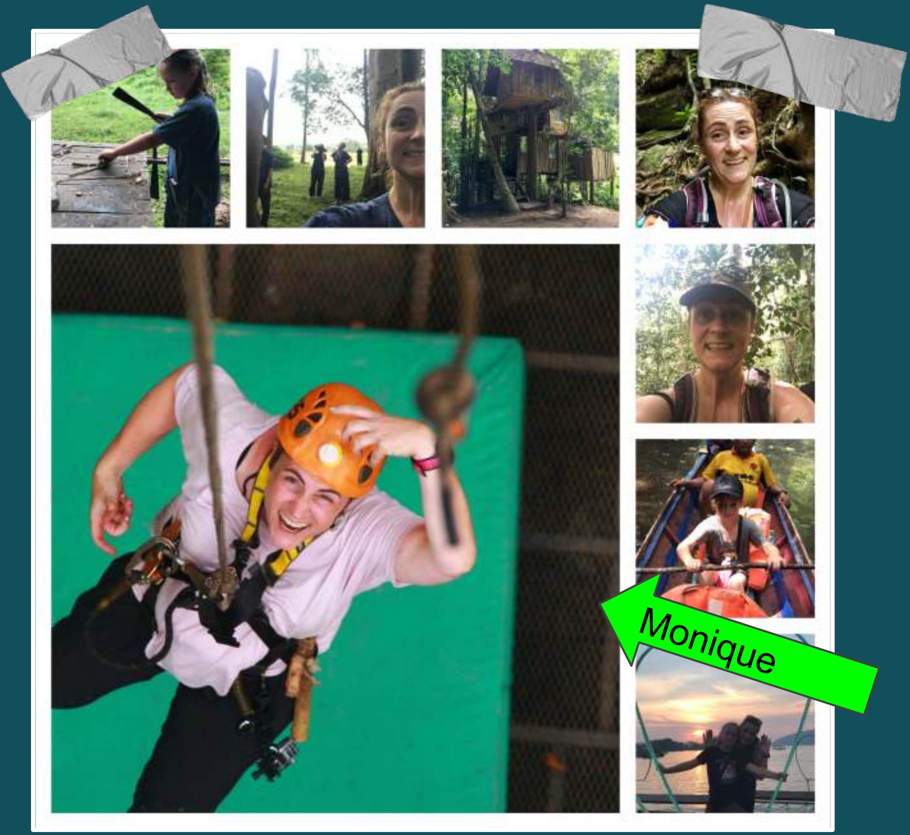
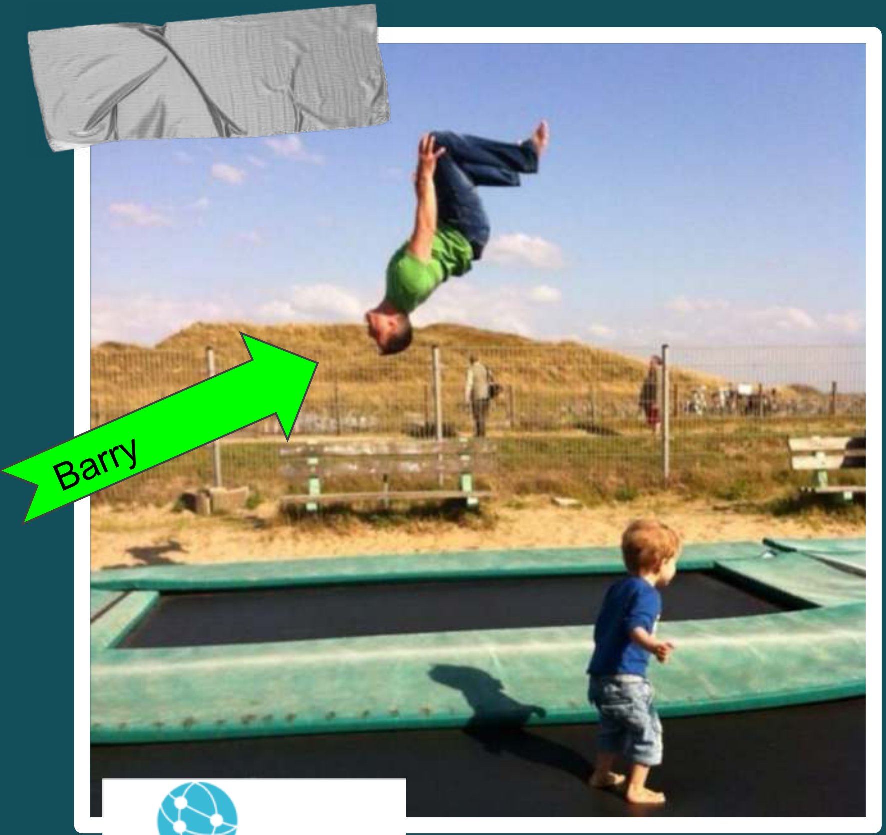
Agile Secure Software

Secure by Agile Design

SPREKERS:
dr. lec. Barry Derksen MSc. MMC CISA CGEIT
drs. Monique Neggers CISA CGEIT CISM CRISC

Secure Software Alliance





STUUR Coaching en Consultancy

inc
I'm in control

Secure
Software
Alliance

Zoom in at Cyber Security controls: What is your status?

1. Do we know all URLs & are those OWASP proof?

2 Abuse cases created?

3 Are our big data analysts screened?

4 Is your testdata scrambled?

5 How many systems are pentested?

6 Prioritized Risk Backlog? For software?

7 ISO27001? By obligation or by heart?

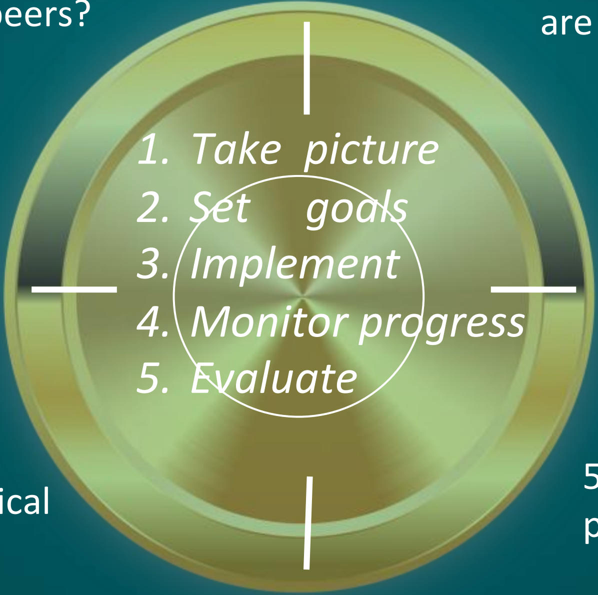
8 When was your last physical security testing?

