International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 6, November – December 2025 ||

DOI: 10.15680/IJCTECE.2025.0806037

Managing Change and Configuration at Scale in IT Service Management (ITSM) Platforms

Mahesh Kumar Damarched

Enterprise Programmer Analyst, Louisville, Kentucky, USA

mahesh.damarched@gmail.com

ABSTRACT: Enterprises with large operations conduct Information Technology (IT) services on a hybrid cloud and distributed infrastructure, and thus, poor change control and quality configuration information rapidly become outages, rework, and audit gaps. The paper addresses the style in which Information Technology Infrastructure Library (ITIL)-aligned practices, Configuration Management Database (CMDB) design, automation, and scale selections on IT Service Management (ITSM) platforms deal with change and configuration. A CMDB is a relational store where configuration items (CIs) are stored, dependencies, and is anticipated to encompass more than devices and software to include applications, documentation, people, and processes, as well as providers. At scale, such dependencies cross between teams and tools, and this is what motivates governance controls such as configuration control boards that approve, verify, and control CMDB-related changes. Configuration management has also been positioned as a basis of ensuring consistency and facilitating a security posture throughout the Information and Communication Technology (ICT) lifecycle, which increases the stakes of CMDB accuracy and traceability. The report aims to achieve five objectives: defining the nature of large-scale change challenges, comparing CMDB models in practice, assessing automation and monitoring of change processes, analyzing platform scaling limitations, and providing a case study to integrate governance, CMDB strategy, and automation into a single working pattern.

KEYWORDS: IT Service Management (ITSM), Change Management, Configuration Management Database (CMDB), Automation, Platform Scalability, ITIL Framework, Enterprise IT Operations

I. INTRODUCTION

ITSM can be defined as the aligned practices, processes, and enabling tools applied to plan, deliver, operate, and improve IT services to business requirements[24]. In the real-world platform considerations, ITSM is manifested in the way people interact with the everyday processes[13]. It is related to incident intake, request fulfillment, change approvals, release coordination, asset and configuration tracking, and reporting. In teams using a single intake path and a record of work, handoffs can be traced, reducing the number of requests lost between teams. The importance of traceability is that most ITSM projects fail when the ownership of processes remains vague, and work stagnates in informal channels[43]. That is why ITSM should be used in conjunction with other tools[19]. ITIL commonly provides the standard language and operating hypothesis that make those workflows within enterprise settings consistent[5][16]. In one of the ServiceNow case studies, based on interviews, respondents associated ITIL use with more positive resource control, more effective communication with customers and suppliers, and standardized processes. Such a standardization is a scaling mechanism, as local practices, once ad hoc, cease to work when cross-departmental and cross-geographical services are involved[32].

The core of this scaling problem lies in change and configuration management. Today, Ahmed et al. reiterate that there is an increased (AI)-driven troubleshooting system that assists in change management[1]. Individually, each change request can appear mundane, but service dependencies, shared infrastructure, identity controls, and actual environmental conditions will determine its downstream consequences. In practice, the toughest part is when teams do not realize that services share components, leading to a narrow scope of change that becomes a large-scale outage. Such a risk increases in distributed ICT environments where the lifecycle integrity can be compromised by uncontrolled changes and weaken the intended security posture[4]. In the case of organizations running distributed ICT systems, configuration management is the practice that ensures integrity and consistency of system elements throughout their lifecycle which supports the desired security posture[4][35]. The importance of that framing lies in the fact that daily operational controls (approvals, audits, baselines) are directly linked to business continuity and risk controls.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 6, November – December 2025 ||

DOI: 10.15680/IJCTECE.2025.0806037

A CMDB is frequently viewed as the memory of enterprise activities. However, it is only in cases when the information in it is correct and useful. According to CMDB planning guidance, CMDB is a relational database in which relationships represent links or dependencies between Configuration Items (CIs). According to Kanji, CMDB fundamentals are deeply rooted in ITL[18]. Without such relationships, reporting and impact analysis suffer because teams cannot trace the dependencies between items. To eliminate the expert system, an example that will assist in this issue is to automatize CI relationship maintenance to ensure that the CMDB remains useful despite daily changes in infrastructure[15]. According to Dande et al's work, a CMDB should have a wide range of information elements (applications, middleware, documentation, people, processes, providers), which is an indicator that the concept of service configuration goes far beyond a hardware inventory culture[6][18]. As the scope widens, the operational model must also expand. In doing so, the CMDB will require that data owners, providers, and consumers be assigned specific roles, lest it become a database of untrustworthy information.

