

## Comparative Evaluation of Selenium Cypress and Tricentis Tosca for Enterprise Test Automation

Aisha I. Abdulwahab<sup>1\*</sup>, Serif Oyindamola Oyesiji<sup>2</sup>, Toussida Fatah Tanguy Minoungou<sup>3</sup>

<sup>1</sup>Cummins Inc., Indiana, USA

<sup>2</sup>FrieslandCampina WAMCO, Lagos, Nigeria

<sup>3</sup>DonorsChoose, New York, New York, USA

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

### Abstract

**Purpose:** Enterprises choosing among Selenium WebDriver, Cypress and Tricentis Tosca routinely rely on feature checklists and short demonstrations that ignore portfolio coverage, maintenance under change, workforce skills, governance and exit cost. This article develops a transparent decision model for that choice. **Method:** A critical integrative review of software testing theory, empirical software engineering methodology and technical documentation for the three tools informs the Enterprise Test Automation Selection (ETAS) model. ETAS applies non-compensatory knock-out gates, then weights six criteria through pairwise comparison with consistency checks, and aggregates normalized ratings. Three illustrative enterprise archetypes demonstrate the mechanics, followed by one-at-a-time crossover analysis and Dirichlet rank-acceptability simulation. **Findings:** The tools differ in architecture rather than quality. Selenium offers protocol-level reach across browsers and languages; Cypress offers fast, retry-based feedback for JavaScript teams inside the browser; Tosca offers model-based, codeless coverage of SAP and packaged applications under commercial licensing. In the illustrative cases, rankings were decisive for the JavaScript and SAP archetypes but fragile for a polyglot regulated bank, where the leading tool changed once maintainability or governance weight passed identifiable thresholds. **Contribution:** The article supplies a gated selection model, an explicit rating rubric, six falsifiable propositions and a staged validation design. **Implications:** Selection should be documented as a weighted, sensitivity-tested decision that is re-examined as the application portfolio changes.

**Keywords:** Selenium WebDriver; Cypress; Tricentis Tosca; test automation tool selection; analytic hierarchy process; sensitivity analysis; model-based testing

**DOI:** 10.66592/jxw2qz26

### 1 Introduction

Test automation tool selection is one of the few engineering decisions that binds an enterprise for years while being made in weeks. A tool choice determines which languages test engineers must write, which applications can be reached at all, how quickly a failing build reports back to a developer, how audit evidence is produced and how expensive it will be to leave. Three tools dominate the conversation in large organizations and represent three distinct architectural bets. Selenium WebDriver drives browsers from outside through a standardized remote-control protocol. Cypress executes test code inside the browser alongside the application under test. Tricentis Tosca abstracts the application into reusable technical models and lets testers assemble test cases without writing code. Classic testing texts have long warned that the value of any testing effort depends on the fit between technique, system and people rather than on the instrument itself (Myers et al., 2011; Kaner et al., 1999; Ammann et al., 2016). That warning applies with particular force to tool selection, where vendor comparisons and community benchmarks tend to report a single winner.

#### 1.1 Problem Context and Motivation

Enterprise portfolios are heterogeneous. A single insurer or manufacturer may run a React customer portal, a Java back office, a Salesforce instance, SAP GUI transactions, SAP Fiori apps and a mainframe terminal, all inside one business process such as order to cash. No single tool reaches every layer with equal ease, and each tool embodies a different theory of who writes tests. Selection therefore behaves like a vendor evaluation problem with several incommensurable criteria, the kind of problem

that data-driven vendor assessment and bid selection models were built to structure (Ahiaeke Patrick et al., 2021). It also behaves like a technology adoption problem, because the productivity promised by any tool is realized only when the workforce is ready to use it (Akhigbe et al., 2024; Afrihyia et al., 2025). Change programmes that ignore this readiness dimension in digital transformation frequently stall after the pilot (Eyetsemitan et al., 2023a). Finally, selection is a governance problem: test evidence feeds release approval, and weak evidence propagates into weak decisions at the executive level (Akhigbe et al., 2025; Adegbite et al., 2025).

The practical consequence is that tool debates inside enterprises often become proxy disputes between communities. Developers favour tools that live in their editor and language; quality assurance centres favour tools that business analysts can operate; procurement favours predictable licensing; and architecture teams favour open standards that reduce lock-in. Without an explicit decision model these preferences are negotiated implicitly, and the loudest constituency wins. Cognitive research on managerial judgment shows that anchoring, availability and confirmation effects are strongest in exactly such multi-attribute, ambiguous choices (Fobellah, 2025a). Scenario-based modelling has been proposed in financial planning as a corrective that forces decision makers to state assumptions and examine how outcomes shift under alternative futures (Filani et al., 2023). The present article brings that discipline to test automation.

## 1.2 Research Gap, Questions and Contributions

Published comparisons of Selenium, Cypress and Tosca fall into three groups: practitioner feature matrices, small laboratory experiments that implement a handful of scripts against a demonstration application, and vendor-sponsored analyses. The first group rarely states weights, the second rarely observes maintenance over successive releases, and the third rarely discloses the scoring method. Structured comparative frameworks do exist for adjacent enterprise tooling decisions, for example the evaluation of Copado, Flossum and Salesforce DX across deployment contexts (Badmus et al., 2020), but no equivalent has been specified for functional test automation tools. Across all three, the evidence standard recommended for empirical software engineering, with explicit design selection, construct definition and threats-to-validity reporting, is seldom met (Easterbrook et al., 2008; Runeson et al., 2009). As a result the field lacks a transparent, reproducible method that converts documented tool properties and organizational context into a defensible recommendation and that states when the recommendation would change. Research on non-functional requirements shows why this matters: qualities such as maintainability and portability are context-relative and cannot be ranked in the abstract (Mairiza et al., 2010).

Three research questions follow. RQ1 asks which architectural properties of Selenium WebDriver, Cypress and Tricentis Tosca materially determine their fit to enterprise automation needs. RQ2 asks how those properties can be combined with organizational context in a transparent multi-criteria model whose weights and ratings are auditable. RQ3 asks how robust the resulting rankings are to plausible changes in stakeholder weights, and which propositions about tool fit can be tested empirically. The article makes four contributions. First, it provides an architecture-level comparison of the three tools that separates documented behaviour from contested claims. Second, it specifies the Enterprise Test Automation Selection (ETAS) model, which combines knock-out gates with pairwise weighting and a rating rubric. Third, it demonstrates crossover and rank-acceptability sensitivity analysis on three illustrative archetypes. Fourth, it states six falsifiable propositions with a staged validation design, consistent with design science expectations for artefact evaluation (Hevner et al., 2004; Wieringa, 2014).

## 2 Theoretical Grounding

Three bodies of theory frame the ETAS model. Multi-attribute decision theory supplies the aggregation logic and the requirement that trade-offs be made explicit. Task-technology fit and sociotechnical adoption theory explain why the same tool produces different outcomes in different teams. Software testing theory, particularly the concepts of test adequacy, test oracles and nondeterministic failure, defines what a tool must actually deliver. Each is used below to derive a specific model feature rather than as background. The combination reflects the design science position that an artefact should be justified by kernel theories and evaluated against the problem it claims to solve (Hevner et al., 2004; Wieringa, 2014).

### 2.1 Multi-Attribute Utility and the Analytic Hierarchy Process

Multi-attribute utility theory holds that when preferences over criteria are mutually independent, the overall value of an alternative can be represented as a weighted sum of single-criterion values. The analytic hierarchy process operationalizes the

weights by asking decision makers to compare criteria in pairs on a ratio scale from 1 (equal importance) to 9 (extreme importance), then deriving weights from the principal eigenvector of the resulting reciprocal matrix. Its practical attraction for tool selection is that pairwise judgments are easier to elicit from engineers and managers than direct percentage weights, and that the method produces a consistency diagnostic that exposes contradictory judgments. Comparable weighted models have been applied to vendor evaluation, site selection and asset valuation, where quantitative and qualitative criteria must coexist (Ahiaeke Patrick et al., 2021; Tonoyan et al., 2022a; Tonoyan et al., 2023). The theory also imposes a warning. Weighted sums are fully compensatory: a very high score on cost can offset an inability to reach a required application. ETAS therefore places non-compensatory gates before aggregation, a design choice derived directly from this limitation of additive utility.

## 2.2 Task-Technology Fit and Sociotechnical Adoption

Task-technology fit theory argues that performance gains follow from the correspondence between task demands, individual abilities and tool functionality, not from tool capability alone. Sociotechnical theory adds that social structures, such as who owns the test suite and who is rewarded for keeping it green, co-determine whether a technical capability is used. For test automation this implies that a code-first tool assigned to a business-analyst testing centre, or a codeless tool imposed on a developer team that reviews tests in pull requests, will underperform its technical potential. Organizational research on technology adoption and employee readiness supports this reading (Akhigbe et al., 2024), as does work on structured mentorship, which finds that capability transfer on the job depends on deliberate coaching rather than exposure (Amarahobu et al., 2025a). Evidence from cross-functional training in engineering teams likewise shows error reduction when specialists learn adjacent disciplines (Anichukwueze et al., 2022). Simulation-based instruction, which lets learners rehearse decisions in a safe environment before applying them, offers one route to closing such gaps (Ebhojie et al., 2023b). ETAS encodes this theory as criterion C4, skills and authoring-model fit, and treats it as a first-class criterion rather than a tie-breaker.

## 2.3 Testing Theory: Adequacy, Oracles and Nondeterminism

Testing theory defines what automation is for. Adequacy criteria specify when a test set is sufficient relative to a coverage model, and the literature on unit test coverage and adequacy established that different criteria subsume each other in a partial order rather than a single scale (Zhu et al., 1997). Ammann et al. (2016) generalize this into graph, logic, input-space and syntax-based coverage, which is relevant because Tosca's test case design module supports combinatorial input-space methods natively while Selenium and Cypress leave test design to the engineer. Oracle theory adds that an automated check is only as good as its expected-result specification. Nondeterminism, the phenomenon of tests that pass and fail on the same code, undermines both: a flaky end-to-end suite is an unreliable oracle. Surveys of unit testing practice already identified maintenance and nondeterminism as leading practitioner problems (Daka et al., 2014), and the process standard ISO/IEC/IEEE 29119-2 requires that test completion criteria and incident handling be defined in advance (ISO/IEC/IEEE, 2018). ETAS derives criteria C2 (reliability and feedback speed) and C3 (maintainability under change) from these concepts.

