Acronis

WHITE PAPER

Why the secure software development life cycle (SSDLC) must be a key criterion in supply chain evaluations

Table of contents

Disclaimer

This white paper is provided for informational purposes only. It reflects lessons learned from Acronis' own ISO/IEC 27001 and IEC 62443-4-1 certification journey, combined with current industry practices, publicly available data and professional insights at the time of writing. It does not constitute legal advice, regulatory guidance or a contractual commitment by the authors or their organizations.

While references are made to international standards (e.g., ISO/IEC, IEC, CSA) and regulatory frameworks (e.g., NIS 2, DORA, the EU Cyber Resilience Act), these are presented for contextual purposes only. Readers remain responsible for interpreting and applying such frameworks in line with their specific business, legal and regulatory environments.

The checklist evaluation tools provided in the appendix are intended as supporting aids to assist in supply chain risk assessments. They do not replace formal audits, certifications, or due diligence processes required by law or industry standards.

Neither the author nor his organizations shall be held liable for any decisions, outcomes or damages resulting from the use of this white paper.

Introduction

The last five years have demonstrated that supply chain risk is one of the most critical vectors in cybersecurity. Notorious supply-chain breaches involving SolarWinds, Log4j and MOVEit demonstrate a new reality: that targeting software suppliers is often the most efficient way for attackers to compromise entire industries at scale, affecting not only IT environments but also organizations operating OT infrastructure. These incidents illustrate how a single compromised software update or dependency can ripple into industrial operations and critical services.

According to ENISA's Good Practices for Supply Chain Cybersecurity,[1] supply chain compromises accounted for 17% of intrusions in 2021, up from less than 1% in 2020, and between 39% and 62% of organizations reported experiencing a third-party cyber incident. This trend has continued into 2025, with the Acronis Cyberthreats Report H1 2025 noting that unpatched vulnerabilities in MSP and RMM software tools remain a primary attack vector exploited by ransomware groups, with consequences that extend far beyond the initial victim.

At the same time, research from SANS[2] shows that while OT security is increasingly recognized as critical, budget allocation and accountability often remain fragmented, leaving gaps between IT and OT risk management that adversaries can exploit.

These trends underscore a simple reality: the security of the supply chain ultimately depends on the practices embedded in the software development lifecycle. Without assurance of an SSDLC, organizations inherit risks that cannot be mitigated by contractual clauses or perimeter defenses alone.

Every software product, whether delivered as an on-premises installation or as a cloud service, is developed by someone. Traditional supplier due diligence typically emphasizes financial health, service-level commitments or infrastructure security. While necessary, these measures often overlook the software development process itself — the point at which vulnerabilities are most likely to be introduced and later weaponized.

This omission creates exposures across all environments — whether IT or OT, cloud or on-premises:

On-premises IT and OT systems: Insecure code shipped into enterprise applications or industrial controllers becomes a permanent part of the environment, where patching delays or incomplete updates can prolong exposure for months or even years.

Cloud IT services and connected OT platforms: Vulnerabilities in a provider's codebase or continuous integration and continuous delivery / deployment (CI/CD) pipeline can silently propagate to thousands of tenants or devices, with little visibility or control for customers.

Hybrid realities: While OT has traditionally been on premises, more industrial systems are now cloud enabled for monitoring and remote management. Conversely, IT workloads still often run on premises in critical sectors for security, compliance or latency reasons. In both cases, software supply chain weaknesses traverse these boundaries, making the development lifecycle a decisive factor in resilience.

[1] European Union Agency for Cybersecurity (ENISA). "Good Practices for Supply Chain Cybersecurity." June 13, 2023. https://www.enisa.europa.eu/publications/good-practices-for-supply-chain-cybersecurity

[2] [SANS Institute] Parsons, Dean. 2025 ICS/OT Cybersecurity Budget: Spending Trends, Challenges, and the Future. March 3, 2025. https://www.sans.org/white-papers/2025-ics-ot-cybersecurity-budget-spending-trends-challenges-future

Past incidents such as the Industroyer malware attack, which disrupted Ukraine's electrical grid by exploiting industrial control systems, underscore how weaknesses in OT can translate into physical-world impact. At the same time, IT-centric attacks like the SolarWinds breach demonstrate how compromised updates in trusted cloud or enterprise software can cascade across entire ecosystems.

