Andrey Sadovykh · Dragos Truscan · Wissam Mallouli · Ana Rosa Cavalli · Cristina Seceleanu · Alessandra Bagnato Editors
CyberSecurity in a DevOps Environment
From Requirements to Monitoring
Springer

VeriDevOps Research Workshop

CyberSecurity in a DevOps Environment

Join the workshop to know the newly Launched book!

Time: 9:30-13:30 (CET)

26TH
October 2023

Online/Arrasate, Spain 2023

presented by Andrey Sadovykh

SOFTEAM UNE MARQUE DE DOCAPOSTE · Åbo Akademi University · montimage · ikerlan · Mälardalen University · ABB · FAGOR FAGOR ARRASATE

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957212

VeriDevOps

Agenda - Part 1 - Requirements

Time	Duration	Topic	Presenter	Organization
9:30	20 mins	VeriDevOps Technical Introduction	Andrey Sadovykh	SOFTEAM
Part I: Security Requirements Engineering				
9:50	20 mins	A Taxonomy of Vulnerabilities, Attacks, and Security Solutions in Industrial PLCs.	Eduard Paul Enoiu	Mälardalen University
10:10	20 mins	Natural Language Processing with Machine Learning for Security Requirements Analysis - Practical Approaches.	Andrey Sadovykh	SOFTEAM
10:30	20 mins	Security Requirements Formalization with RQCODE.	Andrey Sadovykh	SOFTEAM
10:50	10 mins	break	/	/

Agenda - Part 2 - Prevention

Part II: Prevention at Development Time				
11:00	20 mins	Vulnerability Detection and Response: Current Status and New Approaches	Jose Luis Flores	IKER
11:20	20 mins	Metamorphic Testing for Verification and Fault Localization in Industrial Control Systems	Gaadha Sudheerbabu	Åbo Akademi University
11:40	20 mins	Interactive Application Security Testing with Hybrid Fuzzing and Statistical Estimators	Ramon Barakat	FFK
12:00	10 mins	break	/	/

Agenda - Part 3 - Protection

Part III: Protection at Operations				
12:10	20 mins	CTAM: a tool for Continuous Threat Analysis and Management	Laurens Sion	KUL
12:30	20 mins	EARLY - a tool for real-time security attack detection	Tanwir Ahmad	Åbo Akademi University
12:50	20 mins	A Stream-Based Approach to Intrusion Detection	Sylvain Hallé	UM
13:10	20 mins	Towards Anomaly Detection using Explainable AI	Manh Dung	MI
13:30	10 mins	Conclusions	Andrey Sadovykh	SOFTEAM

