Test Design Techniques in Security Testing

by Artem Vasiuk

About Me

Artem Vasiuk

- From Ukraine. Live in Denmark
- In testing since 2004
- Test Manager in Scalepoint

In Scope

- Where to start & to go
- How to design Security checklists
- Process Maturity levels
- Practical challenges

Out of Scope

- Hacking or Cracking techniques
- Pentesting

Reasons

- Breaches ratio increase
- More cracking tools & knowledge
- Area for personal growth
- Career opportunity

Needed effort

- Part of Quality
- Non-Functional requirement
- White-hat hacker mindset

Continuous effort

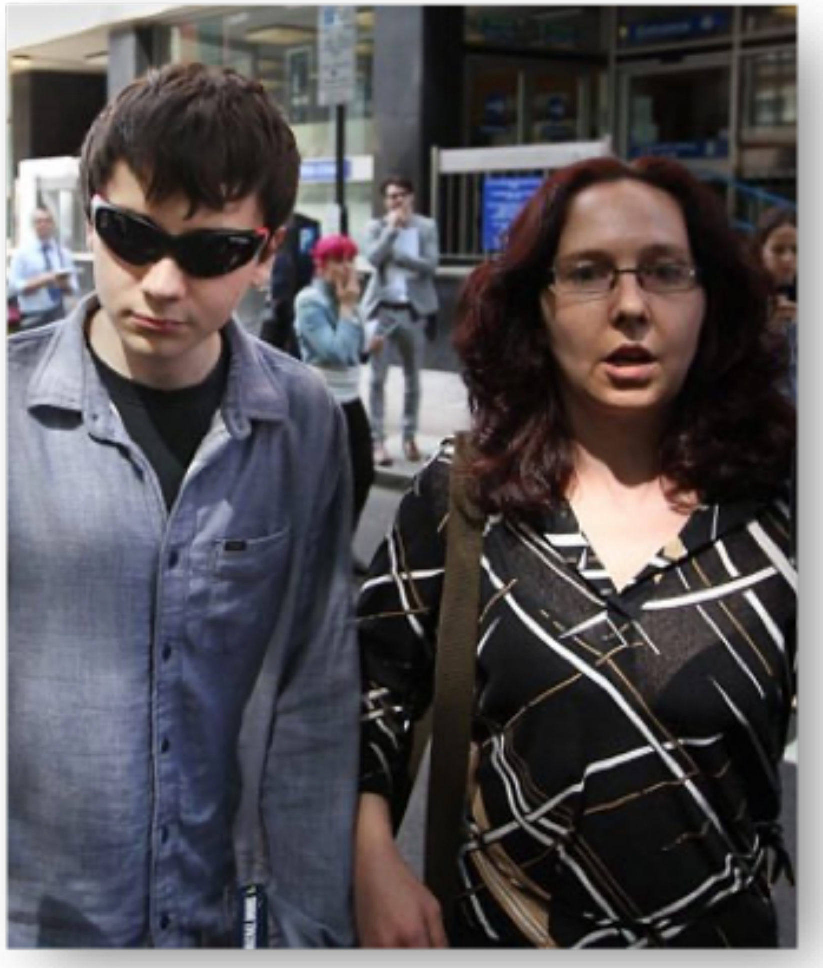
- "We are secure" is not permanent state
- "We use external component" is not an excuse
- Team effort

Why Security?

Meet hackers (expected)

Meet hackers (actual)

Ryan Cleary, 19
(and his mum)



Jake Davies, 18
(and his mum)