In all scenarios — IT or OT, cloud or on premises — the root cause lies in the absence of mature secure software development practices. Without assurance that suppliers have adopted an SSDLC, organizations inherit risks that cannot be mitigated by network defenses or contractual clauses alone.

Global policymakers have recognized this gap. Frameworks such as ISO/IEC 27001:2022, IEC 62443, CSA CCMv4, the EU NIS 2 Directive, DORA and the Cyber Resilience Act explicitly highlight the importance of secure software engineering. These frameworks collectively set the expectation that buyers should evaluate a supplier's SSDLC maturity as part of procurement and ongoing risk management processes.

This white paper builds on these insights to demonstrate why SSDLC must be treated as a baseline trust condition in supply chain assurance and provides practical tools — including a supplier SSDLC maturity checklist — to help organizations integrate this evaluation into their due diligence and audit processes.

The risk of the software supply chain

Supply chain compromises are not isolated incidents — they are systemic risks born from the interconnected nature of modern software ecosystems. Organizations today rely on a complex web of suppliers: direct software vendors, cloud platforms, managed service providers and open-source projects. Each of these actors introduces not only value but also potential vulnerabilities that may remain invisible until exploited.

Beyond the first breach

When attackers compromise a supplier, the real target is often not the vendor itself but its downstream customers. This creates a multiplier effect:

- A flaw in a single codebase can spread across hundreds or thousands of organizations.
- Exploits can propagate silently through trusted update channels, making early detection difficult.
- The impact is not limited to technical disruption: contractual liability, loss of customer trust and regulatory consequences amplify the damage.

This asymmetry is what makes supply chain incidents distinct from traditional breaches: an attacker invests effort once, but the return is widespread compromise. We have seen this scenario play out in both IT and OT domains. In 2023, the 3CX breach showed how attackers could compromise a trusted VoIP provider's software update mechanism, silently distributing malware to thousands of enterprises worldwide.

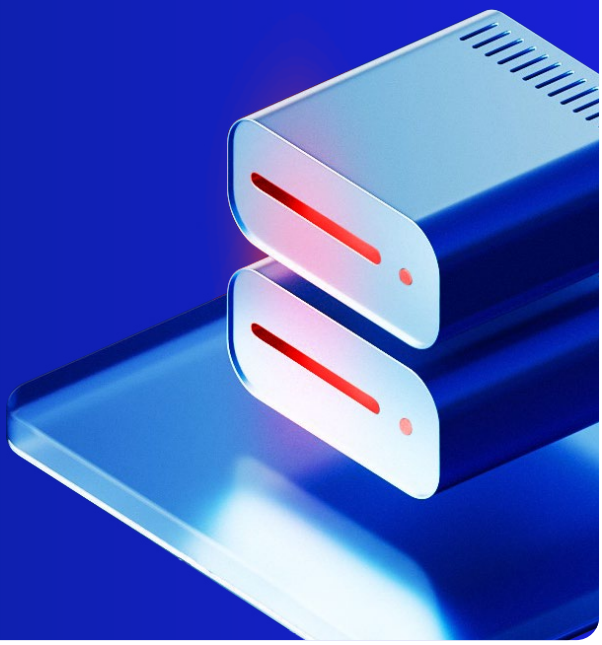
Around the same time, the discovery of the Pipedream toolkit revealed that adversaries were developing similarly sophisticated capabilities against industrial systems, with modular code designed to exploit programmable logic controllers (PLCs) and other OT components. Though different in their targets, both cases demonstrate how weaknesses in development and release practices can ripple downstream, turning a single supplier compromise into a systemic risk.

The hidden weak point: Development practices

Security controls at runtime — including firewalls, endpoint detection and response (EDR) and vulnerability scanning — can reduce exposure, but they cannot retroactively fix insecure code. If vulnerabilities are introduced during design, coding or integration, customers are forced into a reactive position, waiting for the vendor to patch. This structural dependency means that the quality of a vendor's development lifecycle directly shapes the customer's risk surface.

Keystone

Supply chain risk is not about whether a product runs in your data center or in the cloud. It is about the invisible assurance — or lack thereof — in the development processes behind it. This is why evaluating SSDLC maturity must become a baseline element of supplier risk management.



