Getting Started with Security by Design

Rohit Sethi

CEO, Security Compass

Security Compass

Table of Contents

What is Security by Design?

In the current cybersecurity landscape, the need for robust and proactive security measures has never been more critical. Security by Design is a philosophy emphasizing integrating security into systems from the beginning of the development process. Unlike traditional application security approaches that rely on testing to identify vulnerabilities after the fact, Security by Design integrates security activities during the planning, analysis, and design phases well before coding begins. This proactive approach ensures that potential security issues are addressed early, reducing the likelihood of vulnerabilities in the final product.

Key Practices of Security by Design

Security by Design encompasses a variety of processes and tools aimed at embedding security into the development lifecycle. The following key practices are essential components of this approach:

1. **Training** - Educating developers, QA engineers, and other project members in secure development practices. This training covers common vulnerabilities, secure coding techniques, and the importance of security throughout the development lifecycle.

2. **Threat Modeling** - Identifying inherent application threats based on their programming language, frameworks, and deployment environment. Threat modeling helps teams understand what can go wrong and how to prevent these issues.

3. **Security Requirements** - Establishing security and compliance requirements to ensure appropriate safeguards are built into the system. This involves defining security controls and policies that must be implemented throughout the development process.

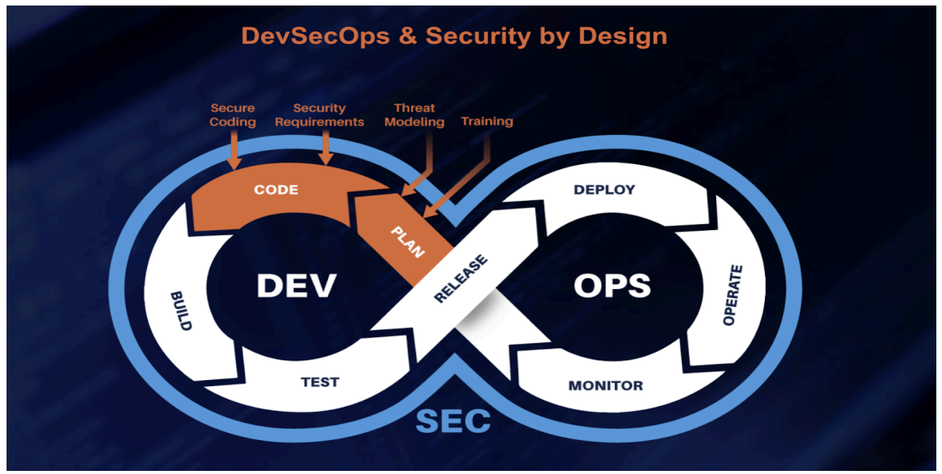
4. **Secure Coding Guidelines** - Creating and adhering to guidelines that promote secure coding practices. These guidelines serve as a reference for developers to follow best practices and avoid common pitfalls that could lead to vulnerabilities.

When to Adopt Security by Design?

While there are many business benefits to adopting Security by Design, starting a program typically requires funding as well as cultural processes and cultural change. In our experience, there are a few factors to consider when deciding whether or not to adopt a security by design program:

1. Custom Software Development

Organizations that build and ship software products or build a significant amount of custom software in-house are more likely to benefit from Security by Design than others. For example, a retailer that typically purchases software is less likely to benefit from the program than a Software As A Service (SAAS) company.



2. DevSecOps Adoption

DevSecOps is a philosophy or approach that integrates security into every step of the software development process and toolset. It emphasizes collaboration between development, security, and operations teams to ensure security is a shared responsibility. DevSecOps leverages automation and continuous integration/continuous deployment (CI/CD) pipelines to embed security checks and controls throughout the development lifecycle. In our experience, organizations typically embrace DevSecOps before they are ready to benefit from Security by Design.

3. Sensitive Data

Organizations that process, store, or transmit sensitive data in custom software are more likely to benefit from security by design. For example, companies that store customers' Personally Identifiable Information (PII), like names and addresses, are more likely to benefit than those that have low confidentiality, integrity, and availability requirements.

Why Adopt Security by Design?

In today's rapidly evolving digital landscape, ensuring robust software security from the ground up has become more critical than ever. Security by Design is a proactive approach that embeds security considerations into every phase of the software development lifecycle, starting from the planning and design stages. This contrasts with traditional methods that rely heavily on testing to identify vulnerabilities post-development. But why should organizations adopt Security by Design? Understanding the value drivers behind this approach is critical to appreciating its benefits.

The Need for Security by Design

Practitioners often articulate the benefits of Security by Design in purely technical terms, such as

"Getting ahead of vulnerabilities" | "Improving maturity in secure SDLC"

However, these benefits might not resonate with non-technical stakeholders.

Moreover, implementing Security by Design represents a long-term, systemic change, which can take years to realize and is often prone to being deprioritized by other initiatives. Successful adoption begins with articulating the business benefits in clear, quantifiable terms that matter to the broader organization.

Value Drivers for Security by Design

There are four primary value drivers for organizations to adopt Security by Design: Reducing Operational Costs, Reducing Risk, Improving Software Security at Scale, and Growing Revenue by Demonstrating Compliance.

1

Reduce Operational Cost

Automate security and compliance to increase efficiencies and reduce total cost of ownership.

Value Drivers for Security by Design

There are four primary value drivers for organizations to adopt Security by Design:

1. Reducing Operational Costs
2. Reducing Risk
3. Improving Software Security at Scale
4. Growing Revenue by Demonstrating Compliance

1 Reduce Operational Cost

Automate security and compliance to increase efficiencies and reduce total cost of ownership.

Without Security by Design:

- ✗ Applications are created with vulnerabilities that are expensive to remediate.
- ✗ Security expertise is costly and hard to find.
- ✗ Development teams are blocked by limited access to security experts.
- ✗ Excessive time is spent on remediating audit deficiencies and preparing for audits.

With Security by Design:

- ✓ Code is written with fewer preventable vulnerabilities, resulting in less rework.
- ✓ Developers take ownership of security outcomes, reducing reliance on scarce security experts.
- ✓ Integrating compliance by design reduces the time and effort spent responding to audits.

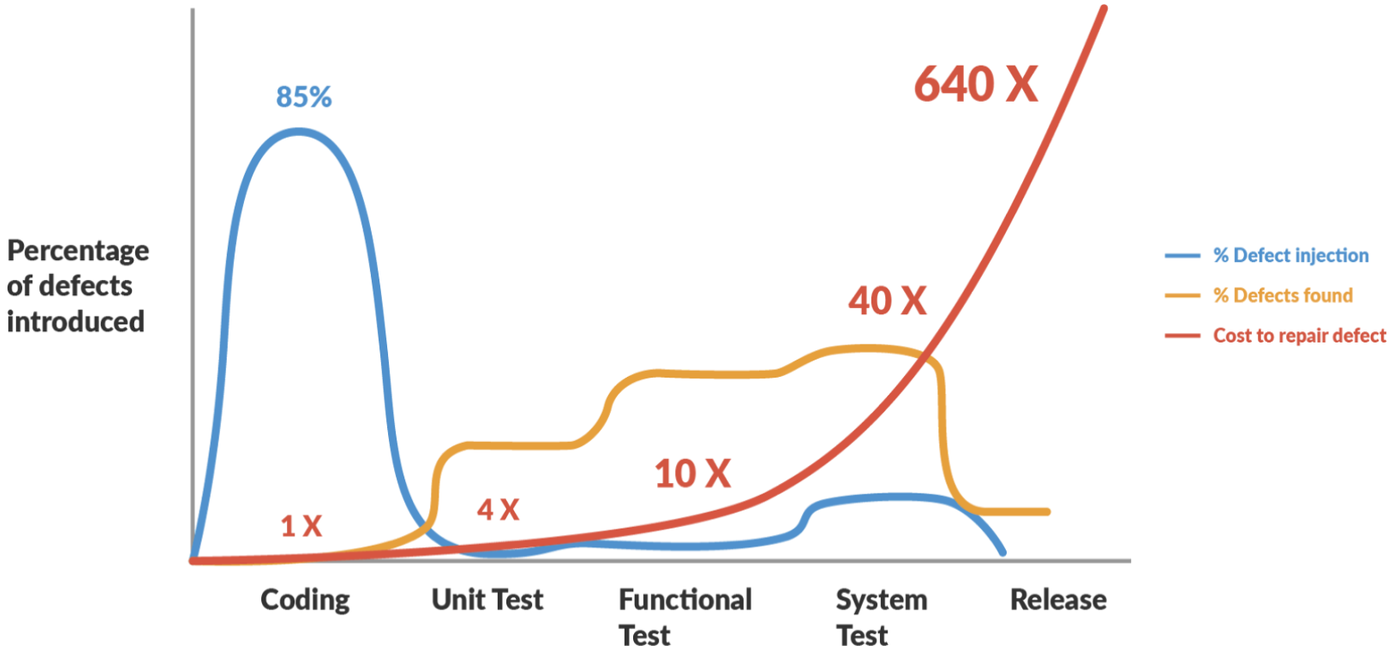


Figure 1: Jones, Capers. Applied Software Measurement: Global Analysis of Productivity.

Considerations for System Implementation:

When you implement systems for Security by Design, consider the following in order to maximize operational cost savings:

- **Manual vs. Automated:** Balance between automated tools and manual processes for efficiency.
- **Knowledge Base:** Ensure technology and compliance knowledge is up-to-date and comprehensive.
- **Education:** Provide ongoing training for non-experts to keep everyone informed.
- **Integration:** Seamlessly integrate security practices with existing tools and processes.
- **Speed of Change:** Implement changes at a pace that the organization can adapt to.

2 Reduce Risk

Reduce security and compliance risk.

Without Security by Design:

- ✗ Vulnerabilities are accepted into production.
- ✗ Remediation rates are lower than desired.
- ✗ Software is flagged as non-compliant during regulatory audits.
- ✗ Security metrics are difficult to obtain and prioritize for action.
- ✗ Organizations are forced to accept avoidable risks of breaches.
- ✗ Vulnerable applications are released, hoping to fix issues in the next release.
- ✗ Incidents and breaches occur due to vulnerable software, leading to fines and changing business priorities.

With Security by Design:

- ✓ Reduced exposure to liability in the event of a breach by following best practices.
- ✓ Fewer vulnerabilities in software, improving overall security posture.
- ✓ Better visibility and tracking of security and compliance efforts.
- ✓ Developers know security and compliance requirements and learn how to remediate vulnerabilities.
- ✓ Reduced risk of fines and lawsuits due to breaches.
- ✓ Lower operational risk, including better customer retention.
- ✓ Fewer regulatory audit findings and lower costs to remediate vulnerabilities when caught earlier in the SDLC.