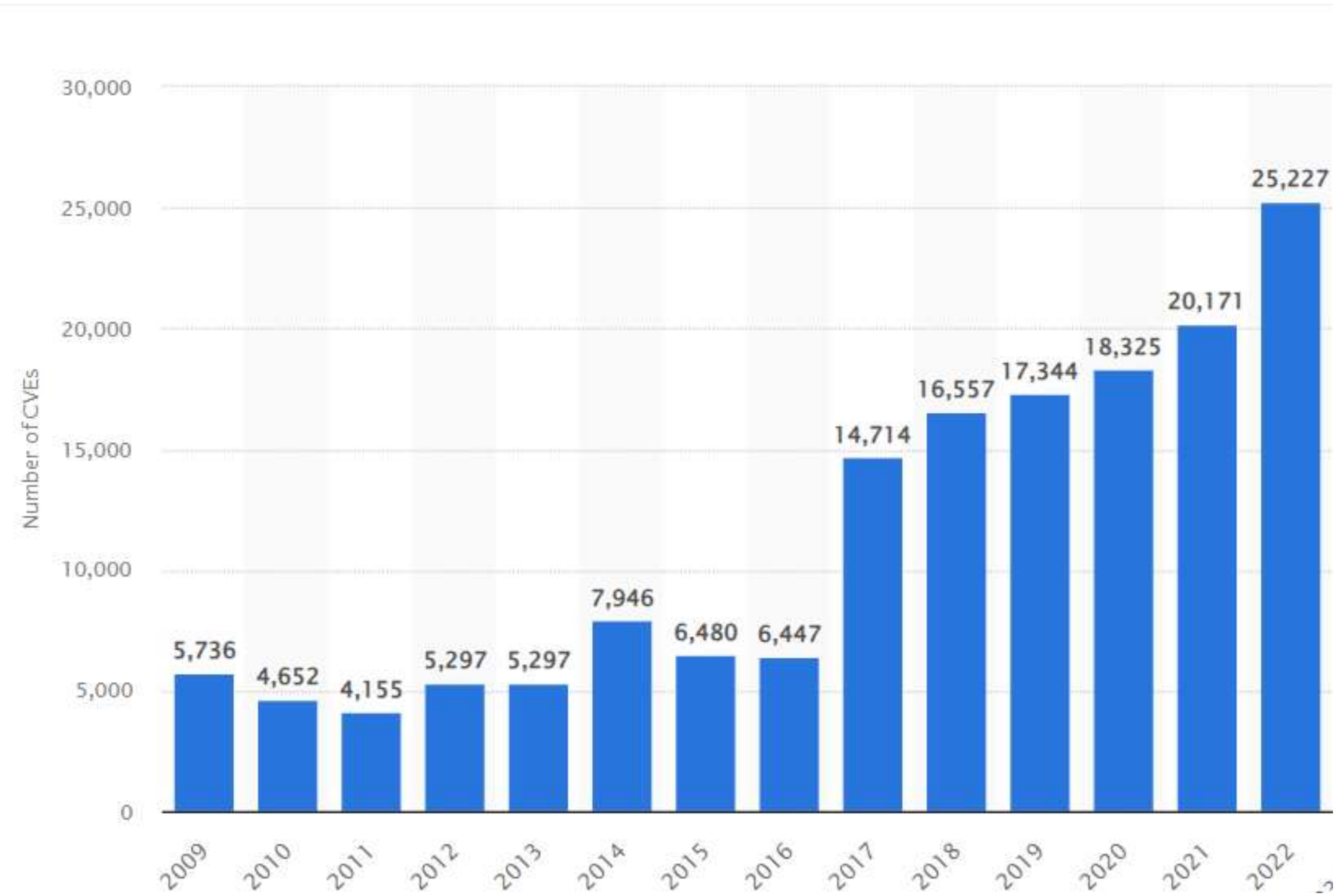
VeriDevOps

Introduction

By Andrey Sadovykh

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 957212

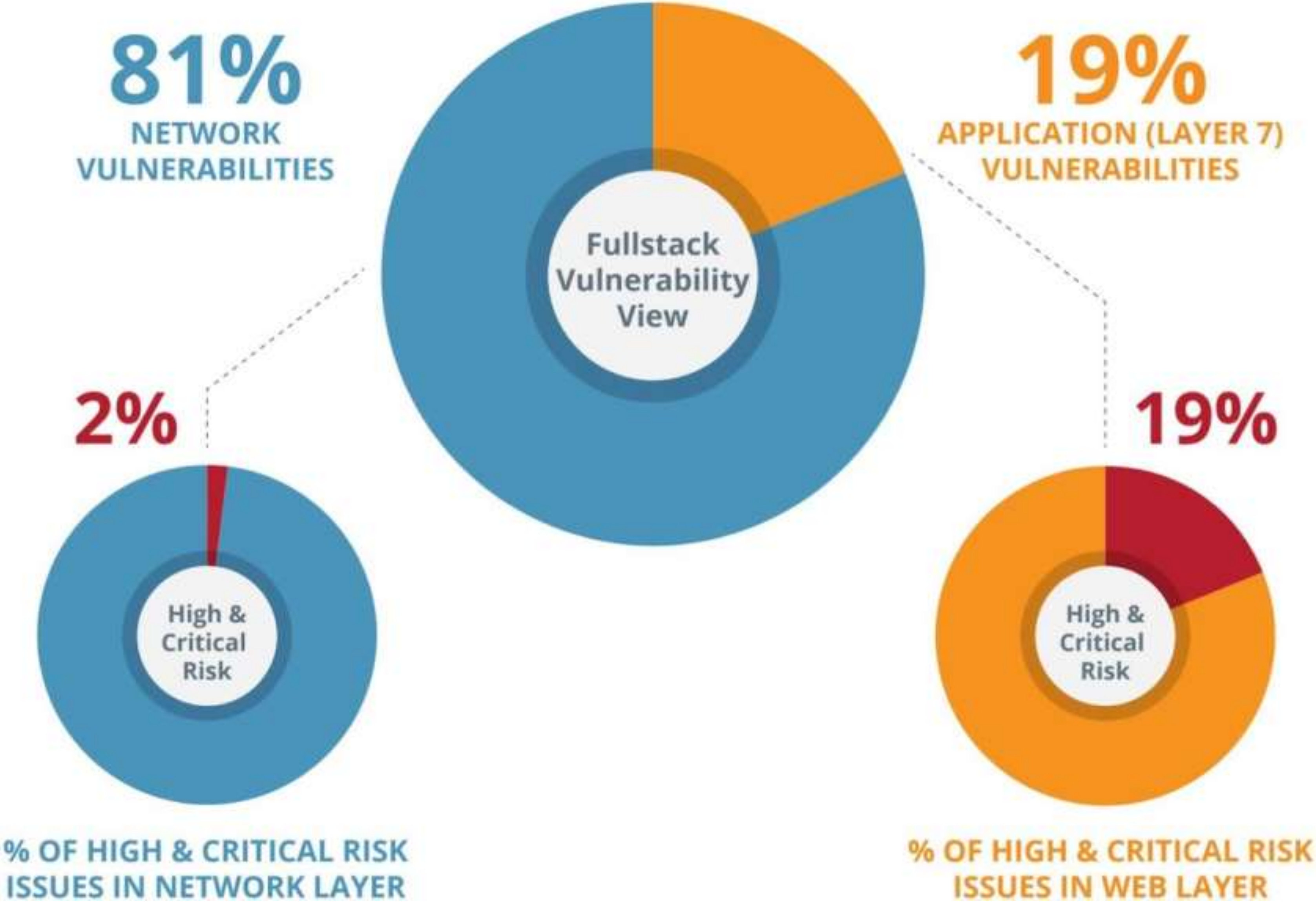
State of industry



Number of common IT security vulnerabilities and exposures (CVEs) worldwide from 2009

