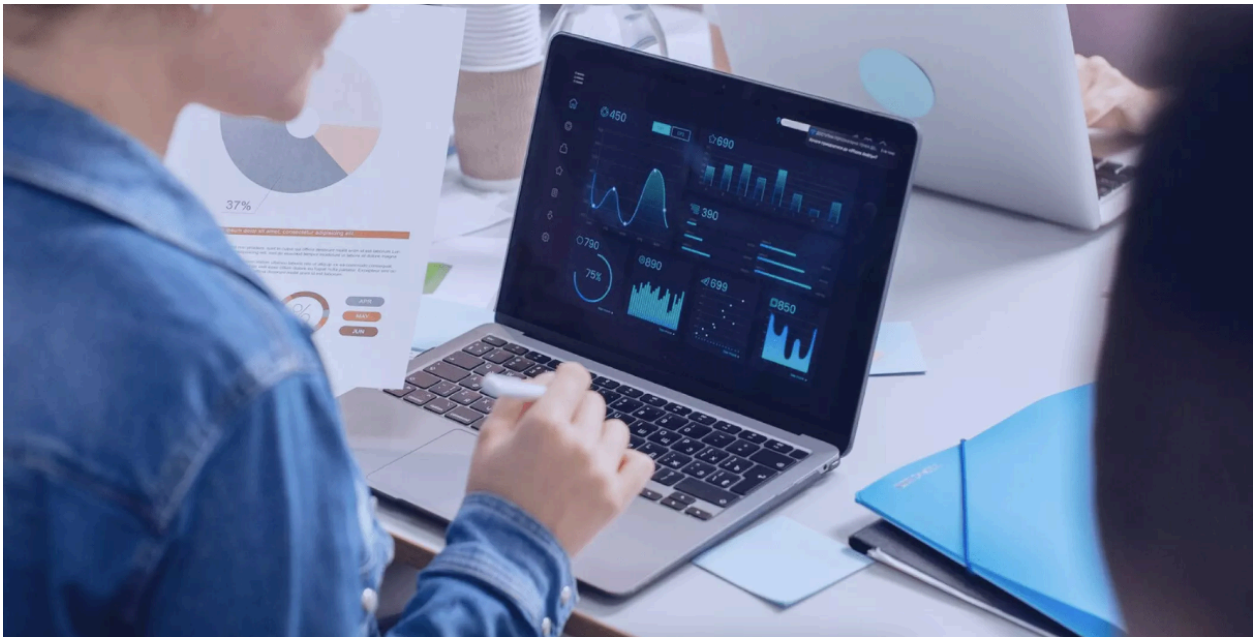
How to Minimize Risks in Software Development?

Introduction

Software projects are an invaluable driver of business transformation, but also one of the riskiest investments for many organizations. This year, the software development market continued to grow rapidly, but so did the types and scale of risks teams had to manage: delivery disruptions, technical debt, supply chain disruptions, and rapidly evolving security threats (now exacerbated by the implementation of AI). This paper examines the most common risks in software development, provides reliable data for 2025, offers practical methods and key performance indicators (KPIs) for risk mitigation, and provides forecasts and recommendations for 2026–2027.

Quick Takeaways

Only about one-third of IT projects delivered in recent CHAOS analyses meet the classic definition of “successful” (on time, on budget, full scope).
Security incidents remain costly but show early signs of improvement in containment: average breach costs in 2025 fell to about $4.44M, helped by faster detection/containment and wider AI use in security. Yet AI adds governance gaps that increase risk if unmanaged.
DevOps and internal developer platforms plus disciplined metrics (DORA-style) remain among the strongest predictors of lower risk and faster recovery – and the 2025 DORA research examines how AI is reshaping delivery.
Forecast: by 2026–2027, AI-native development platforms and LLM-driven features will be mainstream across many teams (Gartner predicts majority adoption trends through 2027), bringing both productivity gain and new governance/security requirements.



The Risk Landscape: What Organizations Face in 2025

Delivery & project risk

Project outcome studies continue to demonstrate a high rate of problematic projects. Recent CHAOS-style industry surveys indicate that only about 30% of projects are fully successful; the rest either suffer from problems or fail. Large-scale efforts and unclear decision-making processes lead to significantly worse results than smaller, focused initiatives.
Why this matters: Failed or problematic projects lead to direct financial losses, delays in time-to-market, and technical debt, which exacerbates future risks.

