COLOMBO SCIENTIFIC PUBLISHING

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE & INFORMATION SYSTEM

E-ISSN: 2536-7919

P-ISSN: 2536-7900

PAGE NO: 89-95

Accelerating Secure, Resilient, and Intelligent Product Development: Integrating AI, Edge Computing, and DevSecOps for Reduced Time-to-Market and Enhanced Reliability

Dr. Helena Martínez

Global University of Applied Sciences, Spain

ARTICLE INFO

Article history:

Submission: October 01, 2025
Accepted: October 15, 2025
Published: October 31, 2025
VOLUME: Vol.10 Issue 10 2025

Keywords:

Time-to-market; DevSecOps; AI in EDA; Edge computing; Formal verification; Dual sourcing; Fault-tolerance

ABSTRACT

The contemporary landscape of product development and system engineering is characterized by converging pressures: the need to reduce time-to-market, to assure security and reliability across increasingly complex hardware–software stacks, and to leverage artificial intelligence (AI) and edge computing effectively. This manuscript synthesizes theoretical perspectives and applied findings from a curated set of references spanning product development lifecycle acceleration, formal verification, DevSecOps integration, AI for hardware and software design, fault tolerance, and supply strategies. We develop an integrative framework that articulates how AI-driven design tools, edge-native architectures, and embedded security practices can be orchestrated to shorten development cycles while maintaining—or improving—functional correctness, security posture, and operational resilience. The paper first situates the challenge by reviewing drivers of time-to-market pressures and causes of project failure. It then explores how predictive analytics, machine learning–assisted EDA (electronic design automation), automated scenario generation for human error assessment, and pre-silicon design-for-test feedback loops can be operationalized. Security is treated as a cross-cutting concern: the adoption of multi-factor authentication, DevSecOps pipelines with static and dynamic analysis, and automated vulnerability management are positioned as essential for modern CI/CD practices. The manuscript further articulates supply-side considerations—such as dual sourcing—to cope with disruption, and cloud-native cost and sustainability tradeoffs for deployment. Methodologically, the paper proposes a mixed-methods conceptual model that integrates formal verification, data-driven predictive monitoring, and fault-injection scenario generation to drive iterative design improvements. Results are described as a set of expected impacts and measurable indicators—reduced cycle time, earlier fault detection, improved vulnerability metrics, and better inventory/sourcing resilience—along with qualitative discussions on limitations, potential failure modes, and future research directions. The work concludes by mapping concrete research and practice agendas to achieve a balance of speed, quality, and security in product and system development. The contribution is both synthetic—bringing together disparate literatures—and prescriptive—offering an actionable, academically grounded roadmap for industry and research communities.

INTRODUCTION

The drive to reduce time-to-market for new products is a strategic imperative across sectors, from consumer electronics and automotive to software and services (Eurenius & Teräväinen, 2020). Market windows narrow as technological capability diffuses and competitors shorten their innovation cycles; at the same time, systems are becoming larger and more integrated—combining silicon, embedded firmware, AI models, cloud services, and edge devices. This increasing heterogeneity intensifies the complexity of verification, validation, and deployment. The literature documents persistent project failures attributable to misaligned expectations, inadequate risk management, technical debt, and deficient integration of security in development workflows (Hughes, Rana, & Simintiras, 2017). Reducing time-to-market, therefore, cannot be pursued as a narrow optimization of calendar days alone; it must be integrated with strategies that preserve correctness, safety, and security.

Reducing cycle time while maintaining quality requires a multi-pronged approach. First, early-stage detection of functional faults and human-error vulnerabilities can prevent expensive downstream rework (Irshad, Demirel, & Tumer, 2020). Second, automation and AI augmentation in design tools—particularly in electronic design automation (EDA) for hardware and FPGA flows—can accelerate iteration and improve design space exploration (Goswami & Bhatia, 2023). Third, edge computing with embedded intelligence enables low-latency, distributed operation and offloads central infrastructure, but introduces new testing and verification challenges (Hua et al., 2023). Fourth, security must be integrated into continuous integration and continuous deployment (CI/CD) pipelines (DevSecOps) through SAST/DAST/SCA to prevent vulnerability accumulation and reduce friction at release (Konneru, 2021). Finally, supply-side strategies like dual sourcing provide resilience against component shortages and geopolitical risks (Goel & Bhramhabhatt, 2024).

