



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Automating the Service Desk: PowerShell-Driven Provisioning, Health Monitoring, and Measurable Reduction of Human Error in Managed IT Security Operations

Kelly Okogbenin^{1*}, Tedum Awi Daniel²

¹Gannon University, Erie, Pennsylvania, USA, okogbeninkelly@gmail.com

²TeKnowledge, Colorado Springs, Colorado, USA, awidaniel@gmail.com

ABSTRACT

This review examines how service-desk automation can strengthen managed IT security operations by reducing preventable human error while improving consistency, traceability, and operational resilience. It evaluates PowerShell-driven provisioning and workflow automation, system-health monitoring, detection and automated remediation, and the measurement of human-error reduction through operational performance indicators. The study adopts a critical integrative review approach, synthesizing literature on IT service management, automation engineering, cybersecurity operations, human factors, governance, access control, and intelligent remediation to establish a unified analytical framework.

The findings show that PowerShell is most valuable when positioned as a governed orchestration layer rather than as an isolated scripting utility. Standardized provisioning, reusable workflows, configuration interrogation, structured logging, validation, and controlled remediation can reduce repetitive manual variation and improve service reliability. However, automation does not eliminate operational risk; flawed logic, excessive privileges, incomplete telemetry, weak validation, and poorly bounded execution can propagate errors rapidly across managed environments. The review further establishes that claims of reduced human error require empirical validation through baseline comparisons, normalized rates, longitudinal analysis, and auditable metrics such as provisioning accuracy, failed changes, configuration deviations, mean resolution time, mean time to detection, mean time to remediation, repeat incidents, rollback frequency, and administrator intervention.

The review concludes that secure automation depends on least privilege, code review, credential protection, approval gates, exception handling, rollback, logging, and human oversight for high-impact decisions. It recommends stronger policy-as-code practices, explainable AIOps, adversarial testing, benchmark datasets, and longitudinal evaluation frameworks to ensure that increasing automation enhances security, accountability, and resilience rather than merely accelerating execution across complex, distributed, and increasingly autonomous managed service environments worldwide.

Keywords: PowerShell automation; service desk; human error; security operations; automated remediation; operational resilience

1. Introduction

1.1 Service Desk Automation in Modern IT

Contemporary service desks are moving from manually intensive support functions toward digitally coordinated control points for enterprise technology services. The traditional model depends on analysts receiving tickets, interpreting requests, executing repetitive administrative steps, documenting outcomes, and escalating unresolved issues. That model becomes increasingly fragile as organizations operate hybrid infrastructure, cloud platforms, distributed endpoints, identity services, security tools, and externally managed environments. IT service management therefore requires service desks to combine user support with process assurance, configuration discipline and traceable operational control. Cross-national evidence shows that operational ITSM processes are often adopted more extensively than strategic processes, reflecting the immediate organizational need to stabilize daily technology delivery through formalized practices (Marrone et al., 2014).

The contemporary service desk is consequently becoming an orchestration layer rather than merely a ticket-routing function. Requests involving onboarding, password and access administration, group membership, software installation, endpoint configuration, service restoration, and routine diagnostics are repetitive yet security-sensitive. When individual analysts perform these activities differently, variability can create inconsistent configurations, delayed resolution, incomplete audit trails, and privilege errors. ITSM research accordingly emphasizes lifecycle-based management, implementation discipline, and structured control of technology services instead of isolated technical interventions (Serrano et al., 2021). Automation

becomes strategically relevant because approved procedures can be encoded into repeatable workflows while escalation remains available for exceptional cases.

Automation, however, should not be reduced to isolated scripting. A script may accelerate a discrete task, whereas mature service-desk automation coordinates triggers, dependencies, approval logic, execution context, validation, exception handling, logging and downstream actions. Ayachitula et al. (2007) emphasized integrating highly automated operational processes with human-centred workflows so that repeatable work can be machine-executed while ambiguous activities remain subject to collaboration and intervention. This preserves human judgment where contextual interpretation remains operationally necessary. This distinction is fundamental in managed security operations, where speed has limited value unless automated actions remain authorized, observable, reversible, and consistent with policy. Automation maturity is therefore defined by how execution is orchestrated and controlled.

The transition is also driven by scale. Enterprise IT teams must maintain reliable services while controlling cost, responding quickly to incidents, supporting continuous change, and governing increasingly heterogeneous technology estates. Automation has therefore been associated with service innovation, reduced repetitive workload, and more consistent delivery of operational tasks (Krishnan & Ravindran, 2017). These pressures are amplified in managed environments because analysts may support multiple customers, tenants, device populations, identity domains, and service commitments. Standardized execution becomes a security requirement as well as an efficiency objective, since repeated manual actions can generate configuration drift, inconsistent permissions, or documentation gaps.

A mature automation model also shifts service management from reactive response toward proactive operational assurance. Traditional service desks become visible after users report failure; integrated monitoring and workflow automation can detect emerging conditions, classify them, and route appropriate actions earlier. Research on IT service operations identifies reactive management, weak measurement, incident classification, and fragmented process interfaces as persistent improvement challenges (Jäntti & Cater-Steel, 2017). Cloud operations strengthen the same requirement for reproducibility: Infrastructure-as-Code demonstrates how provisioning and configuration can be represented in controlled definitions rather than undocumented manual interventions (Özel, Pautz & Schmidt, 2020). These developments recast automation as disciplined execution of approved intent.

Within this transformation, PowerShell occupies a natural but bounded role. In Microsoft-centric environments, it provides a scriptable mechanism for interacting with operating systems, directory services, cloud resources, applications, and management interfaces. Its significance is not that every service-desk activity should become a script, but that standardized administrative logic can be modularized and incorporated into governed workflows. Syed et al. (2020) similarly show that software automation creates value when repetitive work is systematically selected, designed, operated, and controlled rather than automated opportunistically. PowerShell is therefore best understood as an enabling layer within a broader architecture of orchestration, monitoring, policy enforcement, and accountability.

Organizational capability remains equally important. Evidence from South African insurance organizations indicates that implementing ITSM technologies or frameworks without sufficiently embedding their underlying principles can constrain continual improvement and effective service delivery (Mvelase & Smith, 2024). Nigerian scholarship on cloud-based service orchestration likewise describes movement toward integrated platforms combining incident, request, change, configuration, analytics, self-service, and automation capabilities (Ladapo et al., 2023). For managed security operations, automation cannot compensate for weak ownership, unclear approval paths, inadequate skills, or poor process definition. Governance maturity must develop alongside technical capability.

The issue is not whether service desks can automate, but whether they can automate safely, repeatably, and measurably. Smart automation research demonstrates that operational intelligence, real-time monitoring, predictive capability, cybersecurity safeguards, and governance must evolve together if automation is to strengthen resilience rather than merely increase execution speed (Omogun et al., 2024). Modern service-desk automation should consequently reduce preventable manual variation while preserving oversight, evidence, and controlled execution. This framing positions PowerShell-driven provisioning and health monitoring within a wider service-management discipline in which success is judged by standardized execution, visibility of system state, enforceable procedures, traceable actions, and measurable reduction of consequential human error at scale.

1.2 Human Error in Security Operations

Human error in managed IT security operations is better understood as a property of socio-technical work than as evidence that personnel are inherently unreliable. Service-desk analysts operate across identity platforms, endpoints, cloud services, ticketing systems and security controls, often under interruption, time pressure and competing service-level commitments. Repetitive account changes, privilege assignments, configuration edits and incident triage therefore create conditions in which attention can degrade, and cognitive demand can exceed available capacity. Multimodal workload research reinforces the importance of monitoring cognitive burden in complex human-machine environments because excessive workload can impair performance, situational awareness and adaptive decision-making (Adamolekun et al., 2025).

Operational errors also differ in mechanism and consequence. Slips arise when an intended action is executed incorrectly; lapses involve memory or attention failures; rule-based mistakes reflect inappropriate application of procedures; and knowledge-based mistakes emerge when unfamiliar situations require judgment beyond established routines. Cybersecurity reporting remains inconsistent in how such failures are classified, which can obscure whether an incident originated in individual action, workflow design, or latent organizational conditions (Grandhi & Still, 2025). For service desks, this distinction matters because mistyped configuration values, omitted documentation, incorrect group membership, delayed escalation, and unauthorized privilege elevation require different controls.

Evidence from network administrators further shows that information-security failures are shaped by organizational communication, policy, culture, and structure, not merely operator behavior (Kraemer & Carayon, 2007). Weak standardization can leave analysts dependent on memory, informal workarounds or inconsistent local practices. Nigerian cybersecurity research similarly identifies stress, burnout, risky password practices, security fatigue and weak security culture as factors that can increase vulnerability, demonstrating why remediation must address working conditions as well as awareness (Ajufo & Qutieshat, 2023).