## 3 Review Protocol

The study is a critical integrative review rather than a systematic review with meta-analysis. Integrative review is appropriate because the relevant evidence is heterogeneous: normative standards, textbooks on testing theory, empirical software engineering methodology, peer-reviewed studies of test prioritization and traceability, and primary technical documentation that defines how each tool works. The protocol nevertheless follows the transparency principles of PRISMA 2020 for search, eligibility and reporting (Page et al., 2021) and draws its appraisal logic from guidance on selecting empirical methods and on systematic reviews in software engineering (Easterbrook et al., 2008; Dybå et al., 2008). Table 1 summarizes the protocol decisions, and Figure 1 shows how appraised evidence flows into model construction.

### 3.1 Sources, Search Strings and Time Window

Bibliographic searches covered IEEE Xplore, the ACM Digital Library, Scopus, Web of Science and Google Scholar, with a publication window from 1994 to 2025. The lower bound captures foundational work on requirements traceability, which remains the conceptual basis for linking automated tests to requirements and risk (Gotel et al., 1994; Ramesh et al., 2001). Search strings combined tool terms with evaluation terms, for example ("Selenium" OR "WebDriver" OR "Cypress" OR "Tosca" OR "model-based test automation") AND ("comparison" OR "evaluation" OR "selection" OR "flaky" OR

"maintenance" OR "total cost"), and separately ("test automation" AND ("analytic hierarchy process" OR "multi-criteria" OR "decision model")). Technical documentation was consulted for the W3C WebDriver specification, the Selenium project documentation, the Cypress documentation and the Tricentis Tosca documentation, and was used only to establish how features are designed to behave. Governance and cost literature from adjacent enterprise domains was retained where it informed a criterion definition (Adesuyi et al., 2023; Tonoyan et al., 2024b).

## 3.2 Eligibility, Screening and Appraisal

Sources were eligible when they informed at least one of the following: an architectural property of one of the three tools, a criterion used to evaluate automation tools, a method for weighting or aggregating criteria, or a validity threat in tool comparison. Screening proceeded in two stages, title and abstract followed by full text, with reasons for exclusion recorded. Marketing pages without technical specification, tutorials that restated documentation, and benchmark claims that did not disclose tool versions, hardware and application under test were excluded as evidence of performance, although they could be used to identify claims for adjudication. Appraisal applied five questions adapted from case study and empirical methods guidance (Runeson et al., 2009; Easterbrook et al., 2008): whether the tasks compared across tools were functionally equivalent, whether tool and browser versions were disclosed, whether flakiness was measured by repeated execution rather than single runs, whether maintenance was observed across real application changes, and whether authors disclosed vendor affiliation. Evidence failing the first two questions was coded as low weight regardless of its conclusion.

## 3.3 Synthesis Logic and Model Construction

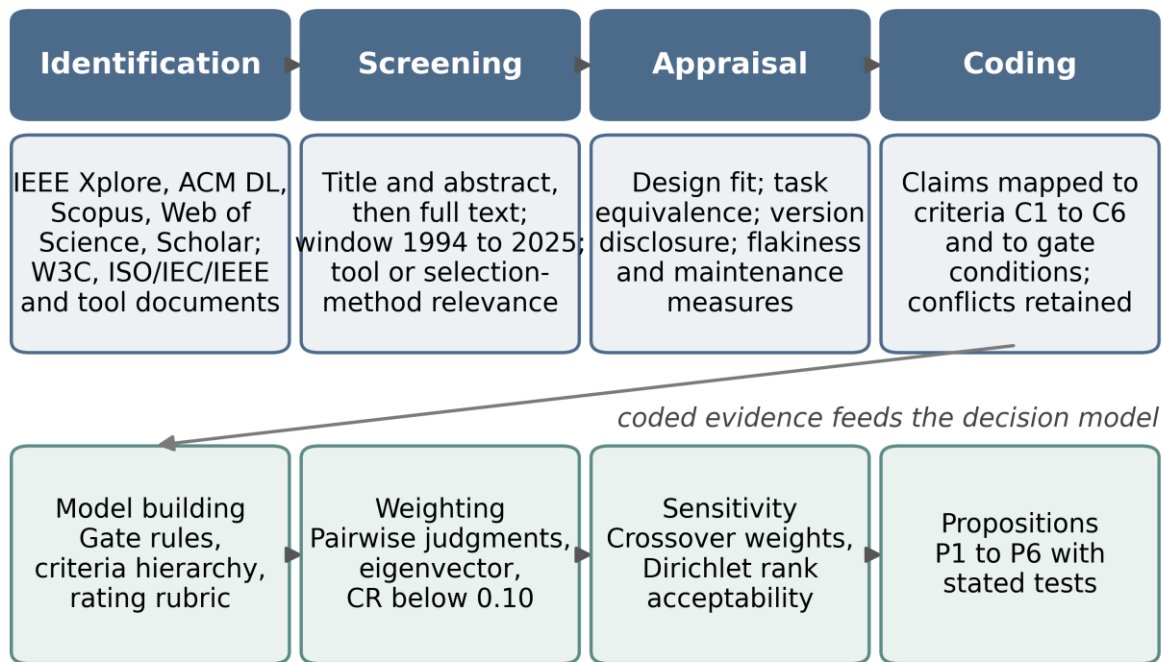
Synthesis used a claim-and-counterclaim logic. Each substantive claim about a tool, such as that Cypress tests are less flaky or that Tosca reduces maintenance, was recorded with the mechanism offered, the conditions under which it was observed and any contradicting evidence. Contradictions were retained and explained by conditions rather than averaged away, following the practice of integrative reviews that preserve heterogeneity as information (Dybå et al., 2008). Coded claims were then mapped to the six ETAS criteria or to gate conditions. Where evidence was insufficient to rate a tool on a criterion in general, the rating was made conditional on the scenario, which is why the illustrative ratings in Section 5 differ across archetypes. The model construction step followed the design science cycle of building and evaluating an artefact against stated requirements, with evaluation here limited to analytical demonstration and sensitivity testing (Wieringa, 2014). Contrasting evidence from quality assurance in other regulated sectors, where assurance is treated as a determinant of supply reliability rather than an inspection afterthought, informed the governance criterion (Asiedu et al., 2024b; Yusuf et al., 2018).

Table 1 lists the protocol elements and the operational rule applied to each.

**Table 1. Review protocol for the comparative evaluation of Selenium WebDriver, Cypress and Tricentis Tosca**

| Protocol element | Operational rule applied                                                                                   | Safeguard against bias                                                |
|------------------|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| Question focus   | Architectural properties, selection criteria, weighting methods and validity threats for the three tools   | Claims unrelated to a criterion or gate excluded with reason          |
| Sources          | IEEE Xplore, ACM DL, Scopus, Web of Science, Google Scholar; W3C, ISO/IEC/IEEE and tool documentation      | Documentation used for designed behaviour only, never for performance |
| Time window      | 1994 to 2025                                                                                               | Tool claims checked against the version current at publication        |
| Search strings   | Tool terms AND evaluation terms; test automation AND multi-criteria terms                                  | Strings logged; backward citation checking                            |
| Screening        | Title and abstract, then full text; two-stage exclusion log                                                | Undisclosed-version benchmarks coded as low weight                    |
| Appraisal        | Task equivalence, version disclosure, repeated-run flakiness, longitudinal maintenance, vendor affiliation | Vendor-affiliated evidence flagged and not used alone                 |
| Synthesis        | Claim, mechanism, condition and counterclaim coding mapped to C1 to C6 and gates                           | Contradictions explained by conditions, not averaged                  |

| Protocol element | Operational rule applied                                             | Safeguard against bias                             |
|------------------|----------------------------------------------------------------------|----------------------------------------------------|
| Output           | Gated AHP model, rubric, sensitivity analysis, propositions P1 to P6 | Illustrative ratings labelled as analyst judgments |



Evidence appraisal (upper row) precedes model construction and testing (lower row).

**Figure 1. Review and model-construction pathway: evidence identification, screening, appraisal and coding (upper row) feed gate definition, weighting, sensitivity analysis and proposition formulation (lower row)**

Figure 1 makes the separation between evidence appraisal and model construction visible. A source enters the model only after it has been coded to a criterion or gate; inclusion in the review does not by itself confer evidential weight. This separation matters most for tool performance claims, which are abundant online but rarely meet the version-disclosure and task-equivalence requirements of Table 1. The same separation between screening signals and validated measures is standard in applied measurement fields, where a marker associated with risk in a cross-sectional sample is not treated as an estimate of that risk for an individual case (Amadi et al., 2025), and where a detection assay is validated before its readings drive decisions (Tawose et al., 2023).

## 4 Architectural Profiles of the Three Tools

This section answers RQ1 by describing how each tool works, because most contested claims about speed, reliability and maintenance trace back to architecture. The descriptions concern designed behaviour documented by the projects and vendor; they are not performance measurements. The testing literature has long distinguished between a tool's mechanism and the outcomes observed when people use it (Kaner et al., 1999), and that distinction organizes the profiles. Traceability research is also relevant, because the tools differ sharply in how easily test artefacts can be linked to requirements and risks (Cleland-Huang et al., 2012). Table 2, at the end of the section, condenses the comparison by criterion.

## 4.1 Selenium WebDriver: Protocol-Level Remote Control

Selenium WebDriver controls a browser from an external process through a browser-specific driver, such as chromedriver, geckodriver, msedgedriver or Apple's safaridriver. The client library sends commands over HTTP using the WebDriver protocol, which the World Wide Web Consortium published as a Recommendation in 2018; Selenium 4 adopted the W3C protocol as its native wire format. Because the protocol is language-neutral, official bindings exist for Java, Python, C#, Ruby and JavaScript, and the same protocol underpins mobile automation through Appium. Selenium Grid distributes sessions across machines; in Grid 4 the architecture is decomposed into a router, distributor, session map, session queue, event bus and nodes, and can run in containers. Selenium Manager, bundled since release 4.6, resolves and downloads matching drivers automatically. The project is open source under the Apache 2.0 licence. Architecturally, Selenium is a library rather than a framework: test runners, assertion libraries, reporting and design patterns such as the page object are chosen by the team. This gives polyglot enterprises reach and freedom, and it places the burden of framework engineering on them, consistent with the long-standing observation that test tooling transfers effort rather than removing it (Myers et al., 2011).