Security & data risk

Major breaches in 2025 (and repeated supply chain incidents) clearly demonstrate that security is more than just an add-on. The IBM/Ponemon report, "The Cost of a Data Breach 2025," found that the average cost of a breach globally has decreased to approximately $4.44 million (a decrease from the previous year due to faster detection and containment), while emphasizing that unmanaged AI implementation dramatically increases the risk of a breach. At the same time, high-profile breaches (in the supply chain or by insiders) continue to lead to mass disclosures and business disruptions.

Supply-chain and third-party risk

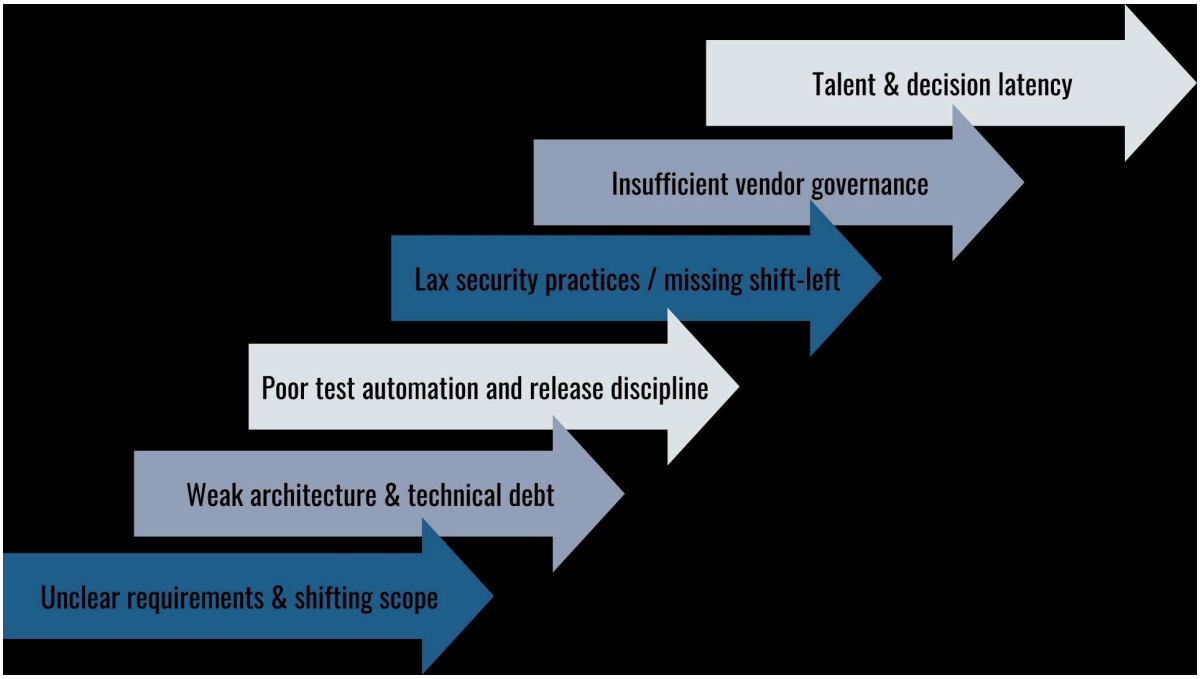
Supply chain breaches are on the rise: multiple recent surveys and 2025 incident data show that over 60% of organizations reported supply chain incidents in the previous year in some regions/segments, making supplier risk a board-level concern. Attackers are increasingly using smaller suppliers as intermediaries.

People, process and AI-era risk

AI tools improve delivery productivity and facilitate programming, but a 2025 study (DORA/Google Cloud, "The State of AI-Powered Development in 2025") found that AI also exacerbates team dysfunction if it lacks clear boundaries. Organizations that implemented AI without governance faced increased vulnerability to configuration errors, data breaches, and ineffective decision-making.

Root Causes: Why Risks Keep Appearing

- Unclear requirements & shifting scope – scope creep without strong product governance remains the top root cause of delivery failure.
- Weak architecture & technical debt – shortcuts early on increase long-term fragility.
- Poor test automation and release discipline – infrequent releases and manual processes cause long MTTR and slow feedback loops.
- Lax security practices / missing shift-left – security treated as post-release adds detection and remediation costs.
- Insufficient vendor governance – no SBOM, poor dependency scanning, or inadequate SLAs.
- Talent & decision latency – slow or poor decisions by leadership correlate with worse outcomes. (The latest CHAOS analysis highlights decision latency as a key differentiator.)



