ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 6 Issue 2 April to June 2026
Paper Id: JETA-V6I2P114/ Page No: 124-134

Review Article

AI -Driven Regression Testing for Policy Lifecycle Scenarios in Multi-Line P&C Insurance

Kiran Babu Boddapati
Acharya Nagarjuna University, Andhra Pradesh, India

Received Date: 26 March 2026 Revised Date: 06 April 2026 Accepted Date: 23 April 2026

Abstract: *AI-assisted regression testing is beginning to transform enterprise insurance systems, especially property and casualty platforms in which policy-lifecycle behaviour spans quoting, binding, endorsements, renewals, reinstatements, cancellations, billing interactions, and claims-related updates across multiple product lines. In this environment, product variation, rule volatility, regulatory constraints, and workflow interdependence increase release risk in ways that cannot be adequately managed through static test inventories alone. This review analyzes literature relevant to AI-based regression testing in multi-line P&C insurance, bringing together the existing literature on regression test selection and prioritization, model-based and combinatoric testing, process mining, predictive process monitoring, explainable AI, and the digitalization of insurance. Major themes include scenario explosion, business-process conformance, risk-based prioritization, data quality, lifecycle-aware test generation, and governance in high-stakes change management. The reviewed studies suggest that AI-assisted regression strategies can improve defect-detection efficiency and test-planning support, although important gaps remain in domain-specific analysis, longitudinal study within policy-administration systems, and integration of process semantics with learned prioritization logic. This topic is becoming increasingly important because insurers now rely on rapidly changing policy platforms whose defects can propagate across underwriting, billing, compliance, and customer service with significant operational and regulatory consequences.*

Keywords: *Artificial Intelligence; Insurance Systems; Policy Lifecycle; Property and Casualty Insurance; Regression Testing*

I. INTRODUCTION

In enterprise software environments characterized by rapid change, strong interdependence, and high business exposure, regression testing has become a critical control mechanism. This is particularly a problem in policy administration platforms that are utilized by property and casualty insurance companies. These platforms support quoting, underwriting referral, policy issuance, endorsements, mid-term adjustments, cancellation, reinstatement, renewal, document generation, and integration with rating engines, claims systems, payment gateways, and regulatory reporting services. A change made to suit a single line of business or in one jurisdiction can cause behaviour change in another place on the platform due to shared elements, product inheritance, reuse of rule templates or shared service interfaces. The trade-off between cost, speed, and confidence has been a long-standing discussion in the literature on regression testing: it is typically impossible to do exhaustive retesting in large systems [1]. This was further emphasized by research on test case prioritization which revealed that sequencing and selection of tests have a significant impact in defect detection given a limited release window [2]. In insurance context, release windows are often compressed by rate changes, product revisions, statutory requirements, and carrier-specific configuration deadlines.

The relevance of this subject matter to the existing research environment is correlated with a larger change in software delivery in the enterprises. The insurance platform of the future is more often dependent on configurable product models, API-orchestrated, workflow engines and event-driven patterns of integrations. These architectures can accelerate product variation, but the same flexibility also expands the number of lifecycle paths that require validation. One individual auto endorsement can have implications on the coverage limits, recalculation of premium, instalments plans, generation of forms, commissions and downstream accounting. Commercial-lines renewal can call on the eligibility regulations, jurisdictional shapes, schedule-rating reasonableness, previous-term migration, and foreign third-party data enhancement. Traditional test suites built around isolated functions are unlikely to capture these longitudinal dependencies. Process-conformance studies show that actual behaviour in information systems often deviates from intended process models, and that event-based analysis can help detect path variation and control-flow deviation [3]. Predictive business-process monitoring extends this logic by using past execution traces to forecast future states and outcomes in running cases [4]. In the case of policy lifecycle, those approaches have a conceptual interface between the history of transactional events and the design of concerns with regression.

This topic is also important across software engineering, business-process management, machine learning, information-systems governance, and insurance operations. From a software-engineering perspective, the core question is how AI methods can improve regression-test selection, prioritization, generation, maintenance, and failure prediction in large-scale systems. In business-process terms the problem is how to maintain conformance in long-running, branching lifecycle situations where the correctness of the situation relies on a temporal ordering of the state along with transitions between states and not just individual, one-step outputs. The digitalization of insurance has changed value creation through configurable products, insurrect integration, data-driven customer interaction, and accelerated release cycles under continuing regulatory and trust constraints [5]. The landscape of multi-line P&C insurers is particularly challenging due to the possibility of having a common platform that can be used to service personal auto, homeowners, commercial auto, general liability, workers compensation, umbrella, inland marine or specialty products with different coverage logic and jurisdiction-specific requirements.

There are a number of issues that are yet to be resolved that are evident through this landscape. One of the problems is that of scenario explosion. The behaviour of policy lifecycle is combinatorial and the number of endorsements, effective-date regulations, billing states, changes of jurisdiction, optional coverages and user roles interact. The other problem is related to domain fidelity. Abstract benchmarks could be achieved well by general-purpose AI testing techniques, but they can fail to detect insurance-specific relationships like out-of-sequence endorsements, cancellation rescission, multi-policy account interaction or document-trigger obligation. The third problem is related to explainability and governance. The decision made in risk-based regressions within the regulated industries have to be traceable and more so where some scenarios are prioritized above others. The fourth problem is related to the data foundation. History logs of events, requirement traces, defect histories and snapshots of policy states are frequently not complete or consistent and restrict the usefulness of learned models. A final problem is the lack of connection between metrics of software-testing and business-operational implications. A release should appear to be technically perfect but could introduce defects to the issue, premium accuracy, compliance paperwork or downstream financial controls.

