

IMPROVING COLLABORATION BETWEEN DEVELOPMENT AND SECURITY TEAMS

Module Overview:

In software development, the success of DevSecOps hinges on effective collaboration between Development (Dev) and Security (Sec) teams. Traditionally, these teams have operated in silos, leading to security challenges and production delays. This module explores strategies for breaking down these barriers, fostering a shared responsibility model, and ensuring security becomes an integrated part of the development process without hindering innovation or productivity.



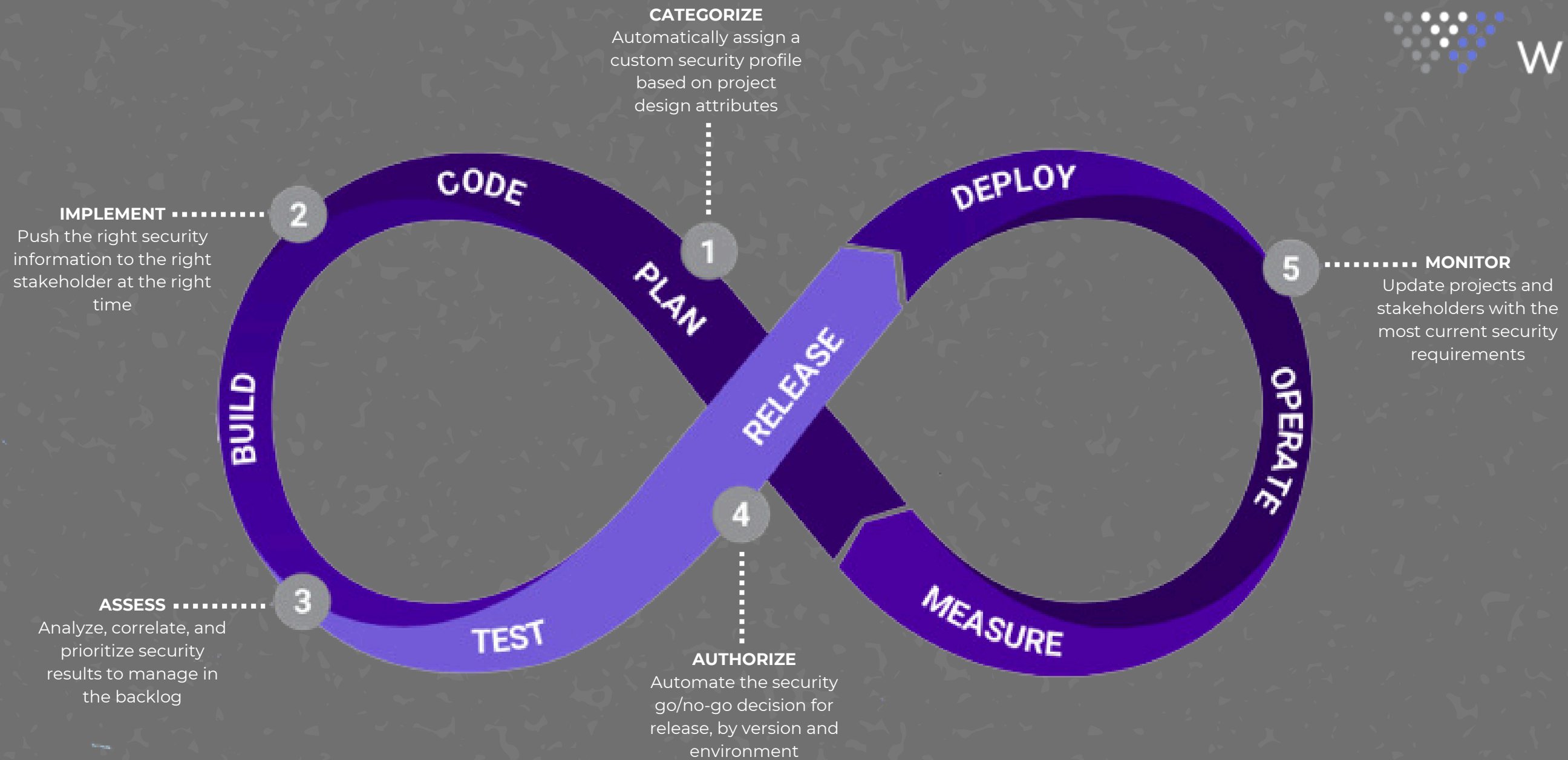
APPLICATION SECURITY IN THE CI/CD PIPELINE



Objectives:

Integrate security checks within the CI/CD pipeline to detect vulnerabilities before they enter production.

