Shifting Left:
5 steps to securing your SDLC

21st September 2021

DevSecOps Specialist Group

WhiteSource

bcs The Chartered Institute for IT

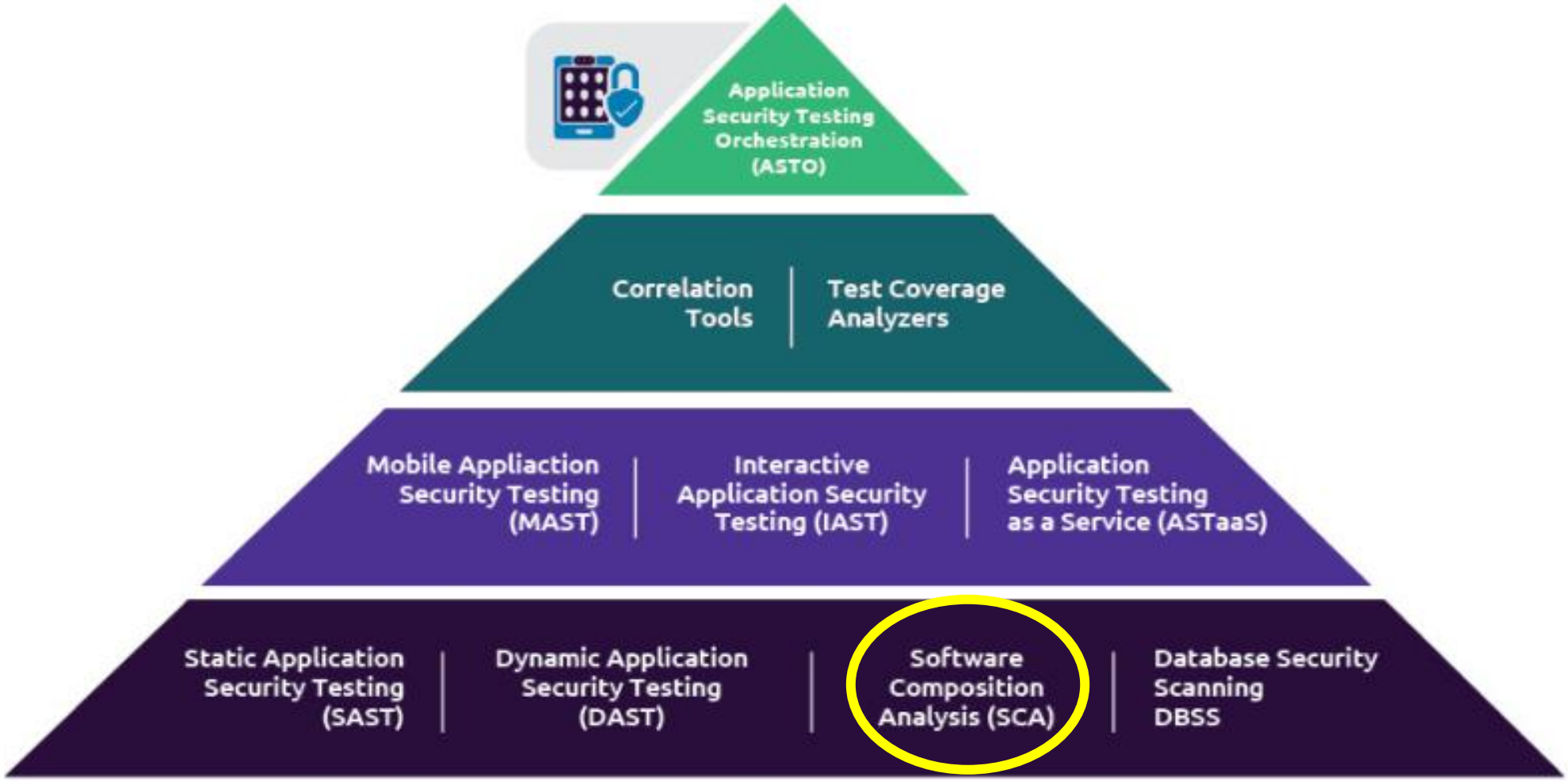
Introduction - Luke Brogan

- Currently working as a Senior Solutions Engineer at WhiteSource
- 11 years in IT
- Worked with BMW Group, Nominet, Arrival and BeyondTrust
- Experience with Application Management, Infrastructure, DevOps, DevSecOps and Organization Technology
- Wife Gina and son Joshua
- Loves everything with four wheels!
- Tries to keep fit running and cycling

