

Systeemityöyhdistys SYTYKE ry:n jäsenlehti N:o 1/2009

A rectangular cake with three distinct layers: chocolate, pink, and white. The cake is decorated with a red ribbon and sits on a silver platter with a spoon.

Tervetuloa Sytyke Ry:n Juhlaseminaariin:

Suomalainen ohjelmistokehitys NYT!

**Sytyke Ry:n 30-vuotisjuhlaseminaari järjestetään Finlandiatalolla
ke 6.5.2009 klo 9.00-17.00.**

Luvassa huippuluentoja alan asiantuntijoilta fokusoidussa paketissa.

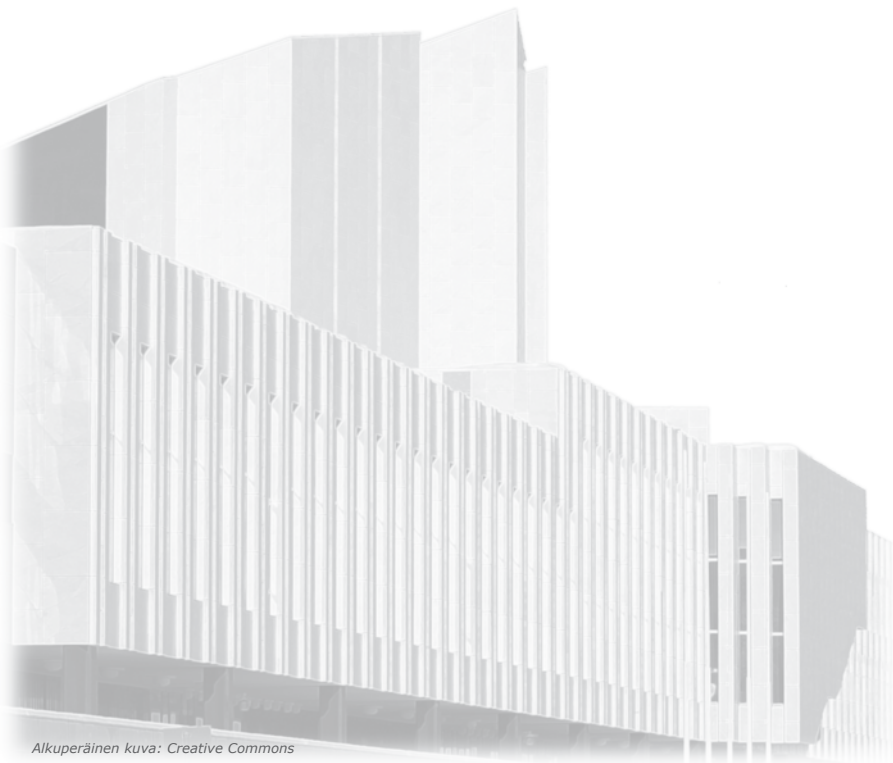
Juhlaluennot ja Tulevaisuuspaneeli sekä kaksi rinnakkaista
osaseminaaria, joissa sisältönä:

- *Systeemyöstä ohjelmistokehitykseen - standardista ketteryyttä
- *Ohjelmistoliiketoiminta ja tuotteistus - kuinka ohjelmistoa tuotteistetaan
- *Arkkitehtuurit/SOA - kokonaisvaltaista ketteryyttä
- *Testaus ja käytettävyys - laadun uudet kasvot

Varaa päivä jo kalenteriisi! Lisätietoja maaliskuun aikana.

Lisätietoja: Timo Sundman, +358 50 32 400 38, timo.sundman@tieto.com

Oikeudet muutoksiin pidätetään.



Alkuperäinen kuva: Creative Commons

Muistathan myös
kevätseminaarin ja -kokouksen 25.3.09

Paikka: Tieto, Kutojantie 6-8, Espoo

Aika: 25.3.09 Kahvitarjoilu 14:30 Ohjelma alkaa 15:00. Kokous klo 17:00.

Ohjelma tarkemmin ja ilmoittautuminen webissä (www.sytyke.org).

Tervetuloa!

Hallitus



Julkaisija

Systeemityöyhdistys Sytyke ry
 Susanna Koskinen
 Talvikkitie 40 A 33, 01300 Vantaa
 p. 09 5607 5363
 f. 09 5607 5365
 sytyke@hennax.fi

Päätoimittaja

Minna Oksanen
 minna.oksanen@gmail.com
 puh. 040 577 6640

Toimitussihteeri

Susanna Koskinen
 sytyke@hennax.fi

Taitto

Speaking Bark Tmi
 Katja Tamminen
 taitto.kt@speakingbark.com

Toimituskunta 1/2009

Matti Vuori
 Päivi Brunou
 Helena Venäläinen
 TestausOSY
 KäytettävyydenOSY

Lisätietoja lehdestä

www.sytyke.org/lehti

Tilaukset

Systeemityölehti sisältyy yhdistyksen
 Tietotekniikan liiton suositusten
 mukaiseen yhdistyksen jäsenmaksuun.
 Vuositilaus 30 €
 Irtonumerot 8 €
 Hyvissä ajoin ennen painatusta tehty
 vähintään 50 kpl lisätilaus 2 €/kpl.
 Tilaukset yhdistyksen toimistosta.

Kansikuva

Matti Vuori

Seuraava numero

Pohjatöitä
 Toimituskunta:
 Minna Oksanen
 Matti Matikainen
 Tarja Raussi
 Erkki Pöyhönen
 Matti Vuori
 Ilmestyy: to 30.4.2009

Painopaikka

T-Print
 Ahokaari 1-3
 05460 Hyvinkää
 Puh. (019) 475 8500

Painos: 2500 kpl

ISSN 1237-0525

16 vuosikerta, nro. 1

Ilmoitushinnat

Takakansi A4 1200 €
 Sisäkannet A4 1000 €
 Sisäsivut 1/1 800 €
 Sisäsivut 1/2 600 €
 Sisäsivut 1/4 400 €

Arvonlisävero 0%

Vakiopaikan vähintään vuodeksi
 varanneille 20% alennus.

Teksti: Matti Vuori, Cybercom Finland
 Päivi Brunou, Qentinel

Pääkirjoitus:

Matkalla kokonaisvaltaiseen laadunvarmistukseen

Laadunvarmistuksesta on erilaisia näkemyksiä ja kulttuureja. Tässä numerossa tuodaan esille joitakin mielenkiintoisia näkökulmia. Yleisesti laatu huomataan silloin, kun siinä on puutteita; projektit myöhästyvät, tai kun uusi äänestysjärjestelmä hukkaa kansalaisten ääniä. Laadunvarmistuksessa ollaan onnistuttu silloin, kun negatiivisia havaintoja ei tehdä. Työtä tehdään siis sen eteen, että työn tulokset jäävät huomaamatta.

Paremmat systeemit + parempi toiminta + paremmat tuotokset + paremmat tulokset = parempi elämän laatu. Kun päivämme hektisessä maailmassa on usein kiireellä kuormitettu, toimivat ja hyvin suunnitellut sekä taidolla toteutetut järjestelmät tekevät arjesta pikkuisen paremman.

Esimerkiksi juuri käytettävyyden, usability-merkityksessä, on teema, joka harvoin nousee esille laadunvarmistuksessa puhuttaessa. Kuitenkin monissa järjestelmissä käytettävyyden on keskeinen laadutekijä. Voiko järjestelmä olla laadukas, jos se ei ole käytettävyydeltään sopiva? Perinteisempää laadunvarmistuksen aluetta, testausta, voi myöskin käsitellä eri sovellusalueiden ja haasteiden kannalta vielä paljon lisää.

Asiantuntijoina jutuissa on pääosin TestausOSY:n ja KäytettävyydenOSY:n jäseniltä. Artikkeleita on myös muilta laadunvarmistuksen asiantuntijoilta.

Toivottavasti löydät tästä numerosta jotakin uutta kokonaisvaltaisen laadun kehittämisessä työssäsi.

Hyviä lukuhetkiä, toivoo toimituskunta.

Sisällys

Pääkirjoitus

Matti Vuori, Päivi Brunou

3

Laadunvarmistus kehittyä

Matti Vuori

4

Käytettävyyttä tutkijoille

Markku Niemi

7

Mallien varmistamaa laatua – mallikelpoista laatua?

Risto Nevalainen

8

Testaus hajautettuna suunnitelmavetoisessa ja ketterässä mallissa

Kari Kakkonen

12

Ohjelmistostandardeja laidasta laitaan

Teemu Vesala

15

SOA – testattuna

Jonas Hinkkanen

17

Vuoden testaaaja 2009 on valittu

TestausOSY

19

Käytettävyyden osana prosesseja

Eeva Kangas, Juha Jääskinen

20

Loppukäyttäjät testaajana – kokemuksia keskeneräisistä tuotteista

Jari Mitro

22

Sama ongelma – erilaiset tulokset. Miten varmistaa käytettävyydestä laatu?

Timo Jokela

24

You need to be smart!

Ivar Jakobsson

27

Navigaattori kourassa rotvalliin reunalla – maailman käytettävyydenpäivä Tampereella

Eija Kaasinen

28

Kirja-arvostelu: Helppokäyttöisen verkkopalvelun suunnittelu

Helena Venäläinen

30

Kuutamolla

34



Kirjoittaja toimii konsultointipalvelujen tuotepäällikkönä Cybercom Finland Oy:ssä (yritys toimi vielä vuodenvaihteessa Plenware) ja työssään konsultoi ja kouluttaa organisaatioita mm. laadunvarmistukseen, testaukseen, käytettävyyteen ja riskienhallintaan liittyvissä asioissa.

Vuori on vuonna 2009 Käytettävyyden OSY:n hallituksen puheenjohtaja ja Testaus OSY:n ohjausryhmän jäsen ja osallistuu globaalin testaussertifikaatin ISTQB kehittämiseen Suomen organisaation FISTB:n hallituksessa.

Laadunvarmistus kehittyy

Tämän lehden teemana on laadunvarmistus. Se on alue, joka muuttuu jatkuvasti, sillä käsityksemme laadusta ja sen varmistamisen tavoista kehittyvät, kun koemme ja opimme uusia asioita. Artikkelini luo tiiviin katsauksen laadunvarmistuksen ajankohtaisiin ilmiöihin.

Käsitys laadusta on oppimisprosessi

Käsitys siitä, mikä on "laatu" on aina sidoksissa ymmärrykseemme järjestelmistä. Mitä enemmän osaamme jäsentää asioita, sitä rikkaampi on käsityksemme laadun keskeisistä elementeistä. Ajatteluun vaikuttaa aina se, mitkä kulloinkin ovat painopistealueet, eli "siitä puhe, mistä puute". Laatuajattelun alkuaikoina oli tekniikan saaminen kuntoon keskeistä ja siksi tekninen laatu oli vallitseva paradigma. Kun tekniikka alkoi olla paremmin hallittua, voitiin näkökulmaa nostaa asiakastarpeisiin. Eli miettiä avoimemmin silmin, mitä asiakkaat oikeasti tarvitsevat ja haluavat.

Samalla koko järjestelmien kehittämisosaaminen kasvoi vahvasti. Esimerkiksi vaatimusmäärittelykulttuuri tuotti ymmärrystä siitä, että järjestelmien oleellisia piirteitä ovat luotettavan oikean toiminnallisuuden lisäksi vaikkapa tietoturvallisuus ja käytettävyys.

Kuitenkin, osakulttuurit eivät aina kohtaa ja erilaiset laatuominaisuudet jäävät helposti laadunvarmistuksen ytimestä erillisiksi saarekkeiksi. Testaus on keskeinen laadunvarmistuksen tekniikka, mutta sen luonne on usein käytännössä jäänyt toiminnallisuustestaukseen ja muiden ominaisuuksien testauksen tarvetta ei edes suunnitella systemaattisesti, vaan niitä otetaan projektien työlle vasta sitten, kun kantapään kautta huomataan, että jotain pitäisi tehdä. Vuonna 2009 pitäisi olla täysin selvää, että käytettävyys ja tietoturvallisuus ovat jokaisen projektin teemoja ja että kuormitustestaus on välttämätön järjestelmäprojektin testaustehtävä.

Tässä on kyse arkisen toiminnan inertiasta, jonka murtaminen on vaikeaa.

Käyttäjakeskeinen laatu

Asiakaslähtöinen laatu ei ole mikään uutinen. Uudempaa on systemaattinen laadun varmistami-

nen eri käyttäjäryhmien näkökulmasta. Käytettävyys on keskeinen laatutekijä ja sen jonkinasteinen varmistaminen on mukana monissa projekteissa - mutta ei läheskään siinä määrin kuin 1990-luvulla arvioitiin. Käyttäjakeskeisen suunnittelun osaaminen ja soveltaminen on lisääntynyt, mutta silloinkin puuttuu laadunvarmistuksen perusperiaatteiden ohjelmistokehityksen kaltaisissa asioissa: osaaminen ja hyvätkään kehittävät prosessit eivät riitä, lopputuloksen laatu on aina varmistettava.

Kuitenkin, niillä alueilla, joilla peruskäytettävyys on saatu kohtuulliseen kuntoon, on näkökulmia jälleen voitu laajentaa. Käyttäjäkokenus on nousut käytettävyyttä ylemmän tason käsitteeksi ja kattaa käyttäjän kokemaa laatua laajemmin kuin ergonomiapohjainen käytettävyys. Ennen kaikkea on kyse eheästä käyttäjän ja tuotteen suhteesta, tunnesiteistä, kokonaisvaltaisesta kokemuksesta.

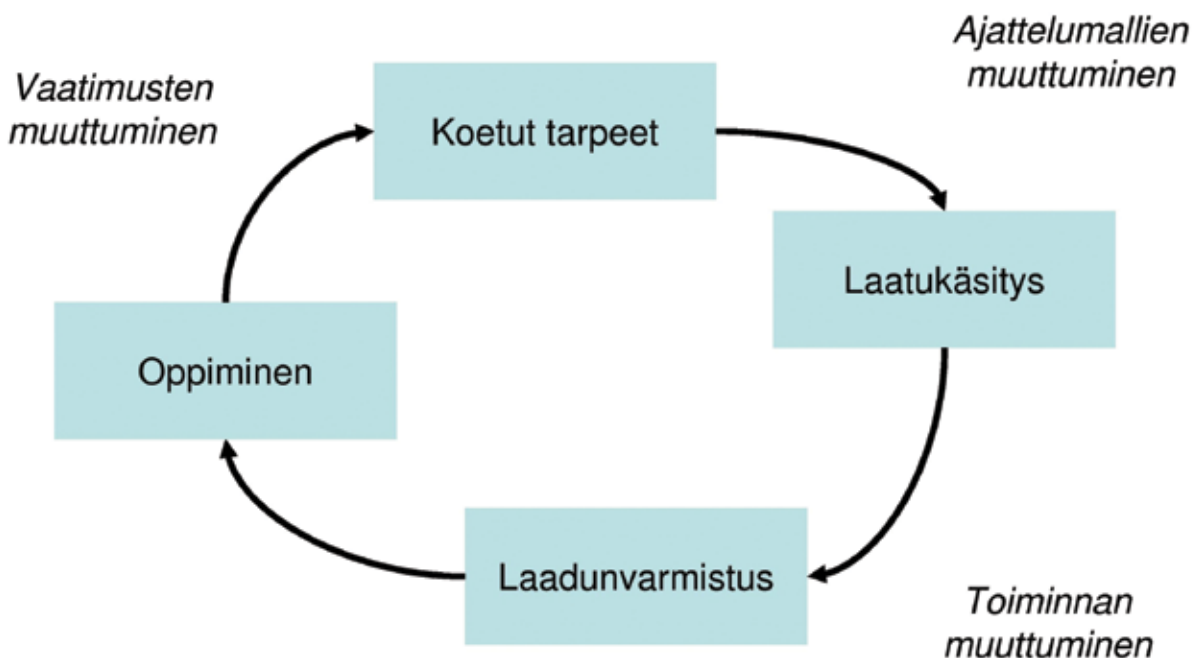
Arvopohjainen suunnittelu on edelleen tälle jatkoa. Olennaista on se, että siinä missä esimerkiksi kulttuurisia tekijöitä on saatettu pitää suunnittelun reunaehtoina, nyt on oivallettu, että tuotteeseen liittyvä kulttuuriset merkitykset ovat osa sen tuottamaa arvoa asiakkaalle ja käyttäjälle. Taiteessa tämä on tietysti ymmärretty aina.

Laadun kohteen abstraktiotason nousu

Kaikkeen laatuajattelun kehittämiseen liittyy laadun kohteen abstraktiotason nousu. Jos tekninen laatu liittyi aiemmin teknisiin yksityiskohtiin, uudemmat laatuajattelut keskittyvät enemmän konsepteihin: tuotteeseen tai järjestelmään liittyviin perusperiaatteisiin, uniikkeihin korkean abstraktiotason piirteisiin, ja kaikki nämä käytön kontekstissa, jossa tuote ei ole yksin, vaan käyttäjän maailmassa.

Vain toimivan kokonaisuuden laadulla on merkitystä

Jos järjestelmä koostuu useasta osajärjestelmästä, millainen on laatu, jos osajärjestelmät toimivat, mutta kokonaisuus ei? Laatu on silloin tietysti nolla. Järjestelmien integrointi tuottaakin projekteissa suurimmat ongelmat ja niinpä myös operatiivinen laadunvarmistus on nostanut näkökulmaansa järjestelmäintegrointiin. Tämä näkyy



käytännössä siinä, että järjestelmäintegroitite-
tauksen tekeminen on systematisoitunut ja sen
osuus projekteissa on kasvamassa.

Toiminnan laadun varmistuksen uudet alueet

Jo varhain on oivallettu, että lopputuloksen
laatu riippuu prosessista, jolla se saadaan aikai-
seksi. Tuotteiden laatuun tarvitaan siis toiminnan
laatua, prosessien sekä teknistä laatua että tar-
koituksenmukaisuutta kulloiseenkin tarkoituk-
seen. Esimerkiksi ISO 9000-sarjassa on kysymys
tiettyyn paradigmaan perustuvasta johtamisen ja
operatiivisen toiminnan laadusta. Ohjelmistoke-
hityksen kypsyysmallit (esim. CMMI ja erilaiset
testauksen kypsyysmallit) ovat luoneet käsitystä
ammattimaiselta toiminnalta edellytettäviltä
menettelyistä. Näitä malleja voidaan käyttää
"todistamaan" toiminnan laatua (tietenkin vah-
voilla varauksilla). Viime aikoina esimerkiksi ITIL
ja ohjelmistohankintojen CMMI (CMMI for Acqui-
sition) tuovat vastaavaa globaalia ajattelua uusiin
organisaatioiden prosesseihin.

Tällaista laajenemista voidaan odottaa edel-
leen. Systeemytyössä ajatellaan usein teknistä
järjestelmätyötä ja projektitoimintaa. Yhtä oleel-
listä on muistaa vaikkapa johtamisen merkitys.
Missä yrityksessä käytetään arkisen johtamisen
kypsyysmallia? Esimerkiksi projektiyrityksen kyp-
syysmalleja sen sijaan on saatavilla ja käytetty
ainakin benchmarking-tarkoituksessa.

Standardien hyödyntämistä haittaa tunnetusti
niiden heikko tuntemus. Esimerkiksi ISO 90003,
joka kertoo hienoja konkreettisia ohjeita ISO
9001:n soveltamiseen ohjelmistokehityksessä, on
useimmille täysin tuntematon. Jo se, että yhden
standardidokumentin henkilökohtainen (!) lisenssi

maksaa kymmeniä euroja auttaa varmistamaan,
että standardeja ei saada organisaatioissa kaik-
kien luettavaksi. Standardointijärjestelmän
laadussa on tässä valtava epäkohta. Onneksi
standardointi on vähitellen uusien avoimempien
teollisuus- ja muiden organisaatioiden ja WWW-
kulttuurin myötä tulossa vapaammin hyödynnet-
täväksi.

Ketterä laadunvarmistus

Hyvä laadunvarmistus on luonteeltaan ketterää.
Uutta asiaa kehitettäessä mietitään kohteen läh-
tökohdista mikä sille on oleellista, millaiset omi-
naisuudet vaativat panostusta kehittämiseen ja
varmistamiseen. Tämän pohjalta tehdään räätä-
löityjä suunnitelmia ja laadunvarmistus tehdään
niillä tavoilla, jotka kulloinkin ovat parhaat mah-
dolliset.

