



Evaluating User Acceptance Testing Effectiveness

Aisha I. Abdulwahab^{1*}, Harouna Wendpanga Yann Christian Sankara², Serif Oyindamola Oyesiji³, Olumide Kumuyi⁴

¹Cummins Inc., Indiana, USA

²Five9, San Ramon, California, USA

³FrieslandCampina WAMCO, Lagos, Nigeria

⁴New York University Abu Dhabi, Abu Dhabi, United Arab Emirates

* Corresponding author: Aisha I. Abdulwahab, ifewahab@gmail.com

Abstract

Background: User acceptance testing (UAT) is the last structured opportunity for business users to confirm that a system supports their work before release, yet it is routinely compressed, staffed by borrowed and tired testers, scripted from happy paths and closed by a signature that certifies attendance more than readiness. Objective: This review asks how UAT effectiveness should be measured and which conditions determine whether UAT detects fitness-for-use failures and informs a genuine release decision. Method: An integrative review of testing, requirements, packaged-software implementation and standards literature published to 2024, with cross-domain work on inspection and diagnostic detection used for analogy, was appraised for construct definition, escape-window specification and defect attribution. Results: Escaped-defect-based detection effectiveness is the most defensible outcome measure, but only when its observation window, severity weights and attribution rule are fixed in advance. Detection depends on scenario representativeness against production process variants, verifiable acceptance criteria, protected and rested business testers, and explicit coverage of configured objects in packaged software; schedule compression moderates all four. Contribution: The VERDICT framework (Validity of scenarios, Escape accounting, Readiness of criteria, Decision authority, Individual tester capacity, Compression, Tailoring), seven falsifiable propositions, formal indicators and a stepped-rollout validation design. Implications: Sign-off should be a recorded decision carrying a real right to refuse. No proposition has yet been tested; each awaits prospective evaluation in live programmes.



Keywords: user acceptance testing, defect detection effectiveness, escaped defects, acceptance criteria, release sign-off, ERP implementation, tester fatigue

1 Introduction

1.1 The Last Gate and Its Reputation

User acceptance testing (UAT) occupies a peculiar position in software delivery. It is the final structured activity before a system reaches the people whose work depends on it, and it is the only test level in which those people, rather than developers or professional testers, execute the checks. Testing standards describe acceptance testing as the stage at which a system is evaluated against user needs, business processes and contractual conditions to decide whether it should be accepted, which makes UAT a validation activity (building the right system) rather than a verification activity (building the system right) (ISO/IEC/IEEE, 2018; IEEE Computer Society, 2014). Testing texts note that the defects remaining at this point are mostly those no earlier level was designed to catch: misunderstood requirements, missing process steps, unusable screens and data that behave unexpectedly (Myers et al., 2011; Kaner et al., 1999). UAT must therefore find the defects that matter most at the point where there is least time to fix them.

Practitioner experience suggests that UAT often fails to carry that burden. Business users are released for a few days and handed project-written scripts to run in an environment that differs from production in data, integrations and access rights. Scripts follow the path that the designers expected rather than the paths that real work takes. Acceptance criteria are vague enough that almost any behaviour can be argued to pass. Testing windows shrink as upstream slippage consumes the schedule while the go-live date stays fixed. At the end, a manager signs an acceptance form, sometimes with known high-severity defects deferred to a post-release list, and the organisation discovers the real acceptance test in the first weeks of production (Eyetsemitan et al., 2023b; Eyetsemitan et al., 2023a). These complaints are anecdotal in form, but they point to identifiable mechanisms that can be measured and governed.

The central claim of this review is that UAT effectiveness is a measurable property of a release process, not an impression. It can be estimated by comparing what UAT found with what escaped within a defined window, and explained by a small set of conditions: representative scenarios, criteria precise enough to act as an oracle, capable and protected testers, packaged-



software configuration in scope, the degree of schedule compression, and a signatory able and willing to refuse. Each condition has a counterpart in literatures on test adequacy, requirements quality, inspection reliability and governance, from which a mechanism-level account can be assembled (Zhu et al., 1997; Mairiza et al., 2010; Sakyi et al., 2022b).

1.2 Gap, Research Questions and Contributions

The review responds to three shortcomings in current knowledge. To begin with, the testing literature has well-developed adequacy criteria for code-facing test levels, from statement and branch coverage to mutation adequacy, and a mature body of work on test prioritisation, but comparatively little on how to judge the adequacy of business-facing acceptance tests whose oracle is a user's judgement rather than a specification (Zhu et al., 1997; Ammann & Offutt, 2016; Daka & Fraser, 2014). Second, where UAT is evaluated at all, it is usually evaluated by activity counts: scripts executed, pass rates and defects raised. These counts rise with effort and fall with scope reduction, but they say little about the defects that were not found, which is the quantity a release decision actually needs (Edivri & Oteri, 2024; Tonoyan et al., 2024b). Third, sign-off is treated in project governance literature as a control point, yet its behaviour as a social act, including the conditions under which signatories feel able to refuse, has received little analytical attention (Eyetssemitan et al., 2020; Aminu-Ibrahim et al., 2024).

The review addresses four research questions. RQ1 asks how UAT effectiveness should be defined and measured so that the measure reflects undetected as well as detected defects. RQ2 asks which conditions of scenario design, acceptance criteria and tester capacity are associated with higher detection in UAT. RQ3 asks how packaged ERP and commercial off-the-shelf (COTS) rollouts, and end-of-project schedule compression, change those conditions. RQ4 asks under what conditions sign-off functions as a governance decision rather than a ritual, and how that difference could be observed. The unit of analysis is the release, meaning a defined scope of change that passes through UAT and a sign-off decision into production, because that is the level at which detection, escape and decision can be linked in version history and incident records (Ramesh & Jarke, 2001; Cleland-Huang et al., 2012).

The article makes four contributions. (1) It formalises UAT detection effectiveness as a windowed, severity-weighted and attribution-ruled measure, shows why unwindowed or unattributed versions are misleading, and proposes complementary estimators based on seeded defects and capture and recapture across independent tester groups. (2) It synthesises evidence on six conditions of UAT detection (scenario representativeness, acceptance-criteria quality,

tester capability and fatigue, packaged-software tailoring, schedule compression and sign-off authority) and adjudicates conflicting findings by specifying when each holds. (3) It proposes the VERDICT framework and derives seven falsifiable propositions, each with a stated test. (4) It specifies a stepped-rollout validation design with process, mechanism, outcome and harm measures, and a threat analysis specific to research on acceptance testing (Wieringa, 2014; Hevner et al., 2004).

2 Conceptual Foundations

2.1 What Acceptance Testing Is For

The purpose of UAT is easily confused with the purposes of neighbouring activities. System and integration testing verify that components and interfaces behave according to design, typically under the control of professional testers who can read specifications and logs. UAT asks a different question: whether the system, as configured and populated with realistic data, allows the business to carry out its processes correctly and efficiently. The ISO/IEC/IEEE 29119 series separates test processes by level and purpose, and requirements engineering standards distinguish verification of requirements from validation against stakeholder needs; UAT belongs to the second family (ISO/IEC/IEEE, 2018; IEEE Computer Society, 2014). A practical consequence is that a UAT defect is not only a failure of code. It may be a missing requirement, a configuration error, a data migration fault, a training gap or a mismatch between the designed process and the way work is done. Schemes that force every finding into a code-defect category discard information that later explains escapes (Kaner et al., 1999; Mairiza et al., 2010).

A second confusion concerns the word acceptance. In information systems research, technology acceptance models explain whether individuals intend to use a system, drawing on perceived usefulness and perceived ease of use. User acceptance testing is not a measurement of that intention. A system can pass UAT and still be resisted, and a system that users like can fail UAT because it posts the wrong ledger entries. The two literatures meet at one point: organisational readiness and employee readiness shape both whether users engage seriously with testing and whether they later adopt the system (Akhigbe et al., 2024; Eyetsemitan et al., 2023a). The review keeps the constructs separate: UAT effectiveness is defined by detection before release and by the quality of the ensuing decision, not by satisfaction with testing.

A third function of UAT is contractual. In outsourced and vendor-led delivery, acceptance is a

legal event that transfers risk, triggers payment milestones and starts warranty periods. Contract terms often define acceptance as the absence of open defects above a severity threshold, or as deemed acceptance after a fixed period without written rejection. Research on contract error reduction and liability mitigation in engineering and energy service contracts shows how ambiguously drafted acceptance and liability clauses shift risk to the party least able to control it (Anichukwueze et al., 2022; Uchechi & Ominyi, 2019). Procurement frameworks for regulated environments make the same point from the buyer's side: acceptance criteria written into a contract become the practical definition of quality whether or not they reflect operational need (Okonkwo et al., 2021). Once acceptance is a payment trigger, both parties acquire incentives unrelated to detection.

2.2 The Business-User Tester as an Imperfect Detector

Signal detection theory offers a precise language for the tester's task. Each scenario execution presents an opportunity in which a failure is either present or absent, and the tester either reports a problem or does not. Detection performance is summarised by two quantities: sensitivity, the ability to distinguish failures from acceptable behaviour, and the response criterion, the tester's threshold for raising a defect. Under the equal-variance Gaussian model, sensitivity is $d' = z(H) - z(F)$, where H is the hit rate on true failures, F is the false-alarm rate on acceptable behaviour, and z is the inverse of the standard normal distribution. The criterion is $c = -[z(H) + z(F)] / 2$. The two respond to different interventions. Domain expertise and realistic data raise sensitivity, whereas social pressure, fear of blame for delaying a release or a belief that defects will be rejected as change requests shift the criterion towards silence (Myers et al., 2011; Obriki & Arumosoye, 2020).

The detection framing connects UAT to fields that measure inspection reliability directly. Non-destructive testing of welds and pressure equipment characterises inspectors and techniques through probability-of-detection curves that relate the chance of finding a flaw to its size and to inspection conditions, and quality standards require that inspection performance be demonstrated rather than assumed (Atta et al., 2021; Atta et al., 2022). Refinery integrity programmes combine inspection results with risk assessment so that inspection effort is concentrated where consequence and likelihood are highest (Atta et al., 2020). Diagnostic laboratories face the same logic, with sensitivity and specificity established under controlled conditions and quality systems designed to prevent drift (Yusuf & Abdulrazaq, 2018; Ogbete et al., 2018). The analogy does not treat business users as calibrated instruments; it holds that

UAT detection can be characterised with similar evidence, including known-positive seeded cases, rather than inferred from scripts completed.

The detector framing also exposes the role of fatigue. Vigilance research has long documented a decrement in detection over sustained monitoring, especially when targets are rare and the task is repetitive. UAT has exactly these properties: most scripted steps pass, true failures are sparse, and testers are often asked to execute long script batches after or alongside their normal work. Studies of human error causation in high-risk industrial activities identify fatigue, time pressure and task monotony as recurrent precursors of missed signals, and heat stress research shows how environmental load compounds cognitive decline over a shift (Obriki & Arumosoye, 2020; Arumosoye & Obriki, 2018). Tester capacity is therefore a time-varying property of the detector, to be measured across the testing day as well as across testers.

2.3 Sign-Off Between Governance Act and Ritual

Institutional theory distinguishes activities that produce the outcomes they claim from activities that signal conformity to expectations while being decoupled from technical work. A sign-off can serve either role. As a governance act, it is a decision by an accountable person who has reviewed evidence about residual risk, who could have refused and whose signature changes what happens next. As a ritual, it is the closing ceremony of a phase whose end date was fixed in advance, and the signature certifies that testing took place. The form itself cannot tell them apart; their correlates can: whether sign-off is ever withheld, whether conditions attached to conditional acceptance are tracked to closure, whether the signatory can explain the residual risks accepted and whether escaped defects later lead to any change in how acceptance is conducted (Obogo et al., 2019b; Sakyi et al., 2022b).

Research on legitimate authority and voluntary cooperation sharpens the analysis. Work on the psychology of tax behaviour distinguishes cooperation grounded in trust in a legitimate authority from compliance extracted by power, and shows that the two produce different quality of cooperation (Kirchler, 2007). Reviews of what motivates compliance point to perceived fairness, reciprocity and social norms as stronger drivers than sanctions alone (Alm, 2019). Transposed to UAT, the signatory stands in a double relation of authority. The project asks the business owner to sign, and the business owner relies on testers who report to line managers rather than to the project. If testers expect defect reports to be treated as obstruction, evidence is filtered before it reaches the signatory; if the signatory expects refusal to be overridden, the decision becomes symbolic. Safety leadership research in temporary project organisations



describes the same upward filtering of bad news (Obogo et al., 2023; Arumosoye & Obriki, 2024).