Considerations for Implementation:

- **Effort Allocation:** Apply effort based on the inherent risk of the project.
- **Education:** Educate stakeholders at the appropriate time, all at once, or as needed.
- **Audit Trails**: Maintain detailed audit trails to ensure compliance.
- **Central Reporting**: Use central reporting to drive behavior and prioritize actions.
- **Policy Conformance**: Ensure conformance to internal policies through linked testing activities.

3 Improve Software Security at Scale

Protect the business and improve application security across the software portfolio.

Without Security by Design:

- ✗ Software release delays occur due to waiting for security sign-offs or remediating vulnerabilities identified late.
- ✗ Unplanned work as vulnerability remediation is not built into the project plan.
- ✗ Frustration among business and development stakeholders due to missed deadlines.
- ✗ Under-resourced application security teams rely heavily on automated testing results.
- ✗ Security experts focus only on high-risk applications, leaving others without thorough security analysis.

With Security by Design:

- ✓ Security requirements are built into planning, turning unplanned work into planned work with measurable outcomes.
- ✓ Faster time to market with fewer delays from remediation and reduced bottlenecks waiting for security experts.
- ✓ Increased knowledge and accountability from development teams, reducing reliance on manual expertise.
- ✓ Improved relationships between development and security teams through better planning and coordination.

Considerations for Implementation:

- **Self-Service for Developers:** Enable developers to access security resources independently to speed up development.
- **Reporting:** Improve oversight with detailed reporting so security stakeholders can assess risk without direct involvement.
- **Compliance**: Make compliance standards clear and easy to understand for developers, avoiding subjective interpretation.
- **Integration**: Ensure seamless integration with existing technology to increase the speed of process execution.
- **Flexibility**: Adapt training and modeling styles to fit different teams and project needs.

4 Grow Revenue by Demonstrating Compliance

Enable rapid, secure, and compliant software delivery to meet market demands.

Without Security by Design:

- ✗ Non-compliance with standards results in decreased market opportunities and loss of revenue streams.
- ✗ Brand damage occurs when products are found to be non-compliant.
- ✗ Obtaining compliance without embedding it into design leads to costly rework and unplanned work.
- ✗ Lack of expertise in various standards and frameworks slows down development.

With Security by Design:

- ✓ Revenue growth by accessing markets that require compliance.
- ✓ Creating barriers to entry against competitors who have not yet achieved compliance.
- ✓ Building secure products required by law leads to additional market opportunities.

Considerations for Implementation:

- **Actionable Guidance:** Translate broad compliance requirements into specific, actionable steps.
- **Normalization:** Normalize compliance requirements across multiple standards to prevent overlap and rework.
- **Integration with GRC**: Ensure compliance by design, which integrates with the broader Governance, Risk, and Compliance (GRC) program to avoid redundant information in multiple systems.
- **Progress Reporting**: The capability to report progress against compliance standards is essential for understanding advancement.
- **Detailed Audit Trails**: Provide sufficient evidence that standards and regulations were adhered to.

Legal Requirements for Product Vendors

In addition to growing revenue by demonstrating compliance, some product vendors are required to build secure products by law.

- **European Cyber Resilience Act (CRA)**: "Cybersecurity is taken into account in planning, design, development, production, delivery and maintenance phase"
- **US Executive Order (EO) 14028**: Shifting cyber responsibility back to manufacturers
- **US Cyber Trust Mark**: A cybersecurity labeling program for smart devices
- **Industry Specific Regulators & Supervisory Bodies**: OSFI (Canada), OCC (US), PCI - Software Security Framework, FDA, etc

Change Environment: Security by Design Required

"Relentless and increasingly sophisticated cyberattacks threaten the ability to realize the full benefits of the 21st century digital economy"

- Executive Order 14028 Update, 2022

"Our aim is to strengthen trust in the connected economy and keep our citizens digitally secure."

- European Union Agency for Cybersecurity (ENISA)

"PR.PS-06: Secure software development practices are integrated, and their performance is monitored throughout the software development life cycle"

- NIST Cybersecurity Framework 2.0 Update, 2024

EXPAND

A first ever EU wide legislation of its kind: the **Cyber Resilience Act** introduces **mandatory cybersecurity** requirements for hardware and software products, **throughout their whole lifecycle**.

- European Union 2022 - 2023

4

Why Do Businesses Need Security by Design?

In today's fast-paced digital landscape, the importance of robust cybersecurity cannot be overstated. Security breaches can lead to significant financial losses, reputational damage, and legal consequences. To mitigate these risks, businesses must adopt a proactive approach to cybersecurity. Security by Design proactively integrates security into every phase of the software development lifecycle, starting from the planning and design stages.

Let's explore the key benefits and how to build a compelling business case for security by design.

Building a Business Case for Security by Design

Security by Design program is a paradigm shift for most organizations. Switching mentalities from finding and fixing defects to building security requires organizational change management. We need to help answer the question "Why do businesses need security by design?. In our experience, it's crucial to quantify benefits to gain buy-in from business stakeholders. Using your organization's data to build the business case is ideal, but in many cases, organizations may lack the necessary data points.

Here, we provide metrics, formulas, and industry data to help you quantify a business case.

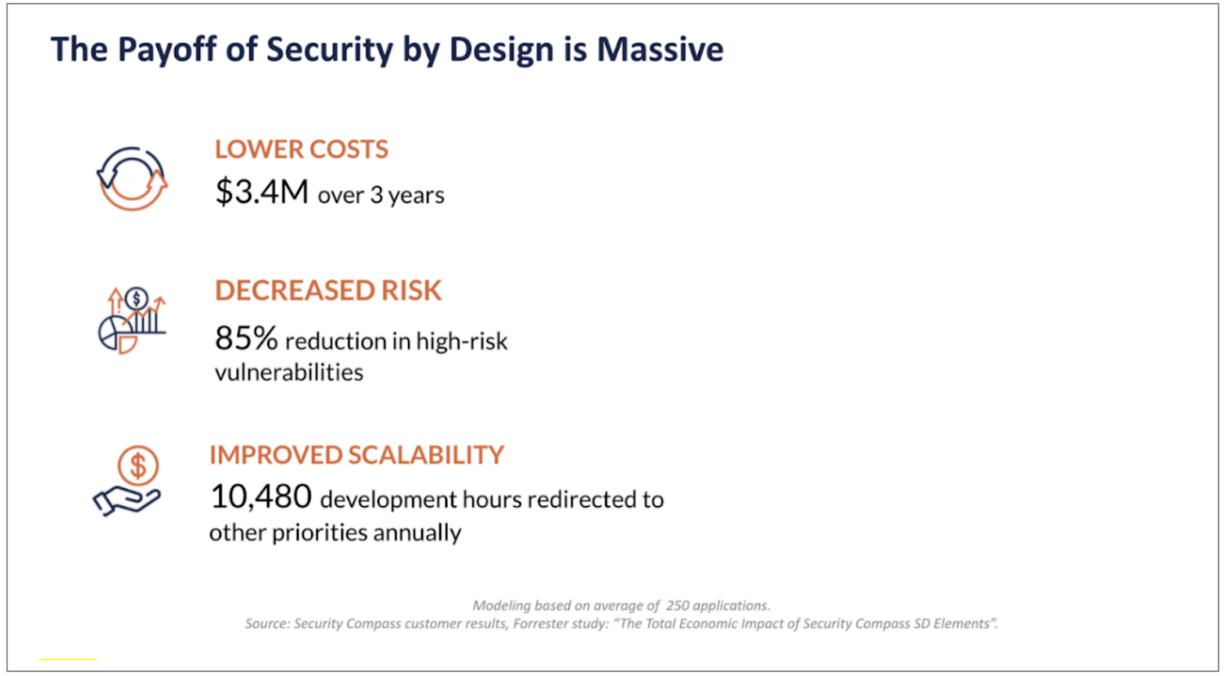


Figure 1: Actual results of analysis from implementing a Security by Design program.

1. Reduce Operational Costs

Security by Design often offers the highest Return on Investment (ROI) among cybersecurity programs. While other initiatives aim to reduce the likelihood of a breach, Security by Design also reduces the costs of securing software compared to reactive approaches.

Here are three primary ways it achieves this:

- **Avoid Vulnerability Remediation**

  Data from our customer base indicates that fixing an average vulnerability costs $50,156. Given that the average application has six high or critical-risk vulnerabilities. Implementing Security by Design can conservatively reduce vulnerabilities by 79% compared to simply testing for security issues after the software has been built. This results in significant cost savings per application. For example, a global company that adopted Security by Design could see estimated savings such as below:

Figure 2: Savings achieved with Security by Design implementation.

• Decrease Time Spent on Compliance

Responding to audits and building artifacts to demonstrate compliance can be onerous for software teams. Taking a by-design approach with built-in audit trails allows organizations to reduce the time and effort required for compliance. This proactive approach ensures that security and compliance requirements are met from the outset, avoiding the need for extensive rework penalties associated with noncompliance.

	YEAR 1
Number of Audits (at Attestations) per year	10
Hours saved per Certification	40
Total hours saved on compliance certifications because of SDE	400
Average Annual Salary of Senior Security Engineers	$150,000
Average Hourly Rate of a Senior Security Engineer	$72
Fully Loaded Adjustment Percentage	4.5%
Fully loaded hourly rate of senior secuirty engineer	$75
Decreased cost spent on complaince certifications	$30,144
Risk adjustment %	10%
Subtotal Risk adjustment	$3,014
Decreased time spent on complaince certifications (risk-adjusted)	**$27,130**

EXPAND

Figure 3: Detailed breakdown of the cost savings achieved through the implementation of Security by Design.

• Reduce Costs with Automation

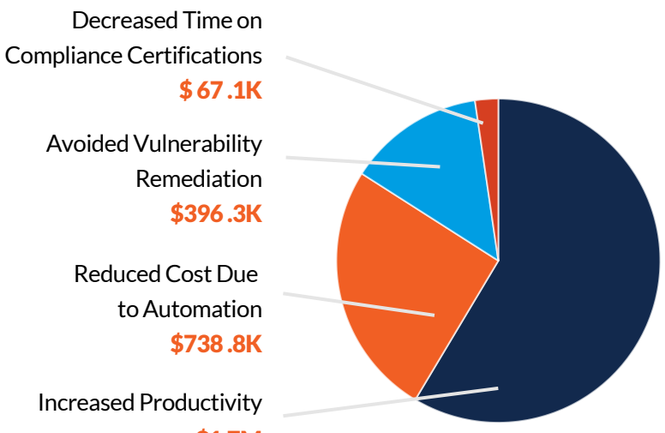
While some Security by Design activities, such as threat modeling and security requirements generation, can be done manually, automation significantly enhances ROI. Automated tools reduce the number of person-hours needed to perform these tasks, leading to substantial cost savings.

