



Erkki Pöyhönen, QA Business Development Manager  
TietoEnator, QACC Quality Assurance Competence Center

Nykyisin TietoEnatorilla vaikuttava Erkki Pöyhönen on aikaisemmin työskennellyt Nokiassa monissa eri testaus- ja kehittämisrooleissa yhteensä 13 vuoden ajan. Hänen tontillaan on ollut erityisesti testauksen menetelmien ja testauskoulutuksen kehittäminen eri yksiköiden tarpeisiin. Vuonna 2001 hän perusti suomalaisen testaajakerhon testaajien osaamisen ja ammatti-identiteetin kehittämiseksi ja vuonna 2003 Suomen testausraadın kansainvälisen testaajsertifikaatin valmisteluun ISTQB:n kautta (<http://www.istqb.org>). Erkki on toiminut myös mm. EuroSTAR2004-konferenssin puheenjohtajana ja on Tietotekniikan Liiton hallituksen jäsen. Nykyisin Erkin roolissa on testauksen palvelujen ja testausosaamisen kehittäminen QACC:ssä, TietoEnatorin testausasiantuntijaorganisaatiossa.

# Kehityksen testaus ja testaustiimi

## ONGELMANA OHJELMISTOKEHITYKSEN TESTAUSOSAAMINEN

**Yksi asia alkaa olla nykyään kaikkien testauspäälliköiden viesti, kun keskustellaan järjestelmän laadusta ja testauksen roolista: Ohjelmiston laatu tehdään oikeissa vaatimuksissa ja huolellisella suunnittelulla ja ohjelmoinnilla; testaus ei tee laatua. Testaus lähinnä tuottaa laatutietoa järjestelmistä ja osoittaa kohteita laadun parantamiselle (lue: ohjelmistovikoja). Kuitenkin on yleistä, että kehitystiimi delegoi "laatuasiat" testaustiimille, etenkin jos asiakkaan kokemaa laatua ei ole kehitysorganisaatiolle konkreettisesti ymmärretty. Jos testaukseen tuleva ohjelmisto on keuhkoa, ei hyväkään testaus tule menemään kovin sujuvasti ja myös loppu-tuotteen laatu tulee jäämään huonoksi.**

Sana "kehitys" tässä tarkoittaa sekä kehitysorganisaatiota (suunnittelu ja ohjelmointi) että myös niiden tekemää työtä, johon yleensä kuuluu suunnittelun ja ohjelmoinnin lisäksi jonkinlaista yksikkötestausta ja integrointia.

Sana "testaus" tarkoittaa usein kahta eri asiaa:  
a) Tekemistä eli Laadunvarmistuksen tehtäviä, jotka koskevat eri muodoissa melkein kaikkia kehitysprojektissa; ja  
b) Ihmisiä eli Erillinen testausorganisaatio tai testaustiimi, "testausfunktio".

Tässä artikkelissa puhutaan erillisen testaustiimin (testauksen merkitys b) mahdollisuuksista ja tarpeista puuttua kehityksen tekemään testaukseen (merkityksessä a).

Varsinainen ongelma on, että kehityksen tuotaman järjestelmän laatu on huono. Ja usein vielä sillä tavalla, että projektissa harva sitä tiedostaa ennen kuin erillinen testaus alkaa. Syynä voivat olla esimerkiksi äsken viitattu, vanhentunut tapa delegoida kaikki laadunvarmistus testausammattilaisillekehittäjien puutteellinen testausosaaminen huono näkyvyys kehitteillä olevan tuotteen laatuun projektin sisällä vallitseva hajaannus projektin tavoitteista, tuotteen vaatimuksista ja prioriteeteista.

## MITEN ASIOIDEN "TULISI YLENSÄ MENNÄ"?

Katsotaanpa "miten asioiden yleensä tulisi mennä". Lainausmerkit ovat ihan tarpeen tässä, koska erilaisten käytäntöjen ja prosessien hyödyllisyys riippuu valtavasti projektin ja organisaation tilanteesta. Vaikka alla lueteltu prosessi on mahdollisimman abstrakti, niin helposti prosessin tehtävät tulkitaan jonkin kontekstin mukaan. Esimerkiksi jossain tilanteessa alla luetellut tehtävät tehdään ajallisesti peräkkäin, toisessa tilanteessa rinnakkain. Samoin toisaalla on tarpeen eriyttää rooleja enemmän ja toisaalla vähemmän. Kyse on siis yksinkertaistetusta näkemyksestä ja toistanakin tässä itsestäänselvyksiä, että on myöhemmin mihin viitata.