9 Analysis?GDPR compliant? Security laws?

10 How many data leakages reported?

11 Developer staff trained in security practices?

12 How good is our SOC compared to peers?



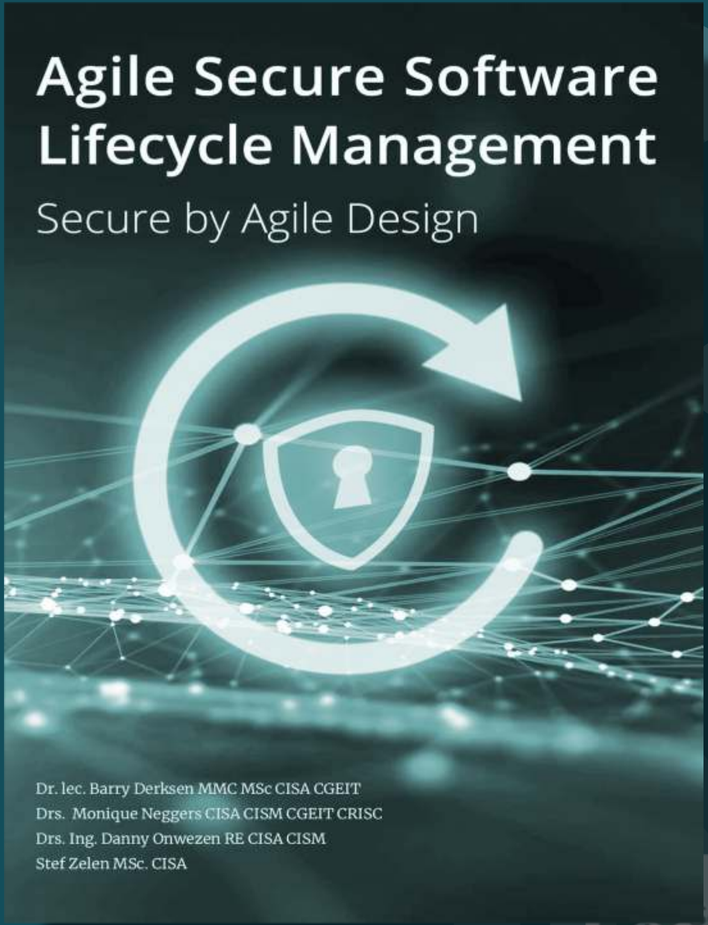
1. Take picture
2. Set goals
3. Implement
4. Monitor progress
5. Evaluate

Secure Software Alliance

Agile Secure lifecycle management

8 Sprints

1. Because we have to!
2. Developers meets hacker
3. Agile beats structure
4. Software Security Fundamentals
5. Introducing Agile secure Software Development
6. Agile Secure Software Framework
7. Maturing Agile Secure Software Life Cycle
8. Ren je Rot



Agile Secure Software Lifecycle Management

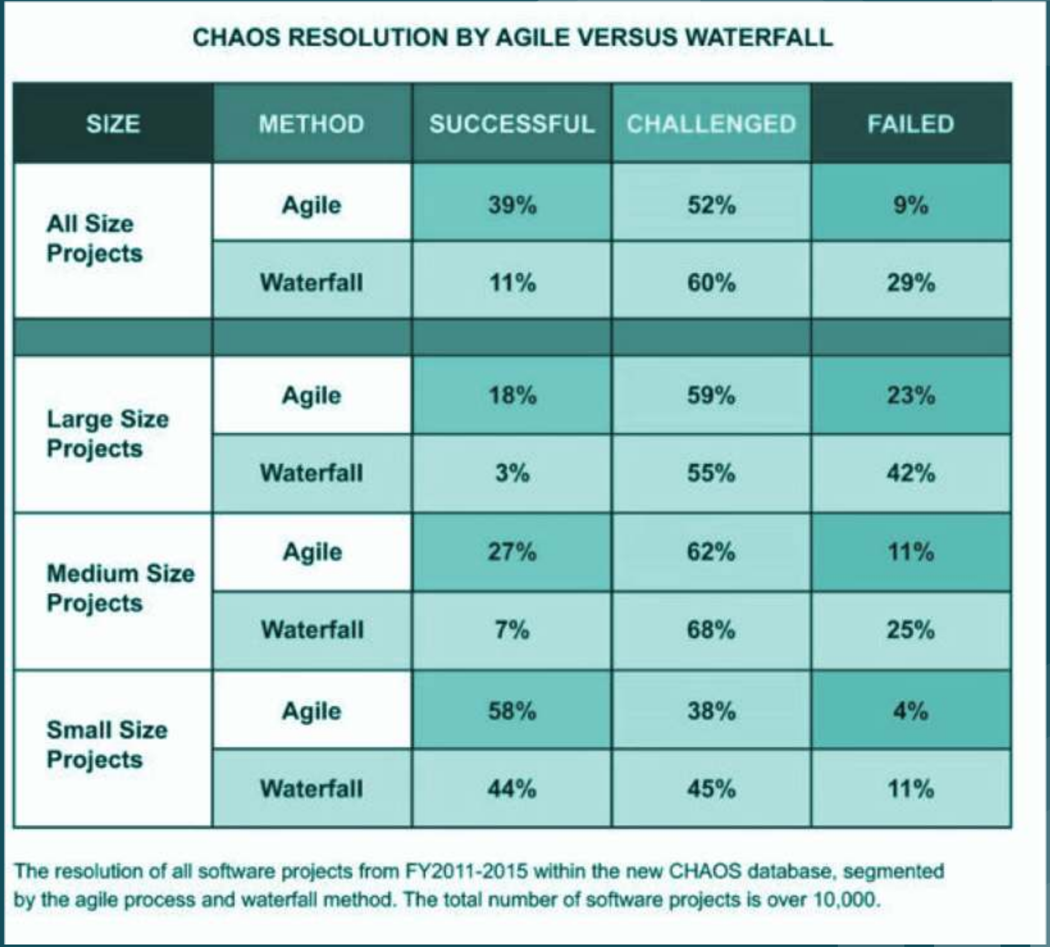
Secure by Agile Design

Dr. lec. Barry Derksen MMC MSc CISA CGEIT
Drs. Monique Neggers CISA CISM CGEIT CRISC
Drs. Ing. Danny Onwezen RE CISA CISM
Stef Zelen MSc. CISA

Sprint 1. Because we have to!