Two recurring weaknesses stand out:

- **Insecure dependencies:** Modern applications are built on layers of open-source and third-party code. Without strict governance, compromised packages (e.g., poisoned libraries, typosquatting attacks) can be introduced unnoticed.
- **Inadequate update assurance:** Patch processes may exist, but without a secure release pipeline (code signing, integrity validation, segregation of environments), attackers can hijack distribution channels.

Different models, same root cause

Whether software runs on premises, in the cloud or is embedded in OT equipment, the delivery model changes the way that supply-chain risks eventually manifest, but the root cause remains the same: insecure development and release practices.

- **On-premises IT and OT:** Insecure code becomes part of the customer's ecosystem, where patching delays can extend exposure.
- **Cloud IT services and connected OT platforms:** A flaw in the provider's code or CI/CD pipeline can propagate instantly across thousands of tenants or devices.
- **Hybrid realities:** OT systems increasingly rely on cloud services for monitoring and remote management, while IT workloads often remain on premises for security, compliance or performance reasons. In both cases, software supply chain weaknesses traverse these boundaries, making the development lifecycle the decisive factor in resilience.

Why this matters for supply chain evaluation

Traditional due diligence: Reviewing certifications, service-level agreements and infrastructure controls cannot uncover the true quality of a supplier's software engineering practices.

The rise of as-a-service consumption models and complex software stacks has made procurement teams de facto risk managers. When selecting suppliers, organizations no longer evaluate just what a product does, but how it is built and maintained. Without this lens, even highly regulated industries can inadvertently import systemic vulnerabilities.

Table 1: On-premises vs. on-cloud SSDLC risk exposure

Aspect	On-premises software	Cloud / SaaS services
Deployment model	Installed and maintained by the customer.	Hosted and operated by the provider.
Patch responsibility	Customer responsible for timely updates and patching.	Provider responsible for patching and securing runtime.
Risk amplification	A vulnerable product compromises one customer environment.	A flaw in a multitenant service can cascade across thousands of tenants.
Visibility	Customer sees runtime behavior but rarely the vendor's development practices.	Customers have almost no visibility into CI/CD pipelines or secure engineering.
Trust dependency	Relies on vendor for secure code at release and disciplined patching by customer.	Relies entirely on provider's SSDLC maturity and operational security.

This analysis of supply chain risks highlights a recurring theme: the origin of most vulnerabilities lies not in the runtime environment, but in the way software is conceived, built and released. To move from reactive defense to proactive assurance, organizations need to shift the focus of supplier evaluations toward the development lifecycle itself. This is where the concept of the SSDLC becomes central.

What SSDLC is and why it matters

Compliance authorities, cybersecurity standards developers and their experienced counterparts in business governance, risk and compliance organizations now understand that defensive technologies can detect the symptoms of and may contain the effects of incidents enabled by software insecurities but cannot address their root causes. That is why the SSDLC is not just a theoretical model. It is a mandatory practice for those who produce and maintain software, and the baseline condition for trust in a supplier. If a vendor cannot show how they embed security into their development process, then every product they deliver carries invisible risks that their customers will eventually inherit.

What SSDLC means in practice

Someone could genuinely think that SSDLC is just another development methodology — something new that competes with Agile, DevOps or Waterfall, or something you must apply on top of what teams already do. In reality, it is nothing like that. SSDLC is simply the discipline of embedding security into the development practices that already exist. Security is not a checklist bolted on at the end, but a requirement that stays with the product from its very first design sketch to its last maintenance patch.

At its core, SSDLC is about "shifting left" — identifying and mitigating risks as early as possible, when remediation is cheapest and most effective. A vulnerability caught during requirements analysis may take hours to address; the same flaw discovered post release could entail a costly, weeks-long response involving emergency patches, customer notifications and regulatory reporting.

In practice, this means adding the right checks at the right time:

Requirements: Capture security and compliance expectations upfront. Many software supply-chain breaches can often be traced to a development process in which security was assumed rather than written down, leading to expensive, painful surprises later.

Design: Apply threat modelling and secure architecture. This is the stage most teams are tempted to skip, yet it is where you decide if the product will stand on solid ground or fragile shortcuts.