Synchronization is the main source of Selenium's reputation for flakiness. Because commands cross a process boundary and the browser renders asynchronously, a test must wait for application state explicitly. Selenium offers implicit waits, which apply a global timeout to element lookup, and explicit waits, which poll for a named condition such as element visibility or clickability. The documentation advises against mixing the two because the interaction produces unpredictable timeouts. Errors such as the stale element reference exception arise when the page re-renders between lookup and action. These are well-understood engineering problems with well-understood remedies, which is why teams with mature Selenium frameworks can report stable suites while teams without them report instability. The WebDriver BiDi effort, a bidirectional protocol under W3C development, and Selenium's access to Chrome DevTools Protocol features extend event listening and network interception, narrowing a historical gap with in-browser tools. Organizational learning models that link maturity to continuous performance improvement describe this pattern well: capability is accumulated through deliberate practice, not purchased (Arumosoyle et al., 2021).

## 4.2 Cypress: In-Browser Execution and Automatic Retry

Cypress takes the opposite architectural position. Test code runs inside the browser in the same run loop as the application, while a Node.js server process handles tasks that require operating system access and acts as a proxy for network traffic. Commands are enqueued and executed asynchronously, and queries and assertions are retried automatically until they pass or a timeout elapses; the default command timeout is four seconds. This design removes much of the explicit wait code that Selenium tests require and explains why developers frequently report that Cypress suites are easier to stabilize. The Cypress runner records DOM snapshots for each command, allowing a developer to step back through test execution, and the `cy.intercept` command stubs and inspects network requests directly. Tests are written in JavaScript or TypeScript only. Supported browsers are Chrome-family browsers, Firefox and Electron, with WebKit support marked experimental; there is no native Safari driver. The application is open source under the MIT licence, while parallelization orchestration, analytics and test replay are offered through the commercial Cypress Cloud service.

The same architecture imposes limits. Because tests execute within the browser security model, multiple tabs and multiple browser windows cannot be driven in one test. Before version 12, released in late 2022, a single test could not visit pages on more than one superdomain, which made single sign-on flows through an external identity provider awkward; the `cy.origin` command, introduced experimentally in version 9.6 and generally available from version 12, now supports cross-origin steps but requires explicit code blocks and serialization of values between origins. Iframes require plugins or workarounds. For enterprise portfolios with identity federation, embedded third-party widgets or Safari-heavy customer bases, these properties are gating questions rather than minor inconveniences. The Cypress case illustrates a general principle from technology adoption research: a tool's advantages are strongest for the population it was designed for and weaken at the organizational periphery (Akhigbe et al., 2024; Eyetsemitan et al., 2024a). Cypress also offers component testing for React, Vue, Angular and Svelte, which moves some checks down the test pyramid and changes the economics of end-to-end coverage.

## 4.3 Tricentis Tosca: Model-Based, Codeless Automation

Tricentis Tosca is a commercial platform built around model-based test automation. Instead of writing scripts, a tester scans the application with the XScan tool to create Modules, which are technical models of screens and controls with their identifying properties. Test cases are assembled from Module attributes with action modes such as Input, Verify and WaitOn, and parameterized through the TestCase-Design section, which supports combinatorial methods including linear expansion and pairwise combination. When a control changes, the Module is updated once and the change propagates to every test case that uses it, which is the mechanism behind Tosca's maintainability claims. Tosca provides engines for web browsers, SAP GUI through SAP's scripting interface, SAP Fiori, Salesforce, desktop technologies such as Windows Forms and WPF, Java applications, mainframe terminals and APIs, and it adds image recognition through Vision AI for surfaces without an accessible object model. Distributed execution runs test events on agents. The platform is licensed commercially, typically per user or on a concurrent basis, and SAP has offered Tricentis-based testing within its own portfolio. Tosca's requirements section weights requirements by frequency and damage class to produce risk-based coverage views, a direct implementation of the traceability principle that tests should be linked to requirements and prioritized by risk (Gotel et al., 1994; Cleland-Huang et al., 2012).

The codeless model changes who can automate. Business analysts and functional testers who understand the order-to-cash or procure-to-pay process can build end-to-end tests across SAP, Salesforce and web layers without learning a programming language. Salesforce environments that handle sensitive data under role-based security are a representative case, because test data and permissions must be modelled together (Badmus et al., 2024). The trade-offs are equally concrete. Test assets live in a proprietary repository rather than in the application's source control, which complicates code review and branching practices that developer teams take for granted. Licence cost is recurring and scales with users, and migration away from Tosca requires rebuilding test logic in another form. Research on privacy-by-design and data sensitivity classification is relevant here because centralised test repositories frequently hold production-like data that must be classified and masked (Ominyi, 2021; Aliliele et al., 2024a). Enterprise data loss prevention architectures apply equally to test data stores (Aliliele et al., 2025b).

## 4.4 Adjudicating Contested Claims

Three claims recur in comparisons and deserve adjudication. The first is that Cypress is faster than Selenium. The mechanism is plausible, since in-browser execution removes a network round trip per command, but published figures seldom control for test design, parallelization, browser startup and CI hardware, and Selenium Grid parallelism can dominate per-command latency at suite level. The defensible conclusion is that Cypress typically shortens the local feedback loop for a developer, while suite-level wall-clock time depends on infrastructure. The second claim is that Selenium is flaky. The mechanism is again real, because explicit synchronization is the tester's responsibility, but nondeterminism also arises from test order dependence, shared test data, animation and backend latency, which affect all three tools. Research on anomaly detection in time series suggests treating flaky failure rates as signals to be monitored over time rather than as fixed tool attributes (Atakpa et al., 2023). The third claim is that Tosca eliminates maintenance. Model reuse does reduce the number of places a change must be made, but it does not remove the need to re-scan changed screens, update test data or redesign flows after business process changes. Interpretable failure prediction work in other engineering domains makes the same point: prediction helps allocate maintenance, it does not abolish it (Komi et al., 2021).

A fourth, less discussed claim concerns artificial intelligence features. All three ecosystems now advertise some form of self-healing locators, AI-assisted test generation or visual recognition. Reviews of artificial intelligence in software testing show real progress in test generation, prioritization and oracle support, but also warn that evidence often comes from controlled settings (Borah et al., 2022). Enterprises considering agentic or generative test tooling must also address new security exposures, including prompt injection and tool poisoning in automated workflows (Nwakamma et al., 2024b), and regulatory readiness obligations under the EU Artificial Intelligence Act where such tools touch regulated processes (Ominyi et al., 2024b). Multi-agent quality control for regulated documents offers a transferable lesson: AI review increases throughput only when a human accountability layer remains (Ominyi et al., 2025a). ETAS therefore treats AI features as modifiers of C3 ratings that must be evidenced, not as a separate criterion.

Table 2 consolidates the architectural comparison by criterion. Each cell states the designed behaviour and the main condition under which that behaviour becomes an advantage or a constraint.

**Table 2. Architecture-level comparison of Selenium WebDriver, Cypress and Tricentis Tosca against the six ETAS criteria**

| Criterion                     | Selenium WebDriver                                                                                   | Cypress                                                                                                    | Tricentis Tosca                                                                                      |
|-------------------------------|------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| C1 Technology coverage        | Any W3C-compliant browser incl. Safari; mobile via Appium; no SAP GUI or desktop without other tools | Chrome-family, Firefox, Electron; WebKit experimental; single tab; component tests for major JS frameworks | Web, SAP GUI and Fiori, Salesforce, desktop, Java, mainframe, API; Vision AI for opaque surfaces     |
| C2 Reliability and feedback   | Stability depends on explicit waits and framework design; Grid parallelism shortens suite time       | Automatic retry of queries and assertions; time-travel debugging; fast local loop                          | Engine-level synchronization; distributed execution; feedback slower for developers outside the tool |
| C3 Maintainability            | Page objects and code review; locator strategy is team responsibility                                | Co-located with app code; refactoring with IDE tools; cross-origin steps add code                          | Module reuse propagates control changes; re-scan and data upkeep still required                      |
| C4 Skills and authoring       | Java, Python, C#, Ruby, JavaScript; engineering skill needed                                         | JavaScript or TypeScript only; natural for front-end developers                                            | Codeless; suits analysts and functional testers; code-first teams may resist                         |
| C5 Integration and governance | Any CI system; reporting assembled by team; traceability via external tools                          | CI integration; commercial cloud for orchestration and analytics                                           | Requirements risk weighting, built-in traceability and reporting; proprietary repository             |
| C6 Ownership and exit cost    | Apache 2.0; cost is labour and infrastructure; low exit cost                                         | MIT core; optional paid cloud; exit requires rewrite for non-JS stack                                      | Commercial per-user or concurrent licence; high exit cost from proprietary models                    |

## 5 The ETAS Decision Model

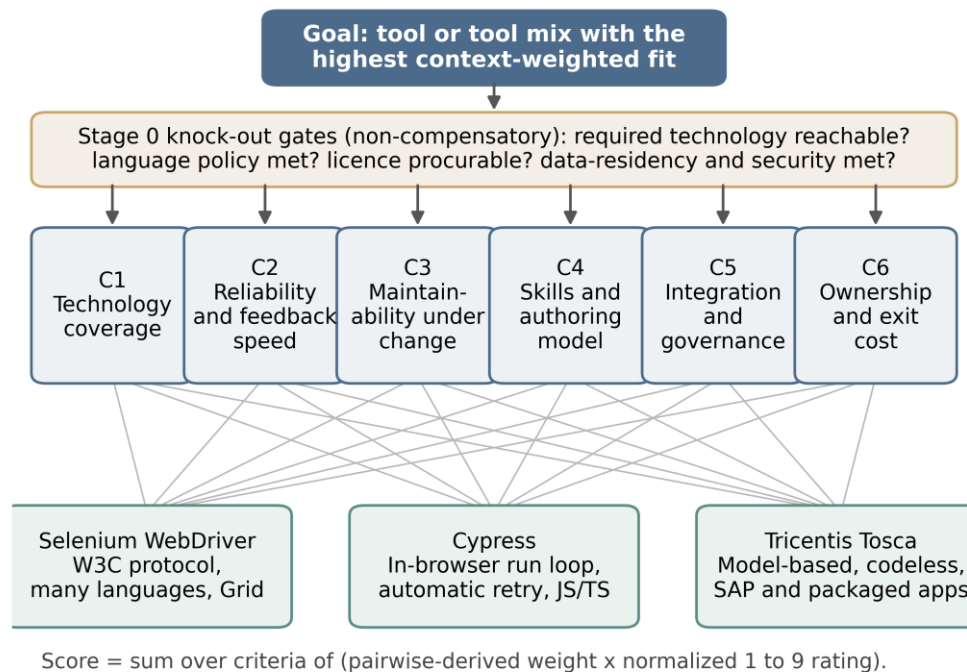
This section answers RQ2 by specifying the model. ETAS has four components: knock-out gates, a criteria hierarchy with operational definitions, pairwise weighting with consistency checking, and a rating rubric with additive aggregation. The components are sequenced so that non-negotiable constraints are handled before trade-offs are made, a principle drawn from the limitation of compensatory aggregation discussed in Section 2.1 and from risk governance architectures that separate hard controls from weighted risk appetite (Adegbite et al., 2025; Falemi et al., 2024). The model is deliberately simple enough to be rebuilt in a spreadsheet, because a selection method that only a specialist can operate reproduces the opacity it is meant to remove. Figure 2 displays the hierarchy.