Tämä on periaate, joka sopii kaikenlaisiin pro-
jekteihin. Viime aikoina on yleistynyt ketterä
ohjelmistokehitys erityisesti käyttäen Scrum-pro-
jektinhallintamallia. Scrum ei sinänsä ota kantaa
ohjelmistokehitys- tai laadunvarmistuskäytän-
töihin, mutta yleisillä tavoilla soveltaa Scrumia
on vahvaa potentiaalia laadunvarmistusmielessä.
Keskeisinä:

- Robustin koodin tuottaminen panosta-
malla yksikkötestaukseen
- Jatkuva matalan tason integrointi, jolla
pidetään koodikanta virheettömänä
- Vaiheittainen kehittäminen edistää vaati-
musten laatua ja koko konseptin laatua

Kuitenkin, Scrum-sovelluksissa on myös puut-
teita. Rikas testaustoiminta edellyttää useita
abstraktiotasoa ja sprintit ylittäviä näkökulmia
laatuun. Järjestelmätestauksen näkökulma on
Scrum-sovelluksissa usein heikko ja automatisoi-

*Laadunvarmistuksen
kehittyminen on
kulttuurinen
oppimisprosessi.*

tuja hyväksymistestejä voidaan pitää lähinnä vitseinä – aito asiakkaan tekemä hyväksymistestaus on monimuotoinen ihmisten tekemä prosessi, joka edellyttää järjestelmähankkeissa hyvin suuria tilaajan panostuksia.

Samoin esimerkiksi käytettävyyden varmistaminen on heikkoa – sujuva vuorovaikutus asiakkaan kanssa ei varmista käytettävyyttä.

Laadunvarmistus lama-aikana

Tällä hetkellä eletään globaalia lamaa ja äärimmäisen niukkuuden aikaa. Voiko olla niin, että toiminnan ja tuotteiden laadusta tingitään, koska ei ole varaa tehdä hyvää? Lopetetaanko esimerkiksi testaus, koska siihen ei ole rahaa?

Jos ajatellaan asioita systeemityön näkökulmasta ja muistetaan edelleen akuutti ohjelmistokriisi, on selvää, että laadun heikentämisen varaa ei ole. Toiminnan laadun on oltava hyvää heikoinakin aikoina. Lamassa onkin se hyvä puoli, että se pakottaa miettimään, mikä järjestelmissä ja palveluissa on olennaista. Tämä on kriittinen lähtökohta laadulle ja sen varmistamiselle. Suurimmat laatuongelmathan kumpuavat juuri konseptitason kysymyksistä – sen ymmärtämisestä, mitä ollaan kulloinkin tekemässä ja mikä sen onnistumisessa on oleellista. Toisaalta, lama-aikana ei ole kertakaikkiaan varaa hankkeiden epäonnistumiseen. Laatuun on siksi kiinnitettävä erityistä huomiota.

Järjestelmähankinnoissa osataan myös etsiä uusia kustannustehokkaita järjestelmätyyppejä, ja avoimen lähdekoodin suosion arvellaan kasvavan merkittävästi. Se on uusi tapa ajatella ohjelmistoista, niiden toimituksesta ja määrittelystä ja

tarvitsee myös uudenlaista systematiikkaa. Uusi JHS-suositukseksi julkaistu Avoimen lähdekoodin hankintaopas tuo toiminnan laatua tälle saralle. Avoimet rajapinnat yleistyvät toivottavasti merkittävästi, sillä niiden vaikutus integrointihankkeiden onnistumiseen on aivan ratkaiseva. Hienoa on myös se, että avoimella lähdekoodin sovelluksilla yhä useammat organisaatiot pystyvät hankkimaan koko henkilöstölleen esimerkiksi laadukkaita testaustyökaluja yksikkötestauksesta vianhallintaan ja viime aikoina myös testauksenhallintaan, missä välineiden kalleus on rajoittanut niiden hankintaa ja käyttömahdollisuuksia.

Niukkuus korostaa laatutoiminnan luonnetta viestintänä. Jos joudutaankin tekemään kompromisseja, kaikkien asianosaisten on tiedettävä, mitä ne kompromissit ovat ja millaisia vaikutuksia niillä on. Silloin ne voidaan hyväksyä tietopohjaisesti.

Kohti 2010-luvun kokonaislaatua

Laadun ja sen varmistamisen parissa tapahtuu koko ajan paljon ja organisaatioissa onkin otettava asiat säännöllisin väliajoin pöydälle ja arvioitava toiminnan laatua: onko käsitys laadusta kattava, tuotetaanko ja varmistetaanko laatu asianmukaisesti kaikissa relevanteissa prosesseissa, muistetaanko perusasiat kun toimintaa ja prosesseja muutetaan.

Siinä, missä toiminnan laatu on edennyt työprosessien detaljeista kokonaisprosesseihin ja lähestymistapoihin, myös järjestelmien laadussa on enemmän nostettava abstraktiotasoa ja mietittävä laatutekijöitä konseptien tasolla. Konseptiajattelun on ajateltu olevan Suomessa heikkoa. Sen kehittämisessä ei ole kyse hypen lisäämisestä, vaan kyseessä on avainasia asiakastarpeiden täyttymisen eli laadun näkökulmasta.

Kuva 2. Laadunvarmistus on parhaimmillaan ketterää ja virtuoosimaisen osaavaa.



Käytettävyyttä tutkijoille

Suomen Akatemia myöntää vuosittain yli 250 milj. euroa tieteelliseen tutkimukseen ja muuhun tieteen edistämiseen. Vuosittain saapuvat lähes 5000 tutkimusrahoitushakemusta ovat käytännössä 100%:sti sähköisessä muodossa. Sama pätee kustakin hakemuksesta lausunnon tuottavaan tieteelliseen vertaisarviointiin, Akatemian toiminnan laadun ja luotettavuuden peruskiveen. Käynnissä olevien järjestelmämuudistusten tuloksena sähköistyvät tutkimusrahoitusprosessirutiinien loputkin osat kuten päätösten allekirjoittaminen ja muutosten hakemuskäsittelyt. Käytettävyyttä on paranneltu vähin erin järjestelmäkehityksen vaiheissa ja tilanne on nyt hyvä ja asiakkaat tyytyväisiä.

Käytettävyyden tilanne vuonna 2003

Vuonna 2003 teetettiin Suomen Akatemian sekä eräiden muiden koti- ja ulkomaisten rahoittajaorganisaatioiden asiointiratkaisuja vertaileva asian- tunti- arviointi. Arvioinnin toteutti Adage Oy. Akatemia sijoittui vertailussa toiseksi. Taulussa 1 on lueteltu korkean sijoituksen taustalla olevia piirteitä. Vuoden 2003 selvityksessä tunnistettiin myös joukko käytettävyyden parantamista vaativia asioita. Listaus niistä on taulussa 2.

Akatemian vastuuhenkilöt ja ohjelmistotoimittajat ovat ottaneet hyvin vaarin havainnoista ja lähes kaikki puutteet on jo korjattu myöhemmissä versioissa. Jotakin on sentään jäänyt tulevankin järjestelmän korjattavaksi - esimerkiksi se, että hakemuksen voisi tulevaisuudessa täyttää omilla tunnuksillaan joku toinen tutkijan puolesta.

Järjestelmämuudistukset tuloillaan

Akatemia korvaa seuraavien kahden vuoden aikana nykyisen kahden erillisen sovelluksen ja järjestelmätoimittajan asiointi- ja rahoitustietojärjestelmäkokonaisuuden yhdellä. Tavoitteena on ketteröidä ylläpitoa ja jatkokehitystyötä ja

minimoida hankalasta ja työläästä järjestelmäintegroinnista johtuvat ohjelmistovirheet. Elinkaarensa päätä tekniseltä toteutukseltaan lähestyvä pitkäikäinen ohjelmisto on joskus kuin alokkaan asento: sitä ei voi korjata vaan se pitää tehdä uudestaan.

Haasteena on edelleenkin suunnitella ja toteuttaa käytettävyys hyvin erilaisten käyttäjäryhmien näkökulmasta. Tutkimusrahoituspäätöksiä päivittäistyönään valmisteleva ja toimeenpaneva tiedeasiantuntija tarvitsee runsaasti enemmän toiminnallisuutta ja tietoa kuin rahoitusta hakeva tutkija. Työn helpottamista haetaan myös asiointijärjestelmän, dokumenttienhallinnan ja toimistojärjestelmän tosiaikaisesti integroivilla ratkaisuilla; asiakirjat talteen ja tarvittaessa esille, tekstinkäsittelyyn ja takaisin järjestelmän haltuun ilman käsin nysväämistä, niitä kuuluisia "nappeja" painelemalla. Käynnissä olevassa kilpailutusprosessissa on mielenkiintoista nähdä, millaista käytettävyyssajattelua meille on tarjolla uuden järjestelmäsukupolven tekemiseen niin teknisessä kuin mentaalissakin mielessä.

Viime syksyinen Akatemian yleisten verkkopalveluiden käyttäjätutkimus totesi merkittävää parannusta tapahtuneen monelta osin. Kehitys näköjään kehittyä tässäkin eli parannettavaakin riittää edelleen. Akatemian yleisiin verkkosivuihin <http://www.aka.fi> ja sähköiseen asiointiin (<http://www.aka.fi/fi/A/Tutkijalle/Kirjaudu-tasta-sahkoiseen-asiointiin/>) voi kuka tahansa tutustua vapaasti vaikka käytettävyyden arviointimielessä. Vinkki: valitse sähköisessä asiointissa Testihaku niin voit leikkiä vapaasti hakulomakkeistolla.



Markku Niemi on projektipäällikkö Suomen Akatemiassa vastuullaan rahoitus- ja asiointitietojärjestelmien uusiminen.

Taulu1.

Akatemian sivuston hyviä piirteitä vuonna 2003 tehdyssä arvioinnissa

- mahdollisuus laatia hakemus vaiheistetusti - "...Sivusto toimii erittäin hyvänä wizard-tyypisenä täytön jaksottajana ja apuna", kuten ulkopuolinen käytettävyyssasiantuntija raportoi.
- automaattinen välitallennus
- koko ajan saatavilla oleva lomakekohtainen ohje
- kenttäkohtainen ohjeistus
- lomakkeen navigaation sitomisen hakemusprosessiin
- pyrkimys käyttää paperilomakkeen numerointia ja otsikointia navigaatiossa
- yhteenveto käyttäjän hakemuksesta, lomakkeista ja tiedoista
- miellyttävä visuaalinen kokonaisuus, rauhallinen layout lomakkeissa ja sivustolla

Taulu 2.

Akatemian sivuston tunnistettuja kehittämistarpeita vuoden 2003 arvioinnissa

- Puutteellisuudet ja epäyhdenmukaisuudet navigaation ja hakemusprosessin kesken
- Keskenäisten kohtien korostamisen puute
- Epäyhdenäisyys pakollisten kenttien merkitsemisessä
- Selkeän käyttäjää tukevan hakemuskohtaisen yhteenvetosivun puute
- Lopullisen hakemuksen toimittaminen vain paperimuodossa
- Kielen valinnan sijainti ja kielivalinnat eivät säily eri verkkosivustojen välillä siirryttäessä
- Uudet ikkunat
- Eteneminen ei ole mahdollista mikäli ei täytä kaikkia kohtia oletetussa järjestyksessä
- Hakemusta ei voi täyttää ja jättää toisen puolesta
- Kenttiin syötetty tieto saattaa hukkaa virheilmoitustilanteessa
- Hakemus poistettava jos jätetty väärään hakuun (hakuvalintaa ei voi vaihtaa)
- Hakijan yhteys- ja organisaatietietojen jälleenkäytön puute



Mallien varmistamaa laatua – mallikelpoista laatua?

Risto Nevalainen,
Senior Advisor, FISMA ry
Risto Nevalainen on toiminut
pitkään laadunhallinnan
asiantuntijatehtävissä, mm.
ISO9001 pätevänä arvioijana
vuodesta 1990 lähtien.

Hän on tehnyt lukuisan määrän
SPICE- ja CMMI-arviointoja
Suomessa ja ulkomailla. Viime
vuosina hän on perehtynyt laa-
dunvarmistukseen vaativissa
turvallisuskriittisissä järjestel-
missä, etenkin ydinvoimaloiden
I&C sovelluksissa. Nevalaisen
vastuualueena FISMA ry:ssä on
ohjelmistojen ja järjestelmien
standardointi sekä prosessien
kehittämisen menetelmät ja
mallit.

**Tässä artikkelissa kuvataan laadunvarmis-
tuksessa käytettyjä ja/tai sitä varten suun-
niteltuja malleja ja standardeja. Ne on tehty
yleensä joko kansainvälisissä standardoin-
tielimissä (ISO, IEC, IEEE) taikka yritysten
muodostamissa konsortioissa. Mallit voi-
daan luokitella monella tapaa, esimerkiksi
niiden pääasiallisen toimialan, varmistuksen
kohteen ja niiden sisältämien menetelmien
mukaan. Uutena asiana artikkelissa esitel-
lään ns. varmistustapaus (assurance case),
jonka avulla voidaan varmistaa laajoja koko-
naisuuksia etenkin vaativissa ohjelmistoissa
ja järjestelmissä.**

minnan ja prosessien yhdenmukaisuutta. Yhteistä
on ainakin osittainen riippumattomuus tekijän ja
varmistajan välillä. Yhdenmukaisuutta korosta-
vat mallit sisältävät usein myös ns. referenssin eli
laatua varmistetaan suhteessa johonkin.

Oheiseen taulukkoon (taulukko 1) on koottu
tekijöitä, joissa erityyppiset laadunvarmistuksen
mallit poikkeavat toisistaan. Toisena ulottuvuu-
tena voidaan nähdä yhdenmukaisuus vs nyky-
tilanteen vahvuus vs kehityshakuisuus. Näiden
mukaisesti taulukkoon on sitten koottu joitakin
rivejä korostamaan kunkin näkökulman keskeisiä
painotuksia ja eroja.

1. Käsitteiden selvennystä

Laadunvarmistuksesta on monia määritelmiä.
Joissakin painotetaan testausta ja lopputuloksen
laatua, joissakin toisissa määritelmissä taasen toi-

Taulukon viimeisenä rivinä on joitakin esimerk-
kejä erityyppisistä laadunvarmistuksen malleista.
Useimmille lukijoille tunnettuja lienevät ainakin
ISO9001, CMMI ja EFQM. Kerron tässä artikkelissa

Taulukko 1.
Laadunvarmistuksen erilaisia
painotuksia ja tyypittelyä.

	Conformance, Compliance	Capability and Maturity	Improvement
Type of requirement in reference(s)	"Shall" type normative requirements, focus on implementation	Target capability or maturity profile/level, better process	Improvement findings, consultative guidance, focus on management
Assurance approach and results	Satisfaction of requirements, Yes/No scale, gap reporting, audit report	Continuous, multi-level scale, NPLF rating, process profiles	Management and team consensus building, prioritisation of findings
Focus areas of assurance	Process, product, competences	Capable and predictable process, risk avoidance	Business goals, better performance
Business case for external assurance	Certification, market requests	Differentiation, customer needs	Improvement, business benefits
Exemplar models	ISO9001, IEC61508, ITIL and ISO20000, Agile	ISO15504, CMMI, various domain models (nuclear, medical, automotive...)	EFQM, Lean Six Sigma, ISO38500 (IT Governance)

näistä lisää, mutta myös hieman uudemmissa ja ehkä vähemmän tunnetuista malleista.

Ohjelmistoteollisuuden ja järjestelmäkehityksen soisi kehittyvän niin, että sen tulokset ovat hyvinkin laadukkaita ja ”kestäviä” (dependable) eivätkä aiheuta riskejä yhteiskunnalle, yrityksille eikä yksilöille. Samaan aikaan kehitys on kohti suurempia, digitaalisia kokonaisuuksia. Esimerkkinä vaikkapa uusi automalli, jonka kehitystyöstä reilusti yli puolet on jo ohjelmistotyötä. Muutaman vuoden päästä meillä on myös valtakunnallinen terveystietojamme sisältävä rekisteri, jonka toivoisi olevan sekä tehokkuutta lisäävä että laadukas. Niinpä tarvitaan nykyistä isompia panostuksia ohjelmistojen laadunvarmistukseen, osin viranomaisvaatimuksista johtuen. Malleilla ja standardeilla on tässä merkitystä, koska ne ovat kaikille osapuolille yhteisiä ja yleensä laajaan konsensukseen perustuvia.

2. Prosessi- ja tuotevarmistuksen mallit

Tunnetuimpia prosessimalleja lienevät CMMI ja SPICE (ISO/IEC15504). Malleilla arvioidaan toiminnan nykytilaa ja haetaan kehityskohteita. Usein malleja käytetään myös toimittajavalinnassa. Nämä mallit edustavat taulukon 1 keskeimmistä saraketta eli niillä mitataan prosessien kyvykkyyttä ja organisaation kypsyyttä. Nämä mallit ovat olleet käytössä jo parikymmentä vuotta, mutta ne eivät ole onneksi jämähtäneet vaan uudistuvat edelleen. CMMI-mallin nykyversio on 1.2, ja vuoden 2010 lopulla voidaan odottaa uutta versiota, CMMI 2.0. Se sisältää verraten joustavan arkkitehtuurin ja joukon käyttökohteen mukaan varioituneita prosessijoukkoja (ainakin ohjelmistot, järjestelmät ja palvelut). Ostajille ja kehittäjille on omat mallinsa. SPICE kehittyy samaan tapaan ja suuntaan kuin CMMI. Nykyversio on julkaistu standardina vuosina 2004 – 2006. Näinä päivinä julkaistaan standardi organisaation kypsyyden arviointiin. Työn alla on lisäksi mm. palvelujen hallinnan arviointimalli. SPICE-standardi on myös kokonaisuudistuksen kohteena lähivuosina.

Ei ISO9001 mallikaan ole unohtunut. Siitä tuli viime marraskuussa uusi versio, ISO9001:2008. Sisältö muuttui monessa kohtaa, mutta yhtään uutta vaatimusta tai tulkinnan merkittävää kiristystä ei tehty verrattuna vuonna 2000 julkaistuun aikaisempaan versioon. Painetta isompaan uudistukseen on epäilemättä olemassa, ja purkautuu seuraavassa versiossa vuonna 2014 tai 2015. Rinnalle on kehittymässä myös tarkempia malleja, mm. projektinhallinnan ISO-standardit ja IT Governance – aihepiirin standardit.

Eri malleilla on verraten paljon päällekkäisyyksiä. ISO9001 on kaikkein pelkistetyin ja soveltuu kaikenlaaisiin tilanteisiin. Vastineeksi se vaatii

kokeneen ja taustateoriat tuntevan arvioijan ja soveltajan, muuten tulokset vaihtelevat liikaa ja mallin käyttö mekanisoiutuu. CMMI ja SPICE ovat tässä suhteessa tarkempia ja parempia, mutta eivät sovellu yhtä notkeasti erilaisiin liiketoiminnan tarpeisiin. Ne ovat myös verraten raskaita, ainakin jos ne otetaan kokonaan käyttöön.

Aika vakiintunut sisältöjako IT-lähtöisissä yleismalleissa on SSS eli Software, Systems ja Services. Tämä kolmen ässän jako on jo toteutettu CMMI:ssä ja ja SPICE:ssa. Minkään yrityksen ei kannata yrittää ottaa kaikkia malleja täysimääräisesti käyttöön, vaan valita ensisijaisin liiketoiminnan ja ansaintalogiikan mukaan. Trendi tuntuisi olevan kohti Service-lähtöisiä malleja, josta osoituksena on ITIL-mallin suosio.

Tuotelaadun puolella ei ole yhtä selkeää ja vakiintunutta tilannetta kuin prosessien puolella. Ohjelmiston laadun arviointiin on kehitteillä ISO25000 standardisarja, perustuen jo julkaistuun ISO9126 sarjaan. Se painottuu laadun mittaamiseen eri menetelmin. Tietoturvan tarkastamiseen on julkistettu oma ISO27000-standardisarjansa.