Credential handling illustrates this interaction between human behavior and system design. In a South African government setting, password reuse, inconsistent training, unclear roles, and fragmented controls were associated with cybersecurity and operational vulnerabilities, indicating that poor process architecture can normalize risky administrative practices (Ngandu, Mwansa & Mkabe, 2025). During incident analysis, analysts also face prioritization and interpretation demands for which procedural guidance can be incomplete, increasing dependence on judgment under constrained time (Spring & Illari, 2021).

Well-governed automation can reduce this exposure by converting approved repetitive procedures into validated, logged and consistently executed workflows. PowerShell-driven control can therefore reduce manual variation in provisioning, configuration checking and remediation when permissions, inputs, exceptions and audit records are governed. Yet automation does not eliminate error: flawed logic, excessive privileges or incorrect assumptions can propagate mistakes rapidly across many assets. Secure automation must consequently reduce opportunities for preventable human error while retaining validation, oversight, rollback, accountability, and operational resilience.

1.3 Review Scope, Objectives, and Contributions

This review examines how service-desk automation can strengthen managed IT security operations by reducing avoidable manual variation while preserving control, accountability, and operational resilience. Its scope is organized around three interconnected domains. The first concerns PowerShell-driven provisioning and workflow automation, with attention to repeatable administrative execution, identity and access tasks, endpoint configuration, standardized request fulfilment, and governed orchestration. The second addresses health monitoring, detection, and remediation, emphasizing the movement from reactive support toward continuous visibility, condition assessment, alerting, and controlled corrective action. The third evaluates whether automation produces a measurable reduction in human error through observable changes in configuration accuracy, failed changes, repeat incidents, escalation frequency, resolution time, administrator intervention, and related operational indicators.

The review is necessary because automation is frequently presented as inherently beneficial, although technical feasibility alone does not establish secure or reliable implementation. Accordingly, the analysis considers security safeguards, operational reliability, governance, auditability, privilege control, exception handling, implementation risks, and measurable outcomes alongside automation capability. Particular attention is given to the possibility that poorly designed automation may reproduce flawed assumptions, accelerate incorrect actions, or distribute errors across multiple systems.

The review therefore pursues three objectives: to clarify the operational role of PowerShell within controlled service-desk automation; to examine how monitoring and remediation can support proactive security operations; and to establish how claims of reduced human error can be evaluated empirically. Its analytical progression moves from foundational service-management and risk concepts to provisioning, monitoring, remediation, performance measurement, governance challenges, and future automation priorities.

The principal contribution is an integrated perspective that connects automation engineering with human reliability and security governance. The review does not attempt to provide a PowerShell programming tutorial, benchmark specific commercial service-management platforms, or evaluate every form of cybersecurity automation. Instead, it concentrates on automation where service-desk execution, managed IT security, human oversight, and measurable operational performance intersect. This framing enables automation effectiveness to be judged by operational consequences rather than implementation novelty alone.

2. Foundations of Secure Service Desk Automation

Secure service-desk automation should be understood as a governed socio-technical system in which technology, operational processes, human judgment, and cybersecurity controls function as interdependent elements. Within managed IT security operations, service-desk activities translate organizational policies into routine administrative actions, making the quality of execution central to both service reliability and security assurance. Automation technologies can reduce variability in these activities by converting approved procedures into repeatable, controlled workflows, while PowerShell provides a practical execution layer for automating administrative tasks across compatible enterprise environments.

The value of automation, however, depends on the controls surrounding it. Human-error exposure is shaped not only by individual actions but also by process complexity, workload, inconsistent procedures, excessive privileges, and weak system design. Automation can reduce these risks when it standardizes execution, validates inputs, records actions, and limits unnecessary discretion. Conversely, poorly governed automation may magnify operational risk by reproducing flawed logic at scale. Least privilege therefore becomes a foundational control, ensuring that automated processes possess only the permissions required to perform authorized functions.

Repeatability and auditability further define secure automation. Repeatable workflows reduce inconsistent outcomes, while audit trails establish accountability, support incident investigation, and enable compliance verification. Governance connects these elements by defining authorization, change control, testing, review, exception handling, and accountability. This conceptual framework therefore evaluates service-desk automation not

simply by speed or task completion, but by its capacity to produce secure, consistent, observable, and accountable operations across managed IT environments. It also provides the analytical basis for examining operational context, automation boundaries, PowerShell capability, human-error risk, and governance principles in the sections that follow.

2.1 Managed IT Security Operations Context

Managed IT security operations depend on an architecture that connects user-facing service management with specialized cybersecurity functions. The service desk occupies a critical position within this architecture because routine requests often alter identities, permissions, endpoints, applications, and configuration states that determine the organization's effective attack surface. In outsourced or co-managed environments, these activities may span several clients, tenants, cloud subscriptions, directory domains, and device populations, making coordination and accountability more difficult. Research on managed security service providers further shows that prevention and detection functions involve interdependent responsibilities, incentives, and information asymmetries, reinforcing the need for clearly governed operational boundaries between customer organizations and external security providers (Cezar, Cavusoglu & Raghunathan, 2014).

This environment differs markedly from a conventional help-desk model. Modern service desks administer password resets, account activation and deactivation, group membership, endpoint enrollment, software access, configuration requests, and first-line incident triage, while security operations centers concentrate on threat monitoring, alert investigation, containment, forensic analysis, and higher-order response. Yet these functions increasingly intersect. Systematic research on security operations centers demonstrates that analysts operate within complex monitoring environments where triage, escalation, coordination, and performance measurement are central to operational effectiveness (Agyepong et al., 2020). A seemingly routine access change can therefore become a security event when privileges are assigned incorrectly, stale accounts remain active, or an exception bypasses approved policy.

Identity administration illustrates this convergence clearly. Cloud environments distribute resources across services and administrative boundaries, so authentication, authorization, identity administration, and audit become inseparable from secure service delivery. Identity and access management implemented as a security service can centralize these controls and support consistent authorization across cloud resources (Sharma, Dhote&Potey, 2016). In practice, service-desk workflows must therefore preserve verification, approval, least privilege, and traceability even when requests are operationally routine.

The same logic applies to endpoint and vulnerability management. Large technology estates contain heterogeneous operating systems, applications, firmware, remote devices, and business dependencies, making patching a socio-technical process rather than a simple installation activity. Evidence from the security-patch literature shows that patch decisions involve multiple stakeholders, competing priorities, testing requirements, deployment risks, and organizational dependencies (Dissanayake et al., 2022). Standardized automation becomes important because inconsistent execution across thousands of assets can create uneven protection, undocumented exceptions, and delayed remediation.

Managed environments also face regulatory and contextual constraints. Nigerian research on secure software practice identifies tensions among cybersecurity requirements, regulatory compliance, organizational culture, skills, and collaboration between operational and security teams, illustrating why technical controls require institutional alignment (Ardo, Bass & Gaber, 2023). Comparable African evidence from South Africa shows that cloud-security evaluation must account for organizational controls, risk, trust, and the conditions under which cloud technologies are adopted (Moyo&Loock, 2021). These perspectives are relevant where service providers operate across jurisdictions or customer environments with differing compliance obligations.

Security-sensitive automation must consequently account for integrity and resilience as well as efficiency. Research on secure data exchange in digitally controlled infrastructure demonstrates the importance of maintaining trustworthy, auditable information flows across interconnected operational systems (Shittu et al., 2022). Likewise, cyber-physical resilience research emphasizes that greater connectivity and adaptive automation improve visibility and responsiveness while introducing vulnerabilities, interoperability concerns, and governance requirements (Shittu, Shittu & Olagunju, 2024).

2.2 Service Desk Processes and Automation Boundaries

Service-desk automation should begin with process suitability rather than technological capability. Activities most amenable to full automation are repetitive, high-volume, rules-based, digitally observable, and sufficiently stable to produce predictable outputs. In managed IT environments, this category commonly includes account creation, standardized access changes, software deployment, endpoint configuration, routine diagnostics, and predefined remediation. Process-selection research demonstrates that automation candidates should be evaluated against explicit feasibility and value criteria, because automating an unsuitable process can institutionalize inefficiency rather than remove it (Costa, Mamede& da Silva, 2023).

A second criterion concerns decision complexity. Mature automation is strongest where business rules are unambiguous, structured data are available, system interactions are stable, and exception rates remain limited. Rule-based assessment models therefore distinguish fully deterministic work from processes requiring contextual interpretation or frequent exception handling (Bédard et al., 2024). Similarly, task-selection methods prioritize frequency, periodicity, and duration when identifying routines whose repetitive execution justifies automation (Choi, R'bigui& Cho, 2021).