Principles and Techniques

Principles

Build Tactics and Strategy

Define Scope of Security Testing

Integrate into SDLC

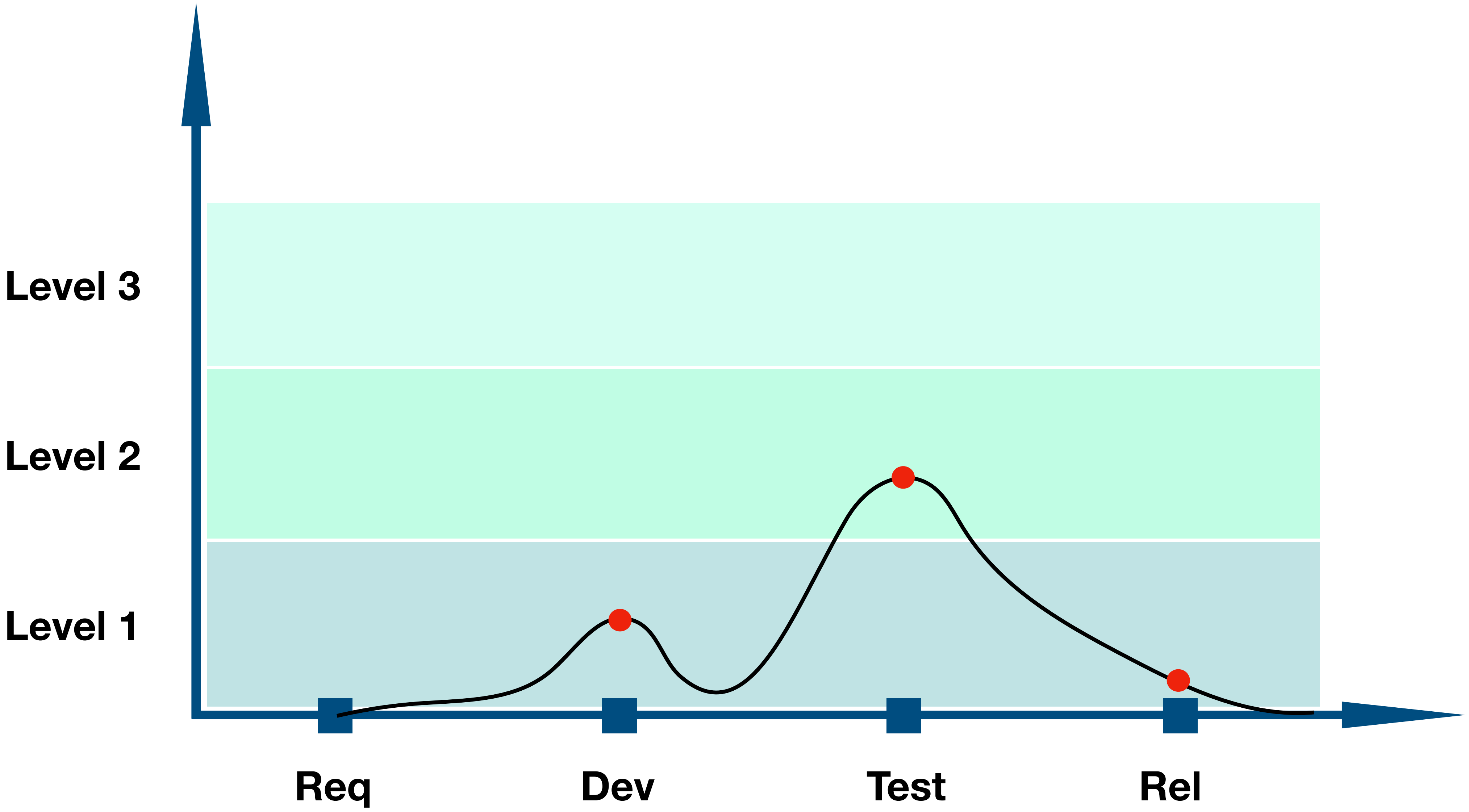
No silver bullet

Techniques

Review and Inspection	*[on Requirements]*
Threat Modelling	*[on Design]*
Code Analysis (SAST)	*[on Development]*
Penetration Testing (DAST)	*[on Testing]*

Let's talk about ....

Process Maturity



Process Maturity Levels

AdHoc	Controlled	Efficient	Optimising
Learn as you go	Organise your efforts	Improve what you know	Integrate the knowledge

What does AdHoc mean?

"For this situation"

"Done for a particular purpose as necessary"

"Informal testing with an aim to break"

Testing WebApp architecture

- Logic flows and flaws
- Types of UI controls
- User input validation
- URL & Body of HTML requests/responses
- HTTP methods

Typical Attack vectors

- OWASP Top 10
- Bypassing validation
- Parameters tampering
- Impersonating

Practice

Challenge #1

Explore JuiceShop for the security flaws

URL: https://www2.owasp.org/www-project-juice-shop/

What's needed for AdHoc?