The governance literature also offers a structural test. A control that has no defined consequence when its threshold is breached is a report, not a control. Risk-based governance frameworks for data protection, cloud configuration and enterprise risk insist that each control has an owner, a threshold, an escalation route and a record of exceptions, and that exceptions expire (Mbonu et al., 2018; Aliliele et al., 2023a; Falemi et al., 2024). Applied to UAT, sign-off is a governance act when there is a written rule stating what evidence is required, who may accept which residual risks, what happens if the rule is not met, and how waivers are recorded and revisited. Where these elements are absent, the review predicts that sign-off will be insensitive to the evidence produced by testing, a prediction that can be tested by relating sign-off timing and conditions to open-defect profiles across releases (Dosunmu & Ogundele, 2019; Jimoh et al., 2023).

3 Review Protocol

3.1 Sources, Search Strings and Eligibility

Evidence on acceptance testing arrives in forms that resist pooling: adequacy experiments, ERP case studies, practitioner UAT frameworks, standards and inspection studies share no common effect metric. An integrative review suits this material, since what these sources can offer is a comparison of how a practice works, where it works and what it changes. Reporting borrows the PRISMA 2020 items that remain meaningful without meta-analysis, namely stated eligibility rules, named databases, a documented appraisal routine and an account of how findings were combined, and the overall procedure follows software engineering guidance on systematic reviews (Page et al., 2021; Kitchenham et al., 2009). The databases were Scopus, Web of Science, IEEE Xplore, SpringerLink, the ACM Digital Library and the AIS eLibrary. Reference lists of core works on test adequacy, test prioritisation and requirements traceability were then mined in both directions, citing and cited, to reach material that keyword searches missed (Zhu et al., 1997; Catal & Mishra, 2013; Gotel & Finkelstein, 1994).

Search strings combined three concept blocks. The acceptance block contained ("user acceptance test*" OR "acceptance test*" OR UAT OR "business acceptance" OR "operational acceptance" OR "sign-off" OR "release readiness"). The effectiveness block contained ("defect detection" OR "escaped defect*" OR "defect leakage" OR "defect removal efficiency" OR



"phase containment" OR "test effectiveness" OR "test adequacy" OR "post-release defect*"). The context block contained (ERP OR COTS OR "packaged software" OR "enterprise system*" OR "business user*" OR "acceptance criteria" OR "test scenario*" OR "schedule pressure"). A cross-domain variant replaced the acceptance block with ("probability of detection" OR "inspection reliability" OR "diagnostic accuracy" OR "vigilance decrement" OR "near miss") to locate detection-reliability analogues. Publications up to the end of 2024 were admitted, and older foundational texts were kept regardless of age (Easterbrook et al., 2008; Kaner et al., 1999).

A source qualified if it offered one of four kinds of contribution (empirical findings, a systematic review, a standard, or a conceptual model specified in enough detail to be tested) and if that contribution bore on at least one of the following: a measure of acceptance-test effectiveness, a condition affecting detection at the acceptance level, the governance of acceptance decisions, or the specific features of packaged-software acceptance. Procedure descriptions that said nothing about effects or conditions were set aside, as were vendor marketing pieces and texts that named acceptance testing only as a lifecycle label. Cross-domain sources were eligible only when the detection or governance mechanism they described had a clear counterpart in acceptance testing, and they were used to define constructs and generate rival explanations, never as evidence about software projects (Runeson & Höst, 2009; Arumosoye & Obriki, 2019).

3.2 Appraisal, Extraction and Synthesis Logic

Appraisal used questions tailored to the measurement problems of acceptance testing rather than a generic quality checklist. For any source reporting detection or escape figures, the appraisal recorded whether the post-release observation window was stated, whether escapes were attributed to the test level that should have caught them, whether severity was defined independently of the team that fixed the defect, and whether release size or change volume was controlled. For sources reporting effects of practices, it recorded temporal ordering, the comparator, the presence of concurrent changes and the plausibility of reverse causation, for example mature programmes adopting better UAT practices because they were already mature (Dybå & Dingsøyr, 2008; Runeson & Höst, 2009).

Extraction used a single matrix in which supporting and contradicting findings were recorded side by side. Fields were: setting (sector, delivery model, custom versus packaged software), unit of analysis, construct and its operational definition, measure and window, outcome,



direction of association, stated conditions and limitations. Synthesis followed a mechanism, context and outcome logic in which a finding was coded to the VERDICT dimension it informed and to the condition under which it held. Where findings conflicted, the synthesis did not average them but asked which contextual difference, measurement difference or selection mechanism could explain the conflict, a procedure that design science methodology describes as treatment of validation evidence against stated conditions (Wieringa, 2014). Table 1 summarises each protocol stage, the rule applied and the bias it was designed to counter; Figure 1 shows how the evidence streams were coded into the framework dimensions.

Table 1. Review protocol for evidence on user acceptance testing effectiveness

Protocol stage	Rule applied in this review	Bias it counters
Question framing	Release as unit of analysis; effectiveness defined by detection before release and quality of the sign-off decision	Drift towards activity counts such as scripts executed
Sources	Scopus, Web of Science, IEEE Xplore, SpringerLink, ACM DL, AIS eLibrary; two-way citation chasing from adequacy, prioritisation and traceability works	Vocabulary gaps between testing, ERP and governance communities
Search blocks	Acceptance terms AND (effectiveness terms OR context terms); detection-reliability variant for analogues; window to December 2024	Missing work that uses leakage, containment or readiness instead of effectiveness
Eligibility	Evidence or specified model on a measure, condition, decision or packaged-software feature of acceptance	Inclusion of procedure descriptions with no account of effect
Appraisal	Escape window stated; attribution rule; independent severity; control for release size; temporal order	Over-reading unwindowed or unattributed escape figures
Extraction	Setting, delivery model, construct, measure, outcome, direction, conditions in one matrix	Selective extraction of supportive findings
Synthesis	Coding to VERDICT dimensions; conflicts adjudicated by condition; cross-domain sources limited to analogy	False generalisation from inspection or clinical settings

Figure 1 makes the asymmetric role of cross-domain evidence explicit. Studies of inspection probability of detection, diagnostic accuracy and safety oversight enter only through the appraisal stage as sources of construct definitions and rival explanations, while direct evidence on testing, ERP delivery and standards informs all seven dimensions. The appraisal criteria listed in the central box, particularly the escape window and the attribution rule, were the most frequent reasons for downgrading the weight of an empirical claim, because a detection figure

without them cannot be compared across releases. The figure also shows that outputs are of three kinds (propositions, indicators and a validation design) and that none of them is presented as an empirical effect estimate (Kitchenham et al., 2009; Borg et al., 2014).

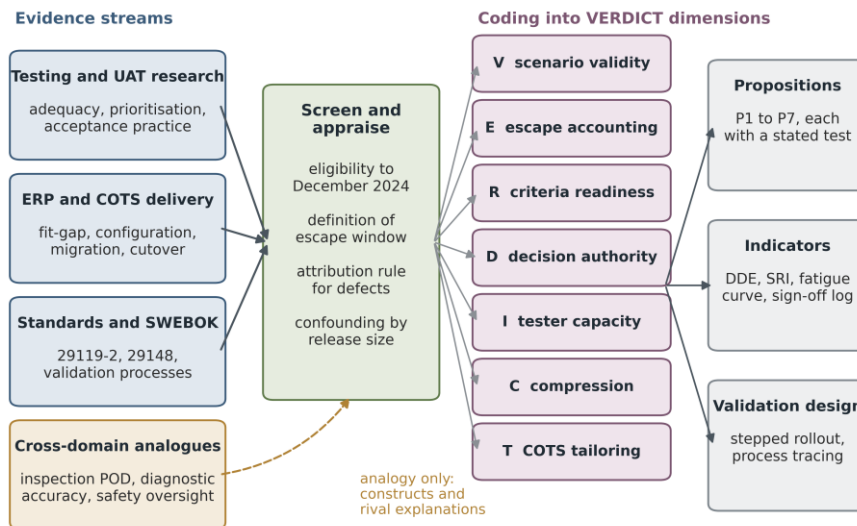


Figure 1. Evidence streams, appraisal criteria and coding of findings into the VERDICT dimension

4 Evidence Synthesis

4.1 Measuring Detection: Escaped Defects and Their Denominators

The most widely used outcome measure for a test level is some form of defect detection effectiveness, also called defect removal efficiency or phase containment. For UAT it can be written as a ratio of the defects UAT found to the defects UAT found plus those that escaped UAT and surfaced later. It is cheap to compute from a defect tracker, but its value depends on three choices that are often left implicit. The first is the observation window after release. The second is severity weighting. The third is the attribution rule that decides which production defects count as UAT escapes rather than as escapes from earlier levels or as new requirements. Left open, they make the measure incomparable across releases and improvable by redefinition rather than better testing (Kaner et al., 1999; Ammann & Offutt, 2016).

A windowed and severity-weighted definition makes these choices explicit. Let D be the set of defects found during UAT, $E(w)$ the set of production defects discovered within w days of release that are attributed to UAT scope, and $s(d)$ a severity weight fixed before release (for

example 8, 4, 2 and 1 for severity levels 1 to 4). Then:

$$DDE_UAT(w) = \sum_{\{d \in D\}} s(d) / [\sum_{\{d \in D\}} s(d) + \sum_{\{e \in E(w)\}} s(e)]$$

Attribution requires a rule stating that a production defect belongs to UAT scope if the affected business process, role or configured object was within the agreed UAT scope and the failure would have been observable with the planned data and environment. Defects in out-of-scope processes are recorded separately as scope escapes, which is itself diagnostic information about scenario selection. Figure 3(a), discussed below, illustrates why the window matters: two releases with the same UAT findings can show different effectiveness at 30 days and a reversed ranking by 180 days, because escapes surface at different rates. Studies of attribution modelling in marketing analytics face a structurally similar problem of assigning credit across touchpoints, and they show that rule-based attribution can systematically bias credit towards the last or first stage unless the rule is validated against held-out evidence (Sanni et al., 2022; Wedraogo & Sanni, 2024).

Detection effectiveness based on escapes has a further weakness: it depends on production usage to reveal the defects that UAT missed, so rarely used functions produce few escapes even when they are badly tested. Two complementary estimators reduce this dependence. The first is defect seeding, in which a small number of realistic failures are deliberately introduced into the UAT environment, for instance an incorrect tax rate on one product class or a missing approval step for one cost centre, and the proportion found estimates the hit rate. The second is capture and recapture across two independent tester groups executing overlapping scenarios. If group A finds n_1 defects, group B finds n_2 , and m are found by both, the Lincoln and Petersen estimator of total defects present in the overlapping scope is:

$$\hat{N} = (n_1 \times n_2) / m, \text{ and estimated undetected defects} = \hat{N} - (n_1 + n_2 - m)$$

Both estimators assume that detectability is broadly homogeneous and that groups are independent, assumptions that are violated when groups share scripts or discuss findings, so they should be treated as bounds rather than point truths. They are nevertheless informative because they can be computed before release, when the decision is still open. The analogy with sampling-based detection in food and feed contaminant screening is instructive: the absence of detection in a sample says little unless the detection limit and sampling plan are known, and quantification requires a validated method rather than a count of negative results (Tawose et al., 2023; Ogbete et al., 2023). Real-world evidence research in health systems makes a parallel argument about using routinely collected outcome data to evaluate interventions, with explicit



attention to confounding and missing data (Ilodigwe & Adesemoye, 2024a).

Escaped defects also need a denominator that reflects exposure. A release that changes three screens and a release that replaces a general ledger should not be compared on raw escape counts. Normalising escapes by the number of changed business process steps, by the number of configured objects touched or by production transaction volume gives rates that can be tracked across releases. Performance-monitoring research in operations and business analytics warns that indicator dashboards without stable denominators produce apparent trends that are artefacts of changing scope (Ogbole et al., 2021; Filani et al., 2022). Anomaly detection methods developed for fraud and financial time series can also surface silent failures, such as postings that balance but land in the wrong account, and so add to E(w) defects that users worked around without reporting (Atakpa & Abetoh, 2022; Atakpa & Fobellah, 2023).

4.2 Scenario Representativeness and Business-Process Variants

The second condition is whether UAT scenarios represent the work that production will see. Code-level adequacy criteria measure coverage against program structure, and prioritisation research has developed measures such as the average percentage of faults detected (APFD) to compare test orderings (Zhu et al., 1997; Li et al., 2007). Neither translates directly to acceptance testing, whose relevant structure is the set of business process variants rather than code paths. A variant is a distinct path through a process, such as a purchase order with a split delivery, a partial goods receipt and a price variance at invoice. Event logs typically show a few variants carrying most transactions and a long tail carrying a disproportionate share of exceptions and rework. UAT scripts written from design documents tend to cover the frequent variants and the designed exception paths but miss the long tail that exists only in operational practice (Catal & Mishra, 2013; Myers et al., 2011). Process efficiency research on CRM and workflow automation in small healthcare practices points in the same direction, since automated patient engagement flows must be validated against the exception paths that real volumes generate and not only against the designed sequence (Eyetsemitan et al., 2024a).

