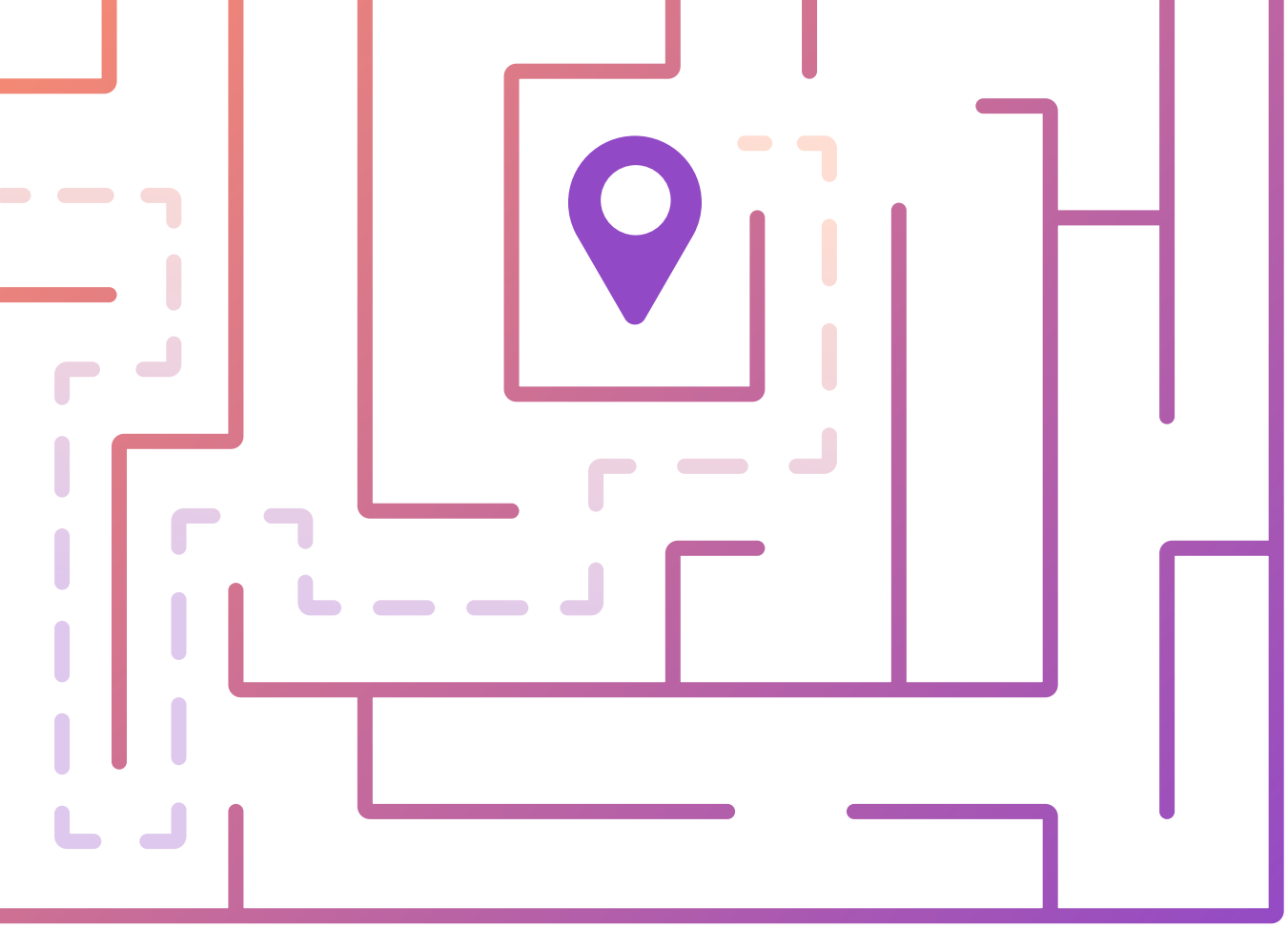

Buyer's Guide: How to Not Screw Up Application Security

A practical guide to building AppSec programs that prioritize risk over noise.

Table of Contents

Executive Summary: The Uncomfortable Truth 3

1. **Strategy First, Tools Later** 4
   - Stakeholder Discovery
   - Risk-Driven Asset Inventory

2. **Build the Business Case: Show Me the Money** 5
   - What Execs Need to Hear
   - Cross-Team Allies That Matter
   - Roadmap to Maturity

3. **Choose Tools That Fit Your Org** 6
   - The Golden Rule
   - Must-Have Tools
   - Tool Selection Tips

4. **Prove It Works: Metrics** 7
   - Success Metrics Cheat Sheet

5. **Common Pitfalls (and How to Dodge Them)** 8

Conclusion: Think Big, Start Small, Stay Focused 9
- Getting Started

Choosing the Right Tool 9

Buyer's Checklist: How to Not Screw Up Application Security 10

Executive Summary: The Uncomfortable Truth

Most Application Security (AppSec) programs fail not because of bad tools, but because they optimize for the wrong things. They chase comprehensive coverage instead of meaningful risk reduction, and measure activity instead of outcomes. They buy enterprise platforms before understanding their own workflows.

