Taming rainbow shitting unicorns

A story on DevSecOps

Dave van Stein | PVIB CISO 28

dvanstein@xebia.com

@Dave_von_S

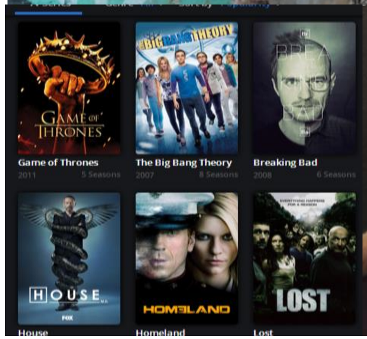
nl.linkedin.com/in/dvstein

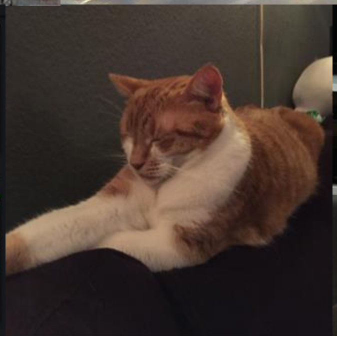
github.com/davevs

CSSLP® Certified Secure Software Lifecycle Professional

DASA DEVOPS FUNDAMENTALS

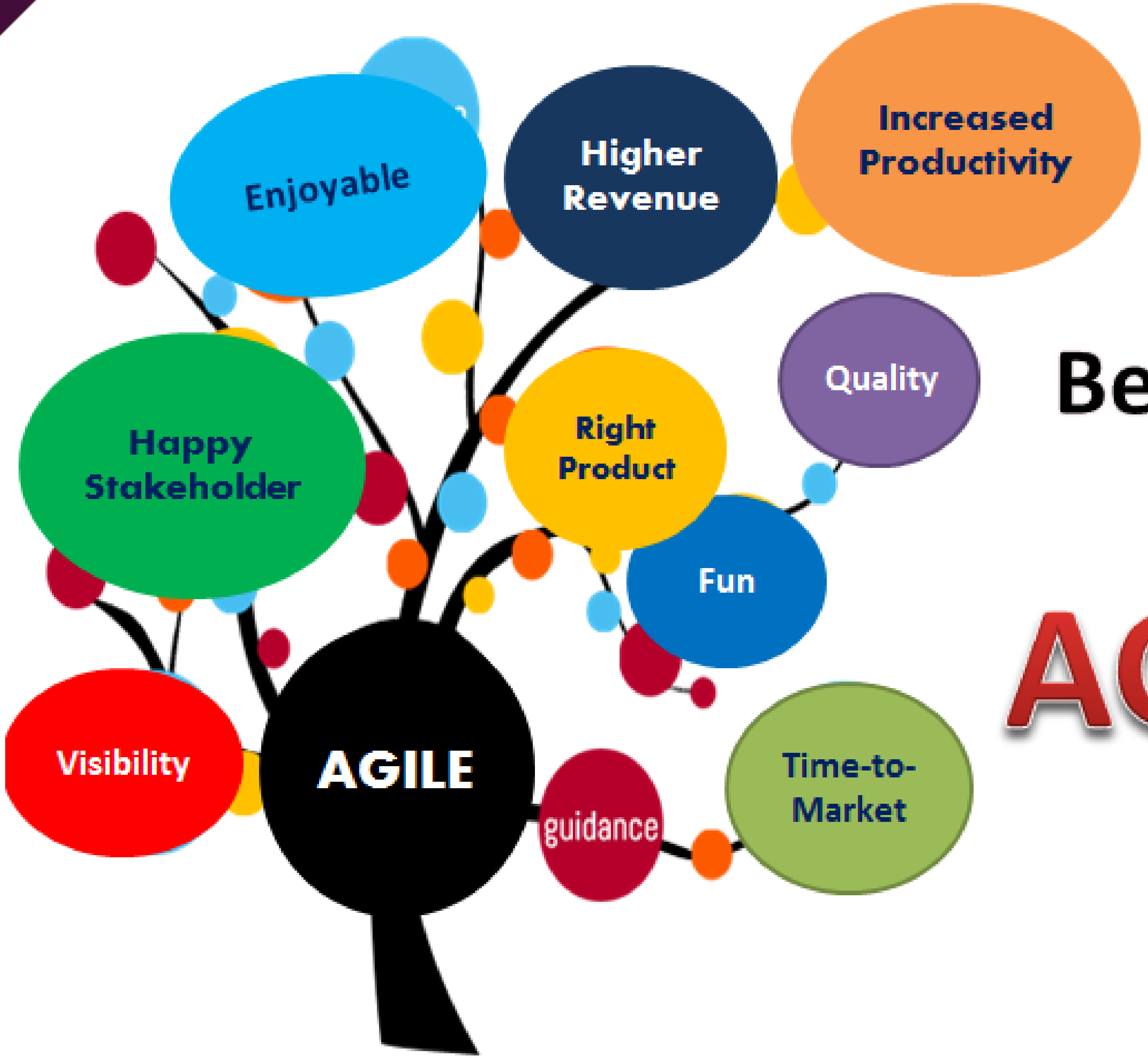
Certified Information Privacy Technologist CIPT iapp





Disclaimer

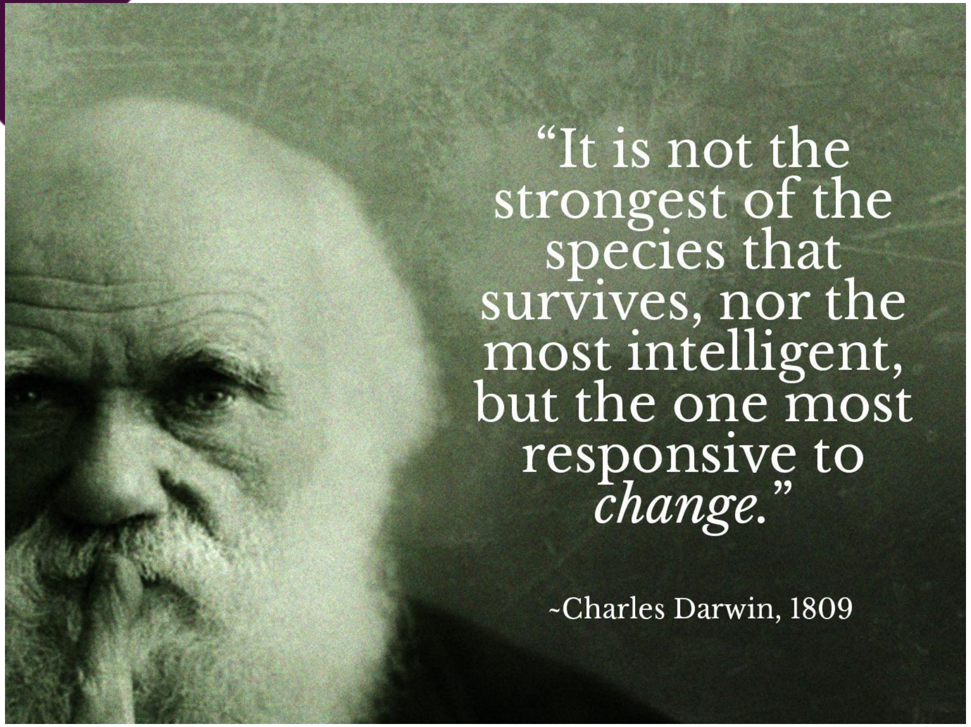
Agile makes you more agile!



Benefits of AGILE

Security?

Security & Agile?



"It is not the strongest of the species that survives, nor the most intelligent, but the one most responsive to *change.*"

~Charles Darwin, 1809

The agile move and adapt rapidly.

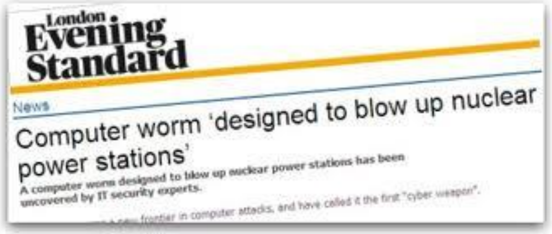
The robust can handle adversity but stay the same.

Xebia

Security has many faces

Sarbanes-Oxley
Financial and Accounting Disclosure Information

HIPAA
Health Insurance Portability
and Accountability Act



London Evening Standard

News

Computer worm 'designed to blow up nuclear power stations'

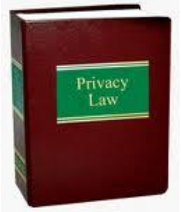
A computer worm designed to blow up nuclear power stations has been uncovered by IT security experts.

... a new frontier in computer attacks, and have called it the first "cyber weapon".

PCI Security Standards Council™

DigiD

NATIONAL SECURITY AGENCY
UNITED STATES OF AMERICA



CERTIFIED CERTIFIED CERTIFIED CERTIFIED
SAS 70
Type II

</xssed>
xss attacks information

FDA U.S. Food and Drug Administration
Protecting and Promoting *Your* Health

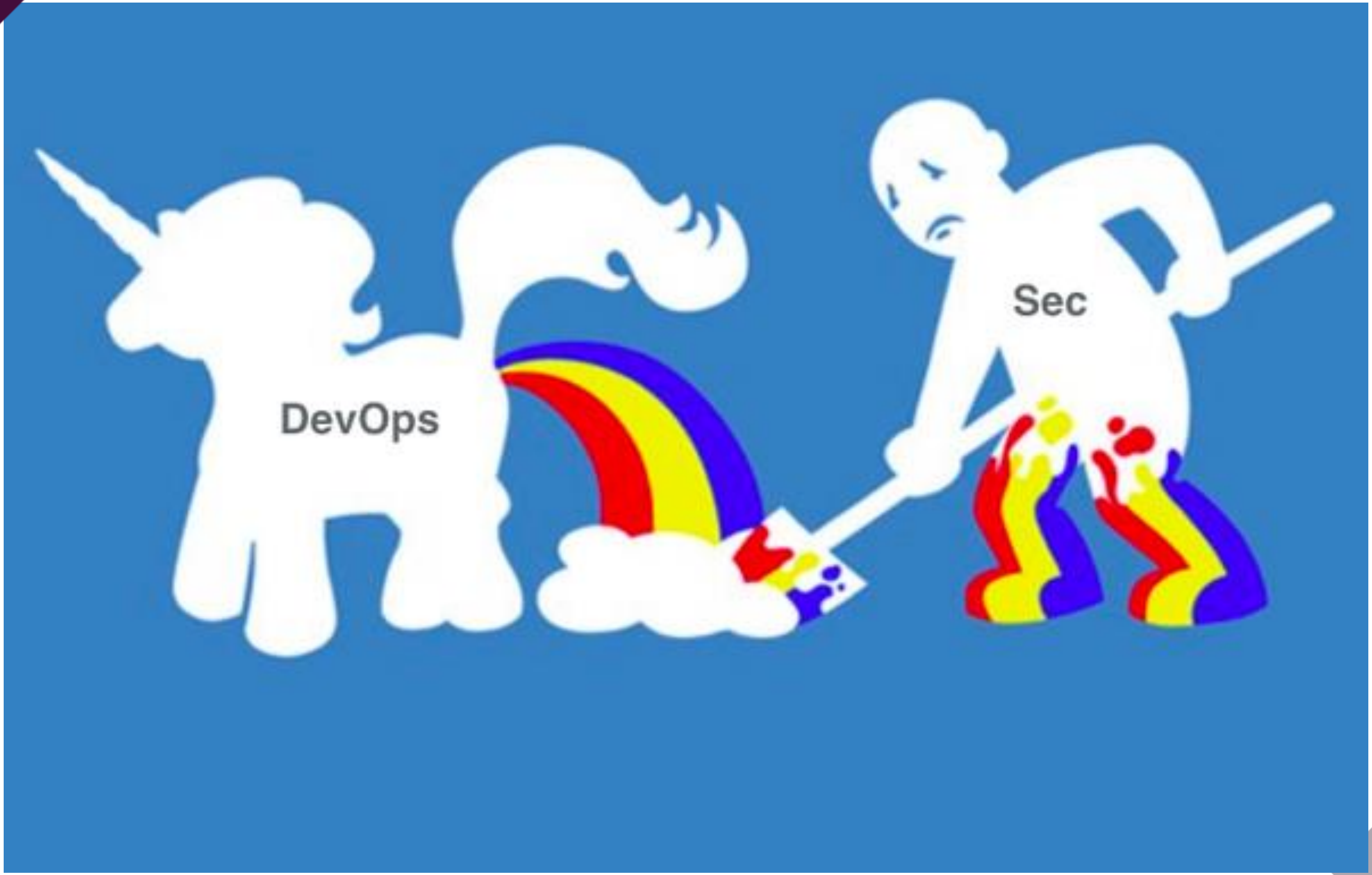
ISO 9001
ISO 27001
BUREAU VERITAS
Certification
BUREAU VERITAS
1828