A scenario representativeness index (SRI) can make this measurable. Let V be the set of process variants observed in a reference period of production or legacy-system data, $p(v)$ the share of transactions following variant v , $k(v)$ a criticality weight reflecting financial, regulatory or customer consequence, and $c(v)$ an indicator equal to 1 if at least one UAT scenario exercises variant v with realistic data. Then:

$$SRI = \frac{\sum_{v \in V} p(v) k(v) c(v)}{\sum_{v \in V} p(v) k(v)}$$

The index is bounded between 0 and 1 and can be computed separately for frequency-weighted and criticality-weighted versions to show whether the scenario set covers the ordinary workload, the dangerous cases or both. Its main limitation is dependence on the reference period, since legacy logs may encode workarounds that the new design removes. Scenario-based modelling in strategic finance shows the value of systematically constructing scenarios that span both expected and adverse conditions rather than relying on a single central case (Filani et al., 2023). Risk-scoring approaches to cross-border payments show how composite scores can weight rare but consequential cases, which is the logic that $k(v)$ encodes (Adesuyi et al., 2024). Traceability research adds a further requirement: each scenario should be linked to the requirement or process step it validates, so that coverage gaps can be seen and so that an escape can be traced back to the missing link (Borg et al., 2014; Cleland-Huang et al., 2012). Representativeness also concerns people and roles, not only paths. A scenario executed by a central finance analyst does not test what a branch clerk with a restricted role, a different screen resolution and a different working rhythm will experience. Research on requirements traceability and non-functional requirements shows that usability, performance under load and access-control behaviour are the attributes most often left implicit and discovered late (Mairiza et al., 2010; Ramesh & Jarke, 2001). Security and access-control behaviour is particularly relevant for UAT in regulated settings, because segregation of duties and role-based access are business rules that only business users can validate as correct. Work on role-based security architecture in CRM platforms handling personal data, and on access control in bring-your-own-device environments, illustrates how role design errors surface as business failures rather than security incidents (Badmus et al., 2024; Ladapo et al., 2023b). Research on diversity in project teams suggests that tester panels drawn from a narrow group miss failure modes that other groups would meet (Amayo et al., 2023c).

4.3 Acceptance Criteria as the Test Oracle

Every test needs an oracle, a means of deciding whether observed behaviour is correct. In UAT the oracle is the combination of written acceptance criteria and the tester's judgement. When criteria are precise, testers can recognise failure without negotiation; when they are vague, the decision about whether a behaviour is a defect is deferred to argument, and the result depends on who argues. Requirements standards specify that individual requirements should be



necessary, unambiguous, complete, singular, feasible, verifiable and correct, and that requirement sets should be consistent and complete (ISO/IEC/IEEE, 2018). Criteria that fail verifiability, for example "the report should be user friendly" or "the process should be efficient", cannot function as an oracle at all. Given, When and Then formats improve verifiability by forcing a precondition, an action and an observable outcome, but not correctness of that outcome (IEEE Computer Society, 2014; Gotel & Finkelstein, 1994).

Acceptance-criteria quality can be measured at the artefact level. An acceptance-criteria defect rate counts, per hundred criteria, the proportion that fail at least one verifiability check: absence of an observable outcome, use of unquantified qualifiers, missing boundary values, missing negative cases or conflict with another criterion. Automated screening can detect several of these failures, and applications of artificial intelligence in software testing include classifiers for requirement smells and generators for test cases from natural-language criteria (Borah et al., 2022). Reviews of language-model evaluation used for release gating caution that automated judges share biases, reward fluent text and can be manipulated by phrasing, which argues for using them as screening aids with human review of flagged items rather than as oracles in their own right (Nwakamma et al., 2024a). Where criteria are generated or checked by autonomous agents, research on securing agentic workflows adds that tool outputs and retrieved content can be manipulated, so provenance of criteria becomes a quality attribute (Nwakamma et al., 2024b).

The evidence on criteria quality contains an apparent conflict. Some practitioner accounts report that detailed acceptance criteria reduce UAT disputes and escaped defects; others report that detailed scripts reduce the exploratory attention of business testers and cause them to confirm the script rather than examine the system. Both can hold. Precise criteria improve the oracle, while prescriptive step-by-step scripts constrain the search. The distinction implies that criteria should specify what counts as correct, while scenarios should leave testers room to vary data, sequence and role in ways that mirror their work. Frameworks for structured validation in lean small-business deployments reach a similar conclusion by combining a minimal set of mandatory checks with time-boxed exploratory sessions led by the most experienced users (Eyetsemitan et al., 2023b). Standardisation research on high-volume procedural work, such as dispute-resolution clinics, likewise finds that standard procedures improve consistency but can suppress the judgement needed for atypical cases unless exceptions are explicitly allowed (Uchechi et al., 2018).

4.4 Business-User Tester Capability, Workload and Fatigue

Business users bring the domain knowledge that makes UAT a validation activity, but they are rarely trained testers and are almost never released fully from operational duties. Three capability components can be distinguished. Domain competence is knowledge of the process and its exceptions. Testing competence is the ability to vary inputs, notice anomalies and record reproducible defects. System familiarity is the ability to navigate the new system well enough that navigation problems are not mistaken for defects, or defects for navigation problems. Professional development research shows that short, practice-embedded training improves performance when it is tied to the actual task and followed by feedback (Joseph & Nwankwo, 2024). Evidence from capability-building interventions in other domains, however, cautions that brief classroom training has modest and fading effects unless it is delivered close to the moment of use (Miller et al., 2015).

Workload and fatigue interact with capability. Business testers typically combine UAT with their normal roles, so testing hours are fragmented and often scheduled at the end of the working day or at weekends near go-live. Vigilance research predicts a decline in hit rate over continuous sessions and a shift in criterion as testers become more willing to accept marginal behaviour. Figure 3(b) illustrates the expected pattern for seeded-defect detection over a six-hour session with and without scheduled breaks; the parameters are illustrative, not estimates. The same pattern appears in safety research on shift work and heat load, where error rates rise late in a shift and are partially recovered by rest breaks (Arumosoye & Obriki, 2018; Obriki & Arumosoye, 2020). Reliability engineering adds a design principle: where a detection function degrades with time on task, redundancy and scheduled renewal are cheaper than attempting to make the component more robust (Gideon et al., 2019).

Capability and motivation are also shaped by the organisation. Employee readiness frameworks for technology adoption emphasise that staff engage with new systems when they understand the purpose of the change and see management commitment, and change-management research in small-business digital transformation reports that users treat testing as a chore when they were not involved in design (Akhigbe et al., 2024; Eyetsemitan et al., 2023a). Enterprise risk models that integrate human resource strategy with operational analytics show how staffing decisions, such as backfilling roles for people seconded to projects, determine whether nominal participation becomes real effort (Falemi et al., 2024). Near-miss research provides a



measurement idea: defects that a tester noticed but did not report because they seemed minor, or because reporting was cumbersome, are near misses that predict later escapes and can be captured through brief end-of-session debriefs (Arumosoye & Obriki, 2019).

4.5 UAT in Packaged ERP and COTS Rollouts

Packaged enterprise systems change the object of UAT. In a custom build, the business tests functions that the team wrote. In an ERP or COTS rollout, most functions were written by the vendor and are presumed to work; what the business must test is the configuration of those functions, the extensions and customisations added to them, the migrated master and transactional data, the interfaces to surrounding systems, the reports and forms, and the roles and authorisations that govern who can do what. Implementation methods begin with fit-gap analysis and plan integration tests, migration rehearsals and cutover rehearsals as separate streams before UAT. When these streams are compressed, UAT becomes the first time that configured processes run end to end with migrated data, and it inherits the defects of every earlier stream (Eyetsemitan et al., 2020; Dogbatsey et al., 2019).

Three packaged-software features deserve particular attention. The first is period-end processing. Month-end and year-end close, reconciliation, accrual reversal and consolidation run rarely and depend on accumulated data, so they are easily omitted from UAT windows that last a few weeks. Research on reconciliation control and financial workflow redesign shows that reconciliation failures often arise at the boundary between configured posting rules and manual adjustments, precisely the area that end-to-end tests with realistic volumes reveal (Dogbatsey et al., 2019). The second is vendor-supplied test scripts. Vendors and integrators often provide standard scripts that exercise delivered functionality, which are useful for regression but mirror the vendor's model of the process rather than the client's, so they raise pass rates without raising SRI. The third is multi-country rollouts, where localisation for tax, invoicing and statutory reporting differs by jurisdiction. Research on harmonising digital trade regulation across jurisdictions illustrates how the same transaction can be subject to different rules, which translates into jurisdiction-specific acceptance criteria (Jones & Ominyi, 2024b). Packaged systems also shift the balance between functional and control testing. Role design, segregation of duties and data-access rules are configured rather than coded, and errors in them are business failures that professional testers may not recognise. Governance frameworks for IT and operational technology convergence in regulated networks, and for zero-trust

architectures in critical infrastructure, emphasise that access rules must be validated against the operational roles that actually exist rather than against organisation charts (Ahmed et al., 2021; Adegbite et al., 2022). Data sensitivity classification and data protection impact assessment matter because UAT with migrated data frequently means UAT with copies of production personal data, which creates privacy obligations and argues for masking and synthetic data where representativeness permits (Aliliele et al., 2024a; Mbonu et al., 2022a); where vendor and client must share test data, privacy-preserving approaches to cross-organisational data management become relevant (Sanni et al., 2023). Upgrade cycles add a recurring regression burden: each vendor release can change delivered behaviour beneath configured processes, which makes a maintained, prioritised acceptance regression suite more valuable than a one-off UAT campaign (Li et al., 2007; Catal & Mishra, 2013).

4.6 Schedule Compression at the End of the Delivery Chain

UAT sits at the end of a dependency chain whose earlier links are subject to slippage, while the go-live date is often anchored by business events such as a fiscal year start, a contract expiry or a regulatory deadline. The result is a structural squeeze: when design, build or integration testing run late, the slack is taken from UAT because it is the last flexible interval before an immovable date. Compression can be quantified as a compression ratio, the ratio of actual to planned UAT calendar days, and as an entry-criteria waiver count, the number of UAT entry conditions (for example, integration test exit, data migration reconciliation, environment readiness) that were waived to start on time. Simulation modelling of process optimisation shows that removing buffer from late stages of a pipeline raises the variance of outcomes more than the mean, which predicts that compressed UAT produces not only more escapes on average but a higher chance of severe escapes (Tonoyan et al., 2023; Tonoyan et al., 2022b). Compression affects UAT through several paths at once. It cuts test cycles, so first-cycle fixes go unretested or are retested only on the happy path. It makes UAT overlap system testing, so business testers spend scarce hours on unstable builds. It increases the number of deferred defects, often reclassified to lower severity so that exit criteria can be met. And it raises pressure on the signatory. Research on risk-based accelerated approval in pharmaceutical regulation offers a disciplined model of how speed can be traded against evidence: accelerated pathways are permitted only with explicit conditions, confirmatory evidence obligations and the possibility of withdrawal (Eze et al., 2024b). Regulatory readiness maturity models similarly



distinguish organisations that document conditional decisions and follow them through from those that treat conditions as formalities (Eze et al., 2022b).

The compression evidence contains a second conflict. Some programmes report that shorter, more intensive UAT windows with dedicated testers perform better than long, part-time windows, while others associate short windows with high escape rates. The conflict dissolves when tester capacity is taken into account. A two-week window with fully released, rested and prepared testers can deliver more effective testing hours than a six-week window in which testers attend two hours a day after their normal work. Research on emergency readiness in energy and process facilities makes a comparable point: readiness depends on rehearsed capability under realistic conditions rather than on the time elapsed since planning began (Arumosoye & Obriki, 2023). Risk pathway studies of lifting operations show how pressure propagates through small, individually defensible shortcuts into serious failure (Arumosoye & Obriki, 2022).

4.7 Sign-Off Practice and Where the Evidence Conflicts

Evidence on sign-off practice is thinner than on the other conditions, but three patterns recur. First, sign-off with known open defects is common and is not in itself a failure; residual risk is unavoidable and acceptance with conditions is often the correct decision. The failure is acceptance without an explicit statement of which risks were accepted, by whom, with what compensating controls and until when. Second, sign-off is often delegated downwards to someone without authority to delay the release, or upwards to a sponsor without detailed knowledge of the evidence, and in either case the link between evidence and decision weakens. Third, sign-off records are seldom revisited after release, so organisations do not learn whether accepted risks materialised. Governance models for partnerships and public health oversight show that accountability requires decisions recorded in a form comparable with later outcomes (Aminu-Ibrahim et al., 2024; Anioke & Atima, 2023a).