B

- ❑ Agile **Secure** Software Development is a contradiction in terms
- ❑ Security is a challenge
- ❑ Agile Secure Development needs professionals
- ❑ Software is everywhere



CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

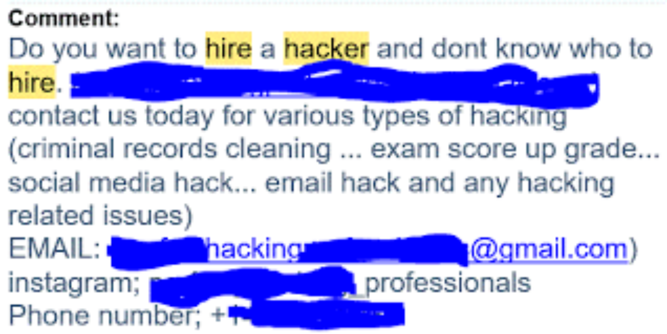
SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011-2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

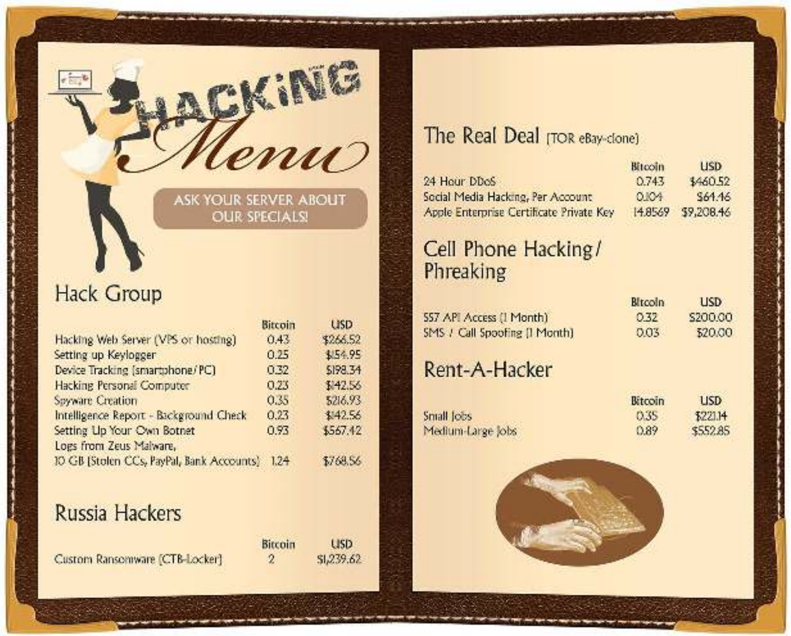
Secure Software Alliance

Sprint 2: Developer meets Hacker

- ❑ Just one flaw is enough
- ❑ Every step needs to be checked on abuse cases
- ❑ Risk based, using CIA



Comment:
Do you want to hire a hacker and dont know who to hire. [redacted]
contact us today for various types of hacking
(criminal records cleaning ... exam score up grade...
social media hack... email hack and any hacking
related issues)
EMAIL: [redacted]hacking[redacted]@gmail.com)
instagram; [redacted] professionals
Phone number; +[redacted]



Hack Group

	Bitcoin	USD
Hacking Web Server (VPS or hosting)	0.43	$266.52
Setting up Keylogger	0.25	$154.95
Device Tracking (smartphone/PC)	0.32	$198.34
Hacking Personal Computer	0.23	$142.56
Spyware Creation	0.35	$216.93
Intelligence Report - Background Check	0.23	$142.56
Setting Up Your Own Botnet	0.93	$567.42
Logs from Zeus Malware, 10 GB [Stolen CCs, PayPal, Bank Accounts]	1.24	$768.56

Russia Hackers

	Bitcoin	USD
Custom Ransomware [CTB-Locker]	2	$1,239.62

The Real Deal [TOR eBay-clone]

	Bitcoin	USD
24 Hour DDoS	0.743	$460.52
Social Media Hacking, Per Account	0.104	$64.46
Apple Enterprise Certificate Private Key	14.8569	$9,208.46

Cell Phone Hacking/ Phreaking

	Bitcoin	USD
SS7 API Access (1 Month)	0.32	$200.00
SMS / Call Spoofing [1 Month]	0.03	$20.00

Rent-A-Hacker

	Bitcoin	USD
Small Jobs	0.35	$221.14
Medium-Large Jobs	0.89	$552.85

EVIL User Stories

❑ **Example #1** "As a hacker, I can send bad data in URLs, so I can access data and functions for which I'm not authorized"

❑ **Example #2** "As a hacker, I can send bad data in the content of requests, so I can access data and functions for which I'm not authorized"

❑ **Example #3** "As a hacker, I can read and even modify all data that is input and output by your application"

