

Test Suite Optimization Using Redundancy Detection Coverage Analysis and Failure Prediction

Toussida Fatah Tanguy Minoungou^{1*}, Aisha I. Abdulwahab², Serif Oyindamola Oyesiji³,

Rosalyn Ezeako⁴

¹*DonorsChoose, New York, New York, USA*

²*Cummins Inc., Indiana, USA*

³*FrieslandCampina WAMCO, Lagos, Nigeria*

⁴*University of Central Missouri, Warrensburg, Missouri, USA*

Abstract

Purpose: Regression suites grow by accretion, and permanent test suite reduction promises lower execution cost, yet deletions are rarely audited for the fault-detection capability they give up. This review asks when coverage-based, behaviour-based and search-based reduction preserves that capability, and what evidence a permanent deletion should carry.

Method: A critical integrative review of software testing research, standards and adjacent measurement literature published up to 2025, appraised for the validity of redundancy criteria, fault-seeding practice and version horizon.

Findings: Coverage equivalence is a weak proxy for behavioural equivalence. Greedy and exact set-cover solutions preserve branch coverage while shedding tests that kill distinct mutants or expose distinct real faults. Similarity-based and multi-objective methods, notably NSGA-II with mutation or fault-history objectives, retain more capability but depend on the quality of the kill matrix and of failure history, which under-represents new and stable code.

Contribution: The article formalises reduction as constrained set cover with explicit loss accounting, proposes the Capability-Preserving Suite Reduction (CPSR) framework with retention guards and a shadow-execution audit, and states six falsifiable propositions with evaluation designs.

Implications: Deletion should be a reversible, audited decision: teams should report fault-detection loss on later versions alongside wall-clock savings, and retain uniquely diagnostic tests even when their coverage is subsumed.

Keywords: test suite reduction; test suite minimization; redundancy detection; coverage analysis; mutation analysis; multi-objective optimization; NSGA-II; failure prediction

Introduction

Regression test suites rarely shrink on their own. Tests are added when features ship, when defects escape and when coverage targets are tightened, but they are seldom removed, because the cost of deleting a test stays invisible until a fault slips through. The result is a suite whose execution time grows faster than the code it protects, and whose individual tests increasingly exercise the same statements, branches and input partitions. Surveys of unit testing practice report that developers write tests incrementally and seldom revisit whether older tests still earn their runtime (Daka & Fraser, 2014), and classic treatments of testing economics have long regarded redundant tests as cost without matching benefit (Myers et al., 2011; Kaner et al., 1999). Test suite reduction, also

called minimization, answers this by permanently removing tests judged redundant with respect to a set of test requirements, usually coverage entities such as statements, branches or definition-use pairs (Ammann & Offutt, 2016; Zhu et al., 1997).

Reduction differs in kind from the two neighbouring techniques with which it is often grouped. Test case prioritization reorders a suite so that faults surface earlier but keeps every test, and regression test selection chooses a subset for one particular change while leaving the stored suite intact (Catal & Mishra, 2013; Li et al., 2007). Reduction is permanent: a deleted test does not run on any future version, including versions whose changes bear no resemblance to the history used to justify the deletion. That permanence changes

the risk calculus. An error in prioritization delays detection, an error in selection affects one build, but an error in reduction removes a detector for as long as nobody notices its absence. Governance research on continuous control makes the comparable point that controls retired because they appeared quiet are the ones whose absence is discovered only after an incident (Adebayo et al., 2025; Akhigbe et al., 2025). Tests also carry institutional memory. A test written after an incident encodes a failure mode that may be documented nowhere else, and, like knowledge passed at shift handover in continuous operations, it is lost silently when its carrier is removed (Ekelemu, 2025b; Amarahobu et al., 2025a).

The Unresolved Problem

The empirical record on reduction contains a durable disagreement. One stream of studies reports that coverage-preserving reduction removes a large fraction of tests with little loss of fault detection; another, using larger programs, seeded faults and differently constructed suites, reports losses that are substantial and hard to predict. The disagreement persists because studies differ in what they hold constant: the adequacy criterion that defines redundancy, the way the original suites were built, whether faults are real or generated by mutation operators, and whether loss is measured on the version used for reduction or on later versions (Zhu et al., 1997; Easterbrook et al., 2008). Three developments have widened rather than closed the gap. Behaviour-based redundancy measures, which compare what tests observe rather than what they execute, promise better retention but lack agreed validation. Multi-objective search, especially the non-dominated sorting genetic algorithm NSGA-II, returns a Pareto front instead of a single answer, which moves the hardest choice from the algorithm to the engineer (Li et al., 2007; Fraser & Arcuri, 2011). Machine-learned failure predictors, trained on build and defect history, are increasingly used to shield tests that seem likely to fail, although the history they learn from is itself a product of earlier testing decisions (Borah et al., 2022; Komi & Adamolekun, 2021).

The gap addressed here is therefore not a shortage of reduction algorithms. It is the absence of an

accepted account of what a permanent deletion should be required to demonstrate, and of how capability lost through reduction should be measured, attributed and bounded. Existing reviews map techniques by algorithm family or fold reduction into prioritization and selection (Catal & Mishra, 2013), which obscures the irreversible character of deletion. Measurement research outside software engineering warns that a quantity optimized as a target tends to detach from the property it was chosen to represent (Sakyi et al., 2022b; Tonoyan et al., 2024b); coverage retained after reduction is exactly such a quantity. A review that treats capability loss as the primary outcome, rather than as a secondary check on size reduction, has not been available for the period ending in 2025.

Research Questions and Contributions

Three research questions structure the review. RQ1 asks under what conditions coverage-based redundancy corresponds to behavioural redundancy, so that removing a coverage-subsumed test leaves fault-detection capability intact. RQ2 asks how similarity-based and multi-objective reduction trade size and execution-time savings against fault-detection loss, and what evidence supports their claimed advantage over greedy set cover. RQ3 asks which biases in the training history distort retention when failure prediction is used to protect tests from removal, and what safeguards counteract them. The questions run from the definition of redundancy, through the optimization that acts on it, to the predictive signals that constrain it (Ammann & Offutt, 2016; ISO/IEC/IEEE, 2018).

The article makes four contributions. First, it formalizes reduction as a constrained set-cover problem in which capability loss appears explicitly, through a loss function defined over mutants, real faults and diagnostic resolution, rather than as a check performed once after the fact. Second, it adjudicates the fault-detection loss controversy by tracing contradictory findings to four identifiable design choices, and states which of those choices a credible evaluation must fix. Third, it proposes the Capability-Preserving Suite Reduction (CPSR) framework, which combines coverage and behavioural redundancy detectors, retention guards for uniquely diagnostic and high-risk tests, multi-objective

selection, and a shadow-execution audit that keeps deletion reversible. Fourth, it states six falsifiable propositions, each paired with an evaluation design, so that the framework can be refuted. The review is conceptual and integrative: no new experiment is reported, and the propositions remain untested (Hevner et al., 2004; Wieringa, 2014).

Theoretical and Formal Foundations

Three bodies of theory make reduction decisions interpretable. Adequacy theory defines what a test suite is supposed to exercise; mutation theory, through the competent programmer and coupling hypotheses, explains why execution alone is an incomplete account of detection; and multi-objective decision theory explains why no single reduced suite is optimal for every stakeholder. Each is used below to derive a specific claim about when deletion is safe. The foundations matter because reduction studies often inherit a definition of redundancy implicitly from the tool that computes coverage, and so never state the assumption that most influences their results. Non-functional requirements sharpen the problem, since properties such as timing, robustness and security are poorly represented by structural coverage entities at all (Zhu et al., 1997; Ammann & Offutt, 2016; Mairiza et al., 2010).

Reduction as Constrained Set Cover

Let $T = \{t_1, \dots, t_n\}$ be a test suite and $R = \{r_1, \dots, r_m\}$ a set of test requirements, for example branches or definition-use pairs. The coverage relation is a binary matrix A in which $a_{ij} = 1$ when test t_j satisfies requirement r_i , and c_j is the cost of running t_j . The classic minimization problem selects a binary vector x that minimizes total cost while every requirement satisfied by T remains satisfied:

$$\min \sum_j c_j x_j \quad \text{subject to} \quad \sum_j a_{ij} x_j \geq 1 \text{ for every } r_i \text{ covered by } T, \quad x_j \in \{0,1\} \quad (1)$$

With unit costs this is the minimum set-cover problem, which is NP-hard, so practical tools use heuristics, or integer linear programming for moderate sizes (Li et al., 2007; Ammann & Offutt, 2016). The additional greedy heuristic repeatedly selects the test that satisfies the most still-unsatisfied requirements and carries a logarithmic approximation

guarantee. Requirement-ordered heuristics instead process first the requirements satisfied by the fewest tests, because the tests that satisfy them are essential, and the GE and GRE variants extract essential tests before removing tests whose requirement sets are subsets of others. Combinatorial optimization in other operational settings, such as warehouse slotting and path planning or prototype system tuning, shows the same pattern: solution quality depends less on the solver than on whether the constraint model represents the loss that matters (Akanbi & Sunday, 2024; Quainoo et al., 2024). Formulation (1) says nothing about faults. The fault-detection capability of a suite is a separate set $F(T)$, the faults or mutants detected by at least one test, and the loss attributable to reducing T to T' is

$$FDL(T') = 1 - \frac{|F(T')|}{|F(T)|} \quad (2)$$

set against size reduction $SR = 1 - |T'| / |T|$ and time reduction $TR = 1 - c(T') / c(T)$, where $c(\cdot)$ denotes total execution cost. Much of the controversy reviewed in Section 4 follows from reporting SR with precision while estimating FDL on a narrow or unrepresentative fault set. Because $F(T)$ is never observed in full, every reported FDL is an estimate conditional on a fault sample, and its credibility depends on how that sample was drawn (Zhu et al., 1997; Myers et al., 2011).