The conflicts in the evidence are therefore mostly conditional rather than contradictory, and Figure 2, introduced in Section 5, is built around the conditions that reconcile them. Detailed criteria help when they define the oracle but hurt when they become prescriptive scripts. Short windows help when tester capacity is protected but hurt when it is not. Vendor scripts help for regression but not for representativeness. The one conflict that is not easily resolved concerns whether independent test teams should run UAT on behalf of the business. Independence

improves testing competence and relieves business staff, and KPI-driven service-delivery frameworks favour it for consistency (Edivri & Oteri, 2024), yet it weakens the domain oracle. The review resolves this provisionally by distinguishing execution from judgement: professional testers can prepare data and execute routine checks, but acceptance of business behaviour and the sign-off decision must remain with business users, a division that the propositions in Section 5 make testable (Kaner et al., 1999; Obriki & Arumosoye, 2019).

A final pattern concerns how organisations use UAT evidence over time. Organisational learning research in safety management describes maturity as the progressive linking of incident evidence to changes in upstream practice, so that the same failure does not recur (Arumosoye & Obriki, 2021). Leadership-driven safety culture transformation in large construction workforces depends on visible response to reported hazards, and hazard identification practice improves when findings from one project are carried into the next (Obogo et al., 2019a; Obogo et al., 2019). In UAT terms, escaped defects are the incidents, and the learning test is whether each severe escape leads to a change in scenarios, criteria, tester preparation or sign-off rules. Continuous monitoring approaches for cloud misconfiguration and for integrated network and security operations show how detection that feeds directly into configuration change can close this loop quickly (Aliliele et al., 2023a; Ladapo et al., 2024).

5 The VERDICT Framework and Propositions

5.1 Framework Structure

The VERDICT framework organises the synthesis into seven dimensions whose initials give it its name. Validity of scenarios (V) is representativeness against production variants, roles and data. Escape accounting (E) is the windowed, severity-weighted and attributed measure of what UAT missed. Readiness of criteria (R) is their verifiability as an oracle. Decision authority (D) is sign-off by a mandated owner under a written rule with a real option to refuse. Individual tester capacity (I) joins domain and testing competence with protected, rested time. Compression (C) is schedule squeeze and waived entry criteria. Tailoring (T) is explicit coverage of configured, extended and migrated objects in packaged rollouts. The framework is a causal hypothesis rather than a maturity scale: its dimensions are not stages to be completed in order, and improving one while neglecting another can leave detection unchanged (Wieringa, 2014; Hevner et al., 2004).

Figure 2 shows the hypothesised structure. Four inputs (V, R, I and T) determine detection in

UAT, expressed as hit rate and sensitivity. Detection informs the sign-off decision (D), which in turn determines release outcomes: escaped defects by severity, business disruption and workarounds. Schedule compression (C) moderates both detection and decision, because it simultaneously reduces effective testing and raises pressure to accept. Escape accounting (E) closes the loop by attributing post-release defects back to scenarios, criteria and tailoring decisions. The structure follows KPI research showing that indicators become useful only when tied to a decision and a feedback route (Tonoyan et al., 2024b; Sakyi et al., 2022b). Table 2 operationalises each dimension with an indicator, a data source, a failure signal and the governance response that the signal should trigger.

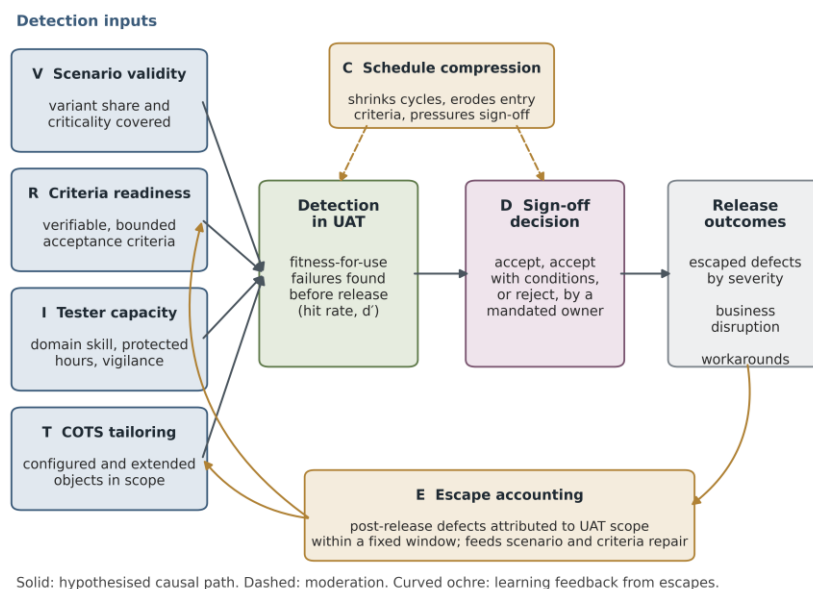


Figure 2. The VERDICT framework: detection inputs, sign-off decision, release outcomes, moderation by schedule compression and learning feedback from escapes

Table 2. VERDICT dimensions, indicators, data sources, failure signals and governance responses

Dimension	Indicator	Data source	Failure signal	Governance response
V Scenario validity	SRI, frequency- and criticality-weighted	Legacy or pilot event logs; scenario trace matrix	Criticality-weighted SRI below agreed floor	Add long-tail and role variants before UAT entry
E Escape accounting	DDE_UAT(30) and DDE_UAT(90), severity-weighted;	Incident and defect trackers with attribution	Severity 1 or 2 escape in UAT scope	Root-cause review; update scenarios and



	scope escapes	field		criteria
R Criteria readiness	Acceptance-criteria defect rate per 100 criteria	Criteria review log; automated smell screen	Rate above threshold at UAT entry	Rework criteria; defer affected scenarios
D Decision authority	Sign-off with documented residual-risk statement; refusal and condition closure rates	Sign-off records; waiver register	Conditions open past expiry; no refusal in many releases	Escalate to accountable executive; revise sign-off charter
I Tester capacity	Protected tester-hours; seeded-defect hit rate by session hour	Timesheets; seeded-defect results	Hit rate falls sharply late in sessions	Shorten sessions; add breaks or second tester
C Compression	Compression ratio; entry-criteria waivers	Plan versus actual; entry checklist	Ratio below 0.6 or two or more waivers	Formal go or no-go with sponsor; conditional acceptance rules
T COTS tailoring	Share of configured, extended and migrated objects exercised	Configuration inventory; migration reconciliation	Untested extensions or period-end processes	Add targeted scenarios; mock period-end run

Two features of Table 2 deserve comment. First, every failure signal is paired with an action, following the principle that a threshold without a defined response is reporting rather than control. Studies of business intelligence architecture and of data-driven risk control in energy infrastructure reach the same conclusion and add that thresholds should be reviewed as evidence accumulates (Mbonu et al., 2021a; Obriki & Arumosoye, 2018). Second, the indicators differ in timing. SRI, criteria defect rate, compression ratio and tailoring coverage are available before or during UAT and can therefore inform the sign-off decision; seeded-defect hit rate is available during UAT; and DDE is available only after release. Risk-based intrusion detection and security operations research uses the same distinction between leading and lagging signals to organise response (Dosunmu & Ogundele, 2020; Okoruwa et al., 2024).

5.2 Propositions and Their Tests

Seven propositions follow from the synthesis. They are stated so that each could be shown false by a feasible study, and each names the evidence that would count against it. They are untested and should be read as a research programme, not as findings. Table 3 summarises them with their predicted direction and the test that would refute them. They are conditional because most UAT practices appear to help only under specific conditions (Easterbrook et al., 2008; Runeson & Höst, 2009).



P1 (scenario validity). Across releases of comparable size, higher criticality-weighted SRI at UAT entry is associated with higher DDE_UAT(90), and the association is stronger for escapes in rarely executed variants than in frequent ones. Refuting evidence would be a null or negative association after controlling for release size and tester-hours. P2 (criteria readiness). A lower acceptance-criteria defect rate at UAT entry is associated with fewer disputed defects and higher DDE, but prescriptive step-level scripts, holding criteria quality constant, are associated with lower detection of unscripted variants. The test compares scripted and exploratory sessions on the same criteria with seeded defects placed inside and outside the scripted paths (ISO/IEC/IEEE, 2018; Eyetsemitan et al., 2023b).

Table 3. VERDICT propositions, predicted direction and refutation test

Proposition	Predicted direction	Evidence that would refute it
P1 Scenario validity	Criticality-weighted SRI raises DDE_UAT(90), most for rare variants	No association after controlling for release size and tester-hours
P2 Criteria readiness	Verifiable criteria raise DDE; prescriptive scripts lower detection of unscripted variants	Scripted sessions find as many off-path seeded defects as exploratory sessions
P3 Tester capacity	Hit rate falls with session length; breaks and pairing attenuate the fall	Flat hit-rate profile; protected hours add nothing beyond calendar days
P4 Compression	Compression lowers DDE and raises severe escapes via fewer tester-hours and retests	Effect vanishes once release size and risk are controlled, with no mediation
P5 COTS tailoring	Coverage of configured and migrated objects predicts escapes better than vendor-script counts	Vendor-script execution predicts escapes equally well
P6 Decision authority	Chartered sign-off is sensitive to open-defect profiles; uncharted sign-off follows the plan date	Equal sensitivity of sign-off to defect profiles with and without a charter
P7 Escape learning	Feedback of severe escapes reduces recurrence in the same process area	Recurrence unchanged over four or more releases despite feedback

P3 (tester capacity). Seeded-defect hit rate declines with continuous session length, and scheduled breaks or paired testing attenuate the decline; protected tester-hours predict DDE better than calendar duration of the UAT window. Refutation would be a flat hit-rate profile across session hours or no incremental predictive value of protected hours over calendar days. P4 (compression). Compression ratio and entry-criteria waivers are associated with lower DDE

and with a higher share of severity 1 and 2 escapes, and the effect is mediated by reduced effective tester-hours and retest cycles. A design that measures both mediators can distinguish this proposition from the rival claim that compressed releases are simply larger or riskier (Tonoyan et al., 2022b; Gideon et al., 2019).

P5 (tailoring). In packaged-software rollouts, the share of configured, extended and migrated objects exercised in UAT predicts escapes better than the share of vendor-standard scripts executed; vendor scripts raise pass rates without reducing escapes. The test is a within-programme comparison across releases or country rollouts with object-level coverage data. P6 (decision authority). Where a written sign-off charter grants a mandated business owner an explicit right to refuse, sign-off timing and conditions are sensitive to open-defect profiles; where no charter exists, sign-off timing is predicted by the planned date and not by defect profiles. Refutation would be equal sensitivity in both conditions (Kirchler, 2007; Aminu-Ibrahim et al., 2024). P7 (escape learning). Programmes that feed each severe escape back into scenarios or criteria show declining escape rates for the same process area over successive releases; programmes that only report escapes do not. The test tracks recurrence at the level of process area across at least four releases (Arumosoye & Obriki, 2021; Obogo et al., 2019a).

5.3 Rival Explanations

Each proposition has a simpler rival that a credible study must rule out. The most important is selection by programme maturity. Well-run programmes tend to score well on every VERDICT dimension and also design better upstream, so associations could reflect general competence. Staged adoption within programmes, with pre-adoption baselines, addresses this better than cross-sectional comparison. A second rival is exposure: releases with fewer escapes may simply have been used less in their first weeks, which the exposure-normalised rates in Section 4.1 are designed to address. A third is reporting behaviour: production defect reporting depends on user willingness and on help-desk classification, so escapes can fall because reporting falls. Telemetry-based detection of silent failures and periodic audit of help-desk categories reduce this risk (Atakpa & Fobellah, 2023; Ogbole et al., 2021).

A fourth rival concerns reverse causation between sign-off and detection. Signatories who plan to refuse may encourage testers to search harder, so the sign-off charter could appear to improve detection when the real driver is the signatory's prior assessment of the release. Process tracing of the timing of signatory interventions can separate these paths. A fifth rival is seasonality:

releases near period-end show more escapes because period-end processes run soon after release, which argues for an escape window spanning a full business cycle. Research on anomaly detection and failure prediction in distributed assets illustrates how seasonal patterns confound failure attribution unless they are modelled explicitly (Komi & Adamolekun, 2021; Shittu et al., 2023). Load forecasting research makes a related point: a deep-learning model with good aggregate accuracy may still err systematically in particular districts, which is a warning that aggregate DDE can hide poor detection for particular roles, sites or countries (Komi, 2023).