Agenda

- What is Software Composition Analysis?
- The Challenge…
- Shifting Left: 5 steps
- Who are WhiteSource?
- Demonstration
- Q&A

Application Security Testing Tools Pyramid



Reference: capgemini.com/2021/04/false-positives-in-web-application-security-take-up-the-challenge/

What is Software Composition Analysis?

- Software Composition Analysis (SCA) is a segment of the application security testing (AST) toolset
- Designed to perform automated scans of an applications code base, including related artifacts
- Detects and identifies all open-source components (BoM), there license compliance data, and any security vulnerabilities
- Some tools also automate the remediation of vulnerabilities

Reference: whitesourcesoftware.com/resources/blog/software-composition-analysis/

The OSS Challenge… in securing your application

- What open-source libraries does our solution use? – The product Manager
- Are these libraries all up to date? – The Security Analyst
- What vulnerabilities are actually effective? - The Developer
- Are we using multiple versions of the same library? – The confused DevOps Engineer
- Are we using commercial friendly licenses? – The Compliance Manager

The impact of Supply Chain Attacks in unprecedented

Microsoft Uber

(2021) Dependency confusion impacted leading tech companies

mimecast

(2021) impacted about 10% of Mimecast customers

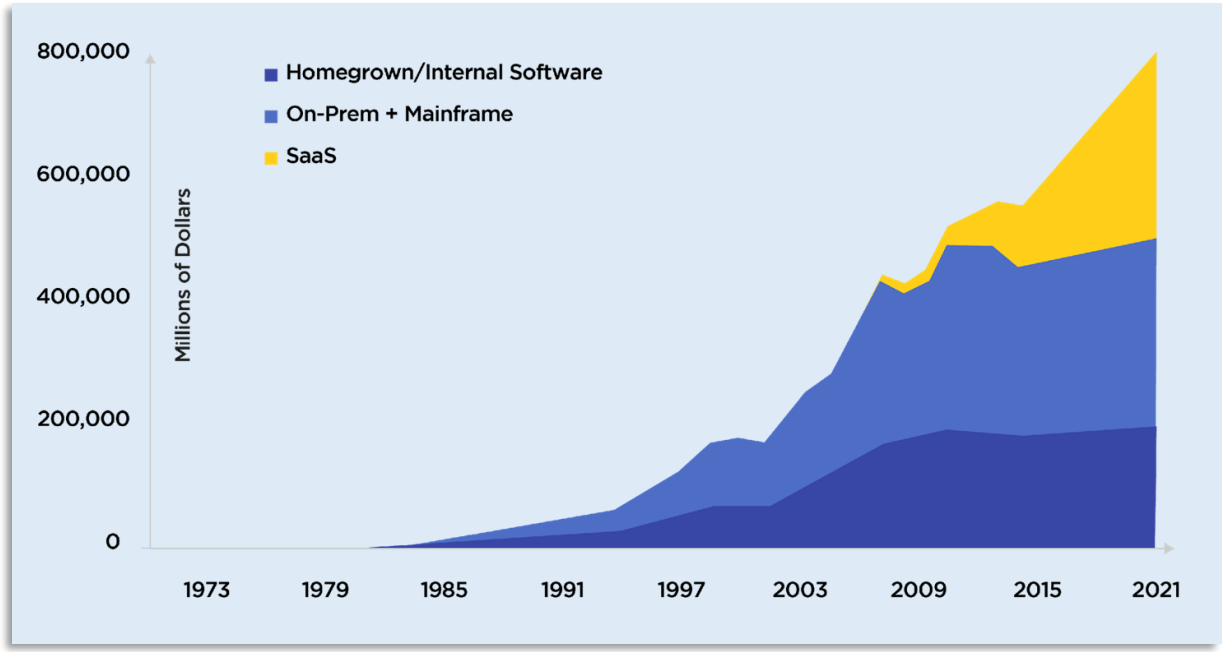
solarwinds

(2020) impacted 18 million customers of SolarWinds

The potential damage of a supply chain attack can be extremely severe: from the consequences of affected application traffic, to abusing access permissions to sensitive systems and data, and leveraging access rights of the trusted software.