Adequacy, Coupling and the Limits of Coverage

Adequacy criteria were introduced as stopping rules for testing: they say when a suite has exercised enough of the program, not whether any particular test is dispensable (Zhu et al., 1997). Using an adequacy criterion as a redundancy criterion reverses its role. Executing a statement is necessary for a fault in that statement to be revealed but not sufficient. The reachability, infection, propagation and revealability model requires that the faulty statement be reached, that it corrupt program state, that the corruption propagate to an observable output, and that an oracle check that output (Ammann & Offutt, 2016). Two tests that reach the same branch may differ in the input values that infect state, in the paths that carry the infection outward and in the assertions

that reveal it. Mutation analysis operationalizes this gap. The competent programmer hypothesis holds that real faults are small syntactic deviations from a correct program, and the coupling effect holds that tests which detect such simple deviations tend also to detect more complex ones. Under these hypotheses the kill matrix K , with $k_{jm} = 1$ when test t_j kills mutant m , is a closer proxy for detection capability than the coverage matrix A . The implication for reduction is direct: a test that is coverage-subsumed but kills a mutant no other test kills is not redundant with respect to detection, and removing it produces a loss that coverage cannot register (Myers et al., 2011; Fraser & Arcuri, 2011).

Oracle strength is the component of this chain that reduction studies most often ignore. A test that executes a branch but asserts only that no exception was thrown contributes reachability without revealability, whereas a test that checks returned values, state changes and side effects converts the same execution into evidence. Two tests with identical coverage can therefore differ sharply in detection power purely through their assertions, and a reducer that breaks coverage ties arbitrarily will sometimes keep the weaker oracle. Checked coverage, which counts only statements whose effects flow into an assertion, and assertion-aware mutation scores are two ways of bringing oracle strength into the redundancy definition. Survey evidence that developers often write tests with thin assertions, and classic advice that a test without a precise expected result is not a test at all, suggest that the weak-oracle case is common rather than marginal (Daka & Fraser, 2014; Myers et al., 2011; Kaner et al., 1999).

Behavioural Redundancy and Diversity

Behavioural redundancy asks whether a test observes something the remaining suite does not. It can be defined on execution traces, on outputs and assertion values, on input data, or on the kill matrix. A test is kill-redundant with respect to a retained suite T' if the mutants it kills are a subset of those T' kills; it is trace-redundant if its execution profile lies within distance ε of some retained test under a chosen metric, such as Jaccard distance over covered elements, edit distance over call sequences, or normalized compression distance over serialized inputs and outputs. The

diversity view, associated with adaptive random testing and with portfolio arguments, holds that failures cluster in contiguous regions of the input domain, so a suite whose tests are spread out is more likely to intersect a failure region than an equally large suite whose tests are clustered (Myers et al., 2011; Kaner et al., 1999). Research on project teams offers a loose but instructive parallel: heterogeneous members surface problems that homogeneous teams overlook, even when both are nominally qualified for the same task (Amayo et al., 2023c). Substitution is the relevant decision. Replacing an input with a cheaper alternative, whether a feed ingredient or a test, requires evidence of equivalence on the outcomes that matter rather than on the compositional measure that motivated the swap (Oluwadele et al., 2025).

Diversity has a second, less discussed role: diagnostic resolution. When several distinct tests fail together, the pattern of failures helps localize the fault. Reduction that keeps one representative per coverage class can therefore preserve detection while degrading localization, because spectrum-based localization depends on the contrast between passing and failing executions across many tests. A suite with one surviving test per class detects the same regression but offers the developer a single failure signal instead of a pattern. Studies of continuous performance monitoring make a parallel observation: consolidated indicators still register that something went wrong but lose the attribution needed to act on it quickly (Tonoyan et al., 2024b; Ogbole et al., 2021; Atakpa & Fobellah, 2023).

Multi-Objective Decision Theory

Reduction has at least three objectives in tension, cost, retained coverage and retained detection capability, to which failure likelihood and diagnostic diversity may be added. When objectives conflict, a candidate suite dominates another if it is no worse on every objective and strictly better on at least one, and the Pareto front is the set of non-dominated candidates. NSGA-II approximates this front through fast non-dominated sorting and a crowding-distance operator that preserves spread along the front, and its output is conventionally assessed with the hypervolume indicator, the volume of objective space

dominated by the front relative to a reference point (Li et al., 2007). Decision theory adds that choosing a point on the front expresses a preference, not a fact about the suite. Selecting the knee, where further savings begin to purchase large capability losses, is a defensible default, but it embeds an exchange rate between minutes of execution and escaped faults that should be stated and owned. Research on KPI-driven decision frameworks and performance governance finds that implicit exchange rates between cost and risk are the ones least often reviewed, and risk-based approval models in regulated industries make the same trade between speed, safety and evidence quality explicit rather than leaving it to defaults (Edivri & Oteri, 2024; Falemi et al., 2024; Eze et al., 2024b).

Review Protocol

The review follows an integrative design, chosen because the relevant evidence is heterogeneous: controlled experiments on benchmark programs, industrial case studies, algorithm papers with benchmark comparisons, standards, and methodological guidance. Pooling effect sizes across these designs would mislead, because FDL estimates depend on fault type and suite construction in ways that are not reported consistently. The protocol therefore borrows reporting discipline from systematic review guidance, including a transparent search, eligibility rules and appraisal, while synthesizing through structured comparison and adjudication rather than meta-analysis. Table 1 summarizes the protocol and Figure 1 shows its stages (Page et al., 2021; Dybå & Dingsøyr, 2008; Easterbrook et al., 2008).

Sources, Search Strings and Time Window

Searches were designed for IEEE Xplore, the ACM Digital Library, Scopus, Web of Science and SpringerLink, with backward and forward snowballing from anchor works on adequacy, search-based testing and traceability. The window runs from 1990, shortly before the first formal treatments of suite minimization, to the end of 2025. The core Boolean string combined three concept blocks: (“test suite reduction” OR “test suite minimization” OR “test suite minimisation” OR “redundant test*” OR

“test redundancy”) AND (coverage OR mutation OR “fault detection” OR similarity OR diversity OR “multi-objective” OR NSGA*) AND (regression OR “unit test*” OR “system test*”). A second string targeted failure prediction used as a retention signal: (“test failure prediction” OR “flaky test*” OR “defect prediction”) AND (test* NEAR/3 (reduction OR retention OR removal)). Standards on test processes and requirements engineering, and methodological sources on empirical software engineering and design science, were added by targeted retrieval rather than keyword search (ISO/IEC/IEEE, 2018; Runeson & Höst, 2009; Wieringa, 2014). Because questions of measurement and accountability recur in adjacent fields, sources on control monitoring, key performance indicators and predictive analytics were admitted where they informed a specific measurement or governance claim, and they are cited only for that claim (Oluwo et al., 2025; Ilodigwe & Adesemoye, 2024a).

Eligibility, Screening and Appraisal

Inclusion required that a source define redundancy or reduction explicitly, report or specify a measure of fault-detection capability, coverage or execution cost, or supply a standard, method or measurement principle used directly in the framework. Sources were excluded when reduction appeared only as a by-product of prioritization without any deletion, when fault detection was asserted but not measured, or when suite construction was undocumented, since suite construction strongly conditions FDL. Screening proceeded in three stages: title and abstract against the concept blocks, full text against the inclusion rules, and design-level extraction. The extraction form recorded the subject programs and their size, the origin of the original suite (hand-written, generated or augmented), the adequacy criterion defining redundancy, the reduction algorithm, the fault source (real, hand-seeded or mutation operators), whether loss was measured on the reduction version or on later versions, and the treatment of flaky tests (Runeson & Höst, 2009; Daka & Fraser, 2014).

Appraisal used five questions adapted from case-study and experimental guidance: whether the fault set was independent of the redundancy criterion, whether loss was estimated out of

version, whether a random subset of equal size or cost served as a baseline, whether variance across repeated runs was reported, and whether the cost model reflected execution time rather than test count (Easterbrook et al., 2008). Quality-assurance practice in manufacturing, inspection and laboratory diagnostics informed the questions on baselines and independence, because sampling and inspection plans in those fields must show that detection power survives a reduction in inspection effort before the reduction is accepted (Asiedu & Asiedu, 2024b; Atta et al., 2022; Yusuf & Abdulrazaq, 2018). Studies that failed the independence question were flagged rather than silently excluded, since their results remain informative about the optimistic end of the loss distribution.