uwv werken aan perspectief

Logius
Ministerie van Binnenlandse Zaken en Koninkrijksrelaties

23% College And University Students Hack IT Systems

One way to do it

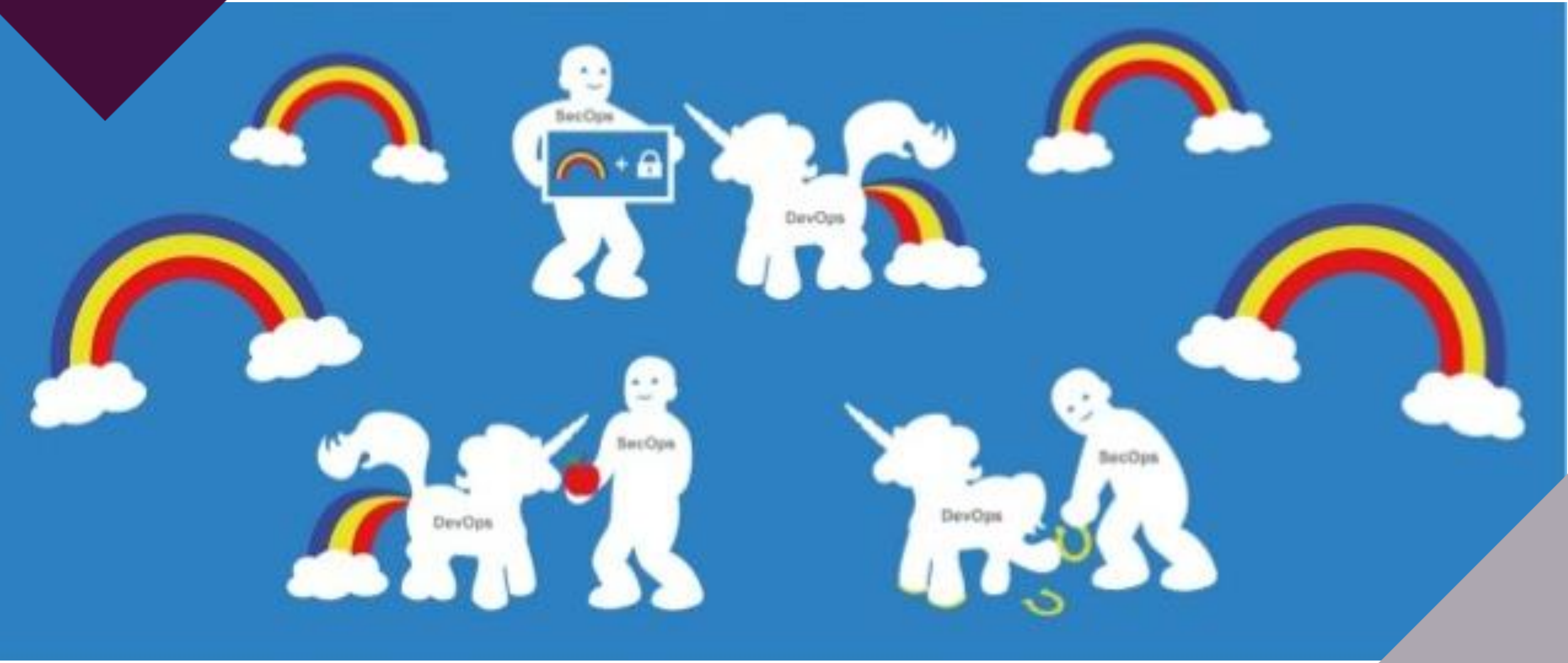


Pentests at the end of sprints

Mandatory policy checks

The Result

A better idea



A change from this

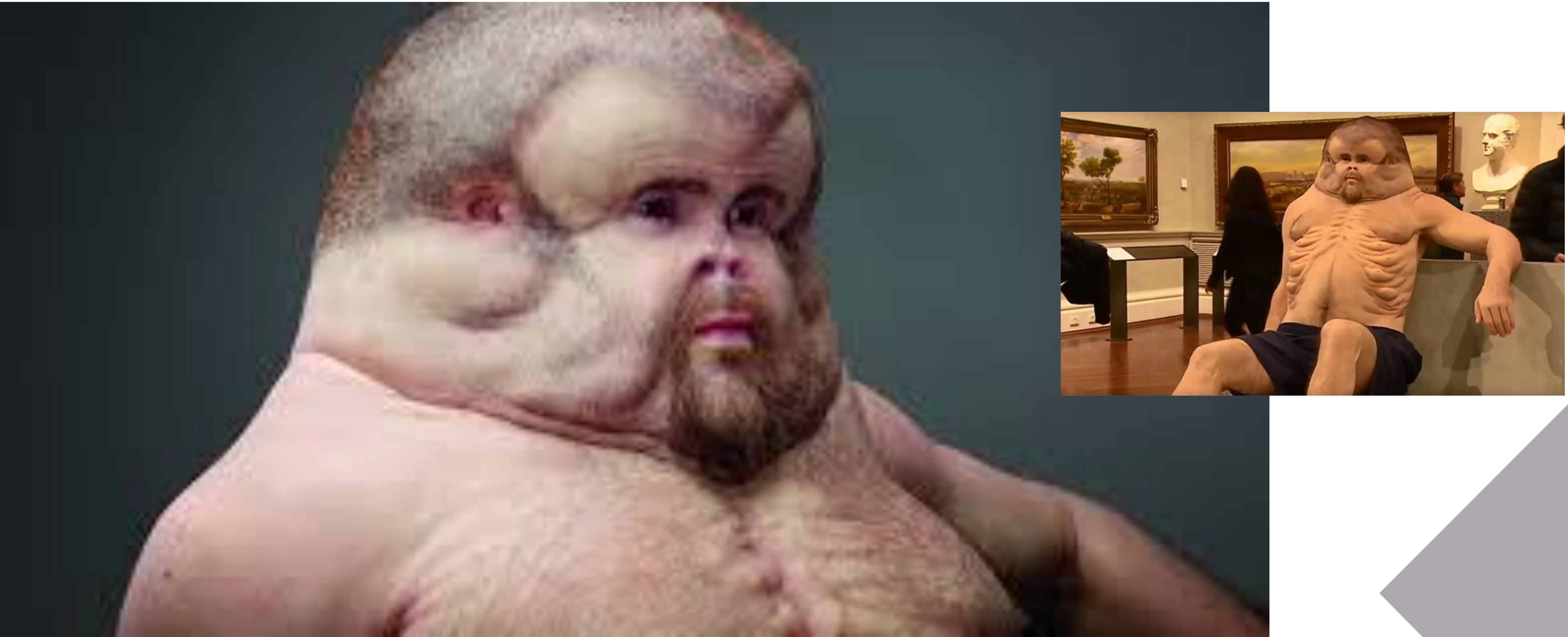
To this

30
km/h
TX-95-FN

A8
Coenplein
S116
S103
S106
info A10
A4
Amstel
A1

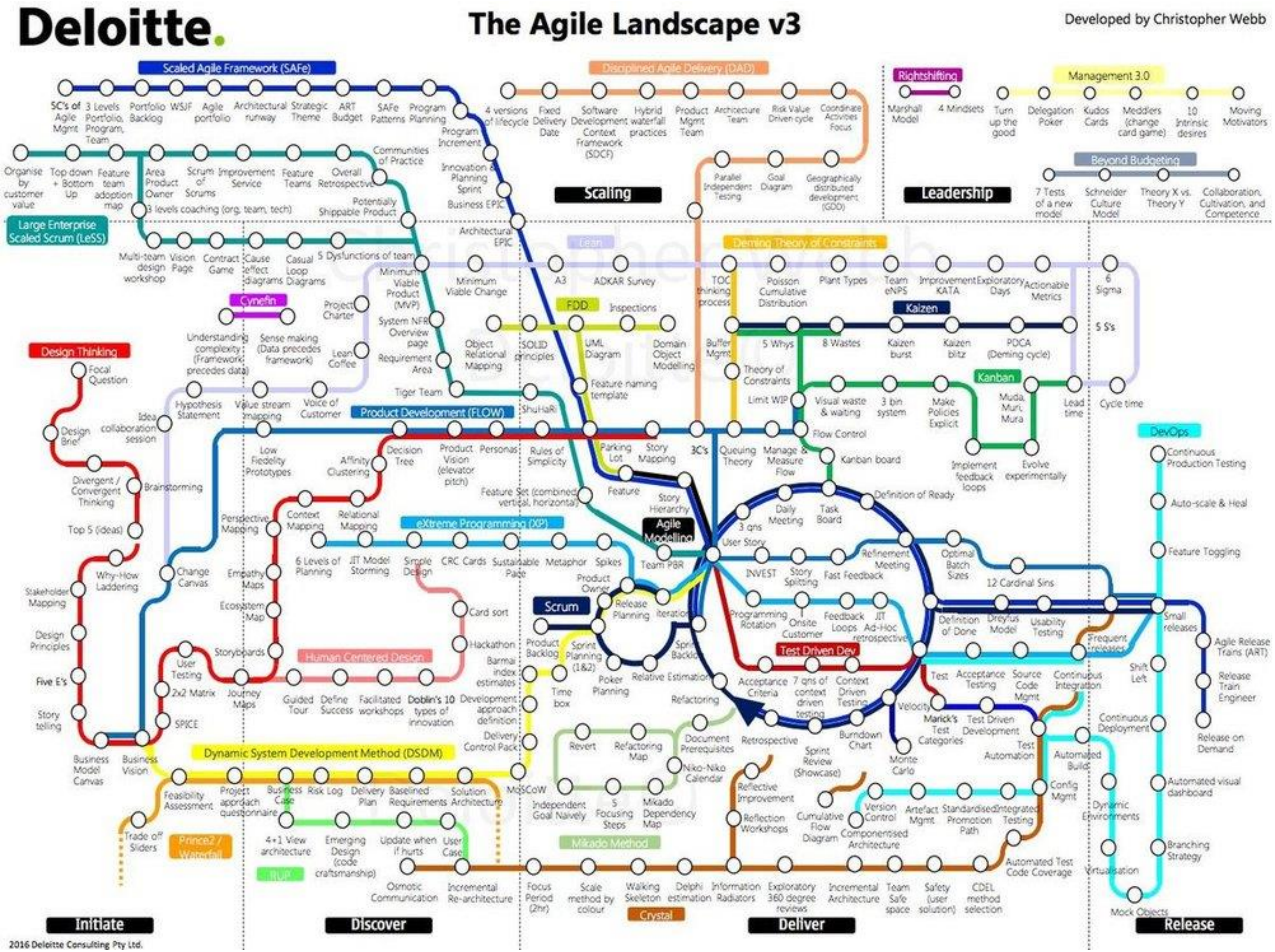
While preventing this

100% secure?

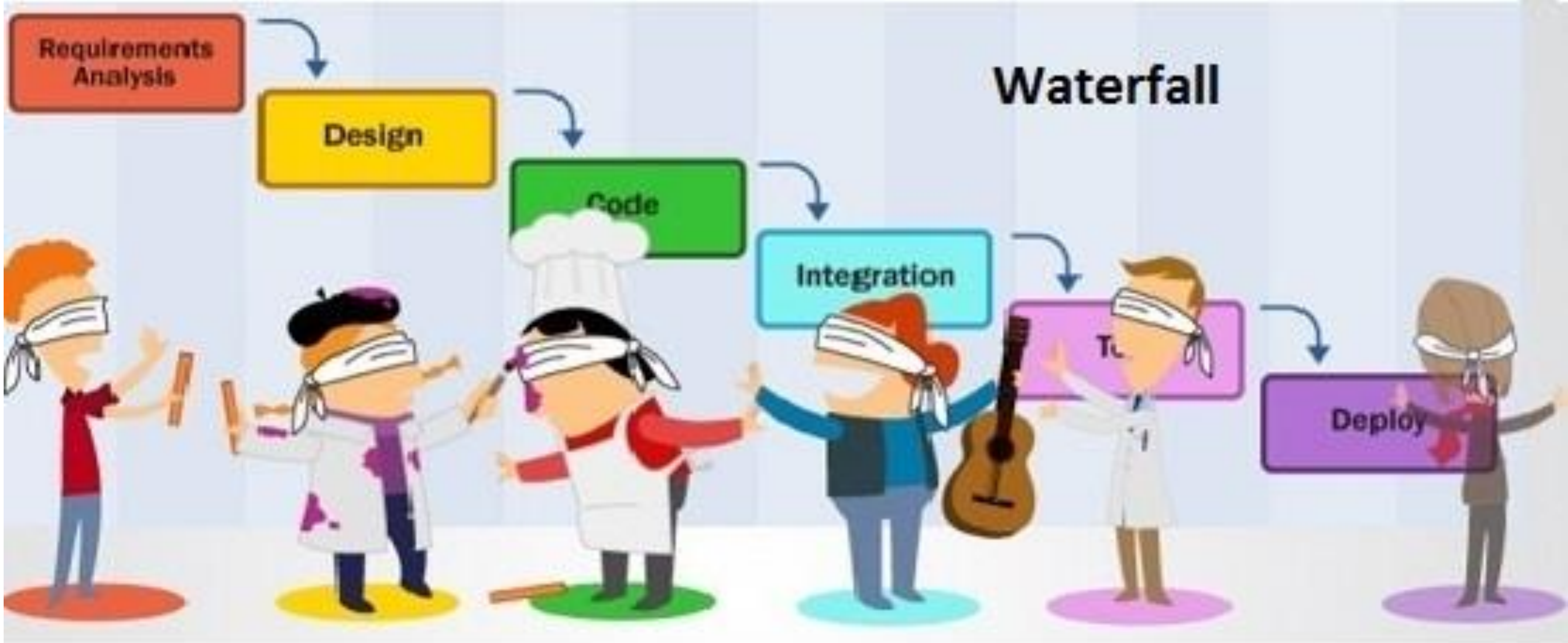


Xebia

Agile? DevOps? DevSecOps?