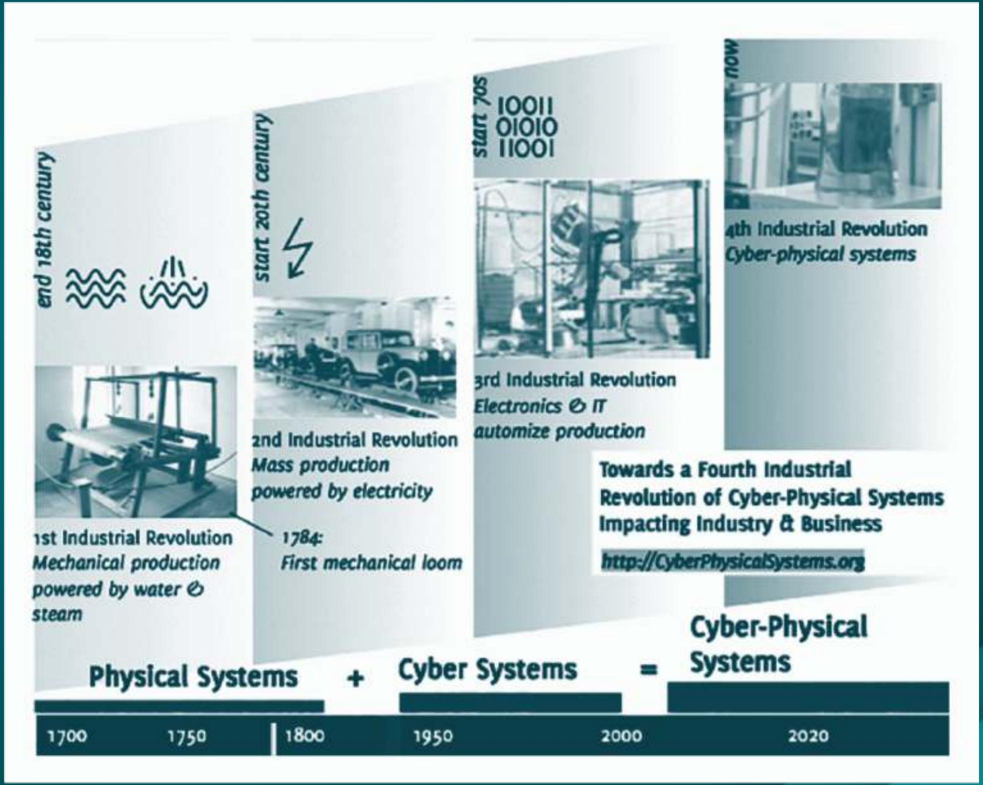
Secure Software Alliance

B

Sprint 3: Agile beats structure

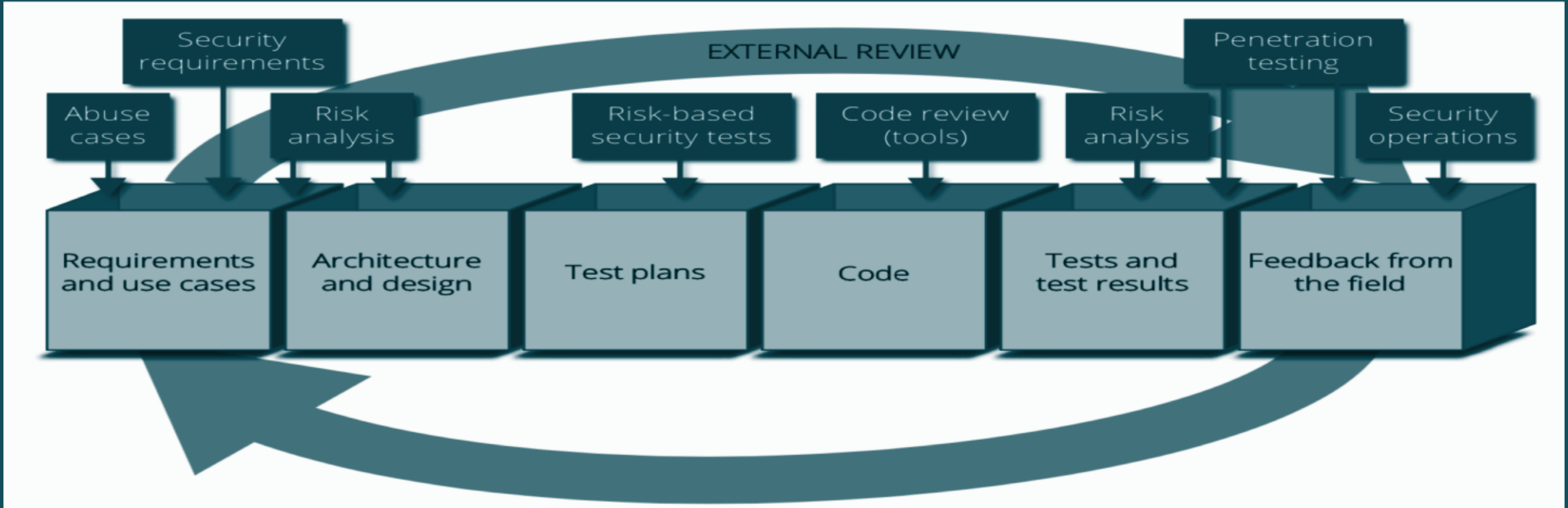
Call for Agile:

- □ Social media
- □ Mobile living
- □ Analytics & Big Data
- □ Cloud
- □ IoT
- □ Chain trends
- □ Risks changes



end 18th century
start 20th century
start 70s
10011
01010
11001
now
1st Industrial Revolution
Mechanical production powered by water & steam
1784:
First mechanical loom
2nd Industrial Revolution
Mass production powered by electricity
3rd Industrial Revolution
Electronics & IT automize production
4th Industrial Revolution
Cyber-physical systems
Towards a Fourth Industrial Revolution of Cyber-Physical Systems Impacting Industry & Business
http://CyberPhysicalSystems.org
Physical Systems
+
Cyber Systems
=
Cyber-Physical Systems
1700
1750
1800
1950
2000
2020