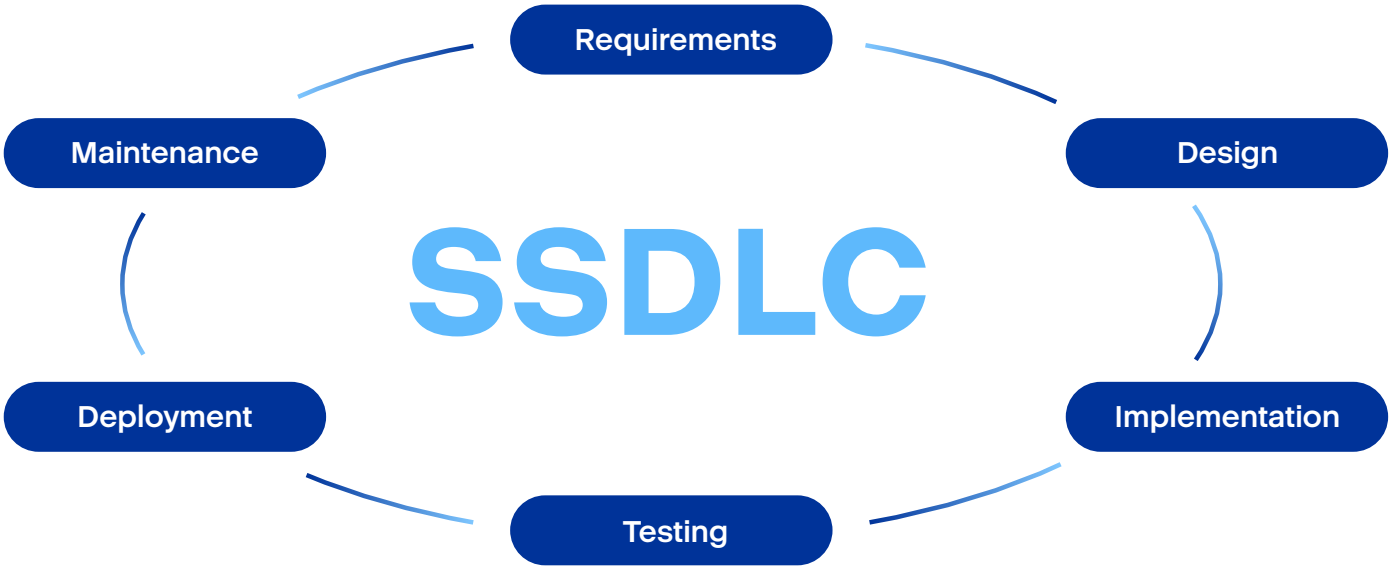
Implementation: Follow secure coding standards, do peer reviews and manage dependencies carefully. Modern software is built on layers of external libraries — if you do not govern them, you are letting attackers choose what goes into your codebase.

Testing: Use static and dynamic analysis, penetration testing and, for critical systems, fuzzing. Best practices suggest that this is what separates "checking the box" from finding weaknesses before they matter.

Deployment: Secure the build and release pipeline. If you do not lock this down with signing and segregation, attackers will not even bother with your code — they will go straight for your build system.

Maintenance: Treat patches and updates as controlled releases. Many costly supply-chain attacks in recent headlines can be traced to exploitation of the update channel itself because it was not protected.

SSDLC does not mean slowing development down or flooding teams with paperwork. It means making security part of the workflow so that vulnerabilities never get the chance to become systemic in the first place.



Why SSDLC matters for the supply chain

When evaluating supply chain risks, it becomes evident that the development process itself is the critical point of leverage. In today's supply chains, that blind spot is dangerous. Whether you are procuring a SaaS service, an enterprise application or an embedded component, what you are really buying is the vendor's development process. If that process is weak, the supplier's vulnerabilities will be silently transferred into your environment.

An SSDLC shifts this dynamic. Instead of leaving customers dependent on reactive fixes, it ensures that suppliers are systematically reducing the likelihood of vulnerabilities entering their products in the first place. For on-premises products, this means fewer flaws are shipped into customer environments, reducing the burden of patching and emergency remediation.

For cloud and SaaS services, where customers have almost no visibility or control over patching, SSDLC is the foundation of trust: if the provider does not apply secure design, testing, and release practices, customers are exposed by default.

This is why SSDLC adoption is a decisive factor in supply chain assurance. It changes the focus from what the product does to how it is engineered, from a promise to fix issues later into a demonstrable ability to prevent them earlier. A product with impressive features but insecure development practices is a liability, not an asset. Conversely, a vendor who can demonstrate SSDLC maturity provides assurance that vulnerabilities are being addressed before they ever reach production.