### 5.1 Knock-Out Gates and Criteria Hierarchy

Stage 0 applies four gates, each answered yes or no for each tool. The reach gate asks whether every application layer that must be automated in scope can be reached by the tool alone or with an approved complementary tool; a portfolio dominated by SAP GUI transactions fails Cypress and unassisted Selenium at this gate. The language gate asks whether the tool's authoring model is permitted by the organization's engineering standards. The procurement gate asks whether licensing can be obtained within budget cycles and vendor risk policy. The security gate asks whether the tool's execution and data flows meet data-residency, network segmentation and access-control requirements; security architecture work on IT and OT convergence, zero trust and access control policy supplies the relevant control language (Adegbite et al., 2022; Ahmed et al., 2021; Ladapo et al., 2023b). A test grid is a network service like any other, and exposed automation endpoints should be protected against misuse in the way any exposed service is protected against amplification and intrusion attempts (Jimoh et al., 2024; Dosunmu et al., 2020). Tools that pass all gates proceed to weighting.

The criteria are defined as follows. C1, technology coverage, is the share of in-scope application layers, weighted by business-process criticality, that the tool can automate natively. C2, reliability and feedback speed, combines the flaky-failure rate

under repeated execution with the median time from commit to test verdict. C3, maintainability under change, is the effort required to restore a passing suite after a representative set of application changes. C4, skills and authoring-model fit, is the proportion of the intended authoring population able to create and review tests without retraining beyond a defined threshold. C5, integration and governance, covers CI/CD integration, requirements traceability, audit evidence and reporting. Reviews of CI/CD pipeline strategies on enterprise platforms treat automated test stages as integral to deployment outcomes, which is why integration is scored as a criterion rather than assumed (Badmus et al., 2018). C6, ownership and exit cost, covers licences, infrastructure, framework engineering labour and the estimated cost of migration away. The definitions follow the recommendation in traceability research that relationships between artefacts be specified in operational terms (Ramesh et al., 2001) and the requirement in ISO/IEC/IEEE 29148 that requirements be verifiable (ISO/IEC/IEEE, 2018).



**Figure 2. ETAS hierarchy: non-compensatory knock-out gates precede six weighted criteria (C1 to C6), which are rated for each of the three alternative tools**

Figure 2 shows that the gates sit between the goal and the criteria, so that a tool failing a gate never reaches aggregation. The lines between criteria and alternatives indicate that every surviving tool is rated on every criterion. The goal box is phrased as a tool or tool mix because the model can also evaluate combinations, such as Tosca for SAP-centred business processes paired with Cypress for a customer-facing single-page application. Evaluating combinations requires rating the mix as a single alternative, including the governance overhead of operating two toolchains, a cost that lifecycle performance evaluation in infrastructure programmes shows to be persistently underestimated (Ogbete et al., 2023; Aminu-Ibrahim et al., 2025c).

## 5.2 Pairwise Weighting and Consistency

For  $n$  criteria, each stakeholder panel completes the upper triangle of an  $n$  by  $n$  reciprocal matrix  $A$ , where  $a_{ij}$  expresses how much more important criterion  $i$  is than criterion  $j$  on the 1 to 9 scale,  $a_{ji} = 1/a_{ij}$  and  $a_{ii} = 1$ . The weight vector  $w$  is the normalized principal eigenvector of  $A$ , defined by equation (1). Consistency is assessed through the consistency index in equation (2) and the consistency ratio in equation (3), where  $RI$  is the random index for a matrix of order  $n$ , equal to 1.24 for  $n = 6$ . By convention a ratio below 0.10 is acceptable, and larger values trigger a review of the judgments rather than an automatic correction. Elicitation should be done separately by role, for example engineering, quality assurance, architecture and procurement, and then combined by the geometric mean of individual judgments, which preserves reciprocity. Recording the dispersion between roles is as informative as the combined weight, because it exposes the proxy disputes described in

Section 1.1. KPI framework research makes the parallel point that accountability depends on documented, attributable measures (Sakyi et al., 2022b).

$$A\mathbf{w} = \lambda_{\max}\mathbf{w}, \quad \text{with } \sum_j w_j = 1 \quad (1)$$

$$CI = \frac{\lambda_{\max} - n}{n - 1} \quad (2)$$

$$CR = \frac{CI}{RI} \quad (3)$$

Judgment quality deserves attention because pairwise elicitation is vulnerable to the same biases as any expert judgment. Anchoring on a recent production incident can inflate the weight of reliability; a recent licence renewal can inflate the weight of cost. Research on cognitive bias in audit and risk decisions recommends structured elicitation, independent first judgments and explicit rationales (Fobellah, 2025a). Diversity of perspective in the panel also improves judgment quality in multinational project teams (Amayo et al., 2023c), and quantitative evaluation of inclusion practices suggests recording which roles were represented (Uchechi et al., 2022). Where teams are distributed across regions, cross-cultural communication research warns that ratio-scale judgments may be interpreted differently, so a calibration exercise on a neutral example should precede elicitation (Lilian et al., 2020).

### 5.3 Rating Rubric and Aggregation

Each surviving tool receives a rating  $r_{ij}$  on a 1 to 9 scale for each criterion  $j$ , anchored by a written rubric. For C1, a rating of 9 means native automation of all in-scope layers, 5 means native coverage of the majority with approved workarounds for the rest, and 1 means coverage of a minority. For C2, anchors are defined on the flaky-failure rate and median verdict time observed in a pilot; for C3 on repair hours per representative change; for C4 on the proportion of authors productive after a defined onboarding period; for C5 on the presence of native traceability, CI integration and audit-ready reports; and for C6 on five-year cost including exit. The global score of tool  $i$  is given by equation (4), where ratings are divided by 9 so that scores lie between 0 and 1. The additive form is justified only for criteria whose values are preferentially independent; C2 and C3 are correlated in practice, because unstable suites also cost more to maintain, and analysts should check that the same evidence is not counted twice (Zhu et al., 1997). Variance analytics used for cost control provide a practical template for tracking realized against planned values of each rating once a tool is in use (Aliliele et al., 2024).

$$S_i = \sum_j w_j \frac{r_{ij}}{9} \quad (4)$$

Cost deserves a specific note. C6 is rated rather than entered as a currency amount so that it can be traded against other criteria on a common scale, but the rating must be derived from an explicit five-year cost model. That model should include licences and cloud services, grid or agent infrastructure, framework engineering and upkeep labour, training, and an exit estimate. Cost governance research in technology-intensive enterprises emphasizes that labour and coordination costs often exceed visible licence costs (Adesuyi et al., 2023), and financial accountability frameworks in large programmes recommend that each cost line have a named owner (Aminu-Ibrahim et al., 2025a; Eboh et al., 2025). Digital supply chain cost models further show that continuity costs, here the cost of an unsupported framework version or an expired licence, belong in the baseline rather than in contingency (Okonkwo et al., 2024e). Simulation-based cost assessment is a useful way to explore these components before committing (Tonoyan et al., 2022b).

### 5.4 Illustrative Enterprise Archetypes

To demonstrate the model, three archetypes were constructed from recurring enterprise patterns. Archetype A is a digital-native software company whose product is a JavaScript single-page application, with developers owning tests in the same repository and a customer base using Chromium and Firefox browsers. Archetype B is a regulated bank with Java and .NET services, a web banking channel with significant Safari usage, federated sign-on, an internal Selenium framework and strict audit requirements. Archetype C is a manufacturer whose order-to-cash and procure-to-pay processes run across SAP GUI, SAP Fiori and Salesforce, with a testing centre staffed mainly by functional analysts. All three tools pass the gates for A and B; for C, Cypress and Selenium pass the reach gate only on the assumption that SAP GUI steps are handled by another tool, and they are retained here to show how the aggregation behaves when that assumption is made. The pairwise judgments and

ratings were set by the authors from the architectural profiles in Section 4 and are illustrative analyst judgments, not measurements. Their purpose is to show the mechanics and the sensitivity behaviour, following the analytical-demonstration stage of design science evaluation (Hevner et al., 2004). Organizational change frameworks for small and large firms caution that such archetypes must later be replaced by the adopting organization's own evidence (Eyetsemitan et al., 2025).

Table 3 reports the derived weights, the consistency ratios and the global scores. All three matrices are acceptably consistent, with consistency ratios between 0.012 and 0.018.