Explore how DevSecOps enables seamless collaboration between security and development teams, enhancing security without slowing down the pipeline.



Embedding Security Into Every Step

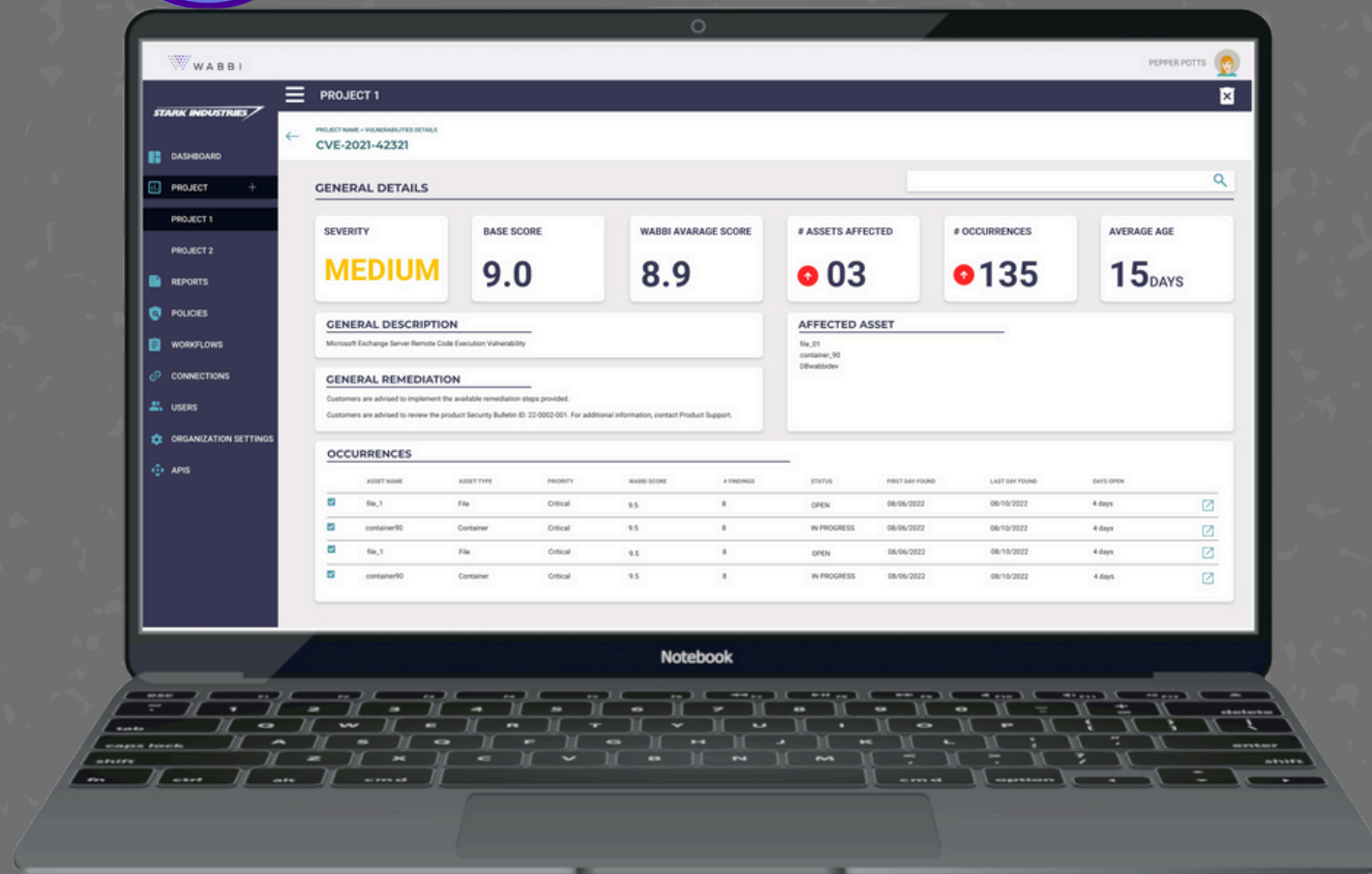
The goal of embedding security into your CI/CD pipeline is to ensure vulnerabilities are caught and addressed before applications go live. This reduces security risks while maintaining the pace of continuous integration and continuous delivery.

Best Practice

Ensure your pipeline includes both SAST and DAST tests, with results sent back to development teams in real-time.