The aim of this review is to analyze how peer-reviewed scholarship on AI-based regression testing can inform policy-lifecycle scenarios in multi-line P&C insurance. The review is concerned with the selection and prioritization of regression tests, industrial test optimization, model-based and combinatorial, process mining, predictive lifecycle monitoring, explainable AI and insurance digitalization as additional sources of knowledge on the area. The literature base is analyzed in the sections below, the conceptual and methodological approaches that prevail reported are organized, reported gaps and findings are discussed, the future research requirements are outlined, and the conclusion section states the main implications on how to design lifecycle-aware regression strategies in insurance platforms.

II. LITERATURE REVIEW

The literature relevant to AI-driven regression testing of insurance policy lifecycles does not belong to a single mature subfield. Rather, it is cross-functional, touching a variety of related fields: regression test engineering, model-based and combinatorial testing, process mining and predictive process monitoring, explainable analytics, and insurance digitalization. The strongest foundation comes from the regression-testing literature Regression test selection, minimization, and prioritization emerged in response to the infeasibility of rerunning entire test inventories after every change [1], [6]. Empirical analyses of prioritization show that the early detection of faults can be enhanced through informed sequencing approaches as opposed to random or arbitrary execution sequence [7]. Search-based methods built upon this by demonstrating that the optimization algorithms can be more effective than the simple heuristics when there are several goals, like coverage and defect exposure, which need to be traded as release constraints are present [8]. Subsequent reviews showed growth in prioritization research but also highlighted fragmented evaluation practices and limited evidence from highly regulated business domains [9].

The second important stream is the one dealing with industrial optimization and test-maintenance realism. Studies on the optimization of test suites in industrial systems indicated that test suites of large sizes can be reduced or rearranged based on the historical execution data, requirement relations and cost-sensitive approaches [10]. Mult release regression suites work also emphasized that industrial test suites do change over time, and tend to become more and more redundant, brittle, and irrelevant across releases [11].

These conclusions are of interest especially in insurance settings since the platforms of policies tend to remain over several years as products, jurisdictions and business regulations vary continuously. A test suite assembled at one stage of product evolution may no longer align with current operational risk unless its composition reflects contemporary lifecycle use and defect concentration. This is consistent with automated test-generation research showing that automation adds value only when behaviour, constraints, and expected outcomes remain aligned with system evolution [12]. This observation has a lot of weight to insurance platforms that are constructed on customizable product models and externalized rules.

The third stream relates to the scenario modelling. Formal or semi-formal descriptions of system behaviour, model-based testing offers a way to generate test artifacts, unlike hand-written scripts [13]. Combinatorial testing can solve the interaction spaces where failures can be based on specific combinations of inputs, states, or configuration variables [14]. These two traditions are especially relevant to policy lifecycles because insurance defects often arise not from single factors, but from combinations of line of business, coverage options, billing state, user role, jurisdiction, effective date, and transaction type.

An endorsement, which is correct on a simple path, can prove unworkable only when it is used with instalment billing, has a history of cancellation before, or has a certain effective-date limit. According to the literature, model-based and combinatorial approaches may help to eliminate blind spots in these interaction-intensive spaces, but both approaches require accurate abstractions of the domain. Practically, lifecycle semantics of P&C insurance can be more difficult to formalize than tasteful behaviour of more limited software realms.

There is a fourth stream that is based on process mining and predictive process monitoring. Research on conformance-checking revealed that event logs can be re-executed on process models, to detect deviations between planned and actual execution [3], [15]. This was further developed in predictive monitoring studies that would learn using historical traces to generalize the remaining time, future occurrences or case results in running process instances [4], [16]. Event-log quality research also shows that partially recorded, noisy, or inconsistent traces can distort downstream analysis and model performance [17]. These contributions are especially useful when it comes to policy administration systems since lifecycle situations are inherently processual. There is no such thing as a single state policy, it follows quote, bind, issue, change, bill, renew or terminate processes, and like any other software it has cross-system side effects and branch points. Process-conscious analytics is thus in a position to reveal the high-risk processes which are missed by the traditional module-based test inventories. Despite this, there is still a lack of direct application to regression testing in literature and there is a dearth of insurance-specific studies.

One of the fifth streams is about governance, explainability, and digitalization of insurance. The research on explainable AI asserts that the use of opaque models can restrict their use and responsibility in situations where the results of automated systems may affect consequential choices [18]. This is not just an academic issue in the context of regression planning of regulated industries.

Such a prioritization model, in which some lifecycle scenarios are de-emphasized, has to be able to justify itself defensibly in case some subsequent software defect impacts compliance or customer results. The study of insurance digitalization also suggests that the carriers are transforming the nature of products, channels, and data applications as well as value-chain relationships, thus making the pace of change faster across the core platforms [5], [19], [20]. This increases the scope of configurability and integration to enlarge the regression risk attack surface. As a result, the literature indicates that there is a need to have both adaptive and auditable testing methods.