Service desks nevertheless contain activities that should remain partially automated. Identity exceptions, privileged-access requests, unusual endpoint behavior, major incidents, and changes with substantial business impact require human authorization, interpretation, or escalation. Research on professional service operations shows that technology creates the greatest value when automation of routine work is combined with augmentation of expert judgment rather than indiscriminate substitution (Spring, Faulconbridge & Sarwar, 2022). Human-in-the-loop design, however, must be substantive: experimental evidence shows that simply allowing human intervention does not guarantee better decisions when users over-trust automated recommendations (Sele & Chugunova, 2024).

Accordingly, approval gates, segregation of duties, escalation thresholds, exception queues, and rollback mechanisms should be designed around risk. RPA evaluation research emphasizes that security, architecture, operability, and organizational strategy are integral selection criteria, not secondary implementation concerns (Kim, 2023). South African analysis of administrative automation similarly stresses that efficiency gains must be balanced against governance, institutional memory, accountability, and sustained human oversight (Willie, 2025).

Organizational readiness determines whether these boundaries can be enforced consistently. Comparative evidence involving the United States and lower-resource settings highlights governance maturity, workforce capability, cybersecurity preparedness, interoperability, and ethical accountability as prerequisites for responsible automation adoption (Ayinde, Ojukwu & Ohunyon, 2025). Service-desk processes can therefore be classified into three operational tiers: fully automated tasks with stable rules and low ambiguity; human-supervised workflows where automation executes bounded actions subject to approval or exception handling; and human-retained activities requiring contextual judgment, high-risk authorization, or novel problem solving. This boundary model preserves the efficiency of automation without surrendering accountability. It also creates a defensible basis for applying PowerShell only where execution logic is sufficiently controlled, testable, auditable, reversible, and proportionate to operational risk exposure.

2.3 PowerShell as an Automation Platform

PowerShell is best understood as an object-oriented administrative automation and orchestration environment rather than merely a command-line shell. Its pipeline passes structured .NET objects between commands, enabling administrators to interrogate, transform, and act on system state without repeatedly converting information into unstructured text. This object model, combined with discoverable cmdlets, providers, reusable functions, and extensible modules, supports standardized administration across heterogeneous Windows estates and increasingly cross-platform environments. The PowerShell Gallery further extends this model by distributing modules and scripts that broaden administrative capability beyond the base installation (Lee, 2020).

For service desks, this architecture is important because routine administration can be expressed as reproducible logic. Windows Server, Active Directory, Group Policy, Azure-related services, and remote administration can be addressed through modular command sets, allowing repeated actions to be executed consistently rather than reconstructed manually for each ticket (Monadjemi, 2017). Desired State Configuration extends this principle from task execution toward declarative configuration management, where administrators define intended states and automate their enforcement across managed infrastructure (Chaganti, 2018). In operational practice, remote administration also enables user management, access changes, server control, and status interrogation without requiring physical proximity to managed systems (Nwabueze & Okon, 2019).

PowerShell's value therefore lies in composability. Scheduled execution, APIs, reusable functions, structured output, exception handling, and remote sessions can be combined into workflows that validate inputs, execute approved actions, return machine-readable results, and generate evidence for later review. Logging is particularly significant because PowerShell can interrogate event data across remote systems, improving the speed with which administrators collect operational evidence and support incident analysis (Tanner, 2019). Such capabilities make automation auditable and reduce inconsistency associated with repeated manual execution.

The same power creates material security risk. PowerShell can manipulate identities, groups, privileges, processes, and remote hosts, so inappropriate shell access or excessive administrative rights can expose Active Directory environments to serious compromise (Senturk & Irmak, 2024). Malicious PowerShell scripts also exploit legitimate automation capabilities, making script validation, behavioral monitoring, and detection controls necessary components of secure administration (Alahmadi, Alkhraan & BinSaeed, 2022).

Consequently, secure service-desk use requires execution controls, protected credentials, least-privilege role design, code review, script signing, constrained execution where appropriate, comprehensive logging, and controlled change management. These safeguards should be embedded within broader IT-governance arrangements because automation reduces manual risk only when policies, responsibilities, controls, and accountability remain explicit and enforceable (Sibanda & von Solms, 2019). PowerShell should therefore function as a governed automation layer: powerful enough to standardize service delivery, but bounded sufficiently to prevent administrative convenience from becoming systemic security exposure. This balance is essential for dependable, traceable, and security-conscious managed IT operations.

2.4 Human Error and Operational Risk

Human error in managed IT security is not a single behavioural defect but an emergent operational risk produced by interactions among people, procedures, technology, workload, and organizational controls. Service-desk work is especially exposed because analysts repeatedly shift between tickets, identity systems, endpoint tools, documentation, and security platforms while working against service-level deadlines. Interruptions and task switching can disrupt procedural memory and generate action slips during information-sensitive computer work, making apparently minor omissions consequential when permissions, credentials, or configurations are involved (Williams et al., 2020).

Risk also increases when operational procedures are inconsistently understood or applied. Nigerian cybersecurity research shows that awareness, policy, social norms, trust, and broader security culture shape how individuals interpret and perform protective behaviour, indicating that errors cannot be separated from their organizational context (Garba, Kaur & Ibrahim, 2023). In service-desk environments, inadequate documentation, ambiguous approval paths, manual data entry, and privilege complexity may therefore transform routine work into opportunities for misconfiguration, unauthorized access, delayed escalation, or incomplete audit evidence.

Human reliability is similarly influenced by competence, autonomy, motivation, and the institutional environment in which security rules are enacted. South African evidence demonstrates that information-security behaviour is associated with psychological and organizational factors rather than compliance being an automatic consequence of policy availability (Gangire, Da Veiga & Herselman, 2021). This supports a socio-technical interpretation of operational risk: configuration drift or access-control weakness may originate in an individual action, yet its probability is conditioned by workload design, tooling, supervision, process clarity, and control architecture.

A rigorous risk perspective must also distinguish proximate mistakes from underlying causes. Root-cause analysis emphasizes that failures frequently arise through interacting technical, procedural, and organizational conditions rather than isolated events, making systematic diagnosis essential for preventing recurrence (Akanbi, Ganiu & Sunday, 2023). In managed IT, one incorrect privilege assignment can propagate into excessive access, while an erroneous configuration template can affect many endpoints and generate service or compliance consequences.

Automation changes this risk profile rather than eliminating it. A human-factors approach to cybersecurity shows that resilient controls require coordinated attention to users, organizational processes, and technical systems instead of treating people as the weakest link in isolation (Pollini et al., 2022). PowerShell-based automation can reduce repetitive manual variation through validated inputs, standardized execution, logging, and predictable error handling. However, flawed logic, excessive permissions, inadequate testing, or poorly bounded scope can reproduce one design mistake across numerous assets. Accordingly, operational controls should be designed to contain both human mistakes and automation-induced failure propagation across environments. Secure automation must therefore combine human oversight with validation, least privilege, rollback capability, auditable execution, and clearly defined escalation boundaries.

2.5 Automation Governance and Security Principles

Automation governance in managed IT security operations should define not only what service-desk technologies are permitted to do, but also who authorizes them, how their behavior is verified, and how accountability is preserved when execution is delegated to code. Governance therefore converts automation from an administrator convenience into an operational capability. A mature framework links policy, technical safeguards, responsibility, oversight, and organizational culture so that automated actions remain aligned with business and security objectives (Da Veiga & Eloff, 2007).

For PowerShell-driven service desks, the central security principle is least privilege. Scripts, service accounts, scheduled tasks, and remote sessions should receive only the permissions necessary for their defined function, with privileged activities separated from routine operations. Attribute-based access-control research demonstrates the value of granting access according to explicit attributes and authorized roles rather than broad, persistent entitlement, reinforcing the need for granular authorization in automated workflows (Shittu & Nabil, 2023). Segregation of duties should complement this model by separating script development, approval, deployment, and high-risk execution where practical.

Governance must also make automation change-controlled and auditable. Scripts should be versioned, peer-reviewed, tested in test environments, approved before production use, and supported by rollback procedures. Execution logs, input records, timestamps, identities, exceptions, and resulting state changes should provide evidence sufficient for incident investigation and compliance review. Nigerian evidence on institutional information-security policies underscores the importance of policy adoption, management commitment, awareness, and readiness in translating security expectations into consistent organizational practice (Ajibesin, Maharazu & Ogundapo, 2022). Without such institutionalization, technically sound controls may be bypassed through informal administrative routines.

Secure automation also requires accountability for exceptions. Failed validations, unexpected system states, privilege conflicts, and incomplete remediation should trigger bounded failure modes and human escalation rather than uncontrolled continuation. Cyber-governance literature emphasizes that effective cybersecurity extends beyond technical management toward coordinated participation, defined responsibilities, strategic oversight, and stakeholder accountability (Savaş & Karataş, 2022). These principles are especially important where service-desk automation spans multiple tenants, identities, endpoints, or compliance regimes.