The classic approach

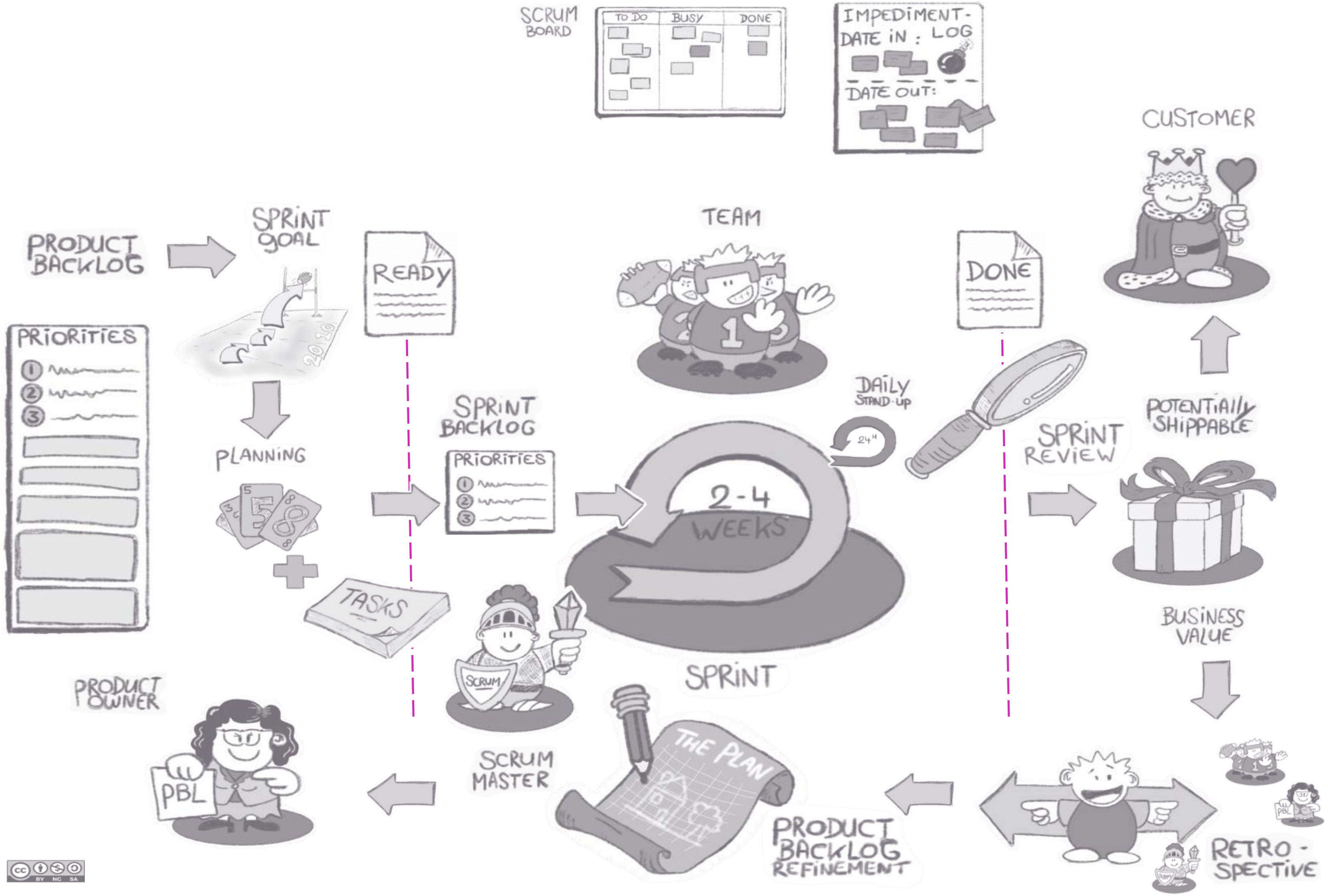


Agile

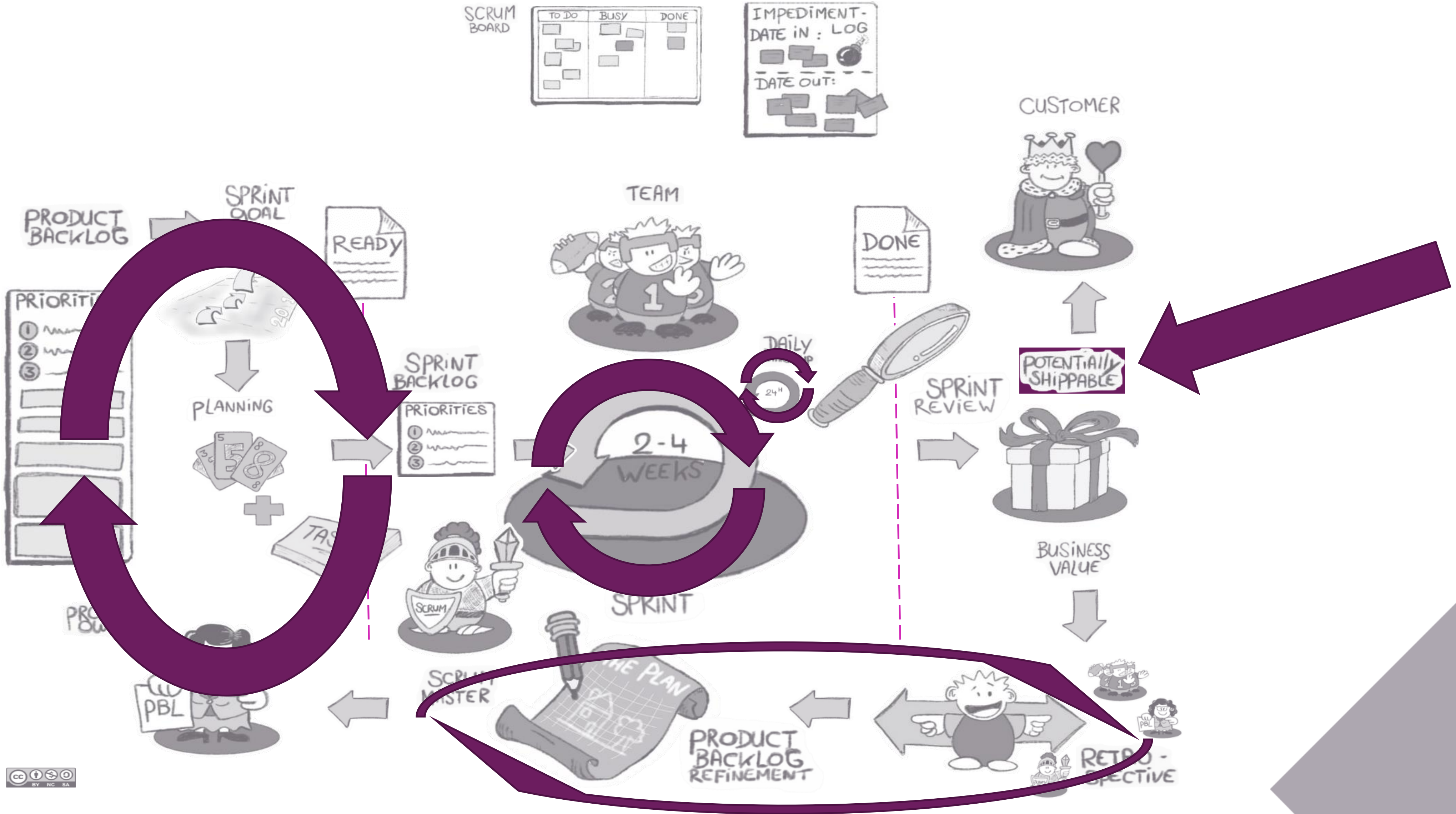
DevOps

DevSecOps

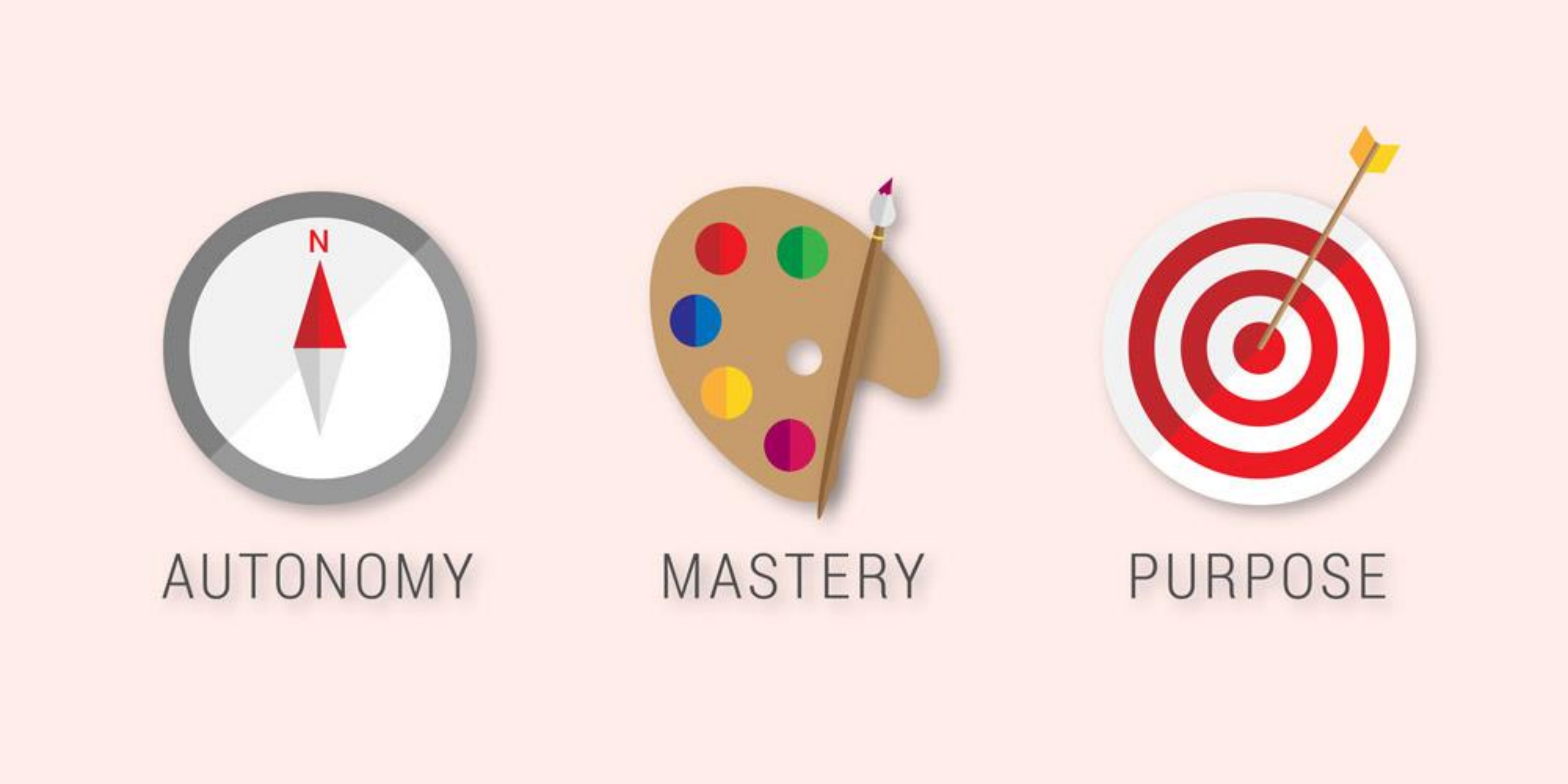
Agile: the important bits



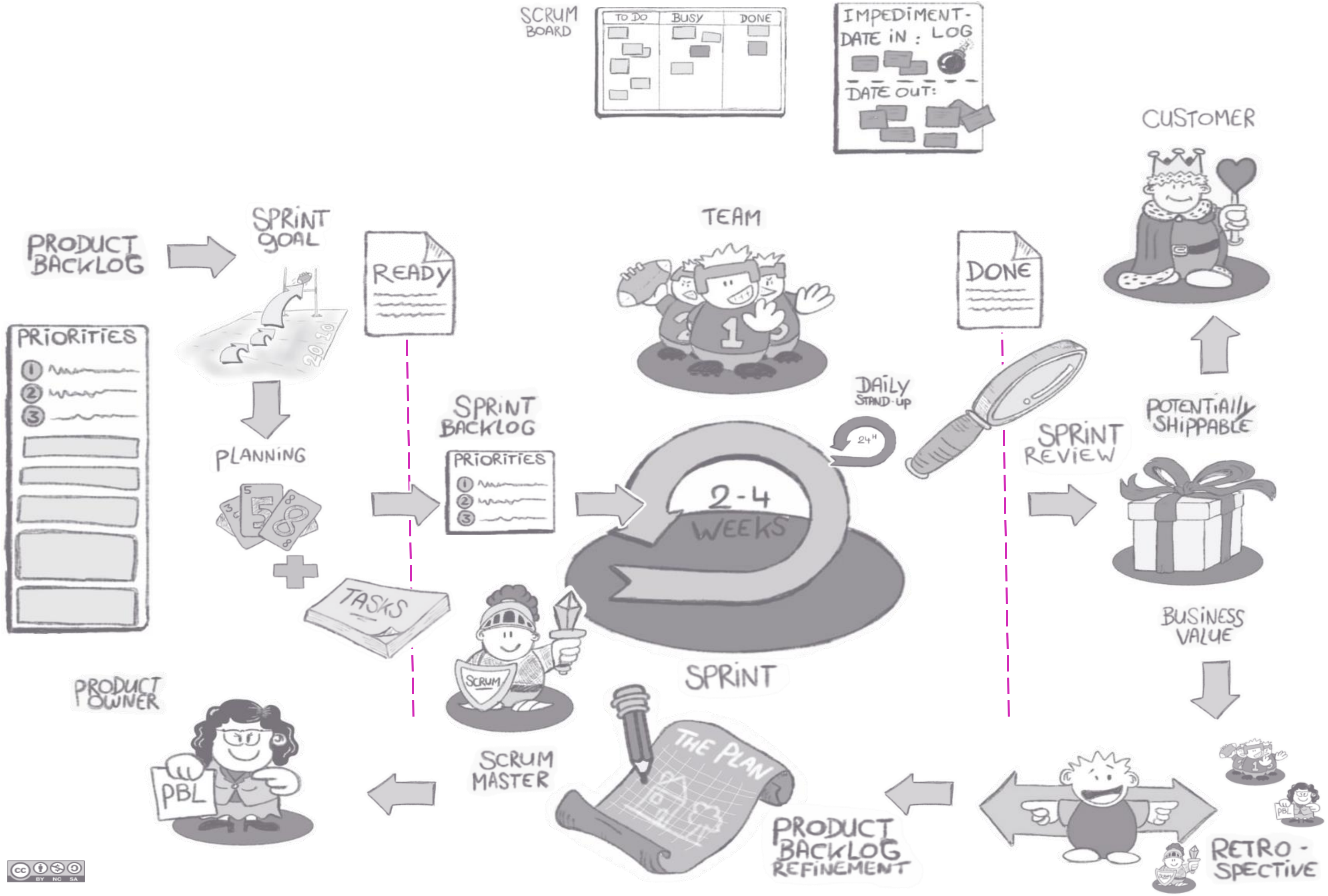
Agile: the important bits



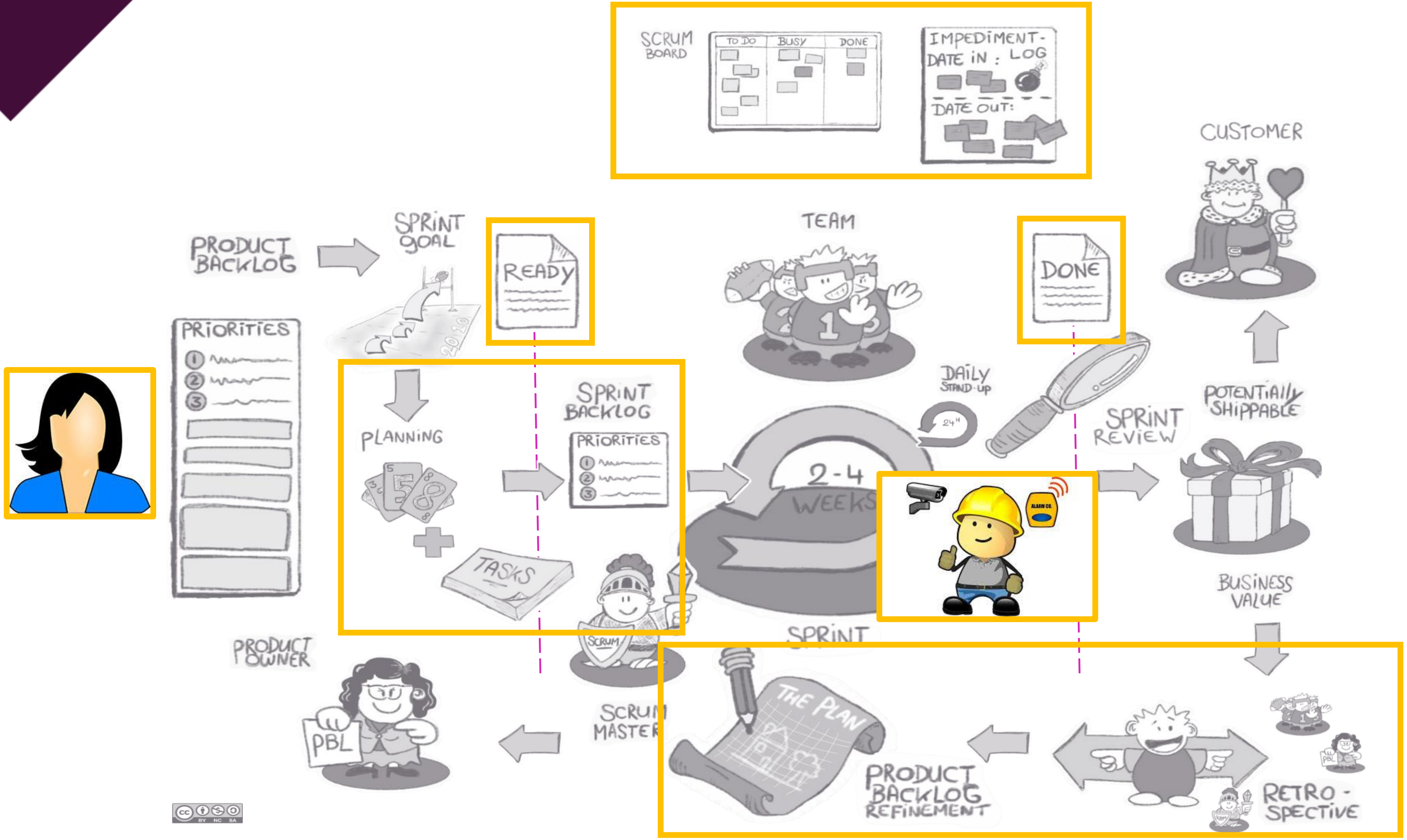
The real goal

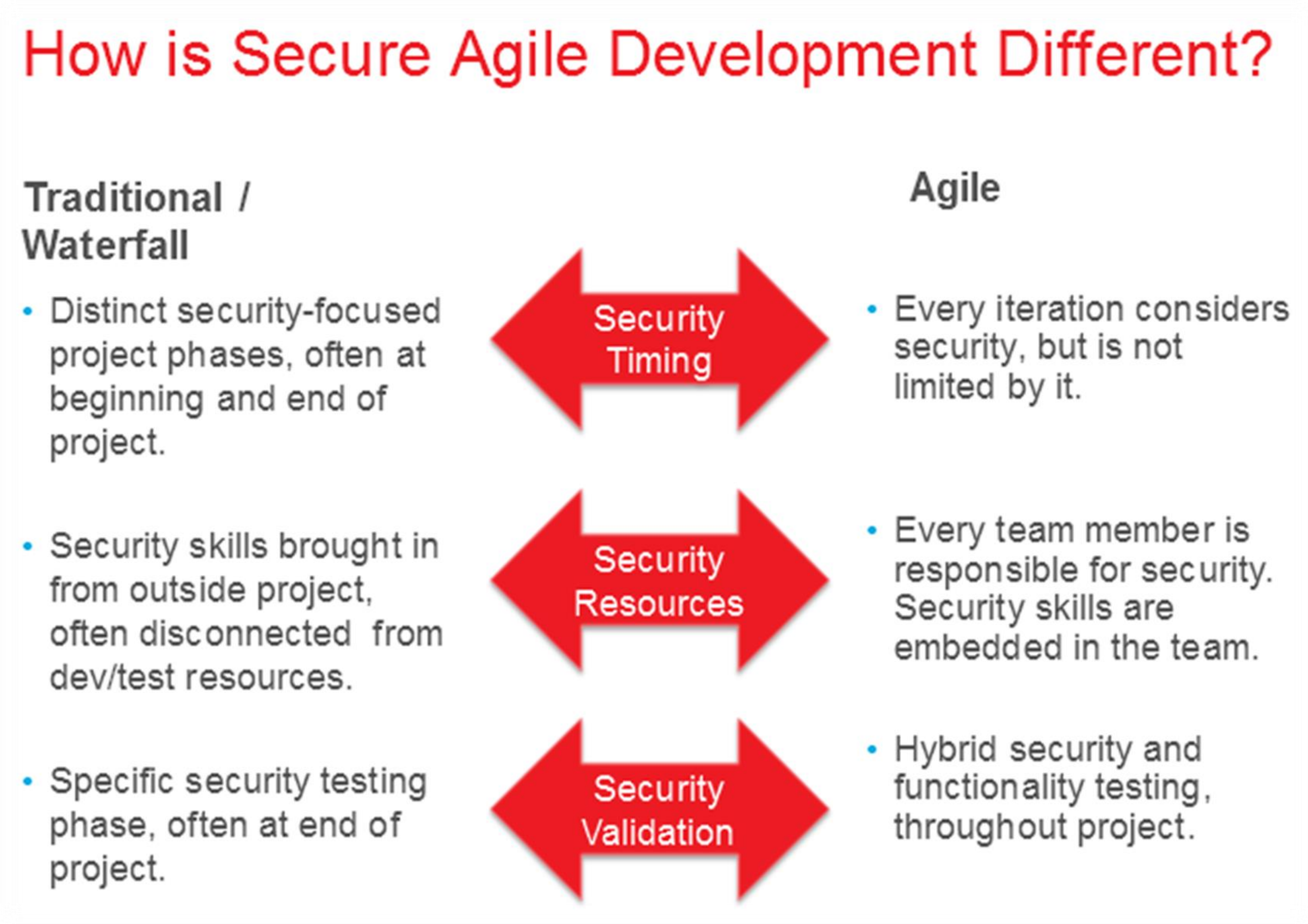


Where does security fit in?



Agile Security





How is Secure Agile Development Different?

Traditional / Waterfall

Agile

- Distinct security-focused project phases, often at beginning and end of project.

Security Timing

- Every iteration considers security, but is not limited by it.

- Security skills brought in from outside project, often disconnected from dev/test resources.

Security Resources

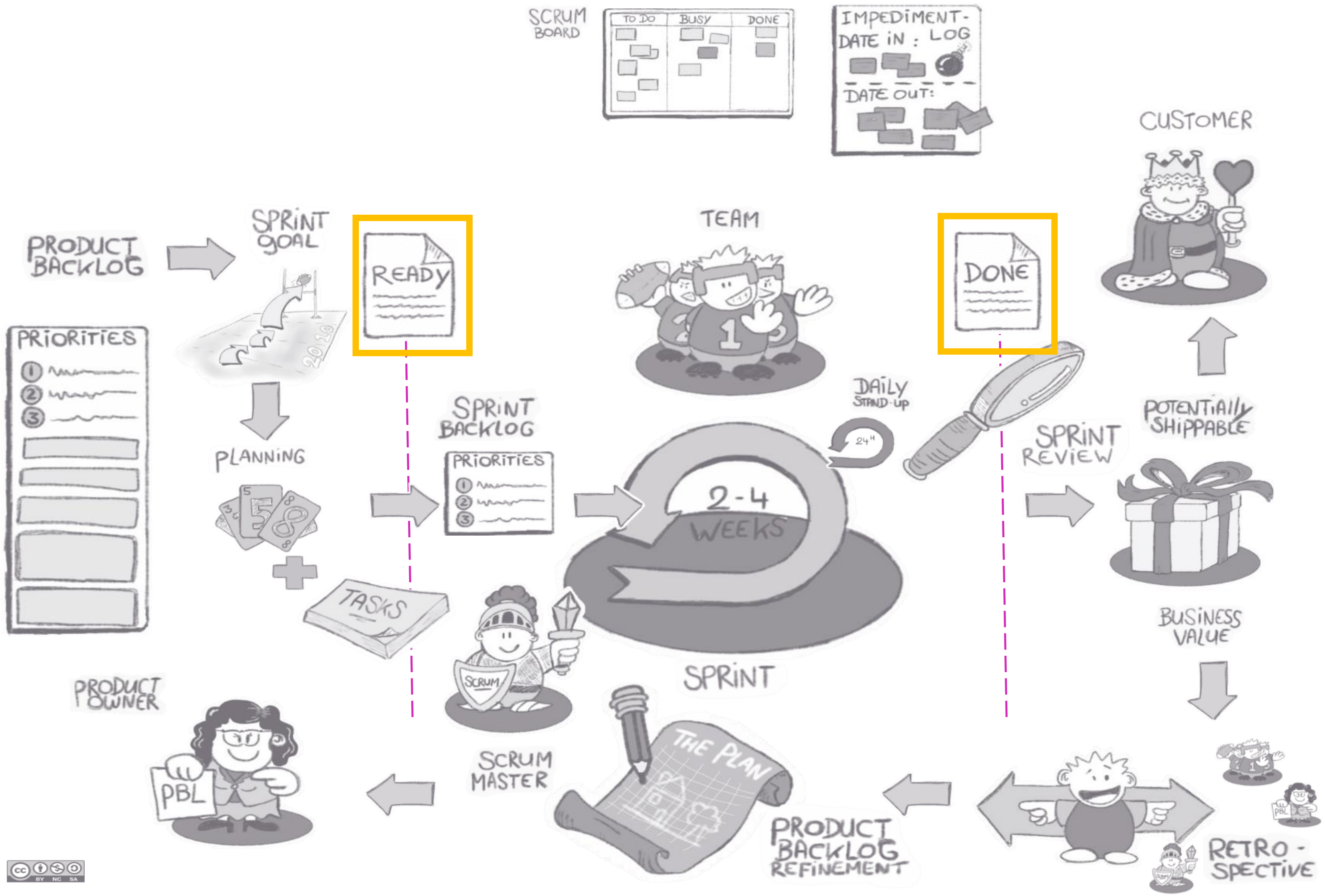
- Every team member is responsible for security. Security skills are embedded in the team.

- Specific security testing phase, often at end of project.

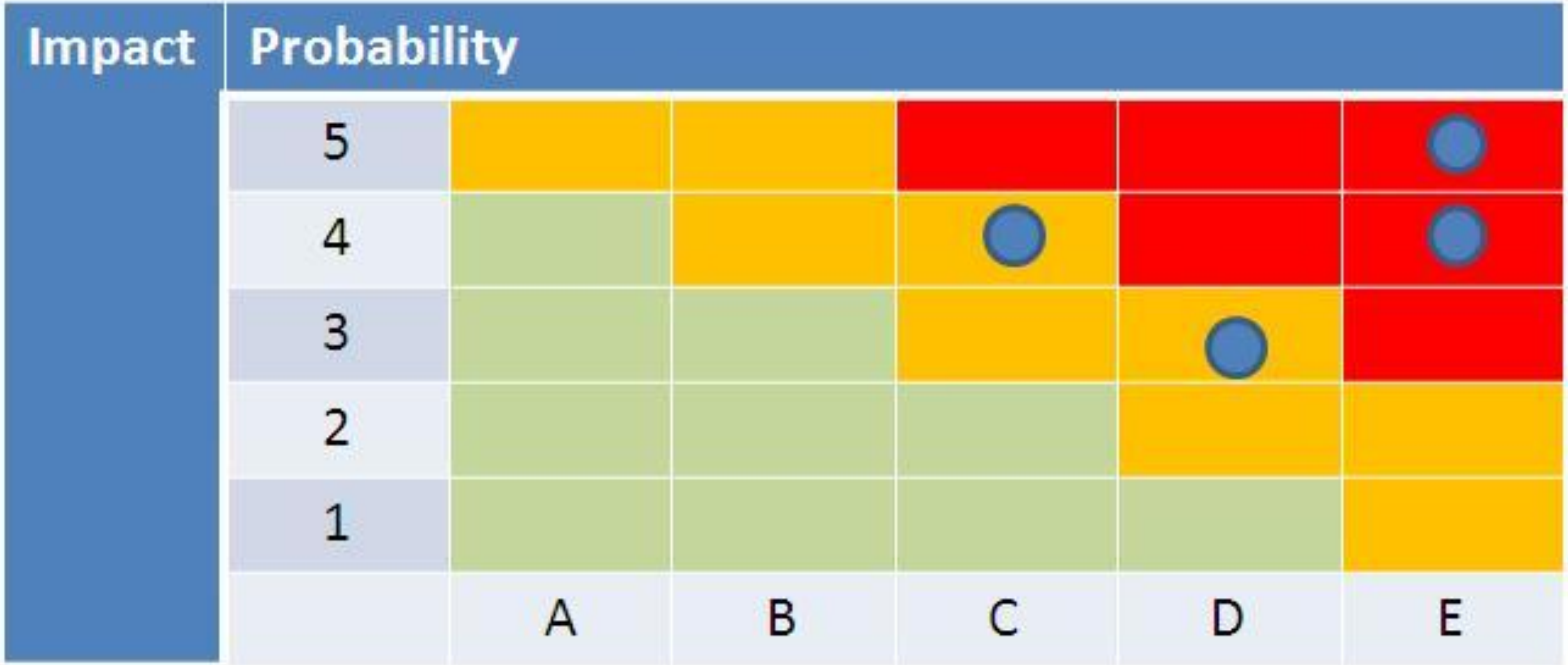
Security Validation

- Hybrid security and functionality testing, throughout project.