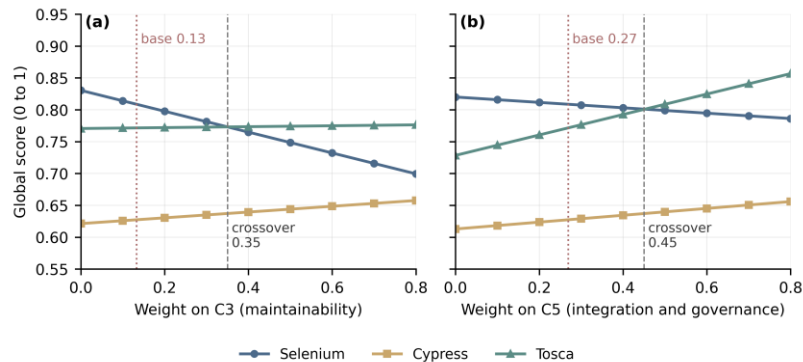
**Table 3. Illustrative ETAS weights, consistency ratios, global scores and rank-1 acceptability for three enterprise archetypes**

| Quantity                         | A: JavaScript SaaS | B: Polyglot regulated bank | C: SAP-centric manufacturer |
|----------------------------------|--------------------|----------------------------|-----------------------------|
| w C1 Coverage                    | 0.069              | 0.281                      | 0.362                       |
| w C2 Reliability and feedback    | 0.262              | 0.122                      | 0.116                       |
| w C3 Maintainability             | 0.157              | 0.133                      | 0.146                       |
| w C4 Skills and authoring        | 0.287              | 0.130                      | 0.175                       |
| w C5 Integration and governance  | 0.082              | 0.269                      | 0.136                       |
| w C6 Ownership and exit cost     | 0.143              | 0.065                      | 0.065                       |
| $\lambda_{max}$ ; CR             | 6.110; 0.018       | 6.087; 0.014               | 6.073; 0.012                |
| Score: Selenium                  | 0.715              | 0.809                      | 0.443                       |
| Score: Cypress                   | 0.903              | 0.627                      | 0.416                       |
| Score: Tosca                     | 0.563              | 0.772                      | 0.907                       |
| Rank-1 acceptability (S / C / T) | 0.00 / 1.00 / 0.00 | 0.95 / 0.00 / 0.05         | 0.00 / 0.00 / 1.00          |

The pattern in Table 3 is informative. In Archetype A, heavy weights on skills fit and feedback speed favour Cypress decisively, and cost weight penalizes Tosca. In Archetype C, coverage dominates and Tosca leads by a wide margin, with Selenium and Cypress close to each other because both lack SAP GUI reach. Archetype B is the contested case. Selenium leads Tosca by only 0.037, because Selenium's coverage of Safari and its fit with Java and .NET engineers are offset by Tosca's governance and traceability strengths. This is the situation in which an unexamined recommendation is most likely to be wrong, and it motivates the sensitivity analysis in Section 5.5. Attribution research in marketing analytics shows the same danger of reading a small margin produced by a particular weighting scheme as a robust ranking (Sanni et al., 2022).

## 5.5 Sensitivity and Rank-Acceptability Analysis

Two sensitivity analyses address RQ3. The first is one-at-a-time crossover analysis. The weight of criterion  $k$  is varied from 0 to 0.8 while the remaining weights are rescaled proportionally,  $w'_j = w_j (1 - w_k) / (1 - w_k)$ , so that they continue to sum to one. The crossover weight is the value at which the leading tool changes. The second is rank-acceptability analysis. Weight vectors are sampled from a Dirichlet distribution centred on the elicited weights with concentration parameter 40, which produces moderate dispersion around each weight, and the share of 20,000 samples in which each tool ranks first is recorded. Figure 3 plots the first analysis for the contested archetype. Scenario-based planning literature treats this kind of exercise as a way of exposing which assumptions a decision actually rests on (Filani et al., 2023), and forecasting research under uncertainty argues for reporting distributions rather than point estimates (Wedraogo et al., 2024).



**Figure 3. One-at-a-time sensitivity for Archetype B: global scores as (a) the weight on maintainability (C3) and (b) the weight on integration and governance (C5) vary, with other weights rescaled proportionally**

Figure 3 shows the two crossovers found for Archetype B. Tosca overtakes Selenium when the maintainability weight rises from its elicited 0.13 to about 0.35, or when the integration and governance weight rises from 0.27 to about 0.45. No single-criterion change within the plotted range lets Cypress lead, because its coverage and skills ratings are low for this archetype. Rank acceptability tells a consistent story: Selenium ranks first in about 95 percent of sampled weight vectors and Tosca in about 5 percent. For Archetypes A and C, the leader ranked first in every sample, and the only crossover found for C required the ownership-cost weight to exceed 0.82, an implausible value. A second form of sensitivity concerns ratings rather than weights. For Archetype B, Tosca would overtake Selenium if its coverage rating rose by about 1.2 points on the 9-point scale, for example if an organization's Safari usage fell or Tosca's browser support for that channel were shown to be equivalent. Decision owners should therefore concentrate evidence gathering on the ratings and weights nearest to a crossover, which is the practical value of the analysis (Tonoyan et al., 2024b).

The analysis also clarifies what a close result means managerially. A small margin with high rank acceptability, as in Archetype B, supports a provisional choice with a defined review trigger, for example re-running the model when the share of SAP or Salesforce traffic in regression scope passes a threshold. A small margin with low rank acceptability would instead support a pilot of both leading tools before commitment. Leading-indicator research in safety governance recommends precisely this pairing of a current decision with monitored signals that would reverse it (Obriki et al., 2019; Obriki et al., 2025). Continuous control approaches to compliance make the same argument for turning periodic decisions into monitored ones (Adebayo et al., 2025).

## 6 Propositions and Validation Design

The ETAS model is a conceptual artefact, and its implications are presented as propositions that have not been tested. Each proposition names the observable that would falsify it and the study design most suited to testing it. Case study and design science guidance both require that such propositions be stated before data collection so that confirmation is not constructed after the fact (Runeson et al., 2009; Wieringa, 2014). Figure 4 arranges the propositions along a staged validation pathway that an adopting enterprise or a research team could follow. The propositions deliberately include conditions under which the intuitive expectation should fail, because a proposition that every outcome confirms carries no information about tool fit.

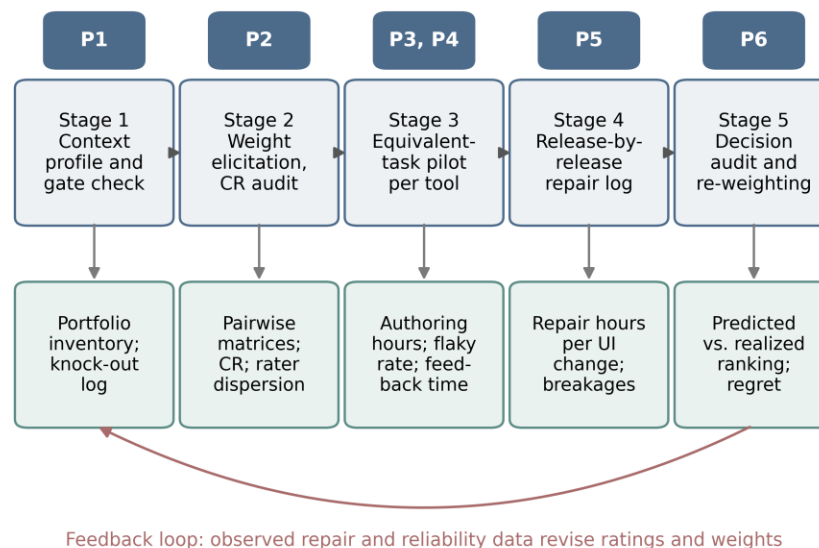
### 6.1 Propositions on Coverage, Weighting and Reliability

P1 (gate primacy): in portfolios where more than a stated share of business-critical process steps run on SAP GUI or other non-browser surfaces, tools that fail the reach gate will show lower end-to-end process coverage after twelve months than tools that pass, regardless of their aggregated scores. P1 is falsified if gate-failing tools, combined with approved complements, achieve equal process coverage at equal cost. P2 (weight dispersion): the dispersion of pairwise weights between engineering and quality assurance roles will predict post-adoption dissatisfaction better than the aggregated score margin. It is falsified if dispersion shows no association with satisfaction or abandonment in a multi-organization sample. P3 (synchronization mechanism): for equivalent end-to-end tasks written by engineers of comparable experience, Cypress suites

will show a lower flaky-failure rate than Selenium suites without a mature explicit-wait framework, but the difference will be absent when the Selenium suite uses a reviewed explicit-wait library. P3 is falsified if the difference persists after controlling for synchronization practice. Test prioritization research shows that measured effects depend heavily on subject programs and fault seeding, which argues for repeated runs across several applications (Catal et al., 2013; Li et al., 2007).

## 6.2 Propositions on Maintenance, Skills and Decision Quality

P4 (authoring population): time to first productive test will be lower for Tosca than for Selenium or Cypress among functional analysts without programming experience, and lower for Cypress than for Tosca among front-end developers. P4 is falsified if the interaction between tool and author background is absent. P5 (model reuse): across a sequence of releases with logged user interface changes, repair hours per change will be lower for Tosca Module-based suites than for script-based suites without page objects, but not lower than for script-based suites with disciplined page objects. It is falsified if Tosca shows a repair advantage over both. P6 (decision quality): enterprises that document gates, weights, crossover points and review triggers will report lower regret, measured as the proportion of decision makers who would choose differently after two years, than enterprises that select by feature comparison. It is falsified if regret does not differ once organization size and portfolio complexity are controlled. Training intervention research outside software shows that capability programmes often yield modest effects unless reinforced by practice, which is why P4 measures time to productivity rather than course completion (Miller et al., 2015; Abah et al., 2025).



**Figure 4. Staged validation pathway for the ETAS model, linking propositions P1 to P6 to study stages and the evidence collected at each stage, with a feedback loop from decision audit to the next context profile**

Figure 4 sequences the tests. Stage 1 profiles the portfolio and records gate outcomes, testing P1. Stage 2 elicits weights by role and audits consistency and dispersion, testing P2. Stage 3 runs an equivalent-task pilot in which the same set of end-to-end scenarios is implemented in each surviving tool, with repeated executions to estimate flaky-failure rates, testing P3 and P4. Stage 4 follows the chosen tool, and ideally a comparison suite, across releases and logs repair effort per interface change, testing P5. Stage 5 audits the decision against realized outcomes and feeds revised ratings into the next cycle, testing P6. Substitution studies in agricultural science illustrate the logic of Stage 3: a candidate replacement is compared with the incumbent on several outcomes under the same conditions rather than on a single headline metric (Oluwadele et al., 2025). Stage 4 reflects lifecycle asset management practice, in which performance metrics are tracked over the whole service life rather than at commissioning (Aminu-Ibrahim et al., 2025c).