Statista

State of industry



81% NETWORK VULNERABILITIES

19% APPLICATION (LAYER 7) VULNERABILITIES

2%

% OF HIGH & CRITICAL RISK ISSUES IN NETWORK LAYER

19%

% OF HIGH & CRITICAL RISK ISSUES IN WEB LAYER

Systems with Multiple Vulnerabilities

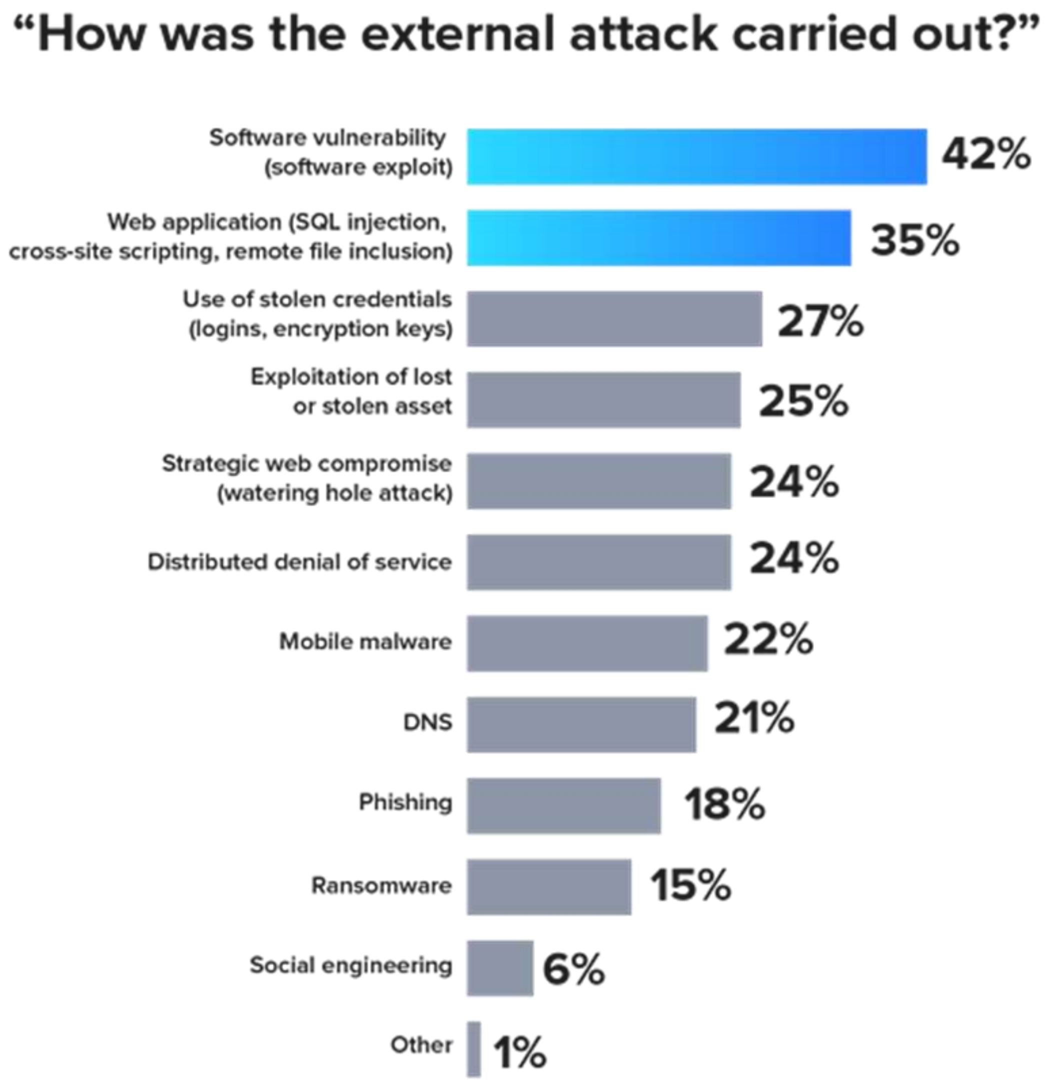
81.58% of systems had at least one CVE

72.11% of systems had more than one CVE

Interestingly, 20.57% of systems had more than 10 CVEs

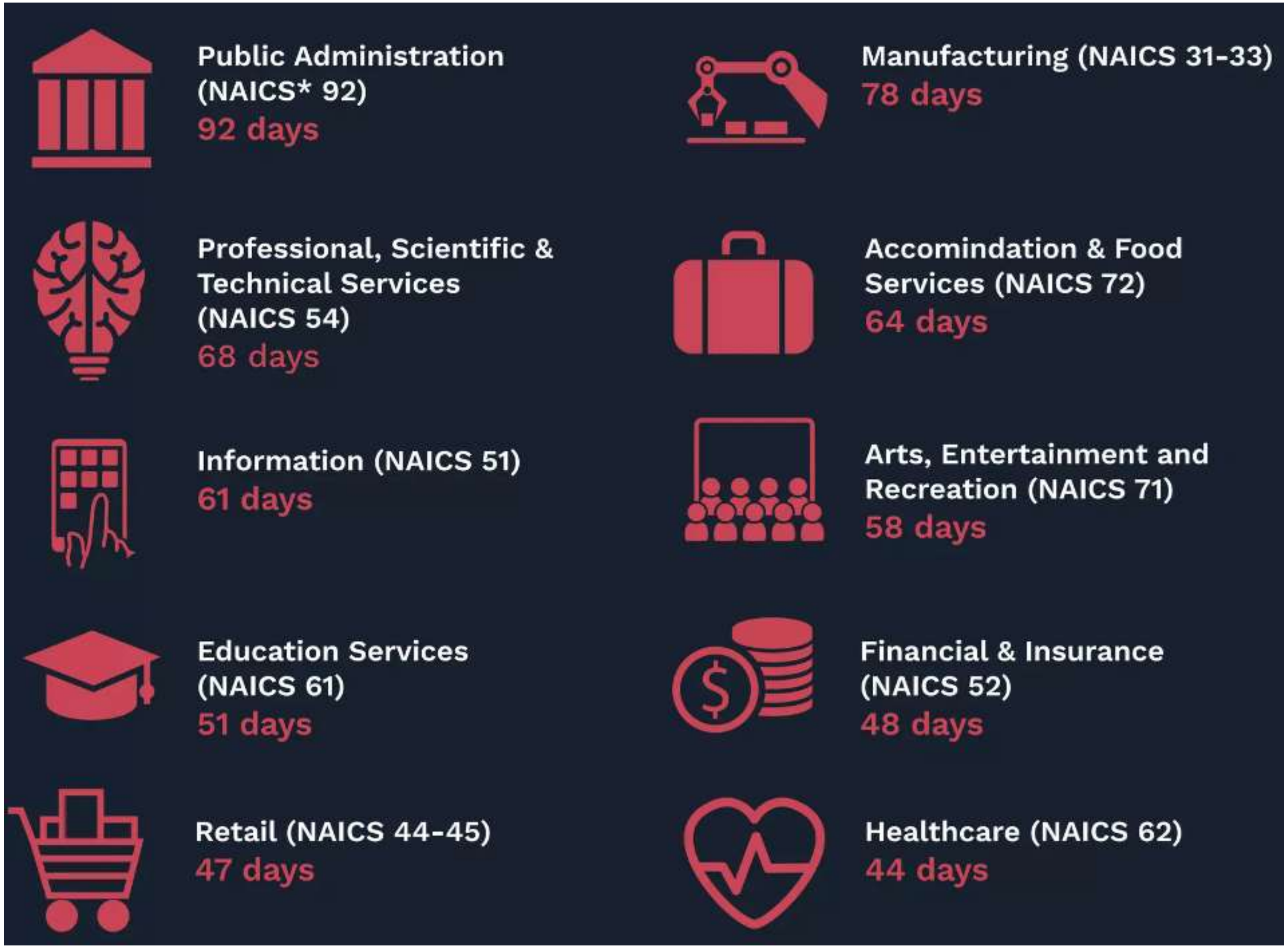
Applications remain the most common attack vector

