Teksti: Teemu Vesala

Ohjelmistostandardeja laidasta laitaan

Teemu Vesala toimii laatukonsulttina Qentinel Oy:ssa. Työssään Vesala käyttää standardeja. Testauksessa hän keskittyy erityisesti tekniseen laatuun ja tietoturvaan. Vapaa-ajallaan Teemu rakentelee legoilla, lukee ja nörtteilee.

Ostaja:
”Miksi muuten luottokorttinumeroiden-käsittelyjärjestelmä katkaisee yhteyden palvelimeen, kun käyttäjä painaa enteriä? Me määrittelimme, että teidän olisi käytettävä telnet-protokollaa.”
(IETF STD-8)

”Vaatimusmäärittelyt eivät kuulu missään tapauksessa hyvään ohjelmiston-hakintaprosessiin! Huomasimme myös, että oikeuksienhallinnassa on virhe, joka antaa jokaisen luottokorttinumeronsa syöttävän tulostaa kaikkien muiden järjestelmässä olevien luottokorttien tiedot.”
(PCI-DSS 1.2)

”Miten niin käytettävyyttä? Sehän on selvästi osa suorituskykyä!” (ISO-25010)

Toimittaja:
”Me käytämme MegaHard:in telnet-toteutusta. MegaHardin ja teidän Moon-palvelinten telnet-toteutukset eivät ole yhteensopivia. Me emme saaneet teiltä minkäänlaista vaatimusmäärittelyä, joten ongelma lienee teidän päässänne enemmän kuin meidän.” (ISO-12207)

”Ette pyytäneet testaamaan tuollaista asiaa, joka ei kuulu toiminnallisen testauksen piiriin, vaan on osa käytettävyyttä.”

Ohjelmistojen standardit ovat muodostuneet hyvin erilaisten tarpeiden pohjalta. Alun keskustelussa tarvittiin tekniikkaan, ostoprosessiin, testauksen sisältöön ja yhteiseen sanastoon kuuluvia standardeja. Eri tyylisiä standardeja tulisi ymmärtää eri tavoin.

Konkreettisin standardisto liittyy yleensä tekniikkaan ja on osa ohjelmiston toiminnallista laatua. Vielä 1990-luvulla käytettiin hyvin yleisesti telnet-protokollaa, kun haluttiin kirjautua työasemalta palvelimelle. Ilman protokollan standardointia jokaisella valmistajalla olisi ollut lopulta oma, toisten valmistajien kanssa yhteensopimaton toteutus. Telnet kuitenkin määriteltiin IETF STD-8:ssa, jonkas pohjalta jokainen toimittaja voi tehdä toisten toimittajien kanssa yhteensopivan telnet-toteutuksen. Ostajan kustannusten kannalta standardit protokollat ovat eduksi. Tekniset standardit on useimmiten otettava hyvin kirjaimellisesti.

Standardoituun rajapintaan voidaan tehdä standardeja testejä. Esimerkiksi Sun on tehnyt Javalleen rajapintojen testaamiseksi työkalut[1].

Standardit voidaan määritellä pakollisiksi jonkin kolmannen osapuolen toimesta. Luottokorttitransaktioihin liittyvä tietoturvastandardi PCI DSS on sitova lähes kaikille, jotka käsittelevät luottokorttien tietoja. PCI DSS 1.2:ssa luodaan järjestelmän infrastruktuurille tietoturvavaatimuksia. Niihin kuuluvat esimerkiksi tunkeutumisen havainnointijärjestelmän käyttö. Standardi luo myös vaatimuksia ohjelmiston testausprosesseille. Siinä listataan asioita, joita websovelluksesta on testattava. Määritelmä on muuttuva ja määrää, että testauksessa on testattava OWASP Top 10-listassa[2] lueteltuja tietoturva-aukkoja. Standardi vaatii monien käytäntöjen olemassa olon, mutta käytännöt voidaan järjestää organisaation omilla tavoilla. Tekniset vaatimukset taas standardissa ovat hyvinkin tarkat.