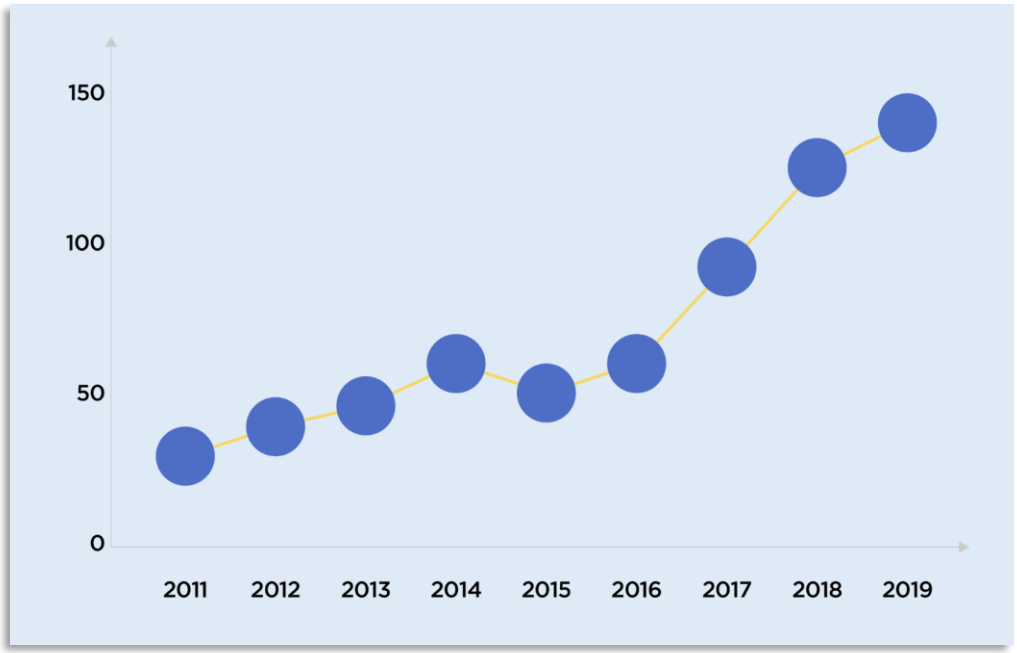
The number of reported breaches with significant impact is rising

As digital transformation and the dev-tool revolution accelerate the pace of software delivery…