In reality: Successful AppSec is more about **organizational cultural change** than technical deployment. When only 2-5% of security findings actually matter, measuring activity does not measure risk reduction.

Building a successful program requires more than just purchasing tools. It demands a strategic approach that balances business needs, technical requirements, and organizational culture: building partnerships between engineering, security, and business stakeholders. The most successful programs are those that grow organically with strong business support and developer buy-in.

This guide provides a framework for evaluating and implementing AppSec solutions that drive real security outcomes.

1

Strategy First, Tools Later

Most AppSec programs fail because they jump straight to shiny tools. But successful security is about building relationships, not just remediation. Start by mapping your org: Conduct a comprehensive stakeholder discovery across your organization. Interview key people in development, operations, legal, and business units to understand their perspectives, concerns, and requirements. This process often takes several weeks or even months, but it's essential for building the network of allies and security champions that will make your program successful.

During these conversations, focus on understanding your organization's risk appetite and compliance requirements. Each person you speak with should be asked who else they recommend you talk to, creating an expanding web of relationships that will become the foundation of your security partnerships. Remember that security is fundamentally about developing relationships, not just throwing vulnerabilities over the wall and demanding fixes.

Stakeholder Discovery

- Interview dev leads, ops, legal, sales, GRC
- Understand risk appetite, regulatory scope, and business pressure points
- Identify security champions early

Once you understand the human landscape, turn your attention to your technical assets. Create a comprehensive inventory that prioritizes applications based on several critical factors. The most common trap organizations fall into is jumping straight to tool evaluation without understanding this business context.

Risk-Driven Asset Inventory

Prioritize apps by:

01 **Public Exposure:** Internet-facing = higher risk.

02 **Data Sensitivity:** PII, financial, IP.

03 **Business Criticality:** Revenue or customer impact.

04 **Interconnectivity:** Blast radius of compromise.

05 **Compliance Scope:** PCI, SOX, GDPR, NIS2, CRA.

Don’t just document systems. Map relationships.

2

Build the Business Case: Show Me the Money

The key to securing budget and organizational support is demonstrating that real security risk exists in your environment. Conduct initial security assessments to identify existing vulnerabilities and document any past incidents or near-misses. Quantify the potential business impact of security failures, whether through direct financial loss, regulatory penalties, or reputational damage.

Compliance requirements often provide the initial justification for AppSec investments. Whether you need to meet PCI standards for payment processing, SOX requirements for financial reporting, or GDPR mandates for data protection, these frameworks create concrete, business-driven reasons to implement security controls. Use these requirements as your initial catalyst to get the ball rolling, then expand your program to address broader security concerns.

What Execs Need to Hear

- Real risks already exist — show initial scans and incident history
- Compliance = urgency (PCI, SOX, GDPR, NIS2, CRA)
- Customers care. AppSec can help close deals

Strategic partnerships across your organization will amplify your efforts and provide ongoing support. Each team brings unique value that extends beyond traditional security boundaries.

Cross-Team Allies That Matter

Team	Superpower	Why You Need Them
Legal	Privacy & risk context	Ties security to liability
GRC	Compliance expertise	Translates security into audit wins
Sales	RFP pressure	Turns security into revenue
Dev	Workflow insight	Makes or breaks adoption

Roadmap to Maturity

0-6 months

Foundation

- Inventory
- Basic scans
- Stakeholder buy-in

6-18 months

Growth

- SAST/DAST
- SCA
- Champions
- Compliance coverage

18+ months

Maturity

- Threat modeling
- ASPM
- Culture transformation

3

Choose Tools That Fit Your Org

The Golden Rule

A tool that doesn't get used is worse than no tool at all.

The modern AppSec toolbelt centers around several core categories of security testing tools, each addressing different aspects of application risk. Integration capabilities often determine whether a security tool becomes a valuable part of your development ecosystem or an isolated system that developers actively avoid.

When evaluating tools, ensure they support the programming languages and frameworks your teams actually use. The most important characteristic is a low false positive rate combined with contextual risk analysis that helps developers understand not just what the vulnerability is, but why it matters and how to fix it.

Must-Have Tools

Tool Type	Purpose	Must-Have
SAST	Source code scanning	IDE integration, contextual fixes
DAST	Runtime testing	API support, safe in production
SCA	Third-party risk	Reachability, license management
IAST	Runtime insight	Low overhead, real-time detection
ASPM	Central brain	Correlation, prioritization, BI dashboards

Tool Selection Tips

01 Test on your own environment, not vendor demos.

02 Involve developers in trials.

03 Prioritize low false positives and workflow fit.
Remember: Only 2-5% of Application Security issues actually matter.

04 Start with pilot deployments, not org-wide rollouts.

4

Prove It Works: Metrics

Security isn't just technical - it's political. You need proof.