Table 1 is a summary of some of the representative studies beginning with reference [6]. The table highlights the breadth of the evidence base and shows that both direct regression-testing studies and adjacent fields concerned with scenario behaviour, process visibility, and governance make relevant contributions.

There are a number of cross-cutting patterns that emerge from this literature. To begin with, the majority of regression-testing studies are made on the assumption that generic software systems are used instead of regulated, product-configurable insurance platforms. Second, process-monitoring studies offer good lifecycle visibility with without typically converting process knowledge into regression-test properties.

Third, insurance digitalization literature talks about increasing complexity and the pace of change but seldom speak in detail about testing methodology. Fourth, issues of governance seem to be becoming more and more relevant with AI involved in release decision making. These trends indicate a major research gap: there is still no well-developed domain-specific knowledge on lifecycle-aware AI regression techniques for multi-line P&C policy systems.

The other limitation that is recurrent is on evaluation. Most software-testing research uses measures of fault detection, execution cost or coverage surrogates. The emphasis of process-mining studies is on conformance, quality of prediction, or trace analytics. When researching insurance, transformation, capability or operating-model change is frequently in the spotlight.

There is a dearth of research linking these layers to a single evaluation scheme capable of measuring whether a regression-testing approach is able to secure operationally significant lifecycle outputs like correct rating, legal-compliant documentation, and policy sustenance, billing accuracy or renewal integrity. This gap makes it difficult to translate generic AI-testing results directly into insurance practice and motivates the conceptual framework developed in the next section.

Table 1: Summary of key findings

Ref	Focus	Key Findings
[6]	Regression test selection techniques	Established that selective regression execution can reduce cost substantially, though empirical comparison remains sensitive to program structure and change profile.
[7]	Empirical test case prioritization	Showed that informed prioritization can detect faults earlier than untreated execution order in repeated regression contexts.
[8]	Search-based regression test prioritization	Demonstrated that optimization-based ordering can improve early fault revelation under constrained execution budgets.
[9]	Prioritization literature review	Reported wide methodological diversity, limited domain realism, and a persistent need for stronger empirical baselines.
[10]	Industrial test suite optimization	Found that execution history, change information, and cost-aware filtering can improve regression efficiency in industrial settings.
[11]	Multirelease regression suite analysis	Identified temporal drift, redundancy accumulation, and maintenance burden in long-lived industrial regression assets.
[12]	Automated software test case generation methods	Mapped a broad set of automation strategies and emphasized representation quality as a key determinant of usable test generation.
[13]	Model-based testing taxonomy	Clarified behavioural modelling alternatives and showed how formalized state and transition structures can improve scenario derivation.
[14]	Combinatorial testing survey	Demonstrated the value of interaction coverage for fault discovery in systems shaped by multi-factor input combinations.
[15]	Process-model replay for conformance and performance analysis	Showed that event-log replay can reveal execution deviations and control-flow mismatches relevant to scenario validation.
[16]	Outcome-oriented predictive process monitoring	Reported that trace-based prediction can anticipate process outcomes, though performance depends strongly on trace quality and feature design.
[17]	Event-log imperfection patterns	Identified systematic quality defects in operational event logs that can distort process analytics and predictive methods.
[18]	Explainable AI in decision support	Argued that transparent model reasoning improves trust and accountability for consequential automated recommendations.
[19]	Insurtech and insurance value creation	Described how digital insurance innovation alters product design, service integration, and operational complexity across the value chain.
[20]	Digitalization effects on insurance companies	Reported that digital technologies reshape underwriting, service, and operations while increasing transformation pressure on legacy systems.

III. MATERIALS AND METHODS

A useful conceptual framework for AI-based regression testing in multi-line P&C insurance should begin from the premise that policy behaviour is lifecycle-based rather than function-based. A rating service, endorsement API, billing connector or forms engine can be technically separate, but production risk typically manifests itself via a scenario path which intersects various services over time. The regression design would therefore consider the lifecycle of the policy as the main unit of analysis and individual tests, traces, and business rules are the observables of the unit. Regression-testing literature contributes mechanisms for selection and prioritization [1], [6], [8], while process-aware literature provides methods for reconstructing and predicting scenario trajectories from event histories [4], [15], [16]. The insurance digitalization literature provides the business case of why such integration is necessary to demonstrate that product and channel modernization has raised configurational complexity along core platforms [5], [19], [20]. Such a combination facilitates a lifecycle-conscious system where AI helps to find scenarios, estimate risks, and order executions.

The literature points to several methodological families. The first is change-based and coverage-based regression analysis. In this case, the tests are chosen or given priority, depending on the modified code, dependency graph, requirement links or historical fault sites [6], [7], [9]. This family is useful because it bases execution decisions on observed system change. Nevertheless, insurance platforms tend to be extremely configuration-based, relying on rule tables, and product models, the effects of which may not be entirely apparent in code-based dependency analysis. The second one is searching based and optimization-based prioritization, where algorithms identify execution orders that optimize a certain goal, like early fault detection, or weighted business risk [8]. These approaches can be adapted to insurance policy design by using scenario criticality, premium volume, regulatory sensitivity, and transaction frequency as optimization signals. Nonetheless, the quality of optimization is related to the relevance and stability of these signals.