Procurement teams are no longer just buying functionality — they are buying the security culture of the supplier. SSDLC is the only way to make that culture visible, documentable and measurable.

Alignment with standards and regulations

Every modern cybersecurity framework now underlines two combined aspects: the importance of securing the supply chain and ensuring that organizations which develop software follow a structured SSDLC. This dual recognition reflects the systemic nature of the risk — vulnerabilities introduced during development can cascade across entire ecosystems, regardless of whether the product is delivered as cloud, on-premises or embedded in industrial equipment.

Keystone

SSDLC is not simply a technical process. It is a compliance anchor and a trust signal across every modern framework, connecting supplier assurance with regulatory obligations.

- ISO/IEC 27001:2022 strengthens general security management requirements for supplier assurance and secure development.
- IEC 62443-4-1 defines explicit SSDLC requirements for industrial and OT suppliers.
- Cloud-specific frameworks such as CSA CCM, ISO/IEC 27017 and ISO/IEC 27018 extend the same principle into service-based environments.
- The U.S.'s CMMC and the EU's NIS 2, DORA and CRA all introduce regulatory obligations for secure-by-design development and supply chain assurance.

The convergence is clear: Regulators, standardization bodies and industry frameworks all point to the same message. Supply chain risk cannot be reduced to contractual controls or certifications alone. It requires evidence that software is being developed securely, in a repeatable and auditable way.

Business value

The business value of SSDLC evaluation in the supply chain extends far beyond compliance. The business impact of SSDLC maturity can be seen across many companies: the ones that adopt it early reduce risks and avoid costly failures, while the ones that ignore it pay heavily later. The consequences are not theoretical. A supplier without a secure development process can become the vector of an attack on the final customer, damaging trust in the entire chain. There are documented cases in which a single compromised update led to loss of availability for critical systems, halting production for days. In other situations, weaknesses in development practices turned into brand and reputational damage that cost far more than the technical fix itself. The financial impact can include regulatory fines, contractual penalties, equity losses in publicly traded companies and the loss of strategic customers.

So, the value is clear:

- **Cost efficiency:** Fixing a flaw during design or coding incurs a fraction of the cost of fixing it after release or, worse, after a breach.
- **Procurement advantage:** SSDLC maturity has become a differentiator in supplier evaluations. Lack of evidence can exclude a vendor from consideration, while demonstrable maturity builds credibility.
- **Operational resilience:** In both IT and OT environments, secure-by-design software and validated updates prevent downtime and protect safety. In OT, this is not just about data — it is about keeping production running and, in some sectors, keeping people safe.
- **Reputational and financial protection:** Avoiding the role of "weakest link" in a supply chain preserves brand trust as well as partner and customer confidence and prevents costly penalties or contract losses.

For those reasons, SSDLC is not optional and not simply overhead. It is the baseline trust condition for modern supply chains. Without it, organizations are forced to defend products that were never built to be secure in the first place.

Criteria to evaluate a supplier's SSDLC

Explaining SSDLC in theory is one thing; verifying it in practice is another. In supply chain assurance, that difference is critical. Evaluating a supplier's SSDLC maturity means going beyond generic questionnaires or paper-based checklists. What matters is obtaining evidence-based assurance that security is consistently embedded into development practices.

This chapter outlines the criteria that procurement and audit teams can apply to turn secure development from a broad principle into a set of concrete, verifiable questions that reveal the true maturity of a supplier's software security practices.

Core evaluation dimensions

From the most relevant international standards, it is possible to extract a six-dimension approach for evaluating how SSDLC is implemented in practice. Each one corresponds to a critical assurance objective:

- Governance and policy ensure that secure development is not left to individual discretion but is managed strategically, with oversight and accountability.
- Risk management and design assurance prevent vulnerabilities at the earliest stage, embedding security considerations before code is written.
- Implementation practices reflect the supplier's discipline in coding, dependency management and developer enablement.
- Verification and validation demonstrate that security requirements are tested and independently confirmed before release.
- Release and deployment security ensure that the integrity of software is protected as it moves into production.
- Post-release maintenance and monitoring prove that security continues after deployment, through vulnerability handling, disclosure and monitoring.

Together, these dimensions give procurement and audit teams a structured way to assess supplier's SSDLC. Appendix A provides a detailed checklist that translates each dimension into concrete, verifiable questions for use in supplier evaluations.