For instance, a company that automated its Security by Design processes saved $2.86 million over three years.

	YEAR 1
Number of Projects Onboard	55
Reduced hours to understand & integrate requirements	16
Fully Loaded Hourly rate of senior security engineer (in $)	$75
Subtotal	$66,000
Fully Loaded Annual salary of senior security engineer (in $)	$150,000
Portion of FTE to maintain requirements for organization	0.33
Subtotal cost of 1/3 time of FTE	$49,500
Reduced Costs	$115,500
Risk adjustment %	15%
Subtotal Risk adjustment	$17,325
Total reduced costs due to automation (risk adjusted)	**$98,175**

EXPAND

Figure 4: Detailed breakdown of cost savings achieved through Security by Design automation in the first year.



Quantified Benefits $2.86M Over 3 Years

Figure 5: Example Quantified Benefits of Security by Design

2. Reduce Risk

Another key benefit of Security by Design is a lower risk due to software due to preventing vulnerabilities in software. Risks are notoriously difficult to measure and communicate to business stakeholders. Many security organizations report on measures like the number of vulnerabilities and Mean Time To Remediation (MTTR) for security, but these aren't necessarily meaningful to non-technical stakeholders.

An alternative method to measure risk in a way that's more intuitive to a non-technical stakeholder is a window of exposure: The number of days that a high or critical risk vulnerability is in production. Since more than one vulnerability may be exposed at one time, it's best to think of this as a unit of measure rather than calendar days.

Quantitative Risk Analysis

FAIR provides a richer mechanism to express risk in business terms. Using quantitative methods, FAIR allows practitioners to express risk in terms of loss exposure in dollars. In practice, FAIR has a learning curve and sometimes faces resistance from practitioners who are skeptical of quantitative risk management for cybersecurity. Using FAIR is outside the scope of this document, but we encourage you to consider it as a method of measuring risk to be presented to executives and boards.

Figure 6: Edge scan 2023 Vulnerability Statistics Report: https://www.edgestan.com/intel-hub/stats-report/

3. Improve Software Security at Scale

Security by Design ensures that security practices are scalable and sustainable across multiple applications and projects. You can assess the impact of scalability in two ways:

1. Determine the amount of time saved per application by using security by design vs. more reactive methods. This results in faster time to market.
2. Determine how many applications a security architect, application security analyst, or other role can effectively serve with and without the security by design program.
3. Finally, if automation is used in the program, you can also assess the speed of using automation vs. manual methods.

Finally, if using automation in the program you can also assess the speed of using automation vs. manual methods.

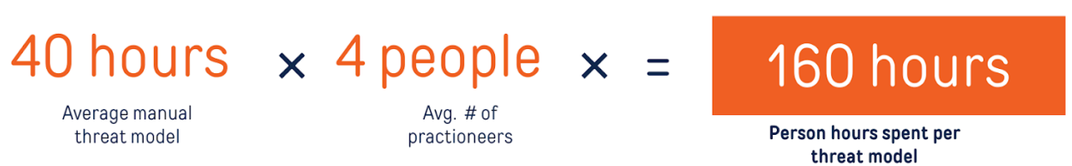


Figure 7: Time Savings Achieved with Security by Design Implementation.

4. Grow Revenue by Demonstrating Compliance

Adopting Security by Design can open new market opportunities and enhance revenue growth by demonstrating compliance with regulatory standards. You can calculate the impact of demonstrating compliance to grow revenue in two ways:

- Estimated sales as a result of demonstrating compliance (or loss of revenue as a result of not showing compliance)
- Understanding the impact on the Total Addressable Market (TAM) as a result of obtaining compliance

Note that in many cases, compliance is not optional. In these cases, you may want to show how the Security by Design method of demonstrating compliance is more efficient than building software, finding audit defects and subsequently fixing them. These calculations are already captured in the “Reduce Operational Costs” driver.

The Cost of Inaction

Implementing Security by Design is a strategic initiative that often competes with more tactical, urgent cybersecurity or IT projects. To effectively prioritize Security by Design, it's useful to articulate the opportunity cost of not implementing this program.

One effective way to communicate this is by quantifying the value drivers and expressing them in monthly terms. For example:

- **Remediation Savings:** If the anticipated remediation savings for 100 applications amount to approximately $5,102,400 annually, delaying the Security by Design program by one month represents an opportunity cost of $425,200.

- **Window of Exposure:** If the total window of exposure for 100 applications is 39,000 days in aggregate over a year, a one-month delay equates to an additional exposure of 3,250 days.

Quantifying the opportunity cost this way helps drive urgency and facilitates a more concrete and quantitative analysis of the benefits, making it easier to prioritize Security by Design over other initiatives.

Qualitative Drivers

Quantitative measurements are critical to any business case. However, in our experience, qualitative drivers are often important in motivating people to change.

One powerful qualitative driver is reputation. It involves asking the question, "When a breach occurs from a vulnerability in our software, do we want to show that we practiced security by design?" Given the rising global awareness of security by design, this is a difficult question for business leaders to say no to.

Other qualitative drivers include improved collaboration between security and software development, turning unplanned work (i.e., vulnerabilities) into planned work, and development teams feeling empowered to make security decisions.

How to Build a Program Plan for Security by Design?

Security by Design is a proactive approach to embedding security into every phase of the software development lifecycle. Once you have established the value drivers and created a compelling business case for Security by Design, the next crucial step is to build a comprehensive program plan.

This plan should include clearly defined goals, a stakeholder engagement plan, and phased execution to ensure successful adoption and implementation. Here's how to build an effective program plan for Security by Design.

Establishing Goals

Setting clear and measurable goals is the foundation of a successful Security by Design program. These goals should be derived from the value drivers and business case you have established.

To ensure alignment and buy-in, we recommend conducting a goal-planning workshop with key stakeholders.

Figure 1: The five elements of a goal-planning workshop

The workshop should follow these five steps:

1. Pain Points: Identify the specific pain points and challenges the Security by Design initiative aims to address. This includes understanding the current security gaps, vulnerabilities, and areas requiring improvement.

2. Goals: Define the overall goals of the Security by Design initiative. These should be broad, high-level objectives that align with the organization's strategic priorities and address the identified pain points.

3. Actionable Objectives: Break down the goals into specific, actionable objectives. These should be clear tasks and initiatives that can be implemented to achieve the broader goals. Examples include conducting regular threat modeling sessions, implementing secure code reviews, and providing secure coding training to developers.

4. Metrics/Measures: Determine the metrics and measures that will be used to track progress and success. These should be quantifiable and include proactive metrics such as the number of threat models created, the percentage of secure code reviews completed, and the number of developers trained.

5. Success Criteria: Establish criteria for success to evaluate the effectiveness of the Security by Design program. This includes setting specific targets for each metric, such as reducing high-risk vulnerabilities by 50% within the first year and defining what successful implementation looks like for each objective.

Example output from a goal planning workshop:

- **Pain Point:** The current process of reviewing applications by a security team member results in a bottleneck and impacts our ability to ship software quickly
- **Goal:** Increase scalability of security design reviews
- **Actionable Objectives:**

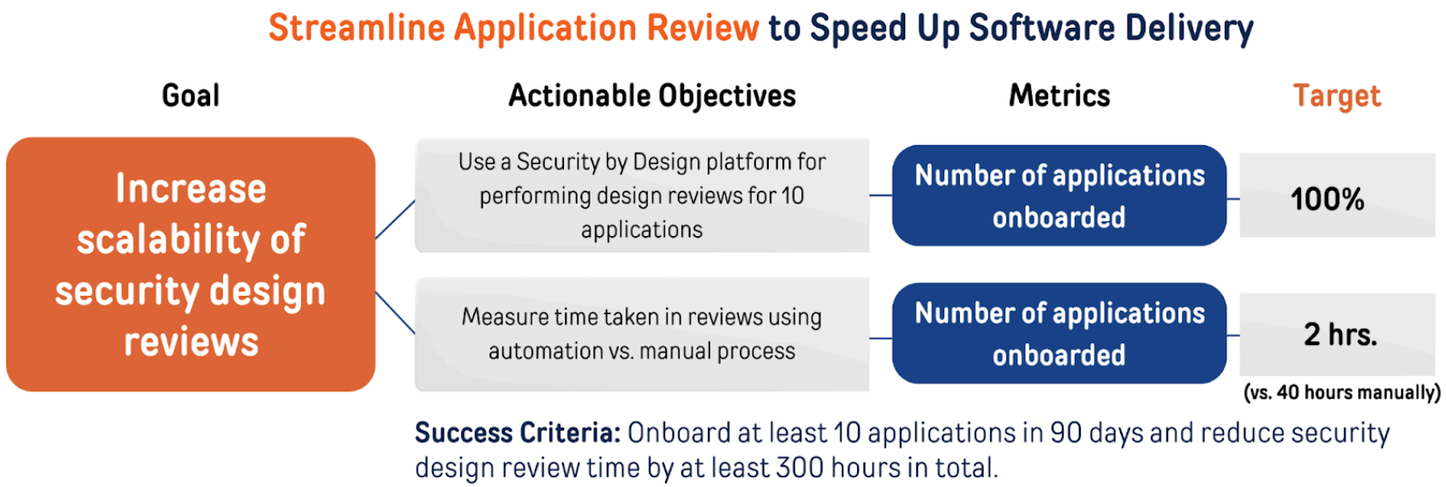
  - Use a Security by Design platform for performing design reviews for 10 applications

    Metric: Number of applications onboarded, Target: 100%

  - Measure time taken in reviews using automation vs. manual process

    Metric: Time to complete process using automation, Target: 2 hrs. (vs. 40 hours manually)

- **Success Criteria:** Onboard at least 10 applications in 90 days and reduce security design review time by at least 300 hours in total.



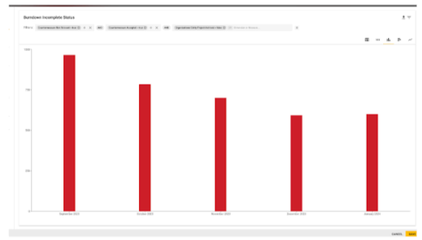
Selecting Metrics and Targets

Selecting metrics and targets is one of the most important aspects of the program. Like W.E. Deming said, “What’s measured gets done.”. Often, application security programs only measure lagging, reactive metrics such as vulnerability count, defect density, and Mean Time To Resolution (MTTR).