A third family consists of model-based and combinatorial scenario generation. The policy transactions could be expressed in model-based testing in terms of states and transitions to generate the paths of scenarios that reflect the lifecycle flow [13]. This can then be further narrowed down to combinatorial methods that concentrate on the interaction between coverage options, user channels, effective dates, jurisdictions, payment plans, and configuration that are unique to lines [14]. The combination is especially appealing to multi-line P&C insurance since defect-prone conditions can often be due to a combination of multiple variables, as opposed to a single input. However, formal models may be hard to sustain in the fast changing policy systems. Excessive configurability can result into model explosions unless abstractions are kept in check and are made to comply with business semantics.

The fourth family comes under process-aware AI. Conformance checking is a method used to compare the event logs with the desired behaviour with the view of identifying deviations along the path [3], [15]. Predictive process monitoring resorts to the past traces to predict the future, subsequent steps or performance features of ongoing cases [4], [16]. Event-log quality research provides an important warning: predictive and conformance techniques are only as reliable as the integrity of the underlying traces [17]. To apply regression testing, process-aware AI is particularly promising since production running logs have empirical information of what lifecycle paths are frequently followed, what lifecycle paths have exception density, and what sequence of transactions are associated with downstream rework or defects. This information has the potential to enhance the traditional regression planning by going beyond the conventional inventories to prioritizing policy scenarios on the basis of evidence. This restriction is that production logs indicate actual use, and not necessarily uncommon but critical compliance or emergency situations.

The conceptual framework derived from this literature is shown in Figure 1. The framework connects change events, lifecycle models, operational traces, AI risk estimation as well as governed regression decisions in a feedback loop. The coordinated utilization of structural change signals, the business-process history and auditable prioritization logic is an important lesson of the reviewed literature reflected in this set-up.

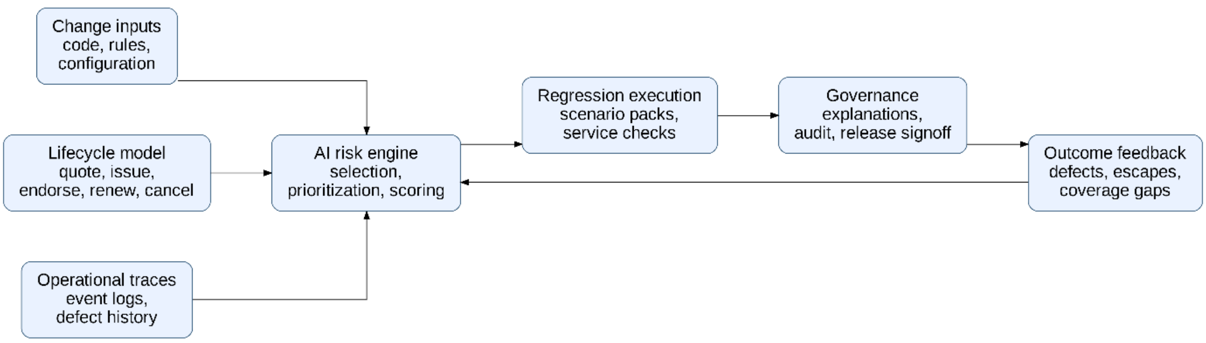

Figure 1: Conceptual framework for AI-driven lifecycle-aware regression testing in multi-line P&C insurance

The figure indicates that regression planning based on AI is not synonymous with a generic test-ordering utility. The logic of scoring is not only dependent on the structure of business lifecycle but also dependent on the operational trace evidence and governance is not a peripheral review process but an official layer. This is particularly crucial in insurance releases where scenarios which are omitted could have compliance and customer-service consequences.

There are a number of patterns of methods that are prominent in literature. Hybrid designs appear more realistic than single-technique approaches. Change-based selection can reduce search space, process mining can highlight path relevance, model-based logic can preserve domain semantics, and AI ranking can allocate scarce execution time, path relevance can be induced by process mining, domain semantics can be preserved by model based logic and limited execution time can be allocated by AI ranking. The other pattern is with regard to temporal feedback. Long-lived enterprise suites have the advantage of learning loops in which program execution outcomes, traces of production incidents and escaped bugs are the basis of a recalibration of future priorities [10], [11]. A final methodological trend concerns explainability. Where some of the test planning is automated, the stake holders must be provided with obvious indicators of why a renewal path is more preferred than an endorsement path or why one line of business is getting more intensive regression coverage than the other [18]. The results are discussed in the discussion that follows these observations.

IV. RESULTS AND DISCUSSION

Studies in software testing and process analytics suggest that AI-assisted regression planning can improve efficiency, although the magnitude of that improvement depends heavily on representation quality, domain structure, and evaluation design. Empirical studies of prioritization research always revealed that informed ordering had the ability to detect faults earlier in comparison to untreated execution sequences [7], [8]. The existence of systematic reviews of regression selection and prioritization also points out that it is possible to reduce costs when the execution is based on the impacted or high-value tests instead of the repetition of the entire suite [6], [9]. Even on their own, these findings justify serious attention to AI-driven approaches. However, the insurance domain introduces an additional layer of situational dependence. Fault significance is not determined by code exposure alone; it also depends on lifecycle position, customer impact, regulatory sensitivity, and propagation into billing, forms, commissions, and downstream financial systems. This can lead to a strategy that works well on generic benchmark programs but poorly in policy platforms in case lifecycle semantics are not well represented.

