Agile Secure Software Lifecycle Management

Danny Onwezen, voorzitter SSA

ISACA Round Table, 4 februari 2019

Secure Software Alliance

About the Secure Software Alliance (SSA)

Goals

- Creation of software security awareness at all levels in the organization
- Stimulate activities that contribute to increase secure software
- Trustee of the (open source) Agile Framework Secure Software
- Follow and contribute to (international) initiatives in the area of secure software development
- Work together with other private and public organizations with similar interests

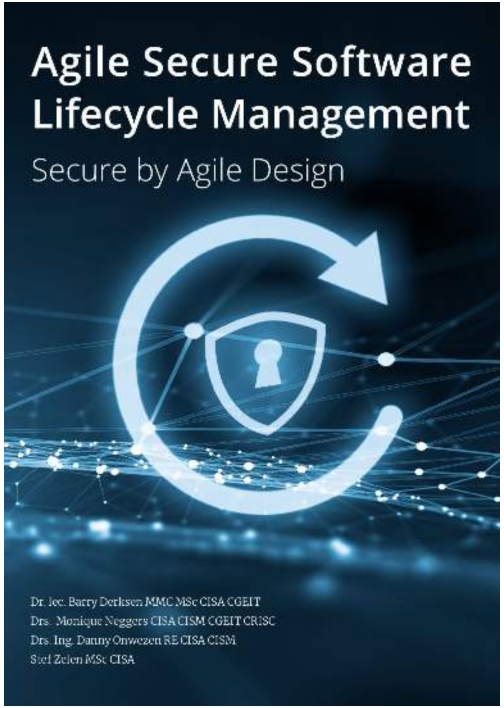
www.securesoftwarealliance.org

Secure Software Alliance

Agile Secure Software Lifecycle Management

Sprints:

1. Because we have to!
2. Developer meets hacker
3. Agile beats structure
4. Software security fundamentals
5. Introducing agile secure software development
6. Agile framework secure software
7. Maturing agile secure software life cycle
8. Roadmap for digital hardware
   and software security



Agile Secure Software
Lifecycle Management
Secure by Agile Design

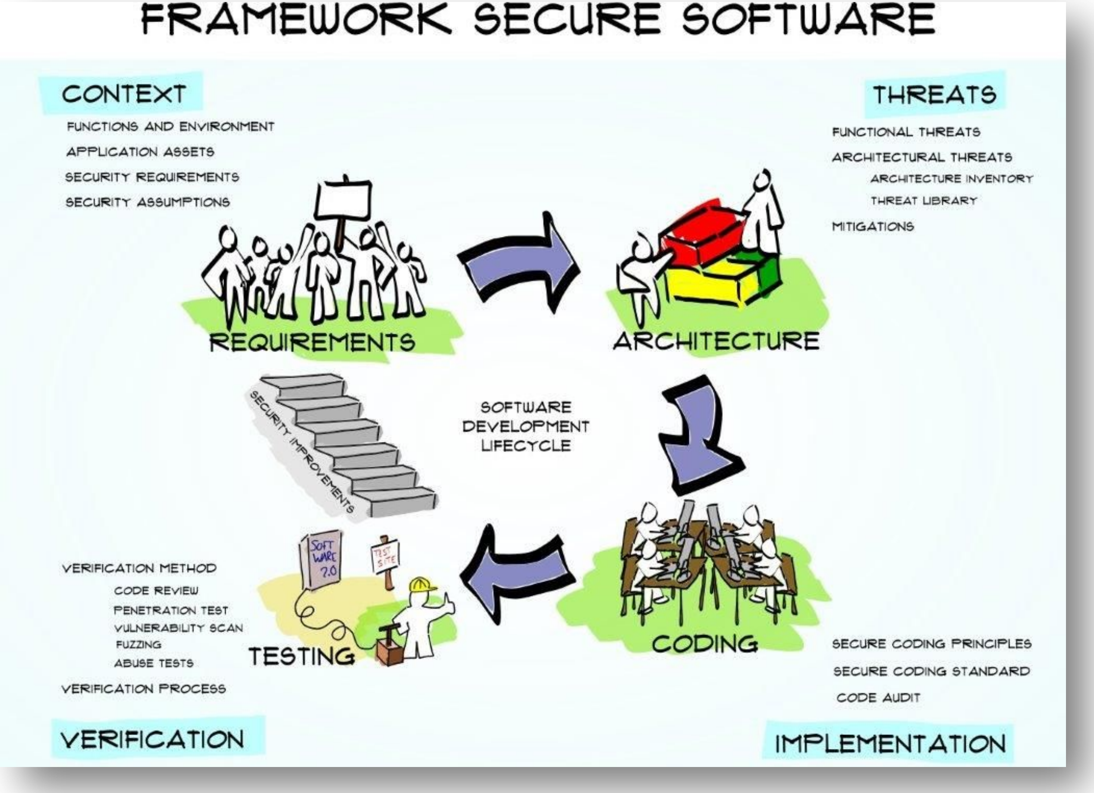
Dr. lec. Barry Derksen MMC MSc CISA CGEIT
Drs. Monique Neggers CISA CISM CGEIT CRISC
Drs. Ing. Danny Onwezen RE CISA CISM
Stef Zelen MSc CISA

Sprint 1: Because we have to!