Frameworks and Patterns to Minimize Risks

Below are the most practical, evidence-backed frameworks and patterns teams should adopt.

Measure the right things – DORA + extended metrics

Use the four DORA metrics (change lead time, deployment rate, change failure rate, and mean time to repair) as a baseline. Research shows that in 2025, teams that track and act on delivery

metrics will be significantly more effective. DORA is exploring how to use AI and the additional metric of "speed of rework" to reflect today's realities. Add business-focused KPIs (customer impact, decision latency) to tie engineering metrics to results.
Actionable KPIs:

- Change failure rate < 15% (best teams).
- Mean time to repair (MTTR) in minutes/hours for critical services.
- Decision latency (the time from the submission of a blocking request to management's decision) is below a defined threshold.

Continuous everything: CI/CD, trunk-based development, and automated testing

Automate your pipeline: test suites, linting, security scanning, and deployment. Frequent, small deployments reduce the attack surface and identify integration issues earlier.
Specific controls:

- Required pipeline stages: unit tests, integration smoke tests, dependency scanning, SCA (software composition analysis).
- Canary releases and releases with feature flags.
- Rollback and automated remediation scripts.

Shift-left security and DevSecOps

Integrate security early: threat modeling during design, SAST, DAST in CI, covert scanning, and automated IaC security checks. By 2025, organizations that integrate AI-powered detection and management will reduce their mean time to containment and their cost. But remember: AI requires access control and audit trails.

Supply-chain hygiene: SBOMs and vendor due diligence

Require a software specification document (SBOM) for all third-party components; conduct regular vulnerability scanning; and be sure to include service level agreements (SLAs) and incident response obligations in contracts. Consider critical vendors as an extension of your risk profile.

Architectural practices that reduce fragility

- Microservices with clear contracts and backpressure patterns.
- Observability-focused design: logs, metrics, traces, structured intervals.
- Resilience patterns: circuit breakers, firewalls, graceful degradation.
- Regular architecture reviews and dependency sprints.

Platform engineering and developer experience

Internal development platforms (IDPs) and well-designed developer workflows reduce variability and minimize the risk of adaptation. Reports as of 2025 show that teams using IDPs achieve measurable productivity gains and fewer configuration errors.

Governance & decision processes

Create a RACI for product decision-making, ensure sprint control, align roadmaps with business outcomes, and reduce decision delays by empowering cross-functional teams. CHAOS analysis shows that fast and informed decisions are closely linked to success.

Security-Specific Playbook

1. Classify data and apply least privilege.
2. Enforce authenticated, token-based access (rotate keys).
3. Implement immutable backups and test recovery (3-2-1 rule). Recent 2025 reporting shows many orgs still miss this control, increasing ransomware risk.
4. Deploy SCA/SAST/DAST in pipelines and fail builds on critical vulnerabilities.
5. Maintain SBOM and continuous dependency monitoring.
6. Add runtime protection for critical services (WAF, EDR, EPP).
7. Institute AI governance: model access controls, data provenance, logging, and red-team tests for LLM behavior. IBM’s 2025 report signals that ungoverned AI systems are more likely to be breached.