Evidence to request

While interviews and questionnaires offer insight, evidence is essential to validate SSDLC maturity. Certifications are often the most valuable form of proof, especially when issued by a well-recognized and independent third party. They demonstrate that a supplier has undergone rigorous external validation and that its processes are not simply self-attested.

However, certifications must be interpreted carefully. The scope of certification is critical: in some cases, suppliers present certificates that cover only a narrow part of their operations, leaving the actual software development lifecycle outside the assessment. For this reason, organizations should not only request the certificate itself but also the auditor's report to understand what was actually evaluated.

Where certification scope is limited, customers should request complementary evidence to build a complete picture of SSDLC maturity. Useful examples include summaries of penetration tests, software bill of materials (SBOM) management records or training logs that demonstrate developer enablement.

When it comes to formal certifications, each standard plays a different role:

- ISO/IEC 27001 provides a general, organization-wide framework for information security management. It is valuable because it shows that a supplier's security is not ad hoc but governed systematically through a management system.

- IEC 62443-4-1 is part of the well-known IEC 62443 family for OT security. It focuses specifically on secure product development lifecycle requirements, and just as important, it can be adopted not only within broader OT certifications but also by companies who wish to certify only their SSDLC practices.
- Cloud service providers should complement the above with frameworks such as the Cloud Security Alliance Cloud Controls Matrix (CCMv4) or ISO/IEC 27017, which extend assurance into the SaaS and cloud domain.

Together, these certifications provide both the broad governance perspective (ISO/IEC 27001) and the specific development assurance (IEC 62443-4-1 and cloud extensions) that organizations should look for when evaluating supply chain partners.

Avoiding common pitfalls

Even when organizations request evidence, evaluations can fall short if they focus on the wrong signals. Common pitfalls include:

- **Overvaluing certificates without context.** A certificate is useful, but only when its scope is clear and supported by the auditor's report. Too often, limited-scope certifications are presented as full assurance.
- **Relying only on runtime security.** Many evaluations concentrate on infrastructure and operations but neglect the lifecycle stage where most vulnerabilities are introduced: development.
- **Ignoring update mechanisms.** A product may launch securely, but weak patching and update processes can create long-term exposure for customers.
- **Assuming OT systems are "offline."** Industrial environments historically relied on network air gapping for security, but increasingly are connected to IT and cloud services. A compromised update can ripple into production lines or critical systems.

Avoiding these pitfalls will require continued evaluation focus on the entire lifecycle. Certificates, questionnaires and infrastructure controls matter, but they only provide real assurance when combined with direct evidence that secure practices are applied consistently across governance, design, implementation, testing, release and maintenance.

SSDLC a baseline for trust

The analysis in this white paper leads to a clear conclusion: Supply chain risk is inseparable from the quality of software development practices. Whether software is delivered on premises or consumed as a cloud service, designed for IT environments or deployed in OT environments, its security depends on the maturity of the processes by which it is conceived, built, tested, released and maintained.

An SSDLC is not an optional enhancement — it is the baseline condition for trust in today's interconnected ecosystem. Evaluating SSDLC maturity in suppliers should therefore be considered a fundamental part of vendor risk management, procurement and audit.

Recommendations for organizations

To integrate SSDLC into supply chain assurance, organizations should focus on five practical priorities:

- **Embed SSDLC criteria into procurement.** Include SSDLC requirements in RFPs, contracts and vendor onboarding processes so suppliers know from the start that secure development is expected.
- **Request structured evidence.** Use the evaluation checklist in Appendix A and ask for certification scopes, auditor reports, SBOM records and testing results as part of due diligence.
- **Align with recognized standards.** Encourage suppliers to adopt relevant certifications such as IEC 62443-4-1 for product vendors and ISO/IEC 27017 or CSA CCMv4 for cloud providers, while recognizing that ISO/IEC 27001 remains the foundation for information security governance.
- **Adopt a maturity-based approach.** Move beyond binary "yes / no" questionnaires; evaluate suppliers along a continuum from ad hoc practices to fully optimized SSDLC maturity.
- **Monitor continuously.** Treat SSDLC assurance as an ongoing process that is integrated into vendor management, contract renewals and enterprise risk programs.

Acronis as a trusted supply chain partner

At Acronis, SSDLC is not a theoretical concept, but a practical discipline applied across all products, including Acronis Cyber Protect Cloud for MSPs, Acronis Cyber Protect for businesses, and Acronis Cyber Protect for OT, in all cases extending to solutions deployed both on premises and in the cloud.