One of the guiding questions this paper addresses is: How can large organizations cope with change and configuration at scale in ITSM platforms without losing control, speed, or data integrity? The objectives are direct to map the key scaling issues in ITSM change management; to compare CMDB models (traditional, hierarchical, federated, hybrid); to evaluate automation and monitoring practices of change workflows; to consider platform-specific scaling constraints (with the concrete anchor ServiceNow); and to present a case study, demonstrating how governance, CMDB strategy, and automation may work as one operational system[33]. One potential method for maintaining scope is to model a single service in the CMDB and validate it before repeating the pattern service by service. This staged method allows auditing at every step, so that each batch of services can be verified before another batch is onboarded.

II. CHANGE MANAGEMENT CHALLENGES

Meyer et al. postulate that an unplanned IT outage can be expensive for any organization[28]. Change management in ITSM controls involves requesting, evaluating, approving, implementing, and reviewing changes that may impact services[14][36]. Defining it is not hard. The hard part is volume and coupling. When infrastructure is operated across distributed, large-scale estates, configuration control functionality should reduce unapproved or undocumented modifications, coordinate among different stakeholders, and evaluate the risk of changes requested by the organization's boards, including the Configuration Control Board (CCBs)[4]. That statement places coordination and authorization at the heart of change control. At the enterprise level, teams are under pressure to move fast. However, the same source enumerates tangible ways in which real environments deviate from the approved baseline, including emergency changes without post-documentation, the absence of change management processes, errant changes, and failed changes[4]. Those are not notional risks; they are operational modes of failure that multiply with the number of weekly deployments.

Distributed infrastructure complicates impact analysis because dependencies span tools and teams. A CMDB can reflect those dependencies, but it makes the list of stakeholders that can be influenced by a change decision longer. In CMDB planning a security-network organization, the CMDB planning scope clearly identifies data providers who generate or combine fundamental data, data consumers who need CMDB data, and data owners of that data[18]. The tension in that division is that change management must make decisions quickly, whereas configuration management must maintain disciplined data ownership and verification. This requires the implementation of a robust IT infrastructure, as argued by Serrano and Pereira[42]. With ownership uncertainty, the CMDB loses its reliability. This loss makes the change advisory process subsequently run on less evidence.

The early stages of platform adoption are usually dominated by human and organizational bottlenecks, despite robust tooling. In the ServiceNow interview research, the respondents discussed initial resistance, training that was not as relevant as it might have been and criticized the workshop organization. This is because those training and adoption gaps directly affect the result of change, as users lacking trust in the system or lacking an understanding of the workflow will either bypass approvals, avoid documentation, or keep their own records outside the system, that is, have a shadow record. This is a concept shown by Veldyaeva et al.[46]. It is essential to note that systems exist within other systems, as illuminated by Zijlstra[49]. The same study continues to state that once overcoming initial resistance, employees appreciated a better overview and some assistance and control over metrics, and modules helped to automate some functions[40]. Adoption is not a fluffy aspect. It is a tool that makes the difference between the platform becoming the actual functioning backbone or another layer of administration.

Change collisions are further increased by technical complexity. Hybrid and federated architecture are common practices in many organizations, resulting in conflicting sources of truth and visibility into states[30]. Hybrid cloud

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 6, November – December 2025 ||

DOI: 10.15680/IJCTECE.2025.0806037

comprises private and public clouds, as explained by Lukkarinen[21]. An actual example can be found in a CMDB-IPAM integration thesis. Here, the author notes the Lack of reporting opportunities due to fragmented data and the absence of relationships between configuration items[15]. It is a direct correlation between the quality of data models and the quality of operational decisions. When the CMDB cannot depict relationships (service-to-server, server-to-IP-range, application-to-dependency), change evaluation becomes guesswork, and the company is forced to rely on tribal expertise rather than verified configuration foundations.

Complex environments require baseline integrity and the organization's capability to reconcile between the planned and actual states to conduct risk assessment and impact analysis[3]. In the paper on the blockchain-based security configuration management, the authors compare the "as-certified" and "as-maintained" status and provide the drift reasons, including unauthorized modifications, process variances, and change control deficiency[4]. In cases where drift is present, impact analysis cannot be decomposed. One configuration baseline is used in a change plan, and the actual environment executes a different one. At this point, verification and auditing functions come into operation. Other configuration verification and audit functions outlined in the same work include security compliance tests and gap analysis to identify the anomalies between the baseline and deployed reality[4]. In practice, audits can be viewed as a team's periodic compliance work; at a large scale, auditing also serves as a feedback mechanism that helps maintain change decisions grounded in truth.

Generally, governmental organizations need to change as the amount of change grows. According to Sarwar et al., it is substantial to lead proactive change through appropriate systems[41]. The management model used in the CMDB planning case translates to a chain in which the manager of the configuration process proposes a CMDB change, demand management reviews it, configuration analysts implement it, and it is verified, audited, and reported as a successful implementation[18]. It is a clear example of change in the CMDB itself being a managed change, including review and verification steps[11]. This type of CMDB-governance discipline is typically skipped by organizations that experience a familiar trend of reduced CMDB-weakening, reduced change-impact-analysis-trust, increased change, with weaker level control keep up with delivery demand trends.