This work synthesizes these strands into a coherent framework. We start by articulating the theoretical and practical knowledge drawn from the provided references, identifying gaps where literature suggests unresolved tensions—such as the interplay between coverage-directed testing and risk of missing critical faults (Gay et al., 2015) or the limitations of purely formal verification in the face of complex, human-in-the-loop systems (Hasan & Tahar, 2015). Building from this base, we propose a methodological approach that combines formal verification techniques, AI-augmented EDA, predictive analytics for DevOps, scenario-based fault generation, and supply-chain risk mitigation to reduce time-to-market while preserving or improving system dependability and security.

METHODOLOGY

This manuscript is conceptual and prescriptive, grounded in cross-disciplinary literatures. The methodological approach proposed for practitioners and research validation consists of several integrated components: (1) Early-stage automated fault-scenario generation, (2) AI-augmented pre-silicon EDA and DFT feedback loops, (3) Formal verification integration for critical components, (4) A DevSecOps pipeline with predictive analytics and automated vulnerability and inventory optimization, (5) Edge-native validation and deployment strategies, and (6) Supply-side resilience through dual sourcing and inventory forecasting. Each element is described in detail below with prescriptive steps and theoretical rationale drawn from the references.

Early-stage automated fault-scenario generation

Irshad et al. (2020) demonstrate that automated generation of fault scenarios in early design stages can surface potential human errors and functional vulnerabilities that would otherwise only become evident during system integration or field operation. The method advocated uses model-based representations of system behavior combined with automated perturbation and scenario enumeration to explore failure modes systematically. Practically, teams should construct abstract system models—capturing user interactions, sensor/actuator mappings, and control logic—and then apply automated perturbation engines to create error-chain scenarios (e.g., miscalibrated sensors, operator input errors, timing violations). Each generated scenario must be mapped to potential mitigations, such as monitoring hooks, enhanced UI warnings, redundant checks, or formalized exception handling.

Automating scenario generation provides two advantages. First, it increases coverage of plausible but non-obvious error chains that human analysts may overlook, thereby improving early design robustness. Second, it creates a prioritized roster of tests that can be employed by AI-driven simulation tools and by human testers in later stages. This approach aligns with the need to shift left—finding defects earlier in the lifecycle to avoid expensive fixes later—and directly supports time-to-market reduction by minimizing rework cycles (Eurenius & Teräväinen, 2020). However, it is crucial to manage false positives and scenario explosion; practitioners should adopt risk-guided pruning strategies (e.g., weighting scenarios by likelihood and impact) to focus resources.

AI-augmented pre-silicon EDA and DFT feedback loops

The role of machine learning in accelerating EDA workflows and enabling smarter design exploration has been increasingly recognized (Goswami & Bhatia, 2023; Mishra et al., 2023). AI models can predict likely problematic regions in designs, suggest optimized placements/routings, and automate time-consuming verification tasks. Lulla (2025) highlights the importance of pre-silicon design-for-test (DFT) feedback loops for GPU productization, illustrating how early DFT insights can prevent costly respins. The methodology proposed integrates AI tools with DFT instrumentation to create continuous feedback: simulation and emulation runs generate telemetry that is fed to ML models, which in turn recommend modifications—such as alternative logic partitioning, different clocking strategies, or targeted formal checks.

A practical pipeline includes instrumentation hooks that collect coverage, timing, and power metrics during simulation. ML models trained on historical design datasets (including wear-out and aging data where available) can predict aging hotspots and propose mitigation via redundancy, guard-banding, or more aggressive DFT insertion (Rendon, 2024). This approach reduces the number of iterative cycles required between design and tape-out, directly shortening development time and improving first-pass yield.

Formal verification integration for critical components

Formal methods remain essential for proving functional properties in critical subsystems where empirical testing alone is insufficient (Hasan & Tahar, 2015). The proposition here is selective formalization: reserve rigorous formal verification for a bounded set of safety- or security-critical modules—such as cryptographic primitives, transaction state machines, or safety monitors—while employing lighter-weight verification for general-purpose blocks. This hybrid approach balances the high cost of formal proofs with the need to maintain high assurance for critical functionality.