Synthesis Logic

Synthesis proceeded in two passes. The first grouped sources by technique family and extracted, for each family, the redundancy definition, the typical magnitude of reported SR and TR, and the conditions under which FDL was reported as low or high. The second pass was adjudicative: where findings conflicted, the design attributes recorded at extraction were compared to test whether the conflict could be explained by a difference in fault source, suite origin, version horizon or baseline. A conflict was treated as resolved only when one design attribute accounted for the divergence across more than one

pair of studies; otherwise it was reported as open. Failure-prediction sources were appraised separately for label provenance, class-imbalance handling and temporal validation, because predictors evaluated with random rather than time-ordered splits overstate performance on future builds (Wedraogo & Sanni, 2024; Komi, 2023). No flow-diagram counts are reported, because the aim is conceptual integration rather than an exhaustive census; the protocol is disclosed so that a replication could reproduce the search and challenge the inclusion decisions (Page et al., 2021; Catal & Mishra, 2013). Table 1 records each protocol decision with the safeguard it carries, and Figure 1 traces the path from retrieval to the propositions, including the loop by which unresolved conflicts were returned to extraction.

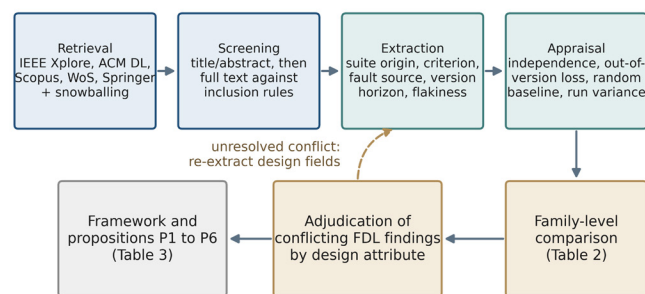


Figure 1: Review protocol stages from retrieval to proposition derivation, with the adjudication loop used for conflicting fault-detection loss findings

Table 1: Review protocol for the synthesis of test suite reduction evidence

Protocol element	Decision rule in this review	Safeguard against bias
Databases	IEEE Xplore, ACM Digital Library, Scopus, Web of Science, SpringerLink; snowballing from adequacy, search-based testing and traceability anchors	Snowballing catches deletion studies indexed under prioritization or selection
Time window	1990 to 2025	Early benchmark studies kept to trace the origin of the loss controversy
Concept blocks	Reduction, minimization or redundancy AND coverage, mutation, similarity or multi-objective AND regression, unit or system testing	Separate string for failure prediction used as a retention signal
Inclusion	Explicit redundancy definition; measured capability, coverage or cost; or a standard or method used directly	Reduction reported only as a by-product of prioritization excluded
Extraction fields	Subject programs, suite origin, adequacy criterion, algorithm, fault source, version horizon, flakiness treatment	Fields chosen to explain, not merely describe, divergent loss estimates
Appraisal	Fault set independent of criterion; out-of-version loss; equal-cost random baseline; run variance; time-based cost	Studies failing independence flagged rather than silently excluded
Synthesis	Family-level comparison, then adjudication of conflicts by design attribute	Conflict left open unless one attribute explains divergence in more than one study pair

Synthesis of Evidence

The synthesis is organized by the redundancy definition each technique family adopts, because the definition, more than the optimization algorithm, determines what capability a reduction can lose. Table 2 compares the families on their redundancy definition, typical method, principal strength, characteristic failure mode and the evidence a reviewer should require. The comparison supports one overarching finding: reduction methods differ less in how efficiently they shrink a suite than in which kind of loss they make visible. Coverage-based methods are efficient and transparent but blind to propagation and oracle strength; behaviour-based methods see more but depend on the representativeness of traces and mutants; predictive methods protect against recurrence but reproduce the blind spots of the history they learn from (Borah et al., 2022; Ammann & Offutt, 2016).

Coverage-Based Minimization and Its Variants

Coverage-based minimization remains the default in tools and in the literature because the coverage matrix is cheap to obtain and the optimization is well understood. Exact integer programming finds a minimum-cardinality suite for moderately sized problems, while greedy, requirement-ordered and GE/GRE heuristics scale to industrial suites and usually land within a few tests of optimal

on benchmark programs (Li et al., 2007). The comparative literature converges on a finding that is practically important and often overlooked: on the same coverage matrix, different heuristics return suites of similar size but different composition, and composition, not size, determines which faults remain detectable. Ties are common, because many tests satisfy identical requirement sets, so tie-breaking rules that look incidental decide which of several coverage-equivalent tests survives. A defensible implementation records its tie-breaking rule and, where possible, breaks ties by secondary capability evidence, such as unique mutant kills or assertion density, rather than by test order or name (Ammann & Offutt, 2016; Myers et al., 2011). Where deletion is carried out by hand from a tool's recommendation, the same ambiguity invites the slips and lapses that human-error models associate with routine, low-salience decisions (Obriki & Arumosoye, 2020).

The choice of criterion matters as much as the algorithm. Statement coverage yields the most aggressive reduction and the largest risk of loss; branch and definition-use criteria retain more tests; and requirement-level coverage, obtained through traceability links from specifications to tests, protects behaviours that code coverage cannot see, such as acceptance scenarios and non-functional constraints (Gotel & Finkelstein, 1994; Ramesh & Jarke, 2001; ISO/IEC/IEEE, 2018). Traceability

Table 2: Test suite reduction technique families compared by redundancy definition, method and characteristic loss

<i>Family</i>	<i>Redundancy definition</i>	<i>Typical method</i>	<i>Principal strength</i>	<i>Characteristic failure mode</i>	<i>Evidence to require</i>
Coverage-based	Requirements subsumed by retained tests (statement, branch, def-use)	Greedy, requirement-ordered, GE/GRE, ILP	Cheap, transparent, deterministic	Drops tests differing in infection, propagation or oracle	Out-of-version FDL on independent faults
Traceability-based	Traced requirements covered by retained tests	Set cover over trace links	Protects acceptance and non-functional behaviour	Incomplete or decayed links	Audit of link precision and recall
Mutation kill-matrix	Killed mutants subsumed by retained tests	Set cover over K	Closest proxy for detection	Equivalent and subsumed mutants inflate uniqueness; analysis cost	Subsumption-reduced mutant set
Similarity or diversity	Trace, input or output within distance ε of a retained test	Clustering, medoid choice, diversity maximization	Preserves diagnostic contrast	Threshold tuning; flakiness mimics diversity	Sensitivity to ε ; flaky tests quarantined
Multi-objective search	Dominated on cost, coverage, capability and diversity	NSGA-II and many-objective variants	Exposes the price of each capability unit	Correlated objectives; unjustified choice of point	Repeated runs, $\hat{\Delta}12$ effect sizes, documented preference
Prediction-guarded	Low predicted failure probability	Classifier score as objective or guard	Protects recurrently failing areas	Survivorship, severity blindness, flaky labels	Time-ordered validation; calibration by stratum

research shows that such links are incomplete and decay without maintenance, and that links recovered by information-retrieval methods arrive as ranked candidates whose precision must be checked before they can anchor a deletion decision (De Lucia et al., 2007; Borg et al., 2014; Cleland-Huang et al., 2012). User acceptance tests are a case in point: they are often coverage-redundant with respect to unit tests yet are the only artefacts that encode the business scenario a release must support (Eyetsemitan et al., 2023b).

Adjudicating the Fault-Detection Loss Controversy

The central empirical dispute in the reduction literature concerns how much capability coverage-preserving reduction sacrifices. Early experiments on small programs whose suites were built to be coverage-adequate reported minimal loss alongside large size reductions. Later experiments on the widely used Siemens programs, with larger suites assembled from random test pools and hand-seeded faults, reported losses that were substantial and varied widely across programs and suites. Read as a contradiction, these results leave practitioners without guidance. Read through the extraction fields of this review, the divergence is largely explained by four design choices (Zhu et al., 1997; Easterbrook et al., 2008).

The first choice is suite origin. When the original suite is itself minimal or near-minimal for the criterion, there is little redundancy to remove and little capability to lose; when it is large and randomly assembled, many tests are coverage-redundant yet behaviourally distinct, and removing them costs detection. The second is fault source. Faults that are independent of the redundancy criterion, such as real regressions or mutants produced by operators unrelated to the coverage entities, expose loss that faults chosen to lie on covered statements cannot. The third is version horizon. FDL measured on the version used for reduction understates loss on later versions, because the retained suite was fitted to the current code and the coverage relation shifts beneath it as the code changes. Engineering assets are judged over their service life rather than at commissioning for the same reason, since performance at handover says little about performance under later load and

modification (Ogbete et al., 2023). The fourth is the baseline. A reduction that loses a tenth of detection sounds poor until it is compared with a random subset of equal cost, which may lose far more, and sounds good until it is compared with a variant that also keeps every unique mutant killer at modest extra cost (Myers et al., 2011; Runeson & Höst, 2009). Coverage retained after reduction behaves like a surrogate endpoint, and health research has long cautioned that interventions judged on surrogate markers must be confirmed against the outcomes those markers are meant to predict (Amadi et al., 2025; Abah et al., 2025).