Another useful output of industrial optimization studies is that the historical execution and defect information can be used to make regression choices in case of large and long-lived test suites [10], [11]. This applies to multi-line P&C insurance since these environments can end up having thousands of automated and semi-automated tests and multiple release cycles. Uneven historical value among tests can be due to product variation, stateful transactions and environment-specific dependencies. The AI techniques can be used to differentiate between scenarios that are often stable and regions with persistent breakage, which in turn can be useful depending on whether the historical failure patterns are still representative after a significant revision of rates, reconfiguration of the product or an integration change of platforms. Literature therefore supports only a qualified conclusion. Historical learning is a potent thing, but it loses its power once change change's causal structure to a greater extent than can be reflected in past data.

Model-driven and combinatorial studies also add an additional understanding of the breadth of lifecycle scenarios. Model-based testing provides a rigorous manner of characterizing state transitions like quote-to-bind, bind-to-issue, issue-to-endorse or cancel-to-reinstate [13]. There are combinatorial techniques that are utilized to handle interaction spaces, such as line of business, coverage form, billing option, jurisdiction, payment status, and effective-date conditions [14]. The combination of these approaches offers a plausible response to the scenario explosion characteristic of insurance platforms. Nevertheless, it is also evident in reported studies that abstraction decisions are conclusive. Models that are too coarse may be missing details of defect prone paths and models that are too fine will be too expensive to maintain. Layered abstraction or more precisely, high-level lifecycle models to frame scenarios and targeted combinatorial expansion at high-risk areas of interaction like endorsements, renewals, cancellations or reinstatements, seems the most promising direction in the insurance context.

Literature on processes provides a solid conceptual contribution especially in insurance cases. Conformance-checking research demonstrates that event logs are capable of reflecting discrepancy between planned process designs, and actual execution behaviour [3], [15]. Further studies on predictive process monitoring indicate that the traces of the past can be used to estimate future moves or risk of the outcome in a case under consideration [4], [16]. When applied to regression testing, this implies that production trace data can tell what policy paths should be retested more frequently or sooner after a change. An example of such findings that a carrier would make includes: a relatively small group of endorsement patterns makes a disproportionate number of downstream exceptions; or that renewal situations with particular underwriting states have a high rate of defect escape. AI-driven prioritization can be fed with such insights. Meanwhile, event-log quality studies caution that such inferences can be biased by a lack of timestamps, irregular identifiers, or incomplete connections of transactions [17]. Insurance platforms typically contain heterogeneous logging conventions across policy, billing, and document subsystems, making trace curation a critical prerequisite.

Another large dimension of reported findings seems to be explainability and governance. Explainable AI studies maintain that understanding reasoning is beneficial in enhancing trust and in responsibly utilizing machine-generated recommendations [18]. This is important in the context of regression planning, as even not taking tests is a decision that entails risk. Relevant parties might not be willing to embrace AI assistance unless the model can point to the indications of change signals, process history or business criticality factors that informed the ranking of scenarios. This fact is reinforced by the insurance digitalization research that demonstrates that digital operating models enhance speed and dependency of core systems [5], [19], [20]. Increased release velocity does not do away with accountability but rather increases the cost of weakly-justified automation. This means that AI-based regression in insurance should be evaluated not only by early fault detection, but also by explanation quality, governance fit, and consistency with release-control practice.

Table 2 is a comparison of major methodological approaches that are discussed in the literature. The analysis shows that both sets of methods have their own benefit but have certain drawbacks. There is no one single strategy that best suits the entire lifecycle, compliance and integration issues that are evident in multi-line P&C settings.

Table 2: Method comparison

Ref	Method	Strengths	Limitations
[6]	Regression test selection based on change impact	Reduces execution burden by focusing on affected areas after modification	Can miss indirect business-process effects when product logic resides in rules or configuration
[7]	Empirical test case prioritization	Improves early fault revelation under limited release windows	Prioritization value depends on stable relevance signals and realistic fault proxies
[8]	Search-based prioritization	Supports multi-objective ordering using weighted risk and coverage criteria	Optimization quality can degrade when business weights are incomplete or unstable
[10]	Industrial test suite optimization	Uses execution history and cost signals to improve suite efficiency in large environments	Historical data may underrepresent rare but severe lifecycle scenarios
[11]	Multirelease suite analytics	Highlights temporal drift, redundancy, and maintenance burden in long-lived suites	Descriptive insights alone do not generate domain-aware future prioritization
[13]	Model-based testing	Captures lifecycle state transitions and supports structured scenario derivation	Model maintenance becomes difficult in highly configurable insurance products
[14]	Combinatorial testing	Exposes interaction faults across multiple input and state dimensions	Interaction spaces still require careful constraint modelling to avoid unrealistic combinations
[15]	Process conformance checking	Identifies divergence between intended and actual lifecycle execution paths	Requires high-quality event logs and well-defined process models
[16]	Predictive process monitoring	Learns from trace history to estimate high-risk outcomes and future path behaviour	Prediction quality varies with feature engineering, trace completeness, and path drift
[18]	Explainable AI for decision support	Strengthens accountability and stakeholder confidence in automated prioritization	Explanation methods can add complexity and may not fully reflect model internals

Another implication of literature is that evaluation metrics should be broader. Measures of standard regression-testing like APFD, reduction in the size of suites or saving in execution-time are still useful [7], [8], [9]. Nevertheless, risk cannot be fully reflected in those measures in policy administration systems. A defect that impacts the cancellation notices, calculation of premiums or even renewal eligibility can have a more significant business impact than a number of lower severity technical failures.