Ethical leadership and policy implementation further shape whether governance controls remain operational rather than symbolic. Evidence from the Southern African Development Community context highlights how leadership quality affects technological transformation and policy execution, emphasizing that governance depends on institutional commitment as well as formal rules (Enaifoghe, Dlamini & Agwuna, 2020). Accordingly, secure service-desk automation should combine least privilege, controlled authorization, code integrity, credential protection, testing, change management, logging, periodic review, segregation of duties, and accountable exception handling. These principles create the conditions under which PowerShell automation can improve consistency and reduce preventable errors without converting speed and scale into amplified operational or security risk.

3. PowerShell-Driven Provisioning and Workflow Automation

PowerShell-driven provisioning is treated as a controlled service-delivery mechanism rather than ad hoc administrative scripting. In managed IT environments, repetitive requests for account creation, access modification, endpoint preparation, software deployment, and user offboarding can become repeatable workflows with defined inputs, approvals, execution states, and outcomes. Service-provisioning research has shown that standardized, high-volume requests are amenable to automation when process steps and supporting resources are aligned (Brenner et al., 2008).

Within identity administration, automation can coordinate account creation, modification, suspension, and deletion across an employee or customer lifecycle. Provisioning frameworks emphasize policy-driven workflows, delegated administration, approval routing, and transaction logging, which directly support service desks handling joiner, mover, and leaver events (Glazer, Robinson & Hamlin, 2022). PowerShell can operationalize these workflows through reusable functions applying validated attributes, assigning approved groups or roles, invoking directory or cloud-management interfaces, and returning structured results to the originating service-management process.

The same orchestration logic extends to ticket-triggered actions. Rule-based ITSM research demonstrates that workflow engines can evaluate request conditions, route approvals, initiate automated activities, and escalate unsafe or exceptional cases rather than executing every change indiscriminately (Kolla, 2021). In this architecture, PowerShell serves as an execution layer beneath governance logic: a ticket or approved service request can trigger a bounded function, while completion status, errors, and resulting state changes are returned for audit and follow-up.

Consistency is an advantage over manual administration. Industrial automation research illustrates how integrated engineering environments standardize process logic, reduce operator variability, and improve repeatability through coordinated execution (Sunday & Omoegun, 2022). Applied to service desks, the analogous objective is not industrial control but deterministic administration: identical approved requests should produce equivalent configuration outcomes unless documented environmental differences require exception handling.

Automation also supports endpoint configuration and software deployment across distributed estates. Research on automated deployment shows that heterogeneous, dynamic systems require explicit management of dependencies, target-state constraints, deployment sequences, and recovery conditions (Arcangeli, Boujbel&Leriche, 2015). PowerShell can contribute by interrogating endpoint state before execution, applying packages or settings conditionally, verifying post-change results, and emitting structured evidence that distinguishes success, partial completion, and failure.

Configuration management further demonstrates why reusable automation should be version-controlled and testable. Automated configuration approaches improve deployment consistency and speed by reducing repeated manual intervention while enabling rollback and reproducible states across environments (Akerle et al., 2024). For service desks, this implies maintaining reviewed functions and modules rather than permitting uncontrolled script fragments to evolve independently across administrators.

Workflow optimization research similarly shows that coordinated automation yields greater value when individual actions are integrated around system-wide objectives rather than optimized in isolation (Akanbi& Sunday, 2024). PowerShell-based provisioning should therefore connect identity, endpoint, application, and ticketing activities through explicit dependencies, approval gates, and outcome checks during onboarding and offboarding, where multiple systems must change coherently.

Automation can also reduce clerical error by replacing repeated manual identification and data-handling steps with machine-mediated transactions. Nigerian research on RFID-enabled library management demonstrates how automated identification can reduce manual involvement, accelerate access to information, and improve process efficiency (Yusuf et al., 2023). In service-desk contexts, comparable benefits depend on authoritative source data, validated parameters, and rejection of incomplete or inconsistent inputs before privileged actions occur.

These efficiencies require strong safeguards. Identity-lifecycle research emphasizes that access-management systems must reconcile intended policy with actual enforcement throughout changing user relationships and privileges (Gunter, Liebovitz&Malin, 2011). PowerShell workflows should therefore incorporate least-privilege execution, protected credentials, input validation, code review, staged testing, signed or otherwise controlled scripts, comprehensive logging, approval evidence, exception capture, and rollback procedures. When integrated with ITSM controls, these features make automated provisioning faster without equating speed with correctness; the objective is traceable, policy-consistent service delivery in which automation narrows preventable variation while preserving human authorization for consequential or ambiguous changes.

4. Health Monitoring, Detection, and Automated Remediation

Health monitoring within managed IT security operations should function as a closed control loop rather than a passive collection of telemetry. PowerShell can interrogate service states, endpoint configuration, event logs, resource utilization, patch posture, account conditions, and security controls, converting heterogeneous operational signals into structured observations that can be evaluated against expected states. Sensor-based fault research demonstrates that continuous observation becomes operationally valuable only when measurements are interpreted against patterns capable of identifying abnormal conditions early (Sunday & Omoegun, 2022).

The progression from visibility to detection requires context. Raw logs, counters, and status checks may reveal symptoms without establishing whether intervention is justified. Predictive-maintenance literature shows the value of moving beyond reactive response toward condition-based assessment that anticipates degradation before failure becomes service-affecting (Sunday et al., 2020). In service-desk environments, scheduled PowerShell health

checks can similarly compare current states with baselines, identify configuration drift, validate service availability, and raise alerts when thresholds or policy conditions are breached.

Digital-twin research reinforces the importance of combining real-time data with modeled expectations to improve fault localization and predictive awareness (Shittu et al., 2023). Condition-monitoring studies likewise demonstrate that diagnostic value depends on transforming continuous signals into interpretable indicators rather than accumulating data without decision logic (Omogun et al., 2023).

Table 1: PowerShell-Enabled Health Monitoring, Detection, Remediation, and Verification Workflow

Operational stage	PowerShell-oriented activity	Required control
Visibility	Query services, logs, endpoints, resources	Complete, trustworthy telemetry
Detection	Compare state against thresholds or baselines	False-positive filtering
Decision	Classify severity and remediation eligibility	Approval and escalation rules
Remediation	Execute bounded corrective action	Least privilege and rollback
Verification	Recheck state and record outcome	Logging and audit evidence

Automated remediation extends monitoring into controlled action. Autonomic-computing principles conceptualize this progression as monitoring, analysis, planning, and execution governed by administrator-defined objectives (Kephart & Chess, 2003). At scale, intelligent healing systems show that repair selection based on monitoring evidence can reduce unavailability, but inaccurate policies or models can select inappropriate actions and propagate failure rapidly (Li et al., 2021).

PowerShell remediation should therefore be bounded by risk. Low-impact actions such as restarting an approved service or reapplying a known configuration may be automated after validation, whereas identity changes, widespread endpoint modifications, ambiguous security alerts, or repeated failures should trigger human authorization. Incident-management research from South Africa similarly emphasizes defined authority, coordinated response procedures, and structured escalation as prerequisites for dependable security response (Pretorius & Ngejane, 2019). Testing, privilege controls, timeout conditions, rollback, loop prevention, post-action verification, and comprehensive logging are therefore essential. Operational resilience is achieved not by maximizing autonomous action, but by shortening detection and recovery times while preserving evidence, reversibility, human control over consequential remediation, and measurable accountability.

5. Measuring Human Error and Operational Performance

Service-desk automation should be evaluated through a measurement architecture that separates activity volume from genuine improvements in security, reliability, and human performance. A credible assessment begins with a pre-automation baseline covering ticket demand, provisioning accuracy, change failures, configuration errors, mean resolution time, escalation frequency, downtime, administrator intervention, and repeat incidents. International ITSM research shows that benefits are often claimed more readily than demonstrated through coherent measurement frameworks, making linkage between objectives, indicators, and outcomes essential (Gacenga, Cater-Steel & Toleman, 2010).

Baseline measures should then be normalized against workload, asset population, user numbers, or transaction volume. Raw ticket throughput may rise simply because demand increases, while shorter resolution time may conceal poorer resolution quality. ITIL-based measurement research therefore supports structured process metrics that connect service-support activities with defined performance objectives rather than relying on isolated counts (Lahtela, Jäntti&Kaukola, 2010). In PowerShell-enabled provisioning, measures should include execution-success rate, provisioning accuracy, rollback frequency, failed-change rate, configuration-deviation rate, and percentage of tasks requiring manual correction.