Security Thresholds

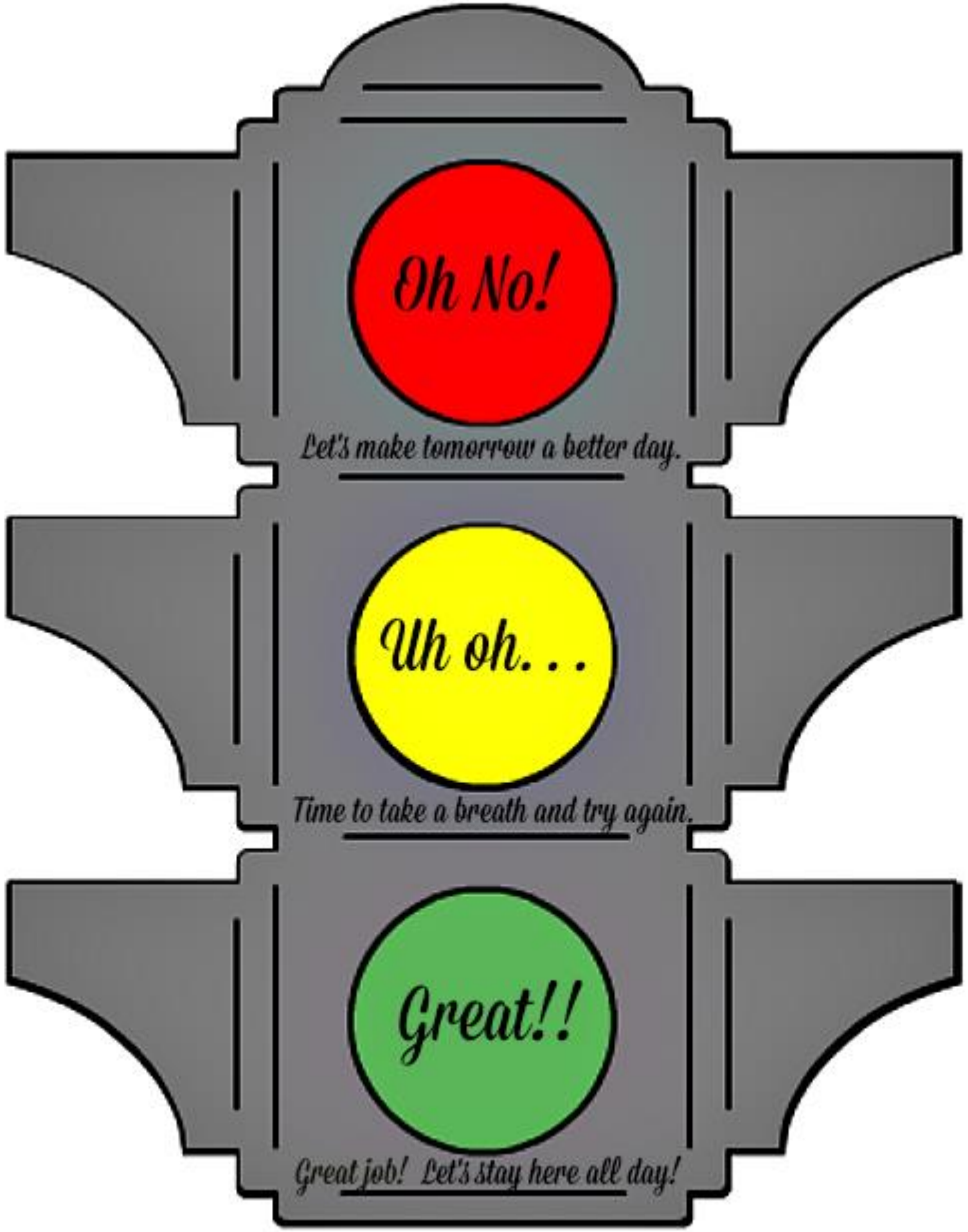


Agile Risk Self Assessment

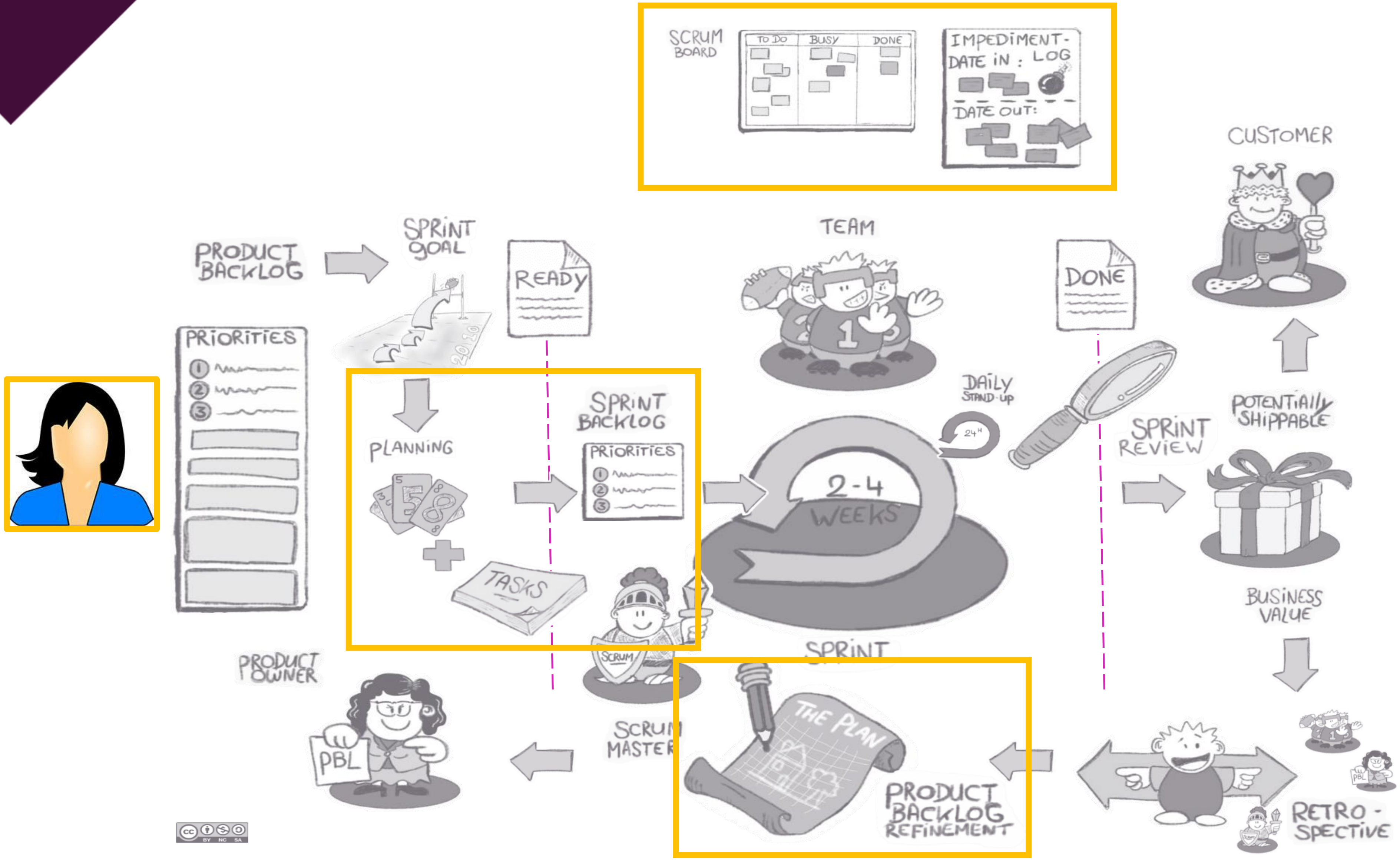


Impact	Probability					
	5	🟧	🟧	🟥	🟥	🟥 ●
	4	🟩	🟧	🟧 ●	🟥	🟥 ●
	3	🟩	🟩	🟧	🟧 ●	🟥
	2	🟩	🟩	🟩	🟧	🟧
	1	🟩	🟩	🟩	🟩	🟧
		A	B	C	D	E

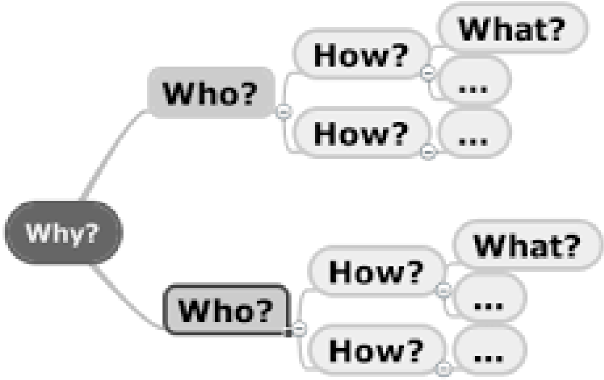
Define thresholds

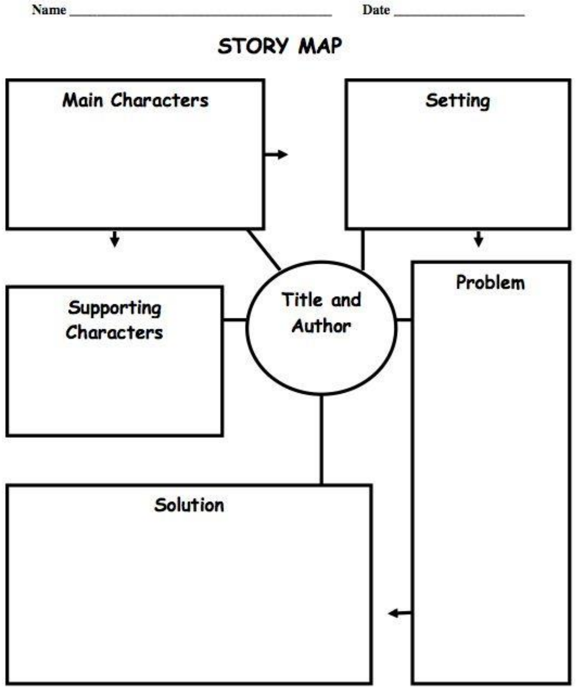

Oh No!
Let's make tomorrow a better day.
Uh oh...
Time to take a breath and try again.
Great!!
Great job! Let's stay here all day!

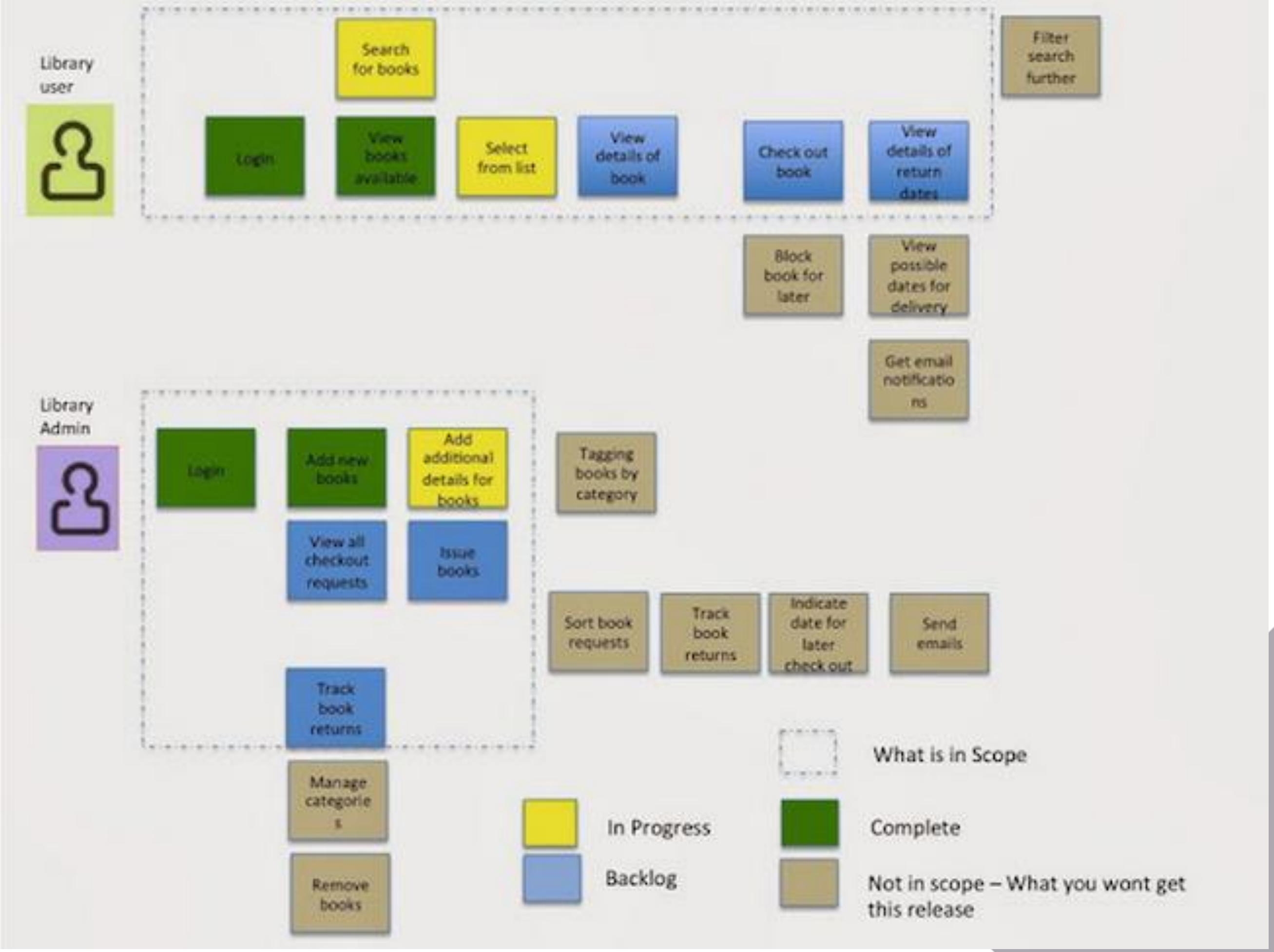
Security Stakeholdership



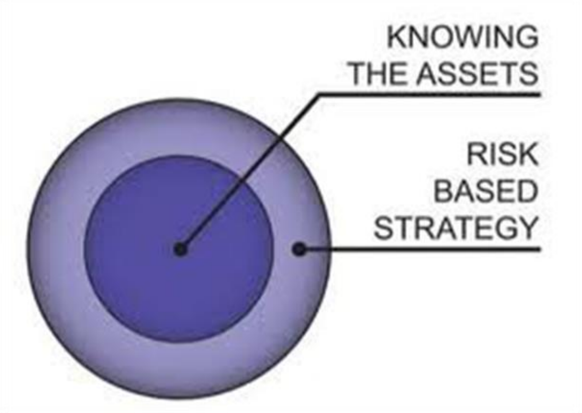
Story mapping

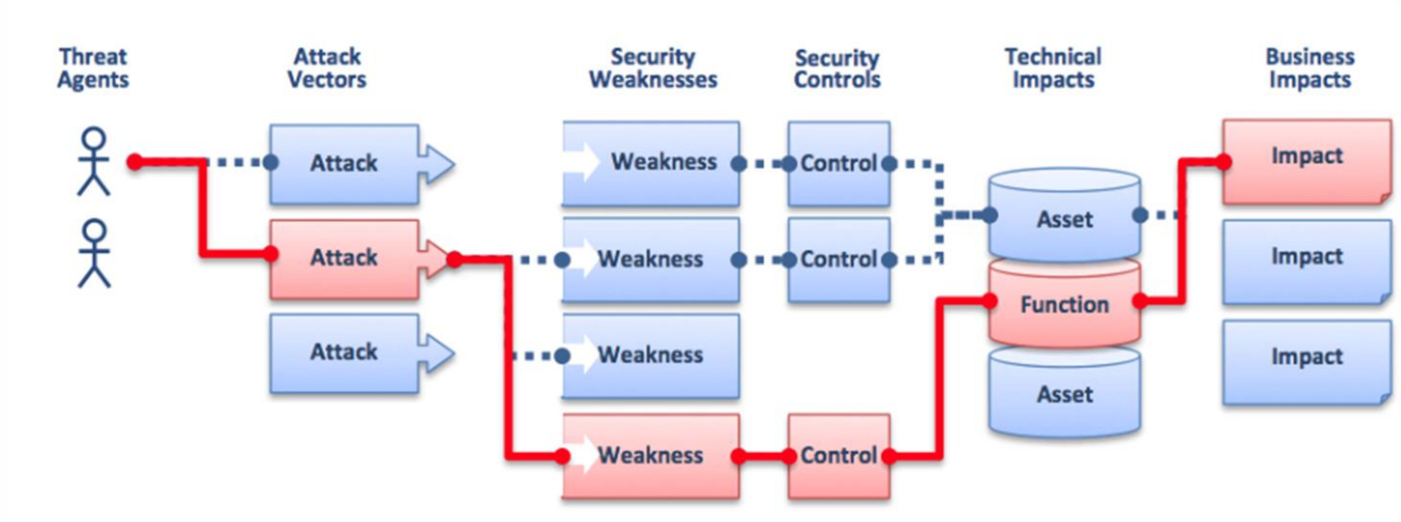


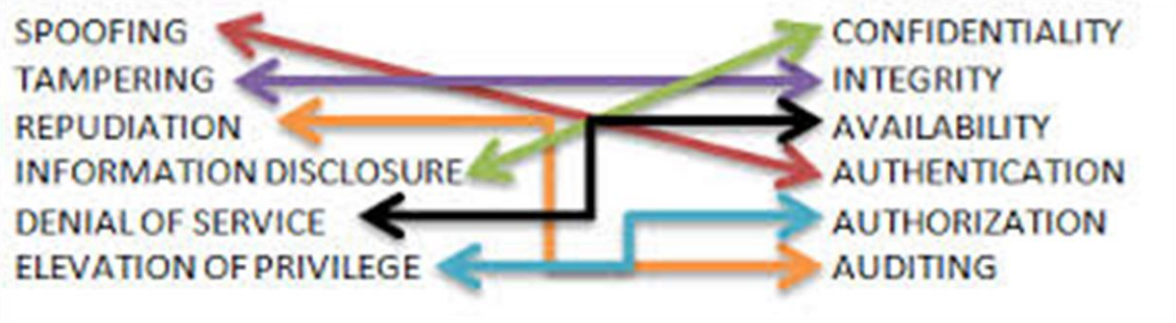




Threat model the story map







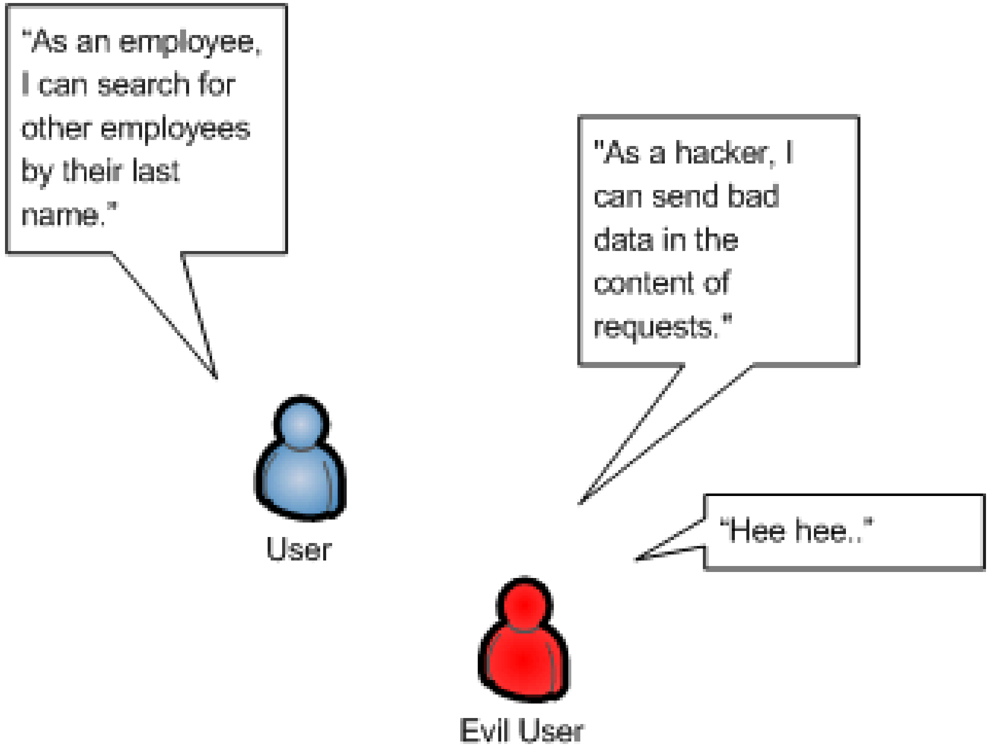
Adding Security User Stories

User Story

As a customer, **I want to** track the shipment of my order **so that** I know when it will arrive.