## 6.3 Measurement Specification for the Pilot

The pilot in Stage 3 requires careful measurement because its results carry most of the evidential weight. The flaky-failure rate should be estimated as the proportion of test executions that fail on an unchanged commit and pass on rerun, over at least several hundred executions per tool, with the random seed, browser version and test order logged. Feedback time should be reported as the median and 90th percentile of the interval from commit to verdict, since tail latency rather than average latency determines whether developers wait for results. Fault-detection effectiveness can be assessed by seeding representative defects and computing detection rates, and where suites are prioritized, the average percentage of faults detected (APFD) can be computed as  $APFD = 1 - (TF_1 + \dots + TF_m) / (n \cdot m) + 1 / (2n)$ , where  $n$  is the number of tests,  $m$  the number of faults and  $TF_i$  the position of the first test that detects fault  $i$  (Li et al., 2007; Catal et al., 2013). Real-time KPI systems research supports publishing these measures on shared dashboards so that pilot results are visible to every stakeholder role (Tonoyan et al., 2024b). Process mining methods can reconstruct the authoring and repair workflow from tool logs to verify that recorded effort matches actual activity (Sanni et al., 2025).

## 7 Discussion

The review and demonstration support a conditional answer to the question of which tool is best: the answer is determined by portfolio composition, authoring population and governance demands, and it can be made explicit and tested. This section discusses the implications for theory and practice, the strongest counter-arguments, and threats to validity. The discussion follows the recommendation of empirical software engineering methodologists that interpretation be kept distinct from demonstration and that alternative explanations be addressed directly (Easterbrook et al., 2008; Dybå et al., 2008). Because the numerical results are illustrative, the discussion concentrates on the structure of the argument rather than on the particular scores.

### 7.1 Theoretical Implications

The first theoretical implication is that tool architecture, not tool quality, is the proper unit of comparison. Selenium, Cypress and Tosca embody different answers to where test code executes, who writes it and where it is stored, and these answers map onto task-technology fit dimensions. The second implication concerns aggregation theory. The demonstration shows that gates and weights perform different theoretical work: gates encode feasibility, weights encode preference, and conflating them produces misleading compensatory results, as the Archetype C scores for Cypress and Selenium illustrate. The third implication concerns testing theory. Treating flakiness as a property of a tool rather than of a tool, practice and application combination is a category error; the proposition P3 formalizes this and makes it testable. Research on requirements traceability recovery using information retrieval supports a fourth implication, namely that traceability is a capability that can be supplied by tooling external to the test framework, so that C5 ratings for open tools should consider the enterprise's wider toolchain (De Lucia et al., 2007; Borg et al., 2014).

A fifth implication concerns organizational learning. Maturity models built on organizational learning describe capability as a stock that accumulates through feedback on performance (Arumosoye et al., 2021), and safety culture research shows how leadership sustains or erodes such stocks (Obogo et al., 2019a; Obogo et al., 2023; Arumosoye et al., 2024). The ETAS feedback loop in Figure 4 treats test automation capability the same way. This reframes the classic build-versus-buy debate: an open framework is a capability stock the enterprise owns, while a commercial platform is partly a stock the vendor maintains on the enterprise's behalf. Neither is inherently superior, but the choice determines who bears the cost of depreciation when browsers, frameworks and business applications change. Governance models for complex enterprises that stress transparency of strategic execution provide the accountability counterpart to this learning view (Akhigbe et al., 2025).

### 7.2 Practical Implications

For practitioners, the most actionable recommendation is to separate the selection decision into the four ETAS components and to document each. A gate log records why a tool was excluded. A weighting record stores each role's pairwise matrix, the combined weights and the consistency ratio. A rating sheet stores the rubric anchor and evidence for every rating. A sensitivity note states the crossover weights and the review trigger. This package makes the decision auditable and reversible, which matters in regulated sectors where test evidence supports release approval (Eze et al., 2024b). Standard operating procedures

are valuable here because they preserve decision rationale across staff turnover (Eyetsemitan et al., 2022), and handover research shows how much information is lost when rationale is carried only in people's heads (Ekelemu, 2025b). User acceptance testing frameworks for smaller deployments offer a lightweight version of the same discipline (Eyetsemitan et al., 2023b).

A second practical implication is that mixed toolchains are often rational. An SAP-centric enterprise may run Tosca for end-to-end business processes, Cypress for component and single-page application tests owned by developers, and Selenium or Appium where mobile or Safari coverage is required. The cost of this mix lies in governance: separate reporting, duplicated test data management and divided skills. Lean process improvement frameworks recommend mapping the full value stream before adding tools, so that each tool owns a clearly bounded part of the flow (Ambali et al., 2021). Security orchestration research shows that automation across heterogeneous tools yields benefits only when event data are normalized into a common view (Dosunmu et al., 2025c), and the same holds for test results. Finally, where test data derive from production, privacy-preserving data architectures and legal risk models for data protection should govern the test environment as strictly as production (Atakpa et al., 2022; Mbonu et al., 2018).

## 7.3 Counter-Arguments and Boundary Conditions

The gated model invites three objections. The first is that multi-criteria models merely formalize prejudice, since weights and ratings are judgments. The response is that ETAS does not remove judgment; it exposes judgment to review, records who held which view, and shows through sensitivity analysis which judgments matter. The second counter-argument is that the market moves too fast for any rating to remain valid; Cypress's cross-origin support changed materially between versions 9 and 12, and Selenium's BiDi work is still evolving. The response is that ETAS ratings are dated and tied to versions, and the Stage 5 audit is designed to revise them. The third counter-argument is that total cost dominates in practice and the rest is rationalization. The Archetype B and C results show that plausible cost weights seldom change the leader, which suggests that cost disputes often mask disagreements about coverage or skills. Risk-based decision models that balance speed, safety and evidence quality face the same critique and answer it the same way, by making trade-offs explicit (Eze et al., 2024b; Eze et al., 2022b).

Boundary conditions limit where the model applies. ETAS assumes that the portfolio is stable enough over the evaluation horizon for coverage ratings to remain meaningful; during a large platform migration, such as a move from SAP ECC to S/4HANA, both the application and its test surfaces change, and transition-period procedural complexity dominates, as operating-cycle research on start-up, shutdown and mode change illustrates in process industries (Amarahobu et al., 2025b). The model also assumes that the organization can run a pilot; very small teams may lack the capacity, in which case the gate and weighting stages still add value but ratings must rely on documentation. Finally, the model addresses functional and regression automation through user interfaces and APIs; performance, security and accessibility testing require different tools and are outside its scope. Resilience planning for systems under stress offers an analogy: models calibrated on normal conditions should not be trusted unaltered during disruption (Aminu-Ibrahim et al., 2025b).

## 7.4 Threats to Validity

Table 4 lists threats specific to comparative evaluation of automation tools and the mitigation each requires. The categories follow the construct, internal, external and conclusion validity structure recommended for case studies in software engineering (Runeson et al., 2009). Two threats are particularly acute for this topic. Version drift means that any comparison ages quickly, and vendor-sponsored evidence can shape published comparisons in ways that are difficult to detect. Analogous issues arise in regulated public health oversight, where performance metrics must be independently verified before they inform regulatory decisions (Anioke et al., 2023a), and in real-world evidence programmes, where observational data require explicit bias controls (Ilodigwe et al., 2024a).

**Table 4. Threats to validity in comparative evaluation of Selenium WebDriver, Cypress and Tricentis Tosca, with mitigations**

| Validity type | Threat specific to tool comparison                                            | Mitigation in ETAS studies                                      |
|---------------|-------------------------------------------------------------------------------|-----------------------------------------------------------------|
| Construct     | Flakiness measured by single runs; maintainability equated with lines of code | Repeated executions per test; repair hours per logged UI change |
| Construct     | Coverage counted as screens rather than                                       | Weight coverage by business-process                             |

| Validity type | Threat specific to tool comparison                                                   | Mitigation in ETAS studies                                  |
|---------------|--------------------------------------------------------------------------------------|-------------------------------------------------------------|
|               | process steps                                                                        | criticality                                                 |
| Internal      | Author skill confounded with tool (experts in one tool, novices in another)          | Match author experience; record prior tool exposure         |
| Internal      | Non-equivalent tasks across tools, such as skipping cross-origin sign-on in one tool | Shared scenario specification and acceptance oracle         |
| External      | Demonstration applications differ from enterprise SPAs, SAP and Salesforce           | Pilot on in-scope applications; report architecture         |
| External      | Version drift (e.g. cross-origin support, BiDi maturity)                             | Record tool, driver and browser versions; date every rating |
| Conclusion    | Small score margins read as decisive rankings                                        | Report crossover weights and rank acceptability             |
| Conclusion    | Vendor-sponsored or self-selected success stories                                    | Disclose affiliations; include abandoned adoptions          |

The threats in Table 4 interact. A study that uses a demonstration application (external threat) also tends to omit sign-on and embedded widgets (internal threat), which are exactly the areas where Cypress's architecture differs most from Selenium's. Similarly, counting coverage by screens (construct threat) inflates the apparent reach of web-only tools in SAP-heavy portfolios. Forecasting research on fairness shows that aggregate accuracy can conceal systematic error in subgroups (Komi, 2023); in tool comparison, aggregate pass rates can likewise conceal poor performance on the most critical process steps. Quality systems in diagnostic laboratories rely on external quality assessment for this reason, and test programmes benefit from the analogous practice of independent review of pilot design (Yusuf et al., 2018).

## 8 Limitations and Future Research

This review has several limitations. The evidence base on long-term maintenance of Selenium, Cypress and Tosca suites in enterprise settings is thin, and much of it is practitioner reporting without version disclosure; the architectural profiles are therefore more secure than any claim about realized outcomes. The illustrative archetypes use analyst judgments rather than elicited stakeholder weights and measured ratings, so the numerical results demonstrate the model's behaviour and must not be read as recommendations for any real organization. The review excluded newer frameworks such as Playwright except where they informed the interpretation of browser protocols, and a complete market evaluation would include them as alternatives. Finally, the additive model assumes preferential independence, which is only approximately true for reliability and maintainability. These limitations are typical of integrative reviews in software engineering, where heterogeneous designs prevent quantitative pooling (Dybå et al., 2008; Page et al., 2021), and of cost and asset models that must be calibrated locally (Ogbete et al., 2023).