Statistical process control offers a complementary means of distinguishing routine variation from meaningful deterioration or improvement. Control limits, trend analysis, and process-capability thinking can reveal whether post-automation performance becomes more stable rather than merely faster (Akanbi, Ganiu& Sunday, 2022b). Workforce-oriented performance analysis is equally important because throughput gains can coexist with fatigue, uneven workload, or shifting human intervention patterns; measurement should incorporate workload distribution and administrator dependency alongside output volume (Akanbi, Ganiu& Sunday, 2022a).

Table 2: Framework for Measuring Service-Desk Automation Performance and Human-Error Reduction

Measurement dimension	Illustrative indicators	Analytical interpretation
Provisioning quality	Accuracy, failed changes, rollbacks	Execution correctness
Automation performance	Success rate, manual intervention	Workflow reliability
Service efficiency	Resolution time, throughput, escalations	Operational productivity
Security performance	MTTD, MTTR, access errors	Security responsiveness

Measurement dimension	Illustrative indicators	Analytical interpretation
Reliability	Downtime, repeat incidents, configuration drift	Service stability
Compliance	Deviations, audit exceptions	Governance effectiveness
Human-error exposure	Manual corrections, repeated mistakes	Preventable variation

For managed IT security, reliability indicators should be interpreted alongside organizational consequences. South African evidence on outsourced IT operations shows that service-provider performance, response time, service quality, and process alignment affect organizational performance, showing why speed alone is insufficient (Mtsweni, Mokwena & Moeti, 2021). Measurement should therefore include mean time to detection, mean time to remediation, repeat incidents, compliance deviations, unauthorized-access errors, and downtime.

Human-error reduction requires evidence that preventable variation decreases after automation. Metrics should compare equivalent pre- and post-automation periods, use consistent definitions, and examine trends longitudinally. Process-assessment research further supports transparent, repeatable metrics for evaluating whether service-management objectives are actually achieved (Asrowardi et al., 2018). Workforce and operational indicators should also be interpreted together because productivity changes may reflect staffing, retention, experience, or workload rather than automation alone (Falemi, Akhigbe & Akin-Oluoyomi, 2023).

Accordingly, causal claims should control for ticket mix, infrastructure changes, staffing levels, seasonal demand, policy changes, and concurrent security initiatives. Audit logs, change records, PowerShell execution histories, incident records, and rollback evidence provide stronger attribution than perception surveys. Reduced human error is empirically demonstrated when normalized error rates decline persistently without deterioration in security, service quality, or reliability.

6. Security Challenges and Future Automation Priorities

Service-desk automation introduces a paradox: the mechanisms designed to reduce repetitive human error can also concentrate privilege, accelerate faulty execution, and expand the blast radius of a single defect. PowerShell is particularly consequential because the same administrative reach that enables efficient provisioning and remediation can be abused for credential access, lateral movement, persistence, and obfuscated execution. Future service-desk architectures should therefore treat automation code, execution identities, modules, and orchestration pipelines as security-sensitive assets subject to continuous validation rather than assuming that trusted administrative tools are intrinsically safe.

A first challenge is automation scope. Scripts that begin as narrow task aids can evolve into interconnected workflows spanning identity, endpoints, cloud services, ticketing platforms, and monitoring systems. As dependencies grow, a faulty assumption or malformed input can propagate across environments before operators recognize the error. Zero-trust research consequently emphasizes continuous verification, granular authorization, visibility, and policy-driven orchestration instead of persistent trust in users, devices, or automated processes (Cao et al., 2024). For PowerShell-driven operations, this implies bounded permissions, explicit execution context, validated parameters, and containment mechanisms capable of stopping automation when system state diverges from expectation.

The transition toward AI-assisted operations adds further uncertainty. AIOps can improve incident classification, assignment, prioritization, and resolution by learning from large operational datasets, but automated recommendations inherit limitations from training data, labels, system dependencies, and incomplete contextual information. Industrial evidence shows that knowledge-based incident automation can reduce resolution effort, yet performance remains dependent on data quality and the fidelity with which operational knowledge is represented (Ahmed et al., 2023). Future service desks should therefore combine machine-generated recommendations with confidence thresholds, escalation rules, explainability, and post-action verification.

Distributed intelligent systems illustrate both the promise and the governance difficulty of increasingly autonomous coordination. Multi-agent architectures can improve adaptability and decentralized decision-making, but they introduce emergent interactions that are harder to predict, test, and audit than deterministic workflows (Akanbi, Ganiu & Sunday, 2025). Similarly, reviews of AI-driven autonomous mobile systems identify scalability, interoperability, safety, and human-machine coordination as continuing research priorities when automation moves from isolated tasks toward coordinated autonomy (Akanbi & Sunday, 2025). For managed IT operations, the analogous priority is controlled autonomy: systems should be permitted to act independently only where the action space, failure modes, and recovery mechanisms are well characterized.

Predictive and model-based automation will also become more prominent. AI-enabled digital twins demonstrate how live telemetry, simulation, and predictive models can anticipate failures and optimize intervention before disruption occurs (Shittu & Shittu, 2024). In service-desk environments, comparable approaches could support configuration-state modeling, predictive incident detection, capacity forecasting, and pre-emptive remediation. However, model drift, incomplete telemetry, adversarial manipulation, and opaque inference can convert predictive advantage into false assurance. Likewise, IoT-enabled distributed control research shows that resilience increasingly depends on integrating monitoring, communication, automation, and adaptive response across interconnected assets (Shittu et al., 2025).

Governance must mature alongside these capabilities. Nigerian research on digital-technology and AI policy identifies persistent gaps in ethical guidance, data protection, and mechanisms for addressing security harms, underscoring the danger of technological adoption outpacing governance

(Alakitan&Makinde, 2025). South African research similarly identifies transparency, privacy, machine bias, responsible AI, stakeholder accountability, and continuous monitoring as central elements of ethical AI governance (Coovadia et al., 2025).

7. Conclusion

The review demonstrates that secure service-desk automation is most effective when technical capability, operational governance, and human oversight are treated as an integrated system rather than as separate concerns. The first objective was achieved by clarifying the role of PowerShell as a controlled automation platform for provisioning, workflow execution, endpoint administration, configuration interrogation, and auditable task orchestration. The analysis showed that its value lies not merely in accelerating repetitive actions, but in improving consistency, repeatability, traceability, and policy enforcement across managed IT environments.

The second objective was met through examination of health monitoring, detection, and automated remediation. The findings indicate that meaningful automation requires a progression from visibility to detection, decision, corrective action, and verification. Automated health checks and bounded remediation can shorten response latency and reduce unnecessary administrator intervention, but only where telemetry is reliable, escalation thresholds are explicit, privileges are constrained, and rollback mechanisms are available. Otherwise, automation may amplify misconfiguration, privilege errors, or flawed logic across multiple systems.

The third objective concerned measurable reduction of human error. The review establishes that such claims must be supported by baseline data, normalized indicators, longitudinal comparison, and auditable operational evidence. Metrics including provisioning accuracy, failed-change rates, configuration deviations, repeat incidents, mean resolution time, mean time to detection, mean time to remediation, rollback frequency, and manual intervention provide stronger evidence than productivity counts alone.

Overall, the study concludes that automation improves security and service reliability when it is governed, observable, testable, reversible, and aligned with least-privilege principles. Managed IT organizations should institutionalize code review, validation, version control, protected credentials, approval gates, audit logging, exception handling, and human oversight for high-impact decisions. Future research should prioritize benchmark datasets, longitudinal evaluations, adversarial testing, explainable AIOps, and policy-as-code approaches that permit greater automation without weakening accountability, resilience, or operational control. This balance remains essential as intelligent orchestration becomes autonomous and interconnected.