The modern AppSec toolbelt centers around several core categories of security testing tools, each addressing different aspects of application risk. Integration capabilities often determine whether a security tool becomes a valuable part of your development ecosystem or an isolated system that developers actively avoid.

Success Metrics Cheat Sheet

Type	Metric Example	What It Shows
Technical	Mean Time to Remediation Vulnerability Escape Rate	Responsiveness Early detection
Business	Audit Pass Rate Customer Security Wins	Compliance maturity Sales impact
Cultural	Champion Engagement Training Completion	Org buy-in Behavior change

Focus on trend lines, not just activity logs.
Executives want outcomes, not effort.

5

Common Pitfalls (and How to Dodge Them)

Shiny Tool Syndrome

What It Looks Like

Buying flashy tools that demo well but don't fit your environment.

How to Dodge It

Buy what fits your workflows and culture, not just what looks cool.

Volume Over Value

What It Looks Like

Getting flooded with 100,000+ findings most of which don't matter.

How to Dodge It

Use tools that help you focus on the 2-5% of findings that actually pose real risk.

Poor Integration

What It Looks Like

Tools that don't plug into CI/CD pipelines or developer IDEs.

How to Dodge It

Prioritize tools that work where developers already live.

Lack of Buy-In

What It Looks Like

No exec support = no budget; no dev involvement = no adoption.

How to Dodge It

Involve both execs and devs early to ensure support and traction.

Conclusion: Think Big, Start Small, Stay Focused

AppSec isn't a one-size-fits-all product. It's a long-term investment in risk management, culture change, and competitive advantage.

Every organization has unique needs based on their business model and risk tolerance, technology stack and architecture, organizational culture and maturity level, and compliance and regulatory requirements. The most successful programs start small with clear value demonstration, prove their worth through concrete risk reduction and business enablement, and scale systematically with strong stakeholder support.

With the right foundation, strategic approach, and commitment to continuous improvement, your AppSec program can become a significant competitive advantage that enables innovation while protecting what matters most to your business. Most importantly, it saves developers time spent handling security alerts, so they can focus on delivering business value.

Getting Started

01 **Start small:** Prove value with 3—5 critical apps.

02 **Build bridges:** Security is a team sport.

03 **Stay focused:** Prioritize risk reduction, not checkbox compliance.

04 **Think long term:** Culture change is the goal, not just tool deployment.

Choosing the Right Tool

Only a fraction of AppSec issues are exploitable, reachable, and impactful. OX enables AppSec and DevOps teams to go deeper than generic prioritization and fix the 5% of vulnerabilities that cause real damage.

From design to runtime, OX stops critical risks before they reach the cloud.

- **Eliminate 95%+ of irrelevant findings** that are unexploitable, unreachable, or pose minimal business risk, with intelligent prioritization
- **Context-aware analysis** that understands your application architecture and data flows
- **Developer-friendly integration** with IDE extensions that provide real-time feedback in familiar environments
- **Reduced friction** by embedding security seamlessly into existing CI/CD pipelines and workflows
- **Catch vulnerabilities before runtime** with proprietary Code Projection technology, comprehensive static and dynamic analysis across the development lifecycle
- **AI-powered 1-Click Fix** that provides easy remediations tailored to your code

OX

Ready to put it into action?
Book a demo

Buyer's Checklist: How to Not Screw Up Application Security

0—6 months

Foundation: Strategy First, Tools Later

Stakeholder Discovery & Network Building

- [ ] Interview C-Suite, dev, ops, legal, GRC, and sales leads to map risks, workflows, and incentives. Understand each stakeholder's KPIs: uptime, velocity, compliance, deal flow, customer SLAs - and where security can support them. Recruit early security champions across departments.

Asset Inventory

- [ ] Inventory all apps, APIs, and services across teams and business units. Prioritize by exposure, sensitivity, criticality, interconnectivity, and compliance scope

Build the Business Case

- [ ] Run initial scans to uncover existing vulnerabilities + Document past incidents. Quantify potential business impact in financial and operational terms. Tie security goals to compliance, customer trust, and uptime. Invest in lasting alliances with legal, GRC, dev, and sales

6—18 months

Growth: Operationalize Security, Measure Outcomes

Tool Selection & Deployment

- [ ] Choose tools that fit your stack, workflows, and developer habits. Ensure integration with IDEs, CI/CD, ticketing, and SDLC systems. Run trials in your own environment — avoid vendor-only demos. Involve developers in evaluation, rollout, and continuous feedback.

Metrics & Measurement

- [ ] Track MTTR and vulnerability escape rate. Track audit pass rates and customer security wins. Track champion engagement and training completion. Report trends over time — not just point-in-time metrics. Translate metrics into business terms for execs and boards.

18+ months

Maturity: Scale What Works

Unified Governance

- [ ] Implement ASPM to unify visibility, correlate findings, and govern remediation. Normalize and prioritize across tools to reduce alert fatigue and noise. Build a consistent reporting cadence for execs, security, and engineering.