People with knowledge and skills

Process Maturity Levels

AdHoc	Controlled	Efficient	Optimising
Learn as you go	Organise your efforts	Improve what you know	Integrate the knowledge

What does Control mean?

The power to influence people's behaviour or the course of events.

Controlled

OWASP Testing Guide

- Process overview
- Test Cases
- Examples
- Follow and adjust to your own needs
- Educate yourself and team

Practice

Challenge #2

Design a 5-step security checklist for Sanity checks

URL: https://www.owasp.org/index.php/Testing_Checklist

Practice

Challenge #3

Apply security checklist on JuiceShop

Controlled

Definition of Done

The acceptance criteria that are common to every single user story.

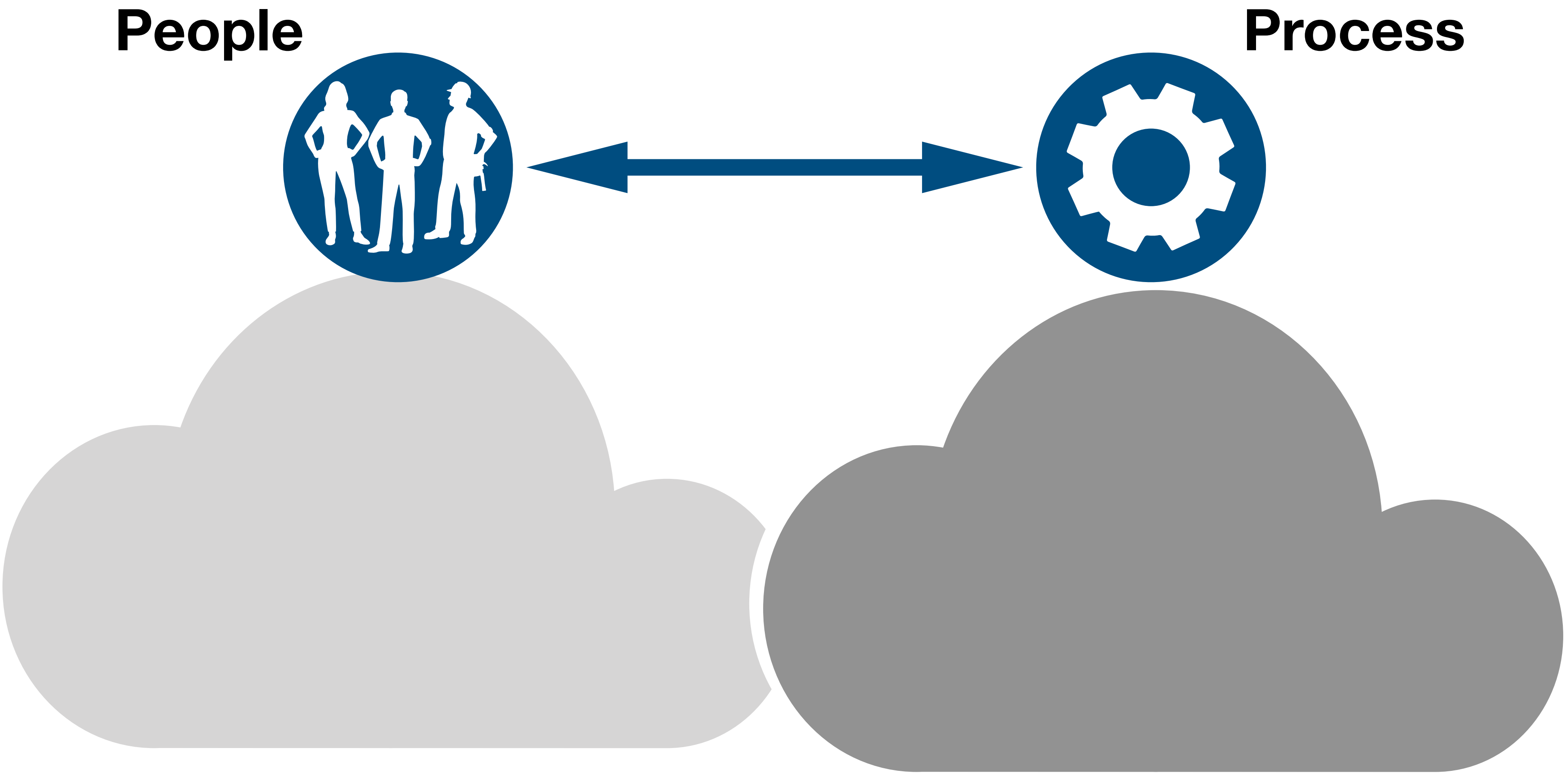
- Code reviewed
- Verified in test environment
- Automated tests written and passed
- Regression testing completed
- Functionality is Security Verified