The evaluation based on insurance would thus involve a scenario criticality, volume of transactions, regulatory significance and downstream propagation in the scoring system. Such domain-specific metrics are rarely reported, creating a significant disconnect between general AI-testing research and insurance practice.

Table 3 is a summary of typical findings that are reported in the literature. The table shows a general tendency toward positive findings for informed regression techniques, model-based scenario design, and process-aware analytics, while also making clear that the outcome categories remain heterogeneous. There are those that are dealing with early fault detection, others with cost efficiency, others with process visibility and others with accountability.

Table 3: Results comparison

Ref	System	Metric	Outcome
[7]	Regression testing environment	Early fault detection	Prioritized execution detected faults earlier than untreated ordering in empirical settings
[8]	Search-based prioritization framework	APFD-style effectiveness	Optimization-based ordering improved early defect revelation relative to baseline heuristics
[10]	Industrial regression suite	Execution efficiency	History- and cost-aware optimization reduced unnecessary regression effort in industrial practice
[11]	Long-lived industrial test inventory	Suite evolution characteristics	Temporal analysis revealed redundancy growth and declining relevance across releases
[13]	Model-based testing setting	Scenario derivation quality	Formalized behavioural models improved systematic coverage of state-transition behaviour
[14]	Interaction-heavy software system	Combination coverage	Interaction-focused design increased exposure to faults triggered by factor combinations
[15]	Process-aware information system	Conformance deviation detection	Event-log replay exposed execution paths not aligned with intended process behaviour
[16]	Predictive process monitoring environment	Outcome prediction quality	Trace-based models estimated future process outcomes with useful but data-sensitive accuracy
[17]	Event-log analysis setting	Log-quality impact	Imperfect logs materially degraded process-analysis reliability and downstream interpretation
[18]	AI-assisted decision context	Interpretability and trust	Transparent recommendation logic improved acceptability for consequential automation

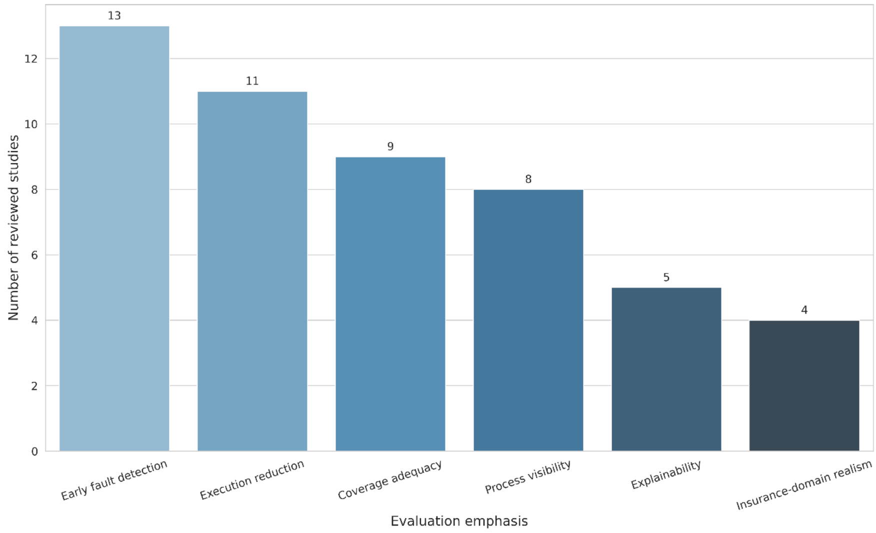

Figure 2: Distribution of reviewed journal studies by year

The literature review results in a coded trend graph (Figure 2) indicating which evaluation emphases seem to be most frequently mentioned as the main concerns. The graph indicates that the efficiency of fault-detection and a decrease in the execution are put in the first place, and the quality of the explanations and the realism of insurance domain are comparatively not directly considered. This bias helps explain the fact that the research on generic regressions does not completely apply to the situation with policy-life cycles.

Figure 2 highlights an important pattern: research on technical efficiency is comparatively mature, whereas domain-specific lifecycle realism remains underdeveloped. This gap is large in the case of multi-line P&C insurance because defects that are critical to the business will occur in infrequent but high impact scenario paths, but not in frequent technical functions.

Figure 3 superimposes the correlations among change signals, the complexity of the lifecycle, and quality of event-logs, AI ranking logic and release confidence. The relationship diagram justifies why, despite its advanced models of prioritization, the poor quality of inputs can still sabotage the advanced models. It also shows the rationale behind the pivotal role that the policy lifecycle semantics should play in the insurance regression design.