The integration requires establishing formal specifications early, aligning them with system requirements, and maintaining traceability between requirements and verified artifacts. In practice, teams should adopt a gated approach: critical modules are subject to formal verification pre-integration, while non-critical modules undergo property-based testing and fuzzing. This mitigates the risk of formally verified components being invalidated by interface errors at integration time.

DevSecOps pipeline with predictive analytics and automated vulnerability and inventory optimization

Security and operational resilience must be embedded in the CI/CD pipeline rather than applied as an afterthought (Konneru, 2021; Kamaruddin & Zolkipli, 2024). The methodology advocates a DevSecOps pipeline that includes SAST (static application security testing), DAST (dynamic analysis), and SCA (software composition analysis) as automated gates, supplemented by predictive analytics that forecast periods of elevated risk or demand.

Predictive analytics—drawing on historical telemetry, incident data, and operational metrics—can forecast bug-prone releases, likely vulnerability hotspots, or supply chain disruptions (Kumar, 2019; Malik, 2025). This forecast then drives proactive measures: earlier, more intensive security scans, targeted fuzzing of suspect modules, or procurement of substitute parts. Malik (2025) emphasizes automating vulnerability management and demand forecasting in CI/CD-powered retail systems; applying analogous automation to product development pipelines reduces human overhead and shortens reaction time to faults, thereby supporting faster release cycles.

Inventory and cost optimization in cloud-native deployments and Kubernetes must also be considered (Pinnapareddy, 2025). Cost and sustainability are operational constraints that influence release timing: unexpectedly high cloud costs or sustainability targets can delay releases. Integrating cloud cost optimization tools in the pipeline, and matching deployment profiles to realistic demand forecasts, enables teams to plan releases with fewer surprises.

Edge-native validation and deployment strategies

Edge computing with AI introduces unique constraints—low latency requirements, intermittent connectivity, and constrained resources—that must be tested in representative environments (Hua et al., 2023). Validation strategies should include hardware-in-the-loop testing, network condition emulation, and deployment canaries that gradually expose new releases to subsets of devices. AI models used at the edge must be validated for robustness under distributional shifts and performance constraints. Techniques include on-device quantization-aware testing, adversarial robustness checks, and incremental model rollouts with rollback capabilities.

Edge-native deployment also requires observability tailored to distributed environments. Telemetry collection and local health monitors allow rapid triage of issues without requiring a full central rollback.

This approach reduces systemic risk and enables faster iterative updates, as issues can be localized and corrected more quickly.

Supply-side resilience and dual sourcing

Supply chain fragility can dramatically increase time-to-market if single-source components are delayed. Dual sourcing strategies—deliberate procurement from multiple suppliers for critical components—mitigate this risk (Goel & Bhramhabhatt, 2024). The methodology recommends integrating dual sourcing decisions into the product development plan early, using scenario analysis to understand cost-performance-resilience tradeoffs. Considerations include supplier qualification time, compatibility testing burden, and inventory implications. Predictive demand forecasting should inform the split between primary and secondary suppliers to balance cost and resilience.

Dual sourcing also intersects with security: suppliers must be assessed for cybersecurity posture and counterfeit risk. Integrating supplier security evaluation into the procurement workflow reduces supply-side vulnerabilities that could introduce malware or faulty components.

RESULTS

Because this manuscript is conceptual, the "results" are descriptive syntheses and projected impacts rather than empirical experimental outputs. Nevertheless, the proposed integrated approach is expected to yield measurable benefits across several axes when adopted in industrial settings. Below we describe anticipated outcomes, measurable indicators, and hypothesized magnitudes based on cross-referenced literature.

Reduced cycle time and rework

Shifting fault detection and scenario enumeration to earlier stages reduces late-cycle rework. Eurenius and Teräväinen (2020) document time-to-market reductions achieved by process changes; when combined with automated scenario generation (Irshad et al., 2020) and AI-augmented EDA (Goswami & Bhatia, 2023), organizations can expect reductions in cumulative development time primarily by cutting the number and severity of integration-stage defects. Practically, teams may observe fewer tape-outs or major software rollbacks and a higher proportion of first-pass successful verifications. While exact percentages vary by domain, analogous case studies in EDA and pre-silicon testing suggest first-pass correction rate improvements that materially shorten iteration cycles (Lulla, 2025).