Sprint 4: Software Security Fundamentals

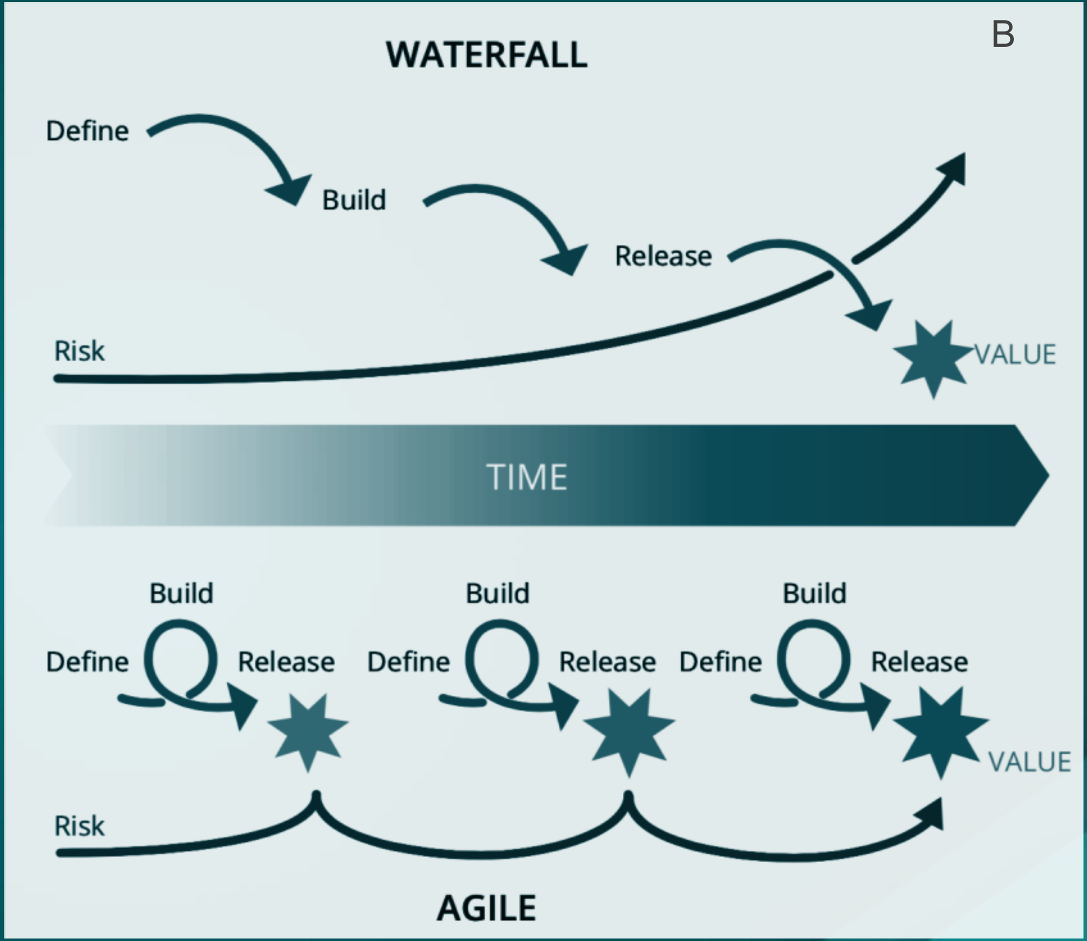


Security measures in SDLC,
Source: Gary McGraw, Software Security in ,2006

Secure Software Alliance

Sprint 5: Agile Secure Software Development

- ❑ Stakeholders part of risk assessment
- ❑ Stakeholders security tests during product review
- ❑ Acceptance criteria security user stories
- ❑ Use Agile retrospectives
- ❑ Group to minimize security damage



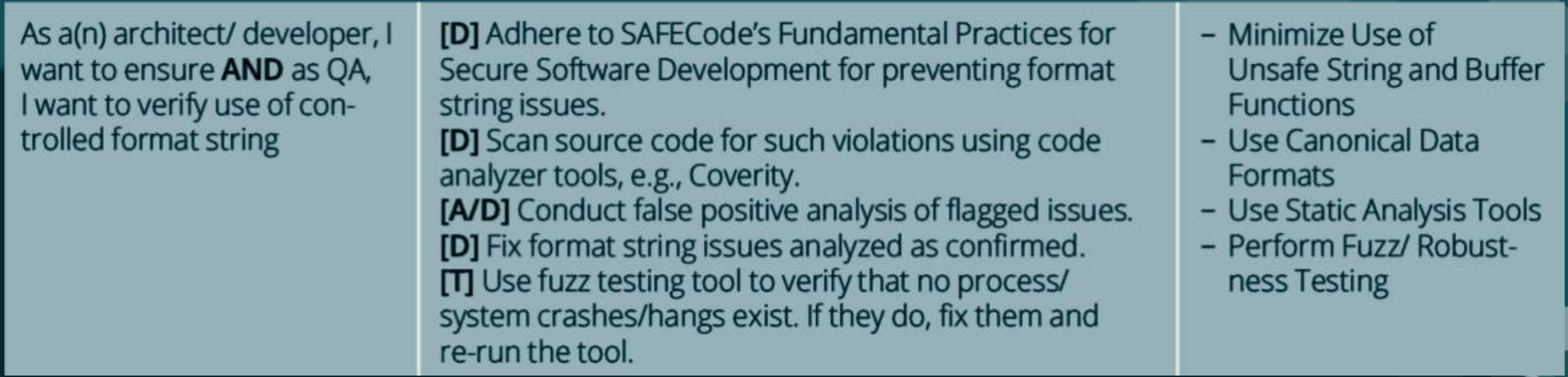
Risk in Waterfall and Agile software development,
Source: Cirdam Group

Secure Software Alliance

B

Sprint 5.1: Adopting security focused stories

- □ Develop Security user stories
- □ Prioritize risk based
- □ decrease risk acceptance level in time
- □ (you’re never finished)



As a(n) architect/ developer, I want to ensure **AND** as QA, I want to verify use of controlled format string	**[D]** Adhere to SAFECode’s Fundamental Practices for Secure Software Development for preventing format string issues. **[D]** Scan source code for such violations using code analyzer tools, e.g., Coverity. **[A/D]** Conduct false positive analysis of flagged issues. **[D]** Fix format string issues analyzed as confirmed. **[T]** Use fuzz testing tool to verify that no process/ system crashes/hangs exist. If they do, fix them and re-run the tool.	– Minimize Use of Unsafe String and Buffer Functions – Use Canonical Data Formats – Use Static Analysis Tools – Perform Fuzz/ Robustness Testing

Figure 15: Security Focused story, Source: Safecode.org

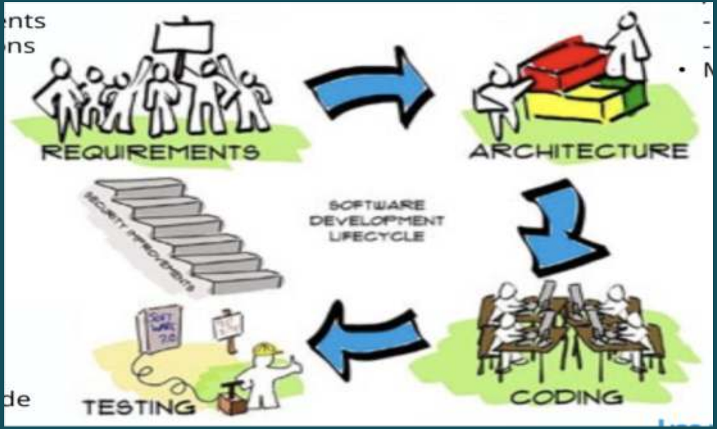
Sprint 6: Agile SECURE Software Development Framework

CONTEXT

- Functions & Environment
- Application assets
- Security requirements
- Security assumptions

VERIFICATION

- Verification method
  - code review
  - penetration test
  - vulnerability scan
  - fuzzing
  - abuse tests
- Verification process



THREATS

- Functional threats
- Architectural threats
  - architecture inventory
  - threat library
- Mitigations