III. CONFIGURATION MANAGEMENT MODELS

The concept of configuration management in ITSM focuses on maintaining a correct model of services, their parts, and interrelationships to ensure that teams can safely operate and modify them, as illustrated by Patel[34]. This is seen in Leon's analogy of accountability and efficiency[20]. Teams in their day-to-day activities also tend to use the CMDB as the last port of call when they want to know what a change would impact before they approve it. A CMDB is the repository of data on the configuration of hardware and software within an organization. It may also include additional aspects of the organization, such as applications, middleware, documentation, people, processes, and providers. The fact that the CMDB is a relational table is important, as dependencies can be interrogated and monitored through relationships. This incorporation supports impact analysis and faster incident investigations.

Conventional CMDB models act more like centralized inventories: one database, one schema, and a load everything attitude. This centralized style may become slow and controversial as soon as the scope increases. It is an aspect that can grow, as each team would want the schema modeled according to its own requirements. That solution is viable on a small scale. However, it is prone to breaking down when the business already has master datasets in other places (data center infrastructure management, identity systems, IP Address Management (IPAM), application catalogs). According to Kanji, CMDB planning guidance, the CMDB is described as a single source of truth for vital information[18]. However, existing master databases are recognized as possible masters, with the CMDB providing the necessary information via Application Programming Interface (APIs). That is already a step towards non-pure centralization and an integration-based model.

It is a hierarchical model, organizing CIs into tiers (service, application, and infrastructure) that facilitates easy navigation but may conceal cross-cutting relationships if the model forces everything into a single tree. Here, teams may find it hard to make incident calls, since a clean hierarchy may still fail to display shared dependencies that are not found within the branch of the tree typically referred to as the main branch. The IPAM-CMDB integration paper demonstrates why the relationships between the configuration items beyond the hierarchy are important. The missing relationships between configuration items led to reporting gaps and reduced visibility[15]. The CMDB model fails to represent many-to-many dependencies (e.g., an application uses many services; one platform has many apps; one IP range has many segments), and as a result, teams cannot reliably trace the impact across domains.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 6, November – December 2025 ||

DOI: 10.15680/IJCTECE.2025.0806037

Federated and hybrid models do not treat the CMDB as the sole storage point but rather as a coordination layer. In practice, such an arrangement can diminish disputes over who owns the truth, since each area retains its own store of authority, while the CMDB maintains the links required by operations teams. The result of the same integration thesis means directional integration of the ServiceNow and NetBox systems, as well as a condition-based automated CI relationship manager that controls relationships[15]. This is a reasonable hybrid model: Authoritative IP data: store in IPAM, ITSM processes and CI records: store in ServiceNow, and then automate relationship maintenance to ensure that operational teams can query dependencies without copying all the data into the CMDB.

The accuracy of configuration directly influences change success rate and efficiency, as fewer surprises lead to lower operational efficiency, as noted by Mao et al[25]. A well-managed approval process may not be effective when an outdated CI record is involved. This is because the reviewers may end up signing a change with the incorrect picture of the environment. Configuration control serves to analyze risk, benefits, and costs of changes and reject or accept requests through governance boards like CCBs[4]. The given mechanism of governance can only perform its functionality when CIs can be traced to configuration states, which the same work terms as configuration identification and configuration traceability to historical, current, and planned states[4]. In other words, in case the model fails to provide the answer to the question of what runs where, and what depends on what, then the organization cannot scale safe change, regardless of how well-polished the workflow screens may be. This is something learned from Mudau et al. discussing the adoption of Zero Trust Architecture (ZTA) across organizational contexts[31].

IV. AUTOMATION OF CHANGE WORKFLOWS AND MONITORING TOOLS

ITSM change workflow automation has the promise of higher throughput and more predictable execution[7][39]. This is despite it only working when the underlying process and data model remain coherent. The interviewees in the ServiceNow interview study were asked whether the platform enabled some of their functions to be automated using its modules, and they said they found it simple to gain a better overview and control their metrics, as indicated by Santos and Rodrigues[40]. These are inferences observed after the initial resistance to Adoption dropped. That relates automation to observability: metrics and visibility enable the teams to trust automation, and trust leads to usage.

Nevertheless, automation does not eliminate governance; it only alters it. In the CMDB management model case, the proposed CMDB field change is to be passed through the demand management evaluation, implementation phase, and verification and audit confirmation before it is marked as successful[18]. As these activities unfold, it is vital to note that change requires resource agility, as noted by Mengistu[26]. This order aligns well with the pattern of automating steps and retaining gates. Routing, evidence capture, and validation can be automated, but approvals and ownership remain.
Event-driven automation and monitoring are important as in large-scale environments, drift and undocumented changes become the rule rather than the exception. The security configuration management literature based on blockchain identifies emergency changes without post-documentation, non-adherence to guidelines, and risky changes without rollback as the causes of divergence between "as-certified" and "as-maintained" states[4]. Generally, security is a significant topic that should be considered critically in any system, as described by Yordanov[48]. That list is like a roadmap for monitoring. It is proposed to identify unwanted changes, take emergency actions, create red flags, and facilitate reconciliation. Verification and auditing functions are defined in the study as comprising compliance checks and gap analysis between the status quo and the status maintained.