Key Considerations:

- **Continuous Testing:** Use automated testing tools that provide immediate feedback during code integration, reducing the time between vulnerability discovery and mitigation.
- **Shift-Left Security:** Start vulnerability detection early in the development process—beginning with the first code commit—so vulnerabilities are caught as soon as possible.
- **Security Automation:** Integrate security testing seamlessly so developers are notified of vulnerabilities within their development environment without disrupting their workflow.





ONLY
32%

of companies integrate
security from the
beginning despite **97%**
believing it is important

Wabbi

IMPROVING COLLABORATION BETWEEN DEVELOPMENT AND SECURITY TEAMS



Objectives:

Foster a Culture of Shared Responsibility: Enable development, security, and operations teams to collaborate effectively using DevSecOps principles to ensure seamless integration of security practices.

Improve Communication and Tool Integration: Develop strategies to enhance communication and align security measures with development goals, while leveraging DevSecOps tools that enable smoother collaboration.

BREAKING DOWN SILOS IN DEVSECOPS

The traditional development model treated security as a checkpoint toward the end of the process, often causing delays as security vulnerabilities were uncovered too late. DevSecOps seeks to bridge this gap by embedding security into every stage of the software development lifecycle (SDLC). To do this, organizations need to dismantle the silos between development, security, and operations teams, fostering a culture of shared responsibility where each team contributes to securing applications.





Key Concepts:

- **Shared Responsibility Model:** In DevSecOps, development, operations, and security teams share ownership of security from the beginning. This means security concerns are addressed early in the process, rather than at the end, reducing bottlenecks and ensuring faster release cycles.
- **Collaboration and Transparency:** Open communication between Dev and Sec teams is crucial to preventing misunderstandings or missed security flaws. Each team needs to understand the priorities of the other, and how their actions impact the security posture of the product.

Objective 1: Understand how siloed teams create bottlenecks and vulnerabilities in the traditional development process.

Objective 2: Explore how a shared responsibility model can address these challenges and improve overall security.



BUILDING A DEVSECOPS CULTURE

For DevSecOps to succeed, organizations need to create a culture where security is an integral part of development, rather than an external process. Security shouldn't be seen as a hindrance but as a necessary function that everyone has a role in supporting. A DevSecOps culture emphasizes accountability, awareness, and continuous improvement.





Strategies for Building a DevSecOps Culture

- **Shared Security Ownership:** Development teams must take responsibility for managing vulnerabilities within their code. This includes performing regular scans, fixing vulnerabilities promptly, and ensuring that security measures are factored into the design and planning stages.
- **Training and Awareness:** Developers should receive continuous training on secure coding practices, the latest vulnerabilities, and risk management strategies. By ensuring developers are well-versed in security, they can address potential flaws early on.

Objective 1: Foster shared security ownership among all teams by embedding security into the SDLC.

Objective 2: Emphasize the importance of continuous training and awareness to keep all teams updated on the latest security practices.



STRATEGIES TO FOSTER COLLABORATION

Collaboration is key to the success of DevSecOps. Without proper tools and processes, communication between development and security teams can still be fragmented, even under a DevSecOps model. Organizations need to implement solutions that facilitate this collaboration in practical ways.





Practical Collaboration Methods

- **Use of Developer-Friendly Security Tools:** The security tools used within the DevOps pipeline must be intuitive for developers. These tools should offer quick feedback and fit seamlessly into the development process, allowing security checks to happen without disrupting the development flow.
- **Regular Security Audits:** Joint security reviews should be a regular part of the development process. These sessions can include reviewing vulnerabilities, remediation strategies, and improving communication between teams. Such audits keep both teams aligned and help identify gaps in the current security practices.

