (as opposed to fail silent), the mechanism refuses to operate and logs the reason it did not operate, not picking a plausible model and erasing work. It would not be safer, only more capable to handle 100 per cent of the population, as it would have to resolve true semantic conflicts without evidence. This is an important consideration about the way in which these mechanisms can be assessed: A reported automation rate only makes sense when used in conjunction with the taxonomy that led to it. The observed distribution of such means then to rephrase what automation has accomplished. It didn't take away the reconciliation control – it divided the population. Most of the obligation (mechanical reconciliation) was transferred to a repeatable and logged operation. Semantic reconciliation, which involves understanding the business intent, was separated, limited in scope and directed to some human. The governance impact is immediate and is a key outcome of this evaluation. The deferred population needs to be owned, have a resolution service expectation and an age threshold, just because that is the population that the automation has deemed to need judgement. Where the headline rate of automation looks good, an invisible ‘backlog' is worse than a visible ‘coordinated, staffed' version of the reconciliation problem – if it is not owned by someone. This argument is okay for P3, but P3 is not a hypothesis tested, as no organisation was found without residual ownership for comparison.

9.4.Threats to validity

This evaluation is preliminary and should be taken as a complement to the results. They are listed below by validity. Construct validity. Transport counts are not a perfect indication of how complex the change is, as they can include one object or many, and an automation rate based on the number of transports will not take the obligation into account. A possible selection effect arises because the eligibility for the retrofit is based on the platform's own logic, which cannot be evaluated by itself. This effort construct is about coordination time, not governance costs. Internal validity. There was no counterfactual to run; the manual and automated approaches were not operated simultaneously on similar populations, so the effort saved cannot be attributed solely to the automation, and there is a possibility of concurrent development of process discipline that is not ruled out. Because it is derived from a staffing pattern, and not instrumented time records, the effort baseline is reconstructed for independent recomputation as shown in Table 5. External validity. The setting is one organisational context and limited programmes - based on this, the automation rate is therefore an existence demonstration and not an industry benchmark. The conflict rate and automation rate are likely to be higher and lower for long-lived brownfield estates with accumulated modification history as opposed to cleaner greenfield implementations, which would increase the likelihood of clean populations of objects. The results are based on ChaRM with Focused Build Standalone Extensions and should not be interpreted as being applicable to Cloud ALM Retrofit which is based on another categorisation model and does not have a direct CSOL equivalent [D2]. Conclusion validity. The only measure of the evaluation is the outcome of reconciliation and effort to displaced coordination. It is not used to measure downstream defect rates or production incidents / audit findings, therefore it is valid to claim partitioning of efficiency and control but not improved operational quality.

10. Discussion

There are four implications to follow out of the artefact and its evaluation. First, ChaRM and Cloud ALM CDM shouldn't be considered the same product and should be included within one enterprise governance model. Second, the authoritative change and deployment history

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

should be clear for each scope as one can only have disagreement when two people have the same status and approve a change, but not the other. Third, approvals and checks, and access and deployment evidence need to be looked at together: a workflow that is not backed by approvals and checks is not a form of governance, and a deployment record without approvals and checks is not a form of governance. Fourth, as Section 10 illustrates, automating a control objective does not take away the governance requirement, but rather assignees it to the design, leaving what is not automated as unassigned.Fourth, as Section 10 shows, the automation of a control objective does not exempt the design from the governance responsibility, but rather assigns it to the design, where what is not automated becomes unassigned. The harder task is the organisation and not the technology. Risk levels, incompatible tasks, rapid verification of development, reusable evidence for test, ownership of sequence and exceptions for release and review of outcomes for business process owners must be established by governance teams.Governance teams must establish risk tiers, incompatible tasks, rapid verification of development, reusable evidence for test, ownership of sequence and exceptions for release and review of outcomes for business process owners. A cross functional design authority is necessary since no one technical team has the entire policy and this authority should take the same scrutiny as production configuration in reviewing policy changes. This condition is in line with capability research on automation, which has found that the outcomes of automation can only be achieved when paired with the corresponding organisational capability [9,10] and ERP evidence, which has shown that the outcomes of the lifecycle are mainly due to organisational factors [1] Since Cloud ALM is delivered continuously, the number of functions and steps within the workflows change more often than in a planned Solution Manager support-package cycle. Assumptions about their capabilities within a governance design should thus be revalidated during migration, but not as frequently as every cycle. Before Cloud ALM can provide equivalent business control [D2] for Regulated or Highly Customised ChaRM scenarios, the following may need to be taken into account: Gap analysis, external workflow, electronic-signature or ITSM integration. This constraint is also significant due to the vendor's stated direction of transition [D3] and therefore the practical and time limited nature of comparability of the control evidence across both platforms for organisations that currently rely on ChaRM.