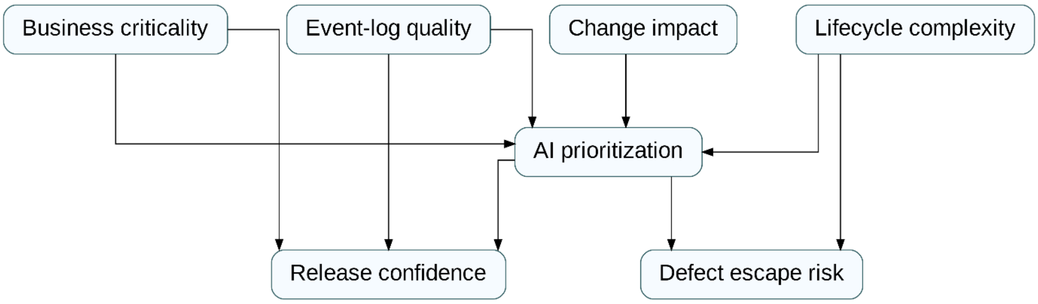

Figure 3: Relationship diagram among major variables in AI-driven insurance regression testing

The diagram reiterates the fact that it is not just test volume that generates release confidence. The aspect of business criticality, event-log integrity, and the complexity of the lifecycle determine whether AI prioritization is reliable or not. This reinforces the claim that insurance regression must be cross-functional as opposed to deploying individual algorithms.

Figure 4 introduces a combined lifecycle-conscientious AI regression in policy administration platforms. The model integrates structural change analysis, scenario modelling, operational-trace learning, controlled prioritization, execution and post-release feedback. This kind of arrangement is consistent with the most powerful consistent lesson of literature: successful regression programs are taught by evolution of software as well as by the production behaviour.

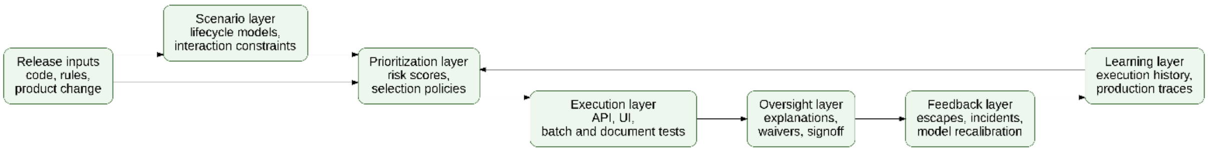

Figure 4. Integrated model for AI-driven regression testing across the policy lifecycle

The combined design specifies two feasible design requirements. To begin with, business semantics and empirical evidence must not be confused, scenario modelling and learning must be separate and yet connected layers to give each other an insight. Second, accountability must remain explicit, because release decisions in insurance are tightly linked to risk acceptance, regulatory sensitivity, and operational responsibility.

On the whole, the literature suggests that AI-based regression testing should be regarded as a decision-support problem as opposed to the limited automation problem. Combination of historical execution data, business-process visibility and structured scenario representations have been observed to perform well. The weak implementation areas are considered to be where no domain semantics is there, no event data, and performance is measured by generic software metrics. Multi-line P&C insurance thus poses a challenging application scenario as well as a research opportunity.

V. FUTURE DIRECTIONS

Future progress will depend on closer integration between process intelligence and regression engineering. The tests prioritization and the augmentation of complexity in monitoring predictive processes are mature techniques that have been established in literature, but without a lot of direct relationship. An avenue of research is the use of policy event traces to identify traces of lifecycle paths of high operational criticality and transfer the lifecycle paths to AI-based regression selection and ranking. This would shift release assurance away from purely code-centric impact analysis toward scenario-centric assurance.

A second priority will be in terms of domain specific representations of multi-line P&C products. The existing regression studies are usually based on generic modules or benchmark systems. Better abstractions that surround the kind of transactions, jurisdiction regulations, effective date semantics, underwriting status, document triggers, and cross system side effects are required in insurance systems. Layered scenario models, constraint-based combinatorial generation, and product-line-based lifecycle graphs may prove useful in the control of the exponential growth of scenarios without undermining business realism.

A third priority concerns explainability, governance, and risk tolerance. In practice, AI-driven prioritization will not be adopted unless release teams can understand why certain scenarios are prioritized and others are not. The future research must investigate the styles of explanations, which fit well in the quality engineering, product control and audit inspection. Uncertainty estimates, the importance of the business and trace based rationale can be of particular use in controlled business environments.

A fourth priority concerns evaluation reform. The early fault-detection measures that are used in combination with a measure of severity of escaped-defects, relevance to compliance, premium-impact risk and lifecycle-path coverage may be useful in future research. Longitudinal industrial research on the policy administration contexts is also demanded. This could be used to learn that AI assistance merely accelerates the process or, in fact, improves the quality of release to endorsements, renewals, cancellations, and reinstatements and other policy modifications which form the actual operations of the insurance.

VI. CONCLUSIONS

AI-enhanced regression testing for policy-lifecycle scenarios has strong potential in multi-line P&C insurance, although its value depends on more than simply automating faster execution or reducing the number of tests. The literature review indicates that the effectiveness of regression increases with the informed selection and prioritization based on the analysis of change, optimization techniques, and past performance of execution. Process-aware techniques add an important lifecycle dimension by revealing path behaviour, conformance gaps, and trace-based signals of future risk.