On large ITSM platforms, control can be automated so that teams run checks before approval (pre-change validation) and after deployment (post-change verification) and then push the alert into the ITSM record so that the system of record has the evidence trail. Chatziamanetoglou and Rantos indicate that when the organization requires stronger tamper-resistance, the permissioned blockchain model suggests immutable CI records and historical traceability that are auditable and recoverable[4]. Integrity automation occurs when a configuration supervision is too large to run by hand.
The primary limitation to automation is integration constraints, as described by Mishra and Ramakrishnan et al.[29][38]. Automation has no sense of missing relationships. According to the CMDB-IPAM integration case, the data silo and the absence of CI relationships diminished reporting capabilities, and the suggested solution involved automating the maintenance of CI relationships to enhance visibility[15]. This demonstrates a realistic sequence of procedures of enterprises. It helps fix relationship coverage and, subsequently, automate change checks and monitoring workflows that rely on the defined relationships.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 6, November – December 2025 ||

DOI: 10.15680/IJCTECE.2025.0806037

V. PLATFORM-SPECIFIC CONSIDERATIONS

The selection of platforms affects how a business manages scalability, customization, and integration, as depicted by Makani and Jangampeta[23]. In practice-oriented research on ServiceNow adoption, interviewees explain ITIL-aligned standard practices. According to Gangula, ITIL has undergone significant changes over time[10][12]. They also state that the selection of the project team and training are determinants of the success of the implementation process. Users report strong support for the project team but also experience some problems. Friedrichsen et al. encourage individuals to consider diverse user experiences when considering optimal change outcomes[9]. They point out that information is not transferred during the decision-making process, and they are not very flexible and communicative[37] [40]. That is important to scale since platform deployment is not a technical rollout but a governance rollout. Many institutions have adopted the systems, as noted by Machaladze[22]. A lack of communication can lead to local exceptions arising from global or multi-department deployments, which subsequently result in integration failures.

Scalability appears in two areas: platform performance and the scalability of the operating model. Kanji shows that the CMDB planning case treats the CMDB as a single source of truth for key information and supports integration between master systems through APIs[18]. It means that platform scalability is based on integration throughput and data reconciliation quality, rather than the size of the database. There is a distinctive growth in data, contributing to what Dritsas and Trigka describe as the "Big Data revolution"[8]. The same source also suggests a horizontal implementation strategy. Generally, implementation is distinctive, as it offers numerous advantages[45]. It represents all services in the CMDB and proceeds to the next. The method is used as a scaling mechanism for global business because it addresses scope creep, maintains stakeholder focus, and establishes a repeatable pattern for service onboarding into the CMDB.
The integration constraint also lies in how data models are processed on the platform. ServiceNow and NetBox are assigned to the CMDB-IPAM integration, with the former acting as the master of the IPAM and the latter as the desired result of the integration of the two parts, which is enhanced data about the devices in ServiceNow with data about the IPAM (unidirectional integration and automated relationship management)[15]. That is an enterprise reality. The ITSM platform does not always have all domain data, so it must ingest and reconcile data without assuming the authoritative role in the origin system. If the platform's configuration data model or integration tooling complicates this, the organization will either recreate data in an unsafe manner or receive partial relationships.

The security architecture also develops platform constraints. Research on blockchain-based configuration management indicates that it can meet the confidentiality, integrity, and availability requirements of distributed ICT systems. Chatziamanetoglou and Rantos attribute it to a permissioned ledger, suggesting that it can also support role-based access control to regulate participant privileges[4]. That is consistent with the practical requirement to limit individuals' ability to modify CI records, grant authorization to make changes, and view sensitive connections within international businesses. When an ITSM platform is scaled across a large number of departments, it must provide access controls at both the workflow and data levels. The CMDB turns into a security liability rather than a control asset at this stage.

VI. CASE EXAMPLE

An example of managing change and configuration at scale in ITSM platforms is a large and distributed infrastructure with many data centers and cloud environments[17], as attributed by Merseedi and Zeebaree[27]. In this context, ITSM workflows and CMDB records are stored using ServiceNow, and IP addressing and network source-of-truth are stored in NetBox. The first state has poor reporting and visibility of service due to the silo existence of configuration data and incomplete CI relationships. It is like the issue outlined in the CMDB-IPAM integration thesis, where one problem is inadequate reporting due to data silos and the absence of links between configuration items. The emergency fixes and undocumented updates also cause the organization to experience the baseline drift, which aligns with the enumerated causes of divergence of the 'as-certified' and 'as-maintained' states, including the emergency change implementations without post-documentation and the absence of change control practices[4].