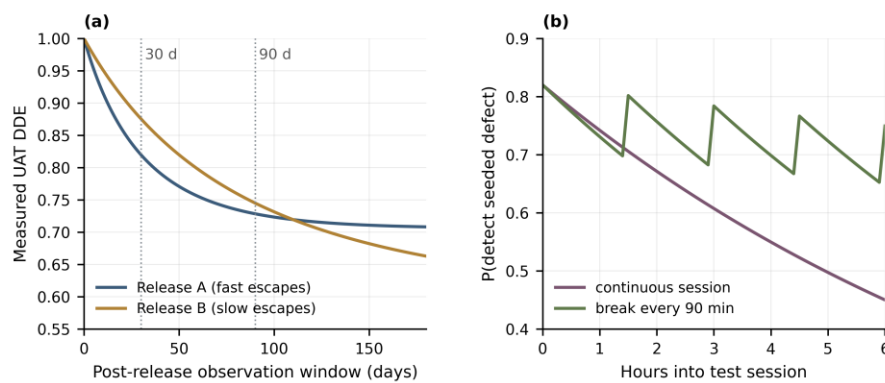


Figure 3. Illustrative dynamics behind two VERDICT measures: (a) measured UAT detection effectiveness as a function of the post-release window for two releases with identical UAT findings; (b) expected decline in seeded-defect detection over a test session with and without scheduled breaks

Figure 3 is illustrative and uses assumed parameters, not data. Panel (a) holds UAT findings fixed at 60 severity-weighted defects and models escapes that accumulate exponentially towards different totals at different rates. Release A appears worse at 30 days but better at 180 days, which demonstrates why DDE comparisons are meaningless unless the window is fixed in advance and long enough to include at least one full business cycle. Panel (b) shows a vigilance decrement in which hit probability declines with continuous time on task and recovers partially after breaks, with a slow residual decline across the day. The panel motivates P3 and the recommendation to measure detection by session hour. Both panels follow the logic of performance-trend monitoring, where the choice of window and baseline determines the apparent result (Tonoyan et al., 2024b; Wedraogo & Sanni, 2024).



6 Discussion

6.1 Theoretical Implications

The review has three implications for theory. First, it positions UAT as a detection system with two separable components, sensitivity and criterion, rather than as a process step. This framing explains why interventions that improve domain knowledge and data realism (sensitivity) and interventions that change the social consequences of reporting (criterion) should be evaluated separately, and why a UAT phase can produce few defects either because the system is good or because testers are unwilling to report. The detection framing links software testing to a body of measurement theory developed for inspection and diagnostics, which offers tools such as probability-of-detection curves and seeded-case validation that are rarely used in acceptance testing (Atta et al., 2021; Yusuf & Abdulrazaq, 2018). It also suggests that acceptance testing is closer to diagnostic laboratory quality than to code verification, since both depend on calibrated human judgement under controlled conditions (Ogbete et al., 2019).

Second, the review treats sign-off as a governance act whose meaning must be demonstrated rather than assumed. Institutional accounts of ceremonial conformity predict that formal controls can persist while being decoupled from the work they supposedly govern, and the VERDICT indicators for decision authority, particularly refusal rates, condition closure and sensitivity of sign-off timing to open defects, provide a way to observe decoupling empirically. The legitimacy literature adds that the quality of cooperation depends on trust in the authority as well as on its power, which implies that sign-off effectiveness depends on testers' willingness to report as much as on the signatory's formal mandate (Kirchler, 2007; Alm, 2019). Readiness research on regulatory regimes for artificial intelligence and on privacy by design in corporate transactions makes a similar distinction between compliance that is documented and compliance that changes decisions (Ominyi et al., 2024b; Ominyi, 2021).

Third, the review shows that the familiar defect removal ratio has three hidden parameters (window, severity weighting and attribution) whose choice can reverse conclusions. Making them explicit turns a descriptive statistic into a measurement instrument whose validity can be assessed at any test level evaluated by escapes. Attribution in turn depends on links between production defects, process steps, requirements and scenarios, links that traceability research shows are expensive to recover after the fact (Borg et al., 2014; De Lucia et al., 2007). Market research frameworks for regulated industrial sectors show a comparable need to fix the

population and denominator before interpreting shares, which is the role of the reference variant set in SRI (Sanni & Attah, 2023c).

6.2 Practical Implications

For practice, the most immediate change is to write a sign-off charter before UAT begins. The charter should name the accountable business owner, the evidence required at sign-off (at minimum SRI, criteria defect rate, open defects by severity, compression ratio and seeded-defect hit rate), the residual risks that may be accepted at each level of authority, the conditions under which acceptance may be conditional, the expiry of conditions and the escalation route if the owner refuses. Performance frameworks for IT service delivery and large commercial organisations show that accountability improves when indicators are few, owned and linked to decisions (Edivri & Oteri, 2024; Sakyi et al., 2022b). Project governance research on multi-stakeholder ventures adds that escalation routes must be agreed before disputes arise, because routes negotiated under deadline pressure favour the party with schedule control (Eyetsemitan et al., 2020).

The second change is to protect tester capacity explicitly. Programmes should budget effective tester-hours rather than calendar days, backfill operational roles for seconded testers, limit continuous test sessions to around ninety minutes with breaks, and pair new testers with experienced ones for their first sessions. Seeded defects, a handful per cycle placed in high-criticality variants, give an early read of detection before release. Capability-building research suggests that orientation delivered in the real environment immediately before testing is more effective than earlier classroom training (Joseph & Nwankwo, 2024; Miller et al., 2015). Workforce strategy research warns that relying on the same small group of experts for every release creates a single point of failure in the detection system (Ilodigwe & Adesemoye, 2024b; Akin-Oluyomi & Akhigbe, 2023b).

The third change concerns packaged-software programmes. UAT scope should be defined from the configuration inventory and migration objects rather than from vendor scripts, and should include at least one mock period-end close with migrated data. Test data should be masked or synthesised where representativeness permits, following data classification and impact assessment practice (Aliliele et al., 2024a; Mbonu et al., 2021). Agile supply chain transformation research shows how embedding IT risk and compliance controls into delivery sprints prevents controls from being bolted on at the end (Mbonu et al., 2020a). Finally, programmes should track DDE at 30 and 90 days with a fixed attribution rule and feed every

severe escape into scenario and criteria repair. Dashboard research suggests presenting these measures with their denominators and uncertainty so that trends are not over-read (Filani et al., 2022).

6.3 Objections and Boundary Conditions

The framework invites objections, and cost comes first. Computing SRI requires event logs, seeding defects requires effort, and capture and recapture requires duplicate testing. The reply is proportionality: seeding a few defects and computing SRI on the top critical processes costs little compared with a single severe escape in finance or supply chain, and the full apparatus is warranted only for high-consequence releases. Reliability and failure analysis research routinely applies more intensive assurance where consequence is higher (Gideon et al., 2019). Risk quantification for rare but severe events in aviation illustrates how low-probability, high-consequence failures justify disproportionate assurance effort relative to their frequency (Adeyelu, 2020). Gaming, such as downgrading escape severity or narrowing scope, is countered by independent severity assignment, recorded scope escapes and periodic audit (Jimoh et al., 2023; Sakyi et al., 2022b).

A second objection is that agile and continuous delivery make UAT obsolete, because acceptance is continuous and releases are small. The framework applies with modification. In continuous delivery, the acceptance decision moves from a phase gate to feature flags, canary releases and product-owner acceptance of increments, but the detection problem remains: someone with domain judgement must validate behaviour, and escapes must be attributed. Empirical reviews of agile development report that customer representatives are often overloaded and under-empowered, which is exactly the tester-capacity and decision-authority problem in another form (Dybå & Dingsøyr, 2008). Test-practice surveys show that developer-level testing is widespread but uneven in quality, which suggests that business validation cannot simply be replaced by automation at lower levels (Daka & Fraser, 2014).

A third objection concerns the scope of UAT relative to specialist testing. Business users should not be expected to test network-level resilience, protocol-level attacks or industrial control security, which require specialist tools and knowledge. Work on denial-of-service mitigation, industrial control anomaly detection and operational technology compliance shows the depth of specialist testing outside UAT (Jimoh & Ahmed, 2024; Adebayo et al., 2023; Adegbite et al., 2020). The boundary condition is that VERDICT covers fitness-for-use validation of business behaviour, including business-level access rules, and assumes that non-functional and

security testing is performed elsewhere. Where organisations rely on UAT to catch these failures, low DDE reflects a scoping error rather than poor acceptance testing. A further boundary concerns very small organisations, where the tester, the signatory and the business owner may be the same person; lean validation frameworks for such settings compress the VERDICT dimensions into a short checklist, but the detection logic still applies (Eyetsemitan et al., 2023b; Onche et al., 2023).

7 Limitations and Research Agenda

7.1 Limitations of the Review

The review has limitations that follow from its design and from the state of the evidence. The empirical literature on UAT effectiveness is small and dominated by case reports, so the synthesis relies heavily on mechanism reasoning, on adjacent literatures and on cross-domain analogies whose transferability is argued rather than demonstrated. Cross-domain sources were restricted to construct definition and rival explanations, yet some analogies may still seem loose. Publication bias is likely: organisations rarely publish evidence of failed UAT or of sign-off under duress, and practitioner accounts over-represent programmes that recovered from problems. Terminology is inconsistent, with defect leakage, escape rate, containment and removal efficiency used for overlapping but not identical measures, which may have caused relevant work to be missed despite broad search strings (Page et al., 2021; Kitchenham et al., 2009).

The framework itself is untested. Its seven propositions are derived from the synthesis and are consistent with the evidence reviewed, but none has been evaluated prospectively, and the illustrative parameters in Figure 3 are not estimates. The indicators in Table 2 have face validity and are computable from data that most programmes hold, yet their reliability, sensitivity to change and resistance to gaming are unknown. The review also privileges enterprise and packaged-software contexts; embedded, safety-critical and consumer mobile contexts have different acceptance regimes, including regulatory certification and staged app-store release, that the framework does not address. Supply reliability research in pharmaceutical manufacturing offers a reminder that quality assurance can be conceptualised as a determinant of downstream outcomes only when the pathway is measured, not assumed (Asiedu & Asiedu, 2024b).

7.2 Validation Design and Research Agenda

The recommended validation design is a stepped rollout across programmes or product lines within one or more organisations, in which each unit moves from usual UAT practice to VERDICT-guided practice in a randomised order, with every release serving as a measurement point. Figure 4 shows the design. The stepped structure gives each unit its own baseline, addresses selection by maturity and allows secular trends to be separated from the intervention. Interrupted time series is a weaker fallback. The design measures four layers per release: process (SRI, criteria defect rate, protected tester-hours), mechanism (seeded-defect hit rate, sign-off dissent and condition logs), outcome (DDE at 30 and 90 days, severity 1 and 2 escapes) and harm (go-live delay, tester overtime and waivers) (Runeson & Höst, 2009; Hevner et al., 2004). Table 4 lists the principal threats to validity for such a study and the corresponding mitigations.

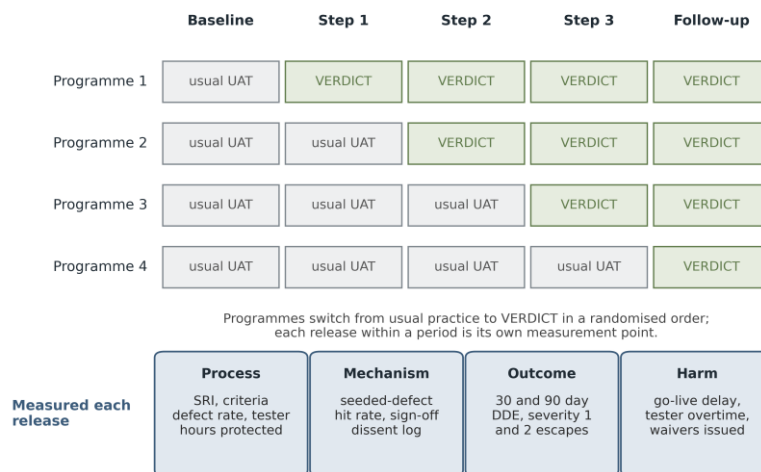


Figure 4. Stepped-rollout validation design for VERDICT with four measurement layers recorded at each release

Figure 4 shows why the design can distinguish the framework's effect from the rivals discussed in Section 5.3. Because programmes adopt in a random order, a programme-maturity explanation would predict improvement unrelated to the adoption step, whereas the framework predicts improvement that follows the step. The harm layer is essential: a framework that reduces escapes by delaying every go-live or exhausting business staff would not be a net improvement, and recording delay, overtime and waivers allows that trade-off to be assessed. Process tracing within programmes, using sign-off records and tester debriefs, should confirm that the mechanism operated, for example that seeded-defect results were actually discussed at



sign-off. Debriefs and post-release user surveys generate large volumes of free text, and AI-enabled methods for synthesising stakeholder and survey data offer a way to code that material consistently across programmes, provided a human analyst audits the coding (Obukadata et al., 2024). Real-time KPI systems and business performance visualisation research provide practical means of collecting these measures without adding heavy reporting burden (Tonoyan et al., 2024b; Ogbole et al., 2021).