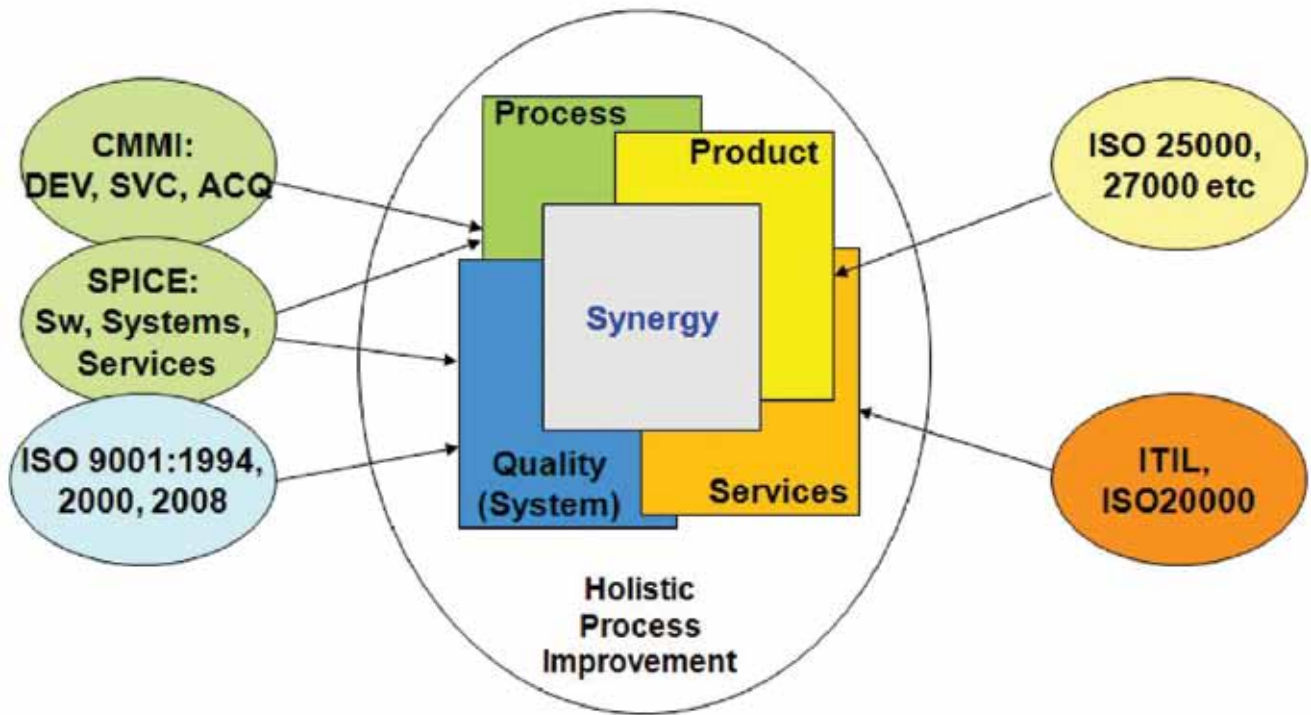
Kuvassa 1 on havainnollistettu eri mallien välisiä eroja ja päällekkäisyyksiä. Nyrkkisääntö voisi olla että millä tahansa yhdellä mallilla saa jo puolet siitä, mitä kuvassa esitetyt viisi erityyppistä mallia kattavat. Kukin yritys voi sitten rikastaa mallien käyttöä oman tarpeensa mukaan. Tyypillinen polku suomalaisissa IT-yrityksissä lienee, että kehitetään ensin ISO9001 vaatimusten mukainen toimintajärjestelmä, jota täydennetään esimerkiksi CMMI- tai ITIL-vaatimuksilla.

Iso trendi on liiketoiminnan kehittämiseen varta vasten tarkoitettujen mallien syntyminen. EFQM onkin jo verraten laajassa käytössä. Monissa IT-yrityksissä puhutaan nykyään Six Sigman käyttöönotosta, ja miksei myös IT – hallintakäytäntöjen (IT Governance) malleista. Yleensä nämä mallit toimivat parhaiten, kun perusmalleista on jo saatu enin irti ja tarvitaan uutta näkökulmaa. CMMI-yhteisössä on puhuttu jo vuosia myös korkean kypsyyden malleista (CMMI tasot 4 ja 5) osittain omana joukkonaan.

3. V&V malleja ja ohjelmistojen laatuluokituksia

Oma kokonaisuutensa laadunvarmistuksessa ovat testaus- ja tarkastuspainotteiset mallit. Usein aihepiiristä käytetään V&V lyhennettä (Verification & Validation). Ne voidaan tyypitellä monella tapaa, samaten kuin yleisetkin mallit. Tässä artikkelissa haluan tuoda esille vain vaatimaan laadunvarmistukseen kehitettyjä malleja ja standardeja, koska ne ovat vähemmän tunnettuja.

Mielestäni paras V&V toiminnan malli on IEEE1012 standardi. Se on kehitetty aikoinaan



Kuva 1.
Eri mallien
synerginen käyttö.

turvallisuuskriittisten ohjelmistojen ohjeistoksi. Sitä voidaan pitää myös hyvänä ja perusteellisenä kuvauksena V&V prosesseista, jakaantuen tarkemmiksi tehtäviksi ja tuloksiksi. Monet yritykset ovat ottaneet IEEE1012 mallin käyttöön lähes sellaisenaan, ja jotkin toimialat vaativat standardia vastaavaa toimintaa toimittajiltaan. Tällaisia toimialoja ovat esimerkiksi terveydenhuolto, ilmaliikenne ja avaruus, autoteollisuus ja ydinvoimalat. Kaikilla näillä toimialoilla on omiakin vastaavia malleja, ja niissä on kunkin toimialan vaatimia tarkennuksia yleiseen malliin nähden.

IEEE1012 sisältää 4-portaisen luokituksen ohjelmiston "kestävyydelle" (dependability). Kestävyys voi sisältää esimerkiksi turvallisuusvaatimuksia (safety, security) ja ominaisuuksia ohjelmiston luotettavuudelle. Tavoitteena on hallita ohjelmiston mahdollisesti aiheuttamia riskejä ja vahinkoja. Nelostason vaatimukset ovat hyvin tiukkoja eikä ohjelmisto saa käytännössä vikaantua lainkaan. Ykköstasolla riittää että ohjelmisto toimii oikein tai pystyy rajoittamaan vahingon siirtymällä esimerkiksi erityiseen turvatilaan. Tämäkin on kova vaatimus useimmille nyt käytössä oleville ohjelmistoille!

Tutumpi laatuluokitus ohjelmistoille ja järjestelmille on ns SIL (Safety Integrity Level). Se on kuvattu standardisarjassa IEC61508 (Functional Safety). Tässä standardissa rajoitetaan enimmäkseen järjestelmien turvallisuuteen. Standardi sisältää runsaasti yksityiskohtaisia kuvauksia ja menetelmiä, jotka on luokiteltu SIL-tasolle 1 - 4 samaan tapaan kuin IEEE1012 standardissakin. IEC61508 on laajalti käytössä toimialoilla, joilla on lakeihin perustuvia turvallisuusvaatimuksia.

Esimerkkeinä mainittakoon rautateiden ja metrojen kulunvalvonnan järjestelmät, terveydenhoidon elektronikka ja auton digitaalisesti valvotut toiminnot. On helppo ennustaa, että IEC61508 standardista tulee perusvaatimus kaikissa järjestelmissä, missä on turvallisuusuuhkia.

4. Varmistustapaus (assurance case)

Esitän tässä artikkelissa yhden verraten uuden mallin ja menetelmän ohjelmistojen laadunvarmistukseen. Kutsun sitä varmistustapaukseksi (assurance case). Turvallisuuskriittisissä järjestelmissä on tehty jo kymmenisen vuotta ns. turvallisuustapauksia (safety case). Niiden tekoon ja laskentaan on kehitetty menetelmiä ja jopa välineitä. Varmistustapaus on hyvin samantapainen, mutta on tarkoitettu yleisempään käyttöön kuin pelkkä turvallisuus. Varmistustapausten tekemiseen on tekeillä standardiperhe ISO/IEC 15026, pääosin IEEE:n resursseilla.

Varmistustapausta voisi luonnehtia jonkin laatuvaikuttamisen todistamiseksi kaikkea tarvittavia ja mahdollisia evidenssejä käyttäen. Laatuvaikuttama (argument) voisi olla vaikkapa "Ohjelmisto ei saa näyttää yksittäisen henkilön tietoja missään olosuhteissa". Usein väittämä on vaikea todistaa sataprosenttisesti, joten siihen liitetään epävarmuus sekä ei-toivottujen seurausten että virhetapausten esiintymisen suhteen.

Evidenssejä väittämän tueksi voivat olla esimerkiksi ohjelmistoprosessin kyvykkyys, erilaiset analyysit ja laskelmat, riippumattomat tarkastukset ja ohjelmiston käyttöhistoria. Esimerkiksi, jos ohjelmistoa on käytetty vikaantumatta jo 50 miljoonaa tuntia, voidaan sitä pitää verraten luotet-

tavana perustoiminnoiltaan. Mutta jos ohjelmistoa käytetään uudella tavalla tai hieman erilaisissa sovelluksissa kuin ennen, käyttökokemusten arvo evidenssinä ei olekaan niin suuri. Tyypillinen tilanne onkin, että tarvitaan monenlaisia evidenssejä ja vasta niiden muodostama kokonaisuus on riittävä väittämän oikeellisuudelle.

Käytännön todistelutilanteissa on useita väittämiä ja tarvitaan useita varmistustapauksia. Ne voivat olla myös todisteita toisilleen taikka jakaantua yksityiskohtaisemmiksi tapauksiksi. Kuvassa 2 on tyypillinen varmistustapaus, kuten se on esitetty standardin ISO/IEC15026 luonnoksessa.

Varmistustapauksen käyttö mahdollistaa suurempien ja laadullisia piirteitä sisältävien asioiden systemaattisen hallinnan. Normaalisissa testauksessa toiminnallisuus puretaan pieniksi osiksi ja niille haetaan testitapaukset osoittamaan niiden oikeellisuus. Testaamalla ei voida kuitenkaan osoittaa esimerkiksi ohjelmiston luotettavuutta, siihen tarvitaan myös laskelmia. Täydentävillä todisteilla voidaan myös kompensoida sitä tosiasiaa, että testikattavuus on harvoin sataprosenttinen ja sen saavuttaminen olisi liian kallista.

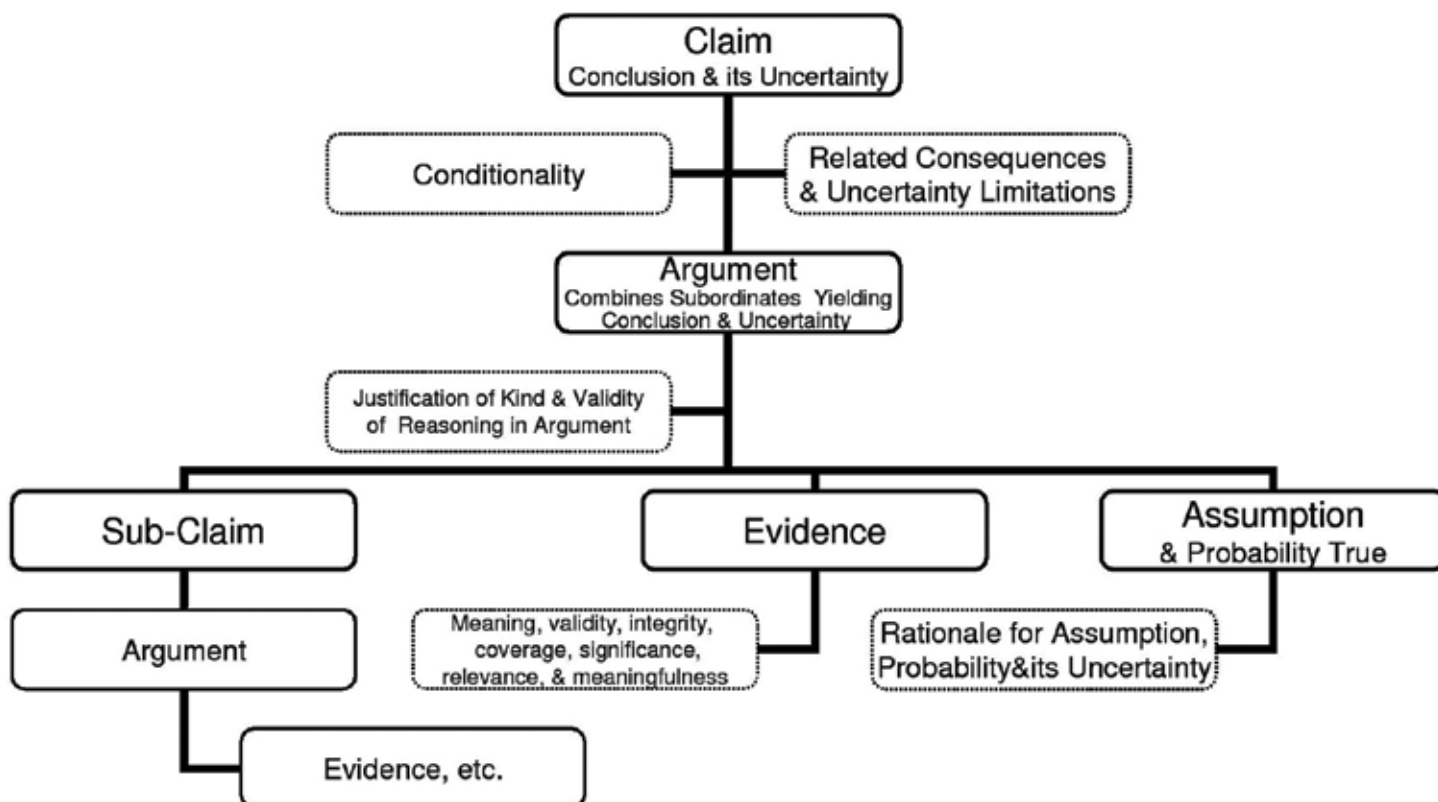
Aika näyttää, tuleeko varmistustapauksista yksi uusi tapa ohjelmistojen ja järjestelmien laadun kohottamiseen. Ensimmäisiä käyttäjiä ovat varmaankin kaikkein vaativimpia ohjelmistoja tekevät yritykset, ja sieltä ne leviävät yleiseen käyttöön.

5. Lopuksi

Muotoilin tämän artikkelin otsikon hieman tulkinvaraisesti. Malleilla on merkitystä ohjelmistojen laadunvarmistuksessa, siitä ei ole mielestäni epäilystäkään. Mutta nykyiset mallit eivät vielä takaa ohjelmistojen mallikelpoisuutta. Onneksi laadunvarmistuksen mallit ovat kehittymässä ja pystyvät käsittelemään entistä vaativampia tilanteita. Lähitulevaisuuskin lienee jatkuvaa rinnakkaineloa eli vaatimukset kasvavat mutta myös varmistuksen menetelmät paranevat.

Laadunvarmistusta käsitellään kesäkuun alussa Turussa pidettävässä SPICE2009 konferenssissa, etenkin SPICE ja CMMI mallien kannalta. FISMA on konferenssin pääjärjestäjä yhdessä Åbo Akademin kanssa. Muissakin FISMAN tukemissa konferensseissa (PROFES 2009, EuroSPI 2009) esitetään sekä teoreettisia tuloksia että käytännön esimerkkejä ohjelmiston laadunvarmistuksesta.

Kuva 2.
Varmistustapauksen
tyypillinen rakenne.





Testaus hajautettuna suunnitelmavetoisessa ja ketterässä mallissa

Kari Kakkonen toimii Testaus ja menetelmät -osastokeskuksen johtajana ja johtavana testauskonsulttina Enderossa. Kari on toiminut testauksen parissa jo vuodesta 1996 ja on konsultoinut ja kouluttanut lukemattomia asiakkaita, projekteja ja yrityksiä eri liiketoiminta-alueilla niin yksittäisissä projekteissa, koulutuksissa kuin kotimaisissa ja ulkomaisissa seminaareissa. Kari on jäsenenä Testaus-osaamisyhteisön ohjausryhmässä, Finnish Software Testing Boardin hallituksessa sekä ISTQB:n kansainvälisessä Expert Level -työryhmässä.

Testaustoimintaa harjoitetaan yhä useammin eri paikassa kuin kehitystiimi tekee kehitystyötään. Monista eri syistä tapahtuvaa testauksen hajautusta yhdistävät yhteiset haasteet. Nämä ovat ratkaistavissa tiiviillä kommunikoinnilla ja yhteisillä tietoteknisillä palveluilla, kuten portaalit ja hallintatyökalut.

Testauksen hajautuksen määrittelyä

Testaus hajautettuna ymmärretään usein erityisesti merkityksessä ”testaus ulkoistettuna eri yrityksen tekemäksi eri maantieteelliseen sijaintiin”. Maantieteellinen sijainti on tietysti eri jo silloin, kun syntyy kehitystiimistä erillinen testaus tiimi oman yrityksenkin kasvaessa. Toisaalta saman testaus-tason testauskin voidaan hajauttaa useisiin sijainteihin. Järjestelmätestaus tiimi voi olla useassa eri yrityksen toimipisteessä ja se voi olla jakanut tuon testaus-tason tehtäviä keskenään. Myös testaustasojen määrä johtaa hajautukseen. Kullekin testaus-tasolle muodostuu usein oma tiimi, joka usein istuu omissa tiloissaan. Jotkin testaus-tasot, kuten hyväksymistestaus, usein organisoituvat oman luonteensa takia automaattisesti hajautetuksi. Jos henkilöt työskentelevät muutenkin erillään, niin myös testaustehtävissä.

Taustaa hajautuksen lisääntymiselle

Testausammattilaisuuden kasvu

Testausta hajautetaan yhä enemmän. Tämä tapahtuu erityisesti testausammattilaisuuden ja yleisen testausymmärryksen kasvun ajamana, jolloin puoliautomaattisesti syntyy kehitystiimistä erillisiä testauskeskittymiä. Erillinen testauskeskittymä yleensä toivoo itselleen oma tilan saadakseen aikaan tiimihenkeä.

Isot hankkeet – monta toimittajaa

Toinen merkittävä tekijä ovat yhä isommat järjestelmähankkeet, jotka jakavat testauksen eri sijainteihin. Isoissa hankkeissa on usein eri asiain keskittyviä testaustasoja (tai perättäisiä vaiheita). Samoin isoissa hankkeissa on usein eri toimittajille tehtäväksi annettuja osia, joka ajaa testauksen tapahtumaan sekä kunkin toimittajan tiloissa että tilaajan tiloissa. Isoissa hankkeissa osa kehitystiimeistä voi toimia suunnitelmavetoisilla toimintatavoilla (esim. V-malli tai vaikkapa iteratiivinen RUP) ja osa käyttää ketteriä toimintatapoja. Tällöin hankkeen tasoinen testaus on hajautettua vaikka yksittäiset osaprojektit voivat välttää hajauttamista hyvinkin tehokkaasti. Ks. kuva 1.

Isossa hankkeessa voi olla eri ohjelmistokehityksille eri osaprojekteissa.

Isot hankkeet – monta tilaajaa

Toisaalta järjestelmän tilaaja itsessäänkin on usein maantieteellisesti hajautunut. On useita toimipisteitä ja käyttäjät istuvat yhdessäkin toimistossa ympäri taloa ilman jatkuvaa kontaktia toisiinsa. Käyttäjistä kootaan usein hyväksymistestaaajat, jolloin hyväksymistestaus helposti on nimenomaan hajautunutta. Isoissa yhteisissä järjestelmissä tilaaja voi olla myös joukko organisaatioita, jolloin hyväksymistestaaajat ovat myös organisatorisesti erillään. Hajautus on astetta laajempaa. Tällaisia hankkeita on paljon mm. julkisella sektorilla ja eläkevakuutussektorilla.

Ulkoistus

Lisäksi hyvän testausosaamisen etsintä ja kustannuspaineet ajavat ulkoistamaan testaustoimintaa eri testausyrityksille ja sitä kautta usein, joskaan ei välttämättä, eri maantieteellisiin sijainteihin. Yksittäinen testaus-taso tai –vaihe, tyy-

pillisimmin järjestelmätestaus, rajataan omaksi tehtäväalueekseen, jolle haetaan osaamiseltaan ja kustannuksiltaan sopiva testausyritys tekijäksi. Tällainen ulkoistus voi olla projektiluonteinen toimeksianto tai pysyvä kumppanuus.