The organization begins by establishing the CMDB operating model and governance lanes. It establishes a configuration control board and allocates duties across demand management, configuration planning, configuration control, and verification and audit[18]. It does this, in accordance with the CMDB planning recommendations, by organizing a CCB and distributing its duties across the relevant management areas. It also visualizes data providers, consumers, and owners to eliminate confusion about who holds which CI attributes[18]. Instead of trying a big-bang import, it uses the suggested horizontal method. This method involves fully representing one service in the CMDB, testing it, and moving to the next.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 6, November – December 2025 ||

DOI: 10.15680/IJCTECE.2025.0806037

The organization then fixes relationship coverage through integration. It implements a one-way collaboration between NetBox and ServiceNow[15]. Babar et al. acknowledge the essence of collaboration in change management. Relationship coverage also uses an automated CI relationship manager that ensures the relationship remains within specified conditions[2]. This directly addresses the previously mentioned visibility failure by establishing relationships and links to support impact analysis and service reporting. The organization then establishes the rule: any change request must mention the affected CIs and the proven relationships in the CMDB to be approved. It follows the verification and audit role outlined in the CMDB management model to ensure that new fields, attributes, and relationships are present as expected after implementation.

The workflow is then redesigned through automation. Standard routing, evidence capture, and status updates are automated using ServiceNow modules, which align with interview findings that the platform can be automated to perform certain functions and provide a better overview and metric control after employees are no longer resistant to it[40]. Despite these implementations, the organization struggles with adoption issues, as the quality of training and information exchange determines the consistency with which teams adhere to the workflow[40]. To ensure that fast fixes do not evade controls, the organization implements post-change validation checks, as noted by Chatziamanetoglou and Rantos[4]. This disposition compares the object's deployed state with the baseline, where gap analysis patterns are defined as configuration verification and auditing functions. In case of validation, such as drift or incomplete documentation, a follow-up task is automatically generated against the same change record.

The organization then enhances high-risk configuration data integrity controls. Chatziamanetoglou and Rantos argue that critical services investigate the concept of permissioned ledgers to maintain tamper-resistant CI persistence and an immutable history of changes with the assistance of role-based access control[4]. The organization embraces the underlying discipline (even without blockchain), rigorous role segregation, auditable approvals, and traceability through configuration states. They are quantified based on the operations outlined in the sources. These include increased visibility and reporting potential, decreased operational costs, and accelerated reaction time to dynamic environments connected to data flows, which enable and automate CI relationships.

VII. LESSONS LEARNED

There are diverse lessons to be drawn from this research. For instance, scale penalizes poor configuration data. The collected literature relates the quality of the relationship and the integrity of the basis to the manner of operation. Change risk analysis is not as feasible when siloed information and a lack of relationships limit reporting. By bypassing documentation and rollback processes during emergency changes, the organization loses its approved baseline[4]. Such a loss reduces the reliability of the organization's services and its security posture. Vibrant enterprises consider the CMDB a dependency map of relationships and invest in verification and audit loops, as compliance checks and gap analysis should be part of routine operations rather than an audit. Stokes argues that the Enterprise Digital Twin (EDT) enables continuous growth by embracing the change that comes with automation[44].

Kanji's study also suggests that the platform can scale cleanly or not, depending on the design of its governance and operating model[18]. An effort in CMDB planning establishes data providers, data owners, and consumers, providing the enterprise with the means to store data and use it as the scope increases. The same findings provide a regulated workflow of CMDB change that consists of evaluation, implementation, and verification[18]. This is easily translated into automated processes that capture evidence, route it, maintain approvals based on risk level, and implement verification as a gate. The horizontal build approach, in which a single service is entirely modeled and expanded, is also a feasible control mechanism against CMDB sprawl.

Lastly, there is a real scaling limitation of people-side adoption. Participants in the ServiceNow interview study mentioned resistance at an initial stage and inadequate training; subsequently, a greater appreciation of enhanced overview and metric control was reported[40]. That arc hints at a pragmatic suggestion. It makes training, communication, and the disclosed nature of decisions part of the control environment, because inadequate training might lead to workflow avoidance, which may result in undocumented modifications and the deterioration of the base. Chatziamanetoglou and Rantos found that when an enterprise seeks greater tamper resistance for configuration records, permissioned blockchain designs offer immutable CI records, traceability, and role-based access control to limit privileges[4]. The lesson is applicable even when organizations do not use that architecture. Tight control of access and traceability enables change management to be more believable at scale[47].