Organizational Practices & People Risks

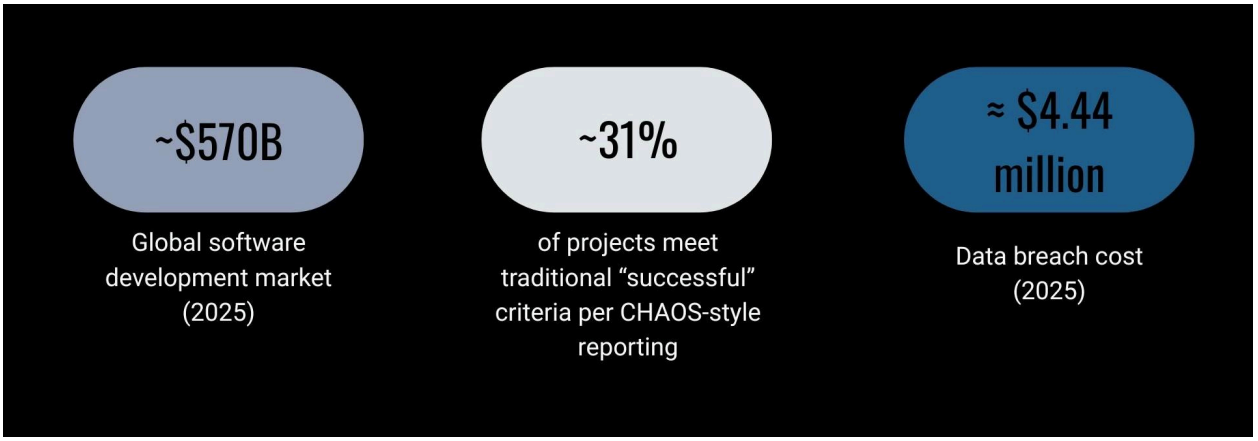
- Hire for T-shaped teams & cross-functional ownershipю Developers, QA, SREs, and security engineers should share responsibility for delivery and production health. Cross-training reduces single-person dependency risk.
- Reduce bus factor with documentation & pair programming. Encourage pair programming, code reviews, and living runbooks. Maintain an up-to-date runbook for critical incidents.
- Invest in continuous learning. Sponsor training for secure coding, platform usage, and AI governance. StackOverflow and industry surveys from 2025 show that developer tool familiarity (including AI tools) materially affects team velocity and quality.

Tools & Automation

- CI/CD: Jenkins X, GitHub Actions, GitLab CI, Tekton.
- SCA/SAST/DAST: Snyk, SonarQube, Checkmarx, OWASP ZAP.
- Observability: OpenTelemetry + Prometheus + Grafana + traces.
- SBOM / Dependency: CycloneDX, SPDX, Dependabot.
- Platform & developer experience: Internal developer platforms built on Kubernetes, Backstage.
- AI-aware governance tools: Model access logging, prompt registries, and data lineage tools (emerging market in 2025). (Choose tools that support audit trails and RBAC.)

2025 Industry Snapshot: Key Numbers

- Global software development market (2025): ~$570B.
- Project success rates: only ~31% of projects meet traditional “successful” criteria per CHAOS-style reporting; many are challenged or fail. Small projects show much higher success.
- Data breach cost (2025): global average ≈ $4.44 million, down vs prior year due to faster containment aided by AI. But US-level costs remain significantly higher in many cases. The AI governance gap was repeatedly highlighted as a major risk.
- Dev/AI trends: 2025 DORA/State of AI-Assisted Development indicates thousands of practitioners reporting early productivity gains – but also new failure modes requiring metric evolution.



Forecasts & What to Expect in 2026–2027

AI-native development & platform acceleration

Gartner and other analysts forecast rapid adoption of AI-native development platforms. By 2027, a majority of engineering teams are expected to be actively building LLM-driven features or leveraging AI agents in development, shifting many workflows to AI-assisted patterns. This increases velocity but also creates new governance/assurance needs (model safety, data leakage, prompt security).
Implication: Expect faster delivery cycles, but plan governance and observability for AI artifacts (models, prompts, datasets).

Security & breach economics

If AI is used responsibly with strong governance, average breach containment times and costs may continue to improve. But ungoverned AI can increase incident risk and exposure. Organizations that do not implement AI governance will likely see disproportionate cost increases due to misconfiguration and data misuse.