The evaluation is for one control objective. The other goals in Table 3 (approval depth, quality gating, segregation of duties, evidence completeness) have not been assessed using production data, or measured results are not accounted for in the operating-model comparison in Table 4. This is because metrics used to define the change failure or lead time vary from organization to organization, making it difficult to compare across studies unless the definitions are provided with the results. The architecture itself has been assessed, and not evaluated with respect to other organisations' instantiations, the main threat to the generality of the design claim.

11. Future research

Four directions follow. First, the effectiveness of the gate coverage should be tested before the staged gating strategy of Section 4.2 should be adopted, as its effectiveness is currently based on adoption evidence and not outcome evidence. Second, comparisons of ChaRM and Cloud ALM evidence models in hybrid landscapes should be made by using the definitions provided in Section 10.2, to determine whether the invariance claim in Table 3 holds with respect to measurement. Third, the automation rate should be defined for "brownfield estates" where a history of modifications to the estate is likely to increase the number of conflicts in the system and hence lower the automation rate. Fourth, Approval and the resulting effect on lead time and control effectiveness should be quantified as a result of policy driven approval. It would be very valuable to have replication reporting the deferred population and its resolution time in addition to the headline automation rate, as the previous analysis indicates that the two indicators have different control meaning.

Conclusion

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

The production-change path in an SAP estate is at the same time a delivery system and a control system. This study has developed a closed loop reference architecture composed of five layers, with the following building blocks: firstly, hybrid transport and deployment mechanisms are at the bottom; secondly, automated verification is above execution; thirdly, SAP ALM orchestration is the core; fourthly, scalable governance above orchestration; and fifthly, measurement and feedback at the top. Platform differences are clearly identified and represented as alternative realisations of the same enterprise control objectives, rather than being abstracted away; ChaRM and Cloud ALM CDM represent two such realisations. Policy flows down through Risk tiers, Roles, Thresholds, Windows and Guardrails; Evidence flows up through Approvals, Tests, Transport and Deployment Logs, Exceptions, Metrics and Operational Outcomes. The central design position for one control objective is supported by the production evaluation. Automated reconciliation on about 2,000 retrofit-eligible transports across N and N+1 parallel landscapes reduced the discharge of the obligation to an estimated 800-1,400 hours of dedicated coordination effort and discharged approximately 95 per cent of the obligation without any manual intervention at the object level. The other five per cent wasn't a failure rate, it was a population that the mechanism rightly refused to make its decision: object conflicts, object locks and vendor-owned content. Automation thus divided the control instead of taking it away, so that the governance responsibility shifted from control to ownership of the residual. The resulting control model can help to alleviate manual effort and increase the likelihood of population-level assurance, but it has to deal with the correlated risk of having a flawed rule applied at scale; and it has to ensure that it is clear how automation will impact on what the control is responsible for. Therefore, controlled exceptions, process conformance, outcome monitoring and periodic policy review are integral parts of the architecture and not some extra reporting features.

References

[1]. [1] S. Katuu, “Enterprise resource planning: Past, present, and future,” New Review of Information Networking, vol. 25, no. 1, pp. 37–46, 2020.

[2]. [2] E. Hustad, V. U. Sørheller, E. H. Jørgensen, and P. Vassilakopoulou, “Moving enterprise resource planning (ERP) systems to the cloud: The challenge of infrastructural drift,” Information Technology & People, vol. 33, no. 3, pp. 910–930, 2020.

[3]. [3] E. Soares, G. Sizilio, J. Santos, D. A. da Costa, and U. Kulesza, “The effects of continuous integration on software development: A systematic literature review,” Empirical Software Engineering, vol. 27, Art. no. 78, 2022.

[4]. [4] F. Zampetti, C. Vassallo, S. Panichella, G. Canfora, H. Gall, and M. Di Penta, “An empirical characterization of bad practices in continuous integration,” Empirical Software Engineering, vol. 25, no. 2, pp. 1095–1135, 2020.

[5]. [5] Z. Akgün, M. Yilmaz, and P. M. Clarke, “Assessing application lifecycle management potentials from an industrial perspective,” in Systems, Software and Services Process Improvement (EuroSPI 2020), Communications in Computer and Information Science, vol. 1251, pp. 342–354, 2020.