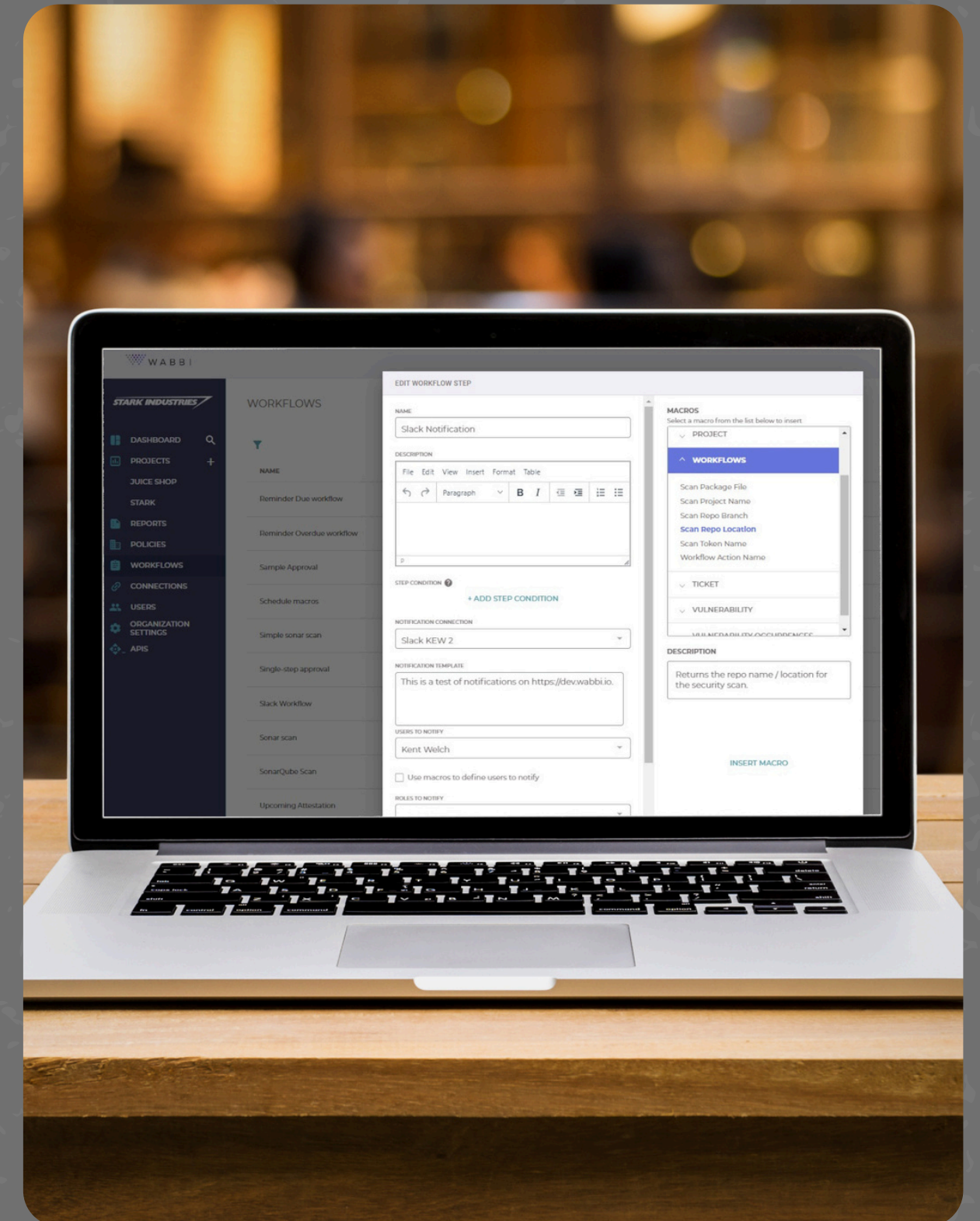
Objective 1: Learn how to integrate developer-friendly security tools into the development pipeline to encourage adoption and ease of use.

Objective 2: Explore the benefits of regular security audits and how they can improve collaboration between teams.



LEVERAGING AUTOMATION FOR SEAMLESS COLLABORATION

Automation is at the heart of DevSecOps. By automating security testing and integrating it into the CI/CD pipeline, development teams can address vulnerabilities quickly, without waiting for manual reviews. Automation reduces the workload on security teams and ensures that developers get instant feedback on security flaws during development.