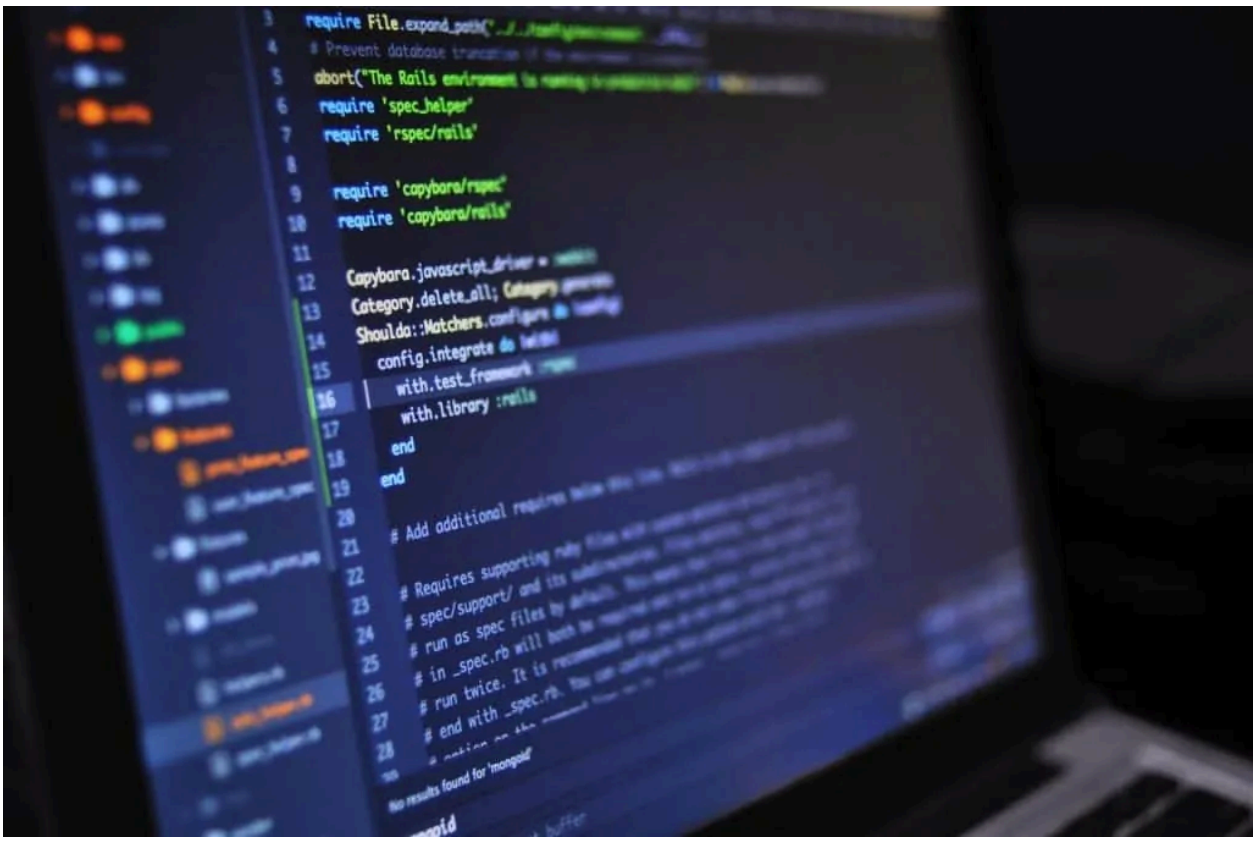
Supply-chain focus becomes mandatory

Regulators and customers will increasingly demand SBOMs, stronger vendor audits, and contractual incident commitments. Treating vendors as first-class risk partners will be standard practice by 2027.

Recommendations for Companies

- Make risk visible to the board. Present a concise risk dashboard: delivery KPIs, security posture, and top vendor exposures.
- Adopt continuous delivery and test automation – smaller, frequent releases reduce risk significantly.
- Shift security left and include security OKRs in engineering goals.
- Invest in observability & runbooks – it pays off in faster incident response and learning loops.
- Treat AI as a first-class risk category – enforce model access control, data governance, and logging.

- Enforce SBOMs and vendor SLAs for critical dependencies.
- Measure decision latency and remove blockers from approval workflows – faster decisions equal fewer prolonged risks.



Short Case Examples

Supply-chain and insider risk: Several 2025 incidents showed how an ex-employee or minor vendor can create mass exposure (e.g., major e-commerce and enterprise incidents reported in 2025). These underscore the need for strict account deprovisioning, least privilege, and vendor controls.
AI governance gap: IBM's 2025 analysis found that organizations adopting AI rapidly without governance were more frequently implicated in costly incidents – demonstrating the need for prompt registries, logging, and model access policy.

Checklist: Quick Controls You Can Apply Today

- Enforce CI pipeline security gates (SCA, SAST, secret-scan).
- Require SBOMs and run continuous dependency monitoring.
- Implement feature flags + canary releases for safer rollouts.
- Keep backups immutable and test recovery quarterly.
- Measure DORA metrics and track MTTR.
- Create AI governance policies and an access registry.
- Run incident tabletop exercises twice a year.

Conclusion

Risk in software development is real and multifaceted – but it is engineering work: measurable, testable, and reducible. The combination of continuous delivery, observability, platform-driven standardization, and mature security (including AI governance) is the proven path to lower risk and better outcomes. 2025 showed progress (faster containment, strong platform benefits), but also revealed new gaps (AI governance, supply-chain hygiene) that organizations must close to succeed in 2026–2027.