The adjudication therefore declares neither stream correct. Coverage-preserving reduction is close to loss-free when the suite is already lean, faults are coupled to the criterion and evaluation is in-version. It is materially lossy when the suite is large and heterogeneous, faults are independent and evaluation is out-of-version, a description that fits most industrial regression suites. It follows that no published FDL estimate transfers to a new suite without re-estimation, and that evaluations built on generated tests need particular care. Suites produced by whole-suite search in EvoSuite are minimized against their own coverage goals during generation, so any further reduction operates on suites that are already lean, and their assertions are regression oracles that capture observed rather than specified behaviour (Fraser & Arcuri, 2011; Daka & Fraser, 2014). Inspection regimes in high-consequence engineering recognize the same pattern: a sampling plan validated on one population of defects does not certify detection power for another, and monitoring designed around known fault signatures misses novel ones (Atta et al., 2021; Shittu et al., 2023; Obriki & Arumosoye, 2019).

Behavioural and Similarity-Based Redundancy

Behaviour-based methods respond to the coverage blind spot by redefining redundancy. Kill-matrix reduction solves formulation (1) with K in place of A, guaranteeing that the reduced suite kills every mutant the original killed; its cost is the mutation analysis itself, which scales with the product of mutants and tests, although selective operators and mutant sampling reduce it substantially (Ammann &

Offutt, 2016). Trace-similarity methods cluster tests by the distance between their execution profiles and keep one or more representatives per cluster, often the medoid or the member with the most assertions; input and output similarity methods apply the same idea to serialized data. The evidence favours behaviour-based methods for retained detection at equal size, subject to three recurrent qualifications (Borah et al., 2022; Fraser & Arcuri, 2011).

First, mutant sets contain equivalent and subsumed mutants that inflate apparent uniqueness, so a test that uniquely kills a subsumed mutant may be protecting nothing a real fault would reveal. Second, similarity thresholds are tuned per system and the literature offers no transferable default for ϵ , so part of any reported gain reflects tuning on the evaluation subject. Third, trace similarity is sensitive to nondeterminism: flaky tests produce variable traces, and a test whose trace varies may look diverse when it is merely unstable. A defensible behavioural criterion therefore uses a subsumption-reduced mutant set, reports how the retained suite changes as ϵ varies, and quarantines flaky tests before any distance is computed. The measurement lesson generalizes: indicators built on noisy or duplicated signals overstate uniqueness unless deduplication and stability checks precede aggregation, a point made in data-governance work on loss prevention, auditable process mining and regulated analytics architectures (Aliliele et al., 2025b; Sanni et al., 2025; Atakpa & Abolaji, 2022). Screening assays in food safety offer a further parallel, since the sensitivity of a detection method must be established on samples independent of those used to calibrate it (Tawose et al., 2023).

Multi-Objective Search and the Choice of a Point

Multi-objective formulations treat reduction as a search for trade-offs. A typical configuration encodes each candidate suite as a bit string over T and evolves a population with NSGA-II under objectives such as minimizing execution time, maximizing coverage, maximizing mutation score or past fault-revealing capability and, in some variants, maximizing diversity (Li et al., 2007). The advantages over single-objective reduction are real: the front exposes the marginal price of each unit of capability, and it frequently shows that most of

the available time savings come at negligible loss while the last few percentage points of savings are expensive. Search-based results are stochastic, however, and credible studies report medians over repeated runs with non-parametric tests and effect sizes such as the Vargha-Delaney $\hat{A}_{12}$ statistic rather than single best runs. Three limitations recur. When fault history is an objective, the search learns to keep tests that failed before, which privileges old, frequently changed code and penalizes tests of new or stable components. When coverage and mutation score are both objectives they are highly correlated, and the front collapses toward a curve that adds little beyond single-objective reduction. When there are more than three objectives, Pareto dominance loses discriminating power, and many-objective algorithms or objective aggregation become necessary (Catal & Mishra, 2013; Li et al., 2007).

The final step, choosing one suite from the front, is under-reported. Most studies either report the whole front, which no pipeline can execute, or pick the knee without justification. Research on multi-criteria choice in procurement and process optimization suggests that an explicit, documented preference, reviewed when the context changes, is more robust than an implicit default (Ahiaeke Patrick et al., 2021; Tonoyan et al., 2022b). Behavioural research adds a caution about anchoring: a headline figure such as a 60 per cent size reduction tends to dominate discussion even when the time saving and the capability loss, which are what matter, are reported alongside it (Fobellah, 2025a). A reduction report should therefore lead with wall-clock TR and estimated FDL, and treat SR as secondary.

Failure Prediction as a Retention Constraint

Failure prediction enters reduction in two ways: as an objective that rewards retaining tests with high predicted failure probability, and as a guard that exempts such tests from deletion. Predictors use features such as recent failure frequency, time since last failure, the textual or dependency coupling between a test and changed files, code churn and test age, and are typically gradient-boosted trees or logistic models. Their appeal is that past failures are the most direct evidence of detection capability, and interpretable models allow engineers to see why a test

is protected (Komi & Adamolekun, 2021; Borah et al., 2022). Reviews of artificial intelligence techniques for secure software testing treat automated regression control as a core application of such models, which makes the retention logic itself a security-relevant artefact that deserves verification in its own right (Mbonu et al., 2021b). Predictive analytics for siting and valuation decisions in other sectors shows the same appeal and the same dependence on whether the training population resembles the decision population (Tonoyan et al., 2022a; Tonoyan et al., 2023).

The review identifies three biases that make predicted failure probability a hazardous sole criterion for permanent retention. The first is survivorship: tests that never ran against a component, because they were deselected or added recently, have no failure history and receive low scores whatever their capability. The second is severity blindness: predictors estimate frequency, not consequence, so a test guarding a rarely changed but safety- or security-relevant path, whose failures are rare and severe, is scored as low value. The third is label contamination by flakiness: intermittently failing tests accumulate failure labels that do not correspond to faults, and a predictor rewards them with retention. Forecasting research outside software shows that models accurate on average can be systematically wrong for subgroups under-represented in training data, and that aggregate accuracy conceals the pattern (Komi, 2023; Wedraogo & Sanni, 2024; Ilodigwe & Adesemoye, 2025a). For reduction, the neglected subgroups are new components and stable critical components. Predicted failure probability should therefore protect tests from deletion but never be the sole reason to delete, and predictors should be validated with time-ordered splits, calibrated, and audited separately on low-history strata (Atakpa & Fobellah, 2023; Nwakamma et al., 2024a).

Cost Models: What Reduction Actually Saves

Size reduction is the most frequently reported outcome and the least informative. Execution time in modern pipelines is dominated by a minority of tests, usually integration and end-to-end tests with heavy fixtures, so removing many cheap unit tests can achieve a large SR with negligible TR. Shared fixtures complicate the arithmetic: removing some

tests in a class may not remove the class-level setup cost, and parallel sharding means that wall-clock savings depend on the critical path through the shards rather than on the sum of test times. Maintenance is a second saving that studies seldom measure but practitioners value, since fewer tests mean fewer brittle assertions to update when behaviour changes intentionally (Daka & Fraser, 2014; Kaner et al., 1999). The cost vector in formulation (1) should therefore combine expected execution time, a fixture-sharing adjustment and an estimate of maintenance burden, and outcome reports should state TR in wall-clock terms on the production pipeline. The trade between throughput and diagnostic accuracy is familiar from laboratory planning, where layout decisions that raise throughput are judged also by their effect on accuracy (Ogbete et al., 2018).

Cost-governance studies in technology-intensive organizations draw the same distinction between visible unit counts and the real drivers of spend, and warn that savings claims built on the wrong denominator erode trust in the programme that produced them (Adesuyi et al., 2023; Aminu-Ibrahim et al., 2025a; Okonkwo et al., 2019). Benefits that are diffuse and delayed, such as lower maintenance effort or faster developer feedback, are hard to quantify, and evaluation research on investments with indirect returns recommends estimating them explicitly rather than omitting them (Uchechi et al., 2022). Digital supply-chain models that track cost alongside operational continuity offer a useful template: a reduction that saves minutes but interrupts continuity of detection has not reduced cost in any meaningful sense (Okonkwo et al., 2024e).