80% of public exploits are published before CVEs are released



Source: Forrester-2021-app-security-report

Meant Time to Remediate - 57 days



Public Administration (NAICS* 92)
92 days

Professional, Scientific & Technical Services (NAICS 54)
68 days

Information (NAICS 51)
61 days

Education Services (NAICS 61)
51 days

Retail (NAICS 44-45)
47 days

Manufacturing (NAICS 31-33)
78 days

Accomindation & Food Services (NAICS 72)
64 days

Arts, Entertainment and Recreation (NAICS 71)
58 days

Financial & Insurance (NAICS 52)
48 days

Healthcare (NAICS 62)
44 days

EdgeScan 2022

Timeline (OWASP guide v4.)

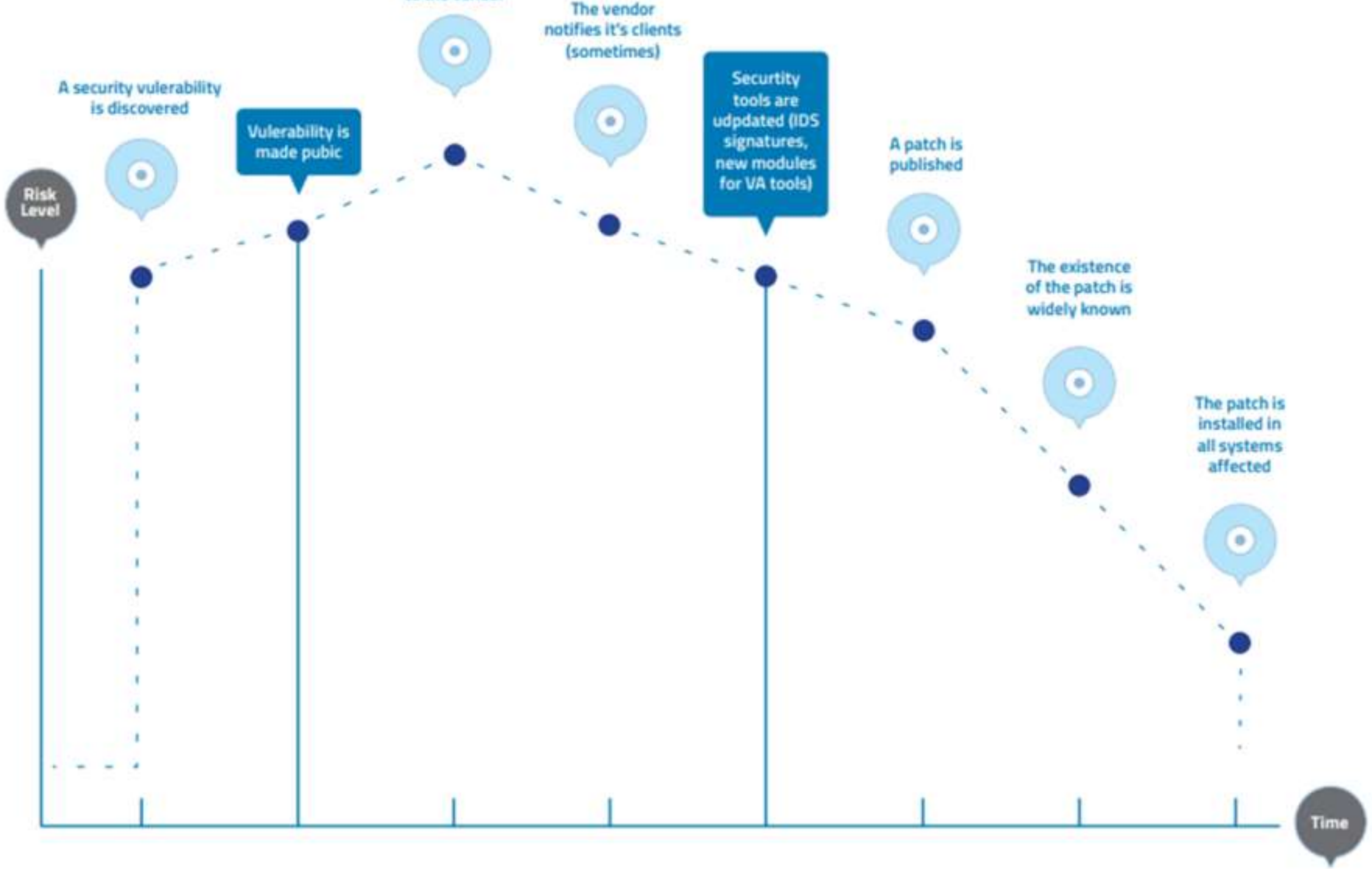

Risk Level
A security vulerability is discovered
Vulerability is made pubic
Vulerability is know to the vendor
The vendor notifies it's clients (sometimes)
Securtity tools are udpdated (IDS signatures, new modules for VA tools)
A patch is published
The existence of the patch is widely known
The patch is installed in all systems affected
Time

Figure 2: Window of Vulnerability

Voice of industry

- Since 2002, the total number of software vulnerabilities has grown year by year **by the thousands**. The peak year seems to have been 2018 for now, but the figures keep rising – **ENISA report for 2018**.
- Upon a breach or failed audit, nearly half of companies (**46%**) took longer than **10 days** to remedy the situation and apply patches, because deploying updates in the entire organization can be difficult – **Voke Media survey**, 2016.
- The average time for organizations to close a discovered vulnerability (caused by unpatched software and apps) is **57 days** – **Edgescan Stats Report**, 2022.
- **37%** of organizations admitted that they don't even scan for vulnerabilities – **Ponemon Report**, 2018.