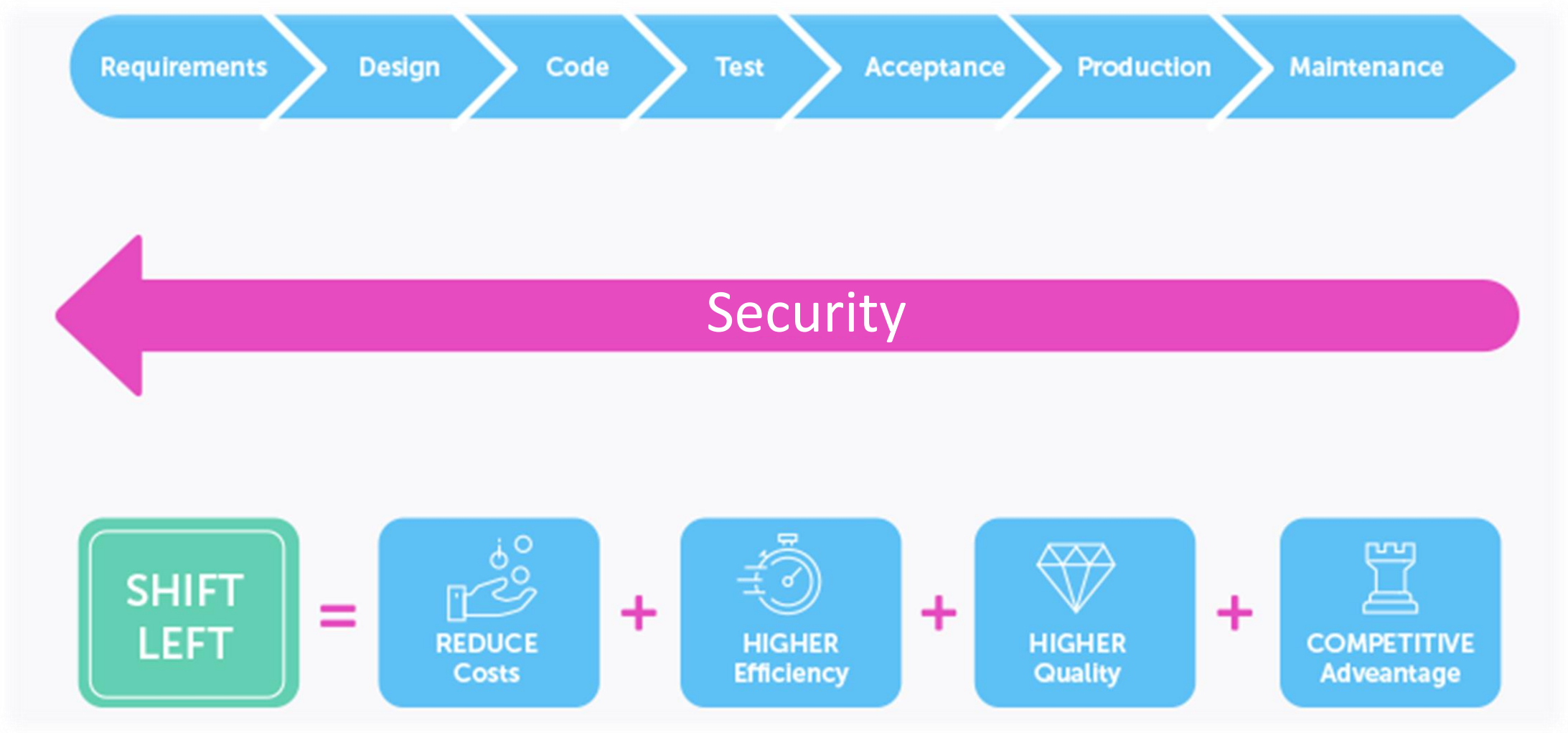
"Software 2019", Battery, 2019

…organizations are increasingly exposed



"Trends in Cybersecurity Breach Disclosures", Audit Analytics, 2020

Shifting left



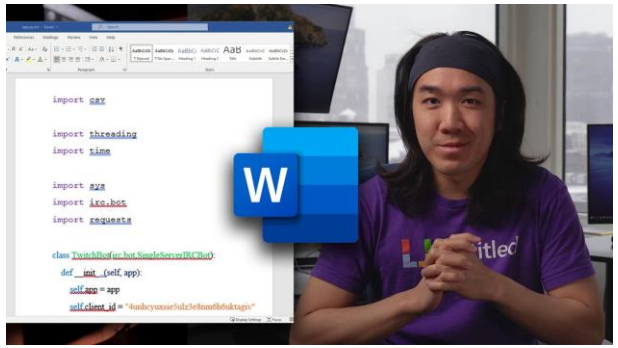
The 5 steps

Shifting Left – Step 1

- **Protect the R&D phase**
- The furthest ‘Left’ we can go for Application Security
- Easiest to fix, with the lowest cost
- Scenario: a sprint that requires new functionality… there is a library for that!
- Developers should be protected with real time vulnerabilities and license violations flagged in the browser
- It should ask the following questions:
  Is this library safe? / Is there a newer version?