IMPLEMENTATION

- Secure coding principles
- Secure coding standards
- Code Audit

Control Testing example

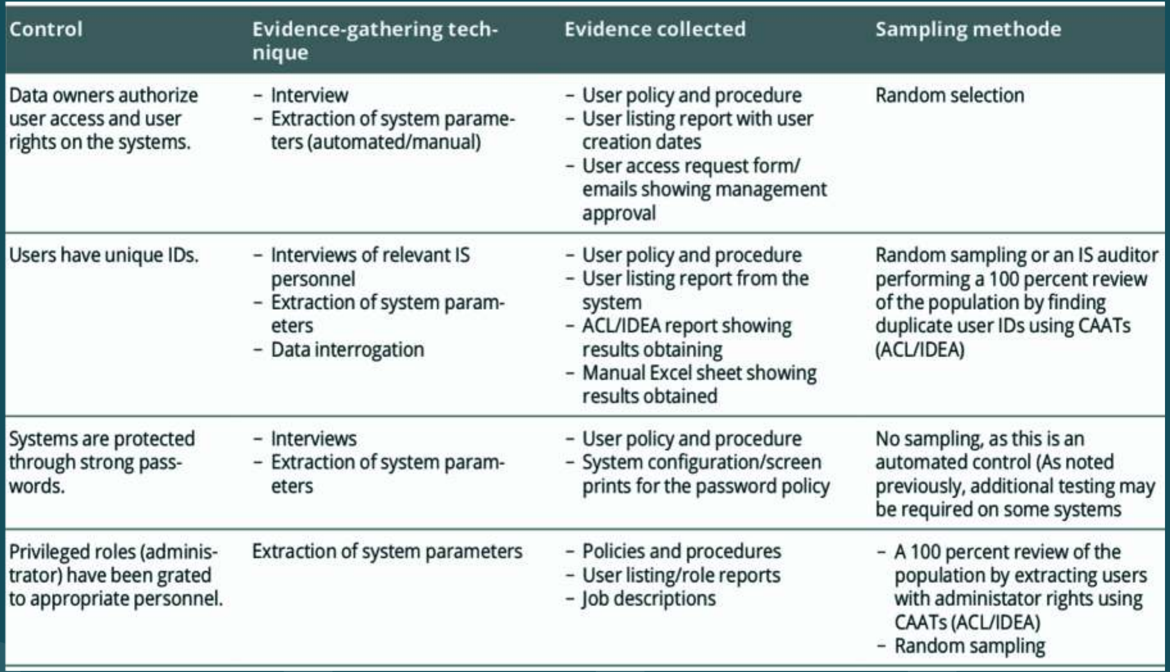
❑Conduct control testing on software releases

❑Derive test cases from known control requirements

❑Employ software specific control testing automation

❑Integrate control testing into development process

❑Utilize automated control testing tools

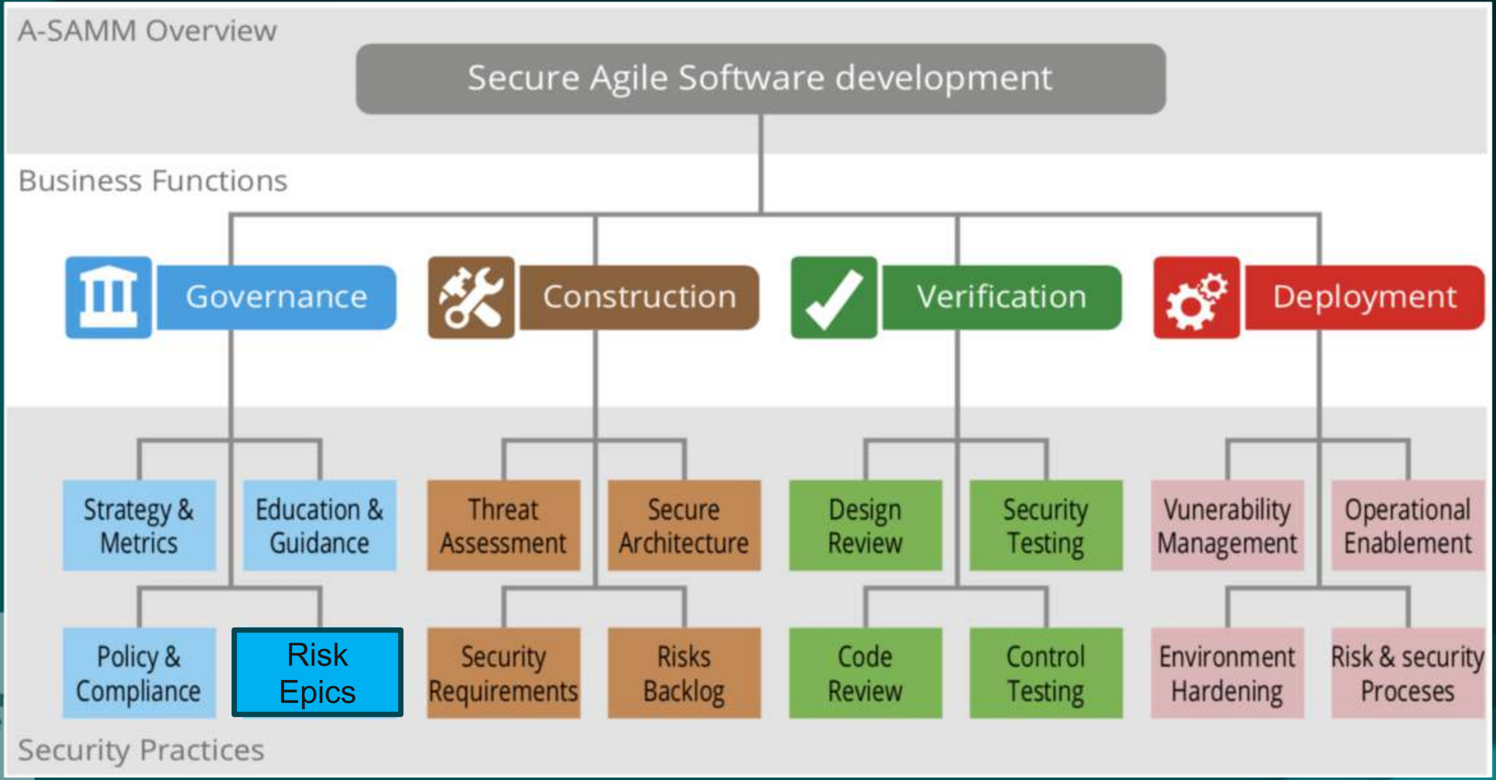


Control	Evidence-gathering technique	Evidence collected	Sampling methode
Data owners authorize user access and user rights on the systems.	– Interview – Extraction of system parameters (automated/manual)	– User policy and procedure – User listing report with user creation dates – User access request form/ emails showing management approval	Random selection
Users have unique IDs.	– Interviews of relevant IS personnel – Extraction of system parameters – Data interrogation	– User policy and procedure – User listing report from the system – ACL/IDEA report showing results obtaining – Manual Excel sheet showing results obtained	Random sampling or an IS auditor performing a 100 percent review of the population by finding duplicate user IDs using CAATs (ACL/IDEA)
Systems are protected through strong passwords.	– Interviews – Extraction of system parameters	– User policy and procedure – System configuration/screen prints for the password policy	No sampling, as this is an automated control (As noted previously, additional testing may be required on some systems
Privileged roles (administrator) have been grated to appropriate personnel.	Extraction of system parameters	– Policies and procedures – User listing/role reports – Job descriptions	– A 100 percent review of the population by extracting users with administator rights using CAATs (ACL/IDEA) – Random sampling