Table 4. Threats to validity specific to evaluating UAT effectiveness and their mitigations

Threat	How it arises in UAT research	Mitigation
Escape-window censoring	Releases observed for different periods; late-surfacing defects missed	Fix 30 and 90 day windows in advance; include one full business cycle
Attribution bias	Teams classify escapes as change requests or earlier-level defects	Written attribution rule; independent classification of a sample
Severity inflation or deflation	Severity negotiated to meet exit criteria or to protect vendors	Severity assigned by a party independent of the fixing team
Reporting suppression	Users work around defects without logging them	Telemetry-based anomaly detection; help-desk category audits
Observation effects	Testers work harder because detection is being measured	Seed defects in all arms; measure over several releases
Seeded-defect realism	Seeded failures easier or harder than natural ones	Seed from historical escape types; rotate seeders
Contamination across arms	Practices diffuse between programmes before their step	Record diffusion; analyse by actual exposure as sensitivity check
Transfer limits	Findings from ERP programmes applied to embedded or consumer software	Report context fields; restrict claims to observed delivery models

Beyond validation of the full framework, the synthesis points to several specific research needs. First, empirical estimation of business-tester detection curves using seeded defects, including the effect of session length, breaks, pairing and domain experience, would convert the vigilance argument into evidence. Second, comparative studies of vendor scripts, configuration-driven scenarios and exploratory sessions in packaged-software rollouts would test P2 and P5 directly. Third, qualitative and archival studies of sign-off decisions, including interviews with



signatories and analysis of waiver registers, would reveal how often sign-off is refused, under what conditions and with what consequences. Fourth, work on the inter-rater reliability of escape attribution would strengthen every escape-based measure; financial inclusion measurement shows how the choice of population and denominator shapes headline indicators (World Bank, 2014). Fifth, research on cooperation across organisational boundaries suggests studying UAT in vendor and client partnerships where testers, signatories and fixers sit in different organisations (Liadi, 2024c; Jones & Ominyi, 2021).

Finally, research should examine the equity and burden of UAT. Business testers are drawn from the operational teams that bear the cost of escapes, and the testing burden falls unevenly across roles and sites. Studies of who is asked to test, who is heard in defect triage and whose processes appear in scenarios would link UAT effectiveness to the fairness of change, using methods from diversity measurement research (Uchechi et al., 2022). Corridor monitoring by unmanned aircraft offers a coverage metaphor for such studies: inspection value depends on whether the whole corridor is flown, not on the number of flights (Dagodzo & Ahiaeké Patrick, 2020). Regulated laboratory design and risk-based assurance research likewise show that quality is built in through design choices rather than verified at the end (Ogbete et al., 2022; Fadayomi et al., 2019).

8 Conclusion

User acceptance testing is effective when it detects the failures that matter to the business before release and when its result shapes a real decision. That effectiveness can be measured. A windowed, severity-weighted and attributed detection ratio, complemented by seeded defects and capture and recapture estimates, gives an honest picture of what acceptance testing found and what it missed. Detection depends on scenarios that represent the work production will see, on acceptance criteria precise enough to serve as an oracle, on business testers who have the skill, the protected time and the rest to notice failure, and, in packaged-software programmes, on explicit coverage of configured, extended and migrated objects. Schedule compression erodes all of these at once and pressures the signatory.

The VERDICT framework brings these conditions together with escape accounting and decision authority into a testable model, with seven propositions that specify what would count against them and a stepped-rollout design capable of testing them. Its practical message is direct. Organisations should write the sign-off charter before testing starts, budget tester-hours



rather than calendar days, scope acceptance from the configuration rather than the vendor's scripts, and treat every severe escape as a reason to change how acceptance is done. Sign-off then becomes what it claims to be: a recorded decision about residual risk, made by someone with the authority and the evidence to refuse.

Funding

No dedicated grant from a public, commercial or charitable funder supported this work.

Competing interests

None declared.

Data availability

The article analyses published literature only; it generated or analysed no new data, and every source used appears in the reference list.

References

1. Adebayo, A., Adegbite, M. P., & Ahmed, M. O. (2023). AI augmented threat detection in industrial control systems: A systematic review of machine learning approaches for ICS anomaly detection. *International Journal of Engineering and Modern Technology*, 9(3), 287-340. <https://doi.org/10.56201/ijcsmt.v9.no3.2023.pg287.340>
2. Adegbite, M. P., Adebayo, A., & Ahmed, M. O. (2022). A security architecture model for IT and OT convergence in regulated energy networks: Design principles and governance alignment. *International Journal of Computer Science and Mathematical Theory*, 8(2), 81-132. <https://doi.org/10.56201/ijcsmt.v8.no2.2022.pg81.132>
3. Adegbite, M. P., Adebayo, A., & Ahmed, M. O. (2020). Securing operational technology networks in electric utilities: A systematic review of NERC CIP compliance and architectural threat mitigation. *International Journal of Engineering and Modern Technology*, 6(3), 103-146. <https://doi.org/10.56201/ijemt.vol.6.no3.2020.pg103.146>
4. Adesuyi, M. O., Akomolafe, O., Olaogun, B. O., Ndukwe, V. U., & Sakyi, J. K. (2024). AI-driven risk scoring model for global cross-border trade payment transactions. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(1), 1569-1581. <https://doi.org/10.62225/2583049X.2024.4.1.5281>



5. Adeyelu, O. O. (2020). An artificial intelligence-driven conceptual model for wildlife strike risk quantification and aircraft airworthiness impact assessment at high-traffic African airports. *Iconic Research and Engineering Journals*, 4(1), 339-368. <https://doi.org/10.64388/IREV4I1-1718482>
6. Ahmed, M. O., Adegbite, M. P., & Adebayo, A. (2021). Zero trust architecture for operational technology in North American critical infrastructure: A framework for implementation and resilience optimization. *International Journal of Engineering and Modern Technology*, 7(1), 66-113. <https://doi.org/10.56201/ijemt.vol.7.no1.2021.pg66.113>
7. Akhigbe, R., Falemi, A., & Akin-Oluyomi, O. T. (2024). An advanced framework for technology adoption that supports employee readiness in modern organizations. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(1), 1637-1659. <https://doi.org/10.62225/2583049X.2024.4.1.5344>
8. Akin-Oluyomi, O. T., & Akhigbe, R. (2023b). A conceptual supply chain leadership model for aligning logistics decisions with enterprise strategy. *International Journal of Advanced Multidisciplinary Research and Studies*, 3(1), 1725-1744. <https://doi.org/10.62225/2583049X.2023.3.1.5340>
9. Aliliele, C., Mbonu, I. S., & Iwuanyanwu, U. (2023a). A conceptual framework for continuous cloud misconfiguration monitoring and enterprise risk mitigation strategies. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(10), 373-394. <https://doi.org/10.32628/CSEIT2361071>
10. Aliliele, C., Mbonu, I. S., & Iwuanyanwu, U. (2024a). A conceptual framework for enterprise data sensitivity classification and regulatory traceability mechanisms. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3103-3124. <https://doi.org/10.62225/2583049X.2024.4.6.5991>
11. Alm, J. (2019). What motivates tax compliance? *Journal of Economic Surveys*, 33(2), 353-388. <https://doi.org/10.1111/joes.12272>
12. Amayo, E. B., Owulade, O. A., & Isi, L. R. (2023c). The role of diversity and inclusion in enhancing project team performance and delivery in multinational environments. *Journal of Frontiers in Multidisciplinary Research*, 4(1), 48-58. <https://doi.org/10.54660/IJFMR.2023.4.1.48-58>
13. Aminu-Ibrahim, A. Y., Ogbete, J. C., & Ambali, K. B. (2024). Governance and accountability models for public private partnerships in healthcare infrastructure development. *International*



Journal of Advanced Multidisciplinary Research and Studies, 4(6), 2943-2960.
<https://doi.org/10.62225/2583049X.2024.4.6.5699>

14. Ammann, P., & Offutt, J. (2016). Introduction to Software Testing (2nd ed.). Cambridge University Press.
15. Anichukwueze, C. C., Ominyi, M., Uzougbo, N. S., & Jones, B. C. (2022). Cross-functional legal education as risk mitigation: Empirical analysis of contract error reduction in engineering teams. *International Journal of Engineering and Modern Technology*, 8(5), 142-180. <https://doi.org/10.56201/ijemt.v8.no5.2022.pg142.180>
16. Anioke, S. C., & Atima, M. E. (2023a). Public health governance models using process optimization and performance metrics for regulatory oversight. *International Journal of Advanced Multidisciplinary Research and Studies*, 3(6), 2534-2548. <https://doi.org/10.62225/2583049X.2023.3.6.5491>
17. Arumosoye, O. M., & Obriki, O. D. (2023). Conceptual model for emergency response readiness and capability in energy and process facilities. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(3), 897-917. <https://doi.org/10.32628/CSEIT25112791>
18. Arumosoye, O. M., & Obriki, O. D. (2024). Conceptual model of safety leadership influence in large temporary project organizations. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3034-3047. <https://doi.org/10.62225/2583049X.2024.4.6.5895>
19. Arumosoye, O. M., & Obriki, O. D. (2022). Conceptual risk pathway model for lifting and rigging operations in heavy industrial construction. *Gyanshauryam, International Scientific Refereed Research Journal*, 5(5), 346-369. <https://doi.org/10.32628/GISRRJ2256237>
20. Arumosoye, O. M., & Obriki, O. D. (2018). Development of an integrated heat stress risk conceptual model for industrial operations in extreme environments. *Iconic Research and Engineering Journals*, 1(12), 141-160. <https://doi.org/10.64388/IREV1112-1714415>
21. Arumosoye, O. M., & Obriki, O. D. (2021). Organizational learning-based conceptual maturity model for continuous safety performance improvement. *International Journal of Multidisciplinary Research and Growth Evaluation*, 2(1), 970-980. <https://doi.org/10.54660/IJMRGE.2021.2.1.970-980>
22. Arumosoye, O. M., & Obriki, O. D. (2019). Systematic review of near-miss and hazard observation data utilization in industrial safety management. *Iconic Research and Engineering Journals*, 3(2), 981-999. <https://doi.org/10.64388/IREV312-1714417>



23. Asiedu, C. S., & Asiedu, A. A. (2024b). Reconceptualizing quality assurance in pharmaceutical manufacturing as a determinant of supply reliability. *International Journal of Health and Pharmaceutical Research*, 9(5), 148-173. <https://doi.org/10.56201/ijhpr.v9.no5.2024.pg148.173>
24. Atakpa, M. I., & Abetoh, N. F. (2022). A systematic review of machine learning advances in financial fraud detection for banking systems. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 8(2), 771-801. <https://doi.org/10.32628/CSEIT23906220>
25. Atakpa, M. I., & Fobellah, A. N. (2023). Anomaly detection in financial time-series data: A conceptual model for healthcare and banking applications. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(2), 967-997. <https://doi.org/10.32628/CSEIT2342441>
26. Atta, S., Adjaklo, E., Asomani, E., & Adenuga, O. M. (2022). Advances in welding inspection standards and quality assurance practices in large-scale power generation projects. *International Journal of Engineering and Modern Technology*, 8(5), 101-124. <https://doi.org/10.56201/ijemt.v8.no5.2022.pg101.124>
27. Atta, S., Tofah, P., & Adenuga, O. M. (2021). Advances in non-destructive testing methods for weld integrity evaluation in high-capacity power infrastructure. *International Journal of Engineering and Modern Technology*, 7(1), 20-47. <https://doi.org/10.56201/ijemt.vol.7.no1.2021.pg20.47>
28. Atta, S., Tofah, P., Kankam, M. A., & Adenuga, O. M. (2020). A critical review of NDT-based integrity management and corrosion risk assessment strategies for refinery process equipment. *International Journal of Engineering and Modern Technology*, 6(2), 19-46. <https://doi.org/10.56201/ijemt.vol.6.no2.2020.pg19.46>
29. Badmus, O., Jooda, D., & Anunagba, C. O. (2024). A privacy-by-design framework for role-based security architecture in Salesforce environments handling sensitive personal data. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3244-3254. <https://doi.org/10.62225/2583049X.2024.4.6.6243>
30. Borah, S., Aliliele, K. C., Rakshit, S., & Vajjhala, N. R. (2022). Applications of artificial intelligence in software testing. In P. K. Mallick et al. (Eds.), *Cognitive informatics and soft computing (Lecture Notes in Networks and Systems, Vol. 375, pp. 727-736)*. Springer. https://doi.org/10.1007/978-981-16-8763-1_60



