

SYSTEEMITYÖ

Systeemityöyhdistys SYTYKE ry:n jäsenlehti N:o 3/2008

**Innovaatioita ja
massatuotantoa**

SAMCOM – LUOTETTAVA KUMPPANI



Samcom tuottaa korkeatasoisia tietojärjestelmäratkaisuita ja -palveluja, joiden avulla asiakkaamme kehittävät liiketoimintaansa ja parantavat kilpailukykyään. Samcom yhdistää alan parhaan osaamisen ja parhaat teknologiaratkaisut takaamaan hankkeiden pysymisen budjetissa ja aikataulussa.

Tutustu Samcomiin osoitteessa www.samcom.fi

Merkitse kalenteriin
Systeemityöyhdistyksen syyskokous
ke 19.11. klo 16.00 alkaen

Paikka ilmoitetaan myöhemmin.

Tervetuloa!

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.

Tietotekniikan liitto ry

Lars Sonckin kaari 12
02600 Espoo

www.ttlry.fi
etunimi.sukunimi@ttlry.fi

jasenasiat@ttlry.fi
p. 020 741 9898
f. 020 741 9889



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.

Julkaisija

Systeemityöyhdystys 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
sb@cabaroo.com

Toimituskunta 3/2008

Lea Virtanen
lea.virtanen@jobit.fi

Lisätietoja lehdestä

www.sytyke.org/lehti

Tilaukset

Systeemityölehti sisältyy yhdistyksen
Tietotekniikan liiton suositusten
mukaiseen yhdistyksen jäsenmaksuun.
Vuositilaukset 30 €
Irtonumerot 8 €

Hyvissä ajoin ennen painatusta tehty
vähintään 50 kpl lisätilaukset 2 €/kpl.
Tilaukset yhdistyksen toimistosta.

Kansikuva

Katja Tamminen

Seuraava numero

4/2008 V. 2009 Systeemityöyhdystys 30 v.
Toimituskunta:
Hallitus
Ilmestyy: to 18.12.2008

Painopaikka

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

Painos: 2500 kpl
ISSN 1237-0525
15 vuosikerta, nro. 3

Ilmoitushinnat

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

Arvonlisävero 0%
Vakiopalkan vähintään vuodeksi
varanneille 20% alennus.

Teksti: Minna Oksanen, lehden päätoimittaja, BI/DW asiantuntija
Digiaassa ja DAMA Finlandin perustajajäsen

Pääkirjoitus:

Pitääkö pyörä keksiä uudelleen?

Pyörän keksiminen on jo niin vanha klisee, että esitellään tässä toinen innovaatio, johon tutustuimme Sytykeen Innovaatioseminaarissa ja josta lehdessä 4/08 kirjoitetaan tarkemmin. Idea on USB-porttiin laitettava tikku nimeltään YUBIKEY, joka toimii fyysisenä kertakirjautumislaitteena, antaen joka kerta yksilöllisen tunnisteen. Erikoiseksi keksinnön tekee se, että laite toimii USB-näppäimistönä, joten se ei tarvitse koneeseen mitään ajureita eikä toisaalta tarvitse paristoja. Toinen erottava tekijä muista kertakirjautumislaitteista (esim. SecurID) on se, että kaikki laitteeseen tarvittavat ohjelmistot on ilmaisia. Businessmallina on siis myydä vain edullisia tunnistuslaitteita ja sen jälkeen sen käytöstä ei tarvitse maksaa lisenssimaksuja.

Samaisessa seminaarissa sovittiin, että hallituksen wikisivuilla testataan syksyn aikana YUBIKEY:n käyttöä ja se raportoidaan tulevissa lehdissä.

Innovaation tuoteistamisessa tärkeää on myös sen toistettavuus. Massatuotannossa tärkeää on prosessin tehokkuus, jotta tuote, palvelu kuten myös ict-ratkaisu saadaan tuotettua mahdollisimman tehokkaasti.

Mitä innovointi vaatii?

Vaatiiko se sopivasti ns. luppoaikaa, jotta aivot voivat työskennellä vapaasti. Miten tämä on mahdollista hektisen työtahdin vaivaamalla alallamme - ei ole lupaa laiskotella. Syntyköö siis enää uusia ideoita, joista kehittyy innovaatioita. Pitäisikö aivoille antaa enemmän vapaa-aikaa? Näistä aiheista lehden teksteissä tarkemmin.

Ajatuksia herättäviä lukuhetkiä!



Sisällys

3

Pääkirjoitus

Minna Oksanen

4

Paljon puhetta ja kirjoitusta innovaatioista

Lea Virtanen

6

Tekeekö innovaatio menestyvän tuoteyrityksen?

Jaakko Salminen

8

Design ei onnistu liukuhihnalla

Tarmo Toikkanen

12

Voidaanko tietojärjestelmiä tehdä toisin?

Kai K. Kimppa

14

Miten kehitimme innovatiivisen sovelluskonseptin – Sohva

Pauli Liikala

16

Monoliittisten prosessien aikakausi on ohi

Päivi Sirviö

19

Mittaaminen kertoo totuuden

Kari Juopperi

22

Valtionhallinnon ostotoimen ratkaiseminen

Paula Miinalainen

24

Kollaboraatioalustat sovelluskehityksessä – mikä muuttuu?

Mikko Holmberg

Perttu Karvinen

Joni Moilanen

27

Testaus on innovatiivista

Risto Paavola

29

Virheettömän ohjelmistotoimituksen haasteet

Matti Matikainen

30

Kuutamolla

Paljon puhetta ja kirjoitusta innovaatioista



Lea Virtanen,
yrittäjä, tietohallinnon
ja projektijohtamisen
asiantuntija sekä
johdon konsultti
JobIT Ky

Kun tämän lehden aihe ”Innovaatioita ja massatuotantoa” päätettiin vuosi sitten Systeemyöhydistyksen suunnitteluviikonloppuna, niin innovaatioista ja massatuotannosta odotettiin kirjoitettavan suunnilleen yhtä paljon. Toisin kävi! Kirjoittajat ovat nähneet kaikkialla innovaatioita ja niistähän koko yhteiskuntamme puhuu ja kirjoittaa tälläkin hetkellä.

Massatuotannolla mammonaa

Patenttien ja keksintöjen kehittäminen ei ole halpaa lystiä. Hyvän idean kehittäminen prototyyppiä ja tuotteeksi vie vuosia. Innovaatiot on saatava tuottamaan, se on oletettavasti lähes kaikkien keksintöjen perimmäinen tarkoitus. Ilman massatuotantoa ei ole menestyviä yrityksiä. Patenteilla ja keksinnöillä ei pitkälle pötkitä, jos ideoita ei pystytä tuotteistamaan ja kaupallistamaan. Tässä on erinomaisesti onnistunut Bill Gatesin perustama yritys. Hän on osannut panostaa sekä tuotteistukseen että markkinointiin. Microsoftin kaltaisia organisaatioita syntyy vain ani harvoin, mutta maailmalla on muitakin esimerkkejä nopeasti maailman valloituksen tehneistä innovaatioista. Bill Gatesilla ja Microsoftilla markkinat ovat olleet kotinurkilla toisin kuin meillä suomalaisilla. Kaksikymmentä vuotta sitten ei voinut arvailla, että tänä päivänä pienessä Suomessa on matkapuhelimia valmistava yritys, joka tekee 25 miljardin liikevaihtoa pelkästään puhelimilla tänäkin vuonna. Onneksemme kiinalaisia on enemmän kuin amerikkalaisia ja saksalaisia, jotka ovat hyljeksineet nokialaisia. Massatuotantoa siis tarvitaan taloudelliseen menestykseen. Massatuotantoa puolestaan voivat synnyttää hyvät tuoteinnovaatiot.

Suomi innovoi

Nettiä selaillessa huomasin, että Sitralla oli innovaatio-ohjelma vuosina 2004-2006. Kahden vuoden innovaatio-ohjelman tuloksena syntyi tavoite: Suomesta innovaatiotoiminnan kärkimaa.

Tämän keksimiseen on kahdessa, kolmessa vuodessa epäilemättä upotettu monen ihmisen työpanos ja paljon valtion rahaa, siis meidän veromaksajien rahaa.

Sitrان johdolla jatkettiin työtä innovoinnin saralla ja kesällä julkaistiin ”Kansallinen innovaatiostrategia”, johon on koottu perusajatuksia innovaatioiden kehittämisestä. Tämä strategiako tekee Suomesta innovaatiotoiminnan kärkimään?

Innovaatioyliopistoko ratkaisuksi

Ajankohtainen innovaatiohanke on innovaatioyliopisto, joka on edennyt ainakin julkisuudessa olevien tietojen mukaan kohtuullisesti. Nyt ollaan tilanteessa, jossa keskustellaan kiihkeästi Helsingin ja Espoon välillä yliopiston sijoituspaikasta. Kummallakin kaupungilla on omat hyvät perustelunsa ja nähtäväksi jää, kumpi kaupunki vetää pidemmän korren. Yliopisto on saanut taakseen vahvan osaajakaartin Koneen Matti Ala-Luhdan johdolla ja varmuuden siitä, että yliopisto aloittaa toimintansa vuonna 2010, mutta hallinnollisen keskuksen paikasta on päästävä sopuun ennen sitä.

Olisiko innovaatioyliopisto askel Sitran projektissa syntyneeseen tavoitteeseen. Innovaatioiden markkinointi on perinteisesti Suomessa ollut se, johon hyvätkin innovaatiot ovat kompastuneet. Kun tekninen ja kaupallinen osaaminen saatetaan yhteistyöhön ja lisätään vielä ripaus taiteellisuutta muotoiluun, löydetään tulevaisuudessa apua kompastelemaan markkinointiin.

Kilpailun innovatiivisuutta

Suomessa on pitkät perinteet erilaisilla innovatiivisuuskilpailuilla. Tuottava idea kilpailussa on kerätty innovaatioita 30 vuoden ajan. Kilpailulla on maineikkaita vetäjiä Keskuskauppakamaria, Kauppalehteä, Suomen Yrittäjiä ja mm. Työ- ja elinkeinoministeriö. Kilpailun ideana on innovatiivisen yritystoiminnan edistäminen.

Toinen kilpailu on INNOSUOMI -08. Tämän vuoden teemana on ympäristö ja energia, joka on osunut oikeaan ajankohtaan, kun Suomi kipristelee korkeiden energiahintapaineiden kanssa ja samalla pyrkii löytämään puhtaampia vaihtoehtoja tulevaisuuden energiaratkaisuiksi EU:ssa tehtyjen periaatepäätösten mukaisesti. INNOSUOMI brändin tavoitteena on edistää käytännönläheisesti ja

jatkuvasti uudistuen luovuutta, osaamista, yrittäjyyttä ja yhteistyötä Suomessa siten, että edellytykset kaikinpuoliseen hyvinvointiin koko maassa paranevat. Hankkeen suojelijana on Tasavallan presidentti.

Systeemityön tuottamia innovaatioita

Meidän alallamme ohjelmistoyrittäjät ovat tuoteliaita innovaatioiden kehittäjiä. Tässäkin lehdessä on esitelty muutama innovaatio, joka on syntynyt ohjelmistoyrittäjien piirissä. Innovaatio on Wikipedian määritelmän mukaan jokin uutuus, tavallisimmin jokin uutuustuote, esimerkiksi teollinen tai tekninen keksintö. Innovaatio voidaan ymmärtää ideana, käytäntönä tai esineenä, jota yksilöt pitävät uutena (Rogers 2003). Innovaatiot jakautuvat karkeasti kahteen luokkaan. On mullistavia innovaatioita ja vähittäisin muutoksin syntyviä innovaatioita.