Security Story

As a fraudster, **I want to** see the details of an order that is not my own **so that** I can learn another person's private information.



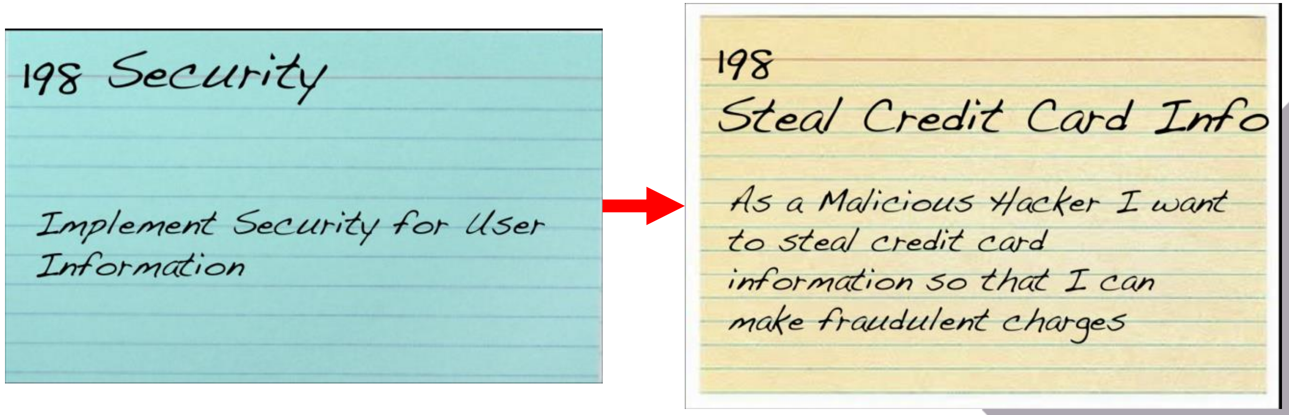
"As an employee, I can search for other employees by their last name."

User

"As a hacker, I can send bad data in the content of requests."

"Hee hee.."

Evil User



198 Security

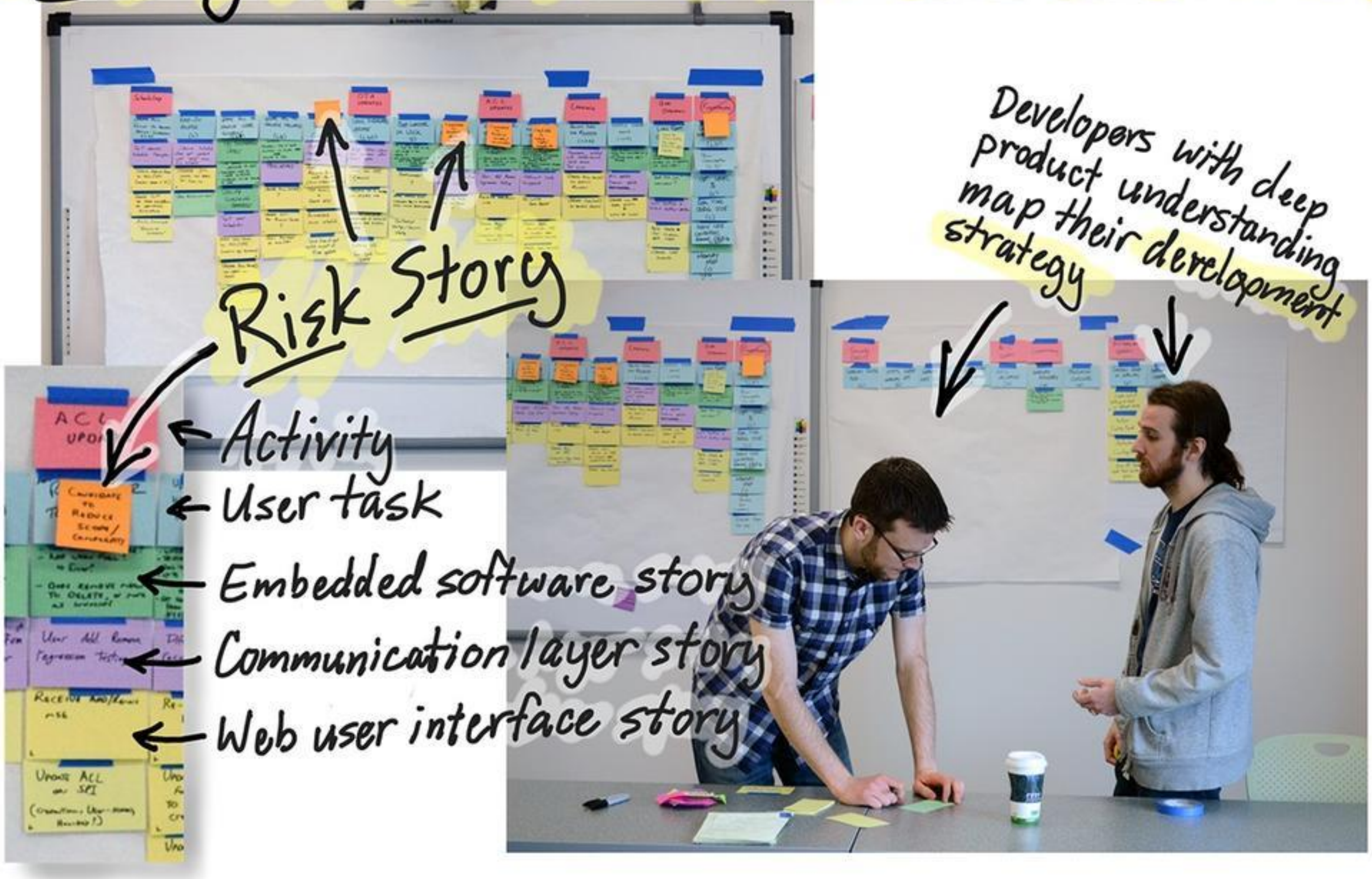
Implement Security for User Information

198

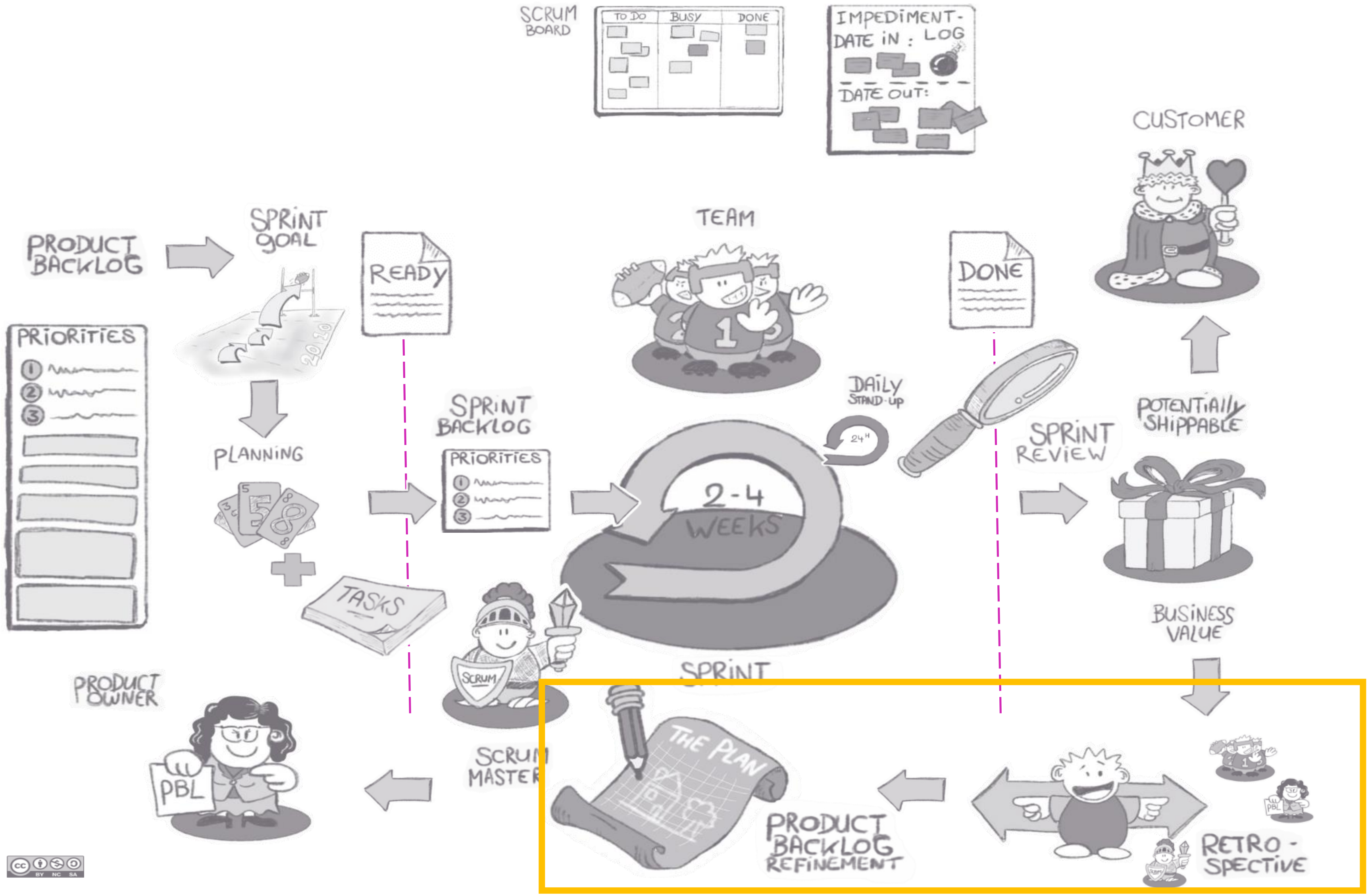
Steal Credit Card Info

As a Malicious Hacker I want to steal credit card information so that I can make fraudulent charges

Adding risk stories to make risk visible


Risk Story
Activity
User task
Embedded software story
Communication layer story
Web user interface story
Developers with deep product understanding map their development strategy
ACL UPDATES

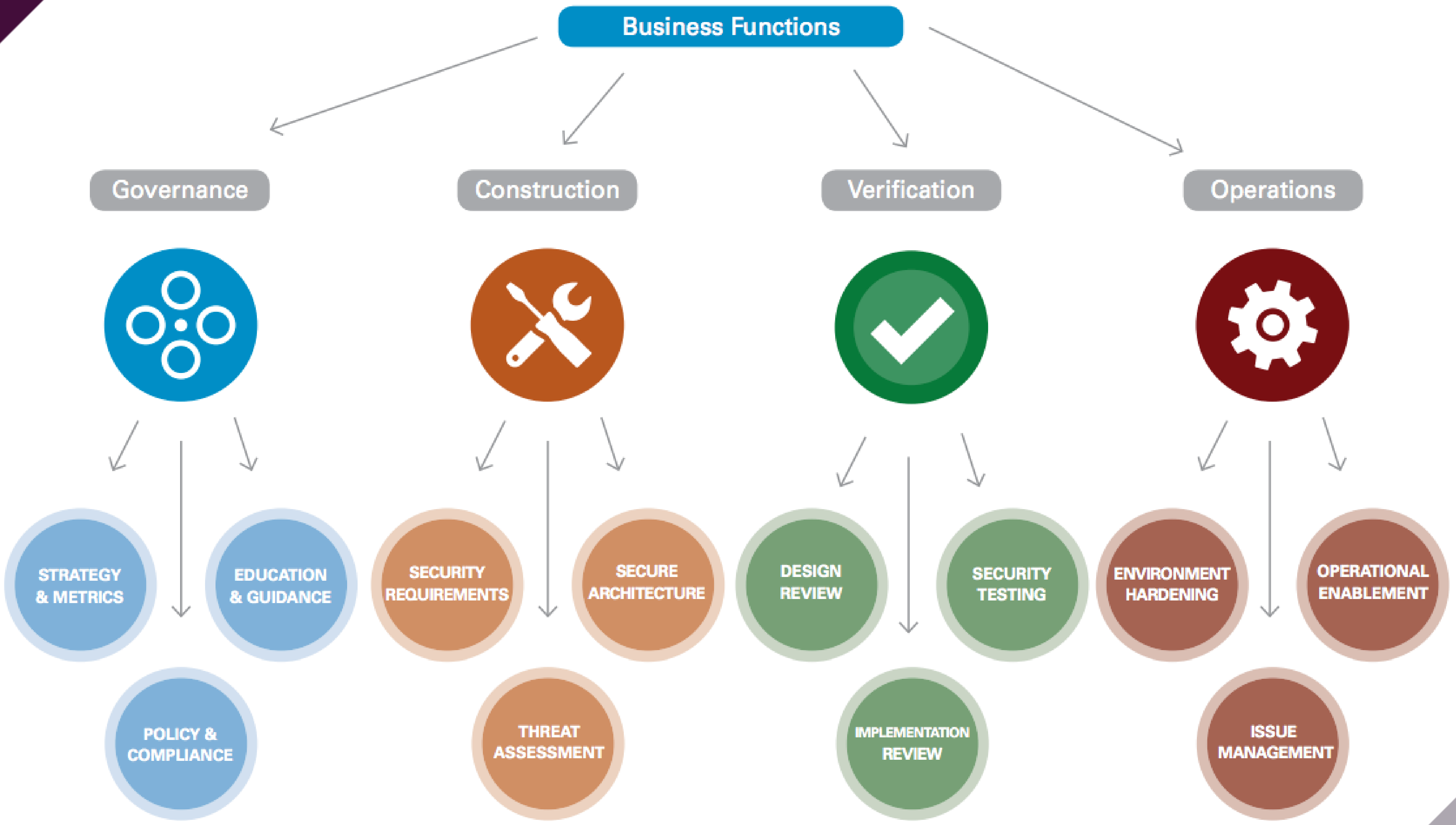
Security Education



Make a plan

SOFTWARE DEVELOPMENT

- SAMM Overview -



Business Functions

Governance

- STRATEGY & METRICS
- POLICY & COMPLIANCE
- EDUCATION & GUIDANCE

Construction

- SECURITY REQUIREMENTS
- THREAT ASSESSMENT
- SECURE ARCHITECTURE

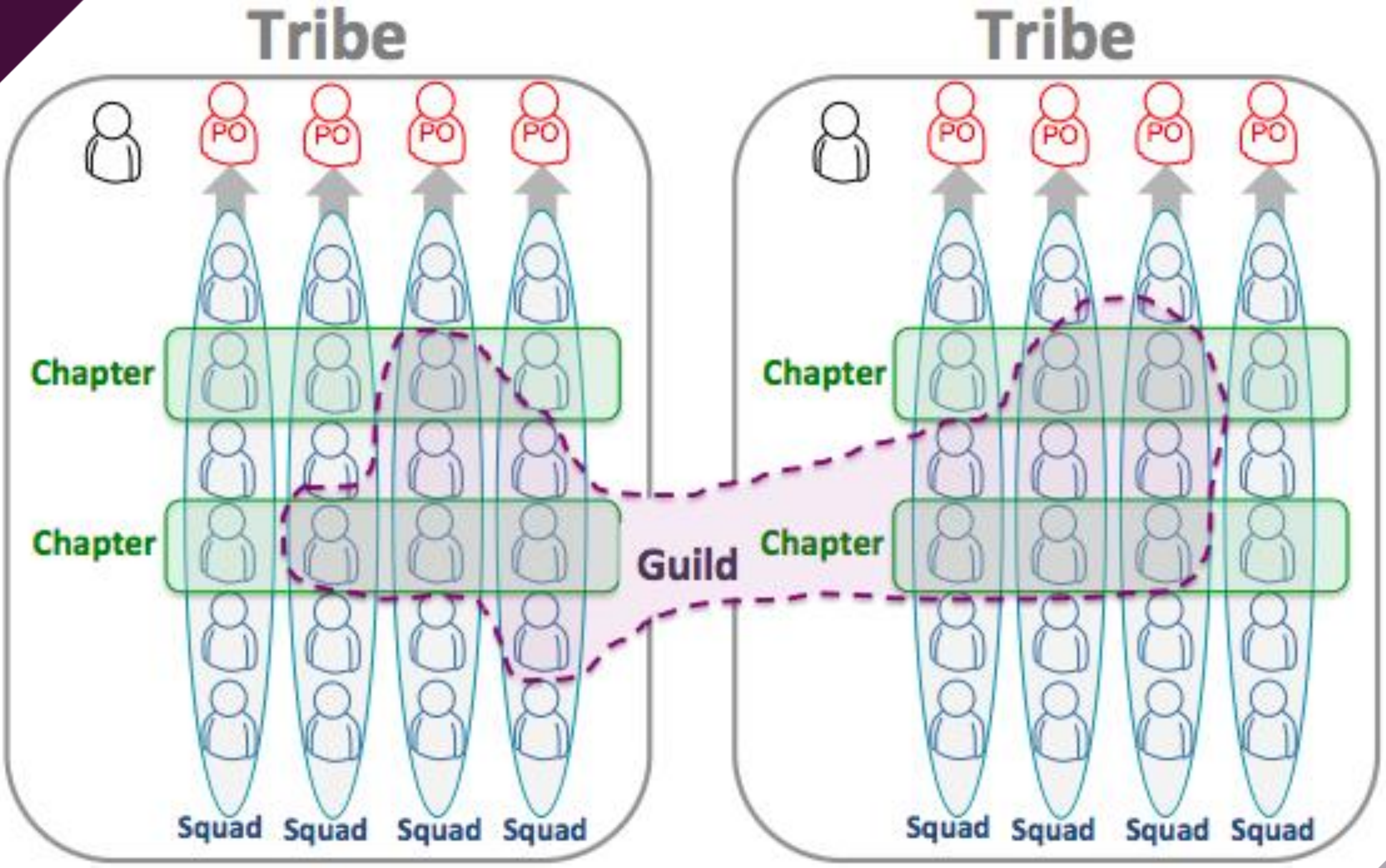
Verification

- DESIGN REVIEW
- IMPLEMENTATION REVIEW
- SECURITY TESTING

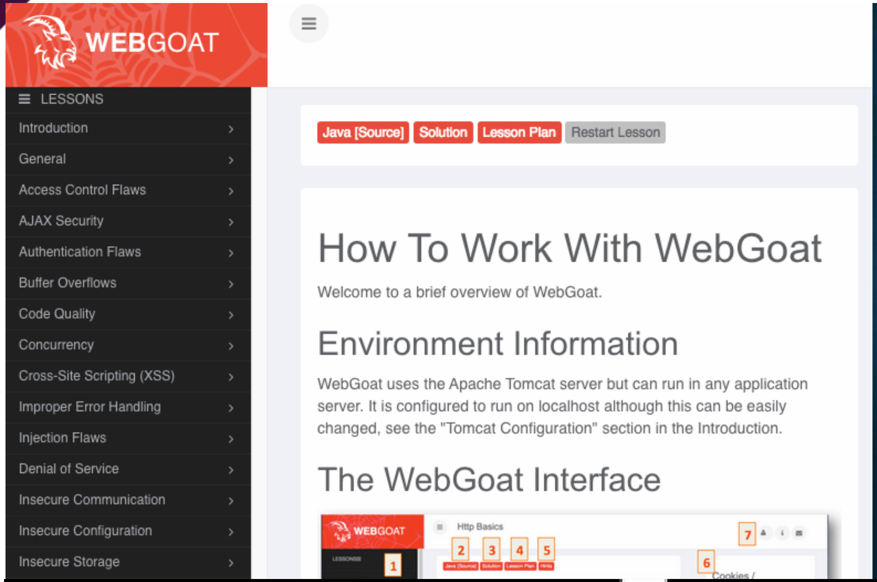
Operations

- ENVIRONMENT HARDENING
- ISSUE MANAGEMENT
- OPERATIONAL ENABLEMENT

Start a Security Guild



Start hacking yourself



WEBGOAT

LESSONS

- Introduction
- General
- Access Control Flaws
- AJAX Security
- Authentication Flaws
- Buffer Overflows
- Code Quality
- Concurrency
- Cross-Site Scripting (XSS)
- Improper Error Handling
- Injection Flaws
- Denial of Service
- Insecure Communication
- Insecure Configuration
- Insecure Storage