If these are the only benchmarks for measurement, security and development teams may not be encouraged to work on Security by Design activities. This is because Security by Design takes time to implement and ultimately impacts the lagging indicators. Security requirements added to a backlog may take months or years to fully implement as they compete with user-facing features for development time.

For executive reporting, tie these metrics back to business value, such as reduced remediation costs, improved compliance, and enhanced risk management.

Examples of proactive metrics include:

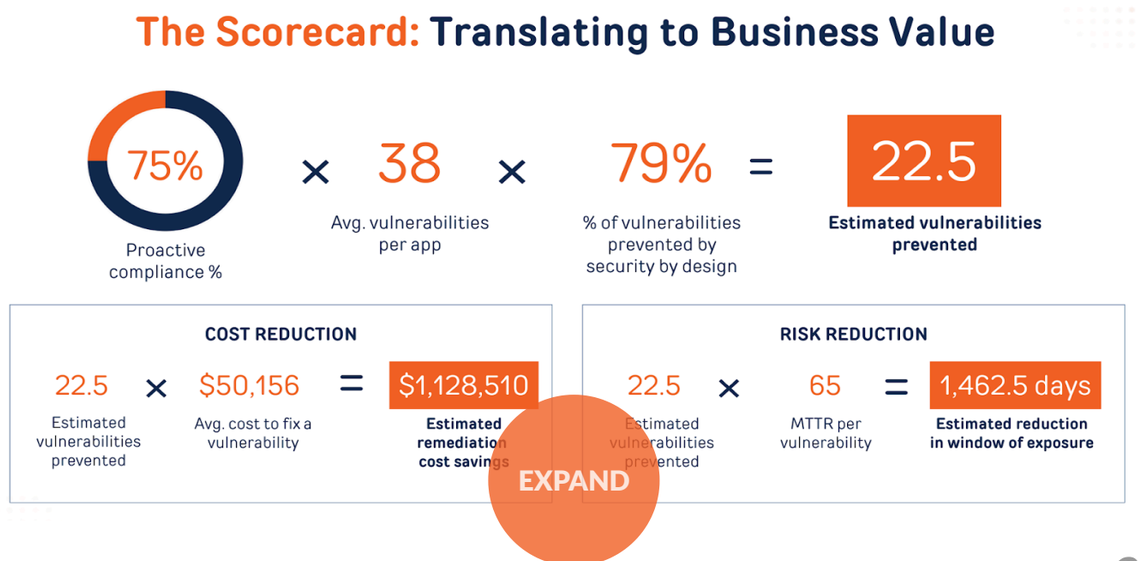


25:1

Proactive Compliance
% of relevant security & compliance controls/requirements implemented

Security Burndown
Time series of tracking of completion of controls/ requirements by application

Coverage Ratio
Average number of applications effectively served per Security Architect (or similar)



Stakeholder Engagement

Creating a business case is just the beginning of launching a Security by Design initiative. It's crucial to engage and gain buy-in from all relevant stakeholders. This can be achieved by outlining the key benefits for each stakeholder group and using a framework like ADKAR to guide your engagement strategy:

1. Initiate

- Objective: Start by explaining the need for Security by Design to the stakeholders.
- Action: Conduct initial meetings to highlight the importance and benefits.

2. Nurture

- Objective: Foster an understanding of the roles and responsibilities each stakeholder has in the program.
- Action: Provide training sessions and resources to build knowledge.

3. Support

- Objective: Offer continuous support and resources to stakeholders throughout the implementation.
- Action: Set up a help desk or support system for ongoing assistance.

4. Participate

- Objective: Engage stakeholders actively in the implementation process.
- Action: Involve them in planning sessions and decision-making processes.

5. Implement

- Objective: Execute the plan with the active involvement of all stakeholders.
- Action: Carry out the defined actions and strategies to embed Security by Design.

6. Review

- Objective: Regularly assess the progress and impact of the implementation.
- Action: Conduct periodic reviews and gather feedback from stakeholders.

7. Evolve

- Objective: Ensure continuous improvement and adaptation based on feedback and changing requirements.
- Action: Update the strategies and plans to address new challenges and opportunities.

Here are some common stakeholder groups and their benefits:

Role	Responsibility	Common Motivators	Desired Outcomes
CISO, Chief Product Security Officer, or Security Leader	- Program executive sponsor - Drives board and technology leadership support	- Scale security team - Reduce risk - Demonstrate compliance - Reduce liability in event of breach	- Increased # of applications served per full time employee - Decreased # of vulnerabilities - Audit artifacts demonstrating compliance with laws, standards, and best practices
CTO or Development Leader	- Top-down support and driving adoption program from development teams	- Demonstrate compliance to improve competitiveness - Reduce liability in event of breach - Improve time to market by reducing security bottlenecks	- Audit artifacts demonstrating compliance with laws, standards, and best practices - Faster project delivery / less time spent on security activities
Application Security Team Member	- Champion program - Drive adoption - Select & administer tools	- Reduce risk across all development teams - Scale limited number of security professionals - Demonstrate compliance	- Increased adoption of security by design activities - Decreased # of vulnerabilities - Increased # of applications served per full time employee - Audit artifacts demonstrating compliance with laws, standards, and best practices
Security Champion	- Lead security by design activities, such as threat modeling on development team - Drive adoption of training - Answer security questions and liaise with central security team	- Reduce risk in their applications - Improve career prospects with security knowledge	- Increased adoption of security by design activities - Accreditations/certifications demonstrating security knowledge
Software Developer	- Participate in security by design activities, including threat modeling - Take security training	- Reduce amount of time spent on security & compliance, such as remediating vulnerabilities and audit findings - Improve career prospects with security knowledge	- Faster project delivery / less time spent on security activities - Accreditations/certifications demonstrating security knowledge
Application Owner (Product Manager or Business Lead)	- Provide business context for threat modeling	- Turned planned work (security vulnerabilities) into planned work (security requirements) - Satisfy compliance requirements	- List of security requirements in backlog - Audit artifacts demonstrating compliance with laws, standards, and best practices
Application/Information Security Architect	- Provide information on company security architecture standards and strategy	- Ensure security is integrated into design	- Reports demonstrating security has been embedded into design
Compliance and Risk Management Lead / Internal Audit	- Assess development teams on compliance status	- Ensures compliance status of applications	- Audit artifacts demonstrating compliance with laws, standards, and best practices

Project Plan

The project plan for rolling out Security by Design will differ depending on the size and goals of your company. However, successful programs generally follow four phases of execution:

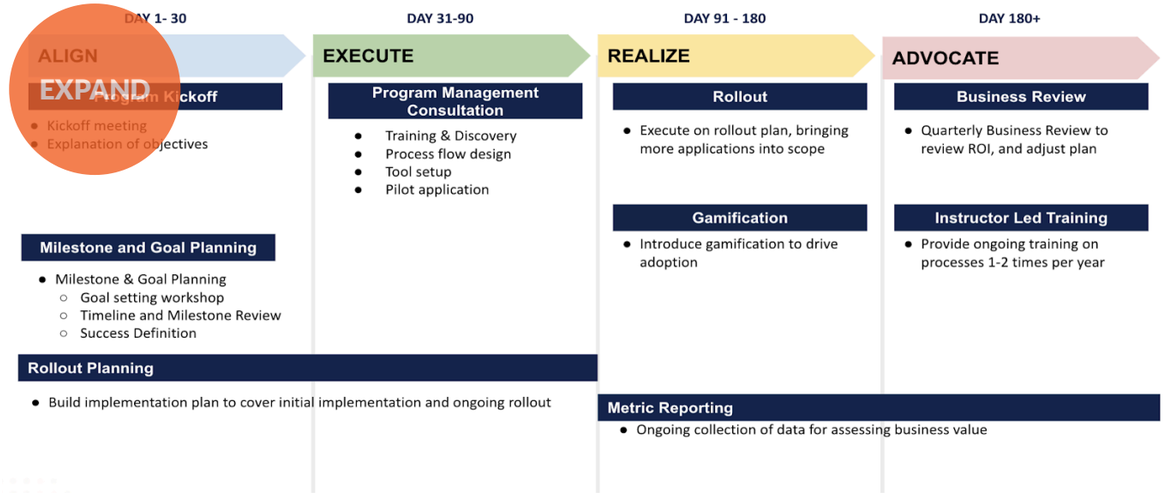
1. **Align:** Create a detailed plan and ensure all stakeholders are aligned with the goals and responsibilities.
    - Conduct goal planning workshop
    - Define scope, metrics, targets, responsibilities, and timeline
    - Secure executive sponsorship

2. **Execute:** Begin implementation with a pilot team, setting up tools and processes to test the approach.
    - Select a pilot team and application.
    - Implement Security by Design practices (e.g., threat modeling, secure code reviews)
    - Gather feedback and refine processes.

3. **Realize:** Expand the rollout based on feedback and results from the pilot phase, measuring impact and making necessary adjustments.
    - Expand to additional teams and applications
    - Measure and report on key metrics
    - Adjust program based on feedback and results

4. **Advocate:** Advocate for the program across a broader set of applications, leveraging successes from earlier phases to drive adoption.
    - Share success stories and case studies
    - Conduct training and awareness sessions
    - Drive continuous improvement and scale the program

Example Plan for a Large Enterprise



Communication Plan

Effective communication is crucial for the successful rollout of Security by Design. Your communication plan should include multiple steps and channels to ensure all stakeholders are informed and engaged.

Key elements of the communication plan include:

- **Kickoff Meetings:** Host one or more meetings to describe the program, answer questions, and set expectations.
- **Mass Email:** Send an email announcing the program to all stakeholders, soliciting participation (if voluntary), or describing necessary changes.
- **Follow-up Emails:** Send periodic emails to provide updates, share progress, and address any concerns.
- **Cascading Communication:** Use security champions to drive follow-up communications within their teams.
- **Additional Channels:** Utilize other channels such as Slack, one-on-one meetings, staff meetings, town halls, and leadership meetings to reinforce messages.
- **Wiki/Intranet Page:** Create a central repository of information, including FAQs, training materials, and progress reports.

Your communications should address the following themes:

- Why the organization is adopting Security by Design
- How success will be measured
- Which processes and tools will be used
- How existing processes will change
- Who is accountable for following the processes
- Stakeholder-specific benefits
- How to provide feedback

What Is “Educate” In The 3E Framework?