The Capability-Preserving Suite Reduction Framework

The synthesis motivates a framework whose organizing principle is that deletion should be justified by evidence of capability redundancy, bounded by explicit guards, and kept reversible until an audit confirms that nothing was lost. The Capability-Preserving Suite Reduction (CPSR) framework has five components: a redundancy evidence layer, a retention guard layer, a multi-objective selection layer, a staged deletion protocol with shadow execution, and a loss audit that feeds back into the

guards. Figure 2 shows the components and their data flows. The framework draws on design science in treating the reduction pipeline as an artefact whose utility must be evaluated against explicit criteria, and on organizational learning models in requiring that audit findings change upstream rules; frameworks that link the parts of a chain to downstream stability make the same demand that local decisions be judged by system outcomes (Hevner et al., 2004; Arumosoye & Obriki, 2021; Ilodigwe & Adesemoye, 2024b; Jones & Ominyi, 2021).

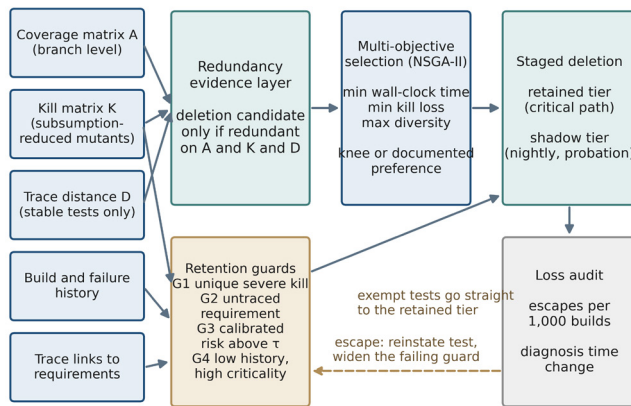


Figure 2: Architecture of the Capability-Preserving Suite Reduction (CPSR) framework, showing redundancy evidence, retention guards, selection, staged deletion and the loss-audit feedback path

Redundancy Evidence and Retention Guards

The evidence layer computes three matrices: the coverage matrix A at branch level, a kill matrix K over a subsumption-reduced mutant sample, and a behavioural distance matrix D over execution traces and serialized outputs, computed only for tests that pass a stability check of repeated runs. A test becomes a deletion candidate only if it is redundant on all three. The conjunction is deliberately conservative, since coverage redundancy alone is necessary but insufficient. The guard layer then exempts four classes of tests irrespective of redundancy evidence. G1 protects tests that are the only detector of any mutant in a severity-weighted operator class, for example boundary-condition and exception-handling mutants on paths tagged as safety- or security-relevant. G2 protects tests traced to requirements, including non-functional requirements, that no retained test covers, using links whose precision has been audited (Cleland-Huang et al., 2012; ISO/

IEC/IEEE, 2018). G3 protects tests whose calibrated failure probability over the next release window exceeds a threshold τ , set per history stratum so that low-history components are not penalized. G4 protects tests with little history but high criticality, reversing the survivorship bias identified in Section 4.5. Guards work like hold points in quality systems for high-consequence work, where an inspection cannot be waived on efficiency grounds alone (Atta et al., 2022; Obogo et al., 2019b; Arumosoye & Obriki, 2022).

Security and privacy enforcement is the clearest case for G1 and G4. The security literature describes access-control, zero-trust and network-hardening controls whose failure is rare in normal operation but decisive when it occurs (Adegbite et al., 2020; Ahmed et al., 2021; Ladapo et al., 2023b; Onche et al., 2023; Jimoh & Ahmed, 2024), and privacy-by-design and data-protection assessment frameworks require such protections to remain demonstrably in force across system change (Ominyi, 2021; Mbonu et al., 2022a; Afrihyia et al., 2024; Badmus et al., 2024). Reviews of legacy control networks and healthcare infrastructure document detection gaps that persisted because the relevant checks were assumed rather than verified (Amadi et al., 2025c; Adebayo et al., 2025). Regression tests that verify these controls rarely fail, rarely change and look coverage-redundant, which is precisely the profile that frequency-based predictors and greedy reducers select for removal; the CPSR guards exist to stop that from happening (Dosunmu & Ogundele, 2020).

Selection, Staged Deletion and the Shadow Audit

The selection layer runs NSGA-II over the deletion candidates only, with objectives of minimizing wall-clock execution time, minimizing kill-matrix loss and maximizing mean behavioural distance among retained tests. The chosen suite is the knee unless a documented preference specifies another exchange rate. Deletion is then staged. Tests selected for removal are not deleted but demoted to a shadow tier that runs off the critical path, nightly or on a sampled fraction of builds, for a probation period long enough to span several release cycles. The shadow audit records every build on which a demoted test fails while all retained tests pass. Such an event, an escape,

is a direct observation of capability loss; it triggers reinstatement of the test and a review of the guard that failed to protect it. Only tests that complete probation without an escape are archived, and archived tests remain retrievable. Staged demotion follows the logic of reversible change management, in which a control is withdrawn gradually while its absence is monitored, and of loss-detection programmes in process plants, which rely on continuous surveying rather than one-off certification (Aliliele et al., 2023a; Ekelemu & Akano, 2025b; Obriki et al., 2025).

The audit's loss estimate is the escape rate per thousand builds, reported with a confidence interval, together with the change in mean diagnosis time for failures on which the retained suite fired, since diagnostic degradation is the loss most likely to be felt before it is measured. When an escape occurs, attribution matters: the review asks whether the escaping test was removed because of a coverage tie, a mis-set ϵ , an uncalibrated predictor or a missing trace link, and the answer determines which guard is widened. Attribution research warns that credit and blame assigned to the last visible touchpoint misrepresent multi-step causation, and the same risk applies to escapes (Sanni et al., 2022). Reinstatement must be quick for the audit to be credible, which calls for the kind of prepared response capacity that readiness models in operational settings describe (Arumosoye & Obriki, 2023; Ogbete et al., 2022). Escape rates and diagnosis-time trends belong in the same executive reporting that carries other quantified operational risks (Adegbite et al., 2025; Sakyi et al., 2022b).

An Illustrative Scenario

Consider a hypothetical service with 4,000 regression tests taking 95 minutes per full run, of which 600 integration tests account for 70 minutes. Branch-level greedy reduction removes 1,900 tests, an SR of 47.5 per cent, but because most of the removed tests are fast unit tests the TR is only 14 per cent. A kill matrix over a 5,000-mutant sample shows that 140 of the removed tests are the unique killers of at least one mutant; restoring them costs 3 minutes per run. Of the remaining candidates, guard G3 exempts 35 tests with high calibrated failure probability and guard G4 exempts 22 low-history tests on a payments path.

NSGA-II over what remains, which now includes integration tests with overlapping traces, finds a knee at which TR reaches 31 per cent with a kill-matrix loss of 0.4 per cent of mutants. During probation the shadow tier records two escapes in 1,200 builds, both from demoted integration tests that exercise a retry path under timeout. Both tests are reinstated, and G1 is extended to cover timeout-handling mutants.

The numbers are invented for exposition, but the pattern they illustrate follows from the synthesis: SR overstates savings, uniquely killing tests are cheap to keep, the largest savings come from overlapping integration tests rather than from unit tests, and escapes surface in rarely exercised error paths. Figure 3 depicts the corresponding trade-off between execution time and retained mutant-kill capability, and locates the greedy solution relative to the approximate Pareto front and to random subsets of equal cost. Accountability research stresses that the value of a worked case lies in making the decision chain inspectable, so that a reviewer can see which evidence justified each removal and which guard intervened (Akhigbe et al., 2024; Edivri & Oteri, 2024).

The scenario also shows why the probation period must be sized statistically rather than by habit. Two escapes in 1,200 builds is an observed rate of about 1.7 per thousand builds, but an exact Poisson interval for two events runs from roughly 0.2 to 6.0 per thousand, so the audit cannot yet distinguish a negligible loss from a material one. Conversely, a probation period with no escapes bounds the rate only loosely: by the rule of three, zero escapes in n builds supports an approximate 95 per cent upper bound of $3/n$, so demonstrating a rate below one per thousand requires about 3,000 clean builds. Teams that run a demoted tier on a sampled fraction of builds must scale the probation accordingly. Forecasting research makes the general point that uncertainty around a small-count estimate should be reported and acted on, not collapsed into a point value (Wedraogo & Sanni, 2024; Easterbrook et al., 2008).

Propositions

Six propositions follow from the synthesis and the framework. Each is stated so that an observable

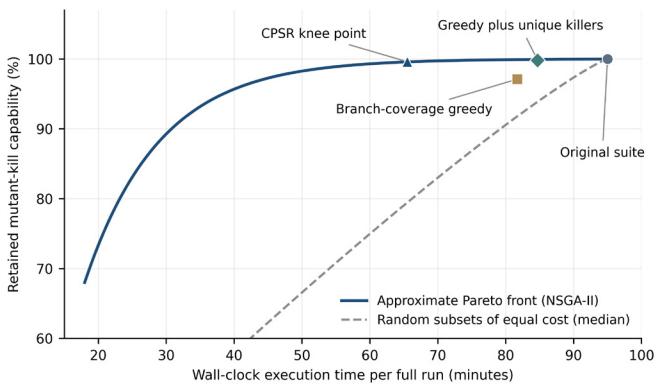


Figure 3: Illustrative trade-off between execution time and retained mutant-kill capability for the hypothetical suite in Section 5.3 (synthetic values)

result would contradict it, and Table 3 pairs each with an evaluation design, a primary measure and a refuting observation. The falsifiability discipline follows empirical software engineering guidance, which asks that each claim specify the comparison, the measurement and the result that would count against it, and process-optimization research in regulated services, which ties each performance claim to a metric that oversight bodies can check (Easterbrook et al., 2008; Runeson & Höst, 2009; Anioke & Atima, 2023a).