- Secure software is a challenge
- Software is everywhere!
- Secure software development needs professionals
- Agile secure software development is a contradiction in terms

Sprint 2: Developer meets hacker

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



Abuse cases

“As an employee, I can search for other employees by their last name”

“As a hacker, I can send bad data in the content of requests”

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"

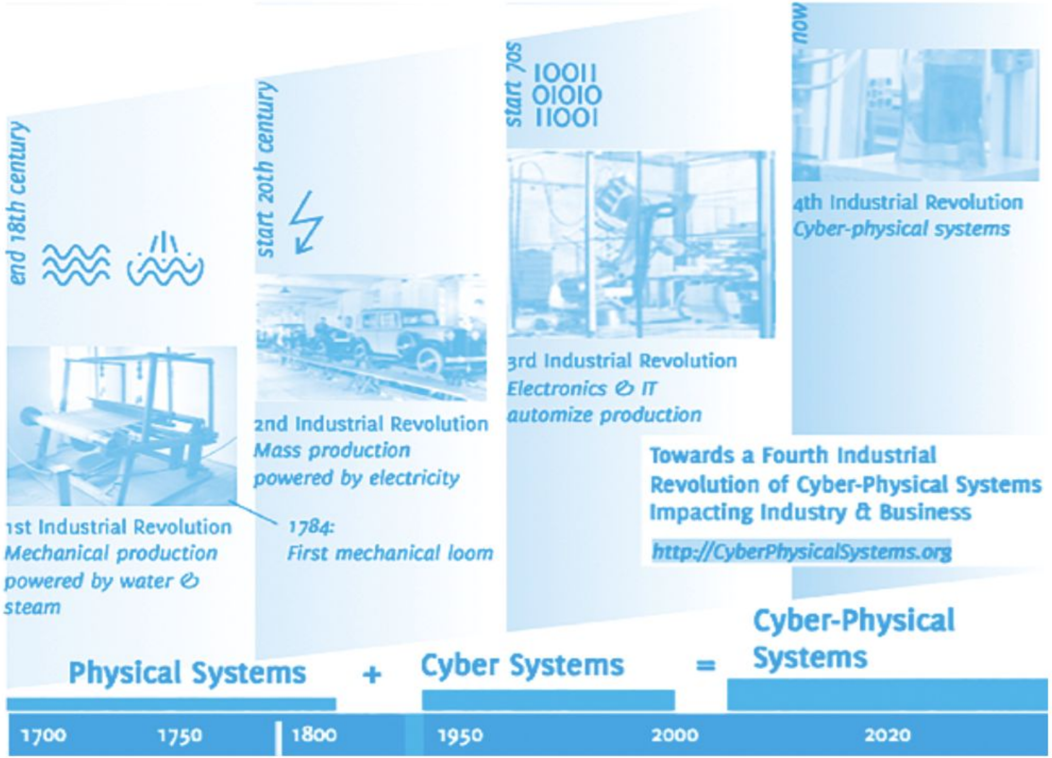
Example #3 "As a hacker, I can read and even modify all data that is input and output by your application"

Sprint 3: Agile beats structure

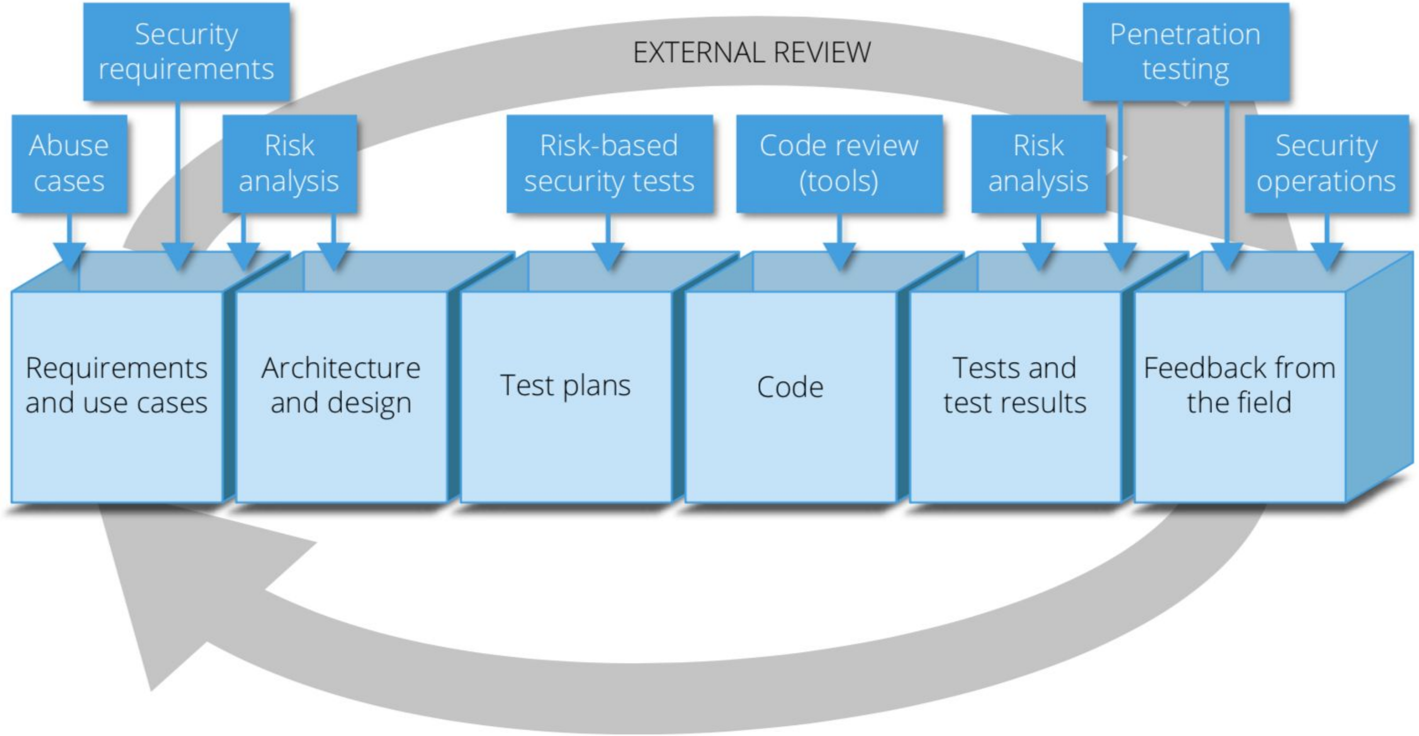
- Social media
- Mobile living
- Analytics & big data
- Cloud
- IoT
- Chain trends

...