Earlier detection of safety and security issues

Selective formal verification applied to critical components reduces the probability of catastrophic failure due to logic errors in those components (Hasan & Tahar, 2015). Paired with SAST/DAST/SCA gates within DevSecOps pipelines, teams can expect a lower incidence of high-severity vulnerabilities making it to production. Malik (2025) argues for automated vulnerability management as a means to reduce mean time to patch; when integrated into the development pipeline, this reduces both exposure windows and the operational burden of emergency patches.

Improved design-space exploration and product quality

AI-augmented EDA tools enable more exhaustive exploration of the design space in less time, detecting suboptimal configurations and performance bottlenecks earlier (Goswami & Bhatia, 2023). Combined with DFT feedback loops, designs can be tuned for testability without incurring substantial schedule delays (Lulla, 2025). The correlation between improved design-space coverage and downstream quality suggests a higher yield and fewer field issues.

Resilience to supply disruptions

Dual sourcing reduces the probability of production halts due to supplier failures, while predictive demand forecasting and inventory optimization reduce the risk of stockouts (Goel & Bhramhabhatt, 2024; Pinnapareddy, 2025). Implemented well, these measures reduce unplanned delays and smooth the release schedule.

DISCUSSION

The proposed integrated framework holds promise, but it encounters non-trivial limitations, tradeoffs, and practical challenges. This section discusses these in depth and outlines directions where further empirical work is needed.

Tradeoffs between speed and assurance

A central tension in accelerating time-to-market is the potential tradeoff between speed and assurance. Faster releases can, if poorly managed, increase the risk of undetected faults or security vulnerabilities. Our approach mitigates this tension through selective formal verification and automated scenario generation; nevertheless, tradeoffs remain. Formal verification is resource-intensive and may bottleneck schedules if applied too broadly. Conversely, relying solely on automated testing and ML-driven suggestions risks missing corner cases that formal methods would catch (Hasan & Tahar, 2015; Gay et al., 2015). The strategic compromise is to identify the minimal set of components warranting deep formal assurance and to augment broader system validation with scenario-based testing informed by automated generation engines.

Quality of AI models and data

AI augmentation's effectiveness depends heavily on the quality and representativeness of training data (Goswami & Bhatia, 2023). In the EDA context, diversity of prior designs, accurate labeling of failure modes, and representative telemetry are essential. New product lines or novel architectures will naturally yield distributional shifts, reducing AI model reliability. Continual retraining and monitoring of AI model performance are necessary, as is conservative operationalization—using model suggestions as augmentations rather than absolute directives.

Coverage-directed testing pitfalls

Coverage-directed test generation is seductive but dangerous if applied naively. Gay et al. (2015) warn that optimizing for coverage metrics can give a false sense of security while missing critical behavioral faults not captured by coverage metrics. To avert this, teams should combine coverage metrics with risk-based prioritization, formal properties, and scenario-driven tests that focus on high-impact behaviours. Metrics must be interpreted in context and complemented by empirical incident data.

Human factors and organizational dynamics

Integrating formal verification, AI tooling, and DevSecOps requires organizational change. Teams accustomed to siloed workflows must adopt cross-functional practices and develop expertise in new tools. Hughes et al. (2017) highlight that IS project failures often stem from poor governance and misalignment among stakeholders. Organizational buy-in, training programs, and phased adoption strategies are essential. Additionally, automated tooling can produce alerts and findings at a scale that overwhelms teams; thus, careful workflow design, prioritization strategies, and automation for triage are necessary.

Supply-chain and procurement complexities

Implementing dual sourcing has cost implications—qualifying multiple suppliers requires time and budget, and increased inventory may raise carrying costs. The optimal tradeoff depends on organization risk tolerance, market volatility, and product margins (Goel & Bhramhabhatt, 2024). Predictive analytics can assist but are not foolproof: unanticipated macro disruptions (e.g., geopolitical events) can still inflict delays. Hence, dual sourcing must be seen as part of a broader resilience strategy, not a silver bullet.