Prosesseihin liittyvät standardit ovat huomattavasti abstraktimpia kuin tekniikkaan liittyvät. ISO-12207 on standardi, joka kuvaa ohjelmistojen elinkaarta. Se antaa kuvauksen niistä käytännöistä, joita hyvän ohjelmistohankinnan tulisi sisältää. Se ei kerro, miten käytännöt toteutetaan. Standardi on abstrakti. Se ei vaadi käyttämään vesiputousmallia, vaan jokainen organisaatio pystyy sovittamaan standardin omiin toimintamalleihinsa ja kehittämään prosessejaan ilman prosessimallin muuttamista standardin mukaiseksi.

ISO-250nn on standardiperhe, jossa osasta riippuen löytyy hyvinkin konkreettisia standardeja. Standardiperheen perustana on ISO-25010:ssa esitelty SquARE-laatumalli. Laatumalli pyrkii kuvaamaan, mahdollisimman konkreettisella tavalla, mitä laatu tarkoittaa. Se tulisi ottaa sellaisenaan, jotta organisaatiot voivat keskustella yhteisellä kielellä samasta asiasta. Samalla kuitenkin on hyvä ymmärtää, että laatumallin jokainen ulottuvuus ei välttämättä koske jokaista ohjelmistoa.

ISO-25010 määrittelee sisäisen ja ulkoisen laadun ominaisuuksiksi funktionaalisuuden, luotettavuuden, suorituskyvyn tehokkuuden, käytettävyyden (operability), tietoturvallisuuden, yhteensopivuuden, ylläpidettävyyden ja siirrettävyyden. Nämä on vielä jaettu pienemmiksi osiksi. Näiden lisäksi määritellään käytönaikainen laatu (quality in use), joka kuvaa käytössä olevan järjestelmän laatuominaisuuksia. Suurin ero näissä on se, että sisäinen ja ulkoinen laatu voidaan mitata testauksen aikana. Käytönaikaisen laadun mittaamiseen taas liittyy aika, joka voi olla hyvinkin pitkä. Käytönaikaisesta laadusta esimerkiksi turvallisuus (safety) sisältää operaattorin terveyden ja turvallisuuden, yleisen terveyden ja turvallisuuden, ja ympäristölle vahingollisuuden. Mittarina voisi olla esimerkiksi operaattoreiden kuolemien määrä vuodessa. Alun keskustelussa olevassa ohjelmistossa tuskin tarvitsee ottaa huomioon turvallisuutta. Sille taas tärkeä ominaisuus on tietoturvallisuus, joka on sisäinen ja ulkoinen laatuominaisuus.

ISO-25010-standardiin liittyy teoreettinen lähestyminen. Lähtökohtaisena ajatuksena standardia luotaessa on ollut, että määritellään toivottu laatutaso käytönaikaiseen laatuun. Se on laatu, jonka loppukäyttäjä kokee. Sen pohjalta määritellään ulkoiset mittarit ja raja-arvot, joilla voidaan saavuttaa vaadittu laatu. Näiden pohjalta taas lähdetään määrittelemään sisäinen mittaristo ja raja-arvot, joilla voidaan saavuttaa haluttu taso. Käytännössä on erittäin vaikeaa löytää korrelaatiota eri mittareiden välillä, joten malli jäänee suurimmaksi osaksi vain teoreettiseksi.

Standardiperheessä ISO-2502n koskee mittausta. Standardeista ovat valmiina vain ISO-25020 ja ISO-25021. Näistä kummatkaan eivät sisällä varsinaisia laatumalliin liittyviä mittareita, vaan mittareiden löytämiseen on käytettävä ISO-9126-2, ISO-9126-3 ja ISO-9126-4 teknisiä raportteja. Tulevaisuudessa kuitenkin olisi tarkoitus korvata nämä ISO-2502n-sarjan standardeilla. Mittareita ei kuitenkaan voida pitää muuta kuin esimerkinomaisina. Niiden pohjalta voidaan sijoittaa organisaation oma ja toimivaksi havaittu mittaristo standardien laatumalliin.

Standardin lukeminen vaatii aina standardin taustan ja kohdeyleisön ymmärtämistä. ISO-12207 ei ole kehittäjien standardi, kun taas IETF STD-8 on.

Viitteet

[1] http://java.sun.com/developer/technicalArticles/JCPtools/

[2] OWASP Top 10 2007 http://www.owasp.org/images/e/e8/OWASP_Top_10_2007.pdf