Risks change!

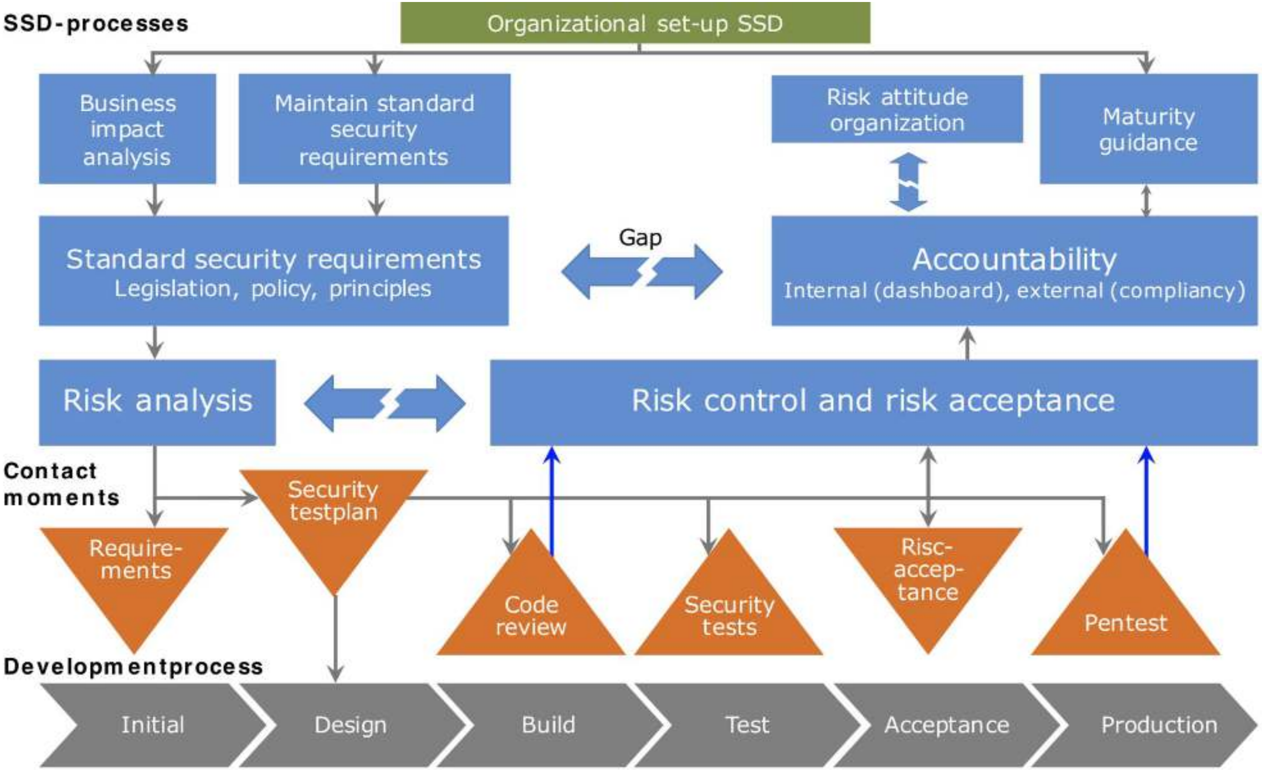


Sprint 4: Software security fundamentals



Secure Software Alliance

Secure Software Development (SSD)