Security integration challenges

Embedding security in CI/CD pipelines is conceptually straightforward but operationally challenging. SAST and SCA can produce noisy results, and overly strict gating risks blocking normal releases, thereby increasing time-to-market rather than reducing it (Konneru, 2021). The solution involves calibrating gates, using risk tiers, and automating triage to reduce the human resource burden. Additionally, security must include supply-side assessments to prevent vulnerabilities introduced via third-party components (Kumar et al., 2016).

Future research directions

Empirical evaluation: The conceptual model proposed should be validated through empirical case studies across domains—consumer electronics, automotive, cloud-native services—to quantify time-to-market improvements, defect reductions, and cost implications.

AI model robustness: Research on robust ML techniques for EDA and predictive analytics—especially under distributional shift and sparse data regimes—is critical. Techniques for uncertainty quantification and explainability are especially relevant to enable practitioner trust.

Human–AI collaboration: Investigate optimal interaction models between engineers and AI tools—when to accept suggestions, how to interpret model confidence, and how to integrate AI-generated artifacts into formal verification workflows.

Edge validation frameworks: Designing standardized protocols and tooling for edge-native validation—covering network emulation, hardware-in-loop testing, and on-device model validation—will help operationalize the edge component of the proposed framework.

Supply-chain AI ethics and security: Research into secure supplier evaluation, provenance tracking, and anti-counterfeiting measures using cryptographic anchors or hardware PUF-based verification (Kumar et al., 2016) can reinforce supply-side trust.

CONCLUSION

Accelerating product development while preserving reliability and security is a multi-dimensional challenge that requires coordinated advances across tooling, processes, and organizational practices. This manuscript has synthesized literature from time-to-market studies, AI in EDA, formal verification, DevSecOps, edge computing, and supply resilience to propose an integrated framework. Core tenets include shifting fault detection left via automated scenario generation, applying AI to accelerate pre-silicon design and DFT feedback loops, selectively applying formal verification for critical components, embedding security into CI/CD pipelines augmented with predictive analytics, adopting edge-aware validation strategies, and employing dual sourcing and inventory optimization for supply resilience.

Implementation of this integrated approach is not trivial: it requires investment in instrumentation and tooling, organizational change to support cross-functional workflows, and continual management of AI model quality. Yet, the potential benefits—reduced rework, faster iterations, improved first-pass yield, reduced vulnerability exposure, and enhanced supply resilience—make it an attractive research and engineering agenda. Future empirical work should evaluate these propositions in real-world development environments to quantify benefits and refine methods.

REFERENCES

1. Eurenius, E., & Teräväinen, B. (2020). Reducing time to market in new product development.
2. Gay, G., Staats, M., Whalen, M., & Heimdahl, M. P. (2015). The risks of coverage-directed test case generation. IEEE Transactions on Software Engineering, 41(8), 803-819.
3. Goel, G., & Bhramhabhatt, R. (2024). Dual sourcing strategies. International Journal of Science and Research Archive, 13(2), 2155. https://doi.org/10.30574/ijsra.2024.13.2.2155
4. Goswami, P., & Bhatia, D. (2023). Application of machine learning in FPGA EDA tool development. IEEE Access, 11, 109564-109580.
5. Hasan, O., & Tahar, S. (2015). Formal verification methods. In Encyclopedia of Information Science and Technology, Third Edition (pp. 7162-7170). IGI Global Scientific Publishing.
6. Hua, H., Li, Y., Wang, T., Dong, N., Li, W., & Cao, J. (2023). Edge computing with artificial intelligence: A machine learning perspective. ACM Computing Surveys, 55(9), 1-35.
7. Hughes, D. L., Rana, N. P., & Simintiras, A. C. (2017). The changing landscape of IS project failure: An examination of the key factors. Journal of Enterprise Information Management, 30(1), 142-165.
8. Irshad, L., Demirel, H. O., & Tumer, I. Y. (2020). Automated generation of fault scenarios to assess potential human errors and functional failures in early design stages. Journal of computing and information science in engineering, 20(5), 051009.
9. Kamaruddin, N. H. C., & Zolkipli, M. F. (2024). The Role of Multi-Factor Authentication in Mitigating Cyber Threats. Borneo International Journal eISSN 2636-9826, 7(4), 35-42.
10. Kamran, S. S., Haleem, A., Bahl, S., Javaid, M., Prakash, C., & Budhhi, D. (2022). Artificial intelligence and advanced materials in the automotive industry: Potential applications and perspectives. Materials Today: Proceedings, 62, 4207-4214.
11. Karwa, K. (2023). AI-powered career coaching: Evaluating feedback tools for design students. Indian Journal of Economics & Business. https://www.ashwinanokha.com/ijeb-v22-4-2023.php