31. Borg, M., Runeson, P., & Ardö, A. (2014). Recovering from a decade: A systematic mapping of information retrieval approaches to software traceability. *Empirical Software Engineering*, 19, 1565-1616. <https://doi.org/10.1007/s10664-013-9255-y>
32. Catal, C., & Mishra, D. (2013). Test case prioritization: A systematic mapping study. *Software Quality Journal*, 21, 445-478. <https://doi.org/10.1007/s11219-012-9181-z>
33. Cleland-Huang, J., Gotel, O., & Zisman, A. (Eds.). (2012). *Software and Systems Traceability*. Springer. <https://doi.org/10.1007/978-1-4471-2239-5>
34. Dagodzo, D., & Ahiaeké Patrick, M. C. (2020). UAV-based pipeline and corridor monitoring: A review of current practices and emerging technologies. *Iconic Research and Engineering Journals*, 3(10), 574-597. <https://doi.org/10.64388/IREV3I10-1716084>
35. Daka, E., & Fraser, G. (2014). A survey on unit testing practices and problems. *IEEE International Symposium on Software Reliability Engineering Workshops*, 201-211. <https://doi.org/10.1109/ISSREW.2014.11>
36. De Lucia, A., Fasano, F., Oliveto, R., & Tortora, G. (2007). Recovering traceability links in software artifact management systems using information retrieval methods. *ACM Transactions on Software Engineering and Methodology*, 16(4), Article 13. <https://doi.org/10.1145/1276933.1276934>
37. Dogbatsey, E. A., Ebhojie, O., & Oyeleye, A. O. (2019). Reconciliation control and financial workflow redesign in emerging market organizations: A conceptual framework. *Zenodo*. <https://doi.org/10.5281/zenodo.20447103> \[Originally numbered 3(5); journal name not captured in source.\]
38. Dosunmu, A. A., & Ogundele, P. O. (2020). Intrusion detection and prevention models for enhancing organizational cyber defense effectiveness. *Iconic Research and Engineering Journals*, 4(6), 310-324. <https://doi.org/10.64388/IREV4I6-1713226>
39. Dosunmu, A. A., & Ogundele, P. O. (2019). Security audit and enterprise risk assessment frameworks for resilient information systems. *Iconic Research and Engineering Journals*, 3(5), 434-447. <https://doi.org/10.64388/IREV3I5-1713225>
40. Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), 833-859. <https://doi.org/10.1016/j.infsof.2008.01.006>
41. Easterbrook, S., Singer, J., Storey, M., & Damian, D. (2008). Selecting empirical methods for software engineering research. In *Guide to Advanced Empirical Software Engineering*



(pp. 285-311). Springer.

42. Edivri, J., & Oteri, O. (2024). A conceptual KPI-driven decision and optimization framework for IT service delivery, portfolio performance, and adoption. *Gyanshauryam, International Scientific Refereed Research Journal*, 7(1), 167-187. <https://doi.org/10.32628/GISRRJ246643>
43. Eyetsemitan, R. A., Ambali, K. B., Oyeleye, A. O., & Fadayomi, O. (2023a). Change management in small business digital transformation: A systematic review and lean change adoption framework. *Gyanshauryam, International Scientific Refereed Research Journal*, 6(6), 521-550. <https://doi.org/10.32628/GISRRJ236648>
44. Eyetsemitan, R. A., Ambali, K. B., Oyeleye, A. O., & Fadayomi, O. (2023b). User acceptance testing in small business technology deployment: A structured validation framework for lean operational environments. *International Journal of Multidisciplinary Research and Growth Evaluation*, 4(6), 1512-1531. <https://doi.org/10.54660/IJMRGE.2023.4.6.1512-1531>
45. Eyetsemitan, R. A., Ambali, K. B., Oyeleye, A. O., & Fadayomi, O. (2020). Multi-stakeholder governance alignment in joint venture operations: A conceptual framework for coordinating business processes in highly regulated environments. *Iconic Research and Engineering Journals*, 4(4), 418-441. <https://doi.org/10.64388/IREV4I4-1716955>
46. Eyetsemitan, R. A., Oyeleye, A. O., Ambali, K. B., & Fadayomi, O. (2024a). CRM and workflow automation in small healthcare practices: A process efficiency framework for scalable patient engagement. *International Journal of Multidisciplinary Research and Growth Evaluation*, 5(6), 1931-1949. <https://doi.org/10.54660/IJMRGE.2024.5.6.1931-1949>
47. Eze, F. I., Akinleye, O. K., & Anene, U. N. (2024b). Development of a risk-based accelerated approval decision model balancing speed, safety, and evidence quality in pharmaceutical regulatory science. *Research Journal of Pure Science and Technology*, 7(6), 120-161. <https://doi.org/10.56201/rjpst.v7.no6.2024.pg120.161>
48. Eze, F. I., Anene, U. N., & Akinleye, O. K. (2022b). Design of an integrated regulatory readiness maturity model for pharmaceutical companies seeking global market entry. *International Journal of Health and Pharmaceutical Research*, 7(1), 77-112. <https://doi.org/10.56201/ijhpr.v7.no1.2022.pg77.112>
49. Fadayomi, O., Abolaji, T. O., Edivri, J., Ogbole, J. I., Okoruwa, P. O., & Akeju, B. (2019). Risk-based cybersecurity assurance and data availability: Limitations, advances and future research opportunities. *Iconic Research and Engineering Journals*, 2(12), 602-617.



<https://doi.org/10.64388/IREV2I12-1713779>

50. Fadayomi, O., Abolaji, T. O., Edivri, J., Ogbole, J. I., Okoruwa, P. O., & Akeju, B. (2019). Risk-based cybersecurity assurance and data availability limitations, advances and future research opportunities. *Iconic Research and Engineering Journals*, 2(12), 602–617. <https://doi.org/10.64388/IREV2I12-1713779>
51. Fadayomi, O., Abolaji, T. O., Edivri, J., Ogbole, J. I., Okoruwa, P. O., & Akeju, B. (2019). Risk-based cybersecurity assurance and data availability: Limitations, advances and future research opportunities. *IRE Journals*, 2(12). <https://doi.org/10.64388/IREV2I12-1713779>
52. Falemi, A., Akhigbe, R., & Akin-Oluyomi, O. T. (2024). A conceptual enterprise risk management model that integrates HR strategy with operational analytics. *International Journal of Multidisciplinary Research and Growth Evaluation*, 5(6), 1730-1753. <https://doi.org/10.54660/ijmrge.2024.5.6.1730-1753>
53. Filani, O. M., Nnabueze, S. B., Ike, P. N., & Wedraogo, L. (2022). Real-time risk assessment dashboards using machine learning in hospital supply chain management systems. *International Journal of Multidisciplinary Evolutionary Research*, 3(1), 65-76. <https://doi.org/10.54660/IJMER.2022.3.1.65-76>
54. Filani, O. M., Nnabueze, S. B., Sakyi, J. K., & Okojie, J. S. (2023). Scenario-based financial modelling for enhancing strategic decision-making and organizational long-term planning. *Journal of Frontiers in Multidisciplinary Research*, 4(2), 251–265. <https://doi.org/10.54660/JFMR.2023.4.2.251-265>
55. Filani, O. M., Nnabueze, S. B., Sakyi, J. K., & Okojie, J. S. (2023). Scenario-based financial modelling for enhancing strategic decision-making and organizational long-term planning. *Journal of Frontiers in Multidisciplinary Research*, 4(2), 251-265. <https://doi.org/10.54660/JFMR.2023.4.2.251-265>
56. Gideon, E. N., Adaramola, T. S., & Fadero, S. (2019). Reliability engineering and failure analysis in complex engineering systems. *Iconic Research and Engineering Journals*, 2(11), 703-727. <https://doi.org/10.64388/IREV2I11-1722490>
57. Gotel, O. C. Z., & Finkelstein, A. C. W. (1994). An analysis of the requirements traceability problem. *IEEE International Conference on Requirements Engineering*, 94-101. <https://doi.org/10.1109/ICRE.1994.292398>
58. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75-105. <https://doi.org/10.2307/25148625>



59. IEEE Computer Society. (2014). Guide to the Software Engineering Body of Knowledge Version 3.0. IEEE.
60. Ilodigwe, L., & Adesemoye, A. C. (2024a). Advances in real-world evidence, predictive analytics, and data-driven decision making for improving health system performance and patient outcomes. *International Journal of Health and Pharmaceutical Research*, 9(5), 174-197. <https://doi.org/10.56201/ijhpr.v9.no5.2024.pg174.197>
61. Ilodigwe, L., & Adesemoye, A. C. (2024b). Beyond technology: A strategic framework for building sustainable and resilient health systems through organizational strategy, operational excellence, and workforce capability. *International Journal of Medical Evaluation and Physical Report*, 8(6), 299-319. <https://doi.org/10.56201/ijmepr.v8.no6.2024.pg299.319>
62. ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29119-2 Software and systems engineering: Software testing processes. International Organization for Standardization.
63. ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29148 Systems and software engineering: Life cycle processes and requirements engineering. International Organization for Standardization.
64. Jimoh, H. O., & Ahmed, M. O. (2024). Analyzing Network Time Protocol (NTP) based amplification DDoS attack and its mitigation techniques. *Journal of Digital Innovations and Contemporary Research in Science, Engineering and Technology*, 12(2), 17-24. <https://doi.org/10.22624/AIMS/DIGITAL/V11N2P2x>
65. Jimoh, H. O., Ahmed, M. O., & Fagbade, M. O. (2023). The role of frameworks in cybersecurity governance. *Journal of Behavioural Informatics, Digital Humanities and Development Research*, 9(4), 7-16. <https://doi.org/10.22624/AIMS/BHI/V9N4P2>
66. Jones, B. C., & Ominyi, M. (2024b). A conceptual framework for harmonizing digital trade regulations in cross-border e-commerce between the United States and Nigeria. *International Journal of Economics and Financial Management*, 9(9), 337-364. <https://doi.org/10.56201/ijefm.v9.no9.2024.pg337.364>
67. Jones, B. C., & Ominyi, M. (2021). Developing a conceptual framework linking West African agricultural value chains to household economic stability. *International Journal of Economics and Financial Management*, 6(2), 89-114. <https://doi.org/10.56201/ijefm.v6.no2.2021.pg89.114>
68. Joseph, O. B., & Nwankwo, C. U. (2024). Professional development for STEM educators: Enhancing teaching effectiveness through continuous learning. *International Journal of Applied Research in Social Sciences*, 6(8), 1558-1574.



<https://doi.org/10.51594/ijarss.v6i8.1370>

69. Kaner, C., Falk, J., & Nguyen, H. Q. (1999). *Testing Computer Software* (2nd ed.). Wiley.
70. Kirchler, E. (2007). *The Economic Psychology of Tax Behaviour*. Cambridge University Press.
71. Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering: A systematic literature review. *Information and Software Technology*, 51(1), 7-15. <https://doi.org/10.1016/j.infsof.2008.09.009>
72. Komi, N. M., & Adamolekun, A. (2021). Interpretable machine learning for early failure prediction in distributed renewable energy assets. *International Journal of Multidisciplinary Research and Growth Evaluation*, 2(6), 1015-1038. <https://doi.org/10.54660/IJMRGE.2021.2.6.1015-1038>
73. Komi, N. M. (2023). When accurate forecasts are not fair forecasts: Deep learning load prediction and equity across districts. *Shodhshauryam, International Scientific Refereed Research Journal*, 6(6), 476-546. <https://doi.org/10.32628/SHISRRJ247102>
74. Ladapo, O. O., Dosunmu, A. A., Jooda, D., & Abolaji, T. O. (2024). Integrated network and security operation center: A systematic analysis. *International Journal of Multidisciplinary Futuristic Development*, 5(1), 65-80. <https://doi.org/10.54660/IJMFD.2024.5.1.65-80>
75. Ladapo, O. O., Jooda, D., Dosunmu, A. A., & Abolaji, T. O. (2023b). Systematic literature review on security access control policies and techniques based on privacy requirements in a BYOD environment. *International Journal of Advanced Multidisciplinary Research and Studies*, 3(6), 2890-2904. <https://doi.org/10.62225/2583049X.2023.3.6.6206>
76. Liadi, K. O. (2024c). Designing a cross-border security cooperation model: Nigeria's foreign policy response to regional terrorism. *Gyanshauryam, International Scientific Refereed Research Journal*, 7(3), 311-338. <https://doi.org/10.32628/GISRRJ247324>
77. Li, Z., Harman, M., & Hierons, R. M. (2007). Search algorithms for regression test case prioritization. *IEEE Transactions on Software Engineering*, 33(4), 225-237. <https://doi.org/10.1109/TSE.2007.38>
78. Mairiza, D., Zowghi, D., & Nurmuliani, N. (2010). An investigation into the notion of non-functional requirements. *ACM Symposium on Applied Computing*, 311-317. <https://doi.org/10.1145/1774088.1774153>
79. Mbonu, I. S., Aliliele, C., Iwuanyanwu, U., & Oluoha, O. M. (2018). A conceptual framework