The 3E framework is a comprehensive approach that helps organizations integrate security into their software development lifecycle. The framework prescribes three sequential steps: Educate, Embed, and Empower, in increasing order of maturity and impact. The first step is "Educate," which focuses on training development teams and business stakeholders on security.

This foundational step ensures that everyone involved in the development process understands the importance of security and is equipped with the necessary knowledge and skills to implement secure practices.

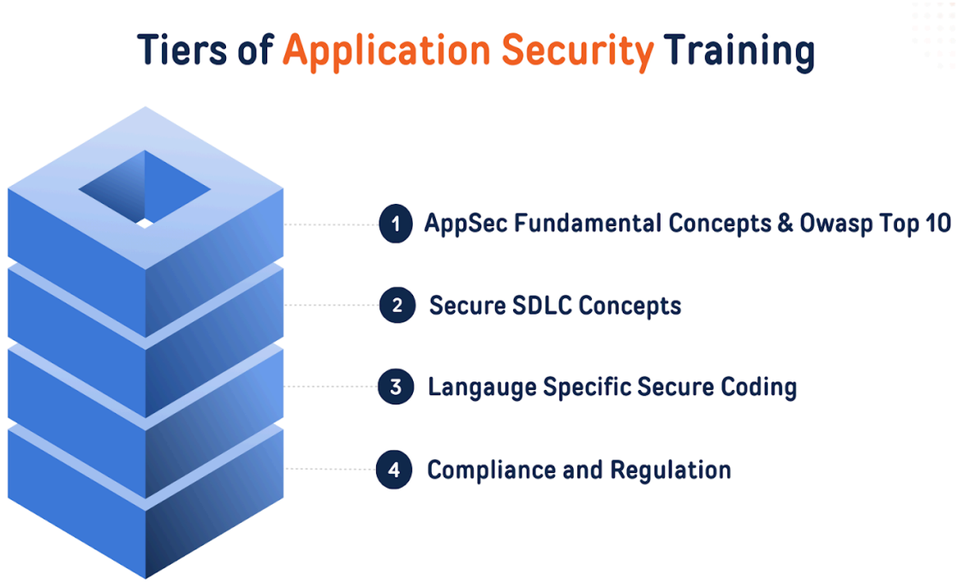
Development Team Training

Standard computer science curriculums do not always mandate security training. As a result, many new programmers are often unfamiliar with application security when they start working. Without knowing about security, development teams may suffer from “ignorance is bliss”: they may believe their products are secure without knowing what application security entails. In our experience, any effort to be “secure by design” is unlikely to be successful unless development teams are aware of the magnitude of application security risks. Data from IBM’s Cost of a Data Breach report shows that employee training has the second highest impact on reducing the cost of a breach after adopting a DevSecOps approach.

Training development teams is a crucial first step for fostering a security-conscious culture within an organization. There are various methods to achieve this, and understanding the preferences and needs of your team is key to selecting the most effective training approach.

What to Train Development Teams On

Application security is a broad domain that you can spend your entire career learning about. We recommend developers start with fundamental education and move on to other topics from there:



Training Methods

A survey conducted by Security Compass in 2024 highlighted several aspects of application security training programs. It revealed whether security training was mandatory or voluntary, the forms of training provided, and the presence of a security champions program.

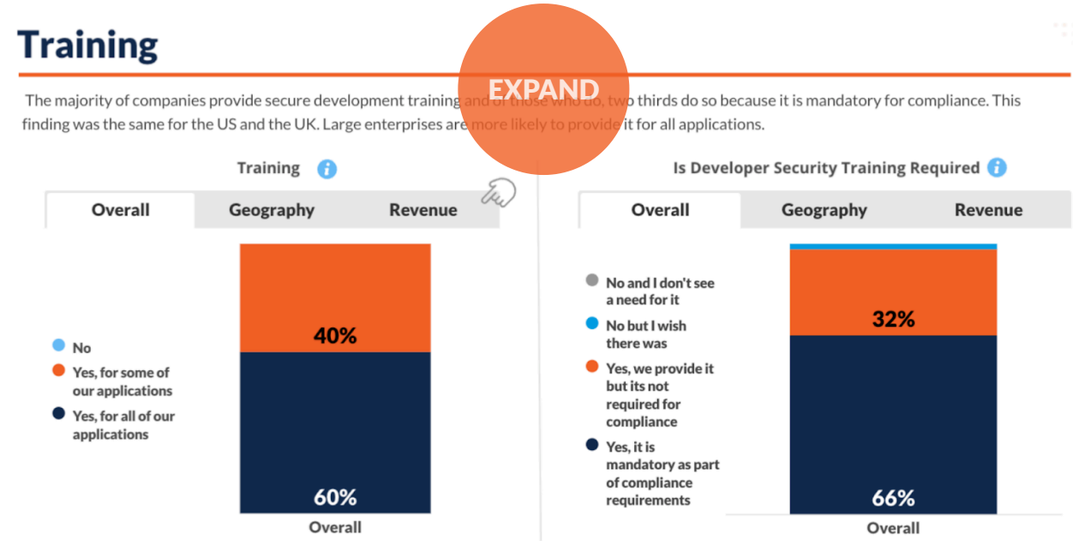


Figure 1: Large enterprises in both the US and UK are more likely to provide it for all applications.

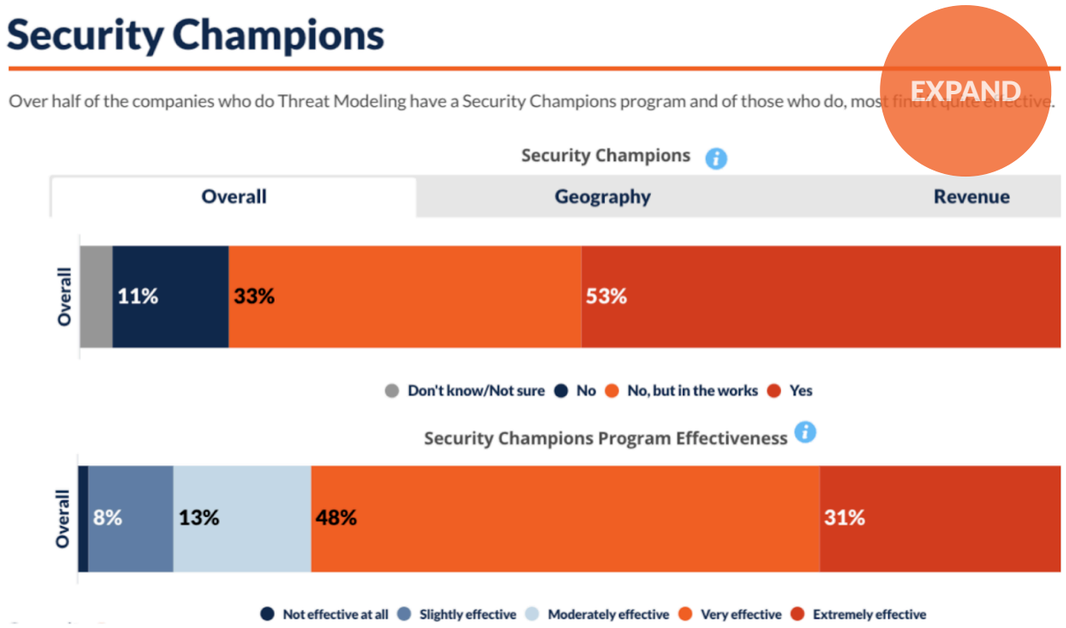


Figure 2: Survey results on the prevalence and effectiveness of security training programs and Security Champions initiatives.

Training

The majority of companies provide secure development training, and of those who do, two-thirds do so because it is mandatory for compliance. This finding was the same for the US and the UK. Large enterprises are more likely to provide it for all applications.

- **Overall:** 66% of developers require security training as part of compliance requirements, while 34% do not have such requirements.
- **Geography:** In the US, 68% of developers must take security training for compliance, whereas in the UK, the figure is lower at 60%.
- **Revenue:** Larger enterprises (with higher revenue) are less likely to mandate security training, while 3% of the respondents wish there were.

Security Champions

Over half of the companies who do Threat Modeling have a Security Champions program, and of those who do, most find it quite effective.

- **Overall:** 53% of companies have a Security Champions program, and 33% don't have but find it effective.
- **Geography:** In the US, 54% of companies have a Security Champions program, while in the UK, the figure is lower at 52%.
- **Revenue:** Companies with less than $1B in revenue are more likely to have a Security Champions program compared to companies with higher revenues. Specifically, 62% of companies with less than $1B in revenue have a Security Champions program, while 49% of companies with over $1B in revenue have such a program.

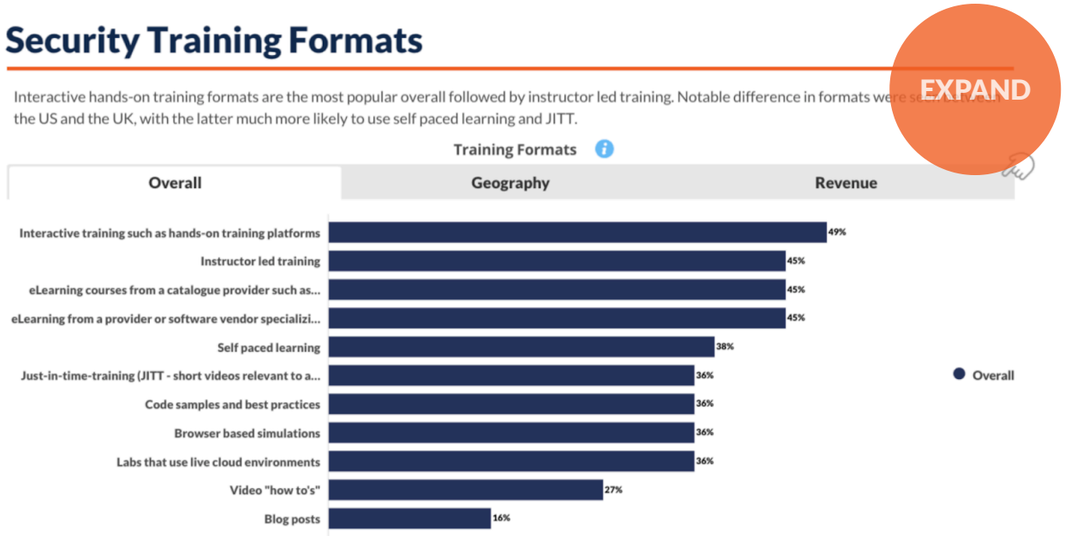


Figure 3: Preference for security training formats by geography and revenue.