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 6, November – December 2025 ||

DOI: 10.15680/IJCTECE.2025.0806037

VIII. CONCLUSION

Scale change and configuration management in ITSM platforms should not be done with workflow tooling. It demands a reliable configuration ground, controlled ownership of configuration information, and verification loops that maintain the working picture in touch with reality. The literature views configuration management as one of the practices that maintain system integrity and consistency throughout the lifecycle phases and ensure a secure posture in distributed ICT systems. They also demonstrate that drift manifests in operational behaviors such as unauthorized modifications, hasty, undocumented fixes, and failed fixes without rollbacks. These behaviors scale rapidly at an enterprise level, and an organization needs to consider change control and configuration verification as part of day-in, day-out disciplines.

According to the research, CMDB is an interrelated database of CIs and their connectivity, designed to encompass a wide range of organizational elements beyond devices. However, the CMDB only facilitates safe change when its information remains consumable. Integration-based solutions, in which master databases are considered authoritative and APIs supply vital information into the CMDB, are appropriate to the enterprise reality outlined in CMDB planning work. The case of CMDB-IPAM integration indicates that siloed information and automated maintenance of CI relationships can enhance visibility, reduce operational costs, and accelerate response times for dynamic settings.

Execution is defined by automation and platform choice, and not governance. The evidence of ServiceNow adoption indicates that the platform will enhance overview, control metrics, and automate certain functions once issues with resistance and training are resolved. Sized operations still require definitions, decision transparency, and verification gates as indicated in the CMDB management model flow proposal, all the way to audit confirmation. Once these elements are synchronized, enterprises can be fast while still maintaining control. These dispositions leverage the CMDB as the dependency map, automation as the execution engine, and governance as the guardrail that prevents scale to rise into chaos.

REFERENCES

1. Ahmed, S., Rahman, A., & Ashrafuzzaman, M. (2023). A systematic review of ai and machine learning-driven it support systems: Enhancing efficiency and automation in technical service management. American Journal of Scholarly Research and Innovation, 2(02), 75-101. https://researchinnovationjournal.com/index.php/AJSRI/article/view/22
2. Babar, Z., Paul, R., Rahman, M. A., & Barua, T. (2025). A systematic review of human-ai collaboration in it support services: Enhancing user experience and workflow automation. Journal of Sustainable Development and Policy, 1(01), 65-89. https://doi.org/10.63125/grqtf978
3. Baptista, B., & Barata, J. (2024). Continuously improving IT service management in the pharmaceutical industry. Procedia Computer Science, 239, 923-930. https://doi.org/10.1016/j.procs.2024.06.253
4. Chatziamanetoglou, D., & Rantos, K. (2023). Blockchain-based security configuration management for ICT systems. Electronics, 12(8), 1879. https://doi.org/10.3390/electronics12081879
5. Dande, F., & Li, X. (2023). Enterprise Service Management Cybersecurity Threats: Exploring Cloud Configuration Management Database (CMDB) Implementation Within Community Colleges. 8th North American Confer-ence on Industrial Engineering and Operations Man-agement. https://www.academia.edu/download/105841961/Enterprise_Service_Management_Cybersecurity_Threats_Exploring_Cloud_Configuration_Management_Database.pdf
6. Dande, F., Li, X., Shofoluwe, M., & McLeod, A. (2024, October). Artificial Intelligence integration in IT Service Management: An ITIL configuration management process review. In Proceedings of the International Conference on Industrial Engineering and Operations Management, Detroit, MI, USA (pp. 9-11). https://www.academia.edu/download/119271703/Artificial_Intelligence_integration_in_IT_Service_Management_An_ITIL_configuration_management_process_review.pdf
7. Davila, A., Janampa, R., Angeleri, P., & Melendez, K. (2020). ITSM model for very small organisation: An empirical validation. Iet Software, 14(2), 138-144. https://doi.org/10.1049/iet-sen.2019.0034
8. Dritsas, E., & Trigka, M. (2025). A Survey on database systems in the big data era: Architectures, performance, and open challenges. IEEE Access. https://ieeexplore.ieee.org/abstract/document/11008610
9. Friedrichsen, L., Vandetta, T., Wheeler, A., Rothwell, C., Bangs, S., Merino, A., & Petersen, K. (2025, April). One IT serving many-using ITSM to drive organizational change and align a federated IT organization. In Proceedings of the 2025 ACM SIGUCCS Annual Conference (pp. 18-21). https://doi.org/10.1145/3675229.3712528
10. Gangula, S. (2025). A comprehensive review of ITIL frameworks for managing large-scale retail cloud operations and challenges.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 6, November – December 2025 ||

DOI: 10.15680/IJCTECE.2025.0806037

https://www.academia.edu/download/125423948/A_Comprehensive_Review_of_ITIL_Frameworks_for_Managing_Large_Scale_Retail_Cloud_Operations_and_Challenges.pdf