Vastavoimia hajautukselle

Ketterät toimintatavat samoissa tiloissa

Vastavoimana testauksen hajautumiselle ketterät toimintatavat lähentävät testausta kehitystiimin ympärille. Ketterissä toimintatavoissa on useimmiten tavoitteena saada päivittäinen kommunikaatio toimimaan yhdistämällä kaikki ohjelmistokehitystyön roolit määrittelystä kehitykseen ja testaukseen samoihin tiloihin. Tämä toimiikin hyvin riittävän pienen ohjelmiston tai järjestelmän kehityksessä. Kun henkilömäärä ei juuri ylitä kymmentä henkeä, kaikki mahtuvat samoihin tiloihin. Hieman isommassa mittakaavassa muutama kymmenen henkeä mahtuu vielä päivittäin yhteiseen projektipalaveriin, esim. Scrum-mallia käytettäessä päivittäiseen 15 minuutin scrumiin.

Ketterät mallit roolittuvat

Usein ketterissä malleissa esitetään utopistisia tavoitteita, joissa jokainen ketterän tiimin jäsen pystyy samoihin tehtäviin. Osaamisen kannalta tämä on lopulta täysi mahdottomuus. Kukaan ei pysty osaamaan kaikkea. Käytännössä kaiken osaamisella tarkoitetaan ohjelmistokehittäjien tasolla suhteellisen monipuolista osaamista ja tehtävien jakoa; jokainen osallistuu itselleen osaksi tulleen toiminnallisuuden määrittelyyn, toteutukseen ja yksikkö- sekä integrointitestaukseen. Sen sijaan määrittelyjen tekemisen ja järjestelmätestauksen osaamiset eriytyvät omiksi rooleikseen. Määrittelyt voivat olla liiketoimintavaatimus- tai arkkitehtuuritason asioita. Järjestelmätestausta laaja-alaisempaakin testausta voi olla. Palataan siis heiluriliikkeenä V-mallista ketterään toimintatapaan menon jälkeen V-mallimaiseen roolitukseen. Eri roolit voivat tietysti silti tapahtua samassa aikaikkunassa. Hajautuksen kannalta nämä eri roolit liikkuvat helposti, yhteisten tilojen käydessä ahtaiksi, erillisiin sijainteihin. Silti ketterä toimintatapa pitää hajautettuakin ryhmää hyvin kasassa – tämä on yksin hajautuksen hallinnan avainasioita.

Osaamisen puute vähentää hajautusta

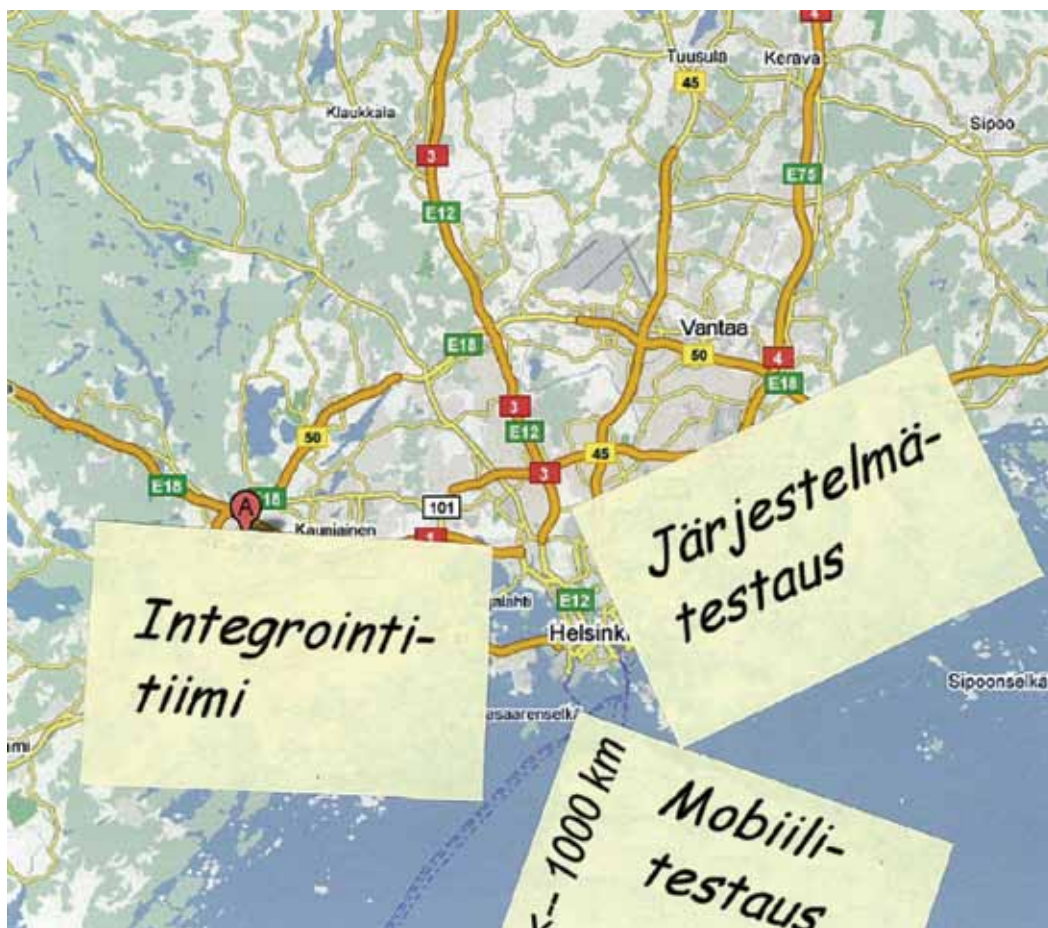
Liiketoimintaosaamisen puute johtaa nykyään yhä useammin ulkoistettuun sijaintiin asti hajautetun testauksen palauttamiseen lähelle kehitystiimiä. Tällöin on ensin ajauduttu ongelmiin ajattelematta, mitä ja miten voi tehokkaasti ulkoistaa. Ratkaisuna tuodaan esim. järjestelmätestaus takaisin omien ihmisten, tai järjestelmäkehityksen kohdalla ohjelmiston toimittajan, tehtäväksi. Ulkoistettu testaus voidaan myös tuoda kehitystiimin lähelle samoihin tiloihin, vaikkakin organisatorinen ulkoistus pidetään osaamis- ja kustannussyiden vuoksi voimassa. Joskus tietyn teknisen arkkitehtuurin testausosaamisen puute voi johtaa samaan tilanteeseen.

Hajautuksen onnistumisen keinoja

Kommunikaatio

Kommunikaatio on aina tärkeässä roolissa ohjelmistokehityshankkeissa. Hajautetun testauksen ollessa kyseessä kommunikaatio on kriittinen asia. Sen täytyy toimia hyvin ja sen täytyy toimia tarvittaessa päivittäin. Kommunikointiin löytyy keinot, kun ne etsitään. Tärkein asia on pitää kommunikointi tiiviinä, kun hajautusta ensimmäistä kertaa luodaan. Ei anneta tilanteen liukua kerran kuussa tai harvemmin tapahtuvaan kommunikointiin. Pikaviestimet kuten Skype, videoneuvottelut, sähköposti ja puhelin on keksitty ja niitä on syytä käyttääkin.

Kuva1.
Isossa hankkeessa voi olla eri ohjelmistokehitysmalleja eri osaprojekteissa.
Kuva: Matti Vuori



Keskitetty seuranta ja suunnittelu

Oivallisia keinoja säännöllisyyden aikaansaamiseen ovat keskitetty korkean tason suunnittelu ja seuranta. Käytännössä tarvitaan kokonaistestausuunnitelma ja vastaavasti kokonaistilanteen etenemisraportti, joiden kautta kaikki näkevät kokonaisuuden, eivätkä vain omaa osaansa kokonaisuudesta. Näiden menetelmien tarkkuustaso luonnollisesti vaihtelee valtavasti riippuen ohjelmistokehitysprosessista – suunnitelmavetoinen vai ketterä. Jotta nämä menetelmät toimivat, tarvitaan yleensä keskitetty testauksen koordinaattorin rooli, joka katsoo kokonaisuutta yli mahdollisten maantieteellisten ja organisaatorajojen. Roolin ei tarvitse olla täysipäiväinen työ, vaikka se sellaiseksi isossa hankkeessa helposti muuttuukin. Onko ensin olemassa keskitetty rooli vai kokonaiskuvan aikaansaamiseen ja ylläpitämiseen tarvittavat menetelmät, on tyypillinen monikana-ongelma. Molemmat kuitenkin tarvitaan.

Läpinäkyvyys

Toisaalta läpinäkyvyys eri sijainneissa tapahtuvan testauksen yksityiskohtiin on sekin tärkeää. Pelkällä kokonaiskuvalla ei pääse pitkälle. Voidaan ajatella, että on aktiivista ja passiivista kommunikointia. Aktiivisessa kommunikoinnissa järjestetään esim. puheyhteys hajautettujen ohjelmistokehityksen eri tahojen välille, esim. statuspalaveri kerran päivässä Skypen välityksellä. Passiivisessa kommunikoinnissa luodaan läpinäkyvyys. Annetaan mahdollisuus kaikille osapuolille ja etenkin koko testauksen koordinaattorille nähdä, mitä tapahtuu, periaatteessa milloin vain. Käytännössä tämä saadaan aikaan käyttämällä keskitettyjä tietoteknisiä ratkaisuja. Pitäisi pystyä luomaan läpinäkyvyyden aikaansaaminen automaattiseksi. Käytännössä tehdään työtä keskitettyjen ratkaisujen kautta, eikä niin, että vain raportoidaan keskitettyihin ratkaisuihin.

Julkaisukäytännöt

Toki käytössä täytyy olla muitakin toimintatapoja. Näistä olennaisimpia ovat päivittämisen koonnin mahdollisuus, sovittu tiivis julkaisusykli testausympäristöihin ja julkaisuselosteet. Päivittäinen koodin koonti eheäksi, yksikkötestatuksi, ajokelpoiseksi ohjelmistoksi tai järjestelmäksi pitää ohjelmiston tai järjestelmä eheänä. Samalla se poistaa hajautuksen perisyntejä – suuret viiveet ja liialliset rinnakkaiset koodin kehityshaarat. Päivittämisen koonnin asennus kuitenkin pitäisi olla mahdollisuus, ei pakote. Testaushan tarvitsee myös testausrauhaa. Toisaalta vanhaa versiota ei kannata liian pitkään testata. Pitäikin sopia tiivistä julkaisusyklistä, jolloin saadaan jo raportoituja vikoja varmistustestaukseen uuden version myötä. Sopiva sykli voi olla esimerkiksi asennus kaksi kertaa viikossa. Lopulta julkaisuseloste, tuo kaikille testausympäristöille olennainen asia, nousee uuteen arvoonsa hajautuksen kohdalla. Hyvästä julkaisuselosteesta selviää, mikä

on muuttunut aiempiin versioihin nähden: mitä uutta, mitä korjattua. Hyvä julkaisuseloste syntyy automaattisesti.

Osaaminen ja tuki

Ehdoton edellytys hajautuksen onnistumiselle on myös se, että kaikilla hajautetuilla testausosapuolilla on riittävät tiedot ohjelmiston käytöstä, teknisestä arkkitehtuurista sekä testauksen toimintatavoista ja työkaluista. Ohjelmiston käytön osaaminen on siis ns. liiketoimintatuntemusta. Kaikkien osapuolien ollessa yhdessä tämä tieto on helpostikin saatavissa. Hajautetusti jokin tieto jää monesti puuttumaan. Täytyy siis organisoida riittävä aktiivinen tuki. Tämä jää keskitetyn testauksen koordinoijan vastuulle.

Keskitetty ympäristö

Tekninen ympäristö on merkittävin tekijä, joka mahdollistaa hajautetun testauksen toiminnan lähes kuin oltaisiin samassa paikassa. Puhutaan erilaisista etätyöratkaisuksista. Testausympäristöihin pitää olla reaaliaikainen pääsy kaikilla osapuolilla, toki käyttötarpeiden mukaan rajattuna. Virtualisoidut testausympäristöt mahdollistavat tämän vielä varhaisempia ympäristöjen jaon teknologioita paremmin, yleisimmin käytetään VMWarea. Tarvitaan yhteinen projektikommunikaation alusta, työtila tai portaali, johon jokainen pääsee lukemaan ja lisäämään tietoja, niin tekstiä kuin tiedostojakin – yleisimmin käytetään wikiä. Myös tiedostopalvelimet tulee jakaa. Testauksen hallinnan ja konfiguraation hallinnan työkalut täytyy myös saada toimimaan paikasta riippumatta.

Yhteenveto

Testauksen hajautus yleistyy, se on tosiasia. Oli syy sitten iso hanke ja sen tuomat useat testaus-tasot tai ulkoistus, testaukselle on järkevää löytää toimivia hajautuksen muotoja. Hajautuksesta saadaan menestystarina, kun hoidetaan hyvin kommunikointi, keskitetty riittävä suunnittelu ja seuranta, hyvät julkaisukäytännöt, läpinäkyvyys eri osapuolten kesken, osaamisen tuen tarjoaminen sekä toimiva hajautettu tekninen ympäristö. Samaan aikaan kannattaa tehdä tiiviissä tiimissä samassa paikassa niitä asioita, mitkä ilman hajautusta voi tehdä. Ei siis luoda hajautettua tiimiä turhaan viereiseen huoneeseen, vaan nautitaan integroidun tiimin nopeasta kommunikaatiosta.

Ohjelmistostandardeja laidasta laitaan



Teemu Vesala toimii laatukonsulttina Qentinel Oy:ssä. 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)

"Vaatusmäärittelyt eivät kuulu missään tapauksessa hyvään ohjelmistonhakintaprosessiin! 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 suorituskkyä!" (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 tyyllisiä 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 stan-

dardointia jokaisella valmistajalla olisi ollut lopulta oma, toisten valmistajien kanssa yhteensopimaton toteutus. Telnet kuitenkin määriteltiin IETF STD-8:ssa, jonka 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 Java-leen 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 tietoturva-vaatimuksia. 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-25000 on standardiperhe, jossa osasta riipuen 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 selaisenaan, 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 kehittäjän 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



Jonas Hinkkanen työskentelee Flanderilla testauksen projektipäällikkönä. Jonaksella on kokemusta sekä yksityisen sektorin että julkishallinnon SOA-hankkeiden testauksen suunnittelusta ja johtamisesta.

SOA – testattuna

Palvelukeskeinen arkkitehtuuri asettaa uusia haasteita testaukselle. Toiminnallisuuden ja liiketoimintalähtöisyyden varmentamisen lisäksi testaustoimenpiteiden on kyettävä yhtälailla varmentamaan palveluiden yhteensopivuus, paljastamaan mahdolliset haavoittuvuudet sekä todentamaan suori-

tuskyky.

Palvelupohjaisten arkkitehtuurien (SOA) yleistyminen organisaatioiden IT-järjestelmäkehityksessä on tuonut liiketoimintaa sekä tietohallinnon järjestelmäkehitystä lähemmäksi toisiaan. Lähen-
tymisestä ja hyvästä vuorovaikutuksesta sekä integraation helpottumisista huolimatta liiketoi-
minnan ja tietohallinnon väliin saattaa edelleen jäädä katvealue josta pahimmassa tapauksessa yksikään osapuoli ei ensisijaisesti vastaa – tes-
taus.

Organisaation laadunvarmistustoimenpiteillä pyritään ensisijaisesti lieventämään niitä liiketoi-
minnan riskejä joita tuotteen tai palvelun käyt-
töönotto aiheuttaa.

Palvelupohjaiseen arkkitehtuuriin perustu-
vien web-palveluiden testaamisessa on kuitenkin tiettyjä piirteitä jotka organisaation tulisi vähin-
täänkin tiedostaa laadunvarmistustoimenpiteitä suunniteltaessa:

Lähestyminen SOA-järjestelmien laadunvarmistukseen

Lähtökohtaisesti SOA testausta tulee lähes-
tyä samoin kuin mitä tahansa muutakin testaus-
projektia. Laadunvarmistustoimenpiteiden tulee pyrkiä lieventämään käyttöönotosta aiheutu-
via liiketoiminnan riskejä. Näistä toimenpiteistä
syntyy lisäksi ohjaus- ja johtoryhmille sekä muille
sidosryhmille tietoa kohteena olevan järjestelmän
laatuvaikutelmasta. Tietoa järjestelmän laatu-
vaikutelmasta voidaan hyödyntää päätöksenteon
tukena.

Lieventääkseen riittävästi riskejä ja tuottaak-
seen kattavasti tietoa järjestelmän laatu-
tuntumasta tulisi testaustoimenpiteiden kohdistua
toiminnallisuuden varmentamisen lisäksi yhtälailla
palveluiden yhteensopivuuteen, haavoittuvuuteen
sekä suorituskyvyn todentamiseen.

Toiminnallisuuden varmentaminen

Web-palveluiden toiminnallisuuden testaami-
sessa tulee keskittyä vaatimuksien ja määrittelyi-
den varmentamiseen. Tarkoituksena on selvittää
vastaavatko rakennetut web- palvelut ja niiden
julkaisemat rajapinnat liiketoiminnan niille aset-
tamia vaatimuksia. Testaaminen voi tapahtua kir-
jattujen käyttötapauksen pohjalta mutta yhtälailla
tärkeää on lisäksi selvittää miten palvelut vastaa-
vat tietynmuotoisiin aineisto-ohjattuihin palvelu-
kutsuihin sekä koeistaa järjestelmää positiivisen ja
negatiivisen testauksen keinoin. Järkevällä palve-
lukutsujen ketjuttamisella voidaan usein saavut-
taa korkea testauksen kattavuusaste suhteellisen
pienellä panoksella. Tästä on edelleen lyhyt matka
toimivaan testiautomaatioon sekä mallipohjaiseen
testaukseen joiden on todettu soveltuvan varsin
hyvin SOA testaukseen. Toiminnallisen testauksen
edellytyksenä on aina kuitenkin hyvin dokumen-
toidut ja saatavilla olevat palvelukuvaukset.

Palveluiden yhteensopivuus

Palvelukeskeisessä arkkitehtuurissa palvelui-
den yhteensopivuuden varmentaminen on ensi-
arvoisen tärkeää. Voidakseen taata sovellusten
liitettävyyden ja uudelleenkäytön sekä yleensäkin
saavuttaakseen SOA:n avulla tavoiteltavat hyödyt
tulee organisaation varmistaa suunnittelemiensa
palveluiden yhteensopivuudesta. Yhtä tärkeää on
varmistaa asianmukainen poikkeus- sekä virheti-
lanteiden hallinta.

Luonteva tapa yhteensopivuuden varmistami-
seksi on tehdä passiivinen WSDL- sekä aktiivinen
SOAP analyysi WS-I Basic Profilea vastaan. Pas-

siivisessä WSDL analyysissä WSDL dokumentti tarkistetaan WS-I Basic Profilea vasten ja pyritään löytämään profiilin vastaisia rakenteita. Aktiivisen SOAP yhteensopivuustestauksen ideana on lähettää palvelulle kutsuja, joilla pyritään löytämään WS-I Basic Profile vastaisia reaktioita. Jokaisesta palvelusta otetaan tyypillisesti muutama operaatio, joilla testataan muutetut SOAP-kutsut.

Haavoittuvuuden huomioiminen

Haavoittuvuuden testaamisen kantavana ideana on muuttaa palvelupyyntöjen sisältöä siten, että kyetään selvittämään miten rakennetut palvelut vastaavat näihin muutettuihin palvelupyyntöihin. Testillä pyritään paljastamaan mahdolliset riskitekijät kuten heikosti toteutetut syötteen vastaanotot ja varmistukset sekä puutteelliset poikkeuksien käsittelyt joita mahdollinen hyökkääjä voisi käyttää hyödykseen.

Syötteinä haavoittuvuuden testaamisessa voidaan käyttää esimerkiksi erilaisia DTD- viitteitä, XML-prosessointi kutsuja sekä XSD-arvo- ja tyyppimutaatioita.

Suorituskyvyn todentaminen

Suorituskykytesteillä arvioidaan palveluiden kykyä käsitellä suurta määrää yhtäaikaista kut-

suja. Suorituskykytestauksen testitapauksina on suositeltavaa käyttää toiminnallisuustestauksen testitapauksia jolloin säästetään sekä suunnitteluaikaa, että varmistetaan testitapausten oikeellisuus.

Suorituskykytestiajoihin on suositeltavaa suunnitella mahdollisimman monipuolisesti erilaisia kuormaprofileja. Vaihtelevilla kuormaprofileilla pystytään varmentamaan palveluiden käytettävyyttä kiiretunteina sekä selvittämään palveluiden käsittelykyvyn sieto piikkikuormia kohtaan. Lisäksi voidaan tarkastella palveluiden suoriutumisesta pitkäkestoisesta yhtäjaksoisesta kuormasta tai ajaa järjestelmä vikatilaan ja tutkia toipumista.