Source: ISACA

B

Sprint 7: Maturing Agile Secure Software Development Life Cycle



A-SAMM Overview

Secure Agile Software development

Business Functions

Governance

Construction

Verification

Deployment

Strategy & Metrics

Education & Guidance

Policy & Compliance

Risk Epics

Threat Assessment

Secure Architecture

Security Requirements

Risks Backlog

Design Review

Security Testing

Code Review

Control Testing

Vunerability Management

Operational Enablement

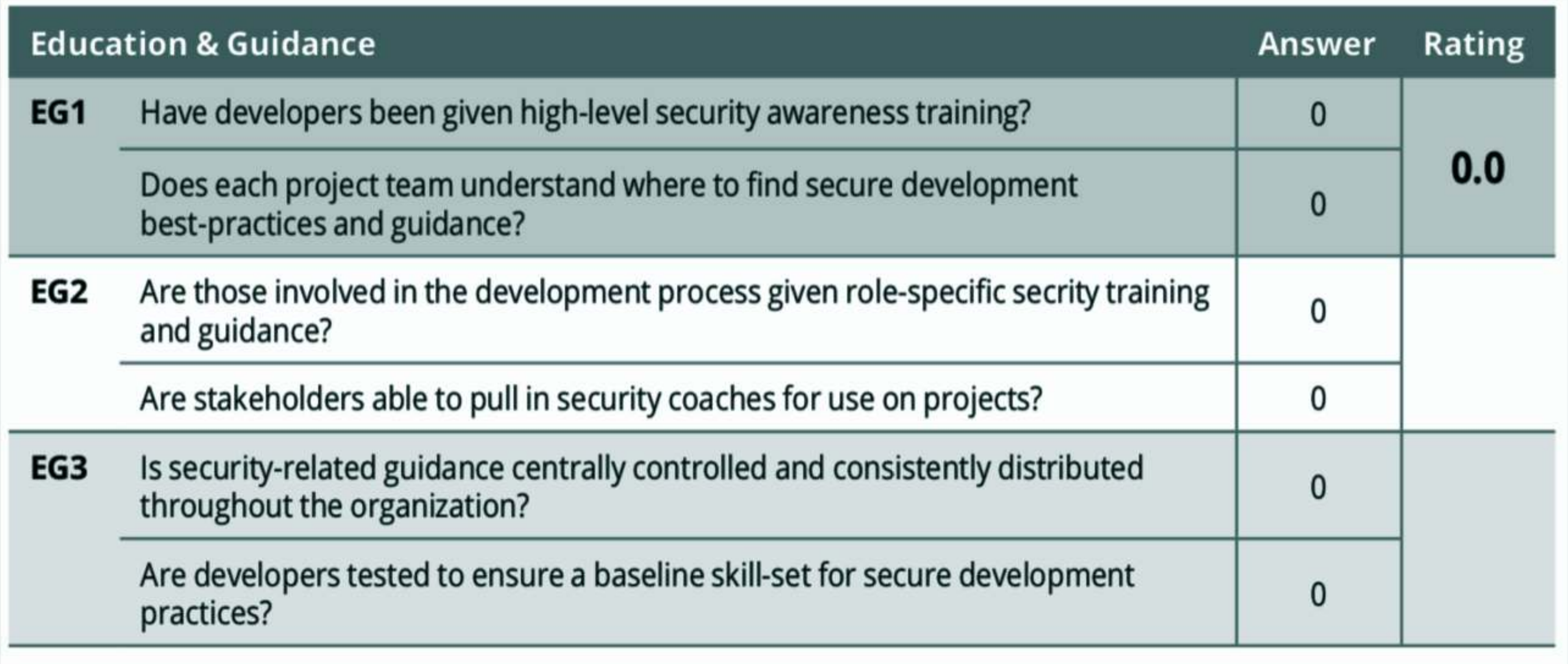
Environment Hardening

Risk & security Proceses

Security Practices

B

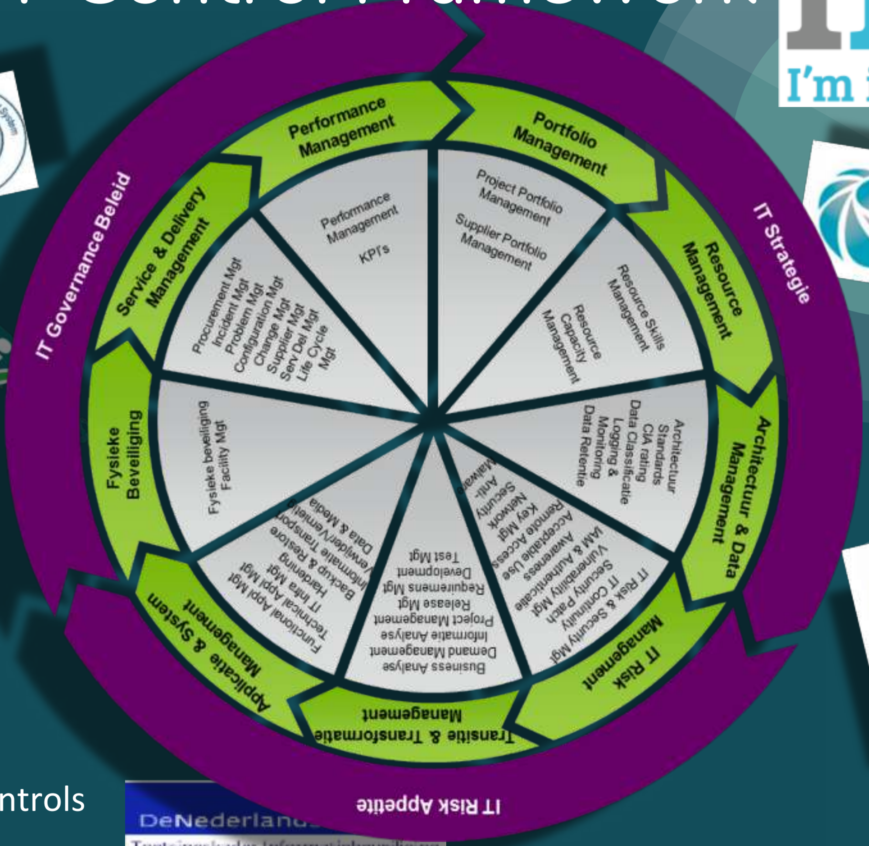
Measure yourself & your supplier



Education & Guidance		Answer	Rating
EG1	Have developers been given high-level security awareness training?	0	0.0
	Does each project team understand where to find secure development best-practices and guidance?	0	
EG2	Are those involved in the development process given role-specific secrity training and guidance?	0	
	Are stakeholders able to pull in security coaches for use on projects?	0	
EG3	Is security-related guidance centrally controlled and consistently distributed throughout the organization?	0	
	Are developers tested to ensure a baseline skill-set for secure development practices?	0	