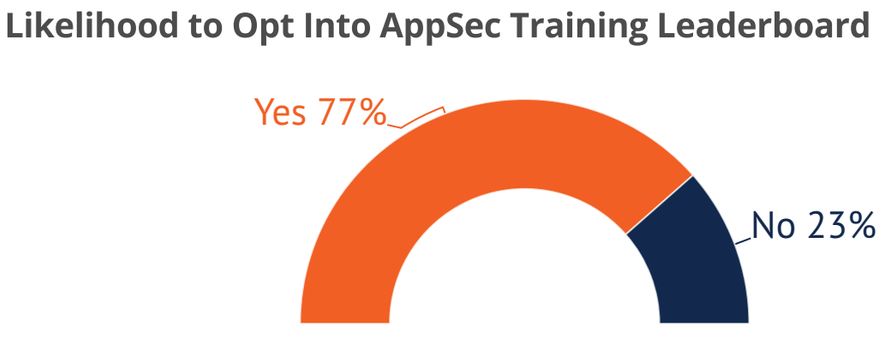
Security Training Formats

Interactive hands-on training formats are the most popular overall, followed by instructor-led training. Notable differences in formats were seen between the UK and the US, with the latter much more likely to use self-paced learning and JITT.

- **Overall:** 49% of companies prefer interactive training such as hands-on training platforms, while 45% prefer instructor-led training.
- **Geography:** The preference for interactive training is slightly higher in the UK compared to the US.
- **Revenue:** Enterprises (<$1B) tend to prefer self-paced learning and JITT, integrating short videos relevant to the application being developed.

Gamification

Gamification, such as using leaderboards to show who has completed the most training, is a popular method among development teams. It drives engagement and encourages continuous learning by introducing a competitive element to the training process.



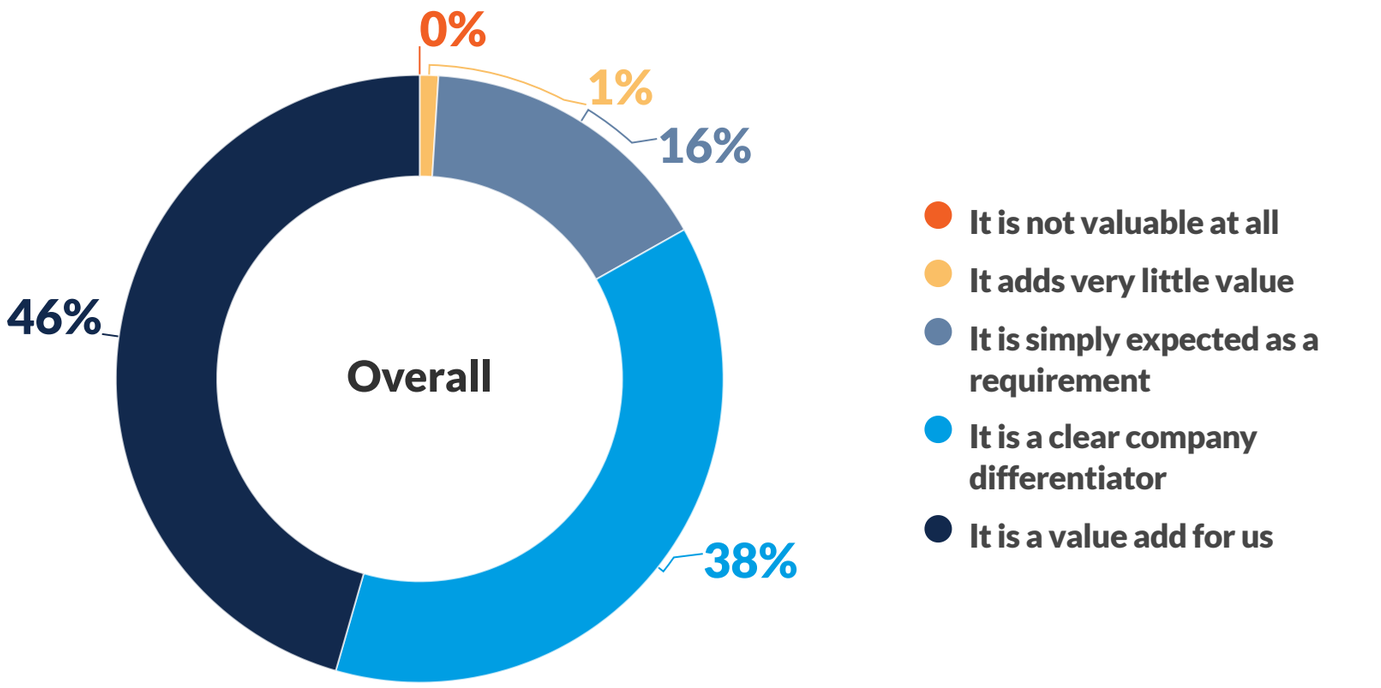
• Interactive vs. Standard Training

Interactive training methods, including hands-on labs and simulations, are known to increase engagement, retention, and the desire to learn more. However, not all training material can be covered interactively. For instance, compliance requirements may be better suited to standard training formats like lectures or reading materials. Thus, a blend of both interactive and standard training is recommended to cover all necessary topics effectively.

• Interactive vs. Standard Training

Accreditation is an important consideration when selecting a training solution. Obtaining credentials from trusted entities such as ISC2, SANS, or well-known academic institutions can be a significant business differentiator. It adds value for external stakeholders and enhances a developer's resume by demonstrating their commitment to continued learning in their field.

Value of Accreditation to External Stakeholders



Considerations for Developer Training

When planning developer training, it's essential to take a strategic approach that addresses the diverse needs of your team and aligns with your organization's overall security goals. Here are some key considerations to keep in mind:

Event-focused vs. Process-focused Training

- **Event-Focused Training**: This type of training is centered around specific events, such as workshops, boot camps, or hackathons. These events are often intensive, providing deep dives into particular topics within a short period. Event-focused training can be highly effective for initial onboarding or when introducing new security practices and tools.

- **Process-Focused Training**: This training is integrated into the ongoing development processes. It includes continuous learning opportunities, such as regular training sessions, on-the-job training, and integration of training materials into daily workflows. Process-focused training ensures that security education is a continuous effort rather than a one-time event, helping to reinforce and build upon existing knowledge over time.

Mandatory vs. Career Development

- **Mandatory Training**: This type of training is required for all developers to ensure that everyone has a baseline understanding of security principles and practices. Mandatory training is crucial for maintaining a consistent security standard across the organization and ensuring compliance with internal policies and external regulations.

- **Career Development**: Positioning training as part of career development can motivate developers to engage more deeply with the material. Offering advanced courses, certifications, and opportunities for professional growth can make security training more appealing. This approach can help develop security champions within the team who are highly knowledgeable and passionate about security.

Comprehensive vs. Targeted Training

- **Targeted Fundamental Training:** This approach focuses on specific areas of security relevant to the developers' current projects or roles. Starting with targeted fundamental training helps build a strong foundation by addressing the most crucial security challenges developers encounter in their work.

- **Comprehensive Training:** Once a solid foundation has been established with targeted training, developers, particularly those in security-critical roles, can move on to comprehensive training. This approach covers a broad range of security topics, providing a well-rounded understanding of application security. Comprehensive training ensures that developers have deep knowledge in all areas critical to their responsibilities.

Creating a Security Culture

A robust security culture must be supported by organizational buy-in that reinforces the importance of security. This includes:

- **Leadership Support**: Visible commitment from leadership to prioritize security.

- **Incentives and Recognition**: Rewarding and recognizing employees who excel in implementing security practices.

- **Continuous Improvement**: Encouraging a mindset of continuous learning and improvement in security.

- **Cross-functional Collaboration**: Fostering collaboration between development, security, and operations teams to integrate security into all stages of the development lifecycle.

Getting Started with Training

Here's how to begin your Educate step of the 3E framework:

1. **Assess Current Knowledge Levels**: Conduct a baseline assessment to understand the current security knowledge and skills within your development team.
2. **Define Training Objectives**: Based on the assessment, define clear training objectives that align with your organization's security goals. For example, ensuring all developers have a baseline level of security education, or ensuring compliance with a standard such as the Payment Card Industry's Data Security Standard (PCI DSS).
3. **Select Training Materials**: Choose appropriate training materials to meet your training objectives.
4. **Implement Training Program**: Deploy training material, ensuring they are interactive and engaging to maximize retention and application.
5. **Evaluate and Iterate**: Continuously evaluate the effectiveness of the training program through feedback and performance metrics, and make necessary adjustments to improve its impact.

If you are just getting started, consider using our free Kontra training modules. These modules provide a solid foundation for your team and help integrate security into your development processes. By considering these factors and taking a structured approach to developer training, organizations can build a knowledgeable and security-conscious development team ready to tackle the evolving challenges of application security.

Business Stakeholder Training

Educating business stakeholders about application security is equally important yet often neglected. These stakeholders play a critical role in supporting Security by Design initiatives. Fortunately, the scope of necessary training for business stakeholders is generally narrower than for development teams. This guide outlines key areas of education for business stakeholders, particularly around value drivers. Here are two free resources to assist with this training:

- **The Case for Security by Design:** This resource explains why Security by Design is essential and how it can benefit the organization.
- **Building a Bridge to Security Island:** This resource helps stakeholders understand their role in supporting security initiatives and how to effectively collaborate with development teams.

What Is “Embed” In The 3E Framework?

The 3E framework for integrating security into the software development lifecycle includes three critical phases: Educate, Embed, and Empower. The "Embed" phase is essential for integrating security practices into the daily workflows and processes of development teams.

“Embed” is the second step in the 3E Framework. Once stakeholders have received baseline education, the next step is to embed security expertise locally into development teams. This is often called a “Security Champions” program. Security Champions act as local experts and advocate for security best practices, ensuring that security is a fundamental aspect of the development process. They will eventually be the champions of adopting a Security by Design mindset and take on responsibility for activities in the Empower phase.

Security Champions

A Security Champion is a member of the development or product team who has volunteered or been nominated to act as the local expert on security for their team. This individual should have an established and trusted relationship with their team and be intimately familiar with their applications, code, and technologies.

Roles and Responsibilities

A Security Champion:

- **Acts as the “Security Conscience”**: Serves as the voice of security within the team, promoting security best practices.
- **Provides Expertise**: Acts as the go-to person for security-related matters and assists in various application security activities.
- **Drives Improvements**: Helps implement and drive security improvements within the team.
- **Integrates Security**: Ensures that security is fully integrated into the development process.
- **Bridges Teams**: Acts as a liaison between the application security team and the development team, facilitating better communication and collaboration.