References

1. Adamolekun, A., Logah, F.X., Alabi, C., Baanye, J., Wilson, M., Ganiu, O., Seong, Y. and Yi, S. (2025) 'Toward a unified framework for multimodal cognitive workload assessment in human-machine interaction systems', SSRN Electronic Journal, SSRN 5235980. DOI: [10.2139/ssrn.5235980](https://doi.org/10.2139/ssrn.5235980).
2. Agyepong, E., Cherdantseva, Y., Reinecke, P. and Burnap, P. (2020) 'Challenges and performance metrics for security operations center analysts: A systematic review', Journal of Cyber Security Technology, 4(3), pp. 125–152. DOI: [10.1080/23742917.2019.1698178](https://doi.org/10.1080/23742917.2019.1698178).
3. Ahmed, S., Singh, M., Doherty, B., Ramlan, E., Harkin, K., Bucholc, M. and Coyle, D. (2023) 'Knowledge-based intelligent system for IT incident DevOps', Proceedings of the 2023 IEEE/ACM International Workshop on Cloud Intelligence and AIOps, pp. 1–7. DOI: [10.1109/AIOps59134.2023.00005](https://doi.org/10.1109/AIOps59134.2023.00005).
4. Ajibesin, A.A., Maharazu, K. and Ogundapo, O. (2022) 'Information security policies in Nigerian institutions: Evaluation and readiness', International Journal of Risk and Contingency Management, 11(1), pp. 1–22. DOI: [10.4018/IJRCM.303103](https://doi.org/10.4018/IJRCM.303103).
5. Ajufo, G.K. and Qutieshat, A. (2023) 'An examination of the human factors in cybersecurity: Future direction for Nigerian banks', Indonesian Journal of Information Systems, 6(1), pp. 1–16. DOI: [10.24002/ijis.v6i1.7278](https://doi.org/10.24002/ijis.v6i1.7278).
6. Akanbi, O. and Sunday, E.A. (2024) 'An integrated path-planning and slotting optimization model for AMR-enabled high-density warehousing', International Journal of Advanced Multidisciplinary Research and Studies, 4(6), pp. 3226–3243. DOI: [10.62225/2583049X.2024.4.6.6181](https://doi.org/10.62225/2583049X.2024.4.6.6181).
7. Akanbi, O. and Sunday, E.A. (2025) 'A systematic review of AI-driven autonomous mobile robots (AMR) in scalable micro-fulfilment centers', International Journal of Advanced Multidisciplinary Research and Studies, 5(6), pp. 2363–2379. DOI: [10.62225/2583049X.2025.5.6.6182](https://doi.org/10.62225/2583049X.2025.5.6.6182).
8. Akanbi, O., Ganiu, O.S. and Sunday, E.A. (2022a) 'A workforce-centric line-balancing model for throughput maximization in multi-shift operations', Shodhshauryam: International Scientific Refereed Research Journal, 5(3), pp. 439–464.
9. Akanbi, O., Ganiu, O.S. and Sunday, E.A. (2022b) 'Comprehensive review of statistical process control (SPC) for precision weight management in global food manufacturing', Gyanshauryam: International Scientific Refereed Research Journal, 5(3), pp. 433–461.
10. Akanbi, O., Ganiu, O.S. and Sunday, E.A. (2023) 'A review of industrial equipment restoration strategies: Integrating design of experiments (DOE) and root-cause analysis', International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 9(10), pp. 444–462. DOI: [10.32628/CSEIT2361074](https://doi.org/10.32628/CSEIT2361074).

11. Akanbi, O., Ganiu, O.S. and Sunday, E.A. (2025) 'A multi-agent AI framework for swarm intelligence in autonomous mobile robot (AMR) fleet coordination', *International Journal of Scientific Research in Humanities and Social Sciences*, 2(1), pp. 81–109.
12. Akerele, J.I., Uzoka, A., Ojukwu, P.U. and Olamijuwon, O.J. (2024) 'Increasing software deployment speed in agile environments through automated configuration management', *International Journal of Engineering Research Updates*, 7(2), pp. 28–35. DOI: [10.53430/ijeru.2024.7.2.0047](https://doi.org/10.53430/ijeru.2024.7.2.0047).
13. Alahmadi, A., Alkhraan, N. and BinSaeedan, W. (2022) 'MPSAutodetect: A malicious PowerShell script detection model based on stacked denoising auto-encoder', *Computers & Security*, 116, Article 102658. DOI: [10.1016/j.cose.2022.102658](https://doi.org/10.1016/j.cose.2022.102658).
14. Alakitan, M. and Makinde, E. (2025) 'Where are the ethical guidelines? Examining the governance of digital technologies and AI in Nigeria', *Policy & Internet*, 17(1), e416. DOI: [10.1002/poi3.416](https://doi.org/10.1002/poi3.416).
15. Arcangeli, J.-P., Boujbel, R. and Leriche, S. (2015) 'Automatic deployment of distributed software systems: Definitions and state of the art', *Journal of Systems and Software*, 103, pp. 198–218. DOI: [10.1016/j.jss.2015.01.040](https://doi.org/10.1016/j.jss.2015.01.040).
16. Ardo, A.A., Bass, J.M. and Gaber, T. (2023) 'Implications of regulatory policy for building secure agile software in Nigeria: A grounded theory', *The Electronic Journal of Information Systems in Developing Countries*, 89(6), e12285. DOI: [10.1002/isd2.12285](https://doi.org/10.1002/isd2.12285).
17. Asrowardi, I., Putra, S.D., Subyantoro, E. and Daud, N.H.M. (2018) 'IT service management system measurement using ISO20000-1 and ISO15504-8: Developing a solution-mediated process assessment tool to enable transparent and SMS process assessment', *International Journal of Electrical and Computer Engineering*, 8(5), pp. 4023–4032. DOI: [10.11591/ijece.v8i5.pp4023-4032](https://doi.org/10.11591/ijece.v8i5.pp4023-4032).
18. Ayachitula, N., Bucu, M.J., Diao, Y., Maheswaran, S., Pavuluri, R., Shwartz, L. and Ward, C. (2007) 'IT service management automation: A hybrid methodology to integrate and orchestrate collaborative human-centric and automation-centric workflows', 2007 IEEE International Conference on Services Computing (SCC 2007), pp. 574–581. DOI: [10.1109/SCC.2007.75](https://doi.org/10.1109/SCC.2007.75). (DBLP)
19. Ayinde, M., Ojukwu, P.U. and Ohunyon, G. (2025) 'Organizational readiness for generative AI integration in healthcare operations: Comparative management capabilities between the U.S. and low- and middle-income countries', *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), pp. 2207–2222. DOI: [10.62225/2583049X.2025.5.6.6007](https://doi.org/10.62225/2583049X.2025.5.6.6007).
20. Bédard, M., Leshob, A., Benzarti, I., Mili, H., Rab, R. and Hussain, O. (2024) 'A rule-based method to effectively adopt robotic process automation', *Journal of Software: Evolution and Process*, 36(11), e2709. DOI: [10.1002/smr.2709](https://doi.org/10.1002/smr.2709).
21. Brenner, M., Dreorodosek, G., Hanemann, A., Hegering, H.-G. and König, R. (2008) 'Service provisioning: Challenges, process alignment and tool support', in Bergstra, J.A. and Burgess, M. (eds.) *Handbook of Network and System Administration*. Amsterdam: Elsevier, pp. 855–904. DOI: [10.1016/B978-044452198-9.50033-1](https://doi.org/10.1016/B978-044452198-9.50033-1).
22. Cao, Y., Pokhrel, S.R., Zhu, Y., Doss, R. and Li, G. (2024) 'Automation and orchestration of zero trust architecture: Potential solutions and challenges', *Machine Intelligence Research*, 21, pp. 294–317. DOI: [10.1007/s11633-023-1456-2](https://doi.org/10.1007/s11633-023-1456-2).
23. Cezar, A., Cavusoglu, H. and Raghunathan, S. (2014) 'Outsourcing information security: Contracting issues and security implications', *Management Science*, 60(3), pp. 638–657. DOI: [10.1287/mnsc.2013.1763](https://doi.org/10.1287/mnsc.2013.1763).
24. Chaganti, R. (2018) *Pro PowerShell Desired State Configuration: An In-Depth Guide to Windows PowerShell DSC*. 2nd edn. Berkeley, CA: Apress. DOI: [10.1007/978-1-4842-3483-9](https://doi.org/10.1007/978-1-4842-3483-9).
25. Choi, D., R'bigui, H. and Cho, C. (2021) 'Candidate digital tasks selection methodology for automation with robotic process automation', *Sustainability*, 13(16), Article 8980. DOI: [10.3390/su13168980](https://doi.org/10.3390/su13168980).
26. Coovadia, H., Marx, B., Botha, I. and Gold, N.O. (2025) 'Building an ethical artificial intelligence corporate governance framework for the integration of emerging technologies into business processes', *South African Journal of Accounting Research*, 39(3), pp. 286–316. DOI: [10.1080/10291954.2025.2523661](https://doi.org/10.1080/10291954.2025.2523661).
27. Costa, D.S., Mamede, H.S. and da Silva, M.M. (2023) 'A method for selecting processes for automation with AHP and TOPSIS', *Heliyon*, 9(3), e13683. DOI: [10.1016/j.heliyon.2023.e13683](https://doi.org/10.1016/j.heliyon.2023.e13683).
28. Da Veiga, A. and Eloff, J.H.P. (2007) 'An information security governance framework', *Information Systems Management*, 24(4), pp. 361–372. DOI: [10.1080/10580530701586136](https://doi.org/10.1080/10580530701586136).
29. Dissanayake, N., Jayatilaka, A., Zahedi, M. and Babar, M.A. (2022) 'Software security patch management: A systematic literature review of challenges, approaches, tools and practices', *Information and Software Technology*, 144, Article 106771. DOI: [10.1016/j.infsof.2021.106771](https://doi.org/10.1016/j.infsof.2021.106771).
30. Enaifoghe, A., Dlamini, N.P. and Agwuna, L.U. (2020) 'African economic integration development in technological transformation: Assessing the importance of ethical leadership for policy implementation in SADC', *Journal of Economics and Behavioral Studies*, 12(6), pp. 11–20. DOI: [10.22610/jebs.v12i6\(J\).3102](https://doi.org/10.22610/jebs.v12i6(J).3102).