EXAMPLE

Controlled

What's needed to have Control?

People working within Process



People

Process

Small Follow-up

What did we talk about?

- Security is a part of product quality
- Testing without specific goals is non-productive
- Sanity Checklists improve your process
- Consider Security when you say "Done"

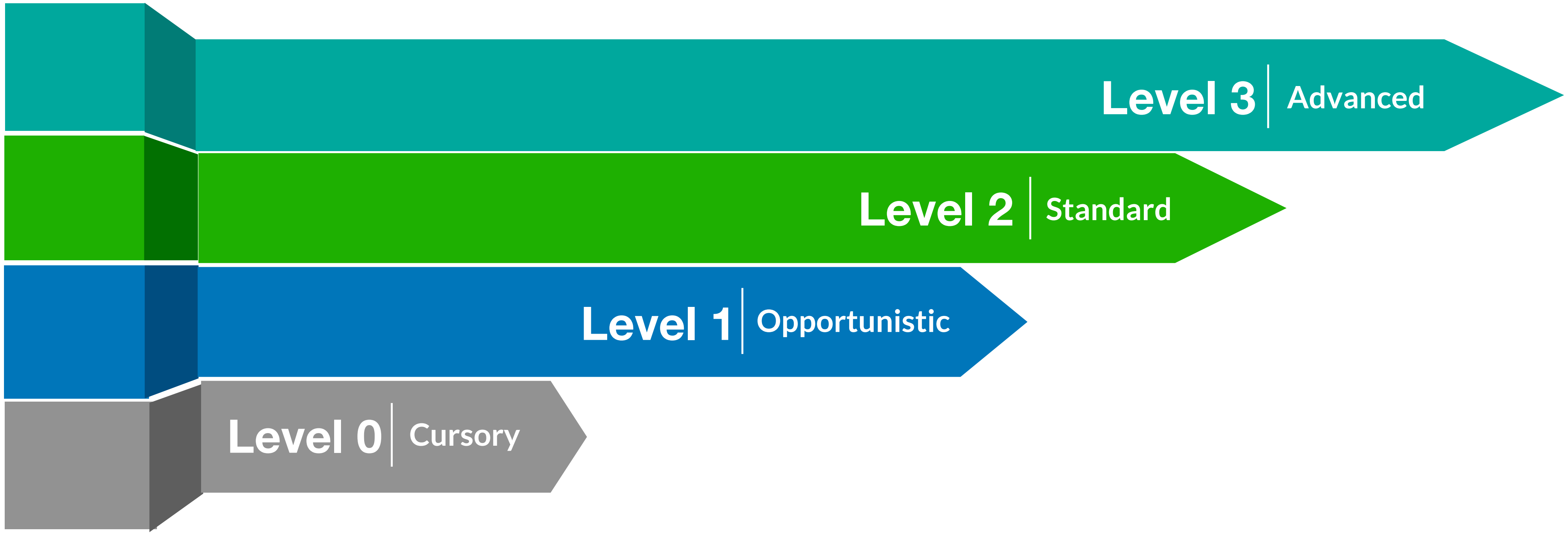
Process Maturity Levels

AdHoc	Controlled	Efficient	Optimising
Learn as you go	Organise your efforts	Improve what you know	Integrate the knowledge

Efficient

Application Security Verification Standard (**ASVS**)

Framework of security requirements that focus on normalising the functional and non-functional security controls required when designing, developing and testing modern web applications.