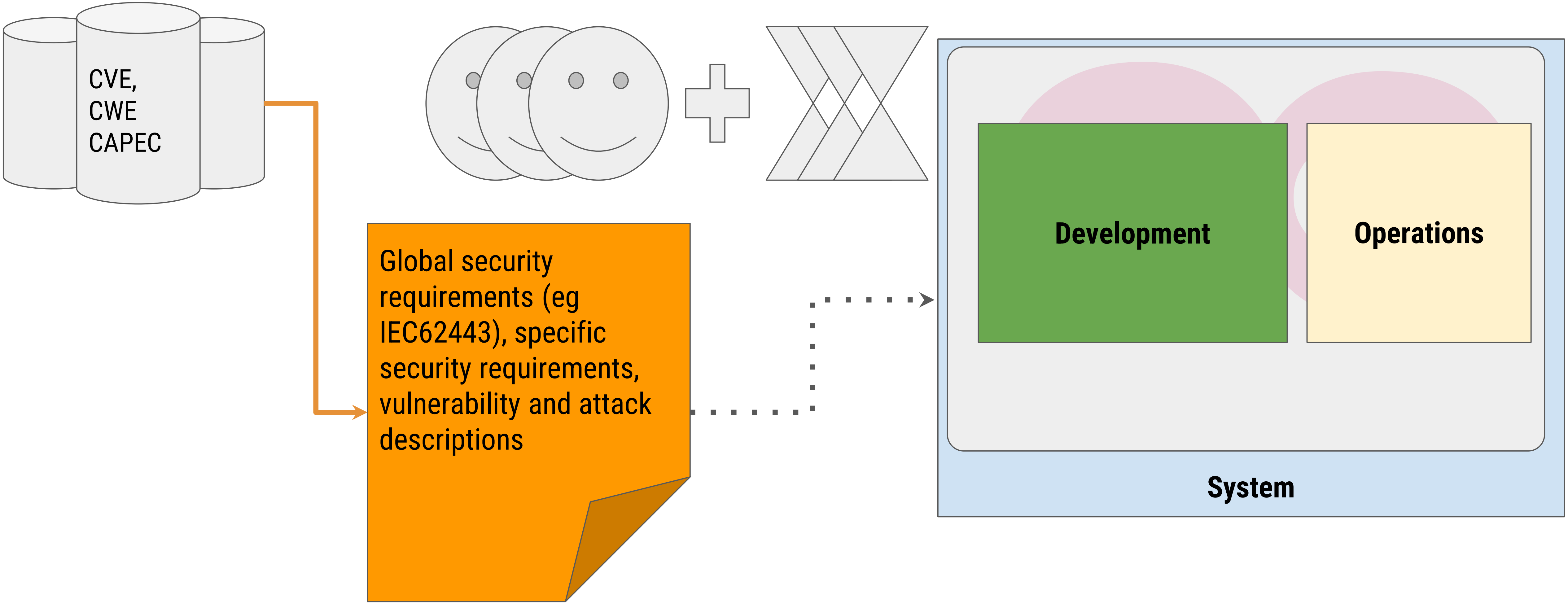
- **58%** of organizations run on '**legacy systems**' – platforms which are no longer supported with patches but which would still be too expensive to replace in the near future – **0patch Survey Report**, 2017.
- More than half of all companies **(55%)** say that when it comes to spending more **time manually navigating** the various processes involved than actually patching vulnerabilities;
- On average it takes **12 days** for teams to coordinate for applying a patch across all devices;
- Most companies **(61%)** feel that they are disadvantages for relying on manual processes for applying software patches;
- Nearly two-thirds of all companies **(65%)** say that it is currently too difficult for them to decide correctly on the priority level of each software patch (aka which update is of critical importance and should be applied first).

Challenges

- Security vulnerability are omnipresent
  - Internet, Cars, Railway, Industry 4.0
- Number of security scenarios explodes
- Vulnerabilities cause losses for end-users
  - increase in the production and maintenance costs