11. González Ferreiro, M., & Newhouse, S. (2022). IT Service management system for EMBL's European Bioinformatics Institute's IT department. F1000Research, 11, 507. https://f1000research.com/articles/11-507
12. Harjanto, A., & Aji, R. F. (2024). Improving IT assets management with ITIL 4 framework. Jurnal Ilmu Komputer dan Informasi, 17(2), 127-143. https://doi.org/10.21609/jiki.v17i2.1195
13. Hasan, M. M., & Islam, M. M. (2023). Reinforcement learning approaches to optimize IT service management under data security constraints. American Journal of Scholarly Research and Innovation, 2(02), 373-414. https://doi.org/10.63125/z7q4cy92
14. Inavolu, M. (2025). Challenges in ITSM digital transformation: A case study. https://urn.fi/URN:NBN:fi-fe2025060963061
15. Jarmolinski, J. (2022). Configuration management database integration with IPAM database for cloud computing company. https://www.theseus.fi/bitstream/handle/10024/750749/jakub_jarmolinski.pdf?sequence=2
16. Jassal, H. (2025). The evolution of IT service management: Navigating digital transformation in the modern enterprise. Journal Of Engineering And Computer Sciences, 4(8), 327-339. https://sarcouncil.com/2025/08/the-evolution-of-it-service-management-navigating-digital-transformation-in-the-modern-enterprise
17. Jyoti, S. N. (2025). ITSM based change management automation in cloud environments: A cross sector empirical study. Review of Applied Science and Technology, 4(02), 440-472. https://doi.org/10.63125/xvjst226
18. Kanji, K. (2024). The planning of a configuration management database for Suomen Erillisverkot Oy: CMDB and management model. https://urn.fi/URN:NBN:fi:amk-202401131384
19. Ketolainen, N. (2025). Assessing the current state of the Service Integration and Management (SIAM) team: Organizational insights and recommendations. https://urn.fi/URN:NBN:fi:amk-2025120131138
20. Leon, P. (2020). Creating accountability and increasing efficiency by implementing an IT service management solution. https://scholarworks.lib.csusb.edu/etd/1047/
21. Lukkarinen, P. (2020). Data center automation-and hybrid cloud system requirements. https://urn.fi/URN:NBN:fi:amk-2020052915435
22. Machaladze, O. (2025). IT infrastructure management in educational institutions using the ITIL framework. International Science Journal of Engineering & Agriculture, 4(2), 215-225. https://doi.org/10.46299/j.isjea.20250402.14
23. Makani, S. T., & Jangampeta, S. (2024). A comparative study of platform engineering tools: Implications for system design and scalability. Journal ID, 1552, 5541. https://www.researchgate.net/profile/Sai-Teja-Makani/publication/381458639_A_Comparative_Study_of_Platform_Engineering_Tools_Implications_for_System_Design_and_Scalability/links/666dd35a85a4ee7261c5a999/A-Comparative-Study-of-Platform-Engineering-Tools-Implications-for-System-Design-and-Scalability.pdf
24. Mándi, Á. (2024). Agile IT service management: an analysis and enhancement of ITIL practices in corporate environments (Doctoral dissertation). https://repositorio.ucp.pt/server/api/core/bitstreams/e185c455-22ed-4f00-9fce-9cedb53a937a/content
25. Mao, H., Zhang, T., & Tang, Q. (2021). Research framework for determining how artificial intelligence enables information technology service management for business model resilience. Sustainability, 13(20), 11496. https://doi.org/10.3390/su132011496
26. Mengistu, F. M. (2020). IT service management agility assessment model: the case of CBE (Doctoral dissertation, Doctoral dissertation, Addis Ababa University). https://www.academia.edu/download/68188151/IT_Service_Management_Agility_Assessment_Model_the_case_of_CBE.pdf
27. Merseedi, K. J., & Zeebaree, S. R. (2024). The cloud architectures for distributed multi-cloud computing: a review of hybrid and federated cloud environment. The Indonesian Journal of Computer Science, 13(2). https://doi.org/10.33022/ijcs.v13i2.3811
28. Meyer, R., Wittmann, S., Le, A. T., & Krcmar, H. (2021). A concept for an adaptive case management system supporting the incident management process. Available at SSRN 3871786. https://dx.doi.org/10.2139/ssrn.3871786
29. Mishra, A. (2019). Exploring ITIL and ITSM change management in highly regulated industries: A review of best practices and challenges. https://www.academia.edu/download/124599203/Exploring_ITIL_and_ITSM_Change_Management_in_Highly_Regulated_Industries_A_Review_of_Best_Practices_and_Challenges.pdf
30. Moussaoui, J. E., Kmiti, M., El Gholami, K., & Maleh, Y. (2025). A systematic review on hybrid AI models integrating machine learning and federated learning. Journal of Cybersecurity and Privacy, 5(3), 41. https://doi.org/10.3390/jcp5030041