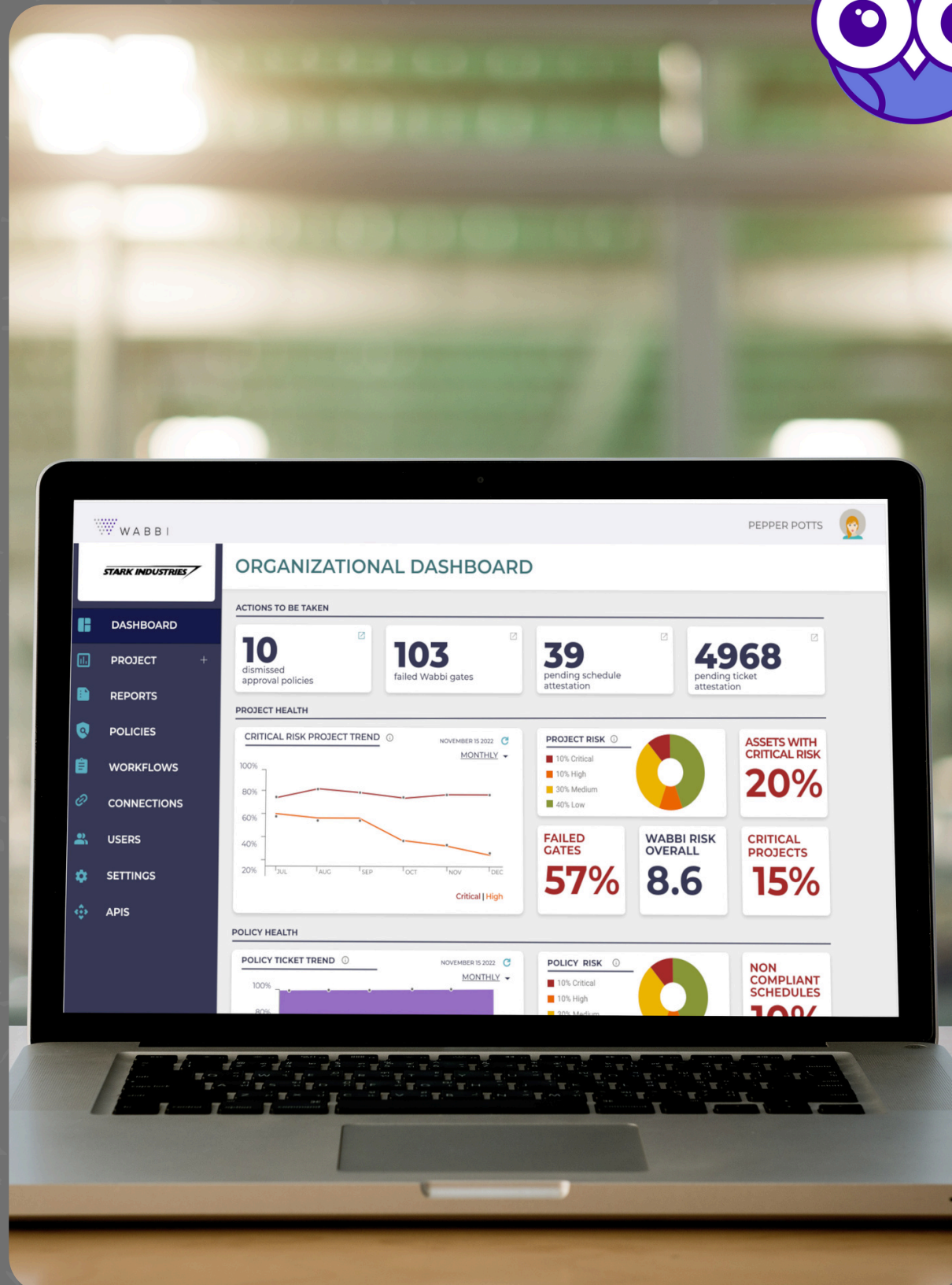


Automating Security Processes

- **Automated Security Testing:** Security tests, including static application security testing (SAST) and dynamic application security testing (DAST), can be automatically triggered within the CI/CD pipeline, providing real-time results to developers as they code.
- **Continuous Monitoring:** Automated tools can continuously monitor the development environment, ensuring that any new vulnerabilities are detected as soon as they arise. Continuous monitoring also provides insights into how well security measures are working over time.

Objective 1: Understand the role of automation in streamlining security checks during development and reducing friction between Dev and Sec teams.

Objective 2: Explore continuous monitoring and its role in maintaining a secure development environment through real-time feedback.



ACTIVITY



Phone a Friend!
Call somebody in a
Dev or Sec team to
brainstorm how to
collaborate.

**Take a look at your
SDLC. Write out the
points where your
organization
integrates security.**



CONCLUSION

Building a DevSecOps culture and improving collaboration between development and security teams are essential for modern software development. By breaking down silos, fostering shared responsibility, and leveraging automation, organizations can deliver secure, high-quality products faster. Encouraging collaboration is not only about tools but also about mindset—security and development teams must work together to achieve a common goal: secure, efficient, and reliable software.



RESOURCES



Understand how DevSecOps impacts different roles



AppSec: Grim Reaper or Archangel

The Future of DevSecOps with AppSec

Understand CVSS

Visit the National Vulnerability Database and read some of the latest vulnerabilities



Stats about integrating security in the SDLC



Wabi-Sabi your SecDevOps

Understand CVE