1. Asiakasta lähellä olevat tahot tuottavat järjestelmän rajauksen asiakasvaatimuksissa
2. Kehitys tuottaa tekniset vaatimukset ja arkkitehtuurin kokonaiskuvan
3. Varsinainen suunnittelu ja toteutus tapahtuvat kehitystiimeissä
4. Valmiit tuotokset julkaistaan integroitavaksi ja testattavaksi
5. Testaustiimi testaa kokonaisuuden

Usealla organisaatiolla on hyviä kokemuksia vaiheiden rinnakkaisuudesta ja roolien sekoittamisesta ja tämä on tärkeä näkökulma ketterissä menetelmissä ja muissa iteroivissa elinkaarimalleissa.

## HALLITTAVUUSONGELMA

Jos halutaan tuottaa riittävän hyvää laatua, tarvitaan siis monta asiaa:

Kommunikoitavissa oleva näkemys tavoitteesta: olen tehnyt koko joukon prosessiarvioiteja ja vaatimukset ovat varmaan yleisin kohtaamani ongelma. Joko niihin ei panosteta riittävästi, luotetaan sokeasti jonkun sidosryhmän tuottamaan dokumenttiin tai sekoitetaan eritasoiset asiat. Lopputuloksena on usein se, ettei projektissa synny yhteisymmärrystä projektin tavoitteista ja niiden

prioriteeteista. Laadulliset tavoitteet jäävät erityisen kehnolle tolalle, koska ei ole luotu yhteistä kieltä niistä puhumiseen ja asiakkaalle riittävä laatutaso on epäselvä.

Osaaminen tavoitteen saavuttamisen keinoista: voidaan asettaa kunnianhimoisia tavoitteita, mutta niiden saavuttaminen edellyttää sekä riittäviä resursseja että myös riittävää osaamista. Paljon hyviä menetelmiä ja työkaluja jää käyttämättä, kun kehityksen energia menee esimerkiksi sovellusalueen ymmärtämiseen. Tässä ei pidä unohtaa myös ei-teknisiä taitoja (kommunikointi, projektinhallinta, ongelmanratkaisu, ...), joiden merkitys korostuu kun projekteja hajautetaan ja liiketoimintaympäristöt monimutkaistuvat.

Kyky havaita todellisuuden ja tavoitteen väli-  
nen ero: tämä on testauksen perinteinen tehtävä. Mutta tämä tarkoittaa myös organisaation kykyä puhua totta ja tehdä päätöksiä tosiasioiden pohjalta. Kovasti arvostamani Jim Highsmith puhuu usein projekti- ja laatutiedosta ja siitä, että tietoa ongelmista voidaan käyttää joko tekosyinä näkyvässä olevalle epäonnistumiselle tai sitten positiivisemmassa tapauksessa varhaisina varoituksina. Vanha projektinhallintasanonta kuuluu näin "huonot uutiset menevät huonommiksi ikääntyessään". Jos järjestelmän laadun systemaattisia ongelmia ei havaita kehityksen aikana, niiden korjaaminen on paljon hankalampaa. Samoin merkitsee valtavasti projektissa vallitseva kulttuuri huonoihin uutisiin: johdon käytös huonoja uutisia kuullessaan määrää sen, saadaanko tulevaisuudessa ongelmat tietoon ennen kuin niistä tulee kriisejä vai aletaanko tietoa pantata ja virheitä peitellä.

Sekä suunnittelijoiden ammattiympäristöä että päälliköiden hyvää hallintoa, joka mahdollistaa järkevän toiminnan: on aivan perusteltu oletus, että projektissa kaikki haluavat tehdä hyvää työtä. Muistan itse miten suunnittelijana minua harmitti, kun jouduin luovuttamaan keskeneräisen sovelluksen pakettiin ja testattavaksi. Asiakas oli

tyytyväinen, mutta itse en saanut tehdä niin hyvää työtä kuin olisin halunnut. Keskeinen projektinhallinnan haaste on moniulotteinen optimointi: sijoittaa resurssit ja mitoittaa kustannukset oikein, että tuetaan liiketoiminnan prioriteetteja. Testauksen tulisi olla koko ajan selvillä projektin tuote- ja liiketoimintariskeistä, että kaikki testaus kohdistuisi sinne missä sillä on eniten merkitystä.

Suuria kokonaisuuksia hallitaan monella tapaa. Yksinkertaisin tapa on tasapäistämisyrimys, jossa koetetaan saattaa eri puolilla tehtävä kehitys ja testaus yhdenvertaiseksi. Hyödylliseksi on koettu vaikkapa laadunvarmistuksen minimi-  
ason asettaminen, eli sovitaan mitä katselmuksia ja testejä kaikki komponentit kokevat niitä muuttaessa. Sitten luonnollisesti riskien mukaan nostetaan kriittisimmän osajoukon testien vaatimustasoa, kattavuutta, jne.

Koodimassojen hallinta tapahtuu pääosin "hajoita-ja-hallitse", eli uustuotannossa iso järjestelmä on jo varhain projektissa jaettu osajärjestelmiin ja ne vielä komponentteihin, sovelluksiin, tms. Usein sopiva modularisoinnin taso löytyy sovellusarkkitehtuurista. Tämä on aivan hyvä tapa, mutta usein tämä jako tehdään etenkin vesiputous-mallisessa prosessissa liian varhain. Kokonaisuus on projektin alkuvaiheessa vielä liian epävarmalla pohjalla, jotta kaikkia osakokonaisuuksia voitaisiin alkaa kehittää itsenäisesti. Toisaalta harvassa projektissa on niin kehittyneet kommunikointimenetelmät, että voitaisiin moneen kertaan muuttaa kokonaisuuden osinjakoa ilman merkittäviä kustannuksia.

## NÄKYVYYSONGELMA TESTAUSTOIMESSA

Kun jouduin vähän aikaa sitten etsimään kokonaiskuvaavaa erillisen testauksen työmäärään ja aikatauluun vaikuttavista tekijöistä, esiin nousi kaksi asiaa: testauksen tavoitteet ja sisään tulevan ohjelmiston laatu. Testauksen tavoitteet tarkoittaa lyhyesti ainakin kahta asiaa: miten hyvää riskienhallintaa (~ miten tiheää seulaa) testauk-



# Ratkaiseva valinta

informaatio- ja tietoteknisiin haasteisiin

**Mermi Business Applications Oy**  
Lars Sonckin kaari 10, 02600 ESPOO  
Puh. 09 540 40 10 · Fax 09 540 40 145  
info@mermit.fi · www.mermit.fi



selta edellytetään ja mitä muita tehtäviä varsinaisen testien suunnittelun ja suorituksen lisäksi testaustiimille asetetaan.

Tämä testattavaksi tulevan ohjelmiston laatu onkin hankalampi käsite. Testauksen sujuvuus riippuu eniten sisään tulevan ohjelmiston kypsyydestä ja eheydestä. Kypsyys tarkoittaa, että ohjelmisto on luotettava ja sisältää sovelluksen tavoitteisiin kuuluvat piirteet. Eheys tarkoittaa, että ohjelmiston toteutus pitää yhtä sisällön ja laadun lupauksen / dokumentaation kanssa.

Tietenkään ei ole realistista odottaa, että kaikki ohjelmisto on korkeaa laatua ja hyvin luotettavaa heti alkuunsa, mutta on testauksen aikeille tuhoisaa, jos järjestelmän komponentit valmistuvat merkittävästi odotettua hitaammin tai oleellisesti huonommassa kunnossa. Esimerkiksi testauskierrosten määrä riippuu paljon siitä, paljonko vikoja järjestelmässä on ja miten paljon jotkut viat peittävät muita vikoja näkymästä. Samoin integroinnin sujuvuus riippuu paljon komponenttien kypsyydestä.

Testaukselle ei ole mikään yllätys että järjestelmissä on vikoja. Mutta testauksella tulee olla näkyvyys järjestelmän laatuun jo kehityksen aikana. Muutoin testauksen panostus suunnittelussa, ympäristöissä ja resursoinnissa voi perustua väärin oletuksiin. Tällöin hukataan projektin loppuvaiheen kallista aikaa sähläämiseen, pikaiseen uudelleensuunnitteluun ja esimerkiksi tehdään osa testauksesta epätarkoituksenmukaisessa ympäristössä. Testien suorittaminen on lähes joka projektissa sen kriittisellä polulla, eli testien myöhästyminen myöhästyttää koko projektia.

## **TESTAUS KEHITYKSEN KONSULTTINA**

Testaukseen kertyy projektien kuluessa paljon tietoa, jota olisi sääli olla käyttämättä organisaation toiminnan kehittämiseksi ja tulevien projektien turvaamiseksi. Vain osa siitä opista ja tiedosta mitä testaajille kertyy, päättyy testaustiimin tai testaustoimen eduksi. Vaatii paljon energiaa, että tieto saadaan kulkemaan ja organisaatio oikeasti oppisi. Tästä löytyy tietoa aiemmin niin suositusta tietämyksenhallinnan (Knowledge Management) ajattelijoilta.

Keskeisin oppi tulee testattavien järjestelmien vahvuuksista ja heikkouksista sekä näiden kautta vastaavasti myös kehitysorganisaation toiminnasta. Paitsi konkreettista vika- ja muuta kirjattua tietoa, testaajat oppivat paljon MuTu-tietoa ("musta tuntuu", ns. hiljainen tacit-tieto), joka harvoin kirjataan minnekään ja harvoin välittyy sinne missä siitä olisi hyötyä.

Toisaalta testaajat oppivat paljon myös asiakkaiden tarpeista ja liiketoiminnan luonteesta, koska

kehityksen alun abstraktit tavoitteet ja toimitusajankohdan realismi kohtaavat juuri testauksessa kaikkein konkreettisimmin. Olisi iso etu, jos tämä kokemus saataisiin tehokkaasti jaettua kehittäjillekin. Hitain tapa opettaa kehitystä on kertoa asia vikaraporttien selityksissä; jokin aloitteellisempi keino olisi paljon halvempi ja tehokkaampi.

Luonnollisesti testauksen tärkein ero muun organisaation osaamiseen on tavallisesti erilaisten laadunvarmistuskäytäntöjen asiantuntijuus. Olen tavannut paljon testaajilla, jotka ovat opiskelleet itseksensä useita kikkoja ja keinoja oman roolinsa ulkopuolelta ollakseen avuksi kehittäjille ja vaatimusten kirjoittajille. Tämä on ilmeisen hyvä idea, koska on todennäköisempää saada viestinsä "tälle tarvitsisi vielä tehdä jotain" läpi, jos on myös antaa ideoita mitä se jotain voisi esimerkiksi olla. Luonnollisesti parasta olisi, että kehittäjillä olisi jo perusopintojensa kautta kaikki tarvitsemansa laadunvarmistustaidot. Testauksen kuitenkin kannattaa panostaa tämänkin puutteen paikkaamiseen, niin kauan kuin näitä tietoja kehitykseltä puuttuu ja kiinnostusta kuitenkin löytyy.

## **NÄKYVYYS JA VAIKUTUSMAHDOLLISUUS KEHITYKSEN LAATUUN**

Testaus on kehityksen lähin (yleensä sisäinen) asiakas, etenkin järjestelmätestauksessa ja järjestelmäintegraatiossa. Koska testauksen tärkeä velvollisuus on tunnistaa järjestelmän puutteita, kehitys asennoituu (toisinaan) testaukseen kuin asiakkaan edustajaan. Luonnollisesti tämä ei ole testauksen oma asema, vaan ainoastaan lainattua valtaa. Jos testaus ei ole palveluorganisaatio ja ole sidosryhmilleen aidosti hyödyksi, ei sitä myöskään arvosteta sisäisen asiakas-roolinsa takia.

Tässä asiakastoimessa testaus tarvitsisi eniten näkyvyyttä tekeillä olevien komponenttien laatuun, kuten jo mainittiin. Tämä auttaa valmistelevaan testausta ja keskittämään testauksen resurssit oikein.

Toinen tarve, ei niin ilmeinen, on ymmärtää kehityksen käytäntöjä. Tämä ei ole kovin suuri ongelma kehityksen ja testauksen ollessa saman organisaation eri puolia, mutta voi mennä todella hankalaksi, jos kehitys tai testaus on ulkoistettu. Ulkoistuksessa ei tietoa kehityksen menetelmistä luontevasti liiku ja asia pitää selvittää erikseen. Tällainen näkyvyys on tarpeen, jotta testaus voi oikein palvella kokonaisuutta, täydentää kehityksen puutteita ja tarvittaessa esimerkiksi auttaa puuttuvan käytännön liikkeelle saamiseen. Ilman aloitteellisuutta ja myöhässä tästä tiedosta on paljon vähemmän hyötyä.