This commitment is validated by independent certifications, including ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, CSA STAR Level 2, and IEC 62443-4-1. Collectively, these certifications demonstrate that Acronis products are engineered with security at their core and assessed against globally recognized standards.

By implementing SSDLC consistently and proving it through certification, Acronis aims to act not only as a technology provider but as a reliable partner in the software supply chain. This approach enables customers and partners to operate with confidence that their cyber resilience is supported by secure development practices and continuous assurance.

About Acronis

Acronis is a global cyber protection company that provides natively integrated cybersecurity, data protection, and endpoint management for managed service providers (MSPs), small and medium businesses (SMBs), enterprise IT departments and operational technology environments in manufacturing and other industrial settings. Acronis solutions are highly efficient and designed to identify, prevent, detect, respond, remediate and recover from modern cyberthreats with minimal downtime, ensuring data integrity and business continuity. Acronis offers the most comprehensive security solution on the market for enterprises, SMBs and MSPs with its unique ability to meet the needs of diverse IT and OT environments.

A Swiss company founded in Singapore in 2003, Acronis has 45 locations across the globe. Acronis Cyber Protect Cloud is available in 26 languages in 150 countries and is used by over 20,000 service providers to protect over 750,000 businesses. Learn more at www.acronis.

About the author

Christian Nicita is a Cybersecurity Governance, Risk and Compliance Specialist at Acronis, with extensive experience in the implementation of international information security standards and regulatory frameworks. His expertise covers the ISO/IEC 27000 family, CSA STAR, IEC 62443-4-1, and regulatory-specific requirements including GDPR, ENS and HDS.

At Acronis, Christian contributes to the definition of corporate security policies, the advancement of risk mitigation strategies and the integration of secure software development practices across product lines. His work supports both compliance assurance and operational resilience, bridging regulatory obligations with technical execution.

With a professional background spanning application security, penetration testing, and secure development, he provides trusted guidance that strengthens the security posture of organizations operating in highly regulated and critical industries.

Appendix A – SSDLC check-list tool

Dimension	Checklist question	Yes	Partial	No	Notes / Evidence Reference
Deployment model	Supplier has a documented secure development policy.	☐	☐	☐	
	Roles and responsibilities for software security are formally assigned.	☐	☐	☐	
	Executive oversight of secure development is in place.	☐	☐	☐	
	SSDLC is integrated into the risk management framework.	☐	☐	☐	
Risk management and design	Security requirements are defined with functional requirements.	☐	☐	☐	
	Threat modelling is performed for new systems and major changes.	☐	☐	☐	
	Design-phase risk assessments are conducted with documented mitigations.	☐	☐	☐	
	Compliance obligations (e.g., GDPR, sector-specific regulations) are embedded into design.	☐	☐	☐	
Implementation	Developers receive regular training in secure coding.	☐	☐	☐	
	Secure coding standards are enforced across projects.	☐	☐	☐	
	Code reviews are mandatory before merging into production.	☐	☐	☐	
	Automated tools (SAST, DAST, IAST, SCA) are used for vulnerability detection.	☐	☐	☐	
	Third-party dependencies are governed (e.g., SBOM in place).	☐	☐	☐	

Dimension	Checklist question	Yes	Partial	No	Notes / Evidence Reference
Verification and validation	Automated security testing is integrated into builds and CI/CD.	☐	☐	☐	
	Penetration tests are performed before major releases.	☐	☐	☐	
	Vulnerabilities are tracked with remediation workflows and timelines.	☐	☐	☐	
	Third-party or independent validation is performed.	☐	☐	☐	
Release and deployment	Build and release pipelines are hardened and monitored.	☐	☐	☐	
	Code signing is used for authenticity of updates.	☐	☐	☐	
	Dev, test and production environments are segregated.	☐	☐	☐	
	Updates are secured with integrity checks, rollback and encryption.	☐	☐	☐	
Maintenance and monitoring	Supplier has a vulnerability disclosure or bug bounty program.	☐	☐	☐	
	Timelines are defined for patching critical vulnerabilities.	☐	☐	☐	
	Customers are promptly notified of vulnerabilities with guidance.	☐	☐	☐	
	Ongoing monitoring is in place for emerging issues	☐	☐	☐	