31. Falemi, A., Akhigbe, R. and Akin-Oluyomi, O.T. (2023) 'A proposed framework for integrating employee retention systems with operational productivity metrics', *International Journal of Advanced Multidisciplinary Research and Studies*, 3(1), pp. 1704–1724. DOI: [10.62225/2583049X.2023.3.1.5339](https://doi.org/10.62225/2583049X.2023.3.1.5339).
32. Gacenga, F., Cater-Steel, A. and Toleman, M. (2010) 'An international analysis of IT service management benefits and performance measurement', *Journal of Global Information Technology Management*, 13(4), pp. 28–63. DOI: [10.1080/1097198X.2010.10856525](https://doi.org/10.1080/1097198X.2010.10856525).
33. Gangire, Y., Da Veiga, A. and Herselman, M. (2021) 'Assessing information security behaviour: A self-determination theory perspective', *Information & Computer Security*, 29(4), pp. 625–646. DOI: [10.1108/ICS-11-2020-0179](https://doi.org/10.1108/ICS-11-2020-0179).
34. Garba, J., Kaur, J. and Ibrahim, E.N.M. (2023) 'Design of a conceptual framework for cybersecurity culture amongst online banking users in Nigeria', *Nigerian Journal of Technology*, 42(3), pp. 399–405. DOI: [10.4314/njt.v42i3.13](https://doi.org/10.4314/njt.v42i3.13).
35. Glazer, I., Robinson, L. and Hamlin, M. (2022) 'User provisioning in the enterprise', *IDPro Body of Knowledge*, 1(8). DOI: [10.55621/idpro.84](https://doi.org/10.55621/idpro.84).
36. Grandhi, S.R. and Still, J.D. (2025) 'Deciphering human error: Improving cybersecurity reporting', *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 69(1), pp. 234–239. DOI: [10.1177/10711813251358790](https://doi.org/10.1177/10711813251358790).
37. Gunter, C.A., Liebovitz, D. and Malin, B. (2011) 'Experience-based access management: A life-cycle framework for identity and access management systems', *IEEE Security & Privacy*, 9(5), pp. 48–55. DOI: [10.1109/MSP.2011.72](https://doi.org/10.1109/MSP.2011.72).
38. Jäntti, M. and Cater-Steel, A. (2017) 'Proactive management of IT operations to improve IT services', *Journal of Information Systems and Technology Management*, 14(2), pp. 191–218. DOI: [10.4301/S1807-17752017000200004](https://doi.org/10.4301/S1807-17752017000200004). (DOI)
39. Kephart, J.O. and Chess, D.M. (2003) 'The vision of autonomic computing', *Computer*, 36(1), pp. 41–50. DOI: [10.1109/MC.2003.1160055](https://doi.org/10.1109/MC.2003.1160055).
40. Kim, S.-H. (2023) 'Development of evaluation criteria for robotic process automation (RPA) solution selection', *Electronics*, 12(4), Article 986. DOI: [10.3390/electronics12040986](https://doi.org/10.3390/electronics12040986).
41. Kolla, S.H. (2021) 'Rule-based automation for IT service management workflows', *Online Journal of Engineering Sciences*, 1(1), pp. 1–14. DOI: [10.31586/ojes.2020.1360](https://doi.org/10.31586/ojes.2020.1360).
42. Kraemer, S. and Carayon, P. (2007) 'Human errors and violations in computer and information security: The viewpoint of network administrators and security specialists', *Applied Ergonomics*, 38(2), pp. 143–154. DOI: [10.1016/j.apergo.2006.03.010](https://doi.org/10.1016/j.apergo.2006.03.010).
43. Krishnan, G. and Ravindran, V. (2017) 'IT service management automation and its impact on the IT industry', 2017 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, pp. 1–4. DOI: [10.1109/ICCIDS.2017.8272633](https://doi.org/10.1109/ICCIDS.2017.8272633). (Redcol)
44. Ladapo, O.O., Jooda, D., Dosunmu, A.A. and Abolaji, T.O. (2023) 'A comprehensive survey on ServiceNow for IT service management', *International Journal of Multidisciplinary Research and Growth Evaluation*, 4(6), pp. 1532–1545. DOI: [10.54660/IJMRGE.2023.4.6.1532-1545](https://doi.org/10.54660/IJMRGE.2023.4.6.1532-1545).
45. Lahtela, A., Jäntti, M. and Kaukola, J. (2010) 'Implementing an ITIL-based IT service management measurement system', 2010 Fourth International Conference on Digital Society, pp. 249–254. DOI: [10.1109/ICDS.2010.48](https://doi.org/10.1109/ICDS.2010.48).
46. Lee, T. (2020) 'Setting up a PowerShell 7 environment', in *PowerShell 7 for IT Pros: A Guide to Using PowerShell 7 to Manage Windows Systems*. Hoboken, NJ: Wiley. DOI: [10.1002/9781119644750.ch1](https://doi.org/10.1002/9781119644750.ch1).
47. Li, R., Cheng, Z., Lee, P.P.C., Wang, P., Qiang, Y., Lan, L., He, C., Lu, J., Wang, M. and Ding, X. (2021) 'Automated intelligent healing in cloud-scale data centers', *Proceedings of the 2021 40th International Symposium on Reliable Distributed Systems (SRDS)*, pp. 244–253. DOI: [10.1109/SRDS53918.2021.00032](https://doi.org/10.1109/SRDS53918.2021.00032).
48. Marrone, M., Gacenga, F., Cater-Steel, A. and Kolbe, L. (2014) 'IT service management: A cross-national study of ITIL adoption', *Communications of the Association for Information Systems*, 34(1), pp. 865–892. DOI: [10.17705/ICAIS.03449](https://doi.org/10.17705/ICAIS.03449). (Macquarie University)
49. Monadjemi, P. (2017) *Windows Server-Administration mit PowerShell 5.1: Eine kompakte und praxisorientierte Einführung*. Wiesbaden: Springer Vieweg. DOI: [10.1007/978-3-658-17666-2](https://doi.org/10.1007/978-3-658-17666-2).
50. Moyo, M. and Loock, M. (2021) 'Conceptualising a cloud business intelligence security evaluation framework for small and medium enterprises in small towns of the Limpopo Province, South Africa', *Information*, 12(3), Article 128. DOI: [10.3390/info12030128](https://doi.org/10.3390/info12030128).
51. Mtsweni, P., Mokwena, S.N. and Moeti, M.N. (2021) 'The impact of outsourcing information technology services on business operations', *South African Journal of Information Management*, 23(1), a1361. DOI: [10.4102/sajim.v23i1.1361](https://doi.org/10.4102/sajim.v23i1.1361).