P1 (coverage insufficiency). For regression suites not generated to be minimal, removing tests that are coverage-redundant but not kill-redundant yields a higher out-of-version escape rate than removing an equal number of tests that are redundant on both criteria.

P2 (version horizon). FDL estimated on the reduction version underestimates FDL measured over

the following five releases, and the gap widens with code churn in the covered components.

P3 (behavioural retention). At equal wall-clock TR, kill-matrix and trace-similarity reduction retain more real-fault detection than branch-coverage greedy reduction, and the advantage is larger for suites with heavy fixtures and diverse inputs.

P4 (predictor bias). Failure predictors trained on build history are less well calibrated on low-history and low-churn components than on high-churn components, and reduction guided only by predicted failure probability concentrates escapes in the former.

P5 (diagnostic degradation). Reduction that preserves detection lengthens the time from first failure to identification of the culprit change, in proportion to the number of same-class sibling tests removed.

P6 (reversibility). Staged deletion with a shadow audit yields a lower escape rate over the following year than immediate deletion at the same TR, and the difference is concentrated in error-handling and timeout paths.

Discussion

The discussion draws out what the synthesis implies for theory and practice, then takes seriously the strongest arguments against the position that reduction should be conservative, audited and reversible. The central claim is that much of the literature has mis-specified the reduction problem as size minimization with a capability constraint checked afterwards, when it is more accurately capability accounting with a cost objective. The

Table 3: CPSR propositions with evaluation designs, measures and refuting observations

Proposition	Evaluation design	Primary measure	Refuting observation
P1 Coverage insufficiency	Replay on project histories with matched removal sets	Escape rate on real regressions in later versions	No difference, or fewer escapes for coverage-only removal
P2 Version horizon	Reduce at version k; evaluate at k and at k+1 to k+5	FDL on disjoint mutants and real faults per version	In-version FDL equal to or above out-of-version FDL
P3 Behavioural retention	Three methods at equal wall-clock TR, 30 or more runs each	Real-fault detection; $\hat{A}12$ effect size	Greedy equal or superior at equal savings
P4 Predictor bias	Time-ordered training; calibration stratified by history	Expected calibration error by stratum; escape location	Equal calibration across strata; escapes not concentrated
P5 Diagnostic degradation	Paired analysis of failures before and after reduction	Time from first failure to culprit change	No increase, or increase unrelated to siblings removed
P6 Reversibility	Staged versus immediate deletion across comparable services	Escapes per 1,000 builds over 12 months	Staged deletion no better than immediate

reframing matters because it changes which evidence a reduction study is obliged to report, and because it changes who in an engineering organization should own the decision to delete. Traceability research has made a similar argument about requirements links, which become useful only when someone is accountable for their currency (Hevner et al., 2004; Cleland-Huang et al., 2012).

Theoretical Implications

Three theoretical implications follow. First, adequacy criteria should be recognized as criteria of sufficiency, not of redundancy. A suite that is adequate for branch coverage is one from which enough has been exercised; it does not follow that every test beyond a minimal adequate subset is surplus, because detection depends on infection, propagation and oracle strength that coverage does not measure (Zhu et al., 1997; Ammann & Offutt, 2016). Second, the coupling effect, usually invoked to justify mutation analysis as a test-generation criterion, has a corollary for reduction that deserves explicit statement: if simple-fault detection couples to complex-fault detection, then losing a unique simple-fault detector risks losing complex-fault detection as well, so kill-matrix uniqueness is a principled deletion veto. Third, diversity serves detection and diagnosis, and the two roles are separable; theories of test effectiveness that ask only whether a fault is detected omit the value a suite has in localizing it. This connects reduction to evidence theory in organizational monitoring, where redundant signals are valued for triangulation as well as for sensitivity, and where risk models distinguish the likelihood of an event from the ability to explain it afterwards (Mbonu et al., 2018; Aliliele et al., 2024a; Ogbale et al., 2021).

A further implication concerns search-based software engineering. In search-based reduction the fitness function is, in effect, a theory of what makes a test valuable, and the results of the search are only as sound as that theory. Fitness functions built from coverage and execution time encode the assumption that capability is structural; adding mutation score encodes the coupling hypothesis; adding fault history encodes the assumption that the future resembles the past. Treating the fitness function as an explicit, testable claim rather than as

an engineering convenience makes the propositions in Table 3 tests of competing theories of test value, not merely comparisons of algorithms. It also explains why algorithm comparisons in this area so often produce small and unstable differences: when two searches optimize the same flawed proxy, the choice of search operator matters far less than the proxy itself (Li et al., 2007; Fraser & Arcuri, 2011; Wieringa, 2014).

Practical Implications

For engineering teams, the synthesis implies a small set of concrete changes. Report TR in wall-clock terms on the real pipeline, never SR alone. Compute the kill matrix, at least on a mutant sample, before deleting anything, and treat unique killers as retained by default. Quarantine flaky tests before computing any similarity or failure-probability measure, since instability contaminates both. Replace immediate deletion with demotion to a shadow tier and a probation period, and publish the escape rate. Assign the deletion decision to an owner accountable for escapes, not to the tool that proposed the removal, and record the exchange rate between minutes and capability that the chosen knee implies. Test process standards already require that completion criteria and residual risk be documented; applying the same discipline to deletion is a modest extension (ISO/IEC/IEEE, 2018; Eze et al., 2022b). Teams adopting AI-assisted testing tools should expect those tools to propose reductions, and should require the same evidence of a proposed deletion whether a person or a model proposed it (Borah et al., 2022; Ominyi et al., 2025a; Nwakamma et al., 2024b). Trust in AI-supported monitoring depends on this kind of reviewable evidence rather than on reported accuracy alone (Akintola et al., 2025; Adams, 2023).

Adoption is an organizational matter as well as a technical one. Controls that impose visible burden without visible benefit invite workarounds, as compliance research on regulatory burden and on the motives behind voluntary compliance both show (Fobellah, 2025c; Alm, 2019). Deletion rules are more likely to be followed when leaders model them, when cross-functional review brings testers, developers and product owners into the same decision, and when the rules are built into the pipeline rather

than issued as policy (Arumosoye & Obriki, 2024; Anichukwueze et al., 2022; Akhigbe et al., 2024). Safety-culture research reaches a similar conclusion about hazard controls: they persist when ownership is clear, when training explains why a control exists, and when operational discipline is reinforced rather than assumed (Obogo et al., 2023; Obogo et al., 2019a; Obogo et al., 2019; Obriki et al., 2022a; Arumosoye et al., 2025). Standardized workflows in high-volume review settings likewise show that consistent procedure reduces variance in outcomes more than individual expertise does (Uchechi et al., 2018).

Counter-Arguments and Boundary Conditions

Four objections recur in debates about suite reduction. The first holds that compute is cheap and reduction is therefore unnecessary: run everything, in parallel. That is true for some organizations and false for others. Where end-to-end suites occupy hardware-in-the-loop rigs, device farms or licensed environments, execution capacity is scarce, and feedback latency shapes developer behaviour whatever the cost (Daka & Fraser, 2014; Kaner et al., 1999). Reviews of test automation in wireless hardware engineering show how scripting frameworks and instrument control strategies bind each test to shared bench equipment, so that suite size translates directly into instrument time (Quainoo et al., 2025b). The framework therefore applies most strongly where capacity is constrained, much as resilience planning for diagnostic infrastructure concentrates on the periods when capacity is under stress (Aminu-Ibrahim et al., 2025b). The second holds that selection and prioritization make reduction redundant, since a change-aware selector will skip irrelevant tests anyway. Selection helps, but it does not remove maintenance burden, and selectors depend on dependency analysis that is incomplete for reflection-heavy, configuration-driven or cross-language systems, which is exactly where a full suite is the last line of defence (Catal & Mishra, 2013; Li et al., 2007).

The third counter-argument holds that conservative guards will leave so many tests that reduction achieves little. The scenario in Section 5.3 suggests otherwise, since unique killers tend to be cheap, but the claim is empirical, and P3 and P6 are

designed to test it. The fourth holds that mutation analysis is too expensive for large suites. Mutant sampling and selective operators bring the cost down to a small multiple of a full suite run, paid once per reduction cycle rather than per build (Ammann & Offutt, 2016). Two boundary conditions limit the framework. It presumes reproducible builds with test-level results, and it presumes that a substantial part of the failure history is trustworthy. Where flakiness is endemic or result logs are incomplete, the prior task is stabilization and reconciliation of the records, and reduction should wait; risk-based assurance work makes the same point that controls cannot be rationalized on data that are themselves unavailable or unreliable (Fadayomi et al., 2019; Dogbatsey et al., 2019).

