

6.

Forbedret sikkerhed med opbygning af software i komponenter

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
- Maintenance
 - Patching
 - Udfasning
- Højkvalitetssoftware = Bedre sikkerhed

Ensartet udvikling

- Code patterns
 - Styling
 - Error handling
 - Testing
- Design patterns
 - Singleton
 - Facade
 - Iterator
 - Og mange flere...
- Dokumentation

External software trust

- Hvilket software kan vi stole på?
- Meget kendt versus ukendt

Common Data Structures

- Cryptografi
- Parsers
- Hashing

Biblioteker, moduler og frameworks

- De fleste problemer er løst af andre
- Python
 - Python XML
 - Python Hashlib
 - JSON