Shifting Left – Step 2

- **Protect the IDE**
- Developers doing what they do best… creating beautiful applications
- The solution should support all major languages, package managers and IDEs
- Remediation and threat analysis should take place in the IDE in real time
- Mitigates risk around C+P of a vulnerable library, or one which violates a license agreement

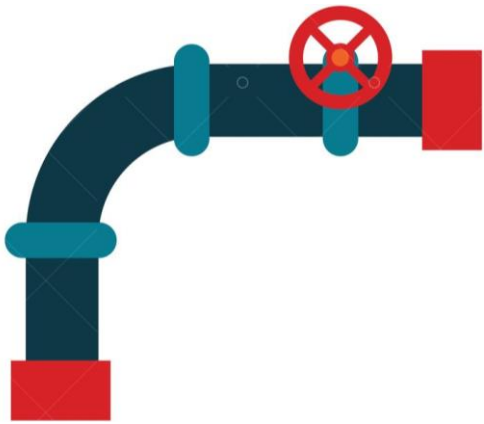


Shifting Left – Step 3

- **Protect the Repository**
- When we commit code to the repository, the code should be scanned to detect vulnerabilities and licenses of open source libraries
- Issues should be automatically raised and logged for review
- Automatic remediation should take place with Pull Requests, with merge confidence levels
- The solution should support all major repositories for flexibility

Shifting Left – Step 4

- **Protect the Pipelines**
- Each pipeline for each environment should be protected with SCA scanning and reporting
- Early detection of policy violations to break the build
- Reports should be accessible from the CI/CD tool
- All major CI/CD vendors should be supported for flexibility
- Scanning should not cause a delay in your pipeline



Shifting Left – Step 5

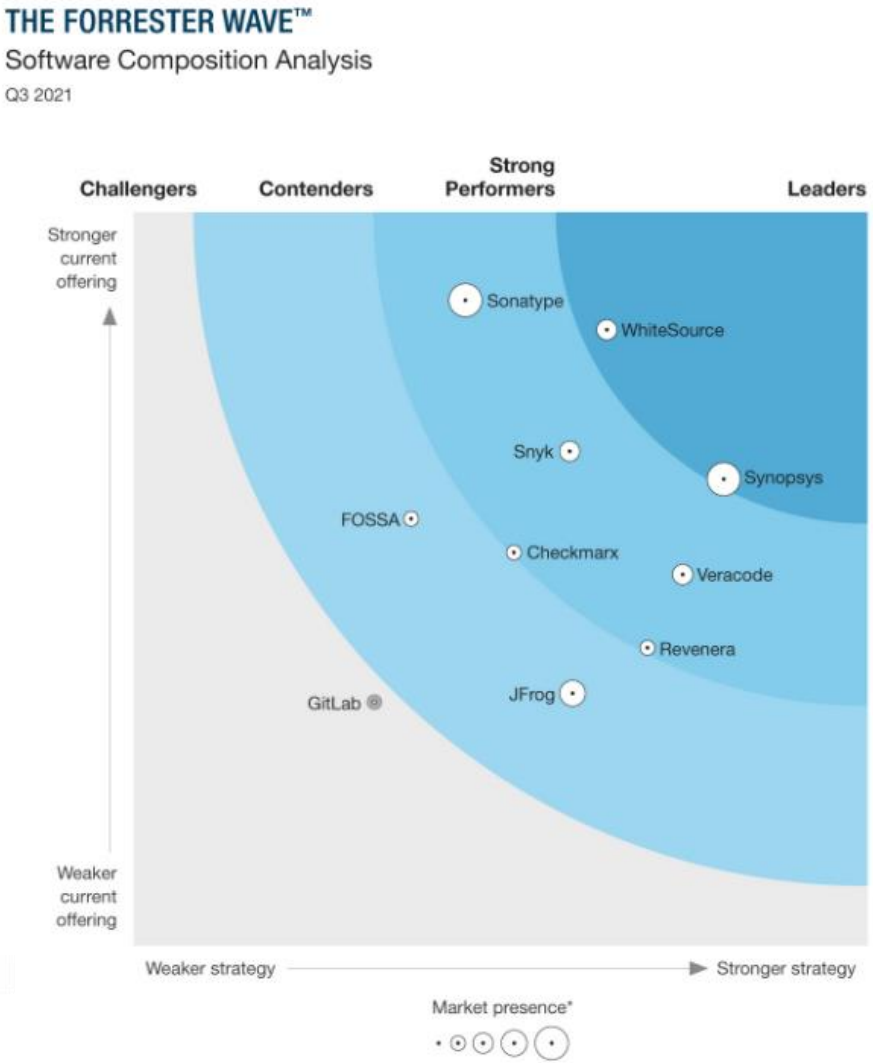
- **Protect the Containers**
- The furthest step to the right, it is much more expensive to fix here
- However, the solution should cover the full SDLC cycle, so it must be supported
- The solution should be able to scan running containers and Kubernetes clusters alike for vulnerabilities and license information