Future research should proceed in four directions. First, multi-organization studies should elicit real pairwise weights by role and test P2, linking weight dispersion to adoption outcomes; organizational readiness research offers instruments that could be adapted (Afrihyia et al., 2025; Ilodigwe et al., 2024b). Second, controlled equivalent-task pilots across several enterprise applications should test P3 and P4 with repeated execution and matched author experience. Third, longitudinal studies following suites across at least a year of releases should test P5, ideally using process mining and anomaly detection on execution logs to separate application change from test fragility (Sanni et al., 2023; Atakpa et al., 2024). Fourth, research should examine AI-assisted authoring and self-healing locators under the same ETAS criteria, with attention to accountability and security in automated workflows (Ilodigwe et al., 2025a; Nwakamma et al., 2024b). Digital twin research on real-time fault detection suggests that pairing a model of the application with execution telemetry could support predictive test maintenance (Shittu et al., 2023), and remote monitoring reviews show how sensor-based surveillance matures from pilot to programme (Dagodzo et al., 2020). Studies of loss detection programmes show the value of systematic attribution when many small failures accumulate (Ekelemu et al., 2025b).

Additional questions concern context. Localization testing across languages raises coverage questions that differ by tool, since text-based locators break under translation while model-based identification may not; comparative translation research in administrative language illustrates how terminology varies systematically across jurisdictions (Kpoka, 2023). Enterprises that also validate hardware, such as wireless products, face a further reach question, because scripting frameworks for instrument

control lie outside the designed scope of all three tools and must be integrated alongside them (Quainoo et al., 2025b). Organizations in emerging markets face different licensing and skills constraints, and studies of value-chain development show how constraints in one link shape feasible choices downstream (Jones et al., 2021). Telehealth research on user trust and regulatory governance indicates that trust in automated evidence depends on transparency about how that evidence is produced (Akintola et al., 2025), and work on device policy in administrative offices shows how endpoint heterogeneity complicates security baselines, a concern for distributed test agents (Onche et al., 2023). Emergency readiness, human error causation, workforce training and safety governance research suggests that capability should be tested under realistic stress rather than assumed, a principle that applies to release-day test pipelines (Arumosoye et al., 2023; Obriki et al., 2020; Obriki et al., 2022a; Obogo et al., 2019b; Obogo et al., 2022).

Finally, the relation between test automation and broader assurance systems deserves study. Inspection standards and non-destructive testing in power generation show how acceptance criteria, sampling and documented evidence combine into assurance (Atta et al., 2021; Atta et al., 2022), and risk pathway and hazard identification models from construction describe how latent conditions propagate into failures (Arumosoye et al., 2022; Obogo et al., 2019). Environmental risk control frameworks link operational discipline to reported performance (Arumosoye et al., 2025), and procedural standardization in high-volume service settings shows that consistency reduces variance in outcomes (Uchechi et al., 2018). Governance of zero trust and cyber assurance adds the availability dimension, since test environments are themselves assets whose unavailability blocks releases (Adebayo et al., 2025; Fadayomi et al., 2019). Data-driven financial decision frameworks provide a template for linking tool investment to business value (Oluwo et al., 2025), and the variance between planned and realized automation benefits is an under-researched outcome in its own right.

## 9 Conclusion

Selenium WebDriver, Cypress and Tricentis Tosca are best understood as three architectural answers to the questions of where test code runs, who writes it and where it is kept. Selenium offers standardized, language-neutral reach across browsers and devices at the price of framework engineering. Cypress offers fast, retry-based feedback for JavaScript teams working inside the browser, bounded by its execution model. Tosca offers model-based, codeless automation across SAP and other packaged applications, bounded by commercial licensing and a proprietary repository. The ETAS model turns this comparison into a documented decision: gates remove infeasible options, pairwise weights make preferences explicit, a rubric anchors ratings to evidence, and sensitivity analysis shows how far a recommendation can move before it reverses. The illustrative archetypes show decisive answers where portfolio and workforce align with one architecture, and a close, reviewable answer where they do not. Six propositions and a staged validation pathway convert the model into a research programme. Enterprises that adopt this approach gain a selection that can be defended, audited and revised as their application landscape changes.

**Funding:** This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

**Competing interests:** The authors declare no competing interests.

**Data availability:** No new data were generated or analysed in this study; all sources are cited in the article.

## References

- Abah, M. A., Okwah, M. N., Oladosu, M. A., Yohanna, N. R., Blessing, C. T., & Vanessa, O. I. (2025). The impact of lifestyle modifications on cardiovascular health: A review of diet, exercise, and stress reduction intervention. *Drug and Pharmaceutical Science Archives*, 5(3), 26-34. <https://doi.org/10.47587/DAP.2025.5301>
- Adebayo, A., Amadi, C., & Ijagbemi, A. O. (2025). From audit to continuous control: Implementation pathways, governance, and organizational readiness for AI-enabled compliance in critical infrastructure. *International Journal of Computer Science and Mathematical Theory*, 11(8), 88-134. <https://doi.org/10.56201/ijcsmt.vol.11.no8.2025.pg88.134>
- Adebayo, A., Anunagba, C. O., & Ozowara, D. E. (2025). A review of zero trust security models and cost effectiveness in healthcare infrastructure. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), 2269-2283. <https://doi.org/10.62225/2583049X.2025.5.6.6051>



- Adegbite, M. P., Adebayo, A., & Ahmed, M. O. (2025). A cyber risk quantification and governance architecture for critical infrastructure: From posture measurement to executive reporting. *International Journal of Computer Science and Mathematical Theory*, 11(12), 232-293. <https://doi.org/10.56201/ijcsmt.vol.11.no12.2025.pg232.293>
- 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>
- Adesuyi, M. O., Kalu, A., & Walawalkar, G. (2023). Data-led cost governance in technology-intensive enterprises. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(3), 877-896. <https://doi.org/10.32628/IJSRCSEIT>
- Afrihiya, E., Akinse, S. G., & Ojukwu, P. U. (2025). Organizational readiness for generative AI integration in healthcare operations: Comparative management capabilities between the U.S. and low- and middle-income countries. *Iconic Research and Engineering Journals*, 8(10), 1673-1697. <https://doi.org/10.64388/IREV8I10-1714670>
- Ahiaek Patrick, M. C., Okonkwo, C. S., Mayo, W., & Okeke, O. T. (2021). Model for data driven vendor evaluation and bid selection using geospatial intelligence. *Shodhshauryam, International Scientific Refereed Research Journal*, 4(4), 426-443. <https://doi.org/10.32628/SHISRRJ214461>
- 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>
- 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>
- Akhigbe, R., Falemi, A., & Akin-Oluyomi, O. T. (2025). An advanced governance model for improving transparency accountability and strategic execution in complex enterprises. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), 1169-1188. <https://doi.org/10.62225/2583049X.2025.5.6.5341>
- Akintola, A. S., Fawehinmi, Y. O., Aiyenitaju, O., Chinemerem, B., & Emmanuella, O. (2025). Digital transformation in telehealth: A systematic review of user trust, privacy, and regulatory governance in AI-powered remote monitoring systems. *Journal of Scientific Research and Reports*, 31(11), 163-182. <https://doi.org/10.9734/jsrr/2025/v31i113658>
- Aliliele, C., Eboh, E. E., & Oluwo, K. (2024). Advances in variance analytics for cost control in manufacturing and industrial enterprises. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3163-3185. <https://doi.org/10.62225/2583049X.2024.4.6.6012>
- 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>
- Aliliele, C., Mbonu, I. S., Uzoka, E., & Iwuanyanwu, U. (2025b). Advances in data lakehouse governance architectures for enterprise data loss prevention and compliance assurance. *Shodhshauryam, International Scientific Refereed Research Journal*, 8(4), 171-213. <https://doi.org/10.32628/SHISRRJ258474>
- Amadi, C. E., Okorafor, U. C., Okorafor, C. I., Okwah, M. N., Akanbi, P. O., Okam, O. V., Udenze, C. I., Mbakwem, A. C., & Ajuluchukwu, J. N. (2025). Association between triglyceride-glucose index and estimated 10-year risk of cardiovascular



disease in Nigerian non-diabetic hypertensives: A cross-sectional study. *PLOS Global Public Health*, 5(7), e0004760. <https://doi.org/10.1371/journal.pgph.0004760>

Amarahobu, C., Ekelemu, O., Akano, O. A., & Adikwu, F. E. (2025a). Structured mentorship models in on-the-job technician development: A review of advancement and promotion coaching schemes, workplace learning theory and the leadership interface in continuous processing environments. *International Journal of Social Sciences and Management Research*, 11(8), 626-656. <https://doi.org/10.56201/ijssmr.vol.11no8.2025.pg626.656>

Amarahobu, C., Ekelemu, O., Akano, O. A., & Adikwu, F. E. (2025b). Transition management and flare reduction across the liquefaction operating cycle: Gas disposition, valve sequencing and procedural complexity in start-up, shutdown and mode change. *International Journal of Engineering and Modern Technology*, 11(8), 516-543. <https://doi.org/10.56201/ijemt.vol.11.no8.2025.pg516.543>

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>

Ambali, K. B., Eyetsemitan, R. A., Oyeleye, A. O., & Fadayomi, O. (2021). Lean Six Sigma for small enterprises: A systematic review and Lite-DMAIC adaptation framework for resource-constrained organizations. *Iconic Research and Engineering Journals*, 5(5), 562-583. <https://doi.org/10.64388/IREV5I5-1716957>

Aminu-Ibrahim, A. Y., Ogbete, J. C., & Iwuanyanwu, O. C. (2025a). Cost control and financial accountability frameworks for national healthcare construction programs. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), 1970-1990. <https://doi.org/10.62225/2583049X.2025.5.6.5700>

Aminu-Ibrahim, A. Y., Ogbete, J. C., & Iwuanyanwu, O. C. (2025b). Infrastructure resilience planning for national diagnostic systems under public health stress conditions. *Gyanshauryam, International Scientific Refereed Research Journal*, 8(1), 340-381. <https://doi.org/10.32628/GISRRJ2582311>

Aminu-Ibrahim, A. Y., Ogbete, J. C., & Iwuanyanwu, O. C. (2025c). Sustainable healthcare infrastructure performance metrics for long-term asset management and value creation. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 11(4), 566-601. <https://doi.org/10.32628/CSEIT251116277>

Ammann, P., & Offutt, J. (2016). *Introduction to Software Testing* (2nd ed.). Cambridge University Press.