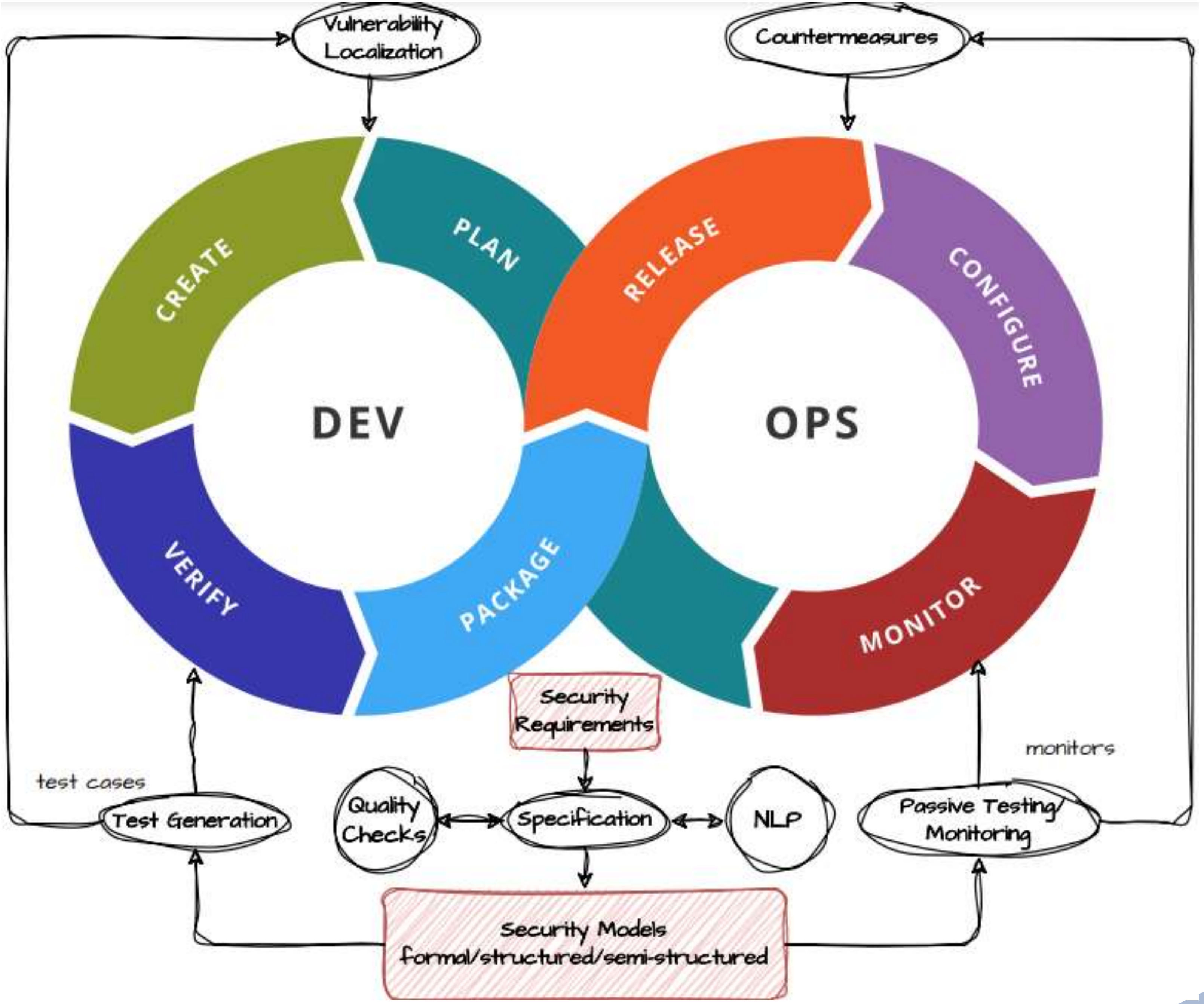
- Security mechanisms have to be built in and reinforced
- Security is difficult to retrofit in design
- Security has to support CI/CD

Typical vulnerability management scenario

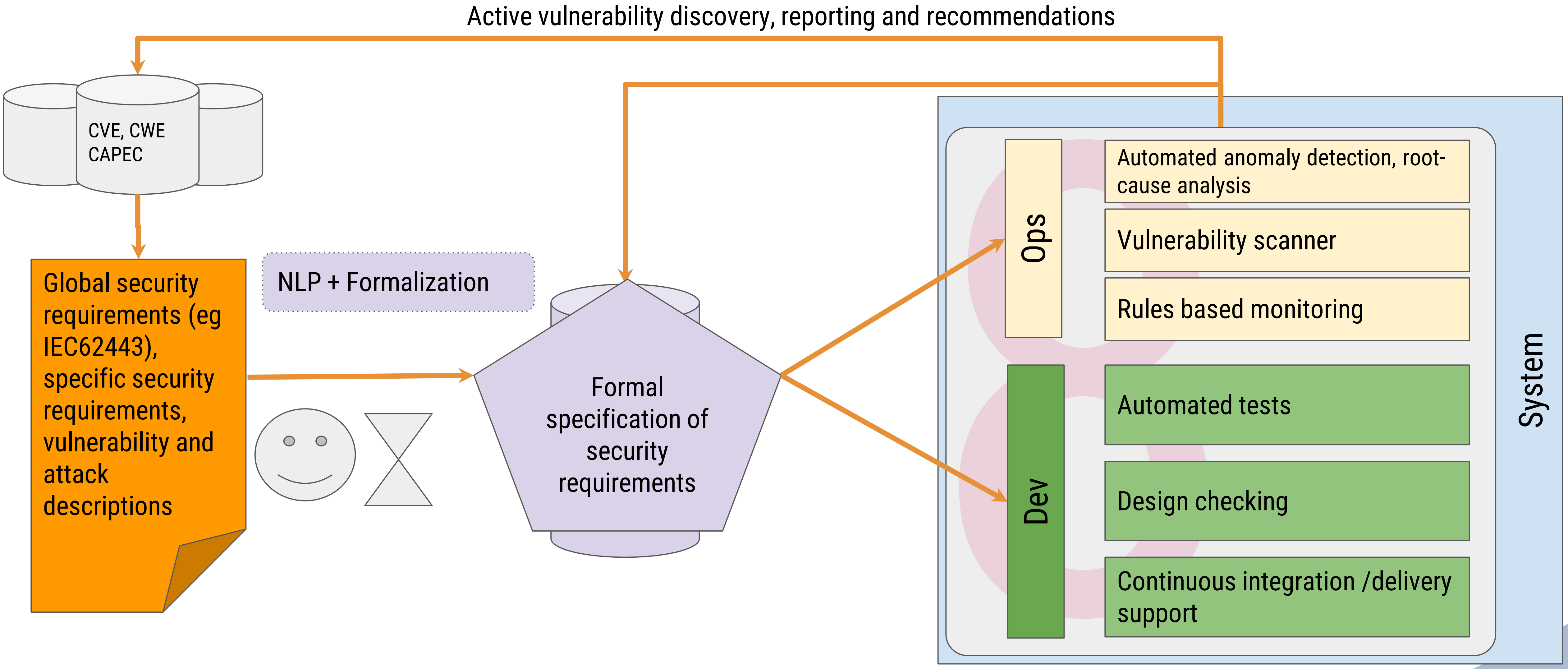


Global security requirements (eg IEC62443), specific security requirements, vulnerability and attack descriptions