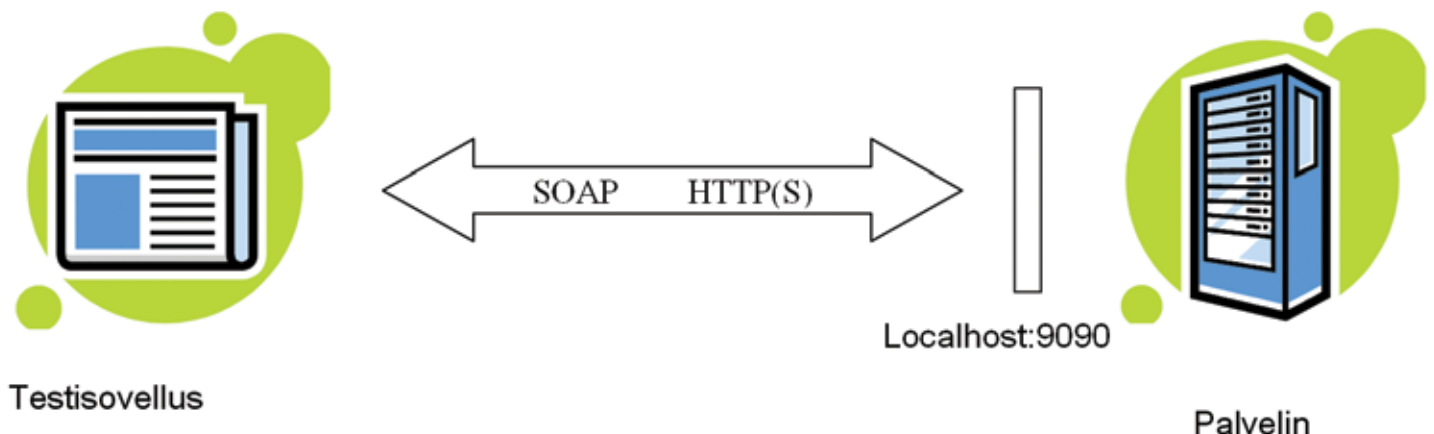
Sanasto

DTD	Document Type Definition
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
WS-I	Web Services Interoperability Organization
WSDL	Web Services Description Language
XML	Extensible Markup Language
XSD	XML Schema

Kuva 1.
Testikonfiguraatio

- ✓ WSDL lataus
- ✓ Toiminnallisuus-, Yhteensopivuus-, Haavoittuvuus ja Suorituskykytestaus

- ✓ Web Service host





Vuoden testaaaja 2009 Ville Vuorenojaa onnittelee edellinen vuoden testaaaja Erkki Pöyhönen (kuva: Kirsi Stenberg, Tieturi)

Vuoden testaaaja 2009 valittu

TestausOSY myönsi Tieturin kanssa Tieturi Testaus 2009 tapahtumassa tänä vuonna kolmannen kerran Vuoden Testaaaja -nimikkeen. Vuoden testaaajaksi 2009 nousi tiukassa 847 ääntä keränneessä äänestyksessä Ville Vuorenoja Digia Oyj:stä. Yhtenä äänestyksen kohteena oli myös Nimetön Testausinsinööri. Tässä artikkelissa julkaistaan kunnianosoituksena kaikille Suomen testaaajille äänestäjien ytimekkäitä kumarruksia nimettömiä arjen sankareita kohtaan.

Onnittelut Vuoden testaaajalle 2009

Vuoden Testaaaja 2009 Ville Vuorenojaa kuvaillaan seuraavasti. Ville on osoittanut ammattitaidollaan ja periksiantamattomuudellaan juuri niitä ominaisuuksia, joita erinomaiselta testaaajalta vaaditaan. Hänellä on ammattimainen ote testaukseen ja testaustiimin vetämiseen sekä dynaaminen ja innovatiivinen asenne. Villen halu kehittää omaa ja tiiminsä ammattitaitoa on onnistuneesti ajanut koko tiimin ja myös testaus-alan etua - kehittäen yrityksemme testauksen laatua, ammattitaitoa ja työkaluja. Onneksi olkoon, Ville!

Kunnianosoituksia Nimettömälle testausinsinöörille

Suurin työ tehdään testaustiimeissä. Siellä pakeretaan se perustyö, kaikenkestävä pohja: testitapausten määrittely, suoritus ja raportointi. Kunniaa ja kumarruksia heille kaikille!

Äänestäjien suusta:

- Olen nähnyt nimettömien testausinsinöörin niin monta sankaritekoa, että ääneni menee ehdottomasti nimettömälle testausinsinöörille. Heidän tekemiensä sankaritekojen palkintona on usein vain pelkkää valitusta, siitä että hankkeen aikataulut kärsivät työntuloksena. Todella harvoin nimetön testausinsinööri saa tunnustusta, siitä uurastuksesta, jonka hän on tehnyt konkreettisesti parantaakseen tuotteen laatua.
- Siellä se kaikki perustyö tehdään, ja sieltä löytyy suunnaton määrä osaamista :)
- Kunnia kuuluu kaikille ruohonjuuritasolla raataville!
- Yksi henkilö voi johtaa ja kehittää prosesseja, mutta ryhmässä erilaiset tulkinnat parantavat lopputulosta
- Tunnettujen nimien sijasta kunnia nimettömille testausinsinööreille 'Software Testing Engineers', joita työskentelee paljon lääketekniikan, mobiilitekniikan ja vaativan ohjelmistokehityksen parissa! Ilman heitä saisimme sutta ja sekunda.
- Nimetön testausinsinööri on se henkilö, joka harvoin jos koskaan saa työstään kiitosta. Jos asiat menevät hyvin, potin korjaa muut tahot. Jos eivät, siitä kyllä saa kuulla.
- Mielestäni jokainen testaaaja on jatkuvasti kasvavan paineen alla, varsinkin yrityksissä, joissa työtä viedään ulkomaille. Bulkkitestaus yritetään tietenkin viedä tehtäväksi sinne missä on halvempaa ja siksin työnkuva muuttuu koko ajan vaativammaksi työn alkuperämaassa. Lisäksi, kummallista kyllä, vielääkään ei ole saatu siihen ryhtiä, että testauksen ajasta otetaan se aika, jonka kehitystyö vaatii. Oli olosuhteet mitkä hyvänsä, on testaus suoritettava yhtä laadukkaasti.
- Aivan liian vähän muistetaan niitä arjen sankareita jotka päivästä toiseen joutuvat ratkaisemaan ongelmia useasti mitään erityistä kiitosta tai huomiota. Ilman heitä konsultitkin olisivat työttömiä.
- Teppo Testaaaja on se tärkein lenkki joka testiprojektissa!
- Hienoa että meillä on lipunkantajia jotka tuovat testaustyön tärkeyttä esiin, mutta arjen raatajat tulisi muistaa myös.
- Bonukset perusduunareille!
- Hän tulee liian harvoin esille. Tässä nyt ääneni hänelle - nimettömälle testausinsinöörille.
- Kaikkein parhaat testaaajat tekevät niin hyvää työtä, saumattomasti, tehokkaasti ja laadukkaasti, että heitä ei koskaan juuri huomatakaan. Kun haetaan _testaaaja_, eikä alan vaikuttajaa, ääni menee "tuntemattomalle testaaajalle".
- Me kaikki testaaajat ansaitaan hieman "näkyvyyttä" ja arvostuksen nousua myös :)
- Kunnia ja kumarrus kaikille heille ja lisäksi muillekin testausalan järeille ammattilaisille, jotka päivästä toiseen, viikosta viikkoon virstanpylvästä tuotteen elinkaaren loppuun asti tekevät osansa ammattitaitoisesti laadunvarmistamisen saralla.
- Tämä on se kaveri joka pitää pyörät pyörimässä.
- Testaaaja kuin testaaaja. Testaaaja saa harvoin kiitosta; päinvastoin, saa kuraa niskaan, kun löytää "bugeja".
- Kunnia sille kenelle se kuuluu, eli kaikille testaaajille tasapuolisesti! :)
- Suurin työ ja kehitys testauksen alalla tapahtuu kuitenkin siellä suorittavassa portaassa jossa ongelmiin törmätään ja ongelmia ratkaistaan.
- Team rules



Eeva Kangas toimii Käyttökokemusyksikön johtajana Digialla. Hänellä on 14 vuoden kokemus erilaisista käytettävyys- ja käyttöliittymäsuunnittelun hankkeista.

Käytettävyystyö osana prosesseja



Juha Jääskinen toimii OpenMethod ohjelmistokehitysmenetelmistön tuotepäällikkönä Digialla.

Vaatimustenhallinta ja käytettävyys ovat tyypillisiä ohjelmistotuotannon teemoja, jotka huomioimalla on mahdollista tehostaa toimintaa ja säästää huomattavia kustannuksia. Jopa yli puolet ongelmista on ratkaistavissa huomioimalla käytettävyys osana kehittämisprosessia. Artikkelin lähtökohtana on käyttäjälähtöinen kehitystyö, jossa käytännön käytettävyysosaamista integroidaan osaksi ohjelmistokehitysmenetelmää.

Mahdollisuuden analysointi ja käyttäjätutkimus
Käytettävyysprojektit aloitetaan asiakastapaamisella, jossa on tärkeintä ymmärtää asiakkaan ja käyttäjän vaatimukset oikein. Usein selviää, ettei asiakas välttämättä edes tiedä tai ole edes miettinyt kuka kehitettävän tuotteen käyttäjä on. Tärkeää on myös analysoida mahdolliset kilpailevat vaihtoehdot, millaisia ne ovat ja mitä uutta juuri tällä idealla saataisiin.

Etenkin laskusuhdanteessa kehitysbudjetit ovat usein tiukoilla ja yritysten on erityisen tärkeää miettiä mihin rahaa satsataan ja mihin ei, mitä kannattaa tehdä ja mitä ei. Monesti houkuttaa ajatus tehdä tuote valmiiksi tuotepäällikön ja ohjelmistokehittäjän yhteistyönä - mitä vähemmän osapuolia, sitä vähemmän tekemistä. Säästetäänkö tällä sitten rahaa tai lopulta edes aikaa?

Business casen kuvaaminen ja konseptisuunnittelu

Ohjelmistokehitys on usein aikaa ja rahaa vievää työtä. Mitä tapahtuu, kun tuote saadaan nopeasti markkinoille? Voidaan ehkä huomata, ettei käyttäjillä olekaan tarvetta tai edes halua

käyttää tuotetta. Tai ehkä tuote onkin liian vaikea-käyttöinen. Tai ehkä tuote vain näyttää niin paljon kamalammalta kuin kilpaileva tuote. Voi olla, että ohjelmistokehitykseen käytetty aika onkin mennyt hukkaan – ehkä säästö tehtiin väärässä paikassa.

Kentällä käydessä käyttäjiltä saa monenlaista tietoa: ketä todelliset käyttäjät ovat, millaisia tarpeita ja toiveita heillä on, mitä ominaisuuksia he tuotteelta haluavat, missä he käyttäisivät tuotteita, miksi ja miten. Päätettäessä jatkaa tuotteen suunnittelua, on tärkeää olla mukana niin tuotepäällikkö, käytettävyysasiantuntija kuin ohjelmoijakin. Kaikkien osaamista tarvitaan, jotta saadaan aikaan tuote, joka on käyttäjille tarpeellinen ja jota käyttäjät osaavat käyttää. Tuotteella on oltava markkinat ja sen pitää olla toteutettavissa nykYTEKNOLOGIALLA. Tuotteesta on tärkeää karsia turhat ominaisuudet ja toteuttaa vain ne, joista käyttäjät tuntuivat innostuvan - kaikki muu on turhaa.

Käyttöliittymän suunnittelu

Tuotteen käyttöliittymää suunniteltaessa voidaan käyttäjiltä saatua palautetta jälleen hyödyntää. Oikean tiedon avulla voimme priorisoida ominaisuuksia, joita käyttäjät usein käyttävät tai ovat jostain muusta syystä käyttäjille tärkeitä. Nämä ominaisuudet kannattaa laittaa käyttöliittymässä hyvin helposti saataville.

Kun tekniset henkilöt ovat mukana jo tässä vaiheessa tuotekehitystä, he sitoutuvat projektiin paremmin ja saavat tärkeää tietoa käyttöliittymän suunnittelusta. Ohjelmoija voi keskittyä vain siihen miten tuotteen toteuttaa. Toisaalta käyttöliittymäsuunnittelijankaan ei tarvitse elää tekno-

logian ehdoilla vaan hän voi suunnitella tuotteen käyttäjien luonnollisiin tarpeisiin ja tapaan tehdä asioita. Käyttäjää ei kannata yrittää pakottaa uusiin tapoihin vaan tuotteen pitää mukautua käyttäjien tapoihin.

Jos tuotteen ostaja ei ole sama henkilö kuin tuotteen käyttäjä, niin käytettävyyden merkitys ostopäätökseen on usein valitettavan pieni. Yrityksiin ostetaan edelleen paljon tuotteita joiden käytettävyys on todella kurja, eikä useinkaan mietitä kannattaako ostaa halpaa tuotetta. Vaikeasti käytettävä tuote vie työntekijöiden aikaa, hermoja ja tehokasta työaikaa. Huonot tuotteet huonontavat myös työviihtyvyyttä.

Prototyypit ja käytettävyydestaus

Ennen tuotteen toteuttamista kannattaa tehdä prototyyppi, sillä sen saadaan tärkeää tietoa siitä, mitä ongelmia käyttäjillä vielä on ja miksi, mitkä asiat toimivat jo hyvin ja mitä muutoksia käyttöliittymään tulee tehdä, jotta tuotteesta tulisi entistä parempi. Käytettävyydesteihin kannattaa ottaa mukaan tuotteen todelliset loppukäyttäjät, emmehän itse maistele kissanruokaakaan ja arvele maistuuko se kissan suussa hyvälle vai ei.

Demojen testauksen jälkeen tehdään käyttöliittymästä ja sen toiminnallisuudesta yksityiskohmainen dokumentti. Kun dokumentti on tarkka, voidaan olla varmempia että toteuttajat tekevät

siitä juuri sellaisen kuin oli ajateltukin vaikka toteutus olisi toisella puolella maailmaa. Samalla voidaan säästyä monilta virheiltiltä, joiden raportointi ja korjaus vievät usein paljon aikaa.

Kun tuote on lopulta toteutettu ja saatu siihen oikeat grafiikat, tehdään vielä yksi käytettävyydestesti, jossa saadaan taas erilaista informaatiota. Tällöin käyttäjä antaa usein palautetta vasteajoista, grafiikasta ja käytetyistä termeistä. Nämä asiat on usein suhteellisen helppo korjata tuotteeseen, kun taas puutteellisten ominaisuuksien tai vaikearakenteinen käyttöliittymä ovat tässä vaiheessa hyvin kalliita korjauksia.

Johtopäätökset

Usein käsitys käytettävyyden toteuttamisesta on edelleen tehdä ensin tuote valmiiksi, antaa se vasta sitten käytettävyydsiantuntijoiden testattavaksi ja toivoo hyväksyttyä tuotetta tai saada hyvin pieniä ja helposti toteutettavia korjauskehdotuksia. Useimmat eivät ole tietoisia käytettävyyden laajuudesta ja kuvitellaan käytettävyyden olevan vain se kakun koriste, pieni kiva nappi käyttöliittymässä, jolla saadaan tuotteesta "cool". Käytettävyys kattaa niin visuaalisen suunnittelun, helpokäyttöisyyden kuin tuotteen hyödyllisyydenkin ja onnistuneesti toteutettuna se on mukana läpi koko ohjelmistokehityksen elinkaaren.





Jari Mitro on 42-vuotias ohjelmistotestausyrittäjä, jonka yritys Test-It Mitro on toiminut kohta vuoden. Hänellä on testauskokemusta noin 9 vuotta ja IT-alan työkokemusta noin 15 vuotta. Keskeisenä kiinnostuksen aiheena hänellä on tuotteiden loppukäyttäjää.

Loppukäyttäjää testaajana – kokemuksia keskeneräisistä tuotteista

Yleisesti ottaen testaus on viime vuosina kehittynyt merkittävästi mutta edelleen loppukäyttäjille tarjotaan keskeneräisiä raa-kileita. Loppukäyttäjää joutuu toimimaan testaajana ja löytämään virheitä, jotka olisi pitänyt löytää järjestelmän eri testausvaiheiden aikana. Artikkelissa pohditaan tätä tilannetta loppukäyttäjän näkökulmasta.

Testaajana virheiden löytäminen testiympäristöissä testauksen kohteista on paitsi tarkoituksenmukaista myös palkitsevaa. Kun taas puhutaan tuotantoympäristöistä, niin virheen löytäminen ei tuota mielihyvää, vaan negatiivisen kokemuksen lisäksi suhtautuminen tuotteen tai palvelun tuottajaan kohtaan muuttuu. Itselläni on viime ajoilta negatiivisia kokemuksia tästä enemmän kuin tarpeeksi. Loppukäyttäjät-testaajana olen lähettänyt kokemuksistani kirjallisia virheraportteja asianomaisille tahoille. Suurimmassa osassa tapauksia vastapuoli ei ole vastannut lainkaan. Joskus on vastattu olevan aivan normaalia, että kaikki ei voikaan toimia vaativan järjestelmäintegraation jälkeen. Tai jopa, tuote on niin uusi, ettei voi toimia vielä luotettavasti tai aina oikein. Pidetään siis normaalina, että tuote on julkaistu kesken-eräisenä tai viallisena.

Loppukäyttäjää ei tarkoituksella etsi virheitä – miksi etsisikään? Tuotteen, joka on ostettu tai valittu muuten käyttöön, pitää toimia oikein. Harva loppukäyttäjät laatii testisuunnitelmaa ja testitapaustuetteloja sekä valitsee testaustekniikoita alkaessaan käyttää uutta tuotetta. Miksi loppukäyttäjät sitten löytävät virheitä? Yksinkertaisesti siksi, että tuotteet sisältävät virheitä, joita ei ole syystä tai toisesta löydetty tuotantoonsiirtoa edeltävissä testeissä.

Kaikkia virheitä ei suurista ja monimutkaisista tuotteista luonnollisestikaan pystytä suodattamaan. Hämmennystä herättää lähinnä, että vasta tuotannossa löydetään ohjelman tai verkkopalvelun käytetyimpien toimintojen virheet tai joudutaan toteamaan todella suuret ongelmat toiminnallisuudessa. Juuri tällaiset seikat saavat normaalin peruskäyttäjänkin epäilemään, joskin aiheetta, onko tuotetta testattu lainkaan. Laatua ei testaamalla pysty luomaan mutta testauksella pitäisi kyetä löytämään poikkeamat vaatimusmäärittäisiin. Asian pitäisi kiinnostaa myös kaikkia tuotteen omistajia ja kehittäjiä, sillä aiheuttaahan virhe sitä suuremman kustannuksen mitä myöhemässä vaiheessa se löytyy.

Tuotteen käyttäjämäärät voivat olla suuria eivätkä kaikki lue käyttöohjeita, jos niitä ylipäänsä on saatavilla tai nekin ovat virheellisiä. Loppukäyttäjillä on kuitenkin jonkinlainen oletus siitä miten tuotteen tai palvelun pitää toimia, esimerkiksi tekemällä vertailua vastaaviin olemassa oleviin tuotteisiin ja palveluihin. Lisäksi on mahdollista, että tuotetta tuntematon käyttäjä tekee hyvää negatiivista testausta, koska "ei osaa" käyttää tuotetta ja löytää sillä tavalla virheitä. Toisaalta loppukäyttäjät voivat toki olla tuotteen tai liiketoiminta-alueen todellisia asiantuntijoita ja huomaavat piankin, jos tuote ei toimi kuten pitäisi tai toimii kuten ei pitäisi.