Kun lehtemme artikkeleita selailee, niin sieltä löytyy innovaatioita, jotka ovat vielä menestysensä alkutaipaleella tavoittelemassa suurempia markkinoita ja toisaalta jokunen, joka on jo löytänyt vakiintuneen jalansijan ainakin lähimarkkinoilla. Tie maailman laajuiseen menestykseen on pitkä ja vaikea. Yrittäjiä läheltä seuranneena olen nähnyt epäonnistuneita ja kalliiksi tulleita yrityksiä, joissa yrittäjä kärsii rohkeudestaan elinikäisellä velkakierteellä. Toisaalta yrittämättä ei synny mitään, ainakaan uutta.

Tuoteinnovaatioiden ohella lehdessä on pohdittu systeemityön näkökulmasta innovaatiota mm. uudelleenkäyttöä ja avoimen koodin hyödyntämistä, virheettömän koodin tuottamisen hankaluutta ja testauksen innovatiivisuutta. Nämä kuuluvat luokkaan vähittäisin muutoksin syntyvät innovaatiot. Lukijan arvioitavaksi jätän näiden innovaatioiden merkityksen kunkin omassa toimintaympäristössä. Varmasti näillä kaikilla systeemityön osa-alueilla on jatkossakin puhaltamassa uusia tuulia ja kun sukupolvet vaihtuvat, niin vanhat asiat vähintään keksitään ja nimetään uudelleen.

Pitkään on yritetty moduulien uudelleenkäyttöä. Pohdin, miksi ohjelmoida haluaa koodata sellaista, mitä on jo koodattu aina uudelleen ja uudelleen. Avoimet rapapinnatkin on keksitty jo kauan sitten. Ehkä yksi asia ohjelmoijan näkökulmasta on se, että ohjelmointityö alkaa tuntua rutiinilta, jos vain kutsutaan valmiita moduuleja. Vapaat kädet ohjelmoijalle antaa mahdollisuuden ongelman ratkaisuun usealla tavalla ja myös mahdollisuuden persoonalliseen ajatteluun. Toinen kysymys on, onko koodi silloin optimoitua ja koodaus tuottavaa. Uudelleenkäytettävyyys on ehkä yksi ratkaisu tuottavampaan työhön, myös testauksen määrä vähenee tätä kautta. Toki välinekehityskin vaikuttaa siihen, että yleismoduuleita koodataan uudelleen ja uudelleen vuosi toisensa jälkeen.

Viherherääminen

Henkilökohtaisesti olen viimeisen vuoden aikana kokenut viherheräämisen. Vihreä ICT on vallannut alaa ja sen merkitystä ei voi enää väheksyä toimittajavalintoja tehtäessä. Imago merkityksen lisäksi vihreä ICT säästää puhdasta rahaa. Organisaatiosta ja laitekannasta riippuen voi olla isokin merkitys sillä ovatko esimerkiksi laitteet yön yli päällä kuluttaen kallista energiaa. Suurin energiahukka tapahtuu yleensä konesaloissa, joiden suunnittelu on harvassa paikassa toteutettu huomioiden tämän päivän energiakustannukset. Onko esimerkiksi energian talteenotto suunniteltu ja toteutettu ja miten tilojen käyttö on optimoitu huomioiden pienenevät palvelinkoot.

Vihreän ICT:n teeman alla tässä lehdessä tutustutaan innovaatioon, jonka avulla tavoitellaan työasemien virrankulutusta vähentämällä säästöjä. Tähän innovaatioon tutustuessani joitakin kuukausia sitten totesin, että tämä sopii erityisesti verkossa olevien kiinteiden työasemien hallintaan ja myös mainiosti tämän vuoden INNOSUOMI teemaan ympäristö ja energia sekä tämän Systeemityölehden teemaan.





Jaakko Salminen on toimitusjohtaja ja koulutukseltaan ohjelmistoliiketoiminnan DI TKK:lta. Ravensoft Oy

Tekeekö innovaatio menestyvän tuoteyrityksen?

Miten palveluja toimittava yritys muuttetaan innovaatiovetoiseksi tuoteyritykseksi? Ravensoft Oy lähti selvittämään tätä ja matka jatkuu edelleen.

Palveluyrityksen kultainen häkki

Palveluliiketoiminnalla on monia ylivoimaisen hyviä puolia varsinkin aloittavalle yritykselle. Se ei vaadi mitään merkittäviä investointeja käynnistettäessä ja laskutus on tasaista sekä suhteellisen varmaa kun hyvät asiakassuhteet on luotu. Liiketoiminta laajenee palkkaamalla lisää työntekijöitä, jotka usein saadaan sijoitettua nykyisten asiakkaiden hankkeisiin. Lisäksi talous on helppo pitää tasapainossa.

Toisaalta palveluliiketoiminnalla on myös vakavia rajoitteita. Sen nopea kasvattaminen on vaikeaa, koska se vaatii aina uusien henkilöiden mukaan ottamista. Vienti muihin maihin on käytännössä mahdotonta. Palveluyritykseen on vaikea saada kerääntymään IPR:ää, koska asiakkaat lähes aina vaativat ne kokonaan itselleen. Näin yrityksen arvo kasvaa vain laskutuksen suhteessa.

Ravensoft Oy päätti 2006 alussa murtautua ulos tästä kultaista häkistä ja hakea nopeaa kasvua laajentamalla tuoteliiketoimintaan. Tuote-innovaatioita yritykseen oli kerääntynyt yllin kyllin vuosien ja projektien varrelta, mutta mitä muuta tarvittiin menestykseen?

Päätöksiä

Kevään ja syksyn 2006 aikana keräsimme kaikki mahdolliset ja mahdottomatkin ideat, joita yritykseen oli kerääntynyt. Vaikka mietimme lopussa pitkään viiden idean välillä, valinta oli lopulta helppo. Green Snapper (silloin nimeltään Sandman eli nuksumatti..) täytti kaikki tuotteelle asettamamme tavoitteet:

- Kansainvälisyys. Tuote ei saa olla sidottu yhteen markkinaan.
- Skaalattavuus. Tuote on oltava käyttövalmis sellaisenaan, ilman asiakaskohtaista muokkaamista.
- Nopea kehitys. Ensimmäisen version tuotteesta täytyy olla tehtävissä alle vuodessa, jotta sitä päästään heti myymään.
- Innovatiivisuus. Tuotteen on oltava uutuus, ei ainoastaan kohennusta olemassa olevaan.

Green Snapper vastasi lisäksi merkittävään nousemassa olevaan trendiin, vihreään IT:hen. Teetimme 2006/2007 vuodenvaihteessa markkinatutkimuksen, joka vahvisti sekä asiakkaiden heräävän kiinnostuksen vihreää IT:tä kohtaan, sekä tuotteemme potentiaalin. Toisaalta tutkimus myös paljasti, että vihreä IT oli markkinasegmenttinä vasta käynnistysvaiheessa, ja joutuisimme tekemään merkittäviä markkinointiponnistuksia sen herättämiseksi.

Se toimii!

Green Snapperin ensimmäinen prototyyppi oli valmis huhtikuussa 2007 ja kuun lopussa se esiteltiin pilottiohjelmaan mukaan tulleeille asiakkaille. Pilottiohjelma antoi meille arvokasta tietoa ohjelmiston käytöstä todellisissa asiakasympäristöissä ja ohjasi kehitystä jo varhain oikeaan suuntaan. Kehitystä jatkettiin aina elokuuhun asti, jolloin perusteelliset järjestelmätestaukset käynnistettiin. Ohjelmisto oli valmis täsmälleen suunniteltuna päivänä, 30.9.2007, kiitos tehokkaan ketterän kehityksen prosessimme ja hyvien teknisten kumppanimme.

Kehityshanke sinänsä oli meille tuttua kauraa, olimmekaan tehneet vastaavia asiakkaillemme jo 10 vuoden ajan. Hankkeen tästä vaiheesta emme itse asiassa olleet missään vaiheessa huolestuneita, koska sen tiesimme osaavamme.

Uuden brändin luonti

Suuremmat haasteet olivat meillä edessä markkinoinnin ja myynnin parissa, koska tähän mennessä asiakkaamme olivat tuoneet kehittämämme tuotteet markkinoille. Lisäksi tiesimme, että uudenlainen tuote vaatii merkittävän panostuksen juuri markkinointiin. Luotimme jälleen ulkopuolisten yritysten apuun sekä brändin luomisessa, että tuotteen julkistuksessa.



Kuva 1 ja 2: Esimerkkejä tuotteille luoduista brändeistä.

Halusimme alusta asti luoda Green Snapperille vahvan brändin, koska sen oli tehokkaasti erottuttava muista ohjelmistotuotteista. Markkina oli uusi, ja matalan profiilin markkinointi ei olisi auttanut meitä.

Tuote julkistettiin 10.10.2007 näyttävässä tilaisuudessa Helsingin Tennispalatsissa. Paikalla oli yli 50 kiinnostunutta asiakasta ja lehdistön edustajaa, ja tilaisuus lehdistönäkyvyyksineen oli menestys.

Tilanne tällä hetkellä

Tuotteemme on herättänyt laajaa kiinnostusta, kuten markkinatutkimuksen perusteella ennakoimmekin. Samoin markkinointimme on ollut erittäin onnistunutta ja olemme onnistuneet saamaan tuotetta tunnetuksi. Myynti on lähtenyt hyvin liikkeelle, ja monet asiakkaat käyttävät jo Green Snapperia.

Missä olemme sitten onnistuneet vähemmän hyvin? Vaikka panostimme myyntiorganisaation rakentamiseen jo keväästä 2007 alkaen, kohtasimme yllättävän paljon vastoinkäymisiä ja selkeää huonoa tuuria juuri tässä kriittisessä osassa. Saimme myyntipäällikön palkattua maaliskuussa 2008, ja sen jälkeen myyntiprosessimme on kiihtynyt voimakkaasti. Alunperin suunnitelmamme oli saada myyntipäällikkö mukaan tiimiin jo lopukesällä 2007.

Mitä opimme?

Kuten ennakoimmekin, uuden tuotteen kehittäminen oli meille helppoa, mutta markkinoille tuominen vaativaa. Tiesimme ottavamme riskin tuomalla uuden tuotteen herääville markkinoille, ja tähän puoleen liiketoiminnan käynnistämiseksi liittyvät suurimmat haasteemme ja ahaa-elämyksemme. Lisäksi kahden hyvin erilaisen liiketoiminnan pyörittäminen samaan aikaan samassa yrityksessä on osoittautunut vaativaksi.

Hankkeemme jatkuu yhä, mutta tässä lopuksi joitakin asioita, joita olemme oppineet matkan varrella. Toivotan muutkin yritykset tälle samalle polulle, meistä kukaan ei kadu tästä hetkeäkään!

- Tuoteliiketoiminta ja palveluliiketoiminta ovat hyvin erilaisia.
- Tuotteen hallinta pitää aloittaa heti alussa ja se jatkuu koko ajan.
- Kahden erilaisen liiketoiminnan pyörittäminen pienessä yrityksessä on hyvin haastavaa.
- Ketterä kehitys mahdollistaa nopean reagoinnin.
- Markkinointiin palaa enemmän rahaa kuin ennakoon luulisi.
- Myyntiin on panostettava vahvasti jo ennen julkistusta.



MAKE IT BLOSSOM

Säästä energiaa, säästä ympäristöä, säästä rahaa



HANKI YLEMPI AMK-TUTKINTO TIETOJENKÄSITTELYN ALALTA!

Työn ohessa opiskelu irroittaa arjesta, verkottaa kollegat, antaa välineitä päivittäiseen työhön ja luo uusia mahdollisuuksia. 90 opintopisteen laajuinen ohjelma on suunniteltu suoritettavaksi kolmen vuoden aikana. Haku tammikuussa 2009 alkaviin opintoihin on 29.9. - 10.10.2008.