Practice

Challenge #4

Map a security checklist with similar ASVS controls

URL: https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

Efficient

BurpSuite application

- Built for Dynamic AppSec Testing
- Manipulating requests
- Automated attacks
- Automated Scanning for vulnerabilities*
- Vulnerabilities reporting*

*Professional version

BurpSuite application

- Intercept requests/responses between browser and server
- Build requests manually
- Crawl a website by automatically visiting every page
- Fuzz applications by sending valid & invalid data

Practice

Challenge #5

Bypass client validation using BurpSuite

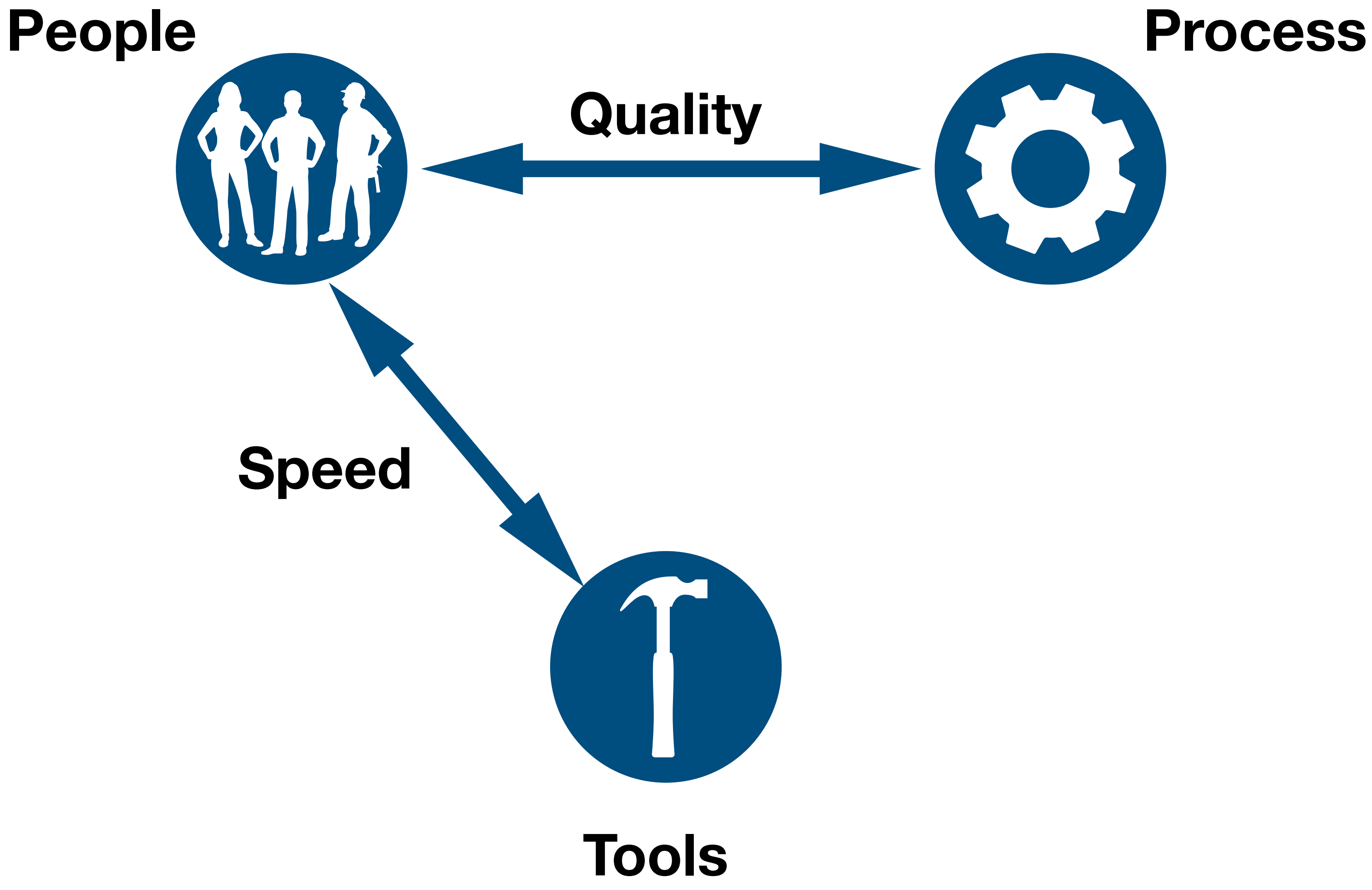
Efficient

sonarqube

- Continuous Code Inspection
- Code quality, Security, Tech Debt, Dependencies
- Numerous plugins (languages, scanning tools, reporting etc.)