Profile of a Security Champion

A Security Champion does not require prior information security or application security knowledge. Ideally, they should:

- Be a developer.
- Be self-motivated and eager to learn new ideas and technologies.
- Have a passion for security and building secure applications.
- Be dedicated to protecting the organization against security threats.
- Be interested in improving their team's security awareness and practices.
- Be influential within their team.
- Not be a manager or have decision-making power.
- Be able to commit 20-30% of their time to security activities.

Responsibilities of a Security Champion

- **Acting as the Voice of Security**: Advocating for security best practices within the team.
- **Participating in Security Discussions**: Engaging in discussions about security and bringing in application security team members when needed.
- **Documenting Security Decisions**: Making and documenting security-related decisions for the team.
- **Identifying Security Needs**: Helping identify user stories or requirements that may require security input.
- **Implementing DevSecOps Initiatives**: Supporting and implementing DevSecOps practices within the team.
- **Prioritizing Security Requirements**: Ensuring that security-related stories and requirements are prioritized in the product backlog.
- **Creating Threat Models**: Developing and documenting threat models for applications.
- **Conducting Security Code Reviews**: Reviewing high-risk code for security vulnerabilities.
- **Implementing Security Tests:** Helping implement security activities and tests within the CI/CD pipeline.
- **Triaging Security Issues**: Managing issues from static and dynamic analysis scans, penetration testing reports, and bug bounty programs.
- **Overseeing Remediation Processes**: Ensuring threats and vulnerabilities are addressed promptly.
- **Staying Updated on Security Trends**: Keeping track of modern security attacks and defenses.
- **Encouraging Team Training:** Motivating team members to complete security training and providing resources for ongoing education.
- **Engaging with the Security Community**: Participating in Security Champion program calls, forums, and events.

Time Commitment and Resourcing

Time commitments for Security Champions vary by company. In our experience, enterprise Security Champions should allocate 20-30% (1-1.5 days) of their weekly time to application security activities. Ideally, each development team should have 1-2 Security Champions to ensure redundancy and coverage.

Creating a Security Champions Program

A Security Champions Program is fundamental to the success of a mature application security program. It helps scale application security in an organization by embedding a security conscience within development teams and leveraging members of the development team to act as the team's key contact for security support.

Keys to Success

- **Motivation**: Ensure Security Champions see the value in their contribution and participation and recognize and reward their efforts. Recognition can include formal positive communication from senior management and opportunities for personal and career growth.
- **Ability:** Provide the necessary training and resources to enable Security Champions to perform their roles competently and confidently. This includes computer-based or instructor-led training, workshops, mentorship, books, and conference attendance.
- **Management Support:** Secure dedicated support from the Security Champion's team and its management. Without buy-in, development tasks may be prioritized over security activities, risking the program's success.

Core Activities

A Security Champions Program can be broken down into four core activities: Recruitment, Training, Engagement and Maintaining Interest, and Program Management. These activities help in building, organizing, and maintaining a Security Champions Program and network.

Figure 1: Core Security Champions program activities

1 Recruitment

Recruitment involves employing multiple strategies to solicit volunteers and identify people interested in participating in the Security Champions program. The primary activity required to establish and grow a Security Champions program is recruitment.

Active Recruitment:

- Performing outreach and advertising of the Security Champions Program.
- Emailing development team executives or senior leadership to nominate Security Champions after the initial advertising of the formal launch.
- Including updates on the Security Champions Program in internal newsletters, with a call to join the program if a development team does not already have a Security Champion.
- Including the Security Champions program in presentations about application security.
- Hosting regular Information Security events focused on application security and inviting development teams.

Passive Recruitment:

- Minimal advertising and effort, relying on word-of-mouth and existing networks to attract participants.

Volunteer vs. Nominations:

- Seeking volunteers for the role ensures motivated and engaged participants. If no volunteers are available, seek nominations from team leads, managers, or peers, but confirm their interest to ensure they are willing participants.

2 Training

As part of participating in the Security Champions program, Security Champions help scale, execute, and support various aspects of Application Security processes. Training helps empower Security Champions and provides them with the ability to support Application Security and the overall security culture within the organization and their respective development teams.

Training may be in the form of shadowing and mentorship, instructor-led training, computer-based training, access to online platforms, or conference and community talks or workshops.

3 Engagement and Maintaining Interest

Ensuring Security Champions remain interested and engaged is crucial to continuous success. Priorities related to their core roles and responsibilities are likely to take precedence over Application Security activities. Therefore, incentivizing Security Champions helps increase the likelihood of continued participation.

Activities associated with Engagement and Maintaining Interest may include organized events (e.g., Capture-the-Flag and Lunch and Learns), an internal security newsletter, growth opportunities (e.g., mentorship, conference attendance, and industry training attendance), recognition (e.g., letter of appreciation and recognition from management), or rewards (e.g., gift cards, t-shirts, and stickers).

4 Program Management

Program Management captures all related activities required to manage and operate a Security Champions program that does not fall into any of the other three categories.

The main objectives of Program Management are to:

- Ensure the program is working and meeting its objectives.
- Ensure Security Champions are engaged and still participating.
- Report on the success of the program.

Incentives and Rewards

Incentives and rewards are crucial for motivating participation. Rewards should be varied and meaningful, aligning with individual values.

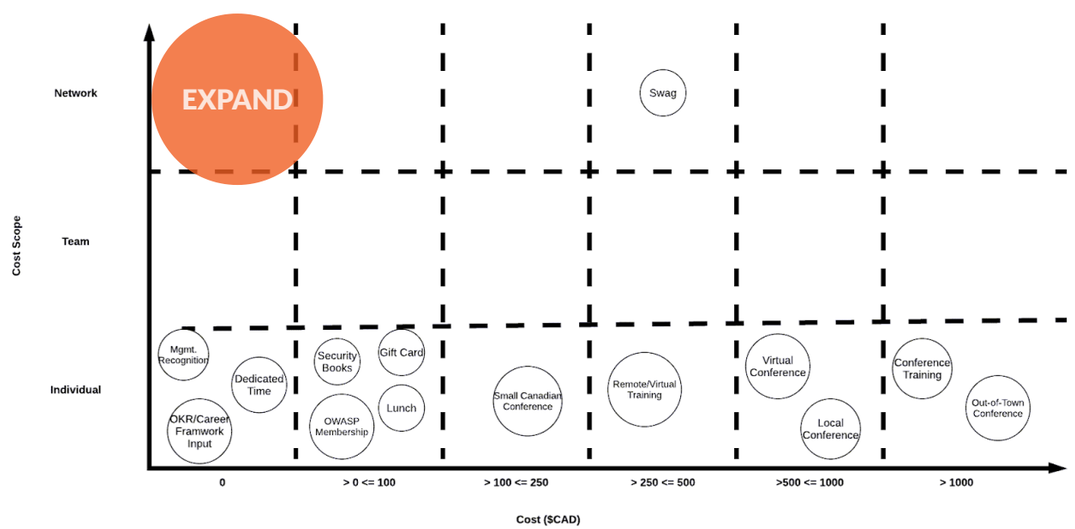


Figure 2: Diagram highlighting potential incentives and rewards, estimated cost ranges, and Security Champions coverage

Examples include

- **Management Recognition:** It's vital to the program for Security Champion work to be explicitly valued and recognized by management as a critical component of a Champion's work responsibilities.
- **Dedicated Time:** The work required of a Security Champion should be recognized as a separate component of the Champion's day-to-day role, and adequate time should be budgeted solely for that specific role.
- **OKR/Career Framework Input:** Along these same lines, Security Champions' work should be explicitly called out in the form of explicit Objectives and Key Results (OKRs). These OKRs should be weighted to provide an exclusive benefit for achievement over team developers who do not take on this additional role.
- **Security Books:** Building good security knowledge and staying on top of emerging security trends is key to the Champions role. Consider allocating a budget for Champions to purchase relevant books to continually stay abreast of these trends.
- **OWASP Membership:** Continuing security education is key for the Security Champions role. Providing memberships in security focused organizations like OWASP can help further that goal.
- **Gift Cards/Lunch:** The Security Champion role is non-trivial and is an additional work commitment to a Champion's normal day-to-day role. Providing them with periodic gift cards can be an appreciated incentive or reward for taking on and executing that role.
- **Conferences/Conference Training:** Similarly, allocating time and budget for security conferences can help further that goal. A conference can take on many different forms. It doesn't always have to be a large, out-of-town conference. Smaller, local, focused, short duration security conferences can be equally valuable. Some conference examples would be:
  - Small local conferences
  - Virtual conferences
- **Remote/Virtual Training:** Providing exclusive or early access to remote/virtual security training can also be an effective way to incentivize and reward Security Champions.
- **Swag:** Swag items (e.g. mugs, cups, hoodies, etc.) can be an effective and fun way to reward all Security Champions and help identify and highlight the Security Champion role within the company and among the teams that they are embedded in.

What Is “Empower” In The 3E Framework?

The "Empower" phase is the final and most comprehensive step in the 3E framework, focusing on empowering development teams with the necessary tools, processes, and knowledge to integrate security seamlessly into the software development lifecycle without the active involvement of security experts.

Following the foundational steps of "Educate" and "Embed," this phase emphasizes effectively implementing and managing security requirements and threat modeling. By empowering teams, organizations can ensure that security is not just an added layer but a core component of their development process. Security champions, as defined in the “Embed” phase, serve as ideal owners of the activities described below.

Threat Modeling

Threat modeling is a crucial process for integrating Security by Design. It was first popularized by Microsoft through their Secure Development Lifecycle and has since gained widespread recognition. Threat modeling helps identify, assess, and mitigate security risks early in the software development lifecycle.

Threat Modeling Considerations

Threat modeling typically seeks to answer four fundamental questions:

1. What are we working on?
2. What can go wrong?
3. What are we going to do about it?
4. Did we do a good job?

Popular methodologies like **STRIDE** (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) guide development teams in identifying threats. However, the effectiveness of manual threat modeling is often limited by the security knowledge of the practitioners involved.

In our experience, manual threat modeling processes typically take 40 - 120 hours to complete and require participation from multiple stakeholders. These time commitments can lead to significant pushback and derail the entire Security by Design program. Therefore, we highly recommend automating threats to streamline the process and maintain development agility. Organizations that use athreat model consistently and at scale can realize all of the four value drivers of security by design.

Security Requirements Generation

The outcome of threat modeling is a set of countermeasures or controls, which become security requirements. Unfortunately, many organizations skip this crucial step, resulting in unaddressed vulnerabilities. It is essential to ensure that security requirements are actionable and integrated into the development process.