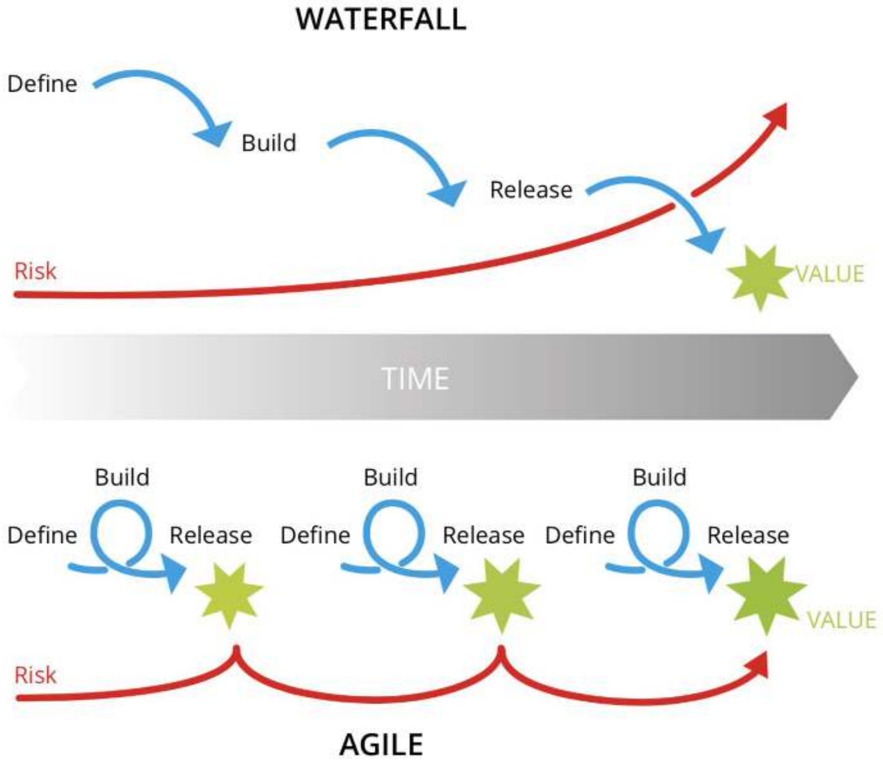


Sprint 5: Agile secure software development

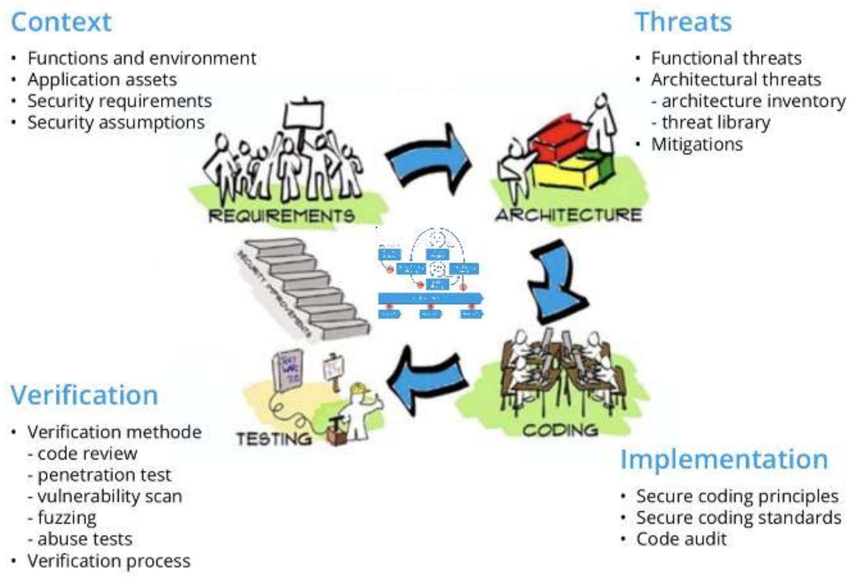
- Stakeholders part of risk assessment

- Stakeholder security tests during product review

- Acceptance criteria for security in user stories



Sprint 6: Agile framework secure software



Context

- Functions and environment
- Application assets
- Security requirements
- Security assumptions

Threats

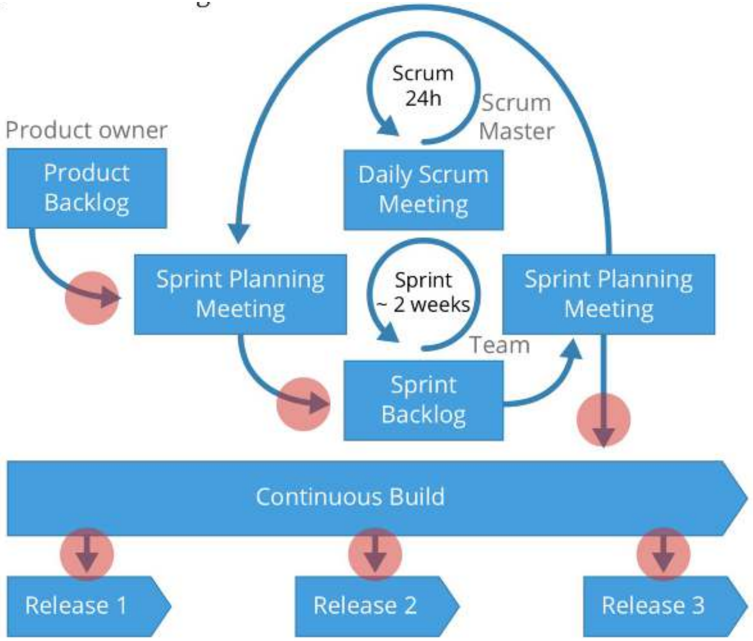
- Functional threats
- Architectural threats
  - architecture inventory
  - threat library
- Mitigations