Java [Source] | Solution | Lesson Plan | Restart Lesson

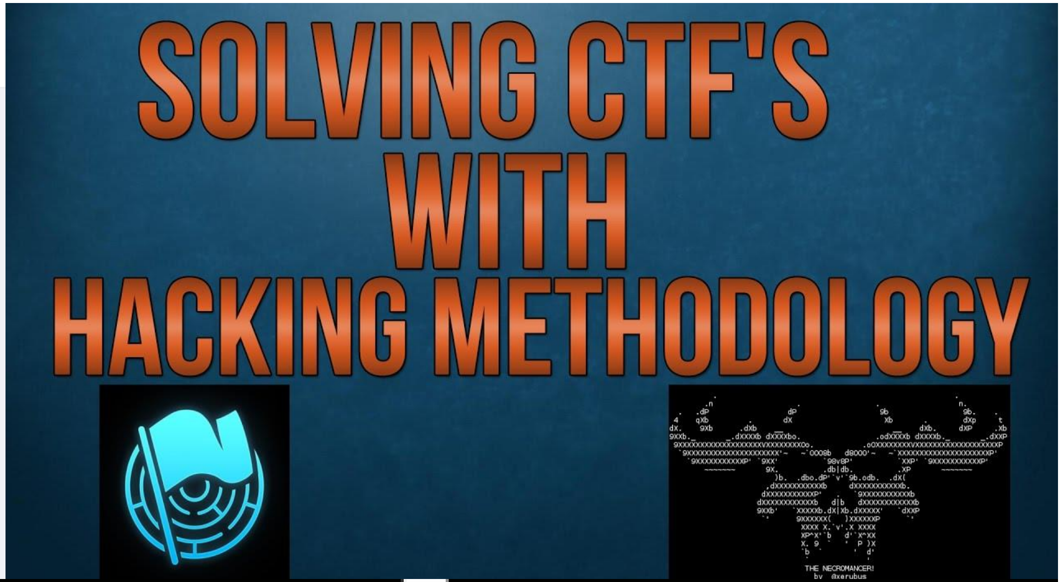
How To Work With WebGoat

Welcome to a brief overview of WebGoat.

Environment Information

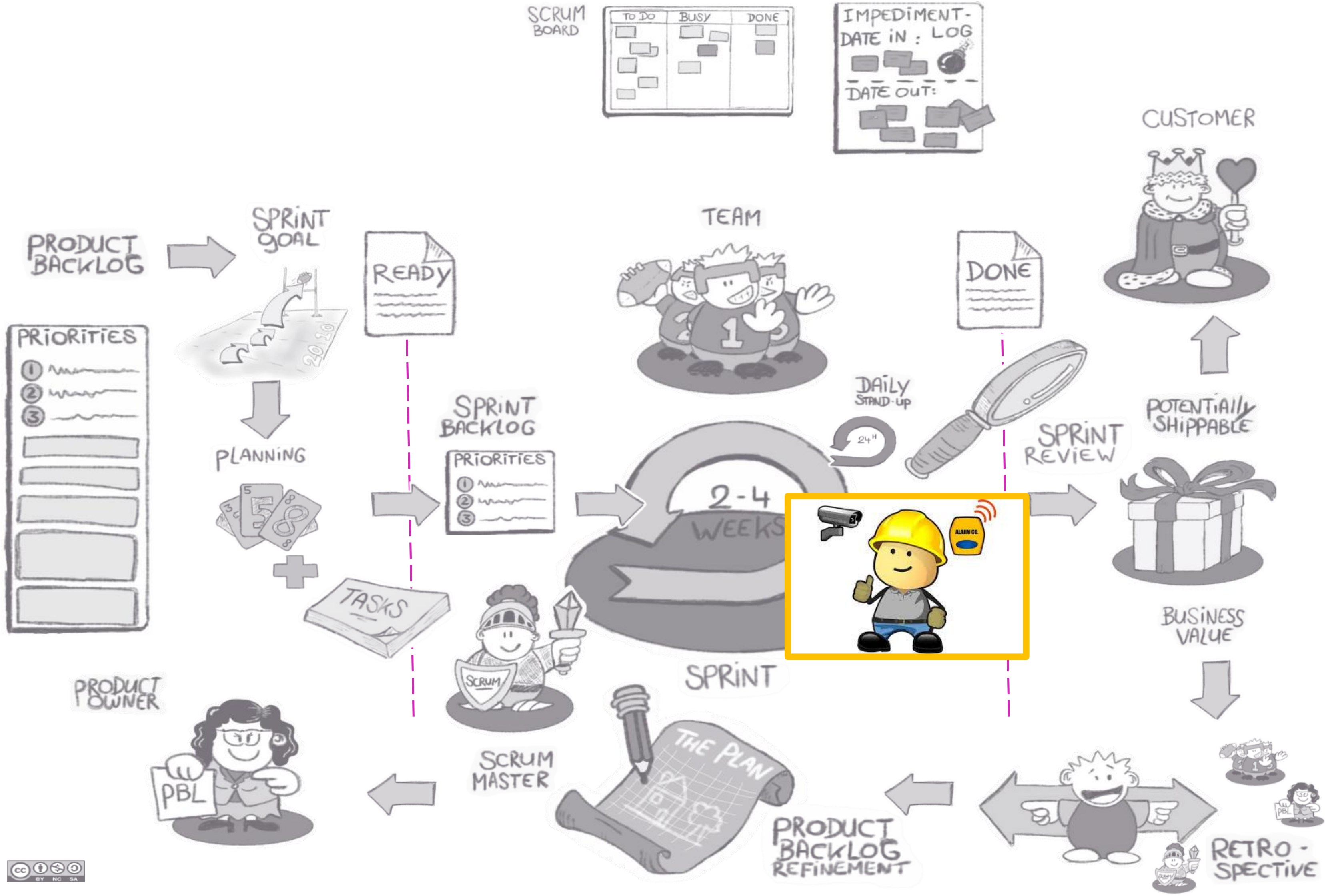
WebGoat uses the Apache Tomcat server but can run in any application server. It is configured to run on localhost although this can be easily changed, see the "Tomcat Configuration" section in the Introduction.

The WebGoat Interface

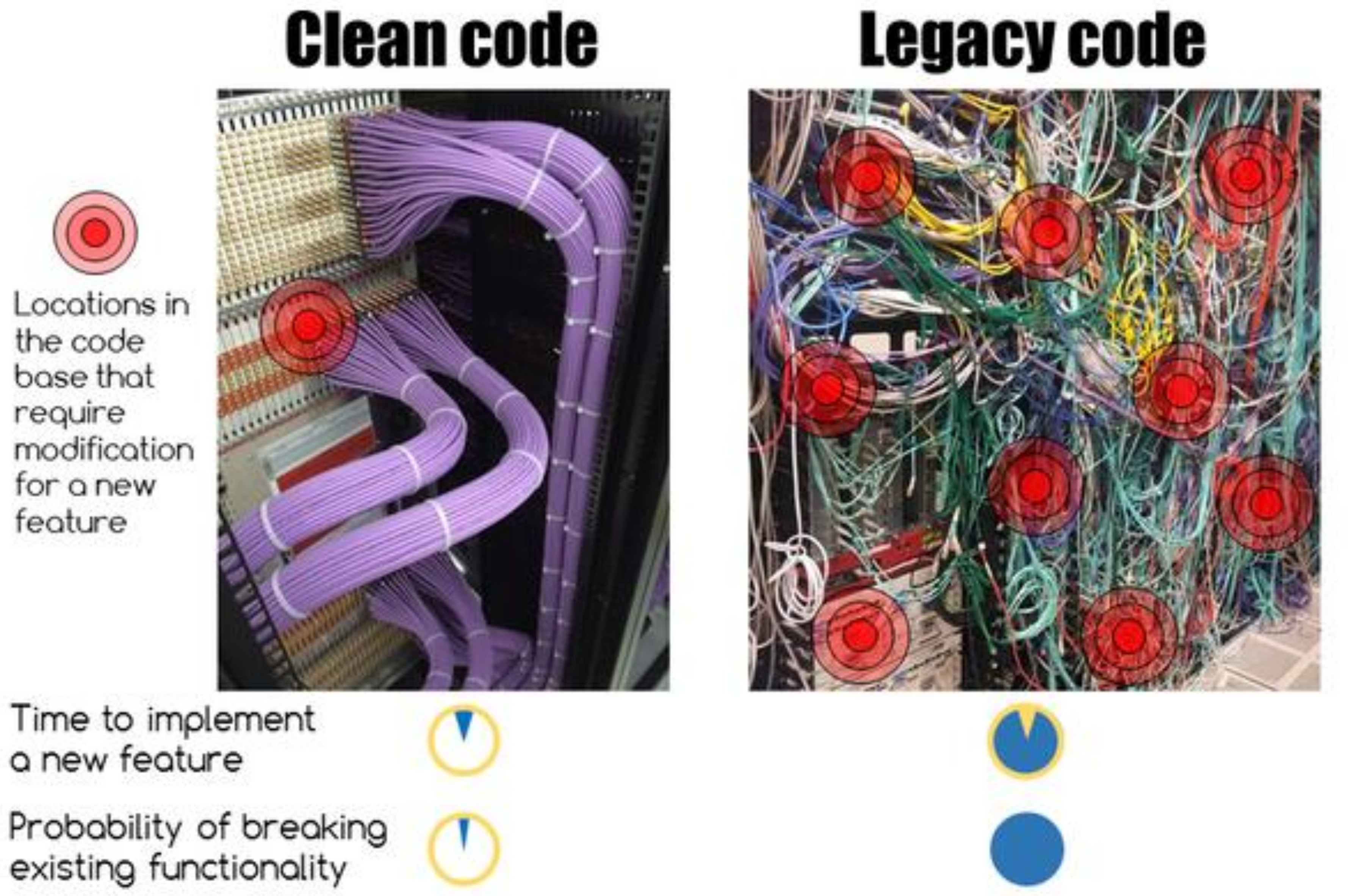


OUR GOAL IS TO WRITE BUG-FREE SOFTWARE. I'LL PAY A TEN-DOLLAR BONUS FOR EVERY BUG YOU FIND AND FIX.
S. Adams E-mail: SCOTTADAMS@AOL.COM
YAHOO!
WE'RE RICH
YES!!! YES!!! YES!!!
11/13 © 1995 United Feature Syndicate, Inc.(NYC)
I HOPE THIS DRIVES THE RIGHT BEHAVIOR.
I'M GONNA WRITE ME A NEW MINIVAN THIS AFTER-NOON!