Who are WhiteSource?

WhiteSource

Who are WhiteSource?

Why WhiteSource?



Accelerating secure product delivery at scale

Industry-leading Knowledge Base

- Broadest language coverage / Highly accurate
- Actionable crowdsourced merge confidence from 300K repos

Automatically remediate issues

- Patented reachability analysis ensures focus on high impact issues
- Actionable results and automated fixes in the developer's native environment

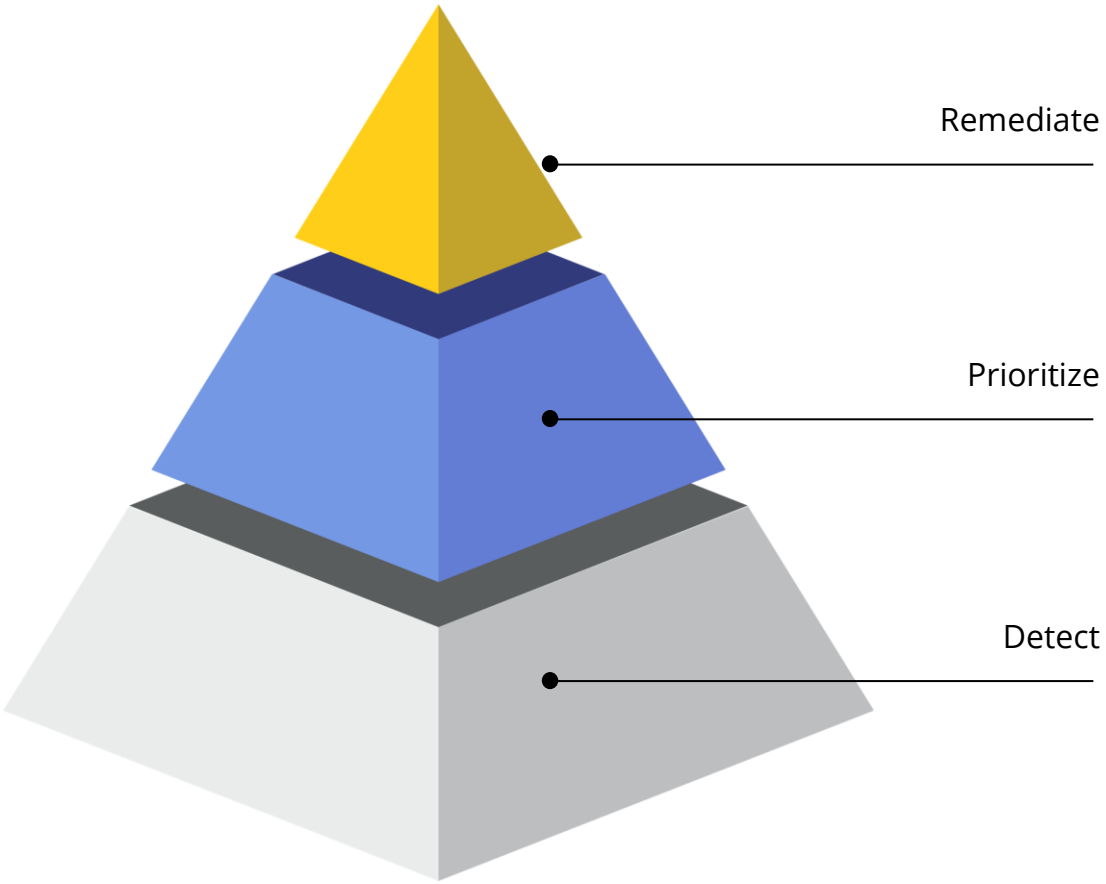
Prevent issues from entering your environment