Verification

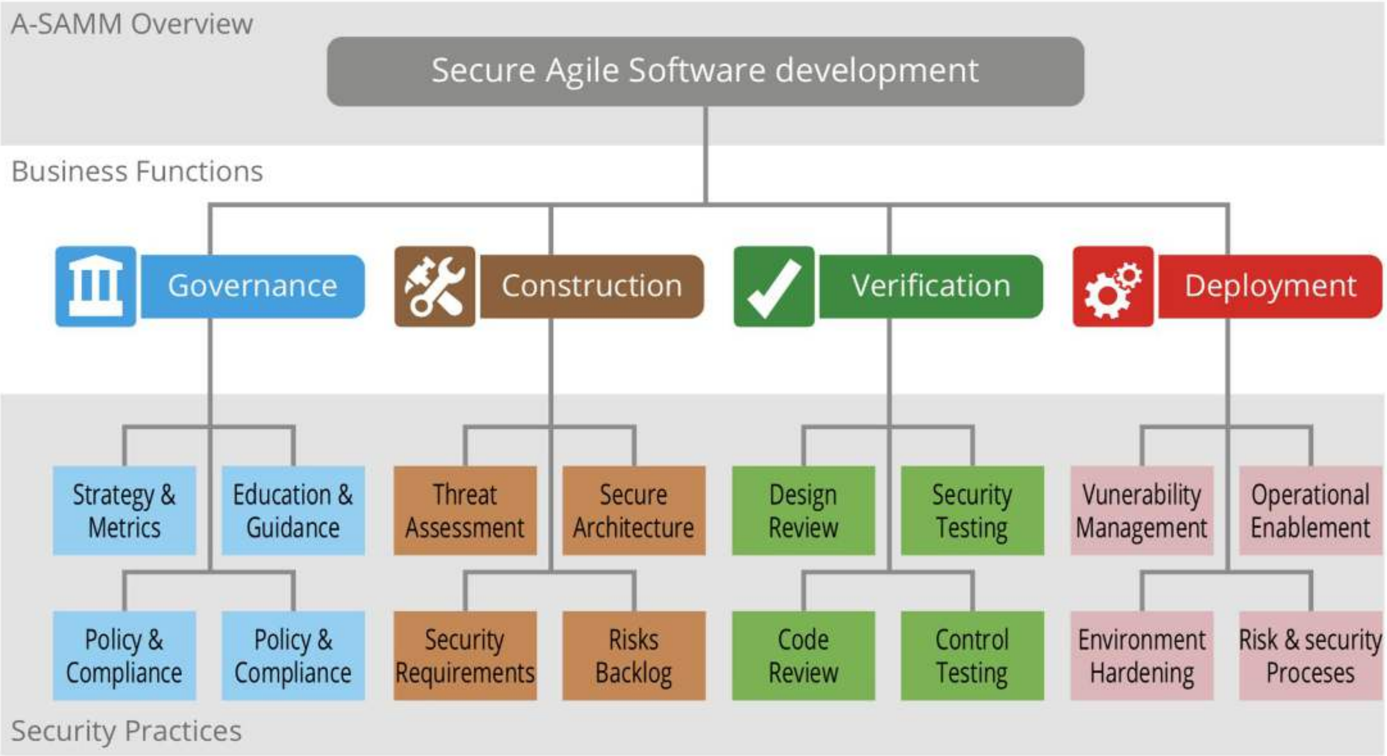
- Verification methode
  - code review
  - penetration test
  - vulnerability scan
  - fuzzing
  - abuse tests
- Verification process