TIETOJÄRJESTELMÄOSAAMISEN KOULUTUSOHJELMA (ylempi AMK)

Helsingissä Pasilassa

Tutustu ohjelmaan:

www.haaga-helia.fi

Lisätietoja:

hakutoimisto@haaga-helia.fi

puh. (09) 2296 5221



HAAGA-HELIA
ammattikorkeakoulu



Tarmo Toikkanen on psykologi ja tietojenkäsittelijä, joka toimii tutkijana Taideteollisen korkeakoulun Medialaboratorion Oppimisympäristöjen tutkimusryhmässä projektipäällikkönä. Hän toimii lisäksi yksityisyrityksenä ohjelmistosuunnittelijana, vapaana toimittajana, kouluttajana, kääntäjänä, kirjailijana ja konsulttina.

Design ei onnistu liukuhihnalla

Taideteollisessa korkeakoulussa osataan design. Design ei TaiKin Medialabrassa tar-koita ohjelmistosuunnittelijan designia (tyyliin uml-kaaviot), vaan jotain ihan muuta. Käyttäjäkeskeisyys tai -lähtöisyys, iteratiivisuus ja designerin vahva rooli ovat joitain onnistuneen ohjelmistoprojektin elementtejä. Tässä artikkelissa kuvataan tiiviisti eräs elinkaarimalli, joka jäsentää tällaista ohjelmistotuotantoprosessia.

Innovaation luonteesta

Innovaatio tarkoittaa uuden menetelmän, idean tai tuotteen keksimistä ja kehittämistä. Olennainen erotus keksintöön on se, että keksityn asian on

Yleistymisen kriittinen vaatimus on helppokäyttöisyys. Suuri yleisö ei hanki laitetta tai palvelua, jota on vaikea käyttää tai ymmärtää. Tämä pätee sekä fyysisiin tuotteisiin että ohjelmistoihin, joskin ohjelmistoissa helppokäyttöisyys tuntuu noin yleensä olevan vaikeammin saavutettavissa. Kenties syynä ovat käytetyt kehitysmenetelmät tai tuotteiden abstraktisuus? Esimerkiksi blogaus yleistyi vasta helppokäyttöisten blogiohjelmistojen ja -palvelujen myötä, vaikka teknisesti blogin pitäminen oli ollut mahdollista kenelle tahansa jo kymmenen vuotta sitä aiemmin.

Otetaan esimerkkinä tyypillisestä ohjelmistosta nimeltä mainitsemaan suomalaisen ohjelmisto-

talon matkalaskujärjestelmä (kuva 1). Aiemmin ulkomailla matkustavat professorit ja tutkijat pistivät kuitit sisäisessä postissa osaston sihteerille, joka hoiti asian eteenpäin. Nykyään jokainen virkamies tekee itse matkasuunnitelmansa tällä järjestelmällä, matkan jälkeen teippaa kuitit erilliselle lomakkeelle, skannaa ne pdf:ksi ja liittää matkalaskuun, jonka tekee tällä järjestelmällä. Prosessi kyllä toimii, mutta professoreiden kallista työaikaa kuluu nyt helposti useampi tunti matkaa kohti, kun he taistelevat järjestelmän kanssa, jonka kohdekäyttäjä on selvästi matkalaskujen tekemiseen erikoistunut taloussihteeriksi. Ja kun järjestelmää ei edes käytetä jatkuvasti vaan tyypillisesti muutaman kerran vuodessa, ehtii kaiken unohtaa käyttökertojen

Kuva 1. Suomalaista tekoa oleva matkalaskujärjestelmä, jonka kohderyhmä on matkalaskuja käsittelevä sihteerit, mutta jota silti käyttävät kaikki työmatkoja tekevät virkamiehet vaihtelevalla menestyksellä ja huonolla tehokkuudella.

yleistyttävä ja saatava aikaan jotain konkreettista parannusta maailmaan. Pöytälaatikkoon tehdyt keksinnöt eivät ole innovaatioita. Eivätkä keksinnöt, joita tuotteistetaan isolla rahalla mutta silti epäonnistuvat. Vasta kun keksintö on yleisessä käytössä, voidaan innovaation katsoa tapahtuneen.

välillä. Tämä on myös esimerkki ohjelmistosta, jossa selvästi keskeisenä arvona on ollut mas-
satuotanto – tehdään yksi yleiskäyttöinen ohjelmisto, jota voidaan myydä kaikkialle minimaalisin muutoksin. Ja maksimoidaan samalla tarvittavan henkilöstökoulutuksen määrä, jolla saadaan lisää tuloja.

Helppokäyttöisen ohjelmiston kehittämisestä

Käyttäjakeskeisyydestä on etenkin uuden Käytettävyyden tilaisuuksissa puhuttu paljon, koska se tuntuu olevan vaikea pala insinööripainotteiselle kehittäjäporukalle. Kuvassa 2 on Oppimisympäristöjen tutkimusryhmän vuosikymmenen aikana kehittämä iteratiivinen, ketterä ohjelmistokehitysmalli "Ohjelmistoprototyyppi hypoteesina", jolla on viimeksi tuotettu EU-projektissa paljon kehuja saanut LeMill-yhteisö opettajille.

Mallin laajuuden näkee siitä, että perinteinen insinöörimäinen ohjelmistotuotanto on puristettu kuvan 2 viimeiseksi vaiheeksi, "Software prototype as hypothesis". Tämä vaihe tehdään ketteriä kehitysmenetelmiä käyttäen, jotta ohjelmistosta saadaan muutaman viikon välein uusi versio testattavaksi ja jotta vaatimuksia voidaan muuttaa koska tahansa.

Joidenkin yritysten mainostama käyttäjakeskeinen kehitys ottaa mukaan "Product design"-vaiheen, jossa hyödynnetään kertakäyttöprototyyppiä, käyttäjätarinoita ja käyttötapauksia. Tässä menetelmässä tuotakin vaihetta edeltää kuitenkin paljon työtä.

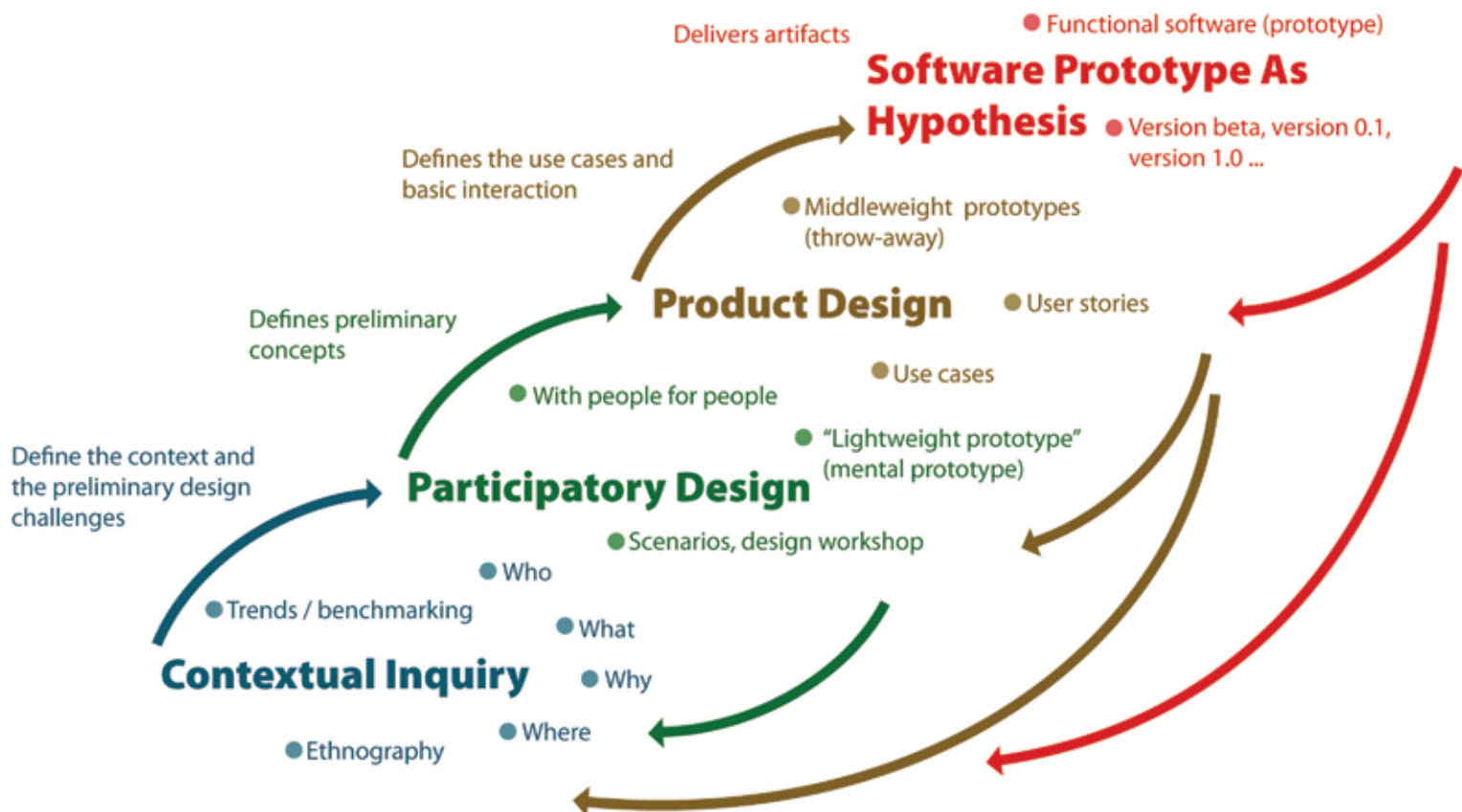
Projekti alkaa "contextual inquiry"-vaiheessa, jossa etnografian, antropologian ja muiden sosio-

logien menetelmien avulla tutustutaan kohderyhmään, sen toimintatapoihin ja -kulttuuriin. Tästä tutkimustyöstä saadaan esiin ensimmäiset design-haasteet. Kohderyhmän ymmärtäminen on tärkeää, sillä muuten tuotettu ohjelmisto voi päällisin puolin ratkaista sille asetetun haasteen, mutta aiheuttaa samalla lisää ongelmia (vertaa aiemmin mainittu matkalaskujärjestelmä).

Nämä design-haasteet viedään kohderyhmästä valittujen edustajien pohdittavaksi "participatory design"-istuntoihin, joissa käydään läpi designereiden laatimia skenaarioita. Istunnot tallennetaan ja litteroidaan. Niitä analysoimalla designerit muodostavat keskeisimmät käsitteet, joita kohderyhmä käyttää oikeassa elämässään ja näiden käsitteiden ympärille aletaan rakentaa design-ratkaisuja.

Vasta näiden vaiheiden jälkeen päästään tavanomaiseen ohjelmiston suunnitteluprosessiin, jota tehdään evolutiivisen prototilun menetelmällä, ketterästi, soveltaen Scrum- ja XP-menetelmien arvoja ja käytäntöjä. Tavoitteena on pitää arkkitehtuuri yksinkertaisena ja automaattisilla yksikötesteillä varmistaa, että ohjelmistoa voidaan refaktoroida ongelmitta. Ohjelmoijat tuottavat 2-4 viikon välein uuden version ohjelmistosta, joka välittömästi otetaan käyttöön ja kohderyhmän validoitavaksi.

Kuva 2.
"Ohjelmistoprototyyppi hypoteesina"-kehitysmalli.