Ohjelmiston hankkijalle viallinen tuote saattaa pahimmillaan tuottaa käyttöongelmien lisäksi merkittäviäkin taloudellisia menetyksiä. Toimimatonta tai virheellisesti toimivaa tuotetta ei voi käyttää mutta liiketoimintaa on kuitenkin harjoitettava. Joskus voidaan palata käyttämään vanhaa järjestelmää mutta se ottaa aikansa eikä aina luon-

nollisestikaan ole mahdollista. Töitä pitää pystyä tekemään ja asiakkaita palvelemaan. Mitä, jos uusi puhelinjärjestelmä ei toimi, tai yhteys tilausjärjestelmästä varastoon tai asiakasrekisterin ja laskutuksen yhteys ei toimi? Entä miten loppuasiakas suhtautuu aiheettomiin huomautuslaskuihin tai virheellisiin tavaratoimituksiin. Ainakin osa asiakkaista saa tarpeekseen ja yritys menettää asiakkaita ja sitä kautta myyntiä. Ainakin osa töistä joudutaan myös tekemään manuaalisesti ja mahdollisesti on tarvetta myös tilapäistyövoiman hankinnalle, eli lisää kustannuksia.

Loppukäyttäjää ei useinkaan kiinnosta mikä ohjelmistokehitysmalli on ollut tuotetta tehdessä käytössä tai tekniset yksityiskohdat eikä projektiryhmän koko tai kuka on hoitanut tuotteen verifoinnin ja validoinnin. Resurssipula tai tiukka aikataulu ei myöskään varmasti hetkauta suuntaa tai toiseen.

Katselmoidaanko vaatimusmäärittelyt, testisuunnitelmat ja testitapausluettelot, testikatavuudet tai testaustekniikat tai missä vaiheessa

testit on suunniteltu tai integroidaanko koko kehityksen ajan sekä muut testauksen kannalta olennaiset asiat eivät yksinkertaisesti kuulu tuotteen loppukäyttäjälle mutta puutteet tai ongelmat niissä välittyvät hänelle. Toimiva tuote tai palvelu on loppukäyttäjällä oikeutetusti toiveena ja hänellä on oikeus reagoida mikäli näin ei ole.

Testauksesta ymmärretään nykyään paljon ja siihen panostaminen on sekä toimittajien että tilaajan yhteisellä vastuulla ja yhteinen etu. Hyvän testauksen avulla vältytään päätyästä aiheeksi uutisiin tai keskustelupalstoille, puhumattakaan kalliista korjaus- ja kenties korvausprosesseista.

"Tai sitten ei...". Tässä tuotteessa olevasta ohjelmistovirheestä jouduin itse kärsimään. Paikalle tarvittiin huoltomies avaamaan ovi avaimella. Sitä ennen sain matkustaa kahden kerroksen väliä puolisen tuntia, vaikka en mitään nappia kerinnyt painaakaan.





FT Timo Jokela on toiminut käytettävyyssalalla 90-luvulta lähtien: VTT:llä, Nokialla, Oulun yliopistolla ja Joticon Oy:ssä. Hän on kehittänyt käytettävyyssuunnittelun prosesseja ja -menetelmiä, sekä niiden arviointimenetelmiä, ja tehnyt käytettävyyteen liittyviä julkaisuja yli 80. Hän on mukana käytettävyyden ISO-standardoinnissa, sekä alan kansainvälisissä verkostoissa.

CUE=
comparative
usability
evaluation

Sama ongelma – erilaiset tulokset

Miten varmistaa käytettävyytestauksen laatu?

Käytettävyytestaus on keskeinen käytettävyyden varmistamismenetelmä. Käytettävyytestaukseen menetelmänä liittyy kuitenkin laatuongelmia: tutkimusten mukaan testauskäytännöt ja testien tulokset riippuvat voimakkaasti siitä, kuka sattuu tekemään testit. Testien tekijöiden tulisi varmistaa testaus toimintansa laatu arvioimalla kriittisesti omia käytäntöjään ja oppimalta muilta. Käytettävyytestien asiakkaiden kannattaisi käyttää useampaa käytettävyydenäkemyksiä, ja harkita muitakin käytettävyysoimenpiteitä.

Tutkimukset osoittavat käytettävyyssuunnittelun laatuongelmat

Käytettävyyssuunnittelu on ehkä keskeisin yksittäinen menetelmä sovellusten käytettävyyden varmistamisessa. Käytettävyyssuunnittelusta tehdään yritysten ja organisaatioiden sisäisesti, sekä kaupallisten toimijoiden toimesta.

Käytettävyyssuunnittelulla tarkoitetaan menetelmää, jonka pääpiirteitä ovat etukäteen valmistellut testitehtävät, testitehtävien teettäminen sovelluksen loppukäyttäjillä, ja tulosten perustuminen käyttäjien havainnointiin (miten käyttäjien suoriutuvat annetuista tehtävistä). Mutta miten hyvää laatua tällainen käytettävyyssuunnittelu on menetelmänä? Miten laadukkaita ovat käytettävyyssuunnittelun tulokset?

Kuvitellaan tilanne, että usea käytettävyyssuunnitteluryhmä testaa saman sovelluksen. Jos käytettävyyssuunnittelu on menetelmänä laadukas, niin jokaisen ryhmän

- tulisi löytää samat ongelmat
- tulisi päätyä samoihin tuloksiin
- tulisi raportoida ne samalla tavoin
- ryhmät olisivat yhtä mieltä ongelmien vakavuudesta ja niiden luokittelusta

Asiaa on tutkittu tieteellisesti kansainvälisissä vertailututkimuksissa. Näiden vertailevien käy-

tettävyyssuunnittelun ("comparative usability evaluation, CUE") vetäjänä on toiminut tunnettu tanskalainen käytettävyyssuunnittelija Rolf Molich.

CUE -tutkimusten idea on siis se, että annetaan sama sovellus usealle eri käytettävyyssiimille itsenäisesti arvioitavaksi, ja verrataan saatuja tuloksia. Testien tekijät ovat olleet ammattimaisia käytettävyyssuunnittelijoita tai -ryhmiä: niin kaupallisia toimistoja kuin yliopiston käytettävyyssuunnittelijoita. Käytettävyyssiimien tekemät evaluoinnit olivat pääasiassa käytettävyyssuunnittelun; joissakin CUE -tutkimuksissa on tehty myös asiantuntija-arviointityyppisiä evaluointeja.

CUE -tutkimusten tuloksia voidaan pitää jokseenkin tyrmäyksenä: käytettävyyssuunnittelun tulokset erosivat huomasti riippuen siitä, kuka testejä oli tehnyt. Tiivistettynä

- ryhmät eivät raportoineet kuin muutaman saman ongelman (joissakin tapauksissa ei yhtään!)
- ryhmät löysivät enimmäkseen vain "omia" ongelmia, ts. ongelmia, joita kukaan muu ryhmä ei löytänyt

Kuvassa 1 on esitetty esimerkkejä CUE -tutkimusten tuloksista.

Lisäksi löytyi merkittäviä eroja testien suorittamisessa: siitä, miten testejä raportoitin, miten testitehtävät oli laadittu jne. Lisätietoja CUE -tutkimuksista löytyy mm. sivuilta www.dialogdesign.dk.

CUE -tutkimusten tuloksista on vaikea tehdä muuta johtopäätöstä, kuin että käytettävyyssuunnittelu menetelmänä ei ole – ainakaan tänä päivänä – tieteellisesti vahva. Voidaan puhua mieluummin kehikosta ("framework") kuin menetelmästä: on tiettyjä yhteisiä piirteitä – loppukäyttäjät mukana, annetut testitehtävät, jne. – mutta tarkemmalla tasolla testaus on tekijänsä näköinen.

Kaikkiaan, tulokset tarkoittavat esimerkiksi, että lausuma "tämä sovellus on käytettävyyssuunnittelun"

CUE-1

- 4 ryhmää testasi Windows Task Timerin
- Löytyi 141 ongelmaa, joita vain 1 sellainen ongelma, jonka kaikki ryhmät löysivät

CUE-2

- 9 käytettävyyssryhmää arvioi Microsoft Hotmailin
- Kaikkiaan löytyi 310 käytettävyysongelmaa
Ei yhtään sellaista ongelmaa, jonka kaikki ryhmät olisivat löytäneet
Vain 2 sellaista ongelmaa, jonka löysi vähintään 6 ryhmää
75% ongelmista oli sellaisia, jotka löysi vain yksi ryhmä
29 vakavista ongelmista sellaisia, jotka löysi vain yksi ryhmä

CUE-4

- 17 käytettävyyssryhmää testasi hotellin varausjärjestelmän
- Kaikkiaan löytyi 340 virhettä
Vain 9 virheistä sellaisia, jotka löysi enemmän kuin puolet ryhmistä
205 (60%) virheistä sellaisia, jotka raportoi vain yksi ryhmä
Näistä 61 kriittisiä tai vakavia

testattu” ei välttämättä kerro kovinkaan paljon. Testattu sovellus on luultavasti parempi kuin jos ei olisi testattu ollenkaan. Mutta auki jää, onko testauksen perusteella löydetty ja korjattu oleelliset ongelmat; puhumattakaan, että onko sovellus riittävän käytettävä.

Mitä käytettävyyssitestien tekijä voi tehdä?

Miten käytettävyyssitestien tekijä voisi osaltaan varmistaa, että hänen tekemänsä testit ovat laadukkaita?

Perusasia tietenkin olisi varmistaa, että omaa hyvät perustiedot käytettävyyden teorioista, käytettävyyden suunnittelun (”käyttäjakeskeisen suunnittelun”) prosesseista, eri käytettävyyssuunnittelun sekä käytettävyysohjeistoista ja -standardeista.

Tässä on kuitenkin haasteena se, että vaikka käytettävyyssuunnittelusta annetaan eri oppilaitoksissa, käytettävyyssuunnittelun ei ole olemassa yleisesti hyväksyttyä kriteeristöä (ns. ”body of knowledge” on kehityksen alla, mutta ei vielä olemassa; ks. http://www.upassoc.org/upa_projects/body_of_knowledge/bok.html). Periaatteessa siis kuka tahansa voi väittää olevansa käytettävyyssuunnittelija. Toisaalta kukaan ei oikeastaan voi väittää, että juuri hän on pätevä.

Toinen perusneuvo – joka myös esitetään CUE -tutkimusten suosituksissa – on se, että tulisi varmistaa ammattitaitoaan oppimalla toisilta käytettävyyssuunnittelijoilta. Oma kokemukseni on, että se, missä ja keneltä käytettävyyssuunnittelun on oppinut, vaikuttaa huomattavasti siihen, miten testauksia tekee. Alan julkaisujen seuraaminen sekä seminaareihin, luentoihin, työpajoihin ja konferensseihin osallistuminen olisi hyödyllistä – alan tutkimuksen ja julkaisujen tekemisestä puhumattakaan. Tai voi vaikka osallistua CUE -tyyppisiin tutkimuksiin, joissa pääsee vertaamaan omia käytäntöjään toisten käytäntöihin. Ammattimainen verkostoituminen yleensä on hyödyllistä; verkos-

tojen kautta kun on mahdollista luontevaan ajatusten ja kokemustenvaihtoon. Koska Suomessa piirit ovat pienet, kansainväliset verkostot ovat tärkeitä.

Yksi mahdollisuus on myös käyttää ulkopuolista käytettävyyssuunnittelijaa auditoimaan käytettävyyssuunnittelun käytäntönsä. Tätä kautta on mahdollisuus saada seikkaperäinen ”toinen näkökulma” omiin käytäntöihinsä. Tällainen ulkopuolinen näkemys on joka tapauksessa luultavasti vähintään ajatuksia tuulettava.

Mitä käytettävyyssitestien asiakas voi tehdä?

Jos CUE -tutkimusten tuloksia tarkastellaan käytettävyyssuunnittelun asiakkaan näkökulmasta, niin tilanne on tietenkin huolestuttava. On selvää, että jotkut testit ovat laadukkaampia kuin toiset. Mutta testien asiakkaalle voi olla haastavaa arvioida sitä, mikä testi on enemmän laadukas ja mikä vähemmän. Mistä tietää, että juuri minulle tehtävien testauksien ja niiden tulosten laatu on hyvää?

Yksi peruslähtökohta käytettävyyssuunnittelun tekijän valinnassa on testin tekijäehdokkaan käytettävyyshistorian läpikäynti. Tässä on huomattava, että kokemusta testien tekemisestä on vain osatotuus – voihan testauskäytännöt olla lähteneet väärille urille alusta lähtien. Erityisesti tulisi tarkastella sitä, mistä edellä oli puhetta: missä määrin tekijäkandidaatti on varmistanut ammattitaitoaan.

Toisena vaihtoehtona voi ajatella tarkistuslistaa, jonka avulla tarkistaa testin ammattimaisuutta:

- edustavatko testitehtävät reaali maailman tilanteita?
- ovatko tehtävät muotoiltu siten, että ne eivät sisällä vihjeitä?
- sisältääkö testiraportti oleelliset kohdat
- erotellaanko tuloksissa selkeästi faktat ja johtopäätökset
- jne.

Tällaisten arviointien tekeminen on kuitenkin haasteellista käytettävyyssalaa tuntemattomalle. Voi olla vaikea arvioida tekijän taustatietojen relevanttiutta. Käytettävyydestien muuttujia taas on paljon, eikä niiden perusteella tehty arviointi ole mekaanista.

Käytännössä asiakkaalla tulisi olla käytettävissä käytettävyyssiantuntijuutta, joka on riippumaton testien tekijästä. Tällainen asiantuntija voi auttaa joko suoraan arvioimaan käytettävyydestin ammattimaisuutta tai valitsemaan sopivaa käytettävyydestin tekijää.

Tilatako käytettävyydestejä ollenkaan?

On tietenkin tilanteita, joissa asiakas ei tilaa suoraan käytettävyydestä, vaan esimerkiksi konsultointityypistä palvelua siitä, mitä hänen kannattaisi käytettävyyden eteen yleensä tehdä (mistä aloittaa, mitä tehdä seuraavaksi jne.).

Tällaisessa tilanteessa eri käytettävyyssiantuntijoiden ehdottamat toimenpiteet luultavasti eroavat vielä enemmän kuin käytettävyydestäuksen suhteen. Tämä yksinkertaisesti siitä syystä, että käytettävyydestäus on kuitenkin ehkä perinteisin ja eniten käytetty yksittäinen menetelmä. Jos eri asiantuntijoiden näkemykset ja käytännöt eroavat tällaisessa perusasiassa, niin käsitykset luultavasti eroavat vielä enemmän silloin, kun puhutaan muista käytettävyyssuunnittelun aktiviteeteista.

Käytettävyydestäus ei todellakaan välttämättä aina juuri se oikea toimenpide. Jokin toinen käytettävyyssaktiviteetti voi olla kriittisempi, riippuen tilanteesta. Esimerkiksi strategisten ja operatiivisten käytettävyydestavoitteiden määrittäminen (ks. <http://www.joticon.fi/JFunnel.html>) yleensä ottaen olisi erittäin hyödyllistä käytettävyydestien perustaksi.

Lopuksi

Kaikesta tässä artikkelissa esitetystä kriittisestä arvioinnista huolimatta käytettävyydestäus on ehdottoman hyödyllinen ja suositeltava menetelmä. Mutta tulee vain ymmärtää ja huomioida menetelmään liittyvät rajoitukset. Ja se, että käytettävyyss varmistus ei ole ensi sijassa sen testusta.

Artikkelissa on korostettu, että käytettävyyssiantuntijoiden tulisi ottaa oppia toisilta. Niinpä itsekin otan mielelläni vastaan palautetta, jos artikkeli sellaisia herättää. Vaikkapa tuosta lopusta mainitusta käytettävyydestavoitteiden määrittämisen hyödyllisyydestä.

Hyvä tietotekniikan liiton jäsen!

Näin päivität tietosi

Voit päivittää jäsentietosi verkkosivuillamme www.ttlry.fi. Tietojen päivittämiseen tarvitset käyttäjätunnuksen (= jäsennumerosi, merkitty jäsenlehtiin) ja salasanasasi (= postinumerosi). Jos olet muuttanut salasanasasi tai kirjautuminen ei muutoin onnistu, voit lähettää tunnusten tarkistuspyynnön osoitteella jasenasiat@ttlry.fi.

Toivomme sinun erityisesti varmistavan, että sähköpostiosoitteesi jäsentiedoissa on oikea.



Henkilökohtaisempaa palvelua - Sinun eduksesi

Tietotekniikan liitto jäsenyhdistyksineen, osaamisyhteisöineen ja kerhoineen haluaa palvella jäseniään henkilökohtaisemmin ja paremmin, tarjota tietoa juuri Sinua kiinnostavista aiheista. Palvelun parantamiseksi olemme uusineet verkkopalvelumme.

Päivität vain tiedot itseäsi kiinnostavista aiheista ja saat tietoa juuri niistä. Voit päivittää valintasi aina halutessasi. Tietoja ei anneta ulkopuolisille tahoille vaan niitä käytetään ainoastaan TTL:n ja sen piirissä toimivien yhteisöjen tarkoituksiin.

Tietotekniikan liitto ry

Lars Sonckin kaari 12	www.ttlry.fi	jasenasiat@ttlry.fi
02600 Espoo	etunimi.sukunimi@ttlry.fi	p. 020 741 9898
		f. 020 741 9889

You need to be smart!

by Ivar Jacobson

One of the most popular movements in software development in recent years is the move toward agility. Today everyone wants to be agile. That is good! However, the essence of being agile is being smart. I have for several years expressed that the most important character you need to have to be a great software developer is to be smart. In several of my columns I have summarized what you need to be successful by saying: You need to be smart! What does that mean? Most people know intuitively what "being smart" means in everyday language, but what does it mean for software.

It is not smart to model everything in UML for instance when building software. It is not smart to model nothing and go straight to code. It is however smart to find exactly that something that is of importance to model and code.

Advice: What is that something? It is about the most essential use cases and in particular about the most essential scenarios through these use cases. It is about the components and in particular the parts of those components that realize these essential scenarios. Thus, it is about the essentials. Now you may ask what makes a scenario essential. An essential scenario is the response to the question: "what are the most important scenarios

the system is supposed to do". Which scenarios exercise the critical parts of the architecture? Which scenarios need to work in order for us to say that the highest technical risks have been eliminated?

It is not smart to write requirements without caring that these requirements are testable. It is smart to make sure the requirements also are test cases.

Advice: try use cases since they are also great test cases.

It is not smart to work in a waterfall manner, first specify all requirements, then do the design and the code, and finally test it all. If you do, you will discover serious problems with performance, architecture, usability too late. It is smart to first build a small skinny system and then build a little bit more, and a little bit more, before you release the system to customers. Each time you build something, you must be able to run, validate and test it.

Advice: use a controlled iterative approach such as the iteration practice in the Unified Process or sprints in scrum.

It is not smart to run off and build a lot of "stuff" before first assessing if you can source the whole or parts of the application (Open Source or commercial offerings).

It is not smart to develop software with a process that cannot scale if your system is successful and

customers want much more. It is smart to use a way of working that is no more than what you really need, but that can grow as you succeed with the product.

Advice: make sure your process has a dial for each interesting process parameter so you can turn the dial to the proper position for your project.

It is not smart to throw out your existing process and adopt a completely new process. That is doomed to fail in most cases. Not everything you did in the past was wrong so why should you start all over. It is smart to improve your process in small manageable steps.

Advice: add one practice at the time.

It is not smart to find a shortcut that in reality becomes a detour, such as skipping requirements and going straight to code. It is smart to do enough requirements to find what to use to build your first increment, your skinny system.

In general it is not smart to be extreme in what you do such as: model everything or model nothing, follow a strict waterfall process or an unstructured iterative approach, throw out what you have and start all over. It is smart to be balanced to do what is needed right now but with an eye to the future.

Above I have given a number of examples. In each case, there are some ideas on how to think about being smart. And each case can be expanded further. You will become smart with experience, but experience on the other hand is not a guarantee for being smart. Of course eventually, it comes back to you.

Smart is not the same thing as being intelligent. You can be intelligent without being smart. And you can be very smart without being very intelligent.

Smart is not the same as having common sense. You can have common sense without being smart, but if you are smart you must have common sense.

Smart is to do exactly right, not to find a broad solution that is just about right.

Let us all become smarter.



Tohtori Ivar Jacobson on ohjelmistomenetelmien vaikutusvaltaisimpia hahmoja, joka on ollut mukana kehittämässä monia ohjelmistoalaa muuttaneita menetelmiä. Näitä ovat: komponentit ja komponenttiarkkitehtuurit, käyttötapaukset, aspektisuunniteltu ohjelmistokehitys, Unified Modelling Language (UML) ja Rational Unified Process (RUP). Viime aikoina hän on keskittynyt käytäntöihin perustuvaan ohjelmistotuotantoon, jonka suosio kasvaa nyt ympäri maailmaa.