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>

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>

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>

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>



- 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>
- 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>
- Arumosoye, O. M., Obriki, O. D., & Ozobu, C. O. (2025). Conceptual framework for environmental risk control and ESG performance through waste handling and operational discipline. *Gyanshauryam, International Scientific Refereed Research Journal*, 8(4), 177-198. <https://doi.org/10.32628/GISRRJ258367>
- 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>
- Atakpa, M. I., Abolaji, T. O., & Abetoh, N. F. (2024). Statistical time-series models for long-range public health trend forecasting: A review and conceptual framework. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 10(6), 2748-2780. <https://doi.org/10.32628/CSEIT2410792>
- Atakpa, M. I., & Abolaji, T. O. (2022). A privacy-preserving data architecture model for regulated industry analytics under GDPR and HIPAA compliance. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 8(3), 781-808. <https://doi.org/10.32628/CSEIT22558>
- 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>
- 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>
- 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>
- Badmus, O., Dosunmu, A. A., & Ozowara, D. E. (2018). A systematic review of CI/CD pipeline strategies in Salesforce DevOps: Tools, practices, and deployment outcomes. *Iconic Research and Engineering Journals*, 2(6).
- Badmus, O., Dosunmu, A. A., Ozowara, D. E., & Anunagba, C. O. (2020). A comparative framework for Salesforce DevOps tooling: Evaluating Copado, Flosun, and Salesforce DX across enterprise deployment contexts. *International Journal of Multidisciplinary Research and Growth Evaluation*, 1(5), 1021-1031. <https://doi.org/10.54660/IJMRGE.2020.1.5.1021-1031>
- 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>
- 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](https://doi.org/10.1007/978-981-16-8763-1_60)

- 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>
- 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>
- Cleland-Huang, J., Gotel, O., & Zisman, A. (Eds.). (2012). *Software and Systems Traceability*. Springer. <https://doi.org/10.1007/978-1-4471-2239-5>
- Dagodzo, D., & Ahiaeke 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/IREV3110-1716084>
- 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>
- 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>
- Dosunmu, A. A., & Ogundele, P. O. (2025c). Security orchestration and automation models for accelerating incident detection and response. *Computer Science and IT Research Journal*, 6(11), 878-894. <https://doi.org/10.51594/csitrj.v6i11.2163>
- 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>
- 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>
- 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.
- Ebhøjie, O., Dogbatsey, E. A., & Oyeleye, A. O. (2023b). Simulation-based tools in graduate accounting education: A conceptual framework for bridging variance analysis theory and practice. *International Journal of Advanced Multidisciplinary Research and Studies*, 3(6), 2851-2862. <https://doi.org/10.62225/2583049X.2023.3.6.6201>
- Eboh, E. E., Aliliele, C., & Odihi, F. (2025). Advances in financial governance and accountability systems in small and medium enterprises. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), 2223-2245. <https://doi.org/10.62225/2583049X.2025.5.6.6013>
- Ekelemu, O., & Akano, O. A. (2025b). Loss detection and integrity surveying in liquefaction header systems: Flare hunting, attribution and programme design for unintended flow to flare. *International Journal of Engineering and Modern Technology*, 11(8), 458-487. <https://doi.org/10.56201/ijemt.vol.11.no8.2025.pg458.487>
- Ekelemu, O. (2025b). Cross-shift information loss and the standardization of logbook handovers: A critical review of communication failure and continuity protocols in continuous processing operations. *International Journal of Engineering and Modern Technology*, 11(8), 397-424. <https://doi.org/10.56201/ijemt.vol.11.no8.2025.pg397.424>



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>

Eyetsemitan, R. A., Ambali, K. B., Oyeleye, A. O., & Fadayomi, O. (2025). An integrated lean-digital framework for scaling small business operations: Synthesizing SOP design, automation, compliance, and change management. *International Journal of Multidisciplinary Research and Growth Evaluation*, 6(6), 1341-1360. <https://doi.org/10.54660/IJMRGE.2025.6.6.1341-1360>

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>

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>

Eyetsemitan, R. A., Oyeleye, A. O., Ambali, K. B., & Fadayomi, O. (2022). Standard operating procedures as strategic assets in small business operations: A systematic review and implementation framework. *Gyanshauryam, International Scientific Refereed Research Journal*, 5(2), 438-465. <https://doi.org/10.32628/GISRRJ225356>

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>

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>

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>

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>

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>

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>

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>



- 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>
- Fobellah, A. N. (2025a). Cognitive biases in financial decision-making: Implications for audit and risk management in large corporations: A conceptual review. *International Journal of Science and Research Archive*, 16(2), 17-22. <https://doi.org/10.30574/ijrsra.2025.16.2.2273>
- 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>
- 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>
- 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>
- Ilodigwe, L., & Adesemoye, A. C. (2025a). Advances, risks, and implementation challenges of artificial intelligence as a force multiplier for clinical decision making and health system efficiency. *International Journal of Medical Evaluation and Physical Report*, 9(7), 165-185. <https://doi.org/10.56201/ijmepr.v9.no7.2025.pg165.185>
- 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>
- ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29119-2 Software and systems engineering: Software testing processes. International Organization for Standardization.
- ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29148 Systems and software engineering: Life cycle processes and requirements engineering. International Organization for Standardization.
- 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>
- 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>
- Kaner, C., Falk, J., & Nguyen, H. Q. (1999). *Testing Computer Software* (2nd ed.). Wiley.
- 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>
- 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>



- Kpoka, M. I. (2023). Comparative analysis of administrative French in Francophone African states and its impact on translation practice. *Journal of Public Administration and Social Welfare Research*, 8(2), 149-163. <https://doi.org/10.56201/jpaswr.v8.no2.2023.pg149.163>
- 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>
- Lilian, I. N., Liadi, K. O., Yeboah, T. J., & Apelehin, A. A. (2020). Understanding cross-cultural communication: Identity, diversity, and global interaction. *Gyanshauryam, International Scientific Refereed Research Journal*, 3(4), 153-176. <https://doi.org/10.32628/GISRRJ21352>
- 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>
- 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>
- 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>
- 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>
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). Wiley.
- 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/wjim.v8.no6.2024.pg217.248>
- 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>
- Obogo, S. F., Obriki, O. D., & Arumosoye, O. M. (2022). Conceptual safety governance model for large commercial facility operations. *Gyanshauryam, International Scientific Refereed Research Journal*, 5(6), 383-410. <https://doi.org/10.32628/GISRRJ225639>
- 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>
- 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>
- 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>



- 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>
- 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>
- Obriki, O. D., Arumosoye, O. M., & Ozobu, C. O. (2025). Conceptual model linking leading safety signals to sustained injury-free project performance. *International Journal of Scientific Research in Humanities and Social Sciences*, 2(3), 233-254. <https://doi.org/10.32628/IJSRHSS252342>
- Obriki, O. D., Obogo, S. F., & Arumosoye, O. M. (2022a). Advances in workforce safety training models for operational risk reduction. *Gyanshauryam, International Scientific Refereed Research Journal*, 5(6), 349-382. <https://doi.org/10.32628/GISRRJ225638>
- 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>
- Okonkwo, C. S., Agbabiaka, J., Mayo, W., & Okeke, O. T. (2024e). Review of digital supply chain models for cost control and operational continuity. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 2836-2846. <https://doi.org/10.62225/2583049X.2024.4.6.5496>
- Oluwadele, J. F., Samuel, O., Tawose, O. M., Ekeocha, A. H., & Adika, A. O. (2025). Evaluation of alternative energy sources to replace maize in Marshall broiler diets: Effects on growth performance, meat quality, and serum biochemistry. *EUREKA: Life Sciences*. <https://doi.org/10.21303/2504-5695.2025> \[Author list truncated in source; verify full authorship.\]
- Oluwo, K., Bola-Sadipe, D., & Aliliele, C. (2025). Conceptual framework for data analytics driven financial decision making in banking sector organizations. *Shodhshauryam, International Scientific Refereed Research Journal*, 8(5), 259-302. <https://doi.org/10.32628/SHISRRJ258529>
- Ominyi, M., Anichukwueze, C. C., & Uzougbo, N. S. (2025a). A conceptual framework for multi-agent AI quality control in the review of regulated documents. *International Journal of Social Sciences and Management Research*, 11(8), 534-557. <https://doi.org/10.56201/ijssmr.vol.11no8.2025.pg534.557>
- 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>
- 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>
- 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>
- 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>



- Quainoo, R., Ogundapo, O., & Asiedu, W. A. (2025b). Test automation in wireless hardware engineering: A comprehensive review of scripting frameworks and instrument control strategies. *International Journal of Engineering and Modern Technology*, 11(10), 405-464. <https://doi.org/10.56201/ijemt.vol.11.no10.2025.pg405.464>
- 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>
- 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>
- 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>
- 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>
- 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>
- 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>
- Sanni, J. O., Iwuanyanwu, U. A., & Essien, M. A. (2025). Problem-oriented process mining for auditable marketing automation lifecycle control. *International Journal of Advanced Multidisciplinary Research and Studies*, 5(6), 1933-1947. <https://doi.org/10.62225/2583049X.2025.5.6.5650>
- 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](https://doi.org/10.34218/IJIERD_14_02_003)
- 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>
- Tonoyan, A., Dada, O., & Ayivi-Donkor, S. S. (2022a). Advances in geospatial analysis: Predictive analytics methods for store location selection and market entry return on investment. *International Journal of Marketing and Communication Studies*, 6(2), 78-105. <https://doi.org/10.56201/ijmcs.v6.no2.2022.pg78.105>
- 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>
- 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>



- 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>
- 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>
- 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>
- 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>
- Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer. <https://doi.org/10.1007/978-3-662-43839-8>
- 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/IREV1I11-1723090>
- 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>