Implementation

- Secure coding principles
- Secure coding standards
- Code audit



Sprint 7: Maturing agile secure software lifecycle



A-SAMM Overview

Secure Agile Software development

Business Functions

Governance

Construction

Verification

Deployment

Strategy & Metrics

Education & Guidance

Threat Assessment

Secure Architecture

Design Review

Security Testing

Vunerability Management

Operational Enablement

Policy & Compliance

Policy & Compliance

Security Requirements

Risks Backlog

Code Review

Control Testing

Environment Hardening

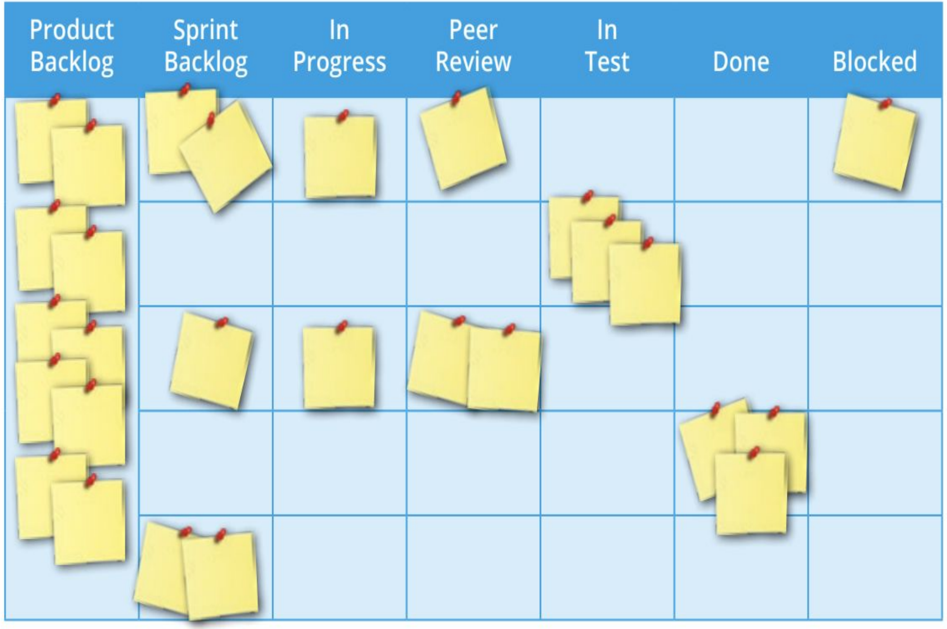
Risk & security Proceses

Security Practices

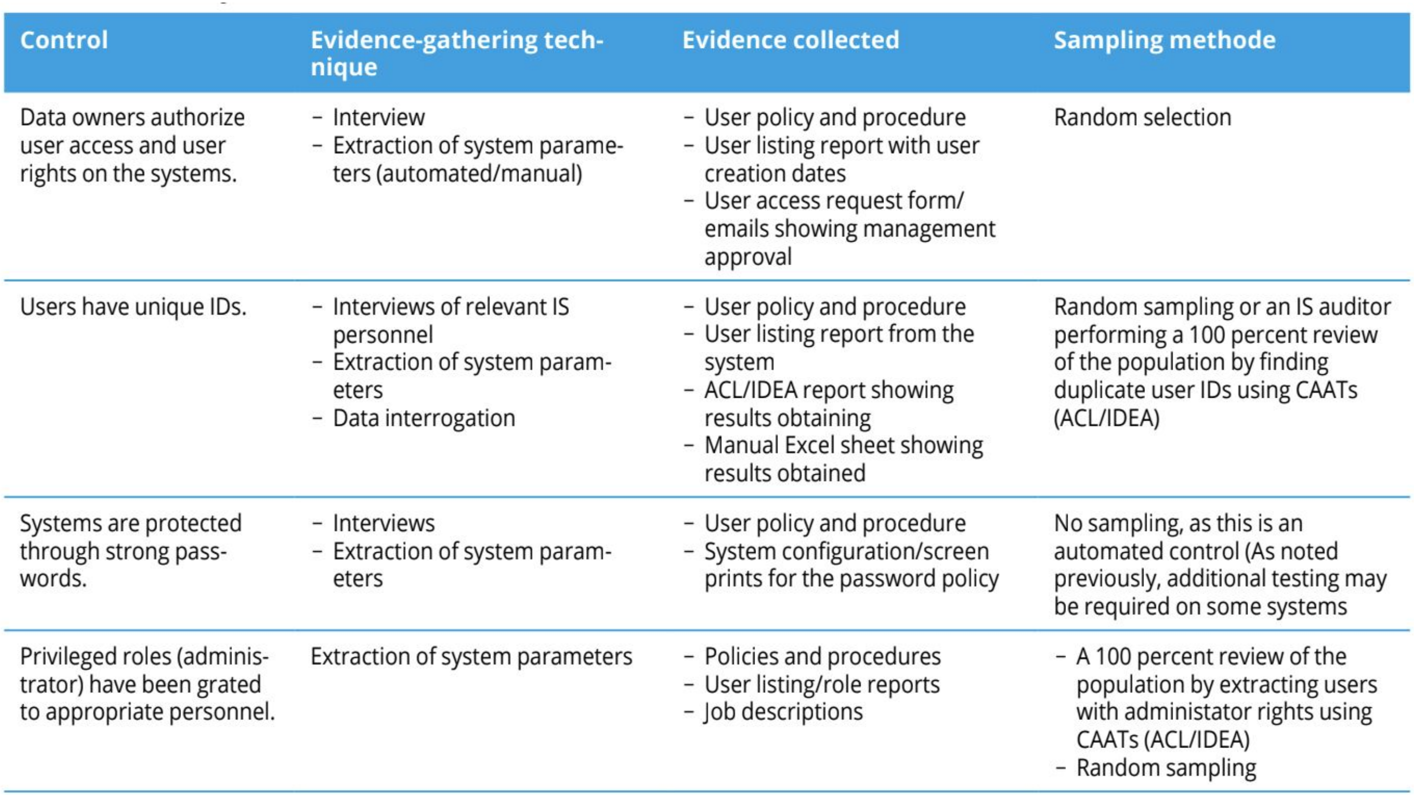
Risk Backlog

- Identifying, analysing and prioritizing risks

- By creating mitigation items in the (product) backlog



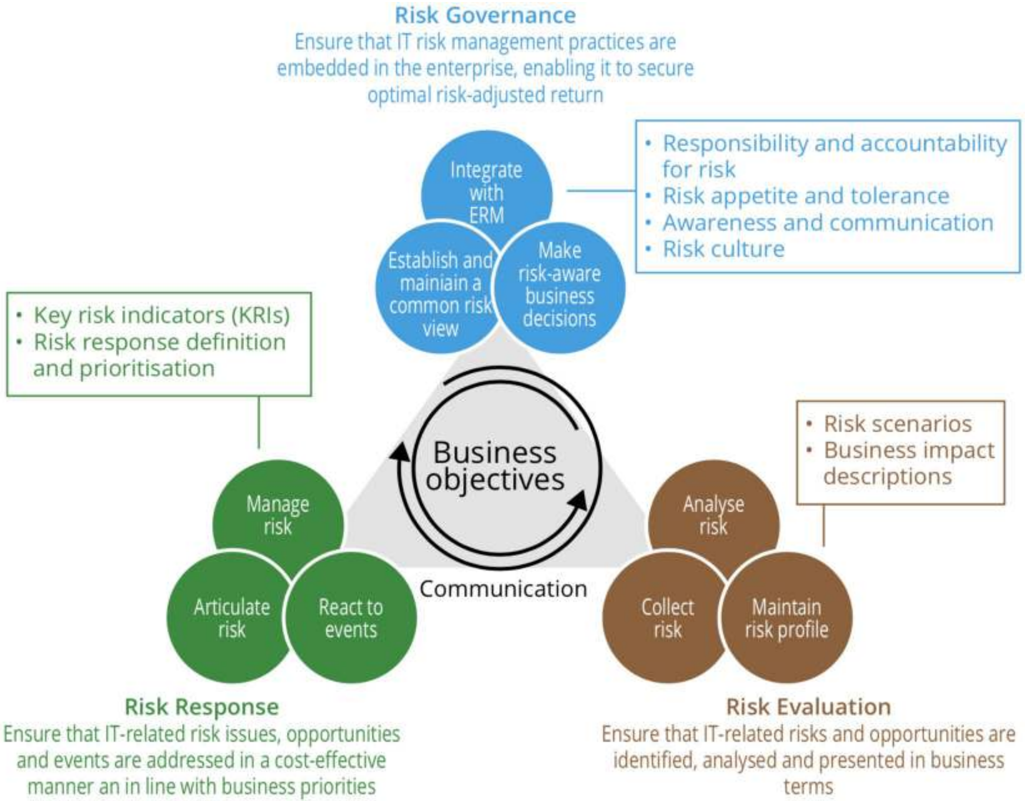
Control Testing



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

Risk & security processes

Integrated Risk Management!