Ivar on ketterien (Agile) menetelmien vahva puolestapuhuja ja kehittää niitä fiksumiksi. Hänen mottonsa on "Kaiken tulisi tulla ketterämmäksi, mutta ketterä ei ole kaikki". Ivar on kirjoittanut kuusi suosittua kirjaa ohjelmistotuotannosta. Ivarin yrityksellä, "Ivar Jacobson International":lla on toimintaa Yhdysvalloissa, Iso-Britanniassa, Kiinassa, Singaporessa, Australiassa, Saksassa ja Ruotsissa.

Tästä Systeemyö-lehden numerosta alkaen Ivar kirjoittaa lehteemme kolumnia ohjelmistotuotannon muutoksen aallonharjalta, tervettä järkeä ja pitkää perspektiiviä unohtamatta.



Kuva 1. Ian McClellandin kertoi puheenvuorossaan miten käytettävyyden varmistamisesta ollaan siirtymässä käyttökokemuksen suunnitteluun.
Kuva: Matti Vuori, Cybercom Finland

Teksti: Eija Kaasinen

Navigaattori kourassa rotvallin reunalla – Maailman käytettävyyispäivä Tampereella



Eija Kaasinen toimii VTT:llä tutkimuskoordinaattorina vastuualueenään ihmisen ja tekniikan vuorovaikutukseen liittyvä tutkimus. VTT:n monitieteinen ihmisen ja tekniikan vuorovaikutustutkimus on integroitu osaksi teknologian sekä tietoyhteiskunnan tutkimista ja kehittämistä. Näin kehitettävistä ratkaisuista saadaan toimivia, käytettäviä sekä käyttäjien arvostamia.

Tampereella järjestettiin 13.11.2008 maailman käytettävyyispäivän seminaari sekä kaikille avoin yleisötilaisuus. Päivän teemana oli liikenne ja liikkumista tukevan tekniikan käytettävyys. Seminaari kokosi Mediamuseo Rupriikkiin ajatuksia vaihtamaan sekä käytettävyyden että tekniikan ammattilaisia niin teollisuudesta, julkishallinnosta kuin tutkimusmaailmastakin. Tapahtumien järjestelyistä vastasivat yhteistyössä VTT, Nokia, Plenware, Tampereen teknillinen yliopisto, Tampereen yliopisto sekä Systeemyöhydistyksen SYTYKE ry:n Käytettävyysosaamis-yhteisö.

Merkitys ja positiivinen käyttökokemus

Päivän pääpuhuja Ian McClelland on tehnyt pitkän uran Philipsillä Hollannissa. Hän painotti puheessaan että suunnittelussa on kiinnitettävä huomio sekä merkitykseen käyttäjälle (value) että merkitykseen liiketoiminnalle. Suunnittelun pitää tähdätä positiiviseen käyttökokemukseen sekä ensikäytössä että vakiintuneessa käytössä. Ian McClelland korosti että suunnittelutiimeissä on oltava tekniikan, käytön ja liiketoiminnan osaajia "from day one". Organisaation rakenteessa on tärkeä päätös kenen tontille kuuluu käyttökoke-

mus. Ian McClelland ehdotti että käyttökokemus voi olla samantasoinen organisaation osa kuin kehitys ja tuotesuunnittelu.

Englanninkielinen termi value taipuu kankeasti suomenkielelle. Se voi tarkoittaa myös käyttäjän arvoja, joista Sari Kujala Tampereen teknillisestä yliopistosta kertoi omassa puheenvuorossaan. Arvot ohjaavat vahvasti käyttäjän toimintaa mutta ne ovat hankalasti ilmaistavia ja tutkittavia. Kun arvot otetaan suunnittelun lähtökohdaksi, pystytään vastaamaan käyttäjän piileviinkin tarpeisiin paremmin. Suunnittelussa pitää päättää painotetaanko tiettyjä arvoja vai suunnitellaanko esimerkiksi liikenneratkaisuja sekä vapaan liikkumisen että vihreiden arvojen lähtökohdista.

Ihmiskeskeisen tutkimuksen ja opetuksen keskittymä Tampereelle

Professorit Veikko Surakka (Tampereen yliopisto) ja Kaisa Väänänen-Vainio-Mattila (Tampereen teknillinen yliopisto) kertoivat uudesta Yliopistoallianssista, joka on Jyväskylän, Tampereen teknillisen ja Tampereen yliopiston yhteistyökonsortio. Konsortion tarkoituksena on harjoittaa pitkälle menevää yhteistyötä ja työnjakoa koulutuksessa, tutkimuksessa ja tukipalveluissa. Keskeinen osa yhteistyötä on Inhimillisen tekno-

logian tutkimus- ja opetusallianssi, INTO. INTO pohjaa yliopistojen nykyiseen yhteistyöhön, jonka keskeinen lähtökohta on monitieteisen ihmisen käyttäytymistä huomioivan informaatio- ja kommunikatio-tekniikan tutkimus ja kehittäminen.

Käyttäjä suunnittelijaksi?

Pirkko Rämä VTT:ltä kertoi ajoneuvojen tieto- ja viestintäjärjestelmien käyttöliittymien suunnitteluperiaatteista. Suunnittelun ohjeiden lisäksi panostetaan autoilijoille suunnattuihin ohjeisiin laitteiden sijoittamisesta sekä kampanjaan hyvistä käytännöistä.

Jussi Kaasinen Nokialta kertoi kokemuksia käyttäjien osallistumisesta Nokia Sports Trackerin suunnitteluun. Alfa-vaiheessa suunnitteluun osallistui aktiivikäyttäjiä omasta yrityksestä, beta-vaiheessa myös ulkopuolisia urheilun harrastajia. Suunnittelu eteni itseohjautuvasti ja keskeinen elementti oli suora yhteys käyttäjän ja suunnittelijan välillä. Myös omassa esityksessäni tulevaisuuden ubi-ympäristöistä korostin käyttäjän muuttuvaa roolia tutkimuskohteesta aktiiviseksi suunnitteluosapuoleksi.

Esitykset kirvoittivat seminaarin loppuun vilkkaan keskustelun siitä pitääkö ja voiko käyttäjä osallistua suunnitteluun. Käyttäjävetoisin avoin innovaatio on vahvasti esillä esimerkiksi ICT-alan strategisen huippuosaamisen keskittymän (ICT SHOK) tutkimusohjelmissa. Käyttäjän osallistumista vastaan esitettiin neljä väitettä: 1) käyttäjä ei ole suunnittelun ammattilainen, 2) suunnittelu ei ole käyttäjän tehtävä, 3) vastuuta vaikeista päätöksistä ei voi siirtää käyttäjälle ja 4) käyttäjän aika on arvokasta. Käyttäjän osallistumista puoltavissa puheenvuoroissa korostettiin sitä että käyttäjä on oman arkensa ja järkevien työkäytäntöjen paras asiantuntija. Käyttäjällä on paljon piilotietoa jota ei saada esiin jos käyttäjää pyydetään kertomaan tarpeet puhtaalta pöydältä suunnittelun aluksi ja jos seuraava mahdollisuus vaikuttaa on esittää pieniä muutoksia jo toteutettuihin ratkaisuihin suunnittelun loppuvaiheessa.

Sovellusalueilla joilla löytyy innokkaita lead user käyttäjiä – esimerkiksi urheilusovellukset - käyttäjien osallistuminen on jo yleinen käytäntö ja se on johtanut hyviin tuloksiin. Toinen hyvä esimerkki on oman asuinalueen suunnittelu koska alueella asuvat ja liikkuvat ihmiset ovat parhaita asiantuntijoita kun vaikkapa alueen liikennejärjestelyjä muutetaan. Samoin työelämän soveluksissa käyttäjillä on työkäytäntöihin liittyvää asiantuntemusta, joka pitää ilman muuta saada suunnitteluun mukaan. Keskustelussa päädyttiin siihen että käyttäjien osallistuminen suunnitteluun täytyy aina miettiä tapauskohtaisesti. Osallistumis- ja vaikuttamismahdollisuuksia pitää olla mutta mitä ja missä muodossa riippuu siitä mitä ja kenelle suunnitellaan.

Miten voin osallistua oman liikenneympäristöni suunnitteluun?

Käytettävyysspäivän tavoitteena on viestittää käytettävyydestä myös tavallisille kuluttajille. Siksi järjestimme Käytettävyysspäivän iltana kaikille avoimen yleisötilaisuuden. Anne Vehmas Ramboll Finland Oy:stä kertoi miten itse kukin voi vaikuttaa oman liikenneympäristönsä suunnitteluun. Esityksen pääviesti oli että vaikutusmahdollisuuksia on mutta omaa aktiivisuutta tarvitaan jotta löytää tietoa valmisteilla olevista hankkeista ja jotta osaa kommentoida oikeaan aikaan. Yleisökeskustelussa päädyttiin ehdottamaan että kaupungeilla pitäisi olla kaupunginosakohtaiset sähköpostilistat, joille voisi rekisteröityä saadakseen tietoa omaa asuinalueen koskevista suunnitelmista riippumatta siitä minkä hallinnon alan asiasta on kyse.

Yleisötilaisuudessa esiteltiin nykyisiä ja tulevia teknisiä ratkaisuja liikkumisen avuksi. Yleisö sai myös tutustua käytännössä navigaattorin avulla harrastettavaan aarteensintään - geokätköilyyn - harrastajien opastuksella.



Kuva 2. Yleisöllä oli paljon kommentoitavaa liikkumisympäristön suunnittelusta ja siihen vaikuttamisesta. Kuva: Stenka Mikael Christiansen, Studio Christiansen Oy.



Kirjan rakenne:

- **I Johdanto**
 - o Verkkopalveluiden käyttäjäkeskeisyys
 - o II Vaatimusmäärittelyn käyttäjäkeskeiset menetelmät
 - o Käyttäjätutkimuksen tekeminen
 - o Käyttäjätutkimuksen analysointi ja tulosten tiivistys
 - o Käsitteet ja terminologia
- **III Verkkopalvelun suunnittelu**
 - o Verkkopalvelun toiminnallisuuden suunnittelu
 - o Järjestelmän rakenteen suunnittelu
 - o Rautalankamalleihin prototyyppien kautta iteroiden
 - o Käyttöliittymän elementit
 - o Visuaalinen suunnittelu
 - o Sisällön tuottaminen
- **IV Käytettävyyden tarkastaminen**
 - o Käytettävyyden evaluointi
 - o Käytettävyydesti

Teksti: Helena Venäläinen

Laatua verkkopalveluihin käyttäjäkeskeisten menetelmien avulla

Kirjan esittely:

**Irmeli Sinkkonen, Esko Nuutila, Seppo Törmä (2008).
Helppokäyttöisen verkkopalvelun suunnittelu. Tietosanoma.**

FM Helena Venäläinen on tietojenkäsittelyn ammattilainen ja tehnyt pitkän uran eri rooleissa osuuspankkien tietojärjestelmien kehittäjänä. Hän on laajentanut osaamistaan viimeisen kymmenen vuoden aikana myös kognitiiviseen ja kasvatustieteen suuntaan. Hän toimii tällä hetkellä OP-Keskuksen ICT -organisaatiossa osaamisen ja menetelmien kehittämissä tehtävissä.

Systeemyöhydistys
Sytyke ry:n johtokunnassa hän toimi vuodesta 2001 alkaen, puheenjohtajana vuosina 2004- 2006. Hän on ollut perustamassa Käytettävyyttä OSYä sekä toiminut sen vetäjänä vuosina 2007 - 2008.

Laadukkaan verkkopalvelun rakentaminen ei ole helppoa. Esiteltävä kirja antaa ohjeita siihen, kuinka laadukas verkkopalvelu rakennetaan käyttäjälähtöisesti.

Kenelle kirja on tarkoitettu

Kirja on ensisijaisesti suunnattu niille yrityksissä työskenteleville henkilöille, jotka ovat vastuussa hyvien verkkosivujen tuottamisesta ja joilla ei välttämättä ole IT-taustaa. Kirja sopii myös seuraaville kohderyhmille:

- IT- tai systeemyön ammattilaiset, jotka eivät vielä tunne käytettävyyden menetelmiä
- käytettävyyden opiskelijat
- käytettävyyssihmiset, vaikka sisältö lieneekin heille tuttua
- tuotteiden ostajat, jotta he osaavat ostaa käytettävyyttä tulematta huijatuiksi.

Kirjan sisältö

Helppokäyttöisen verkkopalvelun suunnittelu -kirja esittelee lähes 350 sivullaan verkkopalvelun käyttöliittymän suunnittelussa käytettäviä käyttäjäkeskeisiä menetelmiä sekä verkkopalvelun suunnittelun vaiheet. Kirja alkaa esipuheella ja kirjoittajien esittelyllä. Kirjan johdannon jälkeen esitellään vaatimusmäärittelyn käyttäjäkeskeiset menetelmät sitten verkkopalvelun suunnittelu ja siinä käytettävät menetelmät ja lopuksi käytettävyyden tarkastamisen menetelmät. Menetelmien soveltamista ja verkkopalvelun syntymistä havainnollistetaan kirjassa esimerkkijärjestelmien avulla: toinen on PKT-säätiön todellisuudessa tuottama, konsultteja välittävä verkkopalvelu A-pooli yksinkertaistettuna ja toinen on kuvitteellinen, koiraharrastajille tarkoitettu verkkopalvelu Koiraviisas.

Johdanto

Johdannossa esitellään verkkopalveluiden käyttäjäkeskeisyyden taustoja ja avataan hiukan käsitteitä käytettävyys ja käyttäjäkokemus. Samoin esitellään lyhyesti erityyppiset verkkopalvelut ja niiden luokittelut. Kirja antaa selkeän kuvan - mm. 20 hyvää syytä - käyttäjäkeskeisen suunnittelun hyödyistä. Johdannossa on esitelty lyhyesti kaksivaiheinen tuotekehitysprosessi verkkopalvelun rakentamiseen sekä verkkopalvelun suunnittelun periaatteet, niiden toteuttamisessa hyödyksi olevat menetelmät sekä tiivistelmä suunnitteluperiaatteista. Liiketoiminnan tavoitteita, vaatimusmäärittelyä ja suunnitteluprosessia on avattu lyhyesti.

Johdannon lopussa on vinkkejä siitä, kuinka käyttäjäkeskeinen verkkopalveluprosessi voitaisiin liittää osaksi yrityksen käyttämää suunnittelumenetelmää, kuten vesiputousmalli, RUP (Rational Unified Process) tai joku ketterä menetelmä. Kirjassa kuvattujen menetelmien ei siis ole tarkoitus korvata nyt käytössä olevia, vaan tuoda rinnalle lisää menetelmiä lopputuloksen laadun varmistamiseksi.

Vaatimusmäärittelyn käyttäjäkeskeiset menetelmät

Kirjan vaatimusmäärittely-osuus keskittyy käyttäjä- ja käytettävyysvaatimuksiin ja käyttöliittymän tietovaatimuksiin. Tässä osassa esitellään myös käyttäjien ryhmittelytavat ja käyttäjäprofiilit esimerkkien kanssa. Kirjan mukaan on liiketoiminnan tarpeiden ja käyttäjien tarpeiden on kohdattava, jos halutaan tuottaa onnistunut palvelu.

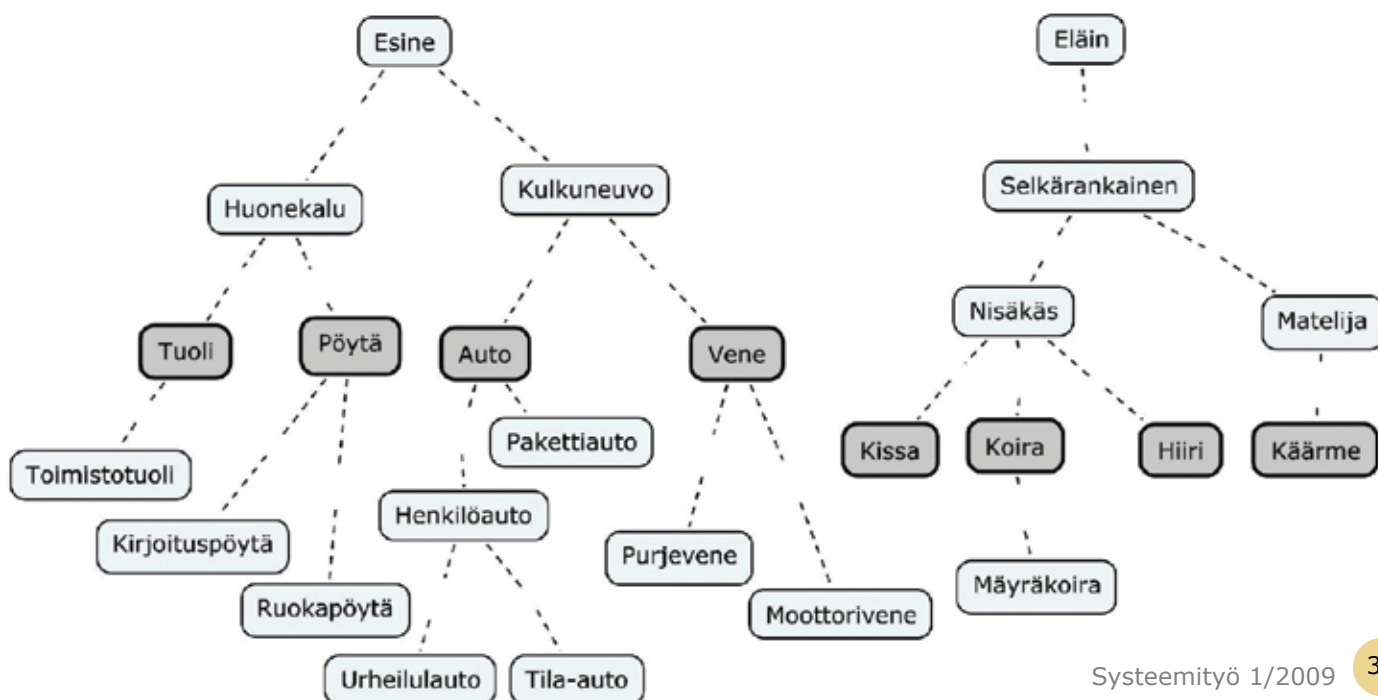
Kaikki käyttäjätutkimuksen tiedonkeruu- ja analysointimenetelmät on esitelty kirjassa mielestäni riittävän seikkaperäisesti, jotta halukkaat voivat ryhtyä niitä soveltamaan ainakin pienemmissä hankkeissa sekä hankki-
maan niistä lisätietoa. Tämä osa kirjasta tuo mielestäni hyödyllistä uutta vaatimusmäärittelyyn, jonka problematiikka saattaa keskittyä käyttäjävaatimusten selvittämisen kustannuksella liiketoiminnallisten ja tietojärjestelmän vaatimusten kuvaamiseen ja hallintaan. Vähäinen ei ole myöskään se kirjassa esitetty näkemys, että projektit pitäisi organisoida siten, että niihin aina kuuluu käyttäjätutkimus.

Kirjasta löytyy mm. seuraavia ohjeita:

- verkkopalvelun strategian miettiminen, jolloin kirkastetaan se, mitä hyötyjä verkkopalvelulla tavoitellaan
- vanhan tuotteen tilan inventointi, tähän kirja antaa perustelujen lisäksi käytännön ohjeita
- kilpailijavertailun tekeminen
- käyttäjien alustavan tarveanalyysin tekeminen työpajojen avulla.

Kirjan mukaan käyttäjätutkimuksen tavoitteena on hankkia palvelun rakentamisen perustaksi tietoa käyttäjistä sekä mm. heidän tavoitteistaan ja tehtävistään. Käyttäjätutkimukset voivat olla kirjan mukaan palvelusta riippuen esimerkiksi lyhyitä ryhmä- tai yksilöhaastatteluja, kyselyitä, pitkäkestoista toiminnan seuraamista tai käyttäjien itse kirjoittamia tarinoita, päiväkirjoja tai muita luotaimia. Myös simultaatioilla ja rooli-
leikeillä voidaan hankkia tietoa käyttötilanteista. Kirja opastaa käyttäjätutkimukseen valmistautumiseen, sen läpivientiin erilaisilla menetelmillä sekä tulosten analysointiin esim. samankaltaisuusanalyysin avulla ja tiivistämiseen taulukoiksi tai persoonien kuvauksiksi sekä toimintatarinoiksi ja skenaarioiksi (taulu 1).