Overview


Vulnerability Localization
Countermeasures
CREATE
PLAN
RELEASE
CONFIGURE
DEV
OPS
VERIFY
PACKAGE
MONITOR
Security Requirements
test cases
monitors
Test Generation
Quality Checks
Specification
NLP
Passive Testing/ Monitoring
Security Models formal/structured/semi-structured

Concept



Active vulnerability discovery, reporting and recommendations

CVE, CWE
CAPEC

Global security requirements (eg IEC62443), specific security requirements, vulnerability and attack descriptions

NLP + Formalization

Formal specification of security requirements

Automated anomaly detection, root-cause analysis

Vulnerability scanner

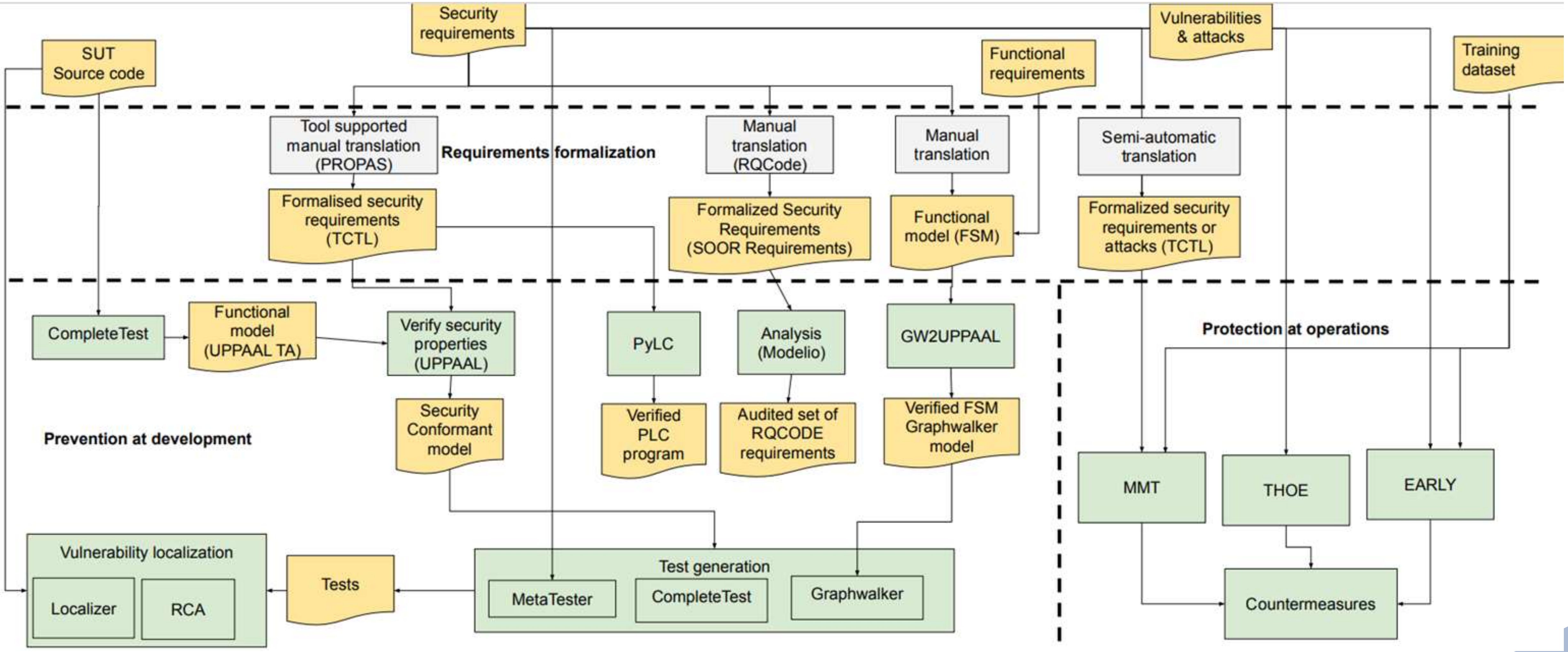
Rules based monitoring

Automated tests

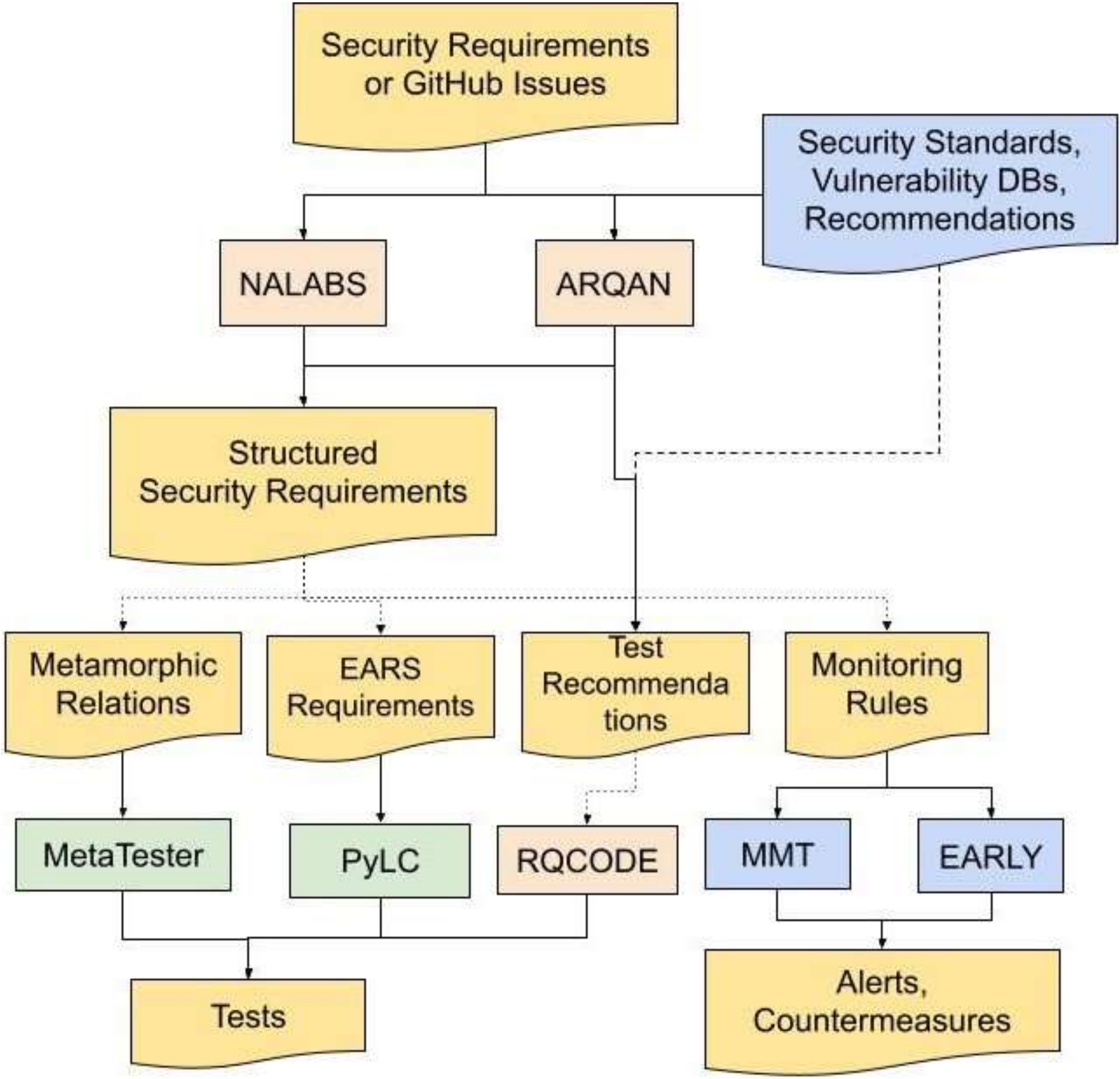
Design checking

Continuous integration /delivery support

Methodology


Security requirements
Vulnerabilities & attacks
SUT Source code
Functional requirements
Training dataset
Tool supported manual translation (PROPAS)
Requirements formalization
Manual translation (RQCode)
Manual translation
Semi-automatic translation
Formalised security requirements (TCTL)
Formalized Security Requirements (SOOR Requirements)
Functional model (FSM)
Formalized security requirements or attacks (TCTL)
CompleteTest
Functional model (UPPAAL TA)
Verify security properties (UPPAAL)
PyLC
Analysis (Modelio)
GW2UPPAAL
Protection at operations
Security Conformant model
Verified PLC program
Audited set of RQCODE requirements
Verified FSM Graphwalker model
Prevention at development
MMT
THOE
EARLY
Vulnerability localization