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 8, Issue 6, November – December 2025 ||

DOI: 10.15680/IJCTECE.2025.0806037

31. Mudau, K., Mudumani, K., & Zwane, S. M. (2025). Zero trust architecture: Frameworks and implementation strategies in modern cybersecurity. Available at SSRN 5637091. https://dx.doi.org/10.2139/ssrn.5637091
32. Narapareddy, V. S. R., & Yerramilli, S. K. (2022). Scaling the service now CMDB for distributed infrastructures. International Journal of Engineering Technology Research & Management (IJETRM), 6(10), 101-113. https://ijetrm.com/
33. Nookala, G. (2024). Adaptive data governance frameworks for data-driven digital transformations. Journal of Computational Innovation, 4(1). https://researchworkx.com/index.php/jci/article/view/16
34. Patel, N. V. (2024). AI-Augmented ITSM: Autonomous incident triage. International Journal of Research and Applied Innovations, 7(2), 10411-10414. https://doi.org/10.15662/IJRAI.2024.0702001
35. Pohn, D., & Hommel, W. (2022). Reference service model framework for identity management. IEEE Access, 10, 120984-121009. https://doi.org/10.1109/ACCESS.2022.3219044
36. Puupponen, A. (2023). Developing the ITSM process of a case company. https://www.theseus.fi/bitstream/handle/10024/789738/Puupponen_Aleksi.pdf?sequence=2
37. Ramakrishnan, M. (2022). Development and evaluation of it service management digital commons-a case study (Doctoral dissertation, University of Southern Queensland). https://research.usq.edu.au/item/wq850/development-and-evaluation-of-it-service-management-digital-commons-a-case-study
38. Ramakrishnan, M., Gregor, S., Shrestha, A., & Soar, J. (2025). Addressing knowledge gaps in ITSM practice with "learning digital commons": A case study. Information Systems Frontiers, 27(3), 965-989. https://doi.org/10.1007/s10796-024-10483-0
39. Ravichandran, N., Inaganti, A. C., Muppalaneni, R., & Nersu, S. R. K. (2020). AI-Powered workflow optimization in IT service management: enhancing efficiency and security. Artificial Intelligence and Machine Learning Review, 1(3), 10-26. https://doi.org/10.69987/
40. Santos, S. B. M., & Rodrigues, N. J. (2024). ServiceNow: Implications and practice within the business environment. Procedia Computer Science, 239, 11-18. https://doi.org/10.1016/j.procs.2024.06.140
41. Sarwar, M. I., Abbas, Q., Alyas, T., Alzahrani, A., Alghamdi, T., & Alsaawy, Y. (2023). Digital transformation of public sector governance with IT service management–A pilot study. IEEE Access, 11, 6490-6512. https://ieeexplore.ieee.org/abstract/document/10018341/
42. Serrano, J. P., & Pereira, R. F. (2020). Improvement of IT infrastructure management by using configuration management and maturity models: A systematic literature review and a critical analysis. Organizacija, 53(1), 3-19. https://sciendo.com/2/v2/download/article/10.2478/orga-2020-0001.pdf
43. Serrano, J., Faustino, J., Adriano, D., Pereira, R., & Da Silva, M. M. (2021). An IT service management literature review: Challenges, benefits, opportunities and implementation practices. Information, 12(3), 111. https://doi.org/10.3390/info12030111
44. Stokes, G. (2025). Enterprise digital twins in financial services: convergence of architecture, operations, and engineering. Authorea Preprints. https://doi.org/10.36227/techrxiv.175624549.98427022/v1
45. Trinidad, M., Orta, E., & Ruiz, M. (2021). Gamification in IT service management: A systematic mapping study. Applied Sciences, 11(8), 3384. https://doi.org/10.3390/app11083384
46. Veldyaeva, E., Fitz, L. R., & Scheeg, J. (2024). Gamifying IT service management education for future IT professionals. https://aisel.aisnet.org/pacis2024/track14_educ/track14_educ/4/
47. Wulf, J., & Winkler, T. J. (2020). Evolutional and transformational configuration strategies: A RASCH analysis of IT providers' service management capability. Journal of the Association for Information Systems, 21(3), 7. https://aisel.aisnet.org/jais/vol21/iss3/7/
48. Yordanov, Y. (2022). Analysis of methods for IT service management processes implementation in higher education institutions. Engineering 4.0 and the Internet of Everything, 85. https://conference.fdiba.tu-sofia.bg/wp-content/uploads/2023/06/FDIBA-Conference_2022.pdf#page=91
49. Zijlstra, K. K. (2022). Boosting fitness for use of configuration management databases by user specific views. https://repository.tudelft.nl/file/File_80a0ec57-f505-44d8-ae75-b16616f52aef?preview=1