Criteria for Effective Security Requirements

For security requirements to be effective, they must be:

- **Clear and Concise**: Easily understood and implemented.
- **Relevant:** Directly applicable to the specific context of the application.
- **Actionable:** Specific steps that can be taken to mitigate identified risks.
- **Testable:** Capable of being validated through testing.
- **Explained:** Providing context on the underlying threat and its mitigation.
- **Compliant:** Aligning with relevant compliance obligations.

In our experience, security requirements are more likely to be addressed if they are tracked inside of a product development team already using such as JIRA. Using a system to track security requirements also helps with creating audit trails for regulatory compliance.

Static Requirements Lists

While static lists like the OWASP Top 10 or NIST 800-53 provide a good starting point, they are often too generic and not directly actionable. Security requirements should be tailored to each specific application to ensure they are relevant and effective.

Secure Coding Guidelines

Secure coding guidelines are another common approach, but they often suffer from being too broad and not traceable. Instead, integrating language-specific implementation guidelines into security requirements can provide more practical and actionable guidance for developers.

Process Change

Implementing Security by Design often requires significant changes to existing development processes. Successful rollouts necessitate clear, specific mapping of how the existing process will change and who will be responsible for each step.

1. **Identify Assets:** Determine the valuable assets in the application.
2. **Identify Threats:** Use threat modeling methodologies to identify potential threats.
3. **Determine Mitigations:** Develop countermeasures to mitigate identified threats.
4. **Implement Security Requirements:** Integrate these countermeasures into the development process as security requirements.
5. **Validate Mitigations:** Test the implemented security measures to ensure they effectively mitigate the threats, for example, by using Static Analysis Security Testing (SAST) tools.
6. **Review and Iterate:** Continuously review and improve the threat modeling process and security requirements based on feedback and new threats.

The following is an example of a process flow diagram illustrating the roles and responsibilities in a threat modeling platform rollout:

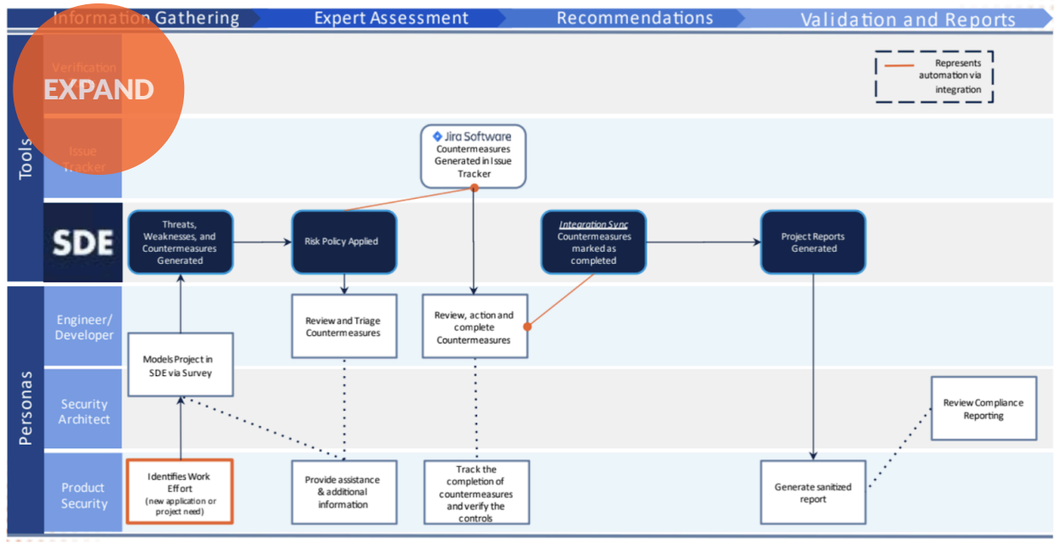


Figure 1: Process flow diagram outlining responsibilities in a threat modeling platform rollout.

Detailed Steps for Implementation

1. **Educate:** Ensure all team members understand the basics of security threats and the importance of threat modeling. Provide training sessions and resources.
2. **Tool Selection:** Choose the right tools for automated threat modeling that fit your development workflow.
3. **Pilot Program:** Start with a small, manageable project to pilot the threat modeling process and security requirement generation.
4. **Feedback Loop:** Gather feedback from the pilot program to refine the process and address any challenges encountered.
5. **Rollout:** Gradually extend the process to larger projects and additional teams, ensuring continuous support and adjustment based on feedback.
6. **Monitoring and Reporting:** Establish a robust monitoring and reporting mechanism to track the effectiveness of implemented security measures and make data-driven decisions for improvements.

Automation in Threat Modeling

Automating threat modeling helps eliminate bottlenecks and enhances the agility of the development process. Developer-centric threat modeling tools provide a more efficient and scalable approach compared to manual methods. These tools integrate seamlessly into the development workflow, enabling continuous and consistent threat analysis.

For a comprehensive understanding, refer to this "whitepaper which outlines the advantages of developer-centric threat modeling over traditional methods.

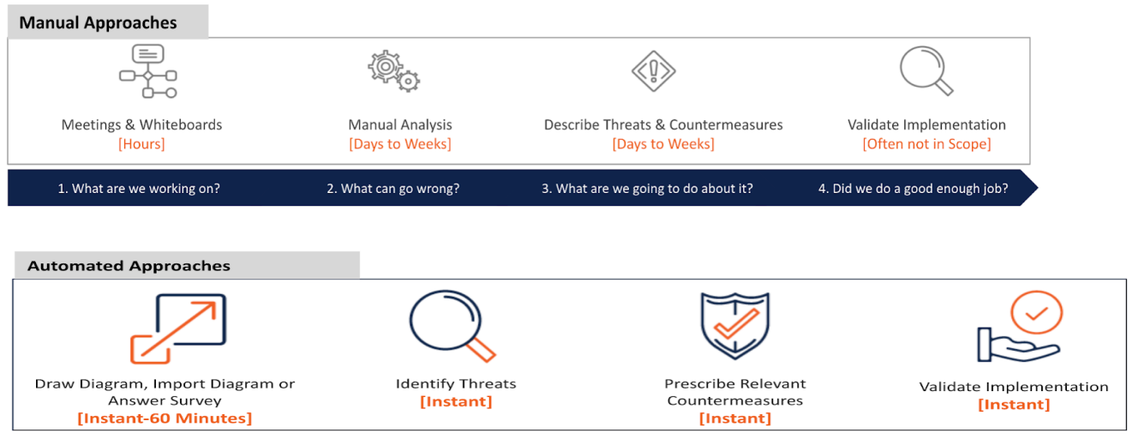


Figure 2: Comparison of manual and automated approaches in threat modeling (visit: https://www.securitycompass.com/threat-modeling/free-threat-modeling-course/)

Conclusion

Security by Design is a foundational approach to building secure software. By embedding security considerations into the earliest stages of the development process, organizations can create more robust, compliant, and high-quality applications.

While it complements aspects of DevSecOps and Application Security, its unique focus on the initial phases of development sets it apart as a critical component of a comprehensive security strategy.

Practitioners often make the mistake of assuming that Security by Design is obviously beneficial, failing to understand that it requires cultural change. Most people understand security as a quality you test for, rather than something you build in. If you are considering implementing Security by Design, we suggest building a business case to facilitate process change.

By understanding and implementing Security by Design, organizations can proactively safeguard their software against the ever-increasing threat landscape, ensuring a secure and resilient digital future. This approach not only enhances software security but also contributes to operational efficiency, regulatory compliance, and overall business success.

Conclusion

Adopting Security by Design is not just about improving technical security measures; it's about driving significant business benefits. By reducing operational costs, mitigating risks, enhancing software security at scale, and enabling revenue growth through compliance, Security by Design offers a comprehensive approach to secure software development.

Understanding these value drivers and effectively communicating them to all stakeholders is essential for securing buy-in and ensuring the successful implementation of Security by Design initiatives. As cybersecurity threats continue to evolve, embedding security from the ground up will be crucial for building resilient, secure, and compliant software systems.

Security by Design is not just a technical initiative but a strategic business imperative. By reducing operational costs, mitigating risks, improving software security at scale, and enabling revenue growth through compliance, Security by Design offers comprehensive benefits that resonate with both technical and non-technical stakeholders. Building a compelling business case with quantifiable benefits is essential for securing executive buy-in and ensuring the successful implementation of Security by Design initiatives. As cybersecurity threats continue to evolve, adopting a proactive approach will be crucial for building resilient, secure, and compliant software systems.

Building a comprehensive program plan for Security by Design is a strategic and detailed process. It involves setting clear goals, selecting the right metrics, engaging stakeholders, creating a phased project plan, and implementing an effective communication strategy. By following these steps, organizations can ensure the successful adoption and implementation of Security by Design, ultimately leading to more secure, resilient, and compliant software systems. As cybersecurity threats continue to evolve, a proactive approach to security is essential for safeguarding digital assets and maintaining a competitive edge.

Conclusion

The **"Educate"** phase in the 3E framework is pivotal for embedding security into the software development lifecycle. By providing targeted, relevant training to both development teams and business stakeholders, organizations can cultivate a culture of security awareness and competence. This foundational step ensures that all participants are prepared to implement secure practices, ultimately leading to more resilient and secure software systems. As cybersecurity threats continue to evolve, investing in education and training remains a critical component of any comprehensive security strategy.

The **"Embed"** phase of the 3E framework is essential for integrating security into the software development lifecycle. By establishing a robust Security Champions Program, organizations can ensure that security practices are embedded within development teams, fostering a culture of security awareness and competence. This approach not only enhances the security of applications but also bridges the gap between development and security teams, leading to more resilient and secure software systems. The success of the Embed phase hinges on motivation, ability, and management support, making it crucial for organizations to invest in these areas to build a strong and effective security culture.

The **"Empower"** phase of the 3E framework is crucial for ensuring that security is deeply integrated into the software development lifecycle. By effectively implementing threat modeling and generating actionable security requirements, organizations can build more secure and resilient software. Empowering development teams with the right tools, processes, and knowledge ensures that security is not just an afterthought but a fundamental aspect of development. This comprehensive approach to security helps organizations stay ahead of potential threats and build trust with their users and stakeholders.

By following the steps outlined in this phase, organizations can ensure that security is an integral part of their development process, leading to more robust and secure software systems. The "Empower" phase, with its focus on automation, effective security requirement generation, and continuous improvement, is the key to sustaining a strong security posture in the dynamic software development landscape.