Validoinnissa selviää aina uutta. Vaikka kohderyhmää on kuinka perusteellisesti seurattu ja tutkittu, eivät he ole kuitenkaan voineet täysin ymmärtää ohjelmistoratkaisua kun siitä on näytetty vain paperiproto tai skenaario. Mitä konkreettisempi ohjelmistoprototyyppi on, sitä paremmin kohderyhmänkin edustajat ymmärtävät, mistä siinä on kyse. On siis odotettavissa, että validoinnissa asiat muuttuvat radikaalistikin. Peruskäsitteet tuskin ovat väärin, mutta käsitteiden priorisointi, esitystavat, interaktiologiikka ja oikeastaan kaikki käyttäjälle näkyvä tulee kokemaan useita muutoksia.

Käyttäjien palautetta ei suoraan käytetä ohjenuorana kehitystyölle, vaan designerit käsittelevät palautteen ja muokkaavat design-ratkaisuitaan vastaavasti. On tärkeä muistaa, että käyttäjät usein haluavat eri asiaa kuin tarvitsevat. Kaikki eivät ole designereita. Saatu yksilöiltä tullut palaute on siis käsiteltävä, jotta saadaan yleisemmin tuotteen toimintaa parantava design-ratkaisu. Tämä ratkaisu voi olla hyvinkin erilainen kuin mitä palautteen antaja olisi toivonut.

Otetaan havainnolliseksi erimerkiksi kolme vuotta kestänyt LeMill-yhteisön kehittäminen. Teknisesti ratkaisu ei mitenkään suuresti muuttunut kunhan perustoiminnot oli saatu toteutettua. Myöskään peruslogiikka ei kaivannut muutoksia, mutta ulkoasun viilaaminen kesti lähes kaksi vuotta ja uusia käsitteitä jouduttiin nimeämään monesti, kunnes kohderyhmän suuhun sopiva esitysmuoto löytyi. Pienet yksityiskohdat, kuten

kenttien esitysjärjestys, sekä etenkin yksinkertaisuuden ja yksiselitteisyyden yhteensovittaminen olivat ne vaikeimmin paikoilleen saatavat elementit.

Nyt kolmen vuoden jälkeen LeMill on saanut EU:n komission arviointipaneelilta loistoarvion. Ja siitä huolimatta, että LeMill on monien maiden opettajille hyvinkin edistysellinen pedagogisessa mielessä, on LeMill valittu jo Viron ja Georgian viralliseksi avoimen oppimateriaalin varannoksi. Yhteisössä on tätä kirjoitettaessa vajaat 3000 opettajaa ja lisää tulee 10% kuukausivauhdilla. Joten jotain siinä on tehty oikein. Ja kun LeMill vaatii opettajilta melkoisen edistysellistä pedagogista otetta, ei se tue kovinkaan hyvin nykyistä eurooppalaista koulutuskulttuuria. Mutta LeMillin jatkuvasti kasvava suosio näyttää siltä, että siinä on keksinnöstä muodostumassa innovaatio.

Tämän ja muiden ohjelmistoprojektien (Fle3, MobileD, Taidekasvatuksen Kuva-arkisto...) onnistumisen perusteella voidaan todeta, että kuvattu ohjelmistokehitysmalli on toimiva. Se ottaa erittäin hyvin huomioon kohderyhmän todelliset tarpeet ja pystyy iteratiivisesti lähestymään optimaalista ratkaisua. Se ei ole kuitenkaan massa-tuotantoa, sillä kukin ratkaisu on yksilöllinen eikä kovin helposti yleistettävissä muille soveltuvaksi. Projektissa on varattava merkittävästi resursseja designereiden työhön muiden ammattilaisten lisäksi ja prosessissa on varauduttava siihen, että designerilta tulee kuukausittain erilaisia käyttötarinoita toteutukseen.

Kuva 3. LeMill-yhteisön etusivu, joka esittelee varannossa olevia opetus- ja oppimistarinoita.

The screenshot shows the LeMill website homepage. At the top, there is a navigation bar with links like 'side', 'en', 'es', 'it', 'fr', 'hu', 'ka', 'is', 'pl', 'ru', 'se'. Below this, the LeMill logo is prominently displayed. To the right of the logo, there are three boxes for 'Webquest', 'Using WIKI for collaborative writing', and 'Wetpaint', each with a corresponding icon and a button labeled 'Sisältö', 'Menetelmät', and 'Työkalut'. Below these, there is a section titled 'Opetus- ja oppimistarina / kokoelma: Media' with a brief description and a 'Lue lisää...' link. Further down, there is a section 'Etsi oppimateriaalia, menetelmiä, työkaluja tai ihmisiä...' with a search bar and a 'Alota' button. At the bottom, there is a footer with statistics: 'LeMill-web-yhteisössä on 2804 opettajaa ja sisällöntuottajaa. Tällä hetkellä LeMillissä on 1400 uudelleenkäytettävää oppimisaihiota, 292 opetus- ja oppimismenetelmän kuvausta ja 456 opetus- ja oppimistyökalun kuvausta. Jos sinun oppimisresursstasi eivät vielä ole LeMillissä, liity yhteisöön niin tehdään ne yhdessä.' Logos for the European Union and YLE are also present.

Innovaatio vaatii keksinnön yleistymistä oikeaan käyttöön, ja yleistyminen vuorostaan on helppoa, kun keksintö toteutetaan muodossa, joka ottaa huomioon kohderyhmän työskentelytavat ja -kulttuurin, siis suunnitellaan kohderyhmän kätehen sopivaksi. Massatuotannolla tehty, yleiskäyttöinen ohjelmisto on luonnollisesti halvempi tuottaa, koska sen monistaminen useille asiakkaille on triviaalia. Valitettavasti harva ohjelmistoyritys pyrkii minimoimaan asiakkaalleen tulevia lisäkustannuksia koulutustarpeen ja menetetyt työajan muodossa, joita molempia huonosti käyttäjilleen soveltuva ohjelmisto aiheuttaa. Ja harvan tarjouskilpailun voittaa kalliimpi, hyvin designattu ratkaisu, joka pitkällä tähtäimellä säästää asiakkaan työntekijöiden työaika ja mielenterveyttä ja parantaa työssäviihtyvyyttä. Design, käytettävyys ja pehmeät arvot ovat vaikeasti numeerisesti mitattavissa, ja siksi monet eivät niitä mitatakaan, vaikkakaan tehtävän vaikeus ei ole hyvä syy jättää sitä tekemättä.

Kuva 4. Esimerkki LeMill-yhteisössä olevasta resurssista, tässä tapauksessa käsitekarttatyökalun kuvauksesta. Ylhäällä on vakionavigointielementit ja vasemmalla muokkaustyökalut.

EXPERICON

Onko Expericon paras työpaikka
projektipäällikölle ja IT-asiantuntijalle?

– www.expericon.fi –



Kai K. Kimppa
Lehtori
Turun Yliopisto
Informaatioteknologian laitos

Voidaanko tietojärjestelmiä tehdä toisin?

Voisiko vapaa ja avoin ohjelmistokehitys-paradigma tuottaa parempia järjestelmiä? Artikkelissa pohditaan yhteistyön ja uudelleenkäytön tarjoamia mahdollisuuksia järjestelmäkehityksessä. Tämä parantaisi pienten ja keski suurten ohjelmistotuottajien mahdollisuuksia toimia kentällä helpottamalla pääsyä markkinoille ja mahdollistaisi organisaatioille niiden käyttämien järjestelmien kehittämisen kilpailuttamisen vapaasti.

Uudelleenkäytön lupaukset eivät ole toteutuneet

Olio-ohjelmoinnin lupaukset moduulien uudelleenkäytettävyydestä ovat toteutuneet vain osittain. Järjestelmiä kehitettäessä edelleen 'keksitään pyörää uudestaan', vaikka tarvetta sille ei modulaarisen järjestelmäkehityksen tultua yhä yleisemmäksi olisi. Ongelman muodostaa takeruminen vanhaan immateriaalioikeusmalliin joka sulkee jo kehitetyt moduulit yrityskohtaisiksi sen sijaan, että niitä voitaisiin käyttää hyödyksi uusien järjestelmien kehittämisessä. Koska tekijänoikeus suojaa erityisesti käännettyä koodia lähdekoodin pysyessä salaisuutena, on lähdekoodin lisensointi hankalaa. Jos lähdekoodin julkaisee, ei kilpailijan niin tarvitse tehdä, jolloin on vaikea todistaa onko kilpailijan käännetty, ajonaikainen koodi perustunut julkaistuun vai itse kirjoitettuun lähdekoodiin. Samankaltaisia ratkaisuja tulee helposti, ja toista kääntäjää käyttämällä on käännetty koodi kuitenkin erilaista, vaikka lähdekoodi olisikin ollut sama.

Immateriaalioikeus vääristää kilpailua myös antamalla monopolin kirjoitettuun koodiin. Vaikka immateriaalioikeuksien tarkoituksena on antaa rajoitettu monopoli, on monopoli tiettyyn koodiin käytännössä rajoittamaton. Viimeisen tekijän kuolinvuodesta seitsemänkymmenen vuoden kuluttua vapautuva koodi on itsestään selvästi ikuisen suojan alla – koodin käytettävyyden yli seitsemänkymmenen vuoden päästä on käytännössä mahdotonta. Toistaiseksi meillä ei ole mitään koodia, joka olisi näin vanhaa¹, ja ottaen ohjelmistotalan kehitysvauhdin huomioon, mikään edes

parikymmentä vuotta vanha koodi ei juurikaan ole käyttökelpoista enää tänään. Puhtaat algoritmit ovat tietysti luku erikseen. Niille ei perinteisesti ole immateriaalioikeussuojaa voinut saada koska niiden katsotaan olevan matemaattisia yhtälöitä jotka ovat tieteellisiä lakeja ja siten immateriaalioikeuksien ulkopuolella. Rajoittamattomia monopoleja pyritään estämään kaikilla muilla aloilla paitsi immateriaalioikeuden alaisilla. Jostain syystä ne siellä katsotaan normaaleiksi vaikka monopolit muuten katsotaan kilpailua vääristäviksi tekijöiksi.

Vapaan ja avoimen ohjelmistokehityksen avulla uudelleenkäyttämään?

Jos nyt modulaarisen ohjelmoinnin aikakaudella siirryttäisiin käyttämään vapaan ohjelmistokehityksen idean mukaista kehitysmenetelmää (jonka puolesta väitöskirjassani esitän muita argumentteja) tulisi ohjelmistokoodin laajempi uudelleenkäyttäminen mahdolliseksi. Organisaation tietojärjestelmät itsessään täytyy kuitenkin tehdä organisaatiokohtaisesti toimiviksi; mitään "hopealuotia" joka ratkaisisi edes jonkun alan kaikkien organisaatioiden tarpeet tuskin koskaan löydetään. Tällöin ohjelmointityö sinällään ei toki loppuisi eikä tarve sovittaa moduulit organisaation tarpeisiin poistuisi, mutta järjestelmäkehitystä ei aina tarvitsisi aloittaa uudestaan vaan jo kehitetyt moduuleita voitaisiin hyväksikäyttää laajalti.

Myös kilpailuasetelma paranisi. Uusien toimijoiden aloittaminen helpottuisi – voisivathan ne näin käyttää aiemmin ohjelmoituja moduuleita hyväkseen – joka lisäisi alan kilpailua. Samaten korjaukset, lisäykset ja uudet osat olemassa oleviin järjestelmiin voitaisiin kilpailuttaa erikseen, eivätkä organisaatiot olisi sidoksissa yhteen järjestelmätoimittajaan. Tämä kannustaisi järjestelmätoimittajia hyvään palveluun ja huolelliseen ohjelmointiin. Tietysti mikäli normaalisti muilla aloilla edellytetty palvelualltius olisi ohjelmistotalallakin normi, käyttäisi organisaatio varmasti mieluiten järjestelmän alkuperäistä toimittajaa jatkokehityksenkin tekijänä; onhan heillä selke-

ästi järjestelmän kehittäjänä myös paras ymmärrys sen toiminnasta. Organisaatio ei kuitenkaan olisi sidottu näin tekemään, jos palvelu ei pelaisi tai organisaation tarpeita ei otettaisi riittävästi huomioon.