Limitations and Future Research Agenda

This section sets out the limitations of the review, a research agenda that would test the CPSR propositions, and the validity threats that such research must manage. The agenda is organized around the propositions in Table 3 so that each line of work has a clear target, and Figure 4 shows how the studies fit together as a single replay-based validation design. The emphasis on replay over historical versions reflects the synthesis finding that in-version evaluation is the most common source of optimistic loss estimates, and the emphasis on field comparison reflects the absence of published staged-deletion studies (Runeson & Höst, 2009; Wieringa, 2014).

Limitations

The review has four limitations. First, it is integrative and conceptual; it reports no new experiment, and the propositions are untested. Second, the reduction evidence base is dominated by benchmark programs in C and Java and by mutation operators designed for those languages, so transfer to dynamically typed languages, data-intensive pipelines and machine-learning components, whose tests assert statistical properties rather than exact outputs, is uncertain. Third, industrial reports of reduction are rare and often lack the design attributes needed for appraisal, so the adjudication in Section 4.2 rests mainly on controlled studies. Fourth, the search may

Table 4: Threats to validity in evaluating capability-preserving reduction, with mitigations

<i>Threat</i>	<i>Reduction-specific risk</i>	<i>Mitigation</i>
Construct	Mutants assumed coupled to real faults; test count used as cost	Validate on real regressions where available; report wall-clock TR
Internal	Same mutants used to reduce the suite and to measure loss	Disjoint mutant samples; real faults held out from reduction
External	Results from lean, generated or single-language suites generalized	Stratify by suite origin, language and fixture weight
Conclusion	Stochastic search; few real faults per version	Thirty or more runs, $\hat{\Delta}12$ effect sizes, intervals on escape rates
Operational	Flaky tests and infrastructure failures counted as escapes	Rerun-based flake triage before an escape is recorded

have missed work published under prioritization or selection labels that also performs deletion, and some adjacent-field sources admitted for measurement and governance claims were appraised against less demanding criteria than the testing studies (Dybå & Dingsøyr, 2008; Page et al., 2021). Conceptual frameworks for emergent practice carry a general risk of internal coherence outrunning empirical support, which the propositions are intended to expose (Komi & Ganiu, 2025).

Research Agenda

Five lines of research would test and extend the framework. A multi-project replay study using open-source histories with linked real regressions, in which reduction is performed at a historical version and loss is measured over later releases, would test P1 and P2 directly and should use time-ordered splits throughout. Controlled comparisons of kill-matrix, trace-similarity and greedy reduction at matched wall-clock savings, with thirty or more repeated runs and effect sizes, would test P3. Calibration studies of failure predictors stratified by component

history would test P4 and could borrow fairness-aware evaluation methods developed for forecasting across unequal populations (Komi, 2023). Cross-project predictors trained without pooling raw data, using federated learning of the kind proposed for cross-organizational analytics, would ease the low-history problem for new projects (Sanni et al., 2023). Field experiments comparing staged and immediate deletion across comparable services would test P6, with process tracing to confirm that escapes were caught by the shadow tier rather than by users. Finally, the framework should be extended to suites for machine-learning components and to tests generated by large language models, where the oracle problem is acute and model-based judges increasingly gate releases (Nwakamma et al., 2024a; Ominyi et al., 2024b). Effects measured under research conditions often shrink when interventions are scaled, so field replication should be planned from the outset (Miller et al., 2015). Figure 4 lays out the validation design that links these studies.

Threats to Validity in Evaluating Reduction

Evaluations of the kind proposed above face validity threats specific to reduction, summarized in Table 4. Construct threats arise when mutants stand in for real faults without evidence of coupling on the subject system, or when test count stands in for cost. Internal threats arise when reduction and evaluation share information, for example when the mutants used to build the kill matrix are also used to measure loss, which guarantees zero measured loss by construction; disjoint mutant samples or held-out real faults are required. External threats arise because suite origin, language and fixture weight shape results. Conclusion threats arise from the stochasticity of search-based methods and from the small number of real faults per version. Operational

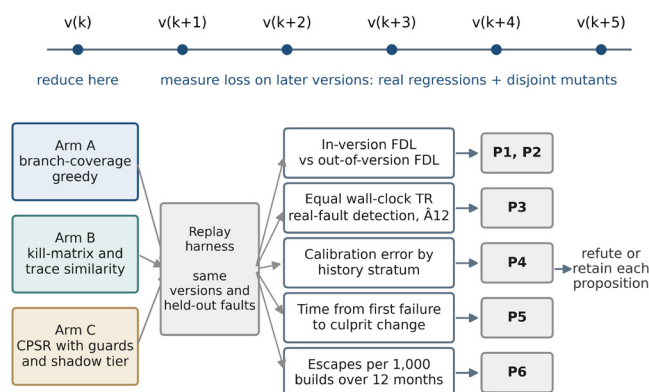


Figure 4: Replay-based validation design linking three reduction arms to the measures that test propositions P1 to P6

threats concern whether findings survive pipelines in which flakiness, parallel sharding and infrastructure failures masquerade as escapes (Runeson & Höst, 2009; Easterbrook et al., 2008). Continuous-control research adds that monitoring arrangements must themselves be audited, or they drift from the conditions under which they were validated (Adebayo et al., 2025; Dagodzo & Ahiaeké Patrick, 2020).

Conclusion

Permanent test suite reduction is justified only when the capability it removes is measured, bounded and recoverable. This review has shown that coverage equivalence, the redundancy criterion most tools apply, is a weak guide to behavioural equivalence, and that the long-running disagreement over fault-detection loss largely dissolves once suite origin, fault source, version horizon and baseline are made explicit. Behaviour-based and multi-objective methods retain more capability, provided their mutant sets are subsumption-reduced, their similarity measures are protected against flakiness and the point chosen on the Pareto front is justified. Failure prediction is a useful shield for tests that catch recurring faults and a poor instrument for deleting tests that guard new or rarely changing critical code. The Capability-Preserving Suite Reduction framework turns these findings into a procedure: conjunctive redundancy evidence, explicit retention guards, multi-objective selection over the remaining candidates, staged demotion with a shadow audit, and feedback from observed escapes into the guards. Its six propositions state what evidence would show it to be wrong. Teams that adopt it will delete fewer tests than a greedy reducer would suggest, recover most of the execution time that matters, and know, rather than assume, what their smaller suite can still detect.

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 reference list.

References

1. 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>
2. Adams, V. (2023). Preserving critical thinking in the era of generative artificial intelligence: Evidence-based frameworks for human-centered learning. *International Journal of Social Sciences and Management Research*, 9(11), 321-332. <https://doi.org/10.56201/ijssmr.v9.no11.2023.pg321.332>
3. 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>
4. 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>
5. 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>
6. Adegbite, M. P., Adebayo, A., & Ahmed, M. O. (2020). Securing operational technology networks in electric utilities: A systematic review of NERC CIP compliance and architectural threat mitigation. *International Journal of Engineering and Modern Technology*, 6(3), 103-146. <https://doi.org/10.56201/ijemt.vol.6.no3.2020.pg103.146>