Security Engineering



Why patching is hard



Clean code

Legacy code

Locations in the code base that require modification for a new feature

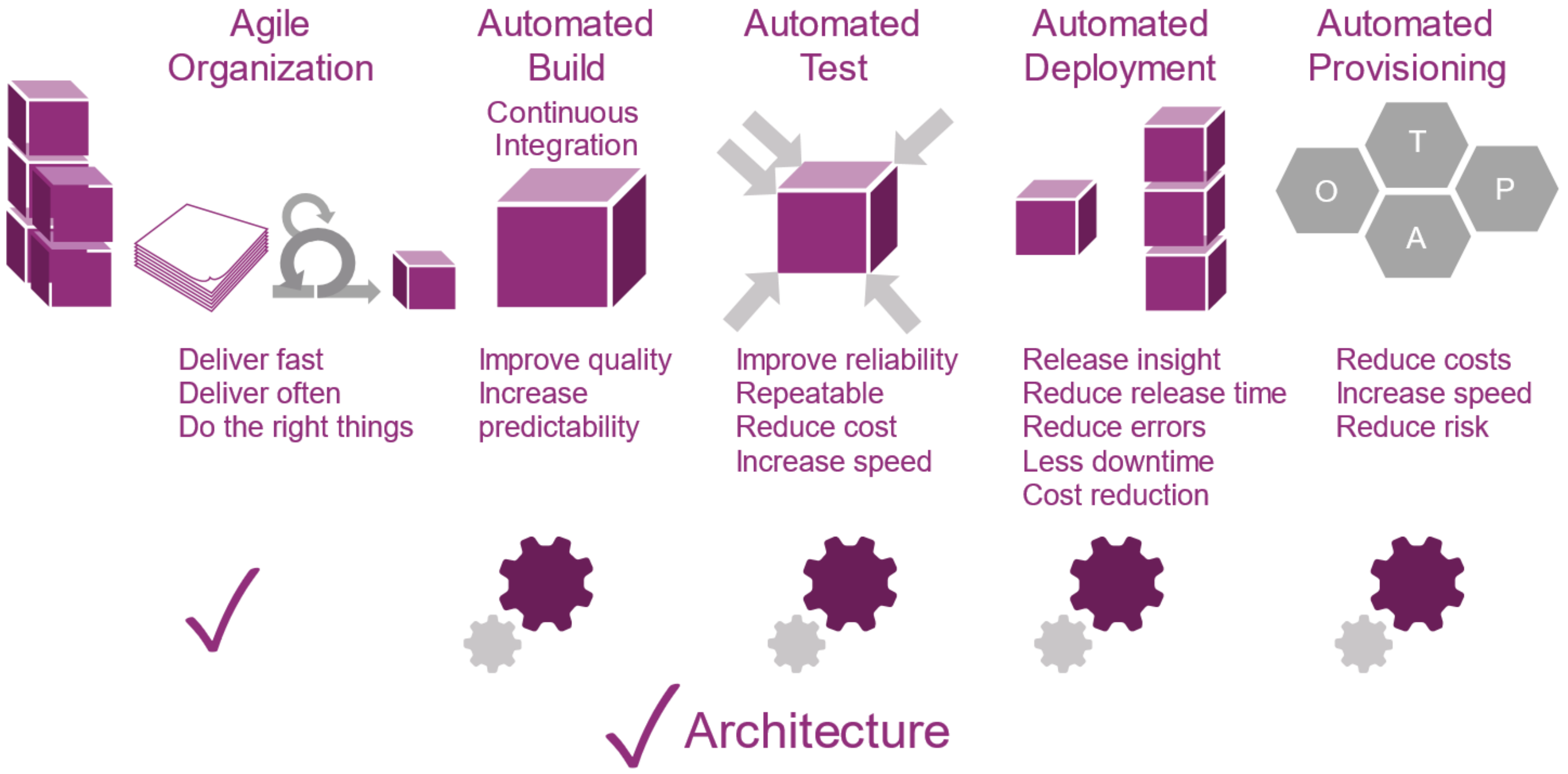
Time to implement a new feature

Probability of breaking existing functionality

Classical deployments

DevOps = automation

FULLY AUTOMATED SOFTWARE DELIVERY PROCESS



Agile Organization

Deliver fast
Deliver often
Do the right things

✓

Automated Build

Continuous Integration

Improve quality
Increase predictability

Automated Test

Improve reliability
Repeatable
Reduce cost
Increase speed

Automated Deployment

Release insight
Reduce release time
Reduce errors
Less downtime
Cost reduction

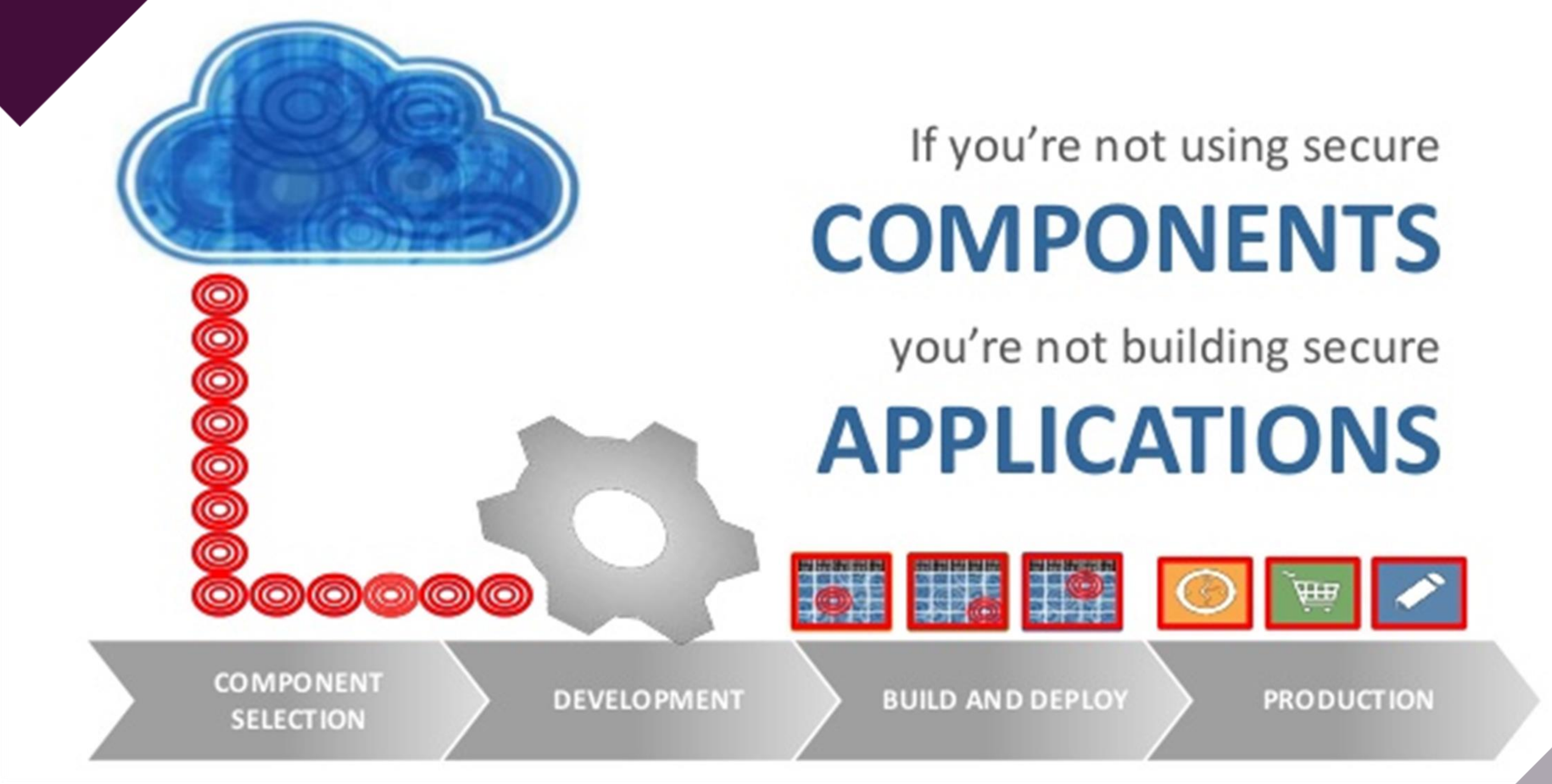
Automated Provisioning

Reduce costs
Increase speed
Reduce risk

✓ Architecture

OTAP in DevOps

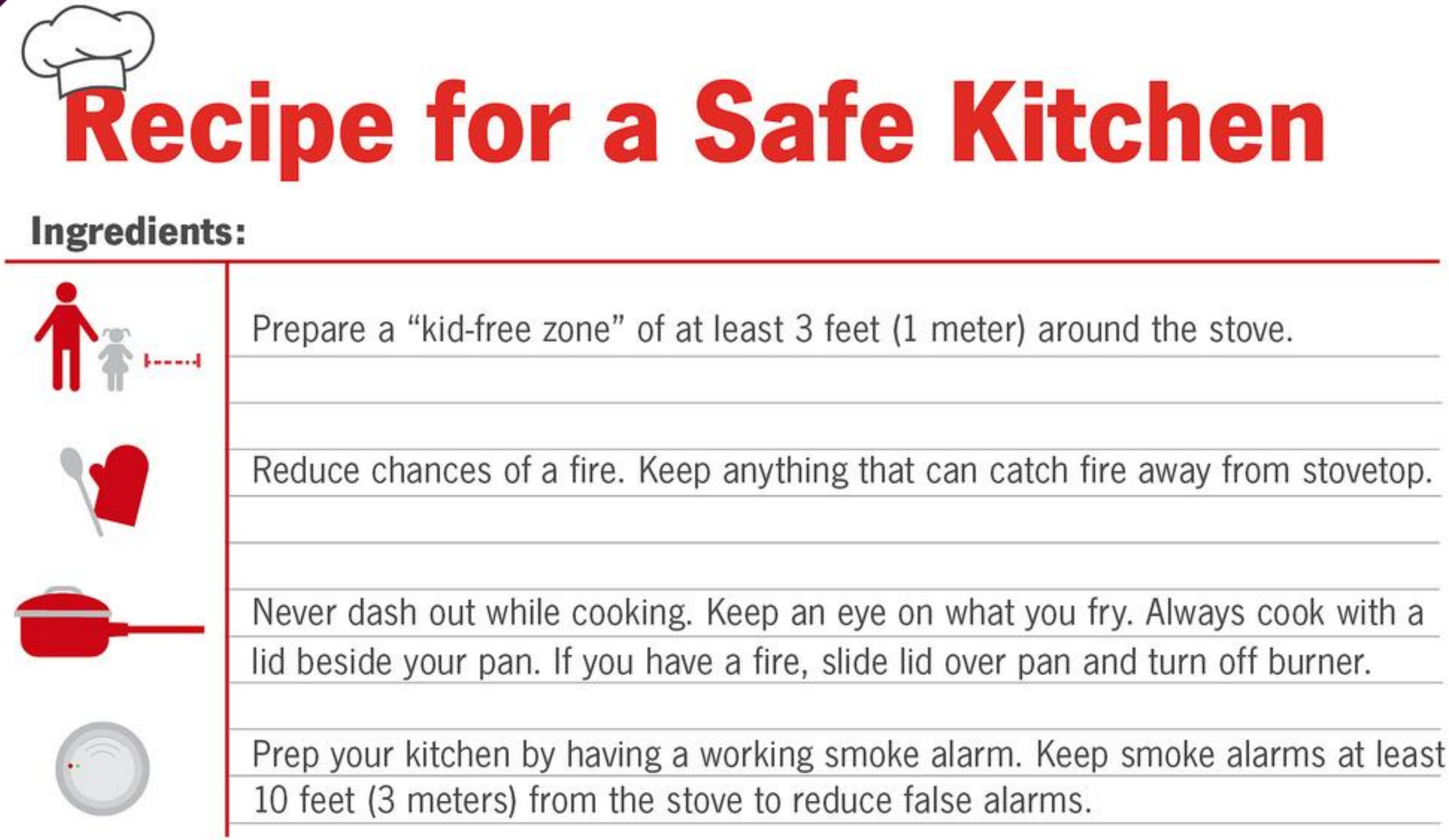
Use Secure Building blocks



If you're not using secure **COMPONENTS** you're not building secure **APPLICATIONS**

Provide sandboxes for experiments

Set some groundrules



Recipe for a Safe Kitchen

Ingredients:

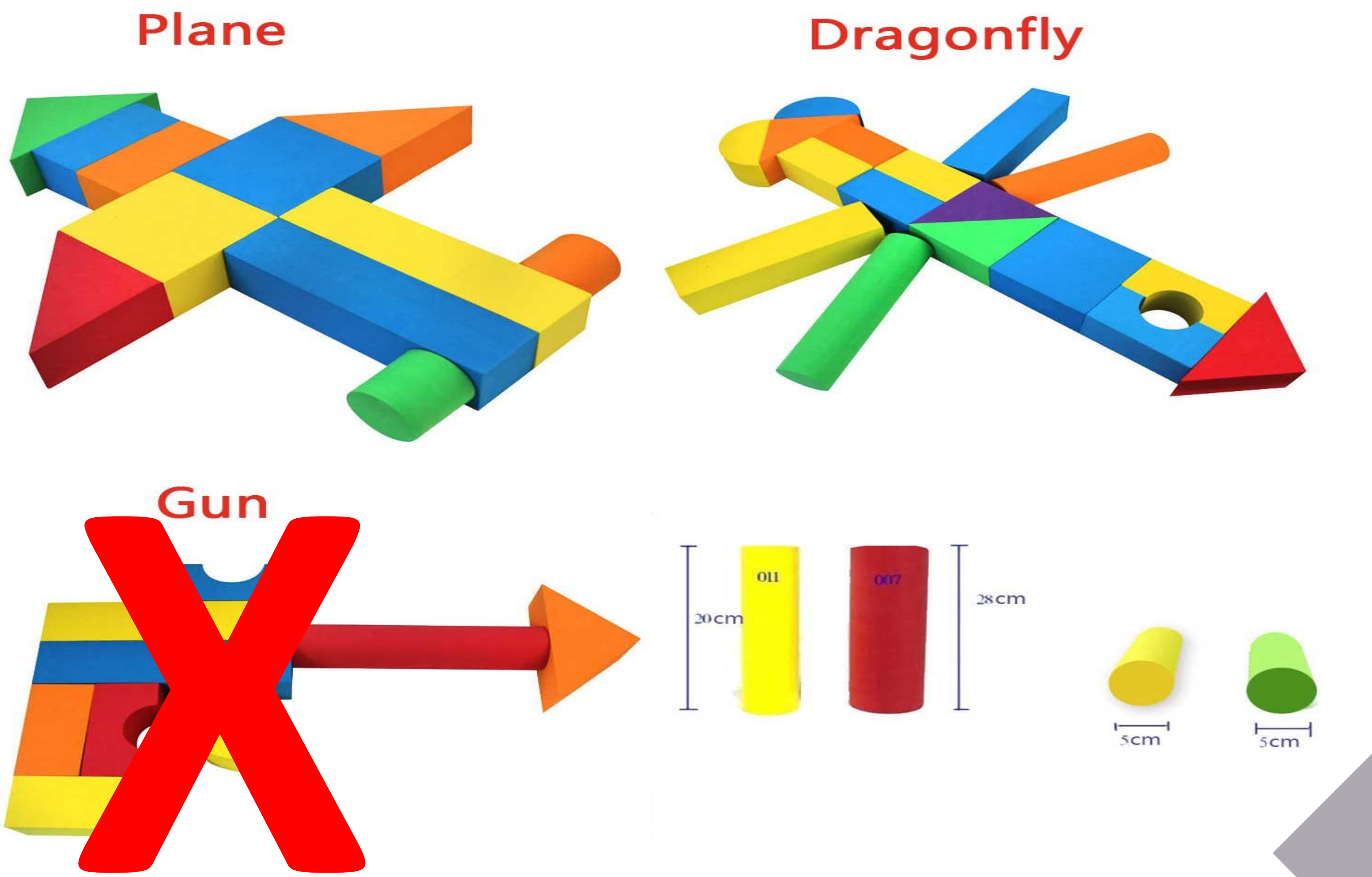
Prepare a “kid-free zone” of at least 3 feet (1 meter) around the stove.

Reduce chances of a fire. Keep anything that can catch fire away from stovetop.

Never dash out while cooking. Keep an eye on what you fry. Always cook with a lid beside your pan. If you have a fire, slide lid over pan and turn off burner.

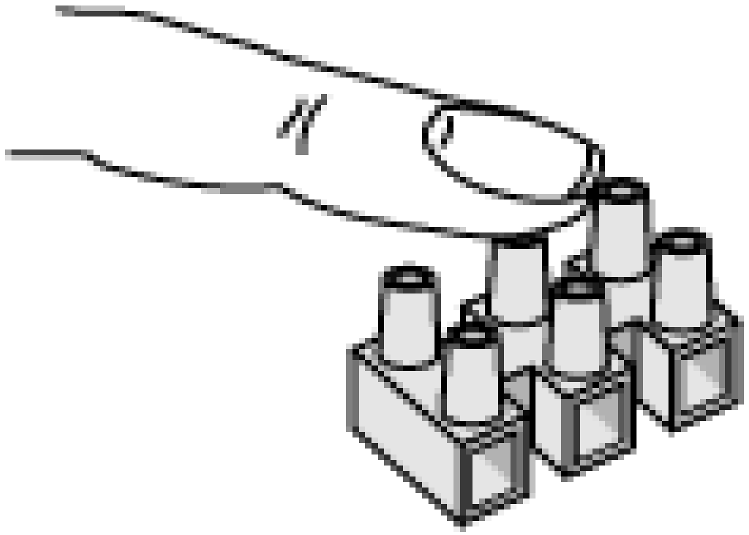
Prep your kitchen by having a working smoke alarm. Keep smoke alarms at least 10 feet (3 meters) from the stove to reduce false alarms.

Set some groundrules

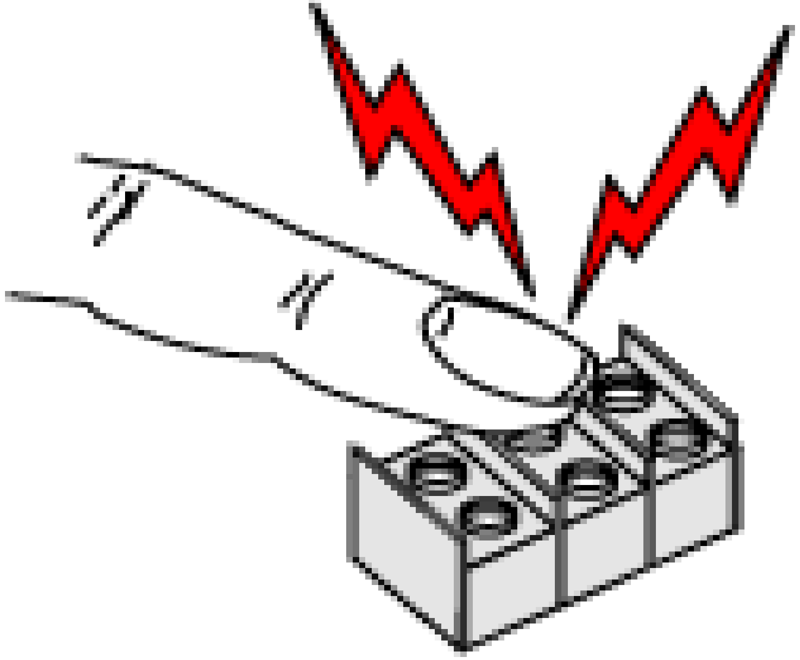


Assess new blocks

The International terminal block has recessed screws and terminals



International terminal block



North American terminal block

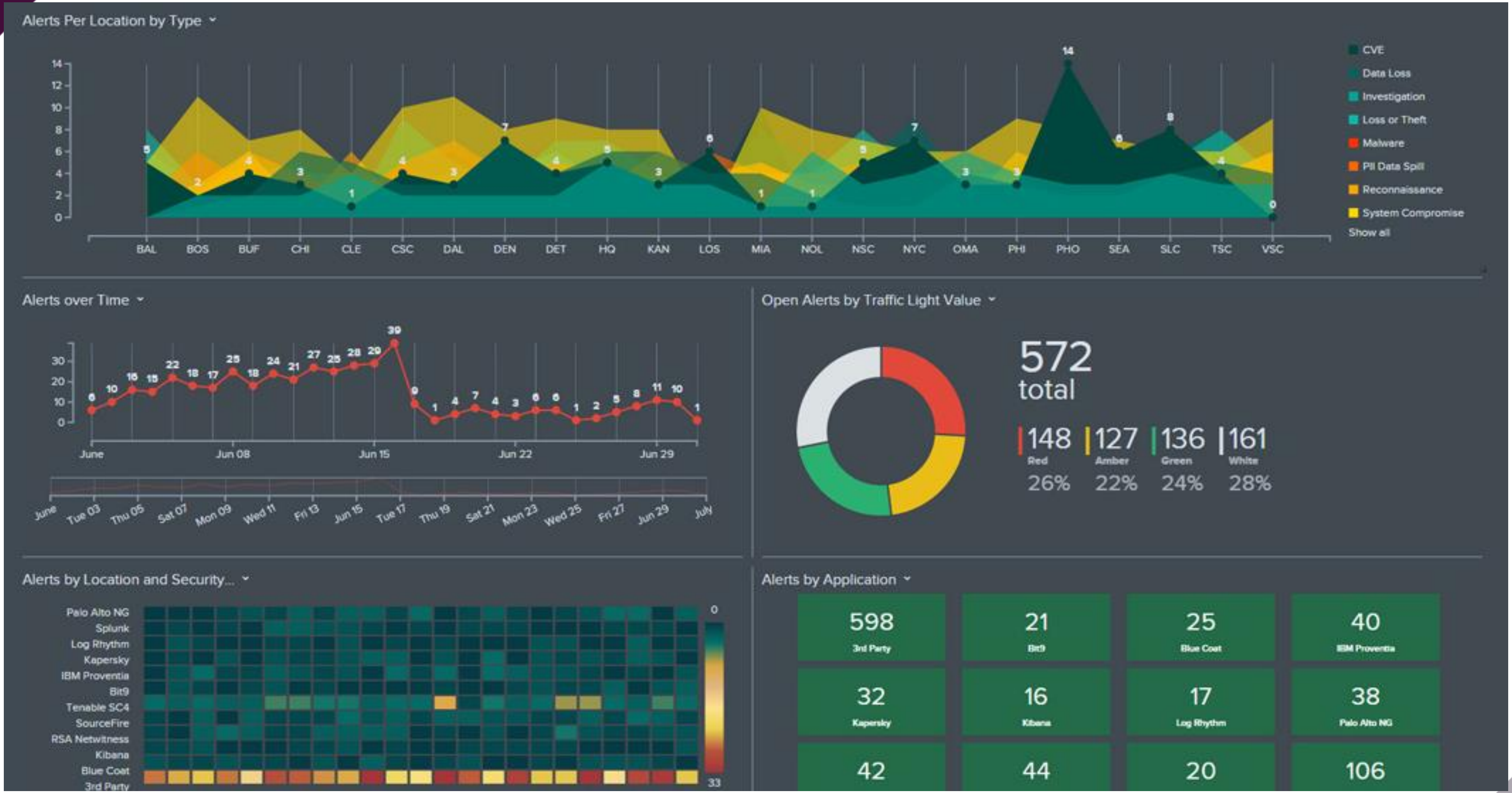
Patching 2.0

Incident response

INCIDENT RESPONSE TEAM

meaning, definition, explanation...

