2.

Sikkerhed i udviklingsprocesser

Sikker software, hvorfor?

- Usikker software
- GDPR
  - Etik
  - Ansvarlighed
- 'Prevention is cheaper than the cure'
- NotPetya omkostninger på $1.2B

Phase	Relative cost to correct
Definition	$1
High-level Design	$2
Low-level Design	$5
Code	$10
Unit test	$15
Integration test	$22
System test	$50
Post-delivery	$100

Hvordan bliver software usikkert?

- Design fejl
  - Privelegier
  - Insecure defaults
  - Defence in depth
- Implementations fejl
  - Input validering
  - Fejlhåndtering
- Maintainence
  - Unpatched software
  - Legacy systemer

Hvordan sikrer man software?

- Højere kvalitet = højere sikkerhed
  - Mindre fejl
  - Test-Driven Development

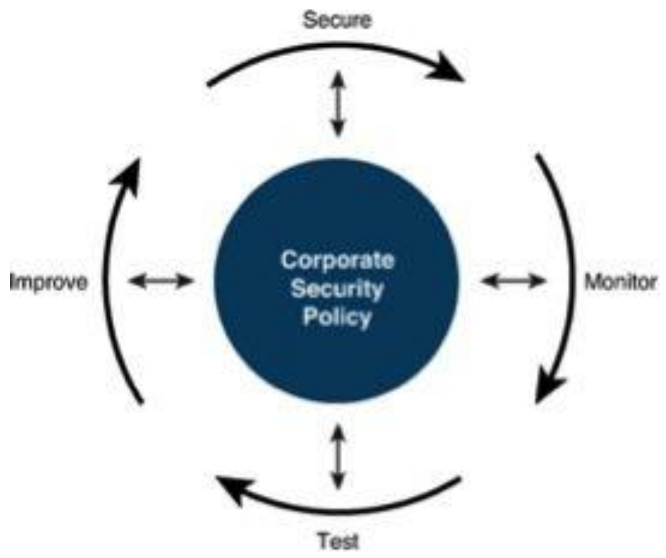
Security principles

- *Minimize attack surface*
- *Establish secure defaults*
- *Principle of least privilege*
- *Principle of defence in depth*
- *Fail securely*
- *Don’t trust services*
- *Separation of duties*
- *Avoid security by obscurity*
- *Keep security simple*
- *Fix security issues correctly*

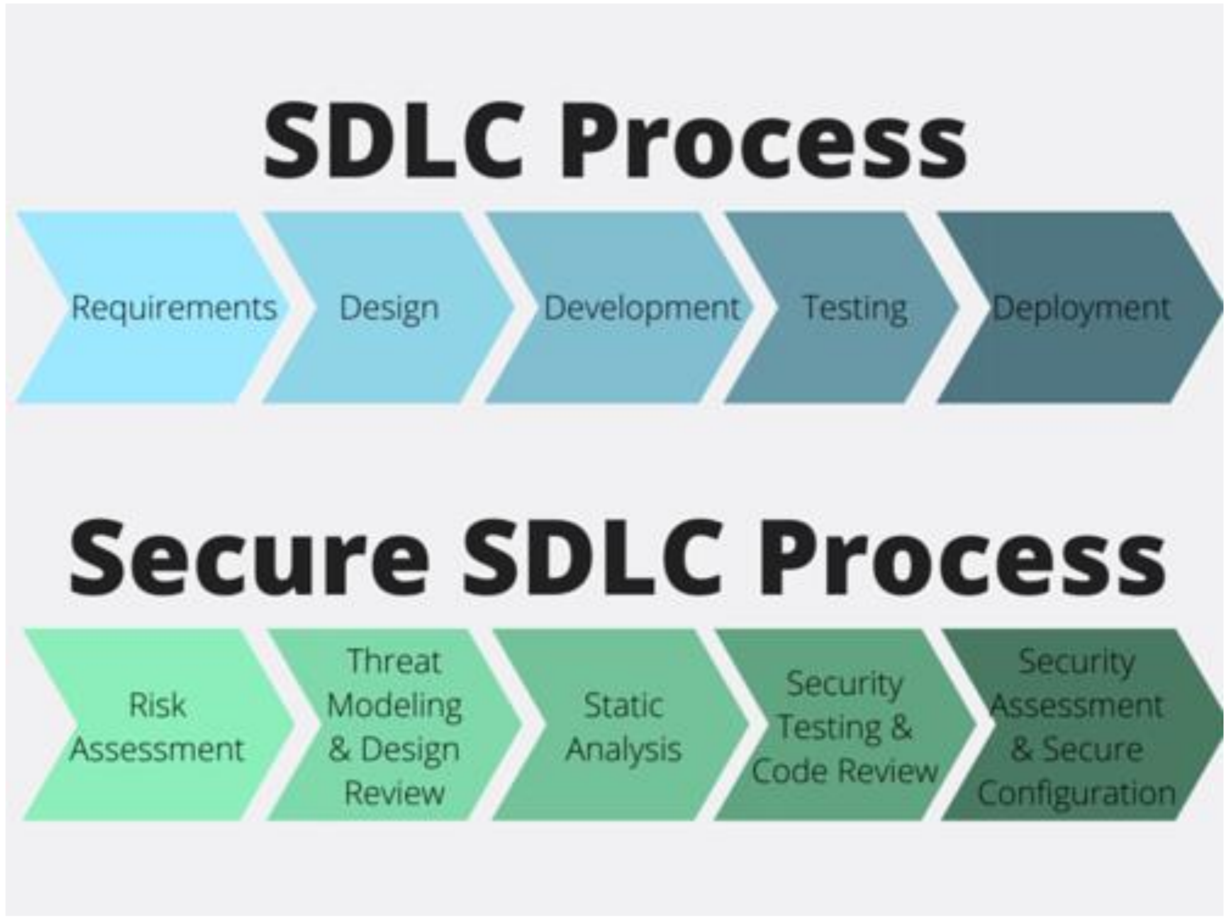
Security is a process

- Bruce Schneier

- … not a product
- Processer og procedurer
  - Vulnerabilities
  - Dokumentation



Secure Software Development Lifecycle


SDLC Process
Requirements
Design
Development
Testing
Deployment
Secure SDLC Process
Risk Assessment
Threat Modeling & Design Review
Static Analysis
Security Testing & Code Review
Security Assessment & Secure Configuration