[6]. [6] S. Vejseli, A. Rossmann, and T. Connolly, “Agility matters! Agile mechanisms in IT governance and their impact on firm performance,” Technological Forecasting and Social Change, vol. 178, Art. no. 121572, 2022.

[7]. [7] G. Ghiotto, L. Murta, M. Barros, and A. van der Hoek, “On the nature of merge conflicts: A study of 2,731 open source Java projects hosted by GitHub,” IEEE Transactions on Software Engineering, vol. 46, no. 8, pp. 892–915, 2020.

[8]. [8] O. Elazhary, C. Werner, Z. S. Li, D. Lowlind, N. A. Ernst, and M.-A. Storey, “Uncovering the benefits and challenges of continuous integration practices,” IEEE

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

Transactions on Software Engineering, vol. 48, no. 7, pp. 2570–2583, 2022.

[9]. [9] R. Amaro, R. Pereira, and M. Mira da Silva, "Capabilities and practices in DevOps: A multivocal literature review," IEEE Transactions on Software Engineering, vol. 49, no. 2, pp. 883–901, 2023.

[10]. R. Amaro, R. Pereira, and M. Mira da Silva, "Capabilities and metrics in DevOps: A design science study," Information & Management, vol. 59, no. 7, Art. no. 103640, 2022.

[11]. S. Rafi, W. Yu, M. A. Akbar, A. Alsanad, and A. Gumaei, "Prioritization based taxonomy of DevOps security challenges using PROMETHEE," IEEE Access, vol. 8, pp. 105426–105446, 2020.

[12]. J. Serrano, J. Faustino, D. Adriano, R. Pereira, and M. Mira da Silva, "An IT service management literature review: Challenges, benefits, opportunities and implementation practices," Information, vol. 12, no. 3, Art. no. 111, 2021.

[13]. S. Kotusev, S. Kurnia, and R. Dilnutt, "The practical roles of enterprise architecture artifacts: A classification and relationship," Information and Software Technology, vol. 147, Art. no. 106897, 2022.

[14]. J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of Internet of Things," IEEE Internet of Things Journal, vol. 7, no. 6, pp. 4682–4696, 2020.

[15]. M. Eulerich, J. Pawlowski, N. J. Waddoups, and D. A. Wood, "A framework for using robotic process automation for audit tasks," Contemporary Accounting Research, vol. 39, no. 1, pp. 691–720, 2022.

[16]. [16] Y. Zhao, K. Damevski, and H. Chen, "A systematic survey of just-in-time software defect prediction," ACM Computing Surveys, vol. 55, no. 10, Art. no. 201, 2023.

[17]. C. Vassallo, S. Panichella, F. Palomba, S. Proksch, H. C. Gall, and A. Zaidman, "How developers engage with static analysis tools in different contexts," Empirical Software Engineering, vol. 25, no. 2, pp. 1419–1457, 2020.

[18]. M. Nachtigall, M. Schlichtig, and E. Bodden, "A large-scale study of usability criteria addressed by static analysis tools," in Proc. 31st ACM SIGSOFT Int. Symp. Software Testing and Analysis (ISSTA), 2022, pp. 532–543.

[19]. K. Dias, P. Borba, and M. Barreto, "Understanding predictive factors for merge conflicts," Information and Software Technology, vol. 121, Art. no. 106256, 2020.

[20]. P. Zerbino, A. Stefanini, and D. Aloini, "Process science in action: A literature review on process mining in business management," Technological Forecasting and Social Change, vol. 172, Art. no. 121021, 2021.

[21]. N. Martin et al., "Opportunities and challenges for process mining in organizations: Results of a Delphi study," Business & Information Systems Engineering, vol. 63, no. 5, pp. 511–527, 2021.

[22]. T. Grisold, J. Mendling, M. Otto, and J. vom Brocke, "Adoption, use and management of process mining in practice," Business Process Management Journal, vol. 27, no. 2, pp. 369–387, 2021.

[23]. A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: A survey of case studies," ACM Computing Surveys, vol. 55, no. 6, Art. no. 114, 2023.

[24]. D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (MLOps): Overview, definition, and architecture," IEEE Access, vol. 11, pp. 31866–31879, 2023.

[25]. A. Barredo Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Information Fusion, vol. 58, pp. 82–115, 2020.

[26]. J. vom Brocke, A. Hevner, and A.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 10 October 2026
Page No: 5854 - 5872
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0758

Maedche, “Introduction to design science research,” in Design Science Research. Cases, Cham, Switzerland: Springer, 2020, pp. 1–13.