Risk Governance

Ensure that IT risk management practices are embedded in the enterprise, enabling it to secure optimal risk-adjusted return

- Responsibility and accountability for risk
- Risk appetite and tolerance
- Awareness and communication
- Risk culture

Integrate with ERM

Establish and mainiain a common risk view

Make risk-aware business decisions

- Key risk indicators (KRIs)
- Risk response definition and prioritisation

Business objectives

Communication

Manage risk

Articulate risk

React to events

- Risk scenarios
- Business impact descriptions

Analyse risk

Collect risk

Maintain risk profile

Risk Response

Ensure that IT-related risk issues, opportunities and events are addressed in a cost-effective manner an in line with business priorities

Risk Evaluation

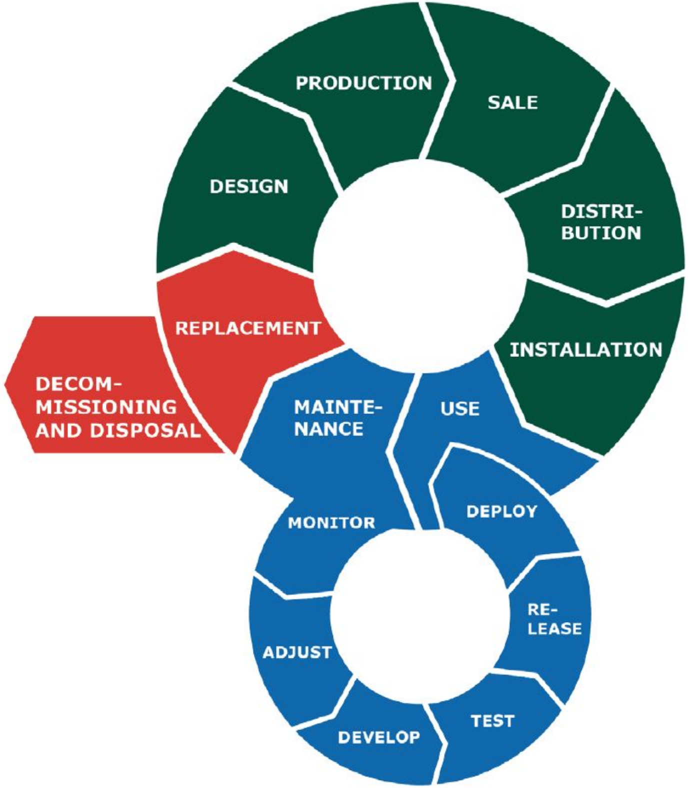
Ensure that IT-related risks and opportunities are identified, analysed and presented in business terms

Sprint 8: Roadmap for digital hardware and software security

Basic principles:

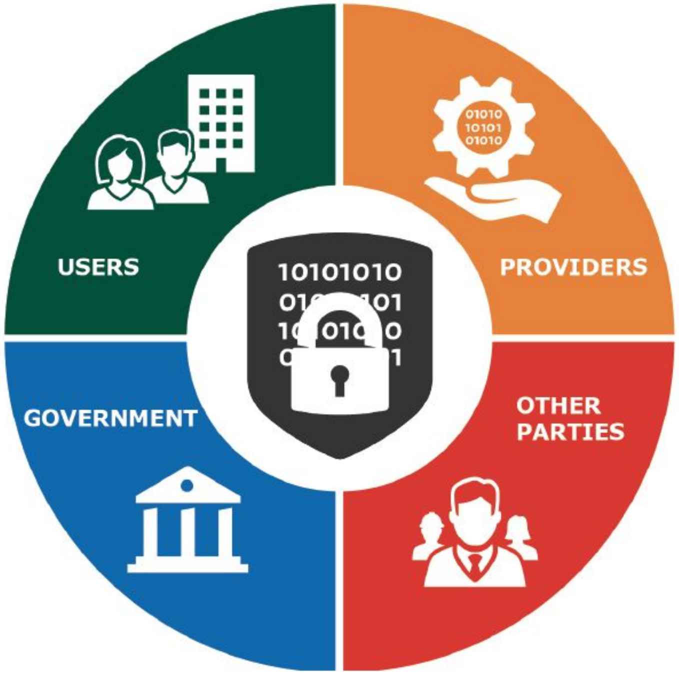
- Product lifecycle approach
- Joined responsibility
- Broad spectrum of instruments
- Balancing public values

All stages in the product life cycle are important for improving the digital security of hardware and software



Secure Software Alliance

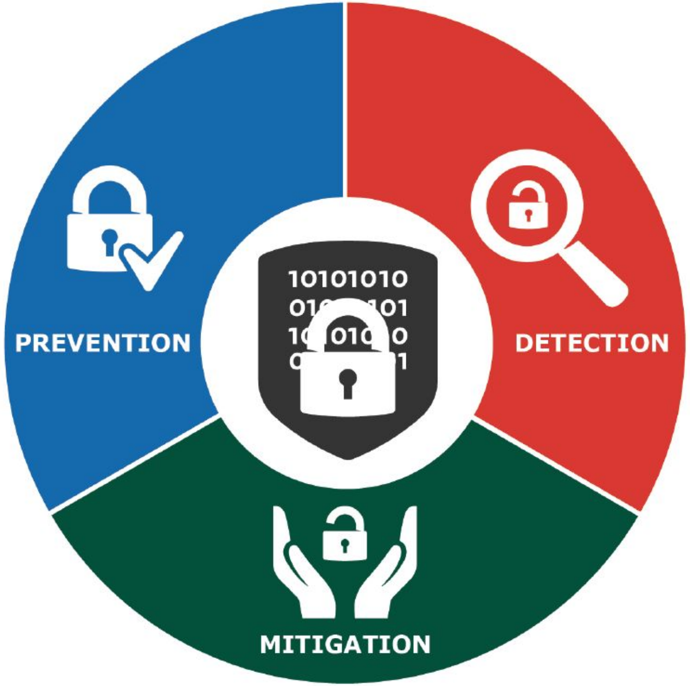
Joined responsibility



- Parties involved have various responsibilities and roles, based on the premise that digitally secure hardware and software requires a concerted approach
- All stakeholders must be involved in promoting digital hardware and software security
- Context based, e.g. B2B, C2B, C2C, critical/non-critical infrastructure) and the party type

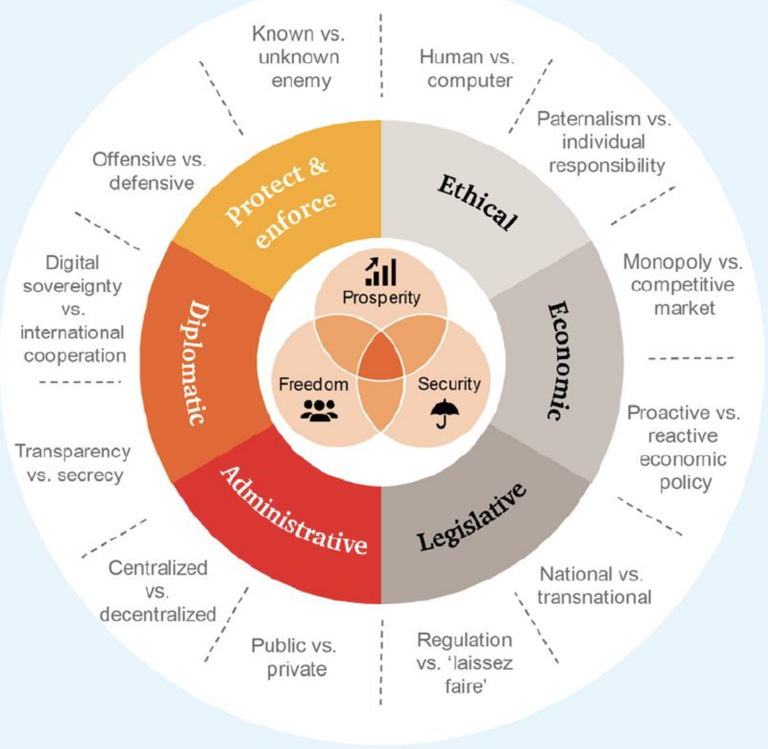
Broad spectrum of instruments

- Promoting digitally secure hardware and software requires a broad spectrum of instruments
- Digital product ecosystem is complex, and vulnerabilities can emerge at various stages of the product life cycle
- Various components of digital products can also result in security risks, each with its own impact



Balancing public values

Core public interests	Dimensions of intervention	Key policy dilemmas
Freedom	Ethical	• Paternalism vs. individual responsibility • Human vs. computer
	Economic	• Protective vs. reactive • Monopolies vs. competitive market
Prosperity	Legislative	• Transnational vs. national • Regulation vs. 'laissez faire'
	Administrative	• Centralized vs. decentralized • Private vs. public
Security	Diplomatic	• Transparency vs. secrecy • Digital sovereignty vs. international cooperation
	Protect & enforce	• Offensive vs. defensive • Known vs. unknown enemy


Known vs. unknown enemy
Human vs. computer
Paternalism vs. individual responsibility
Offensive vs. defensive
Protect & enforce
Ethical
Monopoly vs. competitive market
Digital sovereignty vs. international cooperation
Diplomatic
Prosperity
Freedom
Security
Economic
Proactive vs. reactive economic policy
Transparency vs. secrecy
Administrative
Legislative
Centralized vs. decentralized
National vs. transnational
Public vs. private
Regulation vs. 'laissez faire'

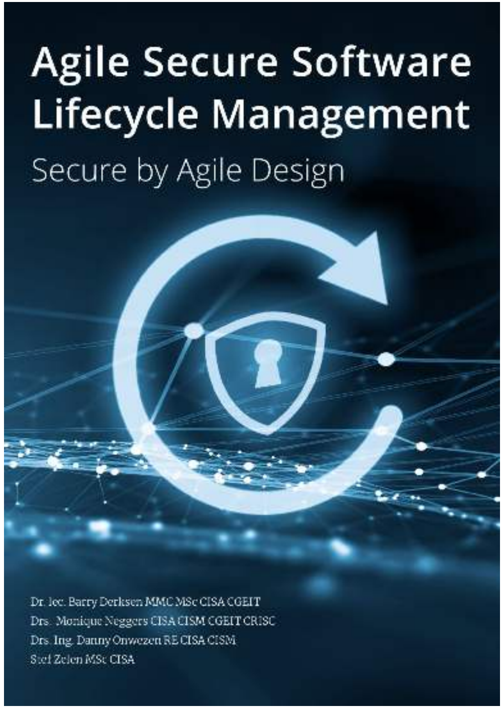
Theses

1. Agile is a team responsibility, this should also be the case with controls
2. Agile is the silver bullet that (finally) makes software development projects successful
3. The product owner must come from IT
4. A risk backlog does not work
5. Work internal audit can be organized agile

Questions?

contact@securesoftwarealliance.org

Thank you for your attention!



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