What's needed to gain Efficiency?

People working within Process with Tools



Process Maturity Levels

AdHoc	Controlled	Efficient	Optimising
Learn as you go	Organise your efforts	Improve what you know	Integrate the knowledge

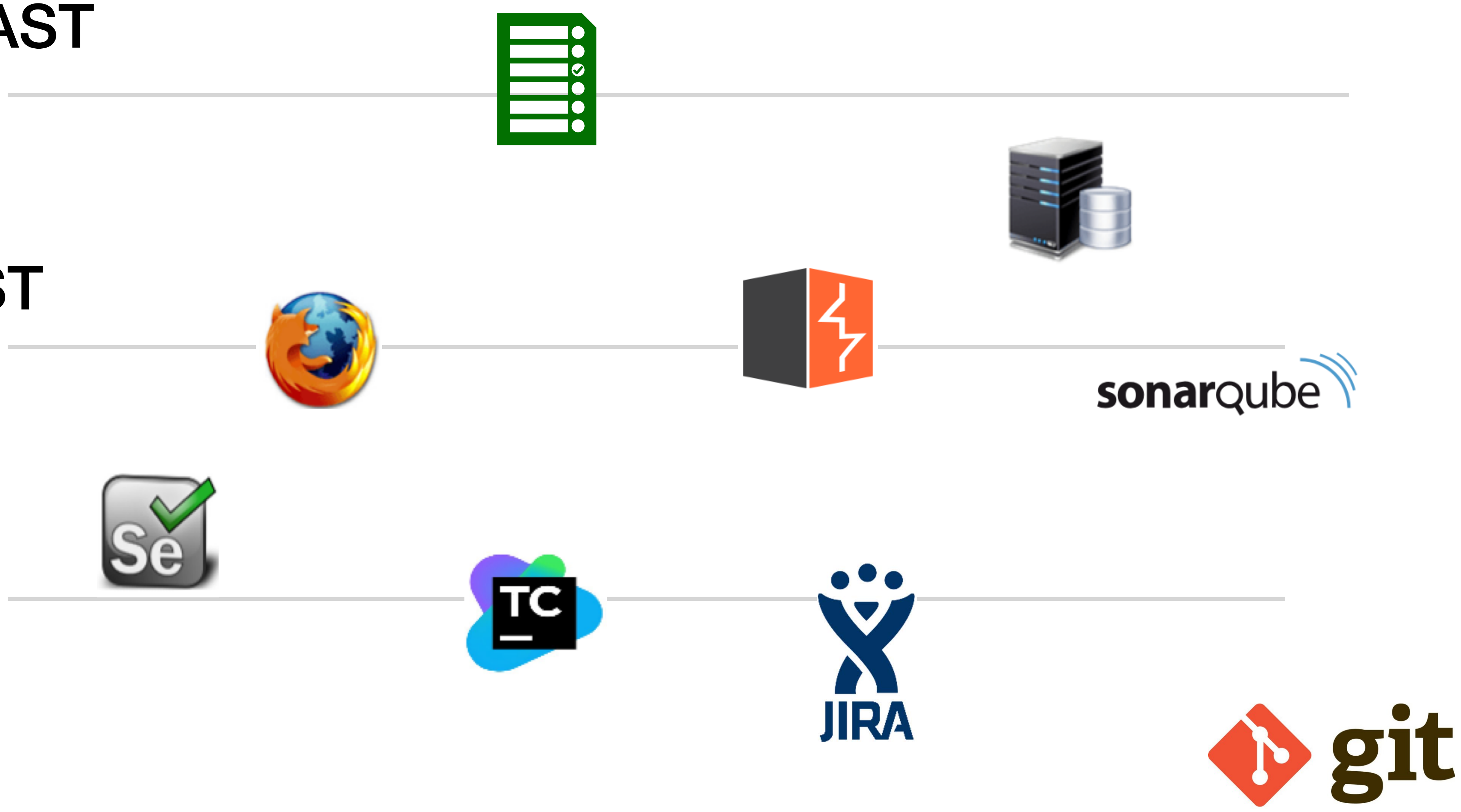
Optimising