Markkinatilanteen muuttuessa myös markkinointistrategioiden täytyisi tietysti sopeutua uuteen tilanteeseen. Koska ensimyyntin (ks. Kuva 1) täytyisi kattaa ainakin merkittävä osa kehityskustannuksista – seuraavien myyntien ollessa selkeästi kilpailumpia muidenkin voidessa hyödyntää kehitettyjä moduuleita – olisi sekä kehittäjällä että ostavilla organisaatioilla nykyistä selvempi motiivi yhteistyöhön toisten kehittäjien ja organisaatioiden kanssa järjestelmiä tilattaessa. Myös kustannusten korvaus siirtyisi enemmän tehdyn työn määrästä riippuvaksi ”lopullisen” tuotteen uudelleenmyyntin sijaan. Koska olen kuitenkin enemmän yhteiskuntafilosofi kuin markkinoinnin ammattilainen, jätän uusien mallien kehittämisen, sikäli kuin ne ajankohtaisiksi tulevat, markkinoinnin ammattilaisille: yritysten markkinointiasetoille ja kauppa- ja korkeakouluihin.

Kuvassa 1 on malli ensimmäisen myyntin osuus järjestelmän tuomasta tulosta. Seuraavien myyntien osuus on mallissa selvästi laskeva koska saman järjestelmän osia tai kokonaista järjestelmääkin on kauppaamassa myös kilpailijoita.

Ehkä paras uudistus esittämästäni mallista seuraisi kuitenkin tietojärjestelmän syvällisen oppimisen mahdollistumisena. Jos organisaatioissa olisi halua ymmärtää järjestelmän toimintaa syvällisemmin, ei mikään estäisi sen omia toimijoita, esimerkiksi IT-osaston työntekijöitä tai vaikka edistyskellisiä käyttäjiäkään tutustumasta järjestelmän toimintaan ohjelmakoodin tasolla. Näin ymmärrys järjestelmän toiminnasta olisi parempi myös organisaation sisällä, eikä kaikkia korjauksia tarvitsisi odottaa sormi suussa ilman mahdollisuutta vaikuttaa niiden tapahtumiseen tai toteutustapaan.

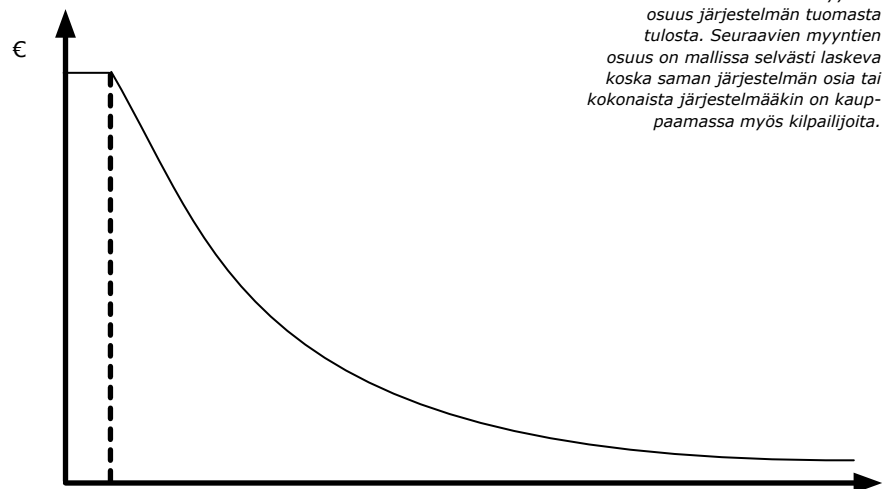
Esittämästäni mallista on joitain esimerkkejä vapaan ja avoimen ohjelmistokehityksen (FLOSS²) piirissä jo nyt³. Ongelmaksi on muodostunut yleisen ilmapiirin vastaisuus vapaan ja avoimen ohjelmakoodin kehityksen kautta tuotetuille järjestelmille. Jopa valtion ja kuntien organisaatioita vaaditaan – yleensä toisistaan erillään – kilpailuttamaan

”vapaila” markkinoilla tarjotut järjestelmäratkaisut. Tämä estää, tai ainakin haittaa järjestelmien hankkimista yhteistyössä, joka taas yleensä sulkee tarvittavat investoinnit FLOSS-kehityksen toteuttamiseksi pois – kun järjestelmän kehittäjä ei voi luottaa uudelleenmyynteihin tai monopoliin järjestelmän jatkotoimittajana ja huoltajana vaan heidän täytyy tehdä alkuperäinen tarjous kokonaiskulut mielessään. Tuntuu ristiriitaiselta lukea monilta tahoilta kuinka kuntien ja valtioiden organisaatioissa pitäisi pyrkiä avoimuuteen toimintatapojen suhteen ja samalla käytännössä tämä järjestelmien suhteen valtaosin estetään. Vanha sanonta siitä kuinka hallinnon oikea käsi ei tiedä mitä vasen tekee tulee väistämättä mieleen.

Lopuksi

FLOSS-malliin perustuva järjestelmäkehitys, niiltä osin kuin se on toteutunut, on muutamaa suurta kansallista projektia lukuun ottamatta kuitenkin jäänyt lähinnä pienten ja keski suurten yritysten pienten järjestelmien kehityksen työkaluksi. Toivottavasti suuretkin organisaatiot voivat ja erityisesti kansallisella ja kunnallisella taholla saavat jatkossa kilpailuttaa tarvitsemansa tietojärjestelmät ottaen kaikki mahdolliset toteutustavat huomioon, olematta ”väkisin” ajettuja käyttämään nykyisin vallitsevaa suljettua ohjelmistokehitystä.

FLOSS
= vapaa ja avoin
ohjelmistokehitys
(Free/Libre/Open-
Source Software)



Kuva 1: Ensimmäisen myyntin osuus järjestelmän tuomasta tulosta. Seuraavien myyntien osuus on mallissa selvästi laskeva koska saman järjestelmän osia tai kokonaista järjestelmääkin on kauppaamassa myös kilpailijoita.

¹Ellei Ada Byronin Bernoullin lukuja laskevaa koodia Babbagen koneelle lasketa.

²FLOSS = vapaa ja avoin ohjelmistokehitys (Free/Libre/Open-Source Software)

³Ks. esim Kolab, <http://www.kolab.org/> tai JBoss, <http://labs.jboss.com/portal/index.html?ctrl=id=page.default.default>

Kirjallisuutta

Kimppa, Kai K. (2007), *Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media*, väitöskirja, TUCS Dissertations, No 83. Osittain ladattavissa: <http://www.tucs.fi/publications/insight.php?id=phdKimppa07a>



Pauli Liikala toimii konsultointijohtajana Fujitsu Services Oy:ssä. Pauli Liikala vastasi Fujitsun kehitysprojektista, jolla innovatiivinen sovelluspalvelukonsepti Sohva kehitettiin.



Sohva

Miten kehitimme innovatiivisen sovelluspalvelukonseptin – Sohva

Tietotekniikan hyödyntäminen on kulkenut pitkän matkan yritysکوhtaisista erillisratkaisuista kohti standardoituja ratkaisuja ja niihin perustuvia palveluja. Tämä kehityskulku näkyi ensin perustietotekniikassa 1990-luvulla, jolloin standardointi ja markkinavoimat huolehtivat siitä, että suurilla yrityksillä tuli vakiintuneet perustietotekniikan ratkaisut. Tämä johti myös perustietotekniikan palveluiden kehittämiseen, josta tunnetuin esimerkki Suomessa on Fujitsu Service Oy:n Patja.

Sama kehityskulku on näkynyt myöhemmin myös sovellusmarkkinoilla, joissa aiemmin räätälöidyt yritysکوhtaiset ratkaisut on korvattu markkinoilla vakiintuneilla toiminnanohjausratkaisuilla. Kuvaan tässä artikkelissa, miten kehitimme innovatiivisen sovellusten ulkoistamiskonseptin. Kun perustietotekniikan palvelu oli nimetty Patjaksi, löytyi tämänkin palvelun nimi huonekaluista, eli sen nimeksi tuli Sohva.

Sohva-palvelun kehittämisen innovatiiviset lähtökohdat

Kun olimme onnistuneet Fujitsussa hyvin perustietotekniikkapalvelun kehittämisessä 1990-luvulla, mietimme miten kehitämme innovatiivisen palvelun sovellusmarkkinoille. Käynnistimme kehitystyön laatimalla tällä palvelulle haastavan ja energisoivan visio. Visioksi määritelimme Riittäisikö puoletkin? Tämän kyseenalaistamisen taustalla oli tietous siitä, että yrityksen sovelluskustannuksista menee valtaosa vanhojen sovellusten ylläpitoon, jolloin uusien liiketoimintaa paremmin tukevien sovellusten kehittämiseen ei jää aikaa eikä rahaa riittävästi.

Seuraavassa vaiheessa täsmensimme vision tarkentamalla, mitkä sovellusten kehittämiseen ja käyttöön kuuluvat asiat voidaan kyseenalaistaa ja mahdollisesti puolittaa. Näitä puolitettavia asioita voivat olla esimerkiksi:

- sovellusten lukumäärä
- sovellustoimittajien lukumäärä
- sovellusten ylläpitokustannukset
- käyttäjien tukipyyntöjen määrä

Fujitsun visiointi tuottaa uusia palveluja



Kehitystyön seuraavassa vaiheessa mietimme, miten tällainen uusi innovatiivinen sovelluspalvelukonsepti kehitetään. Tavoitteenamme oli kehitystyössä yhdistää Fujitsun arvojen mukainen toimintatapa – Ole luova ja toimi kurinalaisesti. Haimme apua kehitystyön lähestymistapaan perinteisen projektityömallin lisäksi mm. ITIL:n versiosta V3, jossa on kuvattu palvelukehityksen elinkaarimalli palvelustrategiasta palvelun jatkuvaan parantamiseen. Toinen ajatuksia herättävä lähde oli Saara Braxin esittämä Palvelutuote systeeminä lähestymistapa. Sen mukaan innovatiivista palvelutuotetta kehitäessä tulee kiinnittää riittävästi huomiota palvelun lopputuloksen

määrittelyyn asiakkaan näkökulmasta. Muut var-
teenotettavat näkökulmat ovat palvelukonsepti,
palveluprosessi ja sen vaatimat resurssit. Saara
Braxin mukaan innovatiivisuus voi kohdistua sekä
palvelukonseptiin että sen yksittäisiin osiin.

Koska kyseessä oli innovatiivinen kehittäminen,
kiinnitimme erityistä huomiota yhteisen näkemyk-
sen muodostamiseen. Kokemuksiemme mukaan
yhteinen näkemys syntyy parhaiten vuoropuhelu-
jen avulla. Toinen kehitystyötä ohjannut keskeinen
ajatus oli kehitystyön versioajattelu ja pidemmän
aikavälin kehitysohjelman laatiminen.