A number of important lessons can be learned based on the review. First, policy platforms should be viewed as lifecycle systems in which it is the scenario paths that become the source of risk, and not a particular function. Second, model-based and combinatorial methods are still relevant in modeling heavy policy behaviour in terms of interactions. Third, regression planning can be complemented by process mining and predictive monitoring which make prioritization based on the observed traces of operations. Fourth, when AI influences release decisions in regulated insurance environments, it must be explainable and well governed.

Continuous gaps are still of concern. Direct peer-reviewed study of AI regression testing in P&C policy administration remains limited. There is a lack of domain-specific evaluation measures, and the quality of operational trace data remains a significant constraint. Little cross-layer evidence on how software metrics are related to insurance-operational outcomes also exists.

Despite these shortcomings, the discipline has come to an extent where a coherent agenda can be seen. The next steps are probably life cycle-sensitive models, trace-guided prioritization, enhanced governance structures and longitudinal industry analysis. In that broader context, AI-aided regression testing in insurance is evolving not merely into a technical-efficiency exercise, but into a risk-based practice of release assurance for complex policy ecosystems.

Interest Conflicts

The author declares that there is no conflict of interest concerning the publishing of this paper.

VII. REFERENCES

[1] Kazmi, R., Jawawi, D. N. A., Mohamad, R., & Ghani, I. (2017). Effective regression test case selection: A systematic literature review. *ACM Computing Surveys, 50*(2), Article 29.

[2] Mukherjee, R., & Patnaik, K. S. (2021). A survey on different approaches for software test case prioritization. *Journal of King Saud University - Computer and Information Sciences, 33*(9), 1041–1054.

[3] Pegoraro, M., Uysal, M. S., & van der Aalst, W. M. P. (2021). Conformance checking over uncertain event data. *Information Systems, 102*, Article 101810.

[4] Márquez-Chamorro, A. E., Resinas, M., & Ruiz-Cortés, A. (2018). Predictive monitoring of business processes: A survey. *IEEE Transactions on Services Computing, 11*(6), 962–977.

[5] Eling, M., & Lehmann, M. (2018). The impact of digitalization on the insurance value chain and the insurability of risks. *The Geneva Papers on Risk and Insurance - Issues and Practice, 43*(3), 359–396.

[6] Garousi, V., Özkan, R., & Betin-Can, A. (2018). Multi-objective regression test selection in practice: An empirical study in the defense software industry. *Information and Software Technology, 103*, 40–54.

[7] Hao, D., Zhang, L., Zang, L., Wang, Y., Wu, X., & Xie, T. (2016). To be optimal or not in test-case prioritization. *IEEE Transactions on Software Engineering, 42*(5), 490–505.

[8] Zheng, W., Hierons, R. M., Li, M., Liu, X., & Vinciotti, V. (2016). Multi-objective optimisation for regression testing. *Information Sciences, 334–335*, 1–16.

[9] Khatibsyarbini, M., Isa, M. A., Jawawi, D. N. A., & Tumeng, R. (2018). Test case prioritization approaches in regression testing: A

systematic literature review. *Information and Software Technology, 93*, 74–93.

[10] Dobslaw, F., Wan, R., & Hao, Y. (2023). Generic and industrial scale many-criteria regression test selection. *Journal of Systems and Software, 205*, Article 111802.

[11] Minhas, N. M., Petersen, K., Börstler, J., & Wnuk, K. (2020). Regression testing for large-scale embedded software development – Exploring the state of practice. *Information and Software Technology, 120*, Article 106254.

[12] Brunetto, M., Denaro, G., Mariani, L., & Pezzè, M. (2021). On introducing automatic test case generation in practice: A success story and lessons learned. *Journal of Systems and Software, 176*, Article 110933.

[13] Ahmad, T., Iqbal, J., Ashraf, A., Truscan, D., & Porres, I. (2019). Model-based testing using UML activity diagrams: A systematic mapping study. *Computer Science Review, 33*, 98–112.

[14] Wu, H., Nie, C., Petke, J., Jia, Y., & Harman, M. (2020). An empirical comparison of combinatorial testing, random testing and adaptive random testing. *IEEE Transactions on Software Engineering, 46*(3), 302–320.

[15] Zaman, R., Hassani, M., & van Dongen, B. F. (2023). Conformance checking of process event streams with constraints on data retention. *Information Systems, 117*, Article 102228.

[16] Teinemaa, I., Dumas, M., La Rosa, M., & Maggi, F. M. (2019). Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Transactions on Knowledge Discovery from Data, 13*(2), Article 17.

[17] Suriadi, S., Andrews, R., ter Hofstede, A. H. M., & Wynn, M. T. (2017). Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. *Information Systems, 64*, 132–150.

[18] Rai, A. (2020). Explainable AI: From black box to glass box. *Journal of the Academy of Marketing Science, 48*(1), 137–141.

[19] Stoeckli, E., Dremel, C., & Uebernickel, F. (2018). Exploring characteristics and transformational capabilities of insurtech innovations to understand insurance value creation in a digital world. *Electronic Markets, 28*(3), 287–305.

[20] Eckert, C., & Osterrieder, K. (2020). How digitalization affects insurance companies: Overview and use cases of digital technologies. *Zeitschrift für die gesamte Versicherungswissenschaft, 109*(5), 333–360.