MICROFT SECURITY DEVELOPMENT LIFECYCLE (MSDL)

- Microsoftin kehittämä ja käyttämä ohjelmistokehitysmalli
- Vista toteuttaa ensimmäisen kerran
- Periaatteet ovat julkisesti saatavilla: https://www.microsoft.com/en-us/sdl/default.aspx
- Minimaalinen malli
- Muuttuva malli
- Ei sidottu käyttöjärjestelmään tai laitteistoon.
- Voi soveltaa erilaisiin ohjelmistokehitysprosesseihin
- Voi käyttää myös pienissä organisaatioissa.

MSDL

- Turvallisuus ja yksityisyyden suoja (privacy) ovat läpinäkyvä osa prosessia.
- Yhdistäminen organisaation omaan ohelmistokehitysprosessiin voi olla työlästä.
  - Edellyttää mm. johdon tukea

MSDL

Viisi aluetta

- Valmennus, politiikka ja organisaationaaliset kyvyt (Training, policy, and organizational capabilities)
- Vaatimukset ja suunnittelu (Requirements and design)
- Toteutus (Implementation)
- Verifiointi (Verification)
- Julkaisu ja ylläpito (Release and response)

MSDL

Neljä kypsyystasoa

- Basic (vähän tai ei ollenkaan valmennusta, prosessia ja työkaluja)
- Standardized
- Advanced
- Dynamic (SDL täydellisenä käytössä läpi organisaation)

MSDL

SDL:n käyttö tulisi Microsoftin mukaan kohdistaa sellaisten sovellusten kehitystyöhön,

- joita käytetään yritys- tai vastaavissa ympäristöissä
- jotka käsittelevät henkilötietoja tai muuta yksityistä tietoa.
- käyttävät säännöllisesti Internetiä tai muuta verkkoa kommunikointiin

MSDL

Erilaisia rooleja

- Advisor/reviewer voi hylätä ja hyväksyä projektitiimin turvallisuus/yksityisyys-suunnitelmät
  - Mieluiten sisäinen ryhmä
  - Kaksi aliroolia, auditoija ja asiantuntija (expert)
- ”Team champions”
  - Koordinoivat ja seuraavat tietoturvan ja yksityisyyden suojan toteutumista

MSDL

Käsitteitä

- ”Quality gate” on vähimmäisvaatimus, tietoturvallisuudelle ja yksityisyydelle (privacy)
- ”Bug bar” on”quality gate”, joka on voimassa koko prosessin ajan.
  - Ei pitäisi löysätä.
- ”Security review” -vaiheessa pitää osoittaa, että vaatimukset täyttyvät
- Agile versio: https://www.microsoft.com/en-us/SDL/Discover/sdlagile.aspx

Vaiheet Microsoft SDL:n kuvauksesta!

VILLE YLIMANNELA: RISK MANAGEMENT IN AGILE SOFTWARE DEVELOPMENT

- Kahden yrityksen haastattelun perusteella löydettyjä ongelmia:
  - Resurssit: riskien hallinta vie aikaa
  - Riskeillä ei ole omistajia
  - Kuka hyväksyy jäljelle jäävät riskit (jäännösriski, residual risk)
  - Tietoturvakoulutuksen puute, erityisesti johtajat
  - Riskien kommunikointi: Mitkä riskit kommunikoidaan omistajalle?
  - Jos riski ei suoraan liity backlogiin, ohitetaan
  - DoD:stä puuttuu tietoturvanäkökulma

THE SECURE AGILE SOFTWARE DEVELOPMENT LIFE CYCLE

- http://www.n4s.fi/2014magazine/article2/assets/guidebook_handbook.pdf

-> Tenttiin

STAATTISET ANALYSAATTORIT

- Ohjelmia, joilla voidaan automattisesti analysoida lähdekoodia ja etsiä virheitä.
- Eivät löydä kaikkea.
- Löytävät kuitenkin nopeasti virheitä, joita ei yhtä nopeasti havaitse manuaalisella koodin katselmoinnilla.

FUZZ-TESTAUS

- Ohjelmia, joilla voidaan analysoida lähdekoodia ja etsiä virheitä
- Perustuu mustaan tai valkoiseen laatikkoon
- Lähettää parametrisoitua satunnaistettua dataa
- Perustuu suureen määrään
- Esim. Oulun yliopistossa kehitetty avoimen lähdekoodin Radamsa https://www.ee.oulu.fi/research/ouspg/Radamsa

TUNKEUTUMISEN TESTAAMINEN (PENETRATION TESTING)

- Tyypillisesti käytetään valmiita työkaluja, joilla tarkistetaan löytyykö testattavasta järjestelmästä haavoittuvuuksia.
- Esim. Metasploit, Nmap
- Hyökkääjät käyttävät kuitenkin, joten parempi käyttää ensin itse.