Fujitsun parhaiden käytäntöjen hyödyntäminen Sohva-palvelun kehitystyössä

Visionamme Riittäisikö puoletkin? on Lean-
filosofian mukainen kyseenalaistami-
nen. Koska Toyota on Fujitsu Japanin
merkittävä asiakas, Fujitsu on
hyödyntänyt Toyotan asiakkuu-
desta saamia oppeja ja
kokemuksia menetelmien
ja palveluiden kehi-
ttämisessä. Fujitsu
UK:ssa on kehitetty
jatkuvaan paran-
tamiseen perustuva
toimintamalli Sense and
Response, jonka Fujitsun
suomalaiset asiakkaat tunte-
vat nimellä Tunto. Fujitsun liike-
toimintalähtöinen tietojärjestelmien
kehittämisen menetelmäkokonaisuus on
nimeltään Macroscopic, jossa on menetelmät
mm. hankesalkun johtamiseen, ict-arkkiteh-
tuurin laatimiseen ja kehityshankkeiden hyöty-
jen varmistamiseen. Patja-konseptin osana oleva
käyttäjien valmennuskonsepti TRIM, sopii myös
sovellusten käyttäjien valmennukseen.

Vanhan sanonnan mukaan morsiamen hääpu-
vussa täytyy olla jotain vanhaa, jotain uutta, jotain
sinistä ja jotain lainattua. Kokosimme samalla
periaatteella Sohva-konseptin keskeiset osat ja
muokkasimme niistä asiakkaille ehyen kokonai-
suuden muodostavan innovatiivisen sovellusten
ulkoistamispalvelun.

Jaoimme palvelukonseptin seuraaviin osiin:

- kumppanuuden johtaminen
- sovellushallinta
- muutosjohtaminen
- kehitys
- palveluintegrointi

Sohva-palvelun pääelementit

Kumppanuuden johtaminen on Sohva-palvelun
johtamismalli, jossa asiakkaan ja Fujitsun johto
asettavat yhteiset tavoitteet ja mittarit kumppa-
nuudelle. Kumppanuutta johdetaan kehitys- ja
palvelusalkkujen avulla.

Sovellushallinnan keskeiset palvelut ovat käyt-

täjätuki, sovellusten ylläpito ja sovellusten pien-
kehitys.

Muutosjohtamisen palvelut auttavat molempien
osapuolten johtoa, kehityshenkilöitä ja sovel-
lusten käyttäjiä ymmärtämään ja hyväksymään
yhteisesti sovitut muutokset sovellusten kehittä-
misessä ja niiden hyödyntämisessä.

Kehityspalvelut liittyvät sekä kumppanuuden
kehityssalkkuun että siihen kuuluviin yksittäisiin
kehityshankkeisiin. Päivi Sirviö on kuvannut toi-
saalla tässä lehdessä Fujitsun käyttämän sovellus-
kehitysprosessin.

Palveluintegrointi yhdistää Fujitsun ja kolman-
sien osapuolien tuottamat sovelluskehityspalvelut.
Integroituja palveluja johdetaan palvelusalkun
avulla.

Fujitsun palvelukehitys ja palveluprosessit saavat julkista tunnustusta

Helsinki 19.8.2008 —

Tieto- ja viestintätekniikan palvelutoimittaja

Fujitsu Services Oy:

lle on myönnetty

it-palveluiden johta-

misen ja hallinnan ser-

tifikaatti, joka on Suomen

ensimmäinen ja ainoa. Huh-

tikuussa 2008 Fujitsu sai tämän

sertifikaatin Patja-asiakkuudelle,

nyt sertifikaatti laajeni perustietotek-

niikkapalvelujen lisäksi myös sovelluksiin,

eli Sohva-palveluun.

ISO/IEC20000-1:2005 on ensimmäinen kansainvälinen standardi tietotekniikkapalvelujen johtamiseen ja hallintaan. Standardi edellyttää yksityiskohtaista näyttöä palvelujen hallinnan laadukkaasta ja tehokkaasta toimivuudesta. Vaatimuksia on peräti 170. Standardi ottaa huomioon myös asiakkaiden muuttuvat liiketoimintatarpeet.

Mikä standardi?

ISO/IEC20000-1:2005-standardi on ensimmäinen kansainvälinen standardi tietotekniikkapalvelujen johtamiseen ja hallintaan. Standardi edellyttää yksityiskohtaista näyttöä palvelujen laadukkaasta ja saumattomasta toimivuudesta.

ISO20000-standardi on linjattu yhteensopivaksi Itilin (IT Infrastructure Library) kanssa. Itil on kokoelma it-palvelujen tuottamisen parhaita käytäntöjä ja menetelmä toiminnan kehittämiseen. Fujitsu on soveltanut Itiliä jo useiden vuosien ajan.



Päivi Sirviö toimii kehityspäällikkönä Fujitsu Services Oy:ssä. Erityisen mielenkiinnon kohteina hänellä on menetelmäkehitys.

Monoliittisten prosessien aikakausi on ohi



Projektit ovat erilaisia ja niin ovat myös menetelmätarpeet. Tämä on kautta aikojen ollut menetelmäkehityksen ongelma. Ivar Jacobson on kehittänyt lääkkeeksi Essential Unified Prosessin (EssUP). EssUPin tehtävät ja tuotokset on palasteltu itsenäisiksi moduuleiksi, käytännöiksi, jotka toimivat prosessien rakennuspalikoina. EssUP tarjoaa mahdollisuuden ratkaista Sohvan sovelluskehityksen haasteet – miten yhtenäistää ja tehostaa toimintaa kun sovellukset ja menetelmät vaihtelevat suuresti. Fujitsun lean-johtamistapa luo omalta osaltaan valmiudet tähän.

Lean on työntekijöitä arvostava ja motivoiva johtamistapa

Fujitsun Sohva on palvelumalli, jossa asiakas luovuttaa sovelluksensa Fujitsun hoitoon. Fujitsun lupaa hoitaa tehtävää ajatuksella Riittäisikö

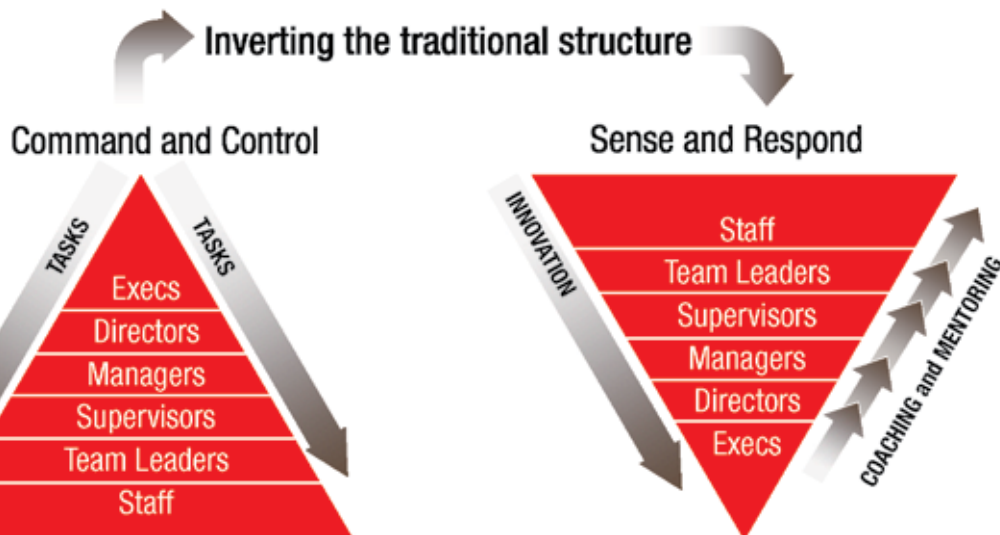
puoletkin? Sohva-palveluun tulee hyvin erilaisia järjestelmiä ja toimintamalleja, jotka tulee mahdollisuuksien mukaan yhtenäistää. On keskityttävä olennaiseen, karsittava turha ja kehityttävä jatkuvasti.

Riittäisikö puoletkin on leania johtamistapaa. Lean tarkoittaa laihaa, kevyttä, notkeaa, ketterää joustavaa ja kilpailukykyistä. Se korostaa jatkuvaa parantamista ja muutosvalmiutta. Tuloksena on laatua ja kustannustehokkuutta. Lea-johtamistapa saattaa ensin kuulostaa tyllyltä, mutta pohjimmiltaan se on humaani ja työntekijöitä motivoiva johtamistapa. Avainasemassa prosessien kehittämisessä ovat työntekijät. Johtamistapa kääntää perinteisen tutun organisaatiokolmion pääläelleen. Johtajat ovat kolmion pohjalla, tukien ja mahdollistaen sen, että työntekijät pystyvät itse toimimaan prosessien kehittäjinä ja ongelmanratkaisijoina.

Lean haastaa sovelluskehitystä

Lean on alun perin kehitetty Toyotan autoteollisuuden prosessien parantamiseen. Fujitsu on keksinyt soveltaa johtamistapaa myös palvelujen tuotantoon. Lean sopii sovelluskehitykseen, mutta asettaa haasteita sovelluskehitysmenetelmille.

Leania tukevien sovelluskehitysmenetelmien täytyy olla ymmärrettäviä, käytettäviä ja nopeasti uusiin tilanteisiin sovitettavia. Menetelmän ylläpidon on oltava tehokasta. Lisäksi menetelmien on tarjottava johdolle näkyvyyttä ja valmiudet seurata monimuotoisten projektien etenemistä. Systeemyö- ja



projektinhallinnan työvälineiden tulee muodostaa saumattomasti ja tehokkaasti toimiva kokonaisuus, joka tukee menetelmän käyttöä ja automatisoi tekemistä.

Sovelluskehitykseen on monia lähestymistapoja. Leanin tavoitteiden saavuttamista tukee ketterä sovelluskehitys. Siinä on paljon yhteistä leanin kanssa. Ketterän lähestymistavan perusajatus on vastata muutoksiin nopeasti ja jopa hyötyä muutoksista. Tavoitteena on saada näkyvyyttä prosessiin, jotta muutostarpeet huomataan ajoissa. Ketteryys keskittyy olennaiseen. Sovelluskehityksessä se tarkoittaa toimivaa koodia. Ketteryys vaatii kommunikointia ja tekee työntekijöistä leanin tapaan työn suunnittelun ja työtapojen kehittämisen avainhenkilöitä. Leaniin pyrittäessä sovelluskehitysmenetelmien olisi siis hyvä tukea ketterää lähestymistapaa.

EssUP rakentuu käytännöistä

Fujitsun valitsema sovelluskehitysmenetelmä on Essential Unified Process (EssUP). Menetelmän taustalla on Rational Unified Prosessin (RUP) modernit sovelluskehityksen perusajatuksat, mutta selkein parannuksina RUPiin verrattuna on EssUPin parempi ymmärrettävyys, käytettävyys, laajennettavuus ja ylläpidettävyys. EssUP sisältää myös ketterän lähestymistavan peruskäytännöt.

EssUP on joukko itsenäisiä moduuleita, käytäntöjä, jotka toimivat prosessien rakennuspalloina. Kukin käytäntö sisältää tuttuja prosessin elementtejä: lopputuotokset, tehtävät tuotosten tekemiselle, tarvittavat kompetenssit ja mallit tekemiselle. Käytännöt keskittyvät johonkin sovelluskehityksen näkökulmaan tai lähestymistapaan, esimerkiksi käyttötapauskeskeiseen vaatimusmäärittelyyn. Käytäntöjen avulla voi helposti määrittää vaihtoehtoisia lähestymistapoja saman asian tekemiseen.

Vaatimusmäärittelyn vaihtoehtoja ovat esimerkiksi perinteinen vaatimusmäärittely tai käytötapauksiin pohjautuva vaatimusmäärittely. Projektinhallintaan liittyviä vaihtoehtoisia käytäntöjä ovat esimerkiksi iteratiivinen projektinhallinta tai vesiputousmalliin pohjautuva projektinhallinta.