- doi.org/10.56201/ijemt.vol.6.no3.2020.pg103.146
7. 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>
8. Afrihyia, E., Ojukwu, P. U., & Akinse, S. G. (2024). Privacy-preserving health data governance models: A comparative review of blockchain and cryptographic strategies in U.S. and developing healthcare systems. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3071-3086. <https://doi.org/10.62225/2583049X.2024.4.6.5919>
9. Ahiaeke 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>
10. 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>
11. Akanbi, O., & Sunday, E. A. (2024). An integrated path-planning and slotting optimization model for AMR-enabled high-density warehousing. *International Journal of Advanced Multidisciplinary Research and Studies*, 4(6), 3226-3243. <https://doi.org/10.62225/2583049X.2024.4.6.6181>
12. 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>
13. 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>
14. 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>
15. Aliliele, C., Mbonu, I. S., & Iwuanyanwu, U. (2023a). A conceptual framework for continuous cloud misconfiguration monitoring and enterprise risk mitigation strategies. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(10), 373-394. <https://doi.org/10.32628/CSEIT2361071>
16. 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>
17. 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>
18. Alm, J. (2019). What motivates tax compliance? *Journal of Economic Surveys*, 33(2), 353-388. <https://doi.org/10.1111/joes.12272>
19. Amadi, C., Adebayo, A., & Ijagbemi, A. O. (2025c). The cyber-physical attack surface of legacy SCADA networks: A systematic review of vulnerabilities, threats, and detection gaps in critical infrastructure. *International Journal of Computer Science and Mathematical Theory*, 11(8), 135-169. <https://doi.org/10.56201/ijcsmt.vol.11.no8.2025.pg135.169>
20. 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>
21. 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>
22. 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>
23. 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>
24. 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>
25. Ammann, P., & Offutt, J. (2016). *Introduction to Software Testing* (2nd ed.). Cambridge University Press.
26. 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>
27. 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>
28. 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>
29. 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>
30. 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>
31. 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>
32. 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>
33. 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>
34. 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>
35. 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>
36. 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>
37. 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>
38. 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>
39. 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
40. 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>
41. 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>
42. Cleland-Huang, J., Gotel, O., & Zisman, A. (Eds.). (2012). *Software and Systems Traceability*. Springer. <https://doi.org/10.1007/978-1-4471-2239-5>
43. Dagodzo, D., & Ahiaeké Patrick, M. C. (2020). UAV-based pipeline and corridor monitoring: A review of current practices and emerging technologies. *Iconic Research and Engineering Journals*, 3(10), 574-597. <https://doi.org/10.64388/IREV3I10-1716084>
44. 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>
45. 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>
46. Dogbatsey, E. A., Ebhojie, O., & Oyeleye, A. O. (2019). Reconciliation control and financial workflow redesign in emerging market organizations: A conceptual framework. Zenodo. <https://doi.org/10.5281/zenodo.20447103> \ [Originally numbered 3(5); journal name not captured in source.]
47. 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>
48. 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>
49. 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.
50. Edivri, J., & Oteri, O. (2024). A conceptual KPI-driven decision and optimization framework for IT service delivery, portfolio performance, and adoption. *Gyanshauryam, International Scientific Refereed Research Journal*, 7(1), 167-187. <https://doi.org/10.32628/GISRRJ246643>
51. 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>
52. 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>
53. 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>
54. 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>
55. 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>
56. 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>
57. 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>
58. 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>
59. 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>
60. 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>
61. Fobellah, A. N. (2025c). Unintended consequences of financial regulations: A study of corporate compliance burdens: A conceptual review. *International Journal of Science and Research Archive*, 16(2), 29-34. <https://doi.org/10.30574/ijrsra.2025.16.2.2275>
62. Fraser, G., & Arcuri, A. (2011). EvoSuite: Automatic test suite generation for object-oriented software. *ESEC/FSE*, 416-419. <https://doi.org/10.1145/2025113.2025179>

63. 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>
64. 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>
65. 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>
66. 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>
67. 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>
68. ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29119-2 Software and systems engineering: Software testing processes. International Organization for Standardization.
69. ISO/IEC/IEEE. (2018). ISO/IEC/IEEE 29148 Systems and software engineering: Life cycle processes and requirements engineering. International Organization for Standardization.
70. 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>
71. 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>
72. Kaner, C., Falk, J., & Nguyen, H. Q. (1999). Testing Computer Software (2nd ed.). Wiley.
73. 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>
74. Komi, N. M., & Ganiu, O. S. (2025). When universal programs deepen inequality: A critical review of intersectional frameworks for climate justice and energy poverty. International Journal of Advanced Multidisciplinary Research and Studies, 5(6), 2380-2412. <https://doi.org/10.62225/2583049X.2025.5.6.6458>
75. 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>
76. 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>
77. Li, Z., Harman, M., & Hierons, R. M. (2007). Search algorithms for regression test case prioritization. IEEE Transactions on Software Engineering, 33(4), 225-237. <https://doi.org/10.1109/TSE.2007.38>
78. Mairiza, D., Zowghi, D., & Nurmaliani, N. (2010). An investigation into the notion of non-functional requirements. ACM Symposium on Applied Computing, 311-317. <https://doi.org/10.1109/SAP.2010.5542002>

- org/10.1145/1774088.1774153
79. Mbonu, I. S., Aliliele, C., Iwuanyanwu, U., & Oluoha, O. M. (2018). A conceptual framework for legal and ethical risk modeling in enterprise data protection governance systems. *Iconic Research and Engineering Journals*, 2(2), 207-226. <https://doi.org/10.64388/IREV2I2-1714911>
80. Mbonu, I. S., Aliliele, C., Iwuanyanwu, U., & Uzoka, E. (2021b). Advances in artificial intelligence techniques for secure software testing and automated regression control mechanisms. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 7(5), 468-496. <https://doi.org/10.32628/CSEIT217565>
81. Mbonu, I. S., Iwuanyanwu, U., Aliliele, C., & Uzoka, E. (2022a). A review of data protection impact assessment models in multi cloud financial infrastructure systems. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 8(1), 589-623. <https://doi.org/10.32628/CSEIT25442>
82. 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>
83. Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). Wiley.
84. Nwakamma, S., Ojukwu, J., & Oyesiji, S. O. (2024a). LLM-as-a-judge for automated model governance: A review of methods, biases, and release-gating practices. *World Journal of Innovation and Modern Technology*, 8(6), 185-216. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg185.216>
85. Nwakamma, S., Ojukwu, J., & Oyesiji, S. O. (2024b). Securing agentic AI enterprise workflows against prompt injection, tool poisoning, memory manipulation, and excessive agency threats. *World Journal of Innovation and Modern Technology*, 8(6), 217-248. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg217.248>
86. 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>
87. 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>
88. 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>
89. 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>
90. 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>
91. 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>
92. 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>
93. 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>
94. Ogbete, J. C., Aminu-Ibrahim, A. Y., & Ambali,

- K. B. (2022). Design standards and operational planning frameworks for scalable blood collection networks. *Gyanshauryam, International Scientific Refereed Research Journal*, 5(6), 267-300. <https://doi.org/10.32628/GISRRJ225635>
95. 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>
96. Ogbete, J. C., Aminu-Ibrahim, A. Y., & Ambali, K. B. (2018). Optimizing laboratory spatial planning strategies to improve diagnostic accuracy, safety, and clinical throughput. *Iconic Research and Engineering Journals*, 2(1), 87-113. <https://doi.org/10.64388/IREV1I7-1713587>
97. Ogbole, J. I., Okoruwa, P. O., Babatope, O. M., & Oyewole, T. (2021). Developing an integrated data visualization model for continuous business performance monitoring and optimization. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 7(5), 441-467. <https://doi.org/10.32628/IJSRCSEIT>
98. Ogbole, J. I., Okoruwa, P. O., Babatope, O. M., & Oyewole, T. (2021). Developing an integrated data visualization model for continuous business performance monitoring and optimization. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 7(5), 441-467. <https://doi.org/10.32628/IJSRCSEIT>
99. 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>
100. Okonkwo, C. S., Ogunwale, O., Okeke, O. T., & Mayo, W. (2019). Conceptual framework for cost reduction through contract negotiation and vendor governance. *Iconic Research and Engineering Journals*, 2(9), 468-482. <https://doi.org/10.64388/IREV2I9-1713121>
101. 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.]
102. 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>
103. 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>
104. Ominyi, M., Anichukwueze, C. C., & Uzougbo, N. S. (2024b). A systematic review of organizational readiness for the EU Artificial Intelligence Act in multinational enterprises. *World Journal of Innovation and Modern Technology*, 8(6), 185-203. <https://doi.org/10.56201/wjimt.v8.no6.2024.pg185.203>
105. Ominyi, M. (2021). Toward a conceptual framework for embedding privacy by design in corporate restructuring and transaction structuring. *Journal of Accounting and Financial Management*, 7(5), 152-173. <https://doi.org/10.56201/jafm.vol.7.no5.2021.p152.173>
106. Onche, V. O., Adegbite, M. P., & Ogbonna, C. S. (2023). BYOD adoption and information-security risk in university administrative offices: Evidence from Nigerian federal universities. *International Journal of Multidisciplinary Research and Growth Evaluation*, 4(6), 1549-1563. <https://doi.org/10.54660/IJMRGE.2023.4.6.1549-1563>
107. Page, M. J., et al. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71. <https://doi.org/10.1136/bmj.n71>

108. Quainoo, R., Ogundapo, O., & Asiedu, W. A. (2024). Modeling the impact of impedance mismatch on wireless link performance: A framework for prototype development and system optimization. *International Journal of Computer Science and Mathematical Theory*, 10(2), 62-116. <https://doi.org/10.56201/ijcsmt.v10.no2.2024.pg62.116>
109. 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>
110. 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>
111. 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>
112. 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>
113. 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>
114. Sanni, J. O., Iwuanyanwu, U. A., Essien, M. A., & Attah, A. (2023). Lifecycle-aware marketing automation using federated learning for secure cross-organizational data management. *Gyanshauryam, International Scientific Refereed Research Journal*, 6(6), 337-364. <https://doi.org/10.32628/GISRRJ236642>
115. 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>
116. 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
117. 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>
118. 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>
119. 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>
120. 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>
121. 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>

122. 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>
123. 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>
124. 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>
125. Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer. <https://doi.org/10.1007/978-3-662-43839-8>
126. Yusuf, H. D., & Abdulrazaq, K. (2018). A conceptual framework for strengthening diagnostic quality systems in resource-limited clinical laboratories. *Iconic Research and Engineering Journals*, 1(11), 145-179. <https://doi.org/10.64388/IREV1I11-1723090>
127. Zhu, H., Hall, P. A. V., & May, J. H. R. (1997). Software unit test coverage and adequacy. *ACM Computing Surveys*, 29(4), 366-427. <https://doi.org/10.1145/267580.267590>