Localizer
RCA
Tests
Test generation
MetaTester
CompleteTest
Graphwalker
Countermeasures

Tool chain examples



Security Requirements or GitHub Issues

Security Standards, Vulnerability DBs, Recommendations

NALABS

ARQAN

Structured Security Requirements

Metamorphic Relations

EARS Requirements

Test Recommendations

Monitoring Rules

MetaTester

PyLC

RQCODE

MMT

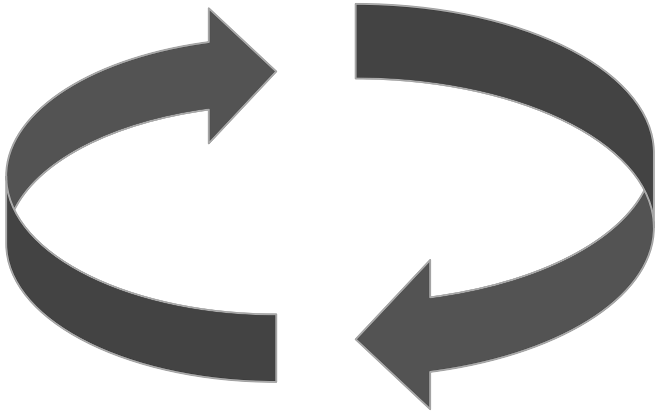
EARLY

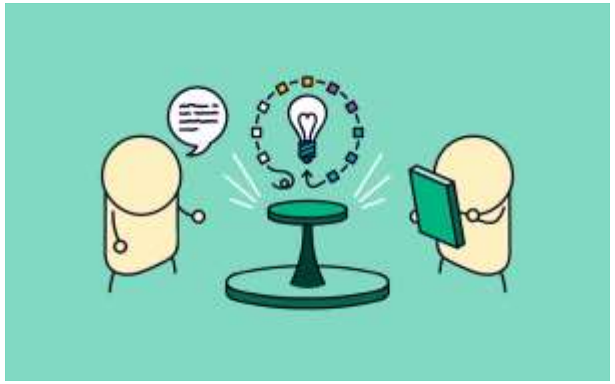
Tests

Alerts, Countermeasures

Key results

10 use cases





20+ papers

12+ tools

Key innovations (RIA) and more

1. NLP datasets and models for Requirements classification and security guidelines mapping.
2. ML-based anomaly detection and root cause analysis.
3. Metamorphic testing generation as intelligent test generation with automated feedback.
4. Vulnerability detection at early stages with scanners.

Next

Time	Duration	Topic	Presenter	Organization
9:30	20 mins	VeriDevOps Technical Introduction	Andrey Sadovykh	SOFTEAM
Part I: Security Requirements Engineering				
9:50	20 mins	A Taxonomy of Vulnerabilities, Attacks, and Security Solutions in Industrial PLCs.	Eduard Paul Enoiu	Mälardalen University
10:10	20 mins	Natural Language Processing with Machine Learning for Security Requirements Analysis - Practical Approaches.	Andrey Sadovykh	SOFTEAM
10:30	20 mins	Security Requirements Formalization with RQCODE.	Andrey Sadovykh	SOFTEAM
10:50	10 mins	break	/	/

Thank You

Contact: Andrey Sadovykh, SOFTEAM

This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No. 957212