- Automatically maintain dependency health
- Protection from software supply chain attacks

Visibility throughout the SDLC

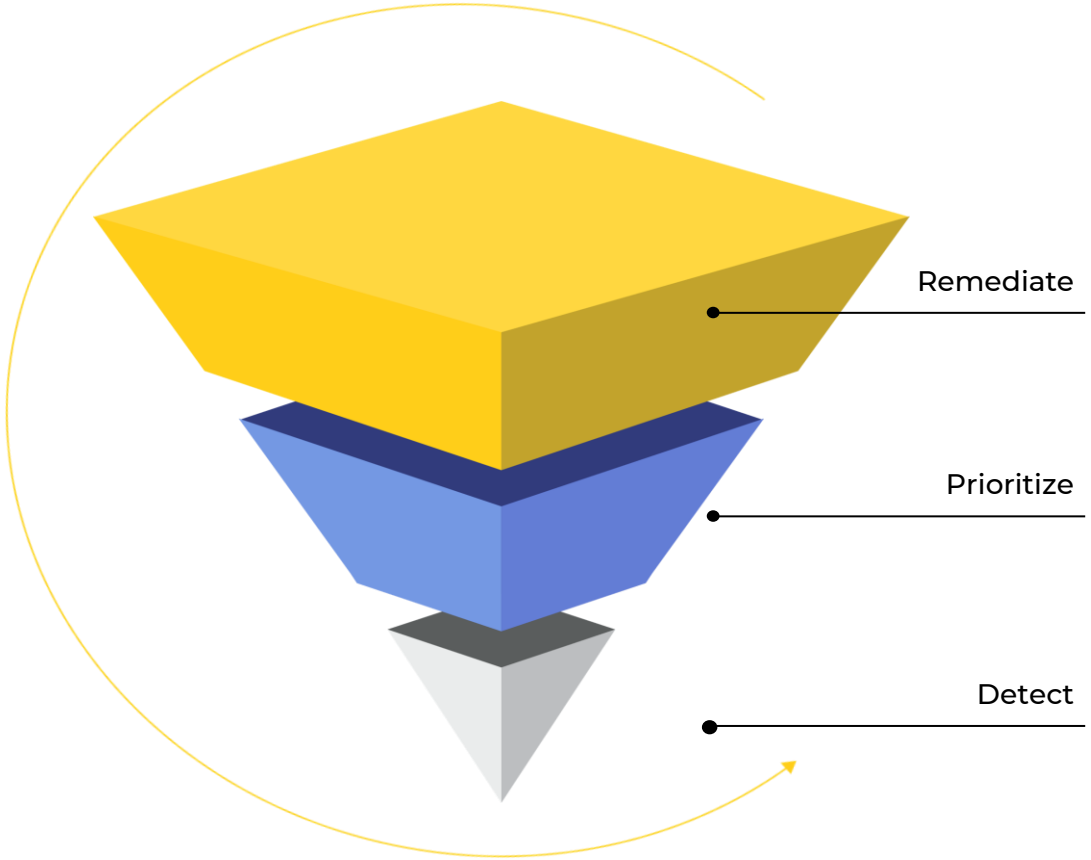
- Set, track and enforce policies
- Track and report on security and compliance posture

The traditional approach



Most of the Effort

Remediation-centric approach



Free tools to use

- **WhiteSource Bolt** – Get real-time security alerts and compliance issues on your open source dependencies within Azure DevOps or GitHub.
- **WhiteSource Renovate** - Save time and reduce risk by automating dependency updates in software projects.
- **WhiteSource Cure** – The first-ever security auto-remediation application designed for custom code.
- **WhiteSource Merge Confidence** - Identifies and flags undeclared breaking releases based on analysis of test and release adoption data across WhiteSource Renovate's early-adopting user base.

Reference: whitesourcesoftware.com/free-developer-tools/

Demo Time!

Any Questions?