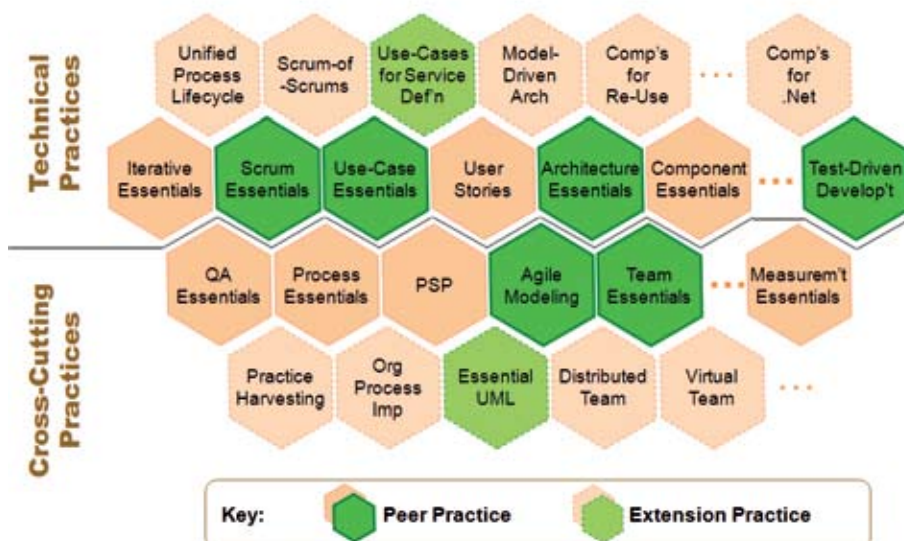
Käytännöt sisältävät myös tuotostensa laadunvarmistuksen. Esimerkiksi käyttötapauskäytäntö määrittää käytötapauksen tekemisen ja niihin pohjautuvan järjestelmätestauksen. Näin käytännöt ovat itsenäisiä moduuleita, joita voi halutessaan käyttää toisistaan riippumatta. Tämä mahdollistaa menetelmän laajentamisen itsenäisiä käytäntöjä kehittämällä – muuhun osaan menetelmää ei tarvitse koskea. Menetelmän ylläpito helpottuu, koska yksittäiseen käytäntöön kohdistuvat muutokset eivät vaikuta menetelmän muihin osiin.

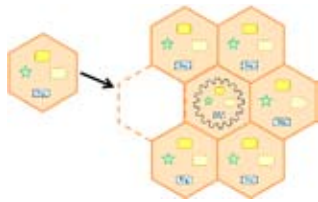
Prosessien rakentaminen käytännöistä on nopeaa ja vaivatonta

EssUPin perusajatus on rakentaa prosessi käytännöistä. Näin prosessiin voi poimia juuri tilanteeseen soveltuvat käytännöt. EssUPissa prosessien rakentamiselle luo pohjan EssUP-kehikko (kernel). EssWork on työväline prosessien rakentamiseen. Prosessien rakentaminen on käyttäjän näkökulmasta hyvin yksinkertaista – käyttäjä poimii EssWorkin alusvetovalikosta haluamansa käytännöt ja EssWork luo prosessin. EssWork on työkalu

Lean tarkoittaa notkeaa, joustavaa ja kilpailukykyistä.

EssUP on joukko itsenäisiä prosessin rakennuspalloja, käytäntöjä.





Käytännöistä rakennetaan prosessi liittämällä käytännöt toisiinsa EssUP-kehikon avulla.

myös projektin tehtävien suunnitteluun. Käyttäjä valitsee haluamansa prosessin, tuotokset ja kunkin tuotoksen tarkkuustason. EssUPin tarjoamista vaihtoehdoista. EssWork luo valintojen pohjalta projektin tehtävät. Prosessien kehittäminen ja säätäminen projektille sopivaksi on joustavaa ja nopeaa. Tämä on merkittävä ero esimerkiksi RUPiin, jossa käytännöt on tiiviisti integroitu.

Projektien työn suunnittelu ja seuranta yhtenäistyy

Sovelluskehitysprojekteille on tunnistettavissa yleiset seurattavat kokonaisuudet, jotka ovat riippumattomia systeemyön tekemisen tavasta. EssUPissa nämä on tunnistettu ja jokaiselle kohteelle on määritelty elinkaari. Ne on sisällytetty EssUP-kehikkoon ja yhdistetty käytäntöjen mukaan tuomiin konkreettisiin lopputuotoksiin. Tämä rakenne mahdollistaa projektien yhtenäisen seurannan. Siten EssUP-projektissa projektipäällikkö

pystyy suunnittelemaan ja seuraamaan projektin sisällön tuottamista, vaikka ei ole asiantuntija eikä tunne käytännön työtapaa ja tuotosten sisältöä. Kommunikointi projektiryhmän kanssa helpottuu. Erilaisten projektien seuranta on yhtenäistä myös johto- ja ohjausryhmälle vaikka itse työn tekemisen tapa ja tuotosten sisältö vaihtelisivat. Projektien hallinta tehostuu, ongelmakohdat ja projektin todellinen tilanne nähdään aikaisemmin.

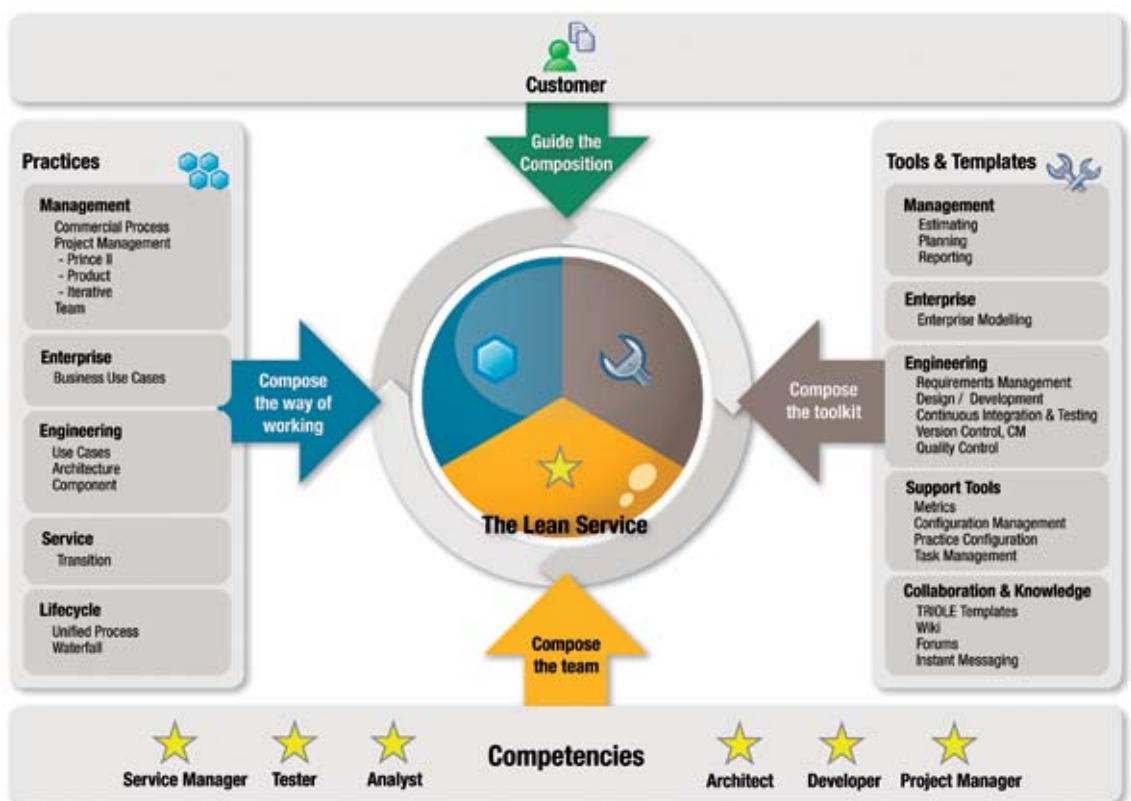
Systeemyön ja projektinhallinnan välineet integroitu

Projektien työvälineet valitaan projektin tarpeista riippuen. Fujitsu on integroinut systeemyön ja projektinhallinnan työvälineet kokonaisuudeksi sekä Java- että Microsoft-kehitykseen. Välineet on sovitettu EssUPiin ja ne ohjaavat ja automatisoivat työn tekemistä ohjeiston mukaisesti. Näin työ tehostuu, yhtenäistyy ja helpottuu.

EssUP-kehikko tarjoaa yhtenäisen näkymän tekemiseen vaikka tekemisen tavat vaihtelevat projektista toiseen kehikon avulla.



Fujitsun ratkaisu lean-johdantavan toteuttamiseen sovelluskehityksessä.



Mittaaminen kertoo totuuden

Suurten ja keskisuurten yritysten ICT-palvelun laadun mittaaminen jää liian usein vain palveluita toimittavien yritysten harteille. Onko pukki oivallinen kaalimaan vartija, vai tulisiko asiakkaan myös itse mitata ostopalvelunsa laatua?

Palvelutasosopimusten räätälöiminen – mahdollistako?

Suomessa ICT-palveluita tarjoavat yritykset ovat jo pitkään kehittäneet palvelun laatuun liittyviä prosesseja ja mittareita. Hinnan sijaan toimittajat ovat alkaneet entistä enemmän kilpailla palveluiden laadulla. Harvassa ovat nykyään ne ICT-palveluiden toimittajat, joilla ei ole vielä valmiita, palvelun laatuun liittyviä palvelutasosopimuksia (SLA, Service Level Agreement), jotka automaattisesti liitetään solmittuun sopimukseen.

Toimittajien valmiita palvelutasosopimuksia leimaa sama ongelma kuin pitkälle paketoituja tuotekokonaisuuksiakin: ne eivät välttämättä sovi kaikille palveluita ostaville yrityksille. Yleensä valmiit, tuotteistetut palvelutasosopimukset sopivat pienyrityksille, puhtaalta pöydältä aloittaville ja peruspalveluiden ostajille. Riippuu pitkälti asiakasyrityksen koosta ja myyjän sekä ostajan neuvottelutaidoista, kuinka paljon valmiita laatumääreitä voidaan räätälöidä, millaisilla ehdoilla ja ennen kaikkea, kuinka paljon se maksaa.

Palveluun laatuun liittyviä sopimuksen osia ei kannata hankintavaiheessa ohittaa. Edullinen palvelun ostohinta voi olla merkityksettömän pieni säästökohde, jos yrityksen ydintoiminta kärsii tai keskeytyy tietoteknisten ongelmien takia.

Palvelutasosopimus kuvaa tuotettavan palvelun vaatimukset

Palvelutasosopimus on myyjän ja ostajan välinen sopimus, jossa määritellään tuotettavan palvelun vaatimustasot. Vaatimustasot, eli tavoitearvoparametrit (SLO, Service Level Objective) määritellään sellaisiksi, että ne on mahdollista saavuttaa ja todentaa ja että ne tukevat ostajan liiketoimintaa.

Näillä tavoitearvoparametreilla konkretisoidaan sopimuksessa sovittu palvelun laatu. Tavoitearvoparametrien täyttymättömyydestä seuraa yleensä rahallisia sanktioita toimittajalle. Esimerkiksi tie-

toliikenneverkossa SLO:t voivat käsittää läpäisykyvyn, käytettävyyden, kulkuaikaviipeen, viipeen vaihtelun, pakettihävikin ja niin edelleen.

Kerran luotua palvelutasosopimusta pitää voida päivittää sopimuskauden aikana ostajan tarpeiden mukaan. On suorastaan utopistista kuvitella, että esimerkiksi viiden vuoden sopimuskaudelle tehty sopimus ei kaipaisi päivitystä suorituskykyparametrien osalta. Sopimuksen kirjoitushetkellä ei ehkä osata huomioida kaikkia tulevia tarpeita. Kenties vielä yleisempää on, että asiakkaan toimintaympäristöön tulee uusia palveluita, jotka asettavat uusia vaatimuksia ns. alemman kerroksen palveluille. Esimerkiksi tietoliikenneverkko on alemman kerroksen palvelu, jota käyttävät useat eri järjestelmät.