Let's see what we have now

Automated DAST

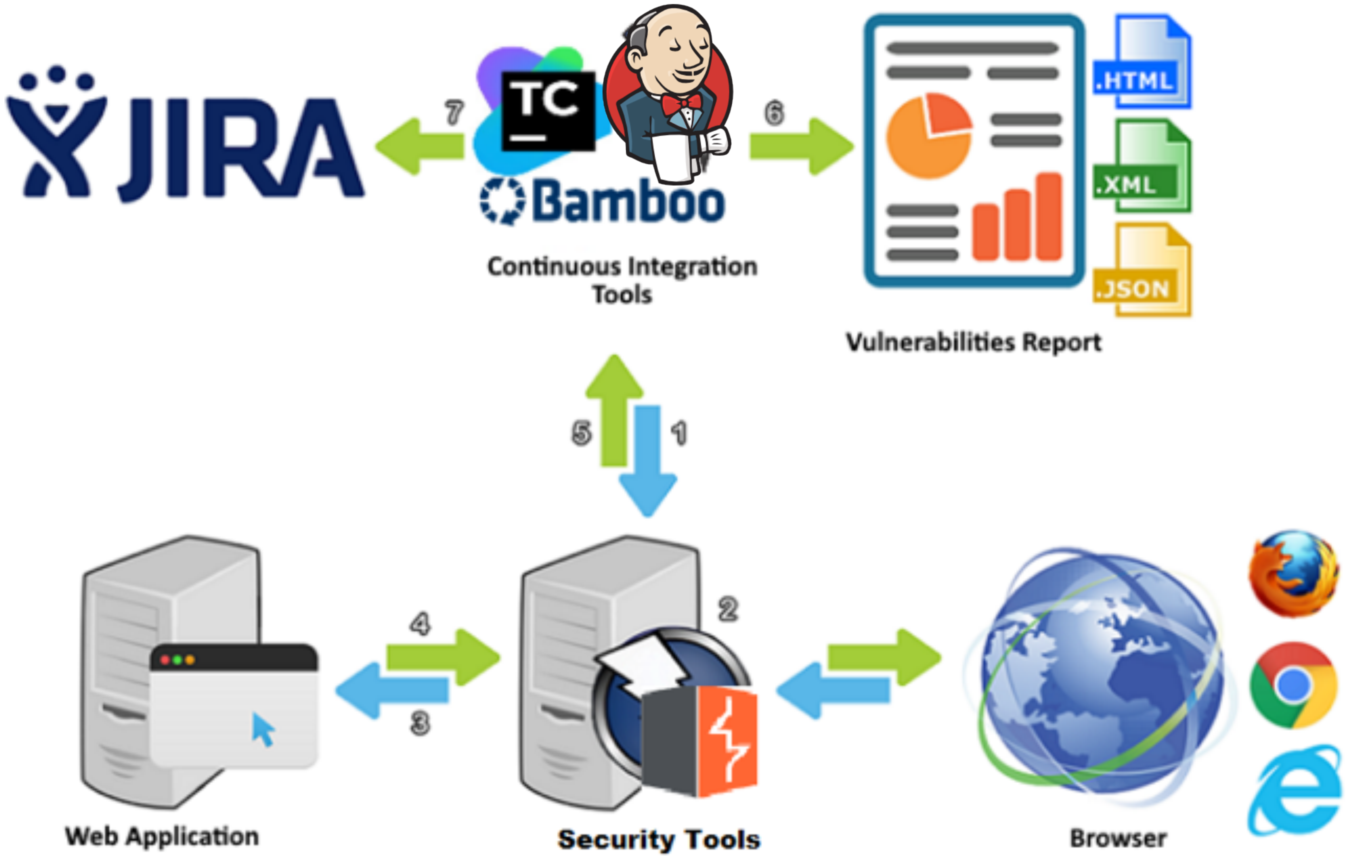
Manual DAST

SAST



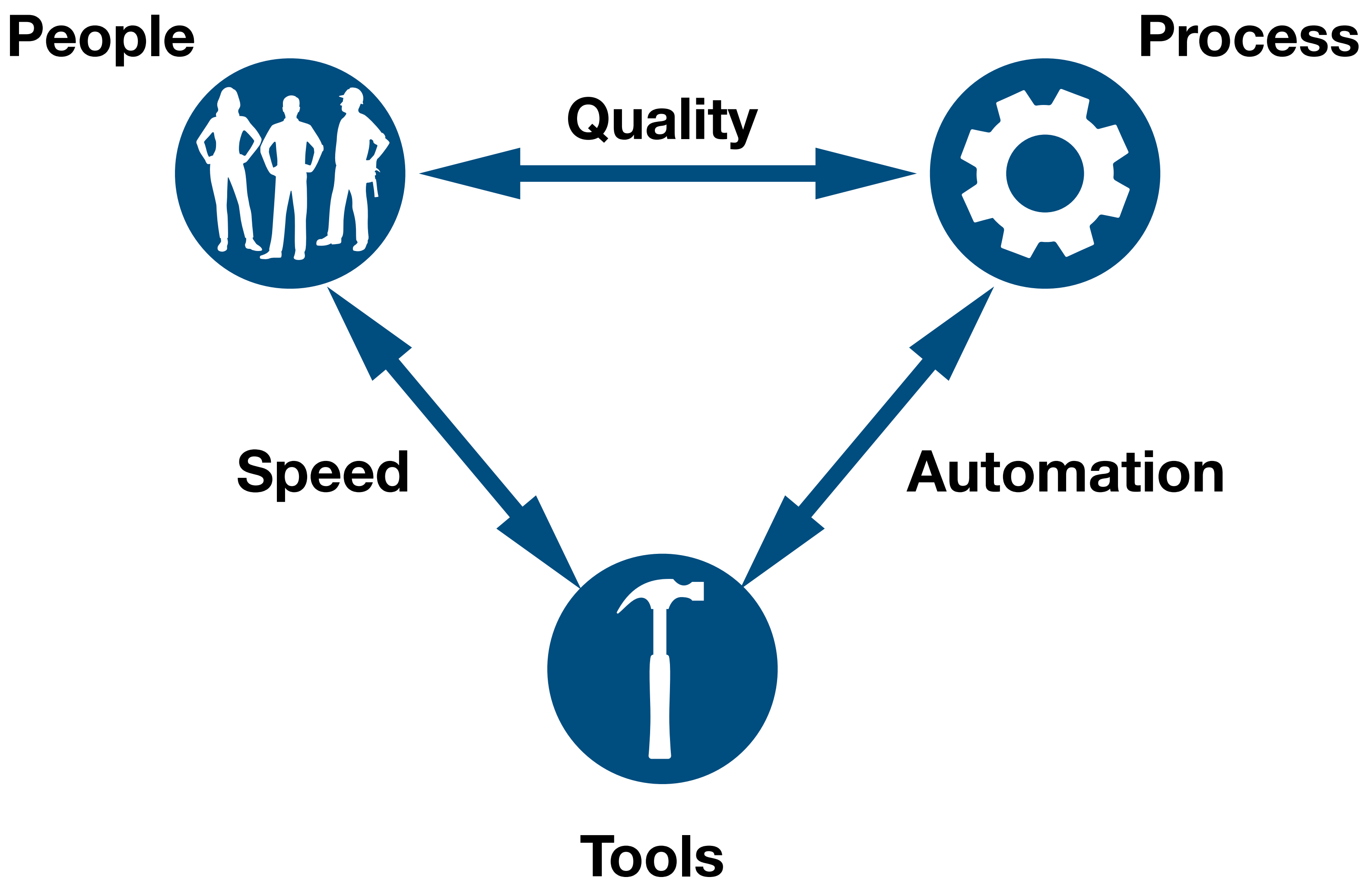
Continuous Testing

- Testing Early
- Testing Often
- Test Everywhere
- Automation



How can we Optimise?

Introduce automation of Tools



Practice

Challenge #6

Automate a scenario and run it through BurpSuite

Security Ambassador

- A Role, not Responsibility
- Concerned about Security related questions
- Knows the drill and is ready to act
- Has good communication skills

The End

Thank you!