Source: SAMM v1.5

Make it part of Integrated© IT Control Framework



COBIT® 5 AN ISACA® FRAMEWORK

Information Security Management System ISO 27001 Certified

PCI

bisl® business information services library

IIIC I'm in control

Secure Software Alliance

MSP®

ITIL®

IT4IT

APPLICATION SERVICES LIBRARY asl

PRINCE2

COBIT® 2019

DeNederlan

Toetsingskader Informatiebeveiliging voor DNB thema-onderzoek 2014

≈17 beleidsstukken

≈ 13 processes

≈ 20 procedures

≈ 45 Standaards with > 250 controls

Summary

- ❑ Software is everywhere!
- ❑ Agile Secure Software Development is a contradiction in terms
- ❑ Security (by design) is a challenge!
- ❑ Just one flaw is enough
- ❑ Evil user stories are a must have
- ❑ Integrating Agile Risk Management is essential
- ❑ Agile Secure Software Development needs professionals

POLDERS
GEMALEN
1
3

Ren je Rot
Ren je Rot

1
2
3

STELLING 1

Agile = AIRgile

STELLING 2

De huidige IT controls bij Agile werken **niet**

Secure
Software
Alliance

STELLING 3

Agile is de zilveren kogel die (eindelijk) IT projecten succesvol maakt.

STELLING 4

De product owner moet van IT komen

Secure Software Alliance

STELLING 5

ERP kan **niet** agile worden aangepakt

STELLING 6

Internal audit werk kan agile worden ingericht

STELLING 7

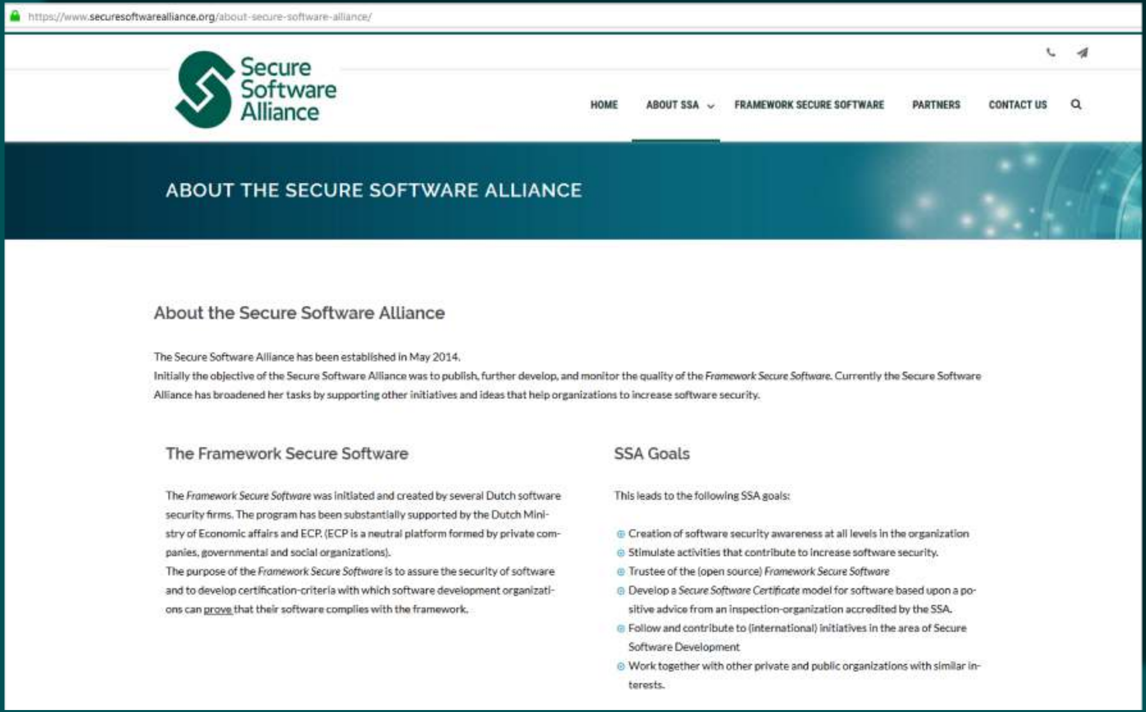
Risk backlog?
Dat **werkt niet**

Secure Software Alliance

STELLING 8

Security in retrospective, dat is een goed start

MENTI: 47 99 45



https://www.securesoftwarealliance.org/about-secure-software-alliance/

ABOUT THE SECURE SOFTWARE ALLIANCE

About the Secure Software Alliance

The Secure Software Alliance has been established in May 2014.
Initially the objective of the Secure Software Alliance was to publish, further develop, and monitor the quality of the *Framework Secure Software*. Currently the Secure Software Alliance has broadened her tasks by supporting other initiatives and ideas that help organizations to increase software security.

The Framework Secure Software

The *Framework Secure Software* was initiated and created by several Dutch software security firms. The program has been substantially supported by the Dutch Ministry of Economic affairs and ECP. (ECP is a neutral platform formed by private companies, governmental and social organizations).
The purpose of the *Framework Secure Software* is to assure the security of software and to develop certification-criteria with which software development organizations can prove that their software complies with the framework.

SSA Goals

This leads to the following SSA goals:

- Creation of software security awareness at all levels in the organization
- Stimulate activities that contribute to increase software security.
- Trustee of the (open source) *Framework Secure Software*
- Develop a *Secure Software Certificate* model for software based upon a positive advice from an inspection-organization accredited by the SSA.
- Follow and contribute to (international) initiatives in the area of Secure Software Development
- Work together with other private and public organizations with similar interests.

Agile Secure Software Lifecycle Management

Secure by Agile Design

Dr. lec. Barry Derksen MMC MSc CISA CGEIT
Drs. Monique Neggers CISA CISM CGEIT CRISC
Drs. Ing. Danny Onwezen RE CISA CISM
Stef Zelen MSc. CISA