12. Karwa, K. (2024). The future of work for industrial and product designers: Preparing students for AI and automation trends. Identifying the skills and knowledge that will be critical for future-proofing design careers. International Journal of Advanced Research in Engineering and Technology, 15(5). https://iaeme.com/MasterAdmin/Journal_uploads/IJARET/VOLUME_15_ISSUE_5/IJARET_15_05_011 .pdf

13. Katsaros, K., Mavromatis, I., Antonakoglou, K., Ghosh, S., Kaleshi, D., Mahmoodi, T., ... & Simeonidou, D. (2024). AI-native multi-access future networks-the REASON architecture. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3507186

14. Konneru, N. M. K. (2021). Integrating security into CI/CD pipelines: A DevSecOps approach with SAST, DAST, and SCA tools. International Journal of Science and Research Archive. Retrieved from https://ijsra.net/content/role-notification-scheduling-improving-patient

15. Kumar, A. (2019). The convergence of predictive analytics in driving business intelligence and enhancing DevOps efficiency. International Journal of Computational Engineering and Management, 6(6), 118-142. Retrieved from https://ijcem.in/wp-content/uploads/THE-CONVERGENCE-OF-PREDICTIVE-ANALYTICS-IN-DRIVINGBUSINESS-INTELLIGENCE-AND-ENHANCING-DEVOPS-EFFICIENCY.pdf

16. Kumar, S., Satheesh, N., Mahapatra, A., Sahoo, S., & Mahapatra, K. K. (2016, December). Securing IEEE 1687 standard on-chip instrumentation access using PUF. In 2016 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS) (pp. 56-61). IEEE. https://doi.org/10.1109/iNIS.2016.024

17. Malik, G. (2025). AI-Driven Security and Inventory Optimization: Automating Vulnerability Management and Demand Forecasting in CI/CD-Powered Retail Systems. International Journal of Computational and Experimental Science and Engineering (IJCESEN). https://ijcesen.com/index.php/ijcesen/article/view/3855/1153

18. Mishra, A., Cha, J., Park, H., & Kim, S. (Eds.). (2023). Artificial intelligence and hardware accelerators. Berlin: Springer. https://link.springer.com/book/10.1007/978-3-031-22170-5

19. Lulla, K. (2025). Pre-Silicon DFT Feedback Loops: Enhancing GPU Productisation Efficiency. International Journal of Computational and Experimental Science and Engineering, 11(3). https://doi.org/10.22399/ijcesen.3778

20. Nyati, S. (2018). Transforming telematics in fleet management: Innovations in asset tracking, efficiency, and communication. International Journal of Science and Research (IJSR), 7(10), 1804-1810. Retrieved from https://www.ijsr.net/getabstract.php?paperid=SR24203184230

21. Pasham, S. D. (2020). Fault-Tolerant Distributed Computing for Real-Time Applications in Critical Systems. The Computertech, 1-29. https://www.yuktabpublisher.com/index.php/TCT/article/view/142

22. Pinnapareddy, N. R. (2025). Cloud cost optimization and sustainability in Kubernetes. Journal of Information Systems Engineering and Management. https://www.jisem-journal.com/index.php/journal/article/view/8895

23. Raju, R. K. (2017). Dynamic memory inference network for natural language inference. International Journal of Science and Research (IJSR), 6(2). https://www.ijsr.net/archive/v6i2/SR24926091431.pdf

24. Rendon, M. J. (2024). 12nm Finfet aging characterization through wear-out sensor design (Doctoral dissertation, University of British Columbia). http://hdl.handle.net/2429/89220.