Palvelutasosopimusten yhteensovittaminen

Nykyään palvelutasosopimukseen kirjataan yleensä eritasoisia luokkia asiakkaan erilaisista tarpeista riippuen. Tietoliikenteen tapauksessa voidaan erotella esimerkiksi toimipisteittäin ne liittymät, jotka tarvitsevat tiukempia tavoitearvoja kuin toiset. Luokkien käyttäminen keventää myös palvelutasosopimuksen päivitystarvetta, koska sopimuskauden aikaisiin muutoksiin voidaan varautua paremmin.

Sen lisäksi, että eri palveluille määritellään yksittäiset palvelutasosopimukset, huomiota pitää kiinnittää myös sopimusten yhteensopivuuteen. Yhteensopivuuden lisäksi niitä palveluita, jotka koostuvat useasta eri toimittajan toteuttamasta palvelukomponentista, tulee tarkastella järjestelmätasolla. Järjestelmätason tarkastelussa verrataan järjestelmän käyttämien osakokonaisuuksien laatumääreiden soveltuvuutta keskenään. Esimerkiksi järjestelmän ylläpidon tapauksessa tulee selvittää vähintään tietoliikenneverkon, palvelinlaitteiston, varusohjelmistojen ja käyttöjärjestelmätoimittajan kanssa solmitut palvelutasosopimukset. Kaikkien laatumääreiden tulee olla yhtäläisiä, keskenään yhteensopivia ja toisiaan tukevia.

Esimerkiksi konesalin tietoliikenneverkon vikaantuminen voi lamauttaa koko palvelun toiminnan. Silloin on ensiarvoisen tärkeää, että tietoliikenneverkon toimittajan kanssa solmittu



Kari Juopperi toimii konsulttina Noval Networksissa. Hän on keskittynyt työssään palvelutasonhallintaan ja ICT-palveluiden kilpailuttamiseen.

”On suorastaan utopistista kuvitella, että esimerkiksi viiden vuoden sopimuskaudelle tehty sopimus ei kaipaisi päivitystä suorituskykyparametrien osalta.”

palvelutasosopimus takaa verkon tukitoimet, jotta palvelun toiminta saadaan palautettua normaalille tasolle mahdollisimman nopeasti.

Integraattori vai monitoimittajaympäristö?

Osa yrityksistä on pyrkinyt poistamaan riskin SLA-sopimusten yhteensopivuusongelmista ja toimittajien kaitsemisesta valitsemalla ICT-toimittajaksi yhden integraattorin. Tämä ei kuitenkaan automaattisesti poista SLA-sopimusten ylläpitovelvollisuutta, yhteensovittamista tai palvelun laadun seuraamista.

Vastuukysymykset ovat integraattorin kanssa kuitenkin yksinkertaisempia, koska mahdollisen vian korjaaminen on aina tämän (tai tämän käytämien alihankkijoiden) vastuulla. Jos integraattoria käytetään, voidaan palvelusopimukseen kirjata palvelusovaatimukset komponenttikohtaisten tavoitearvoparametrien sijaan loppukäyttäjän kokemaa laatua kuvaavan päästä-päähän-menetelmän mukaisesti.

Valvo, mittaa ja pyri jatkuvaan parantamiseen

Oli kyse sitten integraattorista tai useasta erillisestä toimittajasta, palvelutasosopimuksen osana pitää määritellä yksiselitteisesti siihen kirjattujen SLO:iden todentamis- eli mittausmenetelmät. Mittausmenetelmät pitää määritellä tavoitearvojen ohella niin yksiselitteisesti, ettei niistä tule myöhemmin kiistaa sopimusosapuolten välillä. Mittausmenetelmät pyritään määrittämään mitalaite riippumattomasti, eli mittauksia tulee voida tehdä useilla eri valvontatyökaluilla.

Tarkoilla, soveltuvilla ja räätälöidyillä tavoitearvoilla ei ole mitään merkitystä, jos niiden toteutumista seuranta on määritelty epätarkasti,

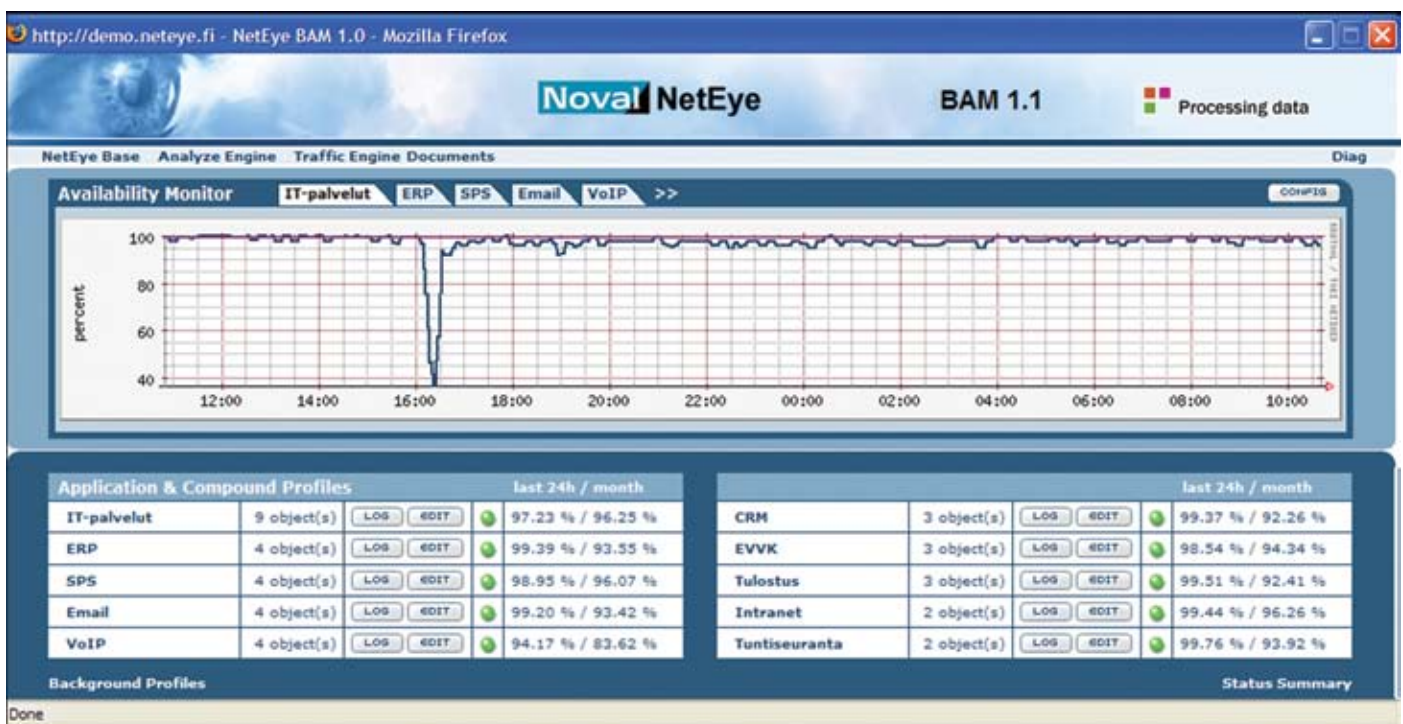
asiakkaan kannalta epäedullisesti tai muuten puutteellisesti. Suositeltavaa on, että toimittajalla ja ostajalla olisi omat, erilliset valvontaratkaisut. Toimittaja käyttää oman valvonnan tuottamaa tietoa operatiivisen toiminnan lisäksi myös sanktioiden maksuperusteena. Asiakkaan suorittama valvonta puolestaan varmistaa toimittajan valvonnan oikeellisuuden. Oma valvonta toimii myös jatkuvaan parantamiseen pyrkivän tietohallinnon kehityskohteiden paikallistamisessa sekä tarjoaa työkalut palvelutasonhallinnalle (SLM, Service Level Management).

Kokonaiskäytettävyyttä kuvaava, valvontajärjestelmän laskema lukuarvo antaa faktista tietoa investointien ja korjaustoimenpiteiden havainnollistamiseen niin tietohallinnon omaan käyttöön, kuin investointirahasta päättävillä portaille. Asiakkaan omaa valvontamenetelmää voidaan myös helposti päivittää ja muuttaa sopimuskauden aikana. Näin ollen puutteelliseksi havaittua valvontaa voidaan jatkuvasti kehittää ja syventää.

Vaikka asiakkaan oman valvontajärjestelmän havaitsemia poikkeamia laatutasossa ei ole palvelutasosopimuksessa sanktioitu, voidaan havaittu poikkeama nostaa esiin toimittajan ja asiakkaan välisissä palaverissa. Jos myyjä-ostaja-suhteesta on päästy aitoon kumppanuuteen, esiin nostettu poikkeama ei jää selvittämättä.

Näiden asioiden lisäksi asiakkaan oma valvonta voidaan virittää yksittäisten sopimuskuoppa-ten tuottamien palveluiden läpi ja näin mallintaa loppukäyttäjän kokema palvelun laatu päästä-päähän-periaatteen mukaisesti. Tällöin mittauksilla voidaan saavuttaa myös liiketoiminnan näkökulma palveluiden toimivuuteen.

BAM-käyttöliittymän päänäkyvä. Ylhäällä valvontaan määritellyt järjestelmät, keskellä kokonaiskäytettävyyden kuvaaja ja alla seurattavien järjestelmien tila- ja käytettävyyssiedot.



Palvelun laadun havainnollistaminen

Reaalimaailmassa asiakkaan oma valvonta on usein puutteellista tai sitä ei välttämättä ole lainkaan. Todennäköisesti suurin syy juontaa juurensa pelkoon omien resurssien riittämättömyydestä.

Yritykset usein luulevat, että oma valvonta on vielä 2000-luvullakin lasikoppivalvontatyyppistä mittarien tuijottamista. Nykyään kuitenkin esimerkiksi sähköpostilla tai tekstiviestillä lähetettävät automaattihälytykset korvaavat henkilöstön valvontatietojen jatkuvan tarkkailun. Noval Networks on kehittänyt oman valvontaohjelmistonsa, Noval NetEye -tuoteperheen, jonka ajavana voimana on ollut käytön helppous ja mittaustapojen monipuolisuus sekä valvontatiedon visuaalinen havainnollistaminen.

NetEye-tuoteperhe koostuu useasta eri ohjelmistokomponentista. Komponenttirakenteen ansiosta asiakas saa ympäristöönsä parhaiten soveltuvat valvontaratkaisut komponenttien perusvalvonnasta aina asiakkaan loppukäyttäjää mallintavaan, päästä-päähän-valvontaan. NetEye:stä tekee poikkeuksellisen sen BAM-ohjelmiston (Business Application Monitor) mahdollistama graafinen, nopeasti sisäistettävä näkymä valvonassa olevien palveluiden toimivuudesta, eli kokonaiskäytettävyydestä. Kokonaiskäytettävyyttä kuvataan prosenttiluvulla nolasta sataan. Nolla tarkoittaa tarkasteltavan palvelun tai palvelukokonaisuuksien täyttä toimimattomuutta ja 100 % puolestaan täyttä toimivuutta.

Palvelujen toimivuus voi tarkoittaa palvelutasosopimuksiin kirjattujen tai asiakkaan kanssa määriteltyjen tavoitearvojen toteutumista. Joissakin tapauksissa halutaan seurata molempia erikseen.

BAM:in ideana on tarjota ajan funktiona piirