Application Security and Cheese

SEQA™

Thank You to Our Sponsors and Hosts!

OWASP
NEW
ZEALAND
owasp.org.nz

QUANTUM
SECURITY

AUT

CyberCX

DATACOM

snyk

Auth0

Checkmarx

HCL AppScan

kordia®

LATERAL
SECURITY

MICRO
FOCUS®

Pulse Security
www.pulsesecurity.co.nz

RedShield

Flux

SEQA™
Information Security

Cobalt

LACEWORK

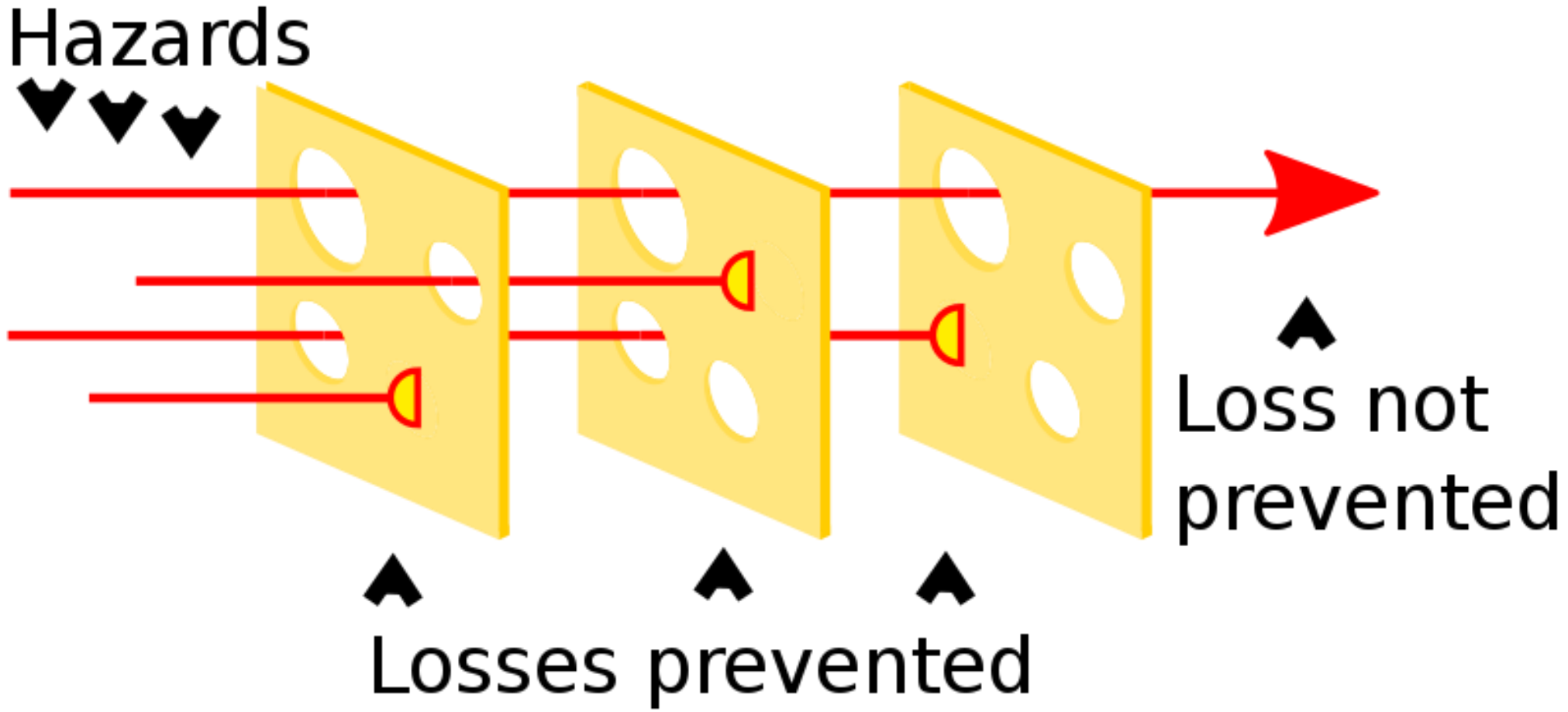
SecureFlag

Without them, OWASP New Zealand Day couldn't happen

Application Security and Cheese?

- What the Swiss Cheese Model is
- Some examples of where it’s been applied
- How it relates to application security
- What a version of good App Sec looks like
- How things can still go wrong & what can be done to prevent them

The Swiss Cheese Model



SEQA

What can cause the holes in the cheese?

Active Failures:

- Slips
- Lapses
- Mistakes
- Violations

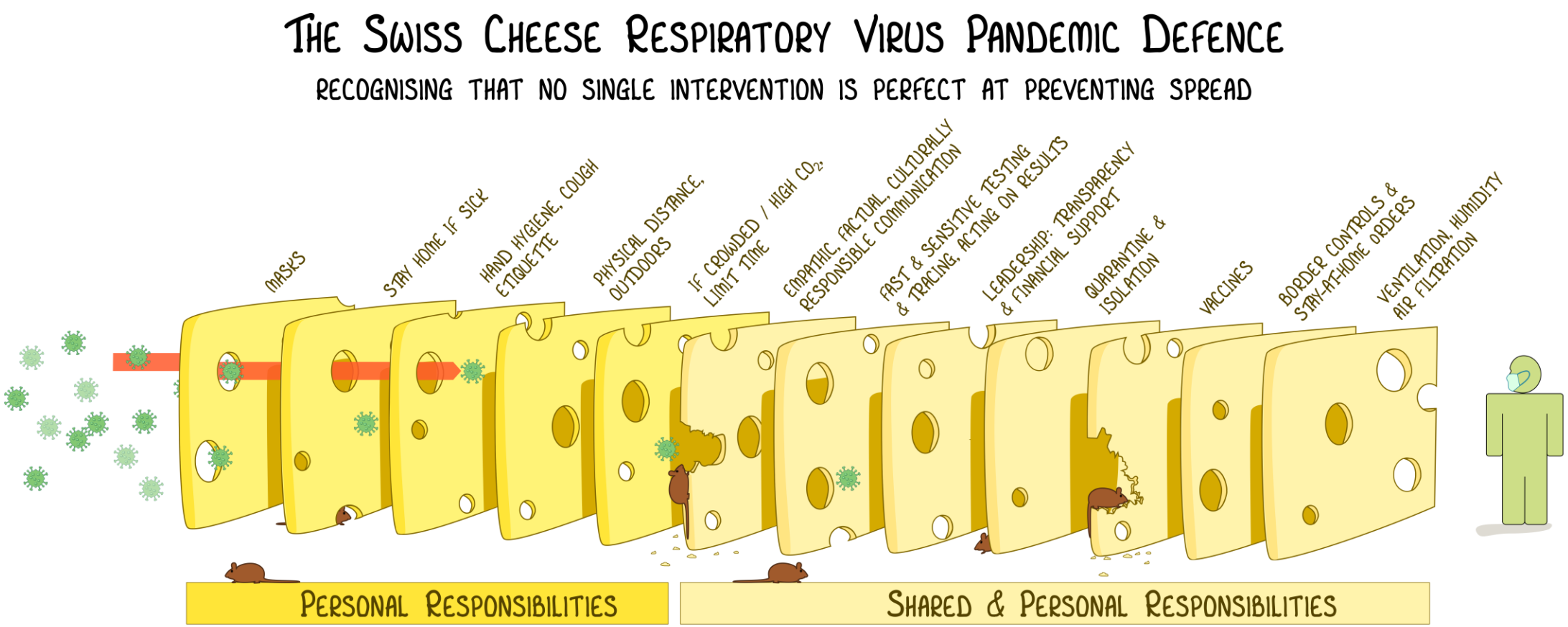
Latent conditions arise from decisions made by:

- Designers
- Builders
- Procedure Writers
- Top level management

Two effects:

- Create error conditions in the local workplace
- Create longstanding holes in the system

Swiss cheese model and COVID

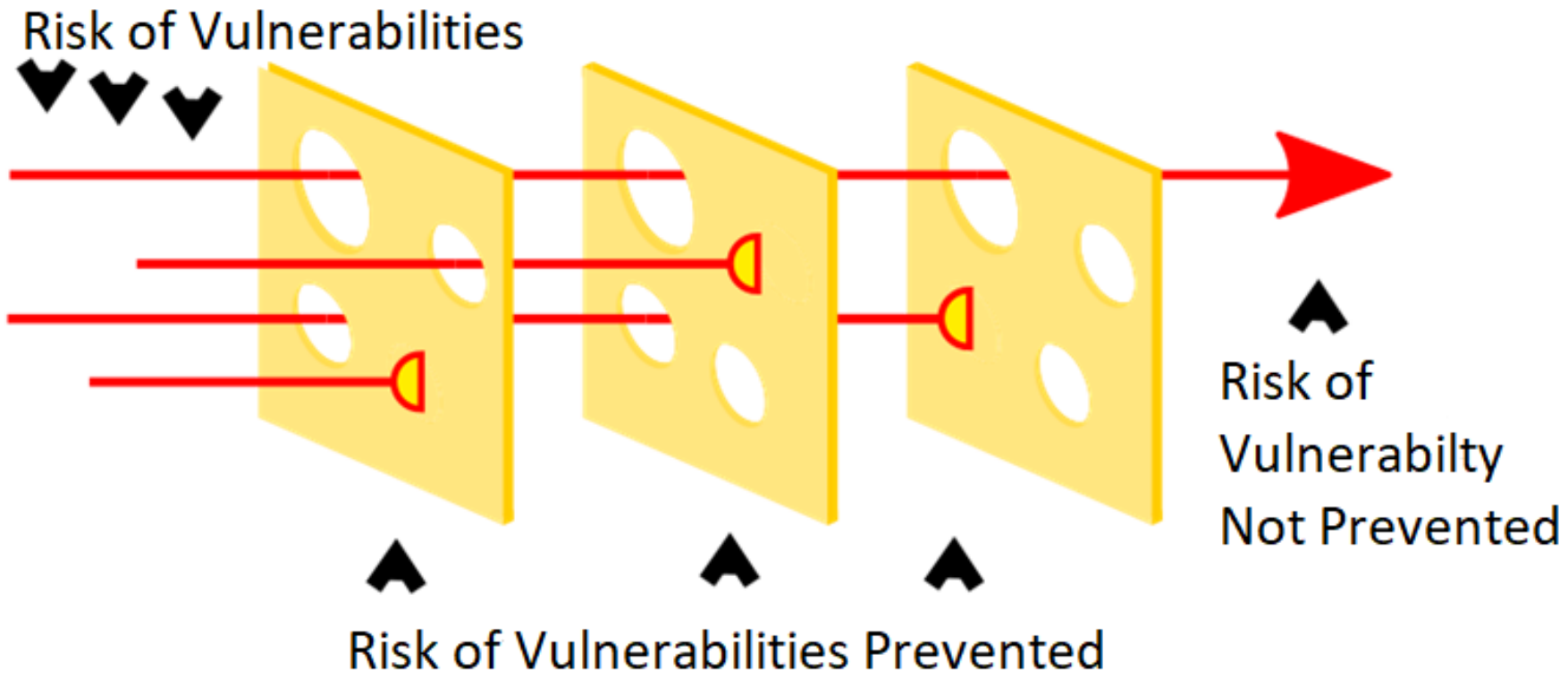


IAN M MACKAY
VIROLOGYDOWNUNDER.COM
WITH THANKS TO JODY LANARD, KATHERINE ARDEN & THE UNI OF QLD
BASED ON THE SWISS CHEESE MODEL OF ACCIDENT CAUSATION, BY JAMES T REASON, 1990
VERSION 4.3
UPDATE: 04SEPT2021

SEQA™

Swiss cheese model applied to App Sec

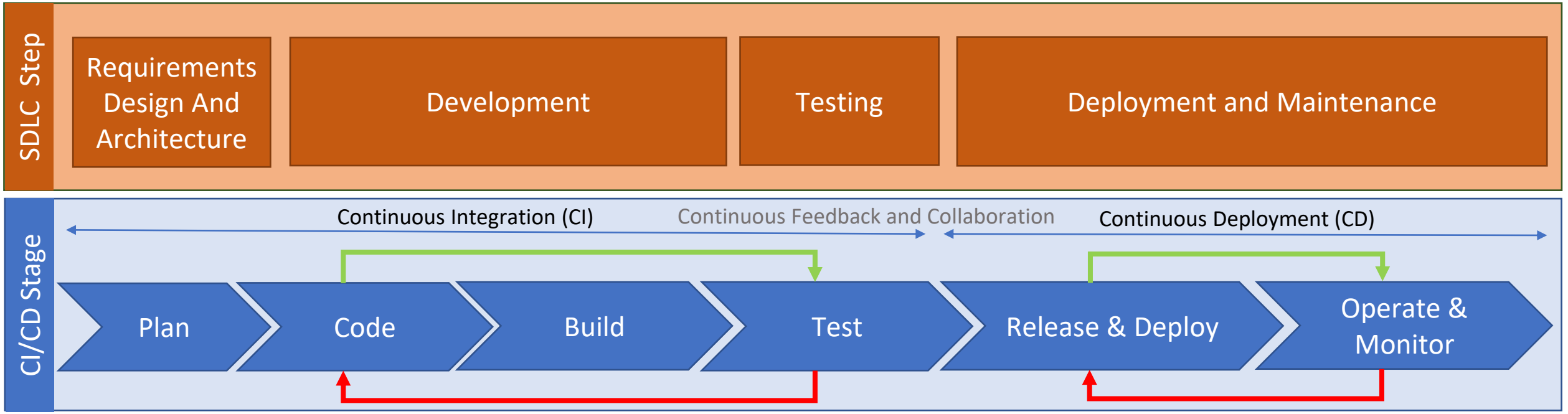
- Can we apply the Swiss cheese model to app sec?
- How can we apply the Swiss cheese model over to app sec?

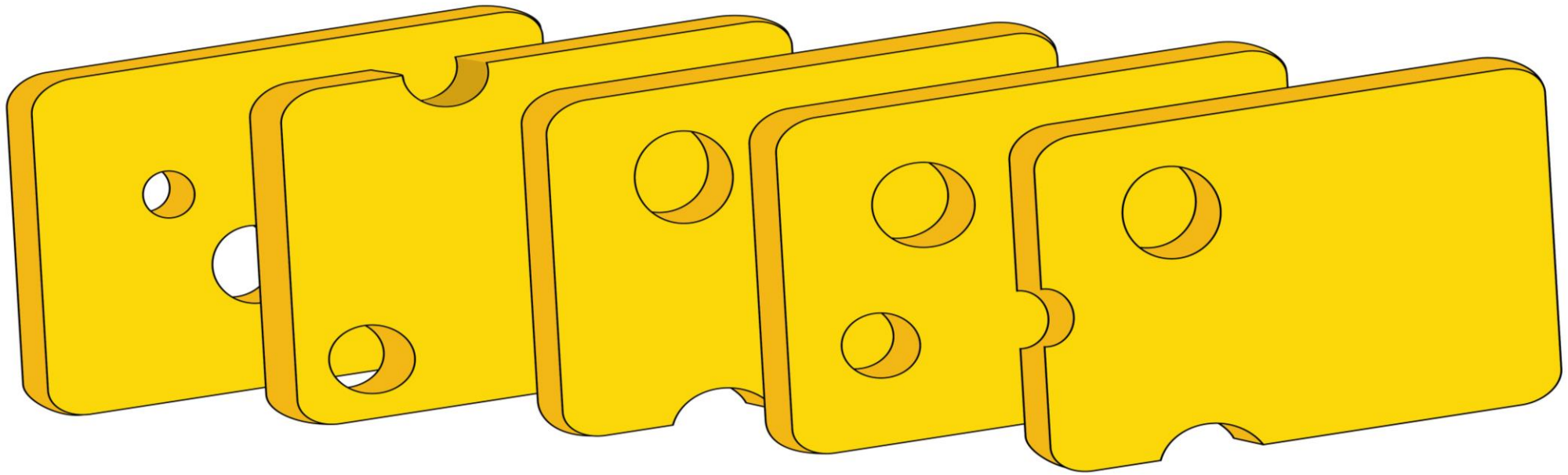


SEQA

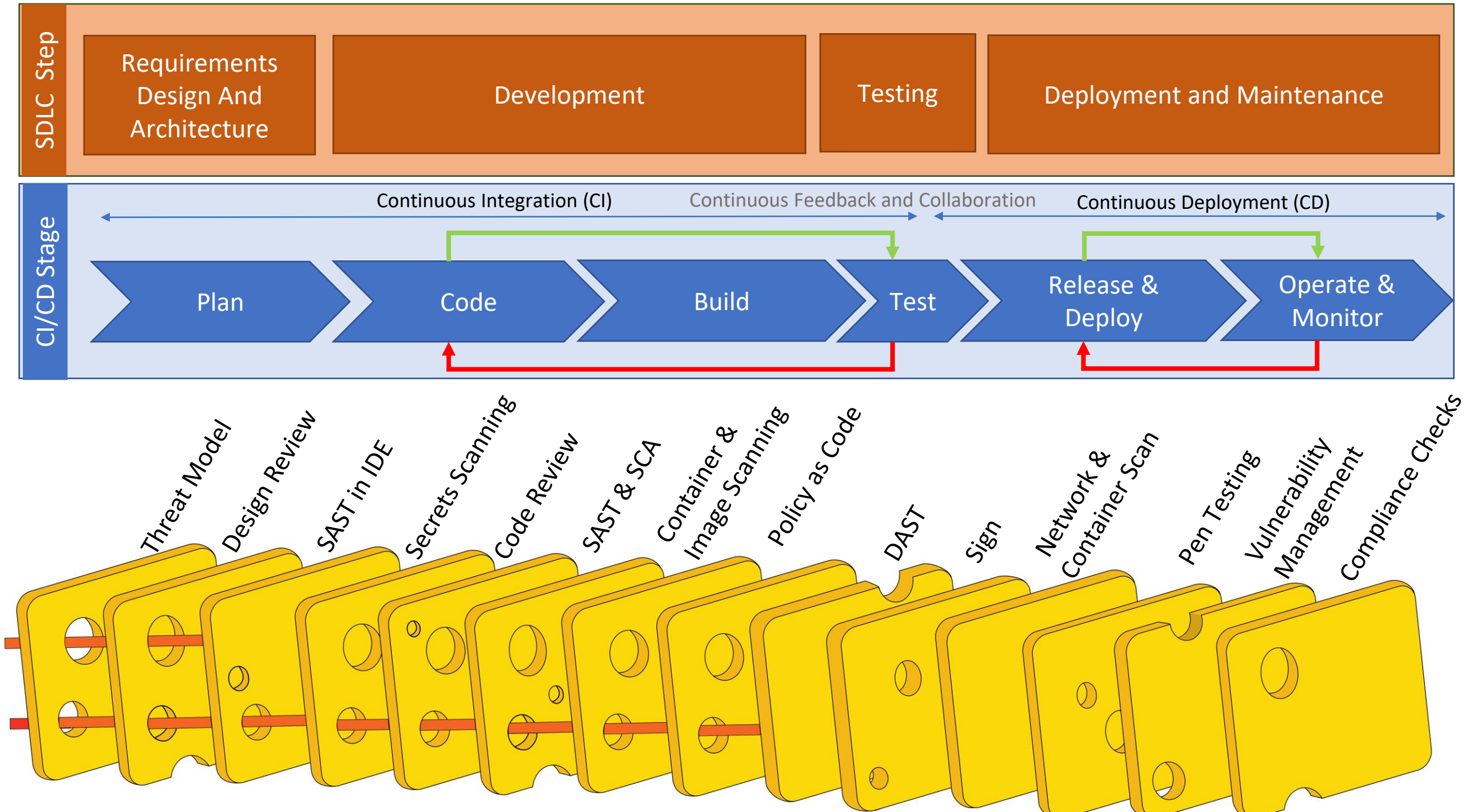
Risks in an Application security context

- Security flaws in the design or architecture of the system.
- Security issues or vulnerabilities in the code.
- Supply chain based vulnerabilities.
- Security issue’s or vulnerabilities in the infrastructure set up
- Malicious Security issues


SDLC Step
Requirements Design And Architecture
Development
Testing
Deployment and Maintenance
CI/CD Stage
Continuous Integration (CI)
Continuous Feedback and Collaboration
Continuous Deployment (CD)
Plan
Code
Build
Test
Release & Deploy
Operate & Monitor



SEQA™


SDLC Step
Requirements Design And Architecture
Development
Testing
Deployment and Maintenance
CI/CD Stage
Continuous Integration (CI)
Continuous Feedback and Collaboration
Continuous Deployment (CD)
Plan
Code
Build
Test
Release & Deploy
Operate & Monitor
Threat Model
Design Review
SAST in IDE
Secrets Scanning
Code Review
SAST & SCA
Container & Image Scanning
Policy as Code
DAST
Sign
Network & Container Scan
Pen Testing
Vulnerability Management
Compliance Checks

What can still go wrong & how can we prevent it

- Not considering security early or on changes
- Setting the thresholds for the scanning tools too high
- Lack of training & supervision for the practice based activities
- Time pressures to delivery functionality
- Lack of skilled security specialists
- Not doing SCA on an ongoing basis

Summary

What can the Swiss Cheese Model teach us about AppSec?

You cant change the human condition but you can change the conditions that humans work under.

- Active Failures – Detective tools to catch these
- Latent Conditions – Design Flaws – Training and Guidance

SEQA™

INTEGRATING **SE**CURITY INTO **Q**UALITY **A**SSURANCE