Monitoring

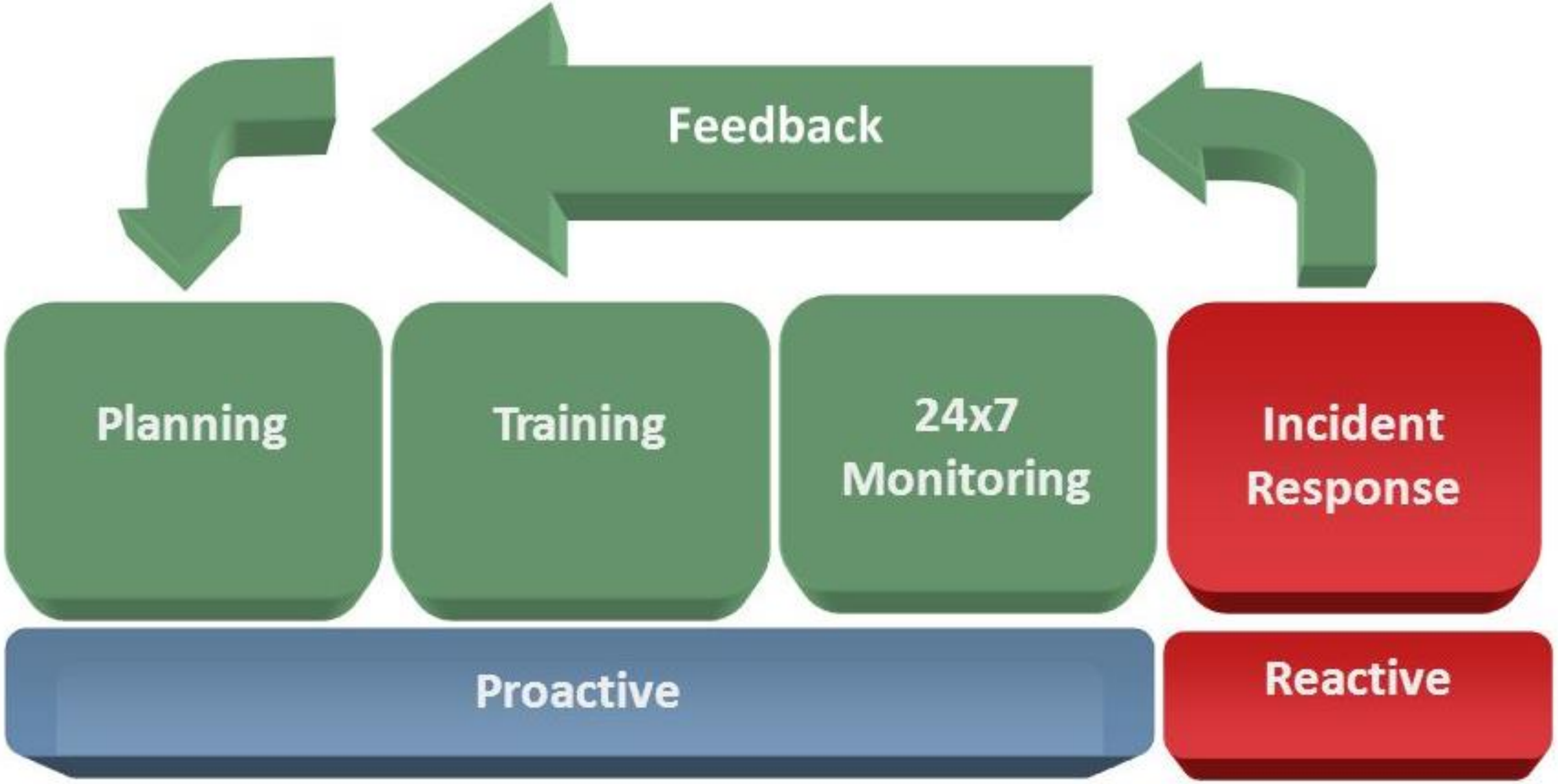


572 total

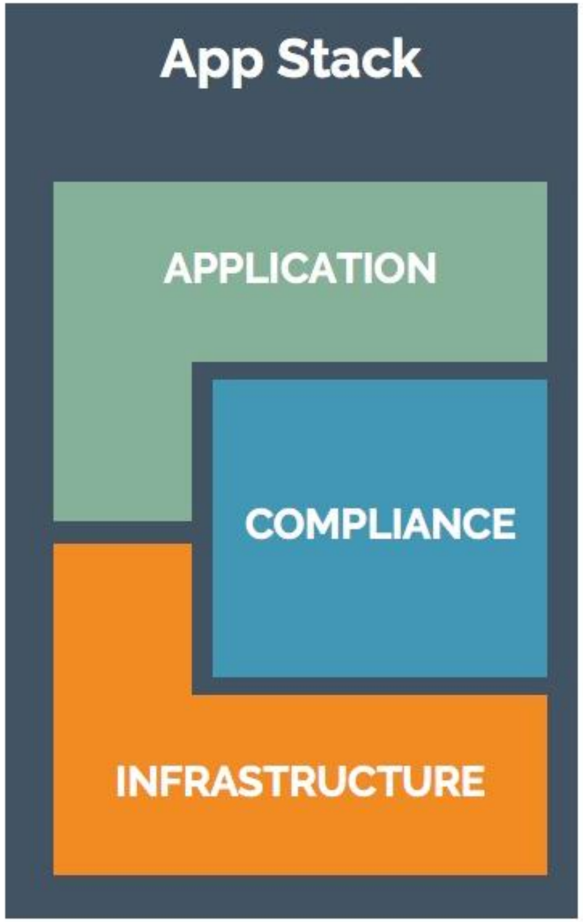
148	127	136	161
Red	Amber	Green	White
26%	22%	24%	28%

598 3rd Party	21 Bit9	25 Blue Coat	40 IBM Proventia
32 Kapersky	16 Kibana	17 Log Rhythm	38 Palo Alto NG
42	44	20	106

Make it a feedback loop



Managing the Application stack as code



Complete application lifecycle managed as code with a frictionless path from laptop to production

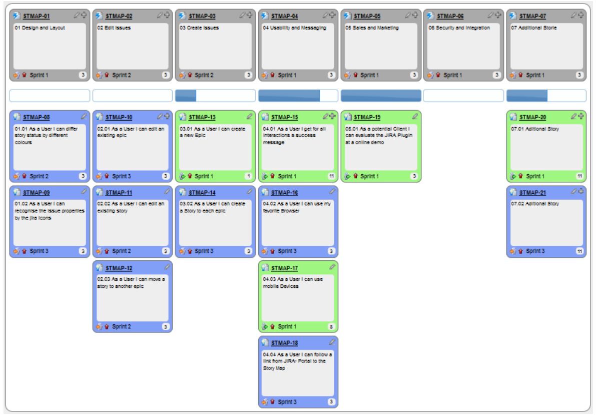
Codify how the application is built, how it runs, and all of its dependencies to free the app from underlying infrastructure

Define policies as code to detect issues before production and discover non-compliance for fast remediation

Manage infrastructure as code to provision, harden, and maintain configuration state

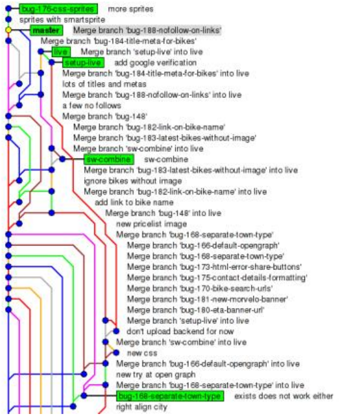
Audit Automation

Mapping



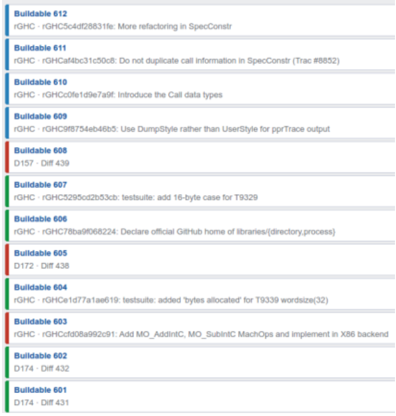
STMAP-01 01 Design and Layout Sprint 1 · 3	STMAP-02 02 Edit Issues Sprint 2 · 3	STMAP-03 03 Create Issues Sprint 2 · 3	STMAP-04 04 Usability and Messaging Sprint 1 · 3	STMAP-05 05 Sales and Marketing Sprint 1 · 3	STMAP-06 06 Security and Integration Sprint 1 · 3	STMAP-07 07 Additional Storie Sprint 1 · 3
STMAP-08 01.01 As a User I can differ story status by different colours Sprint 2 · 3	STMAP-10 02.01 As a User I can edit an existing epic Sprint 3 · 3	STMAP-13 03.01 As a User I can create a new Epic Sprint 1 · 1	STMAP-15 04.01 As a User I get for all interactions a success message Sprint 1 · 11	STMAP-19 05.01 As a potential Client I can evaluate the JIRA Plugin at a online demo Sprint 1 · 3		STMAP-20 07.01 Aditional Story Sprint 1 · 11
STMAP-09 01.02 As a User I can recognise the issue properties by the jira icons Sprint 3 · 3	STMAP-11 02.02 As a User I can edit an existing story Sprint 2 · 3	STMAP-14 03.02 As a User I can create a Story to each epic Sprint 3 · 3	STMAP-16 04.02 As a User I can use my favorite Browser Sprint 3 · 3			STMAP-21 07.02 Aditional Story Sprint 3 · 11
	STMAP-12 02.03 As a User I can move a story to another epic Sprint 2 · 3		STMAP-17 04.03 As a User I can use mobile Devices Sprint 1 · 8			
			STMAP-18 04.04 As a User I can follow a link from JIRA- Portal to the Story Map Sprint 3 · 3			

Version control



```
bug-176-css-sprites   more sprites
sprites with smartsprite
master   Merge branch 'bug-188-nofollow-on-links'
Merge branch 'bug-184-title-meta-for-bikes'
live   Merge branch 'setup-live' into live
setup-live   add google verification
Merge branch 'bug-184-title-meta-for-bikes' into live
lots of titles and metas
Merge branch 'bug-188-nofollow-on-links' into live
a few no follows
Merge branch 'bug-148'
Merge branch 'bug-182-link-on-bike-name'
Merge branch 'bug-183-latest-bikes-without-image'
Merge branch 'sw-combine' into live
sw-combine   sw-combine
Merge branch 'bug-183-latest-bikes-without-image' into live
ignore bikes without image
Merge branch 'bug-182-link-on-bike-name' into live
add link to bike name
Merge branch 'bug-148' into live
new pricelist image
Merge branch 'bug-168-separate-town-type'
Merge branch 'bug-166-default-opengraph'
Merge branch 'bug-168-separate-town-type'
Merge branch 'bug-173-html-error-share-buttons'
Merge branch 'bug-175-contact-details-formatting'
Merge branch 'bug-170-bike-search-urls'
Merge branch 'bug-181-new-morvelo-banner'
Merge branch 'bug-180-eta-banner-url'
Merge branch 'setup-live' into live
don't upload backend for now
Merge branch 'sw-combine' into live
new css
Merge branch 'bug-166-default-opengraph' into live
new try at open graph
Merge branch 'bug-168-separate-town-type' into live
bug-168-separate-town-type   exists does not work either
right align city
```

Continuous Delivery



Buildable 612
rGHC · rGHC5c4df28831fe: More refactoring in SpecConstr

Buildable 611
rGHC · rGHCaf4bc31c50c8: Do not duplicate call information in SpecConstr (Trac #8852)

Buildable 610
rGHC · rGHCc0fe1d9e7a9f: Introduce the Call data types

Buildable 609
rGHC · rGHC9f8754eb46b5: Use DumpStyle rather than UserStyle for pprTrace output

Buildable 608
D157 · Diff 439

Buildable 607
rGHC · rGHC5295cd2b53cb: testsuite: add 16-byte case for T9329

Buildable 606
rGHC · rGHC78ba9f068224: Declare official GitHub home of libraries/(directory,process)

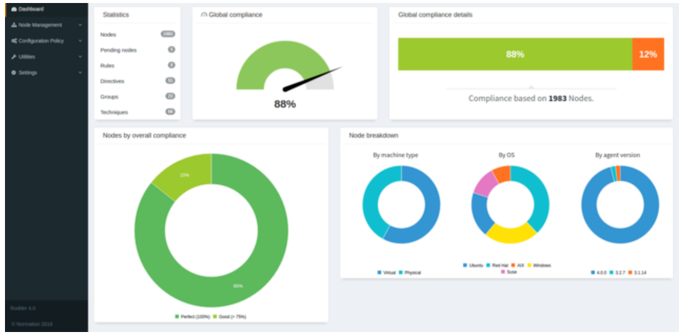
Buildable 605
D172 · Diff 438

Buildable 604
rGHC · rGHCe1d77a1ae619: testsuite: added 'bytes allocated' for T9339 wordsize(32)

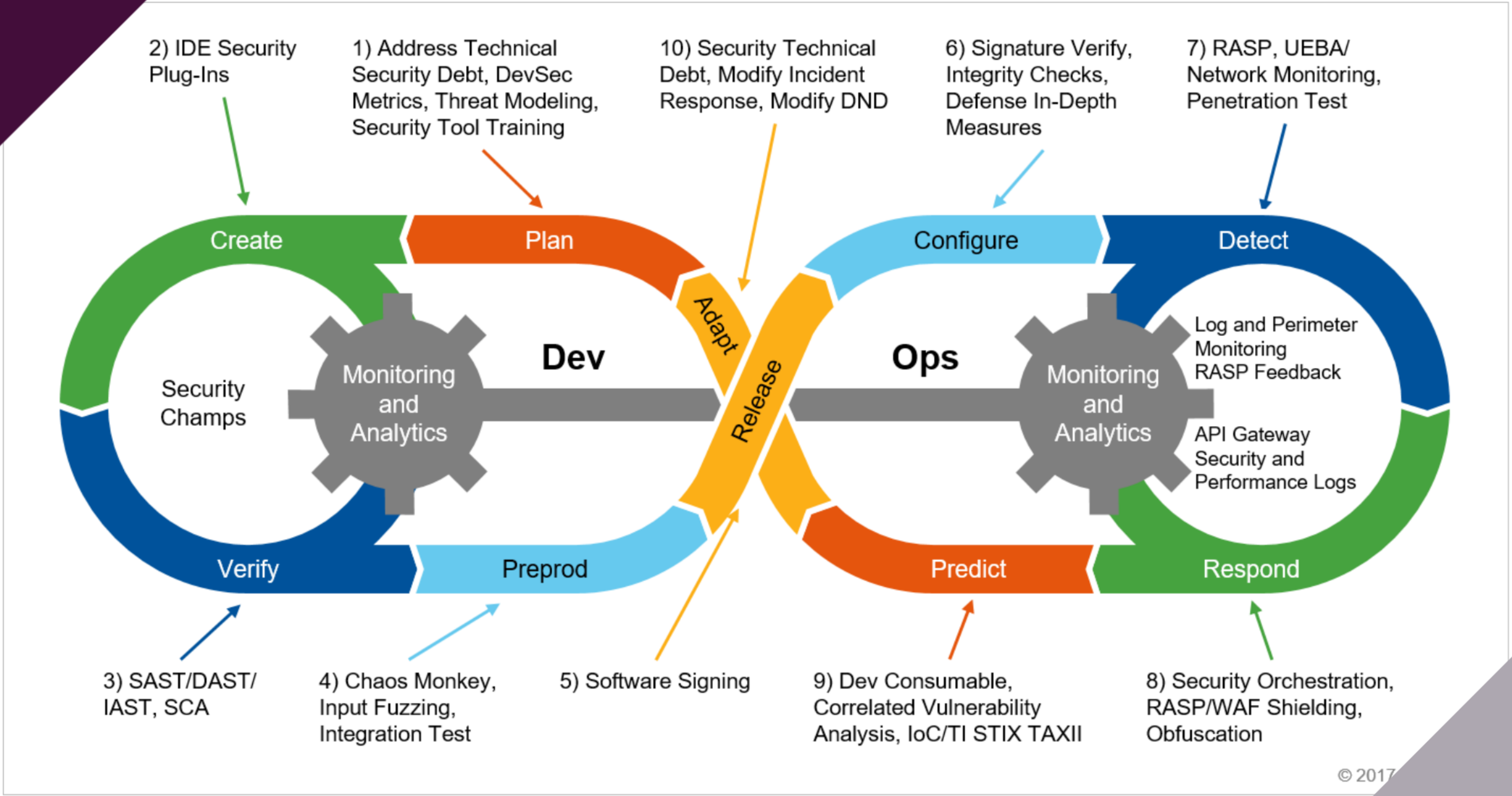
Buildable 603
rGHC · rGHCcfd08a992c91: Add MO_AddIntC, MO_SubIntC MachOps and implement in X86 backend

Buildable 602
D174 · Diff 432

Buildable 601
D174 · Diff 431



Automate everything



1) Address Technical Security Debt, DevSec Metrics, Threat Modeling, Security Tool Training

2) IDE Security Plug-Ins

3) SAST/DAST/ IAST, SCA

4) Chaos Monkey, Input Fuzzing, Integration Test

5) Software Signing

6) Signature Verify, Integrity Checks, Defense In-Depth Measures

7) RASP, UEBA/ Network Monitoring, Penetration Test

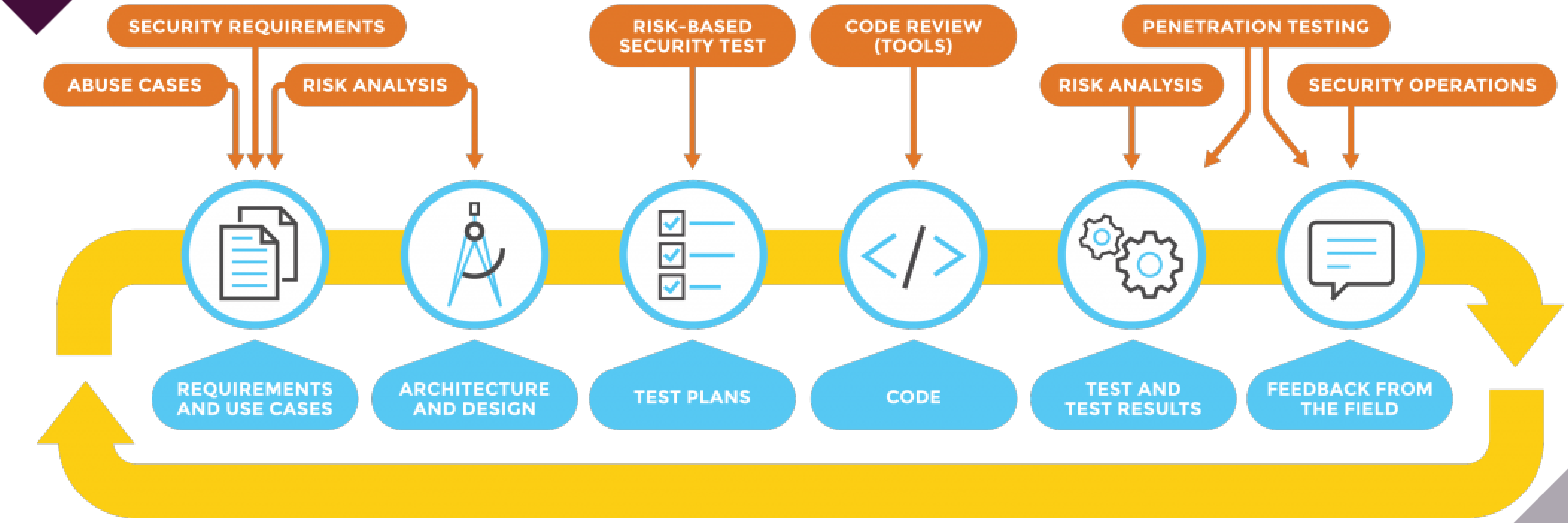
8) Security Orchestration, RASP/WAF Shielding, Obfuscation

9) Dev Consumable, Correlated Vulnerability Analysis, IoC/TI STIX TAXII

10) Security Technical Debt, Modify Incident Response, Modify DND

© 2017

Continuous security



DevSecOps: summary

- Align Dev, Sec, Bus, and Ops
- Standardize and simplify
- Automate, automate, automate
- Know your value
- Attack yourself
- Learn, teach and train

More information!

- Training ‘DevOps for CISO’

  → 19 juli & 6 september

  → https://training.xebia.com/security/devops-for-ciso

  → Discount code: PVIBCISO

- Whitepaper: ‘The IT manager guide to DevOps’

  → https://xebialabs.com/resources/whitepapers/the-it-managers-guide-to-devops/

- Whitepaper: ‘Becoming an agile security officer’

  → https://pages.xebia.com/becoming-an-agile-software-security-officer

SUCCESS
→ go get it →