- for legal and ethical risk modeling in enterprise data protection governance systems. *Iconic Research and Engineering Journals*, 2(2), 207-226. <https://doi.org/10.64388/IREV2I2-1714911>
80. Mbonu, I. S., Aliliele, C., Iwuanyanwu, U., & Uzoka, E. (2021a). A conceptual framework for risk based business intelligence architecture in financial technology platforms. *International Journal of Multidisciplinary Research and Growth Evaluation*, 2(6), 731-746. <https://doi.org/10.54660/IJMRGE.2021.2.6.731-746>
81. Mbonu, I. S., Iwuanyanwu, U., Aliliele, C., & Uzoka, E. (2020a). A conceptual framework for agile supply chain digital transformation with embedded IT risk and ISO compliance controls. *Iconic Research and Engineering Journals*, 3(11), 566-593. <https://doi.org/10.64388/IREV3I11-1714916>
82. Mbonu, I. S., Iwuanyanwu, U., Aliliele, C., & Uzoka, E. (2022a). A review of data protection impact assessment models in multi cloud financial infrastructure systems. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 8(1), 589-623. <https://doi.org/10.32628/CSEIT25442>
83. Mbonu, I. S., Iwuanyanwu, U., Aliliele, C., & Uzoka, E. (2021). A review of VoIP forensic analytics models for financial fraud detection and regulatory compliance monitoring. *International Journal of Multidisciplinary Research and Growth Evaluation*, 2(6), 711-730. <https://doi.org/10.54660/IJMRGE.2021.2.6.711-730>
84. Miller, M., Reichelstein, J., Salas, C., & Zia, B. (2015). Can you help someone become financially capable? *World Bank Research Observer*, 30(2), 220-246. <https://doi.org/10.1093/wbro/lkv009>
85. Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). Wiley.
86. Nwakamma, S., Ojukwu, J., & Oyesiji, S. O. (2024a). LLM-as-a-judge for automated model governance: A review of methods, biases, and release-gating practices. *World Journal of Innovation and Modern Technology*, 8(6), 185-216. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg185.216>
87. Nwakamma, S., Ojukwu, J., & Oyesiji, S. O. (2024b). Securing agentic AI enterprise workflows against prompt injection, tool poisoning, memory manipulation, and excessive agency threats. *World Journal of Innovation and Modern Technology*, 8(6), 217-248. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg217.248>
88. Obogo, S. F., Nwafor, M. I., & Ozobu, C. O. (2023). Conceptual leadership model for safety



- culture development in construction and engineering projects. *International Journal of Scientific Research in Civil Engineering*, 7(6), 121-153. <https://doi.org/10.32628/IJSRCE237554>
89. Obogo, S. F., Ozobu, C. O., & Uduokhai, D. O. (2019a). Advances in leadership driven safety culture transformation in large construction workforces. *Iconic Research and Engineering Journals*, 3(5), 507-523. <https://doi.org/10.64388/IREV3I5-1715497>
90. Obogo, S. F., Ozobu, C. O., & Uduokhai, D. O. (2019b). Development of a construction safety risk governance model for multi contractor high rise projects. *Iconic Research and Engineering Journals*, 2(9), 502-522. <https://doi.org/10.64388/IREV2I9-1715495>
91. Obogo, S. F., Uduokhai, D. O., & Ozobu, C. O. (2019). Systematic review of hazard identification and risk control practices in urban construction projects. *Iconic Research and Engineering Journals*, 2(12), 666-681. <https://doi.org/10.64388/IREV2I12-1715496>
92. Obriki, O. D., & Arumosoye, O. M. (2019). A conceptual framework linking management safety walkthrough frequency and coverage to safety culture outcomes in mega projects. *Iconic Research and Engineering Journals*, 2(8), 355-374. <https://doi.org/10.64388/IREV2I8-1714416>
93. Obriki, O. D., & Arumosoye, O. M. (2020). Conceptual framework for human error causation in high-risk construction and industrial activities. *International Journal of Multidisciplinary Research and Growth Evaluation*, 1(5), 715-727. <https://doi.org/10.54660/IJMRGE.2020.1.5.715-727>
94. Obriki, O. D., & Arumosoye, O. M. (2018). Conceptual modeling of data-driven occupational safety risk control in large-scale energy infrastructure projects. *Iconic Research and Engineering Journals*, 1(7), 169-189. <https://doi.org/10.64388/IREV1I7-1714414>
95. Obukadata, V. O., Ezeako, R., Okoli, I. E., & Alugwe, S. (2024). AI-enabled synthesis of stakeholder and survey data in operations research: A review of advances and applications. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3539-3563. <https://doi.org/10.62225/2583049X.2024.4.6.6931>
96. Ogbete, J. C., Aminu-Ibrahim, A. Y., & Ambali, K. B. (2022). Design standards and operational planning frameworks for scalable blood collection networks. *Gyanshauryam, International Scientific Refereed Research Journal*, 5(6), 267-300. <https://doi.org/10.32628/GISRRJ225635>
97. Ogbete, J. C., Aminu-Ibrahim, A. Y., & Ambali, K. B. (2023). Lifecycle performance



- evaluation of purpose built diagnostic laboratories supporting long term healthcare delivery. *International Journal of Advanced Multidisciplinary Research and Studies*, 3(6), 2605-2621. <https://doi.org/10.62225/2583049X.2023.3.6.5698>
98. Ogbete, J. C., Aminu-Ibrahim, A. Y., & Ambali, K. B. (2018). Optimizing laboratory spatial planning strategies to improve diagnostic accuracy, safety, and clinical throughput. *Iconic Research and Engineering Journals*, 2(1), 87-113. <https://doi.org/10.64388/IREV117-1713587>
99. Ogbete, J. C., Aminu-Ibrahim, A. Y., & Ambali, K. B. (2019). Regulatory compliant design systems for molecular and pathology laboratories in highly controlled environments. *Iconic Research and Engineering Journals*, 3(4), 607-631. <https://doi.org/10.64388/IREV314-1713589>
100. Ogbole, J. I., Okoruwa, P. O., Babatope, O. M., & Oyewole, T. (2021). Developing an integrated data visualization model for continuous business performance monitoring and optimization. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 7(5), 441-467. <https://doi.org/10.32628/IJSRCSEIT>
101. Ogbole, J. I., Okoruwa, P. O., Babatope, O. M., & Oyewole, T. (2021). Developing an integrated data visualization model for continuous business performance monitoring and optimization. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 7(5), 441-467. <https://doi.org/10.32628/IJSRCSEIT>
102. Okonkwo, C. S., Ogunwale, O., Mayo, W., & Okeke, O. T. (2021). Framework for regulatory-compliant procurement in high-risk energy environments. *International Journal of Multidisciplinary Research and Growth Evaluation*, 2(6), 595-605. <https://doi.org/10.54660/IJMRGE.2021.2.6.595-605>
103. Okoruwa, P. O., Fadayomi, O., Abolaji, T. O., Edivri, J., Ogbole, J. I., & Akeju, B. (2024). Enterprise cybersecurity trends and threat evolution, advances and emerging research directions. *Global Multidisciplinary Perspectives Journal*, 1(6), 194-204. <https://doi.org/10.54660/GMPJ.2024.1.6.194-204>
104. Ominyi, M., Anichukwueze, C. C., & Uzougbo, N. S. (2024b). A systematic review of organizational readiness for the EU Artificial Intelligence Act in multinational enterprises. *World Journal of Innovation and Modern Technology*, 8(6), 185-203. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg185.203>
105. Ominyi, M. (2021). Toward a conceptual framework for embedding privacy by design in



- corporate restructuring and transaction structuring. *Journal of Accounting and Financial Management*, 7(5), 152-173. <https://doi.org/10.56201/jafm.vol.7.no5.2021.p152.173>
106. Onche, V. O., Adegbite, M. P., & Ogbonna, C. S. (2023). BYOD adoption and information-security risk in university administrative offices: Evidence from Nigerian federal universities. *International Journal of Multidisciplinary Research and Growth Evaluation*, 4(6), 1549-1563. <https://doi.org/10.54660/IJMRGE.2023.4.6.1549-1563>
107. Page, M. J., et al. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71. <https://doi.org/10.1136/bmj.n71>
108. Ramesh, B., & Jarke, M. (2001). Toward reference models for requirements traceability. *IEEE Transactions on Software Engineering*, 27(1), 58-93. <https://doi.org/10.1109/32.895989>
109. Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14, 131-164. <https://doi.org/10.1007/s10664-008-9102-8>
110. Sakyi, J. K., Filani, O. M., Nnabueze, S. B., Okojie, J. S., & Ogedengbe, A. O. (2022b). Developing KPI frameworks to enhance accountability and performance across large-scale commercial organizations. *Journal of Frontiers in Multidisciplinary Research*, 3(2), 81-93. <https://doi.org/10.54660/IJFMR.2022.3.2.81-93>
111. Sanni, J. O., Atima, M. E., & Attah, A. (2022). Systematic review of attribution modeling methods resolving bias in multi-touch journeys. *Shodhshauryam, International Scientific Refereed Research Journal*, 5(1), 304-334. <https://doi.org/10.32628/SHISRRJ247135>
112. Sanni, J. O., & Attah, A. (2023c). Market research frameworks addressing entry barriers within highly regulated industrial service sectors. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 10(2), 904-930. <https://doi.org/10.32628/CSEIT2342439>
113. Sanni, J. O., Iwuanyanwu, U. A., Essien, M. A., Atima, M. E., & Attah, A. (2022). Adaptive control models for AI-driven marketing automation in financial compliance environments. *Shodhshauryam, International Scientific Refereed Research Journal*, 5(1), 243-270. <https://doi.org/10.32628/SHISRRJ247133>
114. Sanni, J. O., Iwuanyanwu, U. A., Essien, M. A., & Attah, A. (2023). Lifecycle-aware marketing automation using federated learning for secure cross-organizational data management. *Gyanshauryam, International Scientific Refereed Research Journal*, 6(6), 337-



364. <https://doi.org/10.32628/GISRRJ236642>
115. Shittu, M. A., Shittu, H. A., Adeleke, O. J., & 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), 25-44. https://doi.org/10.34218/IJIERD_14_02_003
116. Tawose, O. M., Oluwadele, J. F., Ekeocha, A. H., Odumboni, A. A., & Egbeyemi, O. S. (2023). Aflatoxin detection and quantification in poultry feeds available in selected areas in Ekiti State. *FUOYE Journal of Agriculture and Human Ecology*, 7(2). <https://doi.org/10.62923/fuojahe.v7i2.302>
117. Tonoyan, A., Dada, O., & Ayivi-Donkor, S. S. (2023). A conceptual framework for retail asset valuation: Integrating quantitative pricing and multi-unit financial performance prediction. *Journal of Accounting and Financial Management*, 9(12), 218-250. <https://doi.org/10.56201/jafm.v9.no12.2023.pg218.250>
118. Tonoyan, A., Dada, O., & Ayivi-Donkor, S. S. (2022b). Algorithmic process optimization and cost reduction: A review of simulation modeling and financial impact assessment. *Journal of Accounting and Financial Management*, 8(8), 139-169. <https://doi.org/10.56201/jafm.vol.8.no8.2022.p139.169>
119. Tonoyan, A., Dada, O., & Ayivi-Donkor, S. S. (2024b). Real-time KPI tracking systems: A review of automated performance monitoring and data-driven decision making. *World Journal of Innovation and Modern Technology*, 8(6), 247-281. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg247.281>
120. Uchechi, N., Ominyi, M., Anichukwueze, C. C., & Uzougbo, N. S. (2022). Advances in quantitative evaluation models for measuring corporate diversity and inclusion investments. *World Journal of Finance and Investment Research*, 6(2), 56-101. <https://doi.org/10.56201/wjfir.v6.no2.2022.pg56.101>
121. Uchechi, N., Ominyi, M., & Jones, B. C. (2018). Procedural standardization and workflow optimization in high-volume dispute resolution clinics: A systematic review. *Iconic Research and Engineering Journals*, 2(6). <https://doi.org/10.64388/IREV2I6-1722789>
122. Uchechi, N., & Ominyi, M. (2019). Liability mitigation and dispute prevention in high-value energy service contracts. *Iconic Research and Engineering Journals*, 3(6). <https://doi.org/10.64388/IREV3I6-1722790>
123. Wedraogo, L., & Sanni, J. O. (2024). Machine learning models addressing uncertainty in



- cross-channel campaign performance forecasting accuracy. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 2875-2890. <https://doi.org/10.62225/2583049X.2024.4.6.5648>
124. Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer. <https://doi.org/10.1007/978-3-662-43839-8>
125. World Bank. (2014). *Global Financial Development Report 2014: Financial Inclusion*. World Bank. <https://doi.org/10.1596/978-0-8213-9985-9>
126. Yusuf, H. D., & Abdulrazaq, K. (2018). A conceptual framework for strengthening diagnostic quality systems in resource-limited clinical laboratories. *Iconic Research and Engineering Journals*, 1(11), 145-179. <https://doi.org/10.64388/IREV1111-1723090>
127. Zhu, H., Hall, P. A. V., & May, J. H. R. (1997). Software unit test coverage and adequacy. *ACM Computing Surveys*, 29(4), 366-427. <https://doi.org/10.1145/267580.267590>