52. Mvelase, N. and Smith, R. (2024) 'IT service management best practices in insurance industries', in Moloi, T. and George, B. (eds.) *Towards Digitally Transforming Accounting and Business Processes: Proceedings of the International Conference of Accounting and Business iCAB, Johannesburg 2023*. Cham: Springer Nature, pp. 71–91. DOI: [10.1007/978-3-031-46177-4_5](https://doi.org/10.1007/978-3-031-46177-4_5). ([University of Johannesburg](#))
53. Ngandu, M.R., Mwansa, G. and Mkabe, Z. (2025) 'Strengthening cybersecurity in a government department by addressing password management challenges and human factor vulnerabilities', *Discover Computing*, 28, Article 148. DOI: [10.1007/s10791-025-09659-2](https://doi.org/10.1007/s10791-025-09659-2).
54. Nwabueze, E.E. and Okon, E.E. (2019) 'Improved smartphone application for remote access by network administrators', *Journal of Information Security*, 10, pp. 250–262. DOI: [10.4236/jis.2019.104014](https://doi.org/10.4236/jis.2019.104014).
55. Omoegun, G.O., Sunday, E.A., Essien, M.A. and Oluokun, O.A. (2023) 'Vibration-based condition monitoring of rotating machinery using LabVIEW', *International Journal of Scientific Research in Civil Engineering*, 7(6), pp. 82–108. DOI: [10.32628/IJSRCE237529](https://doi.org/10.32628/IJSRCE237529).
56. Omoegun, G.O., Sunday, E.A., Essien, M.A. and Oluokun, O.A. (2024) 'Implementing smart automation solutions in medium-scale industries using SCADA systems', *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 10(5), pp. 1207–1231. DOI: [10.32628/CSEIT241061242](https://doi.org/10.32628/CSEIT241061242). (DOI)
57. Özel, A., Pautz, T. and Schmidt, N. (2020) 'Infrastructure as Code als Maßnahme zur Cloud-Automatisierung – Hilfestellung zur Auswahl des richtigen Werkzeugs', *HMD Praxis der Wirtschaftsinformatik*, 57(5), pp. 936–948. DOI: [10.1365/s40702-020-00657-0](https://doi.org/10.1365/s40702-020-00657-0). ([Springer](#))
58. Pollini, A., Callari, T.C., Tedeschi, A., Ruscio, D., Save, L., Chiarugi, F. and Guerri, D. (2022) 'Leveraging human factors in cybersecurity: An integrated methodological approach', *Cognition, Technology & Work*, 24, pp. 371–390. DOI: [10.1007/s10111-021-00683-y](https://doi.org/10.1007/s10111-021-00683-y).
59. Pretorius, M. and Ngejane, H. (2019) 'Best practices for establishment of a national information security incident management capability (ISIMC)', *The African Journal of Information and Communication*, 24, pp. 1–20. DOI: [10.23962/10539/28656](https://doi.org/10.23962/10539/28656).
60. Savaş, S. and Karataş, S. (2022) 'Cyber governance studies in ensuring cybersecurity: An overview of cybersecurity governance', *International Cybersecurity Law Review*, 3, pp. 7–34. DOI: [10.1365/s43439-021-00045-4](https://doi.org/10.1365/s43439-021-00045-4).
61. Sele, D. and Chugunova, M. (2024) 'Putting a human in the loop: Increasing uptake, but decreasing accuracy of automated decision-making', *PLOS ONE*, 19(2), e0298037. DOI: [10.1371/journal.pone.0298037](https://doi.org/10.1371/journal.pone.0298037).
62. Senturk, Z. and Irmak, E. (2024) 'Windows Active Directory Domain Services and enterprise network security: Analysis and measures for PowerShell access', *Gazi University Journal of Science Part C: Design and Technology*, 12(3), pp. 475–487. DOI: [10.29109/gujsc.1447924](https://doi.org/10.29109/gujsc.1447924).
63. Serrano, J., Faustino, J., Adriano, D., Pereira, R. and da Silva, M.M. (2021) 'An IT service management literature review: Challenges, benefits, opportunities and implementation practices', *Information*, 12(3), Article 111. DOI: [10.3390/info12030111](https://doi.org/10.3390/info12030111). ([MDPI](#))
64. Sharma, D.H., Dhote, C.A. and Potey, M.M. (2016) 'Identity and access management as Security-as-a-Service from clouds', *Procedia Computer Science*, 79, pp. 170–174. DOI: [10.1016/j.procs.2016.03.117](https://doi.org/10.1016/j.procs.2016.03.117).
65. Shittu, H. and Nabil, M. (2023) 'Smart supply chain management with attribute-based encryption access control', 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), pp. 198–204. DOI: [10.1109/CCWC57344.2023.10099268](https://doi.org/10.1109/CCWC57344.2023.10099268).
66. Shittu, H.A. and Shittu, M.A. (2024) 'AI-powered digital twins for predictive maintenance and operational optimization of renewable energy systems', *International Journal of Science, Architecture, Technology and Environment*, 1(7), pp. 151–163. DOI: [10.63680/ijstate0324085.011](https://doi.org/10.63680/ijstate0324085.011).
67. Shittu, H.A., Shittu, M.A. and Olagunju, F. (2024) 'Cyber physical resilience in digital substations: IoT-enabled adaptive protection for secure DER integration', *International Journal of Science, Architecture, Technology and Environment*, 1(3), pp. 81–99. DOI: [10.63680/ijstate032532.08](https://doi.org/10.63680/ijstate032532.08).
68. Shittu, M.A., Adeniji, I.O., Shittu, H. and Opara, I.S. (2022) 'Blockchain-assisted secure data exchange architectures for SCADA-controlled power systems', *Iconic Research and Engineering Journals*, 6(3), pp. 359–380. DOI: [10.64388/IREV6I3-1714041](https://doi.org/10.64388/IREV6I3-1714041).
69. Shittu, M.A., Shittu, H., Adeleke, O.J. and Adedokun, O.J. (2023) 'Digital twin modeling for real-time monitoring and fault detection in smart substations', *International Journal of Industrial Engineering Research and Development*, 14(2), pp. 25–44. DOI: [10.34218/IJIERD_14_02_003](https://doi.org/10.34218/IJIERD_14_02_003).
70. Shittu, M.A., Shittu, H., Akintayo, T.A. and Opara, I.S. (2025) 'IoT-enabled microgrid and BESS integration in ADMS: A framework for climate-resilient grid operation', 2025 IEEE Symposium on Wireless Technology & Applications (ISWTA), pp. 1–9. DOI: [10.1109/ISWTA68114.2025.11330023](https://doi.org/10.1109/ISWTA68114.2025.11330023).
71. Sibanda, M. and von Solms, R. (2019) 'Devising a strategy for IT governance implementation in municipalities: A case study of South Africa', *The Electronic Journal of Information Systems in Developing Countries*, 85(2), e12067. DOI: [10.1002/isd2.12067](https://doi.org/10.1002/isd2.12067).
72. Spring, J.M. and Illari, P. (2021) 'Review of human decision-making during computer security incident analysis', *Digital Threats: Research and Practice*, 2(2), Article 11, pp. 1–47. DOI: [10.1145/3427787](https://doi.org/10.1145/3427787).

73. Spring, M., Faulconbridge, J. and Sarwar, A. (2022) 'How information technology automates and augments processes: Insights from artificial-intelligence-based systems in professional service operations', *Journal of Operations Management*, 68(6–7), pp. 592–618. DOI: [10.1002/joom.1215](https://doi.org/10.1002/joom.1215).
74. Sunday, E.A. and Omoegun, G.O. (2022) 'Automation of process control systems using Siemens TIA Portal: A case study approach', *International Journal of Scientific Research in Mechanical and Materials Engineering*, 6(5), pp. 30–55.
75. Sunday, E.A. and Omoegun, G.O. (2022) 'Smart fault detection in HVAC systems using sensor-based monitoring', *International Journal of Scientific Research in Science, Engineering and Technology*, 7(4), pp. 380–403. DOI: [10.32628/IJSRSET229473](https://doi.org/10.32628/IJSRSET229473).
76. Sunday, E.A., Omoegun, G.O., Essien, M.A. and Oluokun, O.A. (2020) 'Transitioning from reactive to predictive maintenance in mechanical systems', *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 6(6), pp. 425–447.
77. Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S.J.J., Ouyang, C., ter Hofstede, A.H.M., van de Weerd, I., Wynn, M.T. and Reijers, H.A. (2020) 'Robotic process automation: Contemporary themes and challenges', *Computers in Industry*, 115, Article 103162. DOI: [10.1016/j.compind.2019.103162](https://doi.org/10.1016/j.compind.2019.103162). (Utrecht University)
78. Tanner, N.H. (2019) 'Managing logs', in *Cybersecurity Blue Team Toolkit*. Indianapolis, IN: Wiley, Chapter 9. DOI: [10.1002/9781119552963.ch9](https://doi.org/10.1002/9781119552963.ch9).
79. Williams, C., Hodgetts, H.M., Morey, C., Macken, B., Jones, D.M., Zhang, Q. and Morgan, P.L. (2020) 'Human error in information security: Exploring the role of interruptions and multitasking in action slips', in *HCI International 2020 – Posters. Communications in Computer and Information Science*, 1226, pp. 622–629. Cham: Springer. DOI: [10.1007/978-3-030-50732-9_80](https://doi.org/10.1007/978-3-030-50732-9_80).
80. Willie, M.M. (2025) 'Artificial intelligence and automation in administrative practices: A critical examination of the impact on institutional memory and governance frameworks', *Advances in Human Resource Development and Management*, 1(1), pp. 25–35. DOI: [10.55578/hrdm.2507.003](https://doi.org/10.55578/hrdm.2507.003).
81. Yusuf, M., Akintunde, A., Habeeb, S. and Quadri, A. (2023) 'Radio Frequency Identification (RFID) based library management system', *International Journal of Technology and Systems*, 8(2), pp. 21–35. DOI: [10.47604/ijts.2041](https://doi.org/10.47604/ijts.2041).