Taulu 1.



Kirjan käsitteet ja terminologia -osuudessa pyritään ratkomaan ongelmia, joita verkkopalvelussa esiintyvät käyttäjille tuntemattomat käsitteet ja termit aiheuttavat. Käsitteiden merkitys ja suhteet esitellään samoin käsitteiden muodostamat hierarkiat. Kirjassa on selvitetty, miten verkkopalvelun ja käyttäjien käsitteistöjen erot sekä heidän erilaiset tavoitteet verkkopalvelun käytössä vaikeuttavat palvelun oppimista ja sen ymmärtämistä. Kirja toteaa meillä systeemityöläisille tutun UML:n (Unified Modeling Language) tai OWL:n (Web Ontology Language) rajoitukset käsitteiden kommunikoimiseen käyttäjien kanssa. Tilalle tarjotaan käsitekarttoja (Concept Map), joiden tekemiseen on tarjolla myös ilmainen työkalu (Cmap Tools). Kirja esittelee käyttäjälähtöisen käsitteistöön vaiheet ja menetelmät melko lyhyesti ja osan lopussa esitelläänkin käsiteanalyysikirjallisuutta lisätiedon hankkimiseksi.

Verkkopalvelun suunnittelu

Kirjan verkkopalvelun suunnittelu -osassa neuvotaan ensin verkkopalvelun toiminnallisuuden suunnittelua käyttötarinoiden tai kuvatarinoiden avulla. Niiden avulla kerrotaan, miten käyttäjät tulevat toimimaan uudella palvelulla. Käyttötarinoita hyödynnetään palvelun informaatioarkkitehtuurin ja yksittäisten työprosessien suunnittelussa. Kirjassa mainitaan lyhyesti suunnittelumenetelminä myös vuokaaviot, miellekartat ja systeemityöläisille ennestään tutut käyttötapaukset, jotka tehdään toteutusta varten yleistämällä ja yhdistämällä yksittäisiä käyttötarinoita.

Seuraavaksi kirja ohjeistaa, kuinka verkkosivuston tai verkkosovelluksen rakenne ja informaatioarkkitehtuuri suunnitellaan. Kirja esittelee hierarkisen perusrakenteen lisäksi muutaman muunkin rakenteen sekä informaatioarkkitehtuurin suunnitteluun sopivan korttilajittelutekniikan. Sitten kirja esittelee muutaman sivun verran, kuinka tehdään rautalankamallien - so. toiminnallisuuden kuvausten - työstäminen iteroiden prototyyppien avulla. Käyttöliittymän elementti-osassa on ohjeistettu mm. navigoinnin, linkkien ja hakutoimintojen käyttöä sekä verkkolomakkeiden ja niihin liittyvien kontrollien suunnittelua.

Seuraava luku käsittelee visuaalista suunnittelua, joka on tyypillisesti ostopalvelu, sillä työ edellyttää graafikon tai käyttökokemusasiantuntijan taitoja. Luku melko lyhyt ja se on kirjoitettu myös palvelun ostajan näkökulmasta.

Verkkopalvelussa tarvitaan myös sisältöä, informaatiota tai palvelua. Verkkopalvelun tietosisällön tuottamis-osassa on ohjeistettu verkkotekstin kirjoittamista sekä tekstiä että kirjoittamisprosessin näkökulmista mielestäni hyvinkin kattavasti.

Käytettävyyden tarkastaminen

Kirjassa esitellään ilman käyttäjiä toimivia arviointimenetelmiä sekä testikäyttäjiä vaativia testausmenetelmiä. Ensimmäisenä on kuvattu heuristiseen arviointiin ja tarkistuslistoihin perustuva arviointi ja esitelty joukko hyödyllisiä tarkistuslistoja. Asiantuntija-arvioinnin luku on lyhyt, mutta käytettävyydestä on asiaa parikymmentä sivua. Kirjassa on mm. ohjeistettu pienimuotoisen, ilman laboratoriota läpivietävä käytettävyydestä, jota jokainen käyttöliittymän suunnittelijan pitäisi käyttää omassa työssään apuna. Käytettävyyden paras oppi on seurata oman tuotteen käytettävyydestä, jolloin saattaa joutua huomaamaan, että käyttäjät eivät osaa käyttää sitä.



Yhteenveto

Valtaosa kirjan menetelmistä on ihmistieteiden ja käytettävyyden tuntemia ja koestettuja. Uutta on niiden esittely suomeksi verkkopalvelun suunnitteluun liitettynä ja selkeiden esimerkkien avulla valaistuna. Kirjan kattama alue on rajauksista huolimatta laaja, jolloin asioissa ei voida mennä tarkalle tasolle eikä yksityiskohtiin. Kirja on miestäni mainio perusteos ja erinomainen reseptikokoelma sovellettavaksi ja muunneltavaksi kaikille systeemityöläisille, jotka haluavat laajentaa osaamistaan ja kehittää parempia verkkopalveluita käyttäjäkeskeisin menetelmin. Kurseille ei ole useinkaan aikaa mennä, jolloin asiaa voi edistää eteenpäin hyvä kirja, josta helposti näkee, että tässä kohtaa voi tehdä näin tai tässä kohtaa pitää ymmärtää tämä asia.

Systeemityölehden lukijoilla saattaa jo olla osaamista käyttäjäkeskeisistä menetelmistä, mutta koska ne kuitenkin ovat alallamme uusia tulokkaita, uskon tästä monipuolisesta kirjasta löytyvän kokeneellekin systeemityön ammattilaiselle uutta ajateltavaa ja sovellettavaa.

Kirjan hinta kustantajalla on 62 euroa, kustantaja antaa siitä alennusta oppilaitoksille. Irmeli on luvannut kirjasta alennusta kaikille Käytettävyyden OSYn jäsenille, alennuksen saa tilaamalla kirjan Irmeliltä (irmeli.sinkkonen@adage.fi). Nyt kaikki kipin kapin kirjaostoksille!

Comic Strip Dialogue:

- Panel 1:** Hauska haukotteleva koira etusivulla, se taitaa olla näyttelyssä odottamassa kehään pääsyä
- Panel 2:** Näyttelyhaku, Säännöt, Vinkit, Junior handler, Tuomarit...Hm.... katsotaanpa Näyttelyhakua
- Panel 3:** Valikossa lukee: "Näyttelyt", klikkaanpa sitä
- Panel 4:** Voin uudella järjestelmällä hakea näyttelyä koiran roturyhmällä, rodulla, koiran nimellä, kennelnimellä, syntymäajalla, tuomarin nimellä, näyttelyajankohdalla näyttelypaikkakunnalla, ja mitä useampia ehtoja annan, sitä lyhyemmän listan saan tulokseksi
- Panel 5:** No nyt voin katsoa tuosta lähellä olevat loppukesän näyttelypaikat
- Panel 6:** Ja tuloslistasta voin vielä karsia pois ne, näyttelyt, jotka eivät kiinnosta
- Panel 7:** Voin myös ilmoittautua suoraan palvelusta valitsemini näyttelyihin

Table shown in Panel 5:

Kmmj Kjkaj	KmKjkaj	4.6.09
Kj Kjkaj	KjKjkaj	4.6.09
Kmmj Kja	Kja	7.6.09
Kmmj Kjkaj	Kmmaj	12.7.09
Km Kjkaj	KmKjkaj	21.07.09



Laatua laadun päälle

Istuttuani parisen viikkoa jälkimakustelemassa edellisen artikkelin visuaalista lopputulosta, syöksähti päätoimittaja koputtamatta sisään. Syöksähdysten yhteydessä toimistoni ovi kolahti seinään ja väliin jäänyt muovinen roskaämpäri lähti paineen vaikutuksesta lentoon kieppuen akselinsa ympäri. Roskiksen sisältö: banaaninkuoret, teepussit, energiajuomatölkit ja tyhjät sipsipussit hajaantuivat muuten niin geometrisen täsmällisesti asettuneen huoneeni harmoniaan. Ja jotenkin katosivat siihen.

”Seuraava teema: Laadunvarmistus”, näin luki siinä paperilappusessa, jonka päätoimittaja roskiksen ja sen sisällön lentoratoja hämmästyttävästi väistellen pamautti pöydälleni, sanomatta mitään, häipyen yhtä fortissimo. Asia selvä, käsittelemme laadunvarmistusta.

Mutta katsotaanpa asiaa hieman kauempaa. En aio syöksyä opettamaan arvoisille ja tietoteknisesti järkyttävän ammattimaisille lukijoilleni laadunvarmistuksen eri aspektoja, sen tekevät muut tässä lehdessä. Voin toki tulla yrityksiinne luennoimaan tästäkin aiheesta sivuverokortilla ja autan mielelläni yleisöä löytämään sisäisen laatusankarinsa. Taksani ovat edullisia.

Varmistushan sanana viittaa siihen, että jonkin asian olemassaolo voidaan epäilyksettä todeta. Mutta tässä on pieni ongelma. Nimittäin, otetaanpa esimerkiksi lähes samanlainen sana ”turvavarmistus”. Jos jossain vuoristokiiveskelyliivin koukkukiinnikkeessä on valmistajan mukaan turvavarmistus, niin oletko valmis hyppäämään sata kertaa tyhjän päälle kaiken painosi tuon turvavarmistuksella varustetun kiinnikkeen varaan laitaen? Jos kiipeilyliivi muuten vaikuttaisi huteralta ja maksaisi todella vähän? Ja vaihtaa aina uuden huteran liivin päällesi? Jossain vaiheessa saattaisit pukea sellaisen, jossa turvavarmistus kuitenkin pettäisi. Tai jossain vaiheessa saattaisit pukea sellaisen, joka repeäisi kappaleiksi kaikkialta muualta paitsi siitä turvavarmistetusta kiinnikkeestä.

No, turvavarmistus siinä kiinnikkeessä eittämättä olisi. Mutta olisiko se kiinnike siltikään turvallinen? Entä olisiko se koko liivi turvallinen?

Jonkin verran turvaa aivan varmasti olisi, mutta kun turva ei sinänsä ole mikään yksiselitteinen mittari. Laatukaan ei ole.

Laatu taas voidaan karkeasti ottaen jakaa kahteen pääluokkaan: hyvä laatu ja huono laatu. Huonolaatuisia asioita sanotaan usein laaduttomiksi ja hyvälaatuisia laadukkaiksi, vaikkakin huonolaatuisissakin sitä laatua on paljon, se vain on sitä huonompaa laatua. Laadunvarmistus taas mielletään yksinomaan hyvään laatuun. Jos ette tätä jo tiedneet. Mutta takaako laadunvarmistus sitten laatua? Ja kumpaa laatua laatua? Laadunvarmistuksen laatu nimittäin onkin sitten jo hyvin kummallinen asia. Vähän niinkuin turvavarmistuksen turvakin. Laadunvarmistus voidaan tehdä laadulla, joko hyvällä laadulla tai huonolla laadulla. Tuotteen laadunvarmistuksen laadun laadukkuuden varmistaminen voikin olla seuraava suuri hypetettävä tietotekniikan trendi.

Hetkinen. Jostain vilkahti ajatuksiini jotain samanlaista. Eikös eräissä ideologisesti hyvin vahvoissa maissa ollut jonkinlainen tuollainen systeemi? Jokaisella oli valvoja, jolla oli valvoja, jolla oli valvoja ja niin edelleen. Siellä vain ei tiedetty, että kuka kenenkin valvoja oli. Valvonta oli joka tapauksessa hyvin korkealla tasolla, joskaan senkään laadukkuus ei sitten tainnut pidemmän päälle toimia. Mutta laadunvarmistuksen laadun laadukkuuden varmistamisessa voisimme ottaa parhaat asiat tuostakin mukaan. Esimerkiksi voisimme perustaa Laadunvarmistuksen Laadun Laadukkuuden Varmistamisen Keskuskomitean Politbyroon!

Tää on viimeinen testi,
laatumme yhtykää
niin huomispäivänä softat
niin virheettä pyörivääää!!

Krhmm.. Anteeksi. Ei saisi innostua ihan noin paljoa, päätoimittajakin on havainnut ylitsevuotavaisen pursuiluni ja maailmanmestaramaisesti pystyy aivan totisena kysymään, että onko lääkitykseni kunnossa. Edelleenkin ihmettelen sitä pokerinaamataitoa. No, joka tapauksessa, pitääkää hyvän laadun lippu korkealla. Vappunahan voisimme järjestää vaikka pienen lippu- ja banderollisaattueen, joka näin kunnoittaisi virheettömien ja oikeamielisten ohjelmistojen laadunvarmistajia, noita pyyteettömästi paremman maailman puolesta taistelijoita, jotka itsestään ja hyvinvoinnistaan välittämättä ovat menneinä vuosikymmeninä uhrautuneet virheitä viliseviä ohjelmistoja vastaan käydyssä suuressa laadunvarmistuksellisessa sodassa, jossa laatua riistävät koodiongelmien on saatu vähennettyä, jotta meillä kaikilla olisi mahdollisuus tasa-arvoiseen ja laadukkaaseen elämään. Rohkeasti he ovat käyneet vaikeuksia päin aseinaan vain katselmointimenettelyt, auditoinnit ja testaus. Aika velikultia. Siskoja unohtamatta.

SYTYKE ry on vuodesta 1979 toiminut valtakunnallinen systeemityöntekijöiden ammatillinen yhdistys, joka kehittää alan ammattilaisten välistä yhteistyötä ja tutkimustoimintaa.

Teemayhdistyksen jäseneksi voivat liittyä kaikki systeemityöstä kiinnostuneet yksityiset henkilöt, yhdistykset ja yritykset. SYTYKE ry:n toiminta-alueena on koko Suomi. SYTYKE on Tietotekniikan liitto Ry:n jäsenyhdistys.

Lisätietoja SYTYKE ry:stä: www.sytyke.org

TOIMISTO

Susanna Koskinen
Systeemityöyhdistys Sytyke ry
Talvikkitie 40 A 33
01300 Vantaa
p. 09 56075363
f. 09 56075365
sytyke@hennax.fi

Hallitus 2009

Puheenjohtaja Mitro Kivinen

p. +358 40 589 2724
mitro.kivinen@kolumbus.fi

Seppo Takanen

p. +358 50 581 0140
seppo.takanen@codebakers.fi

Marianne Malila

malila68@gmail.com

Juha Jääskinen

juha.jaaskinen@digia.com

Timo Sundman

timo.sundman@tietoenator.com

Tiina Kiuru

tiina.kiuru@ri.fi

Jyri Partanen

jyri.partanen@sulake.com

Varajäsenet

Ilkka Pirttimaa

p. +358 50 389 0022
ilkka.pirttimaa@stockmann.fi

Altti Lagstedt

altti.lagstedt@haaga-helia.fi

Liittokokous- edustajat

Minna Oksanen

minna.oksanen@gmail.com

Seppo Takanen

seppo.takanen@codebakers.fi

Lauri Laitinen

lauri.laitinen@nokia.com

SYTYKE ry:n hallituksen sähköpostilista:

hallitus@sytyke.ttlry.fi

Sytyttääkö? - Liity jäseneksi



Systeemityöyhdistyksen jäseneksi liitytään Tietotekniikan liiton kautta (<http://www.ttlry.fi/>, 020 741 9898, jasenasiat@ttlry.fi) valitsemalla jäsenyhdistykseksi Systeemityöyhdistys ry. Nykyinen Tietotekniikan liiton jäsen voi liittyä joko vaihtamalla jäsenyhdistystä tai liittymällä lisäjäseneksi.

Tietotekniikan liiton henkilöjäsenmaksu vuonna 2009 on alkaen 49€ ilman lehtiä, 1 lehti 63€. Eri-tyisryhmien hinnoittelusta lisätietoja Tietotekniikan liitosta. Lisäjäsenyys maksaa 12€/yhdistys.

Osaamisyhteisöt 2008

Systeemityöyhdistyksessä toimitaan niin yhdistystasolla kuin aihepiireittäin erikoistuneissa osaamisyhteisöissä. Monipuolisessa tarjonnassamme löytyy jokaiselle jotakin. Vaihtoehtona on myös perustaa omalle kiinnostukselleen uusi osaamisyhteisö - SYTYKE-hallitus toivottaa toimintaehdotukset tervetulleeksi. Osaamisyhteisön toimintaan pääset mukaan liittämällä postia vetäjälle.

JavaSIG - on Javan käyttäjien ja harrastajien intresiryhmä, vetäjänä Simo Vuorinen.
simo.vuorinen@oracle.com

ProjektiOSY - PrOSY pyrkii yhdistämään Systeemityön projektitoiminnasta ja sen kehittämisestä kiinnostuneet, vetäjänä Petteri Puurunen.
petteri.puurunen@tieto.com

TestausOSY - FAST on testauksen keskustelu- ja yhteistyöverkosto, vetäjänä Maaret Pyhäjärvi.
maaret.pyhajarvi@iki.fi

DAMA Finland keskittyy tiedon, informaation ja tietämyksen hallintaan, yhteyshenkilönä Minna Oksanen.
minna.oksanen@gmail.com

ViestintäOSY järjestää yhteistoimintaa viestintäsovellusten alueella, vetäjänä Tapani Ranta.
tapani.ranta@generum.fi

RELA keskittyy relaatiotietokantoihin vetäjänään Lauri Pietarinen.
lauri.pietarinen@relational-consulting.com

MallinnusOSY jakaa tietoa tietojärjestelmien mallintamisesta, vetäjänä Juha Jääskinen.
juha.jaaskinen@digia.com

KäytettävyyösY vaihtaa kokemuksia käytettävyyden kehittämiskeinoista sekä kehittää omaa käytettävyysosaamistaan. Vetäjänä ja yhteyshenkilönä Helena Venäläinen.
helena.venalainen@op.fi

SOA SIG on keskittynyt palveluarkkitehtuuriin (Service-Oriented Architecture, SOA). Vetäjänä Janne J. Korhonen.
jjk@jannekorhonen.fi

KÄYTTÖKELPOISTA KETTERÄSTI

Kun on saatava valmista pian, ei ole varaa virheisiin. On toimittava heti, jotta tulos saavutetaan. Tuloksen on oltava toimiva ja käytettävä. Työn on oltava laadukasta ja turha tekeminen on karsittava. Vaatimukset ovat kovat. Onko sinulla varaa olla valmentautumatta? Varmista valmiutesi vastata vaatimuksiin, tule oppimaan!

PROSESSIT KUNTOON

Liiketoimintaprosessien kehittäminen	26.-27.3.
ARIS prosessien mallintamisessa	23.-24.3.
BPMN prosessien kuvaamisessa	7.4. ^{Oulu} • 4.6. ^{Hki}
Microsoft BizTalk Server 2006 -sovelluskehitys (2933)	23.-27.3.
Tietojärjestelmän tehokas käyttöönotto	23.3.
CRM mitattavassa liiketoiminnassa	16.-17.3.

PROJEKTIT KETTERIKSI

Certified ScrumMaster	21.-22.4.
Certified Scrum Product Owner	23.-24.4.
Ketterän ohjelmistoprojektin läpivienti	23.-25.3.
Windows Communication Foundation (WCF)	23.-25.3.

LAATU LUJAKSI

Laatukäytännöt ohjelmistotuotannossa	18.-19.5.
Määrittely ja suunnittelu (UML 2.1)	15.-17.4.
Vaatimusten määrittely ja hallinta	21.-23.4.
Käyttöliittymän suunnittelu	22.-23.4.

TESTAUS TAVAKSI

ISEB ISTQB Foundation Certificate -valmennusohjelma	30.3.-2.4. ^{Tre} • 20.-23.4. ^{Hki}
Testauksen valmennusohjelma	23.-25.3. ^{Hki} • 4.-6.5. ^{Tre}
Test Driven Development Workshop (TDD, .NET/Java)	8.-9.4.

ARKKITEHTUURI AJANMUKAISEKSI

Tietojärjestelmäarkkitehdin valmennusohjelma	27.4. alk.
Järjestelmäintegroinnin valmennusohjelma	30.3. alk.
Microsoft BizTalk Server 2006 käyttöönotto ja hallinta (2934)	2.-3.4.
SOA ja Web Services -yleiskatsaus	27.3.

Ole aikaasi edellä! | www.tieturi.fi.

puh. 09 4315 5333 | kurssit@tieturi.fi | Helsinki | Turku | Tampere

OLE
AIKAASI
EDELLÄ

 **tieturi**
www.tieturi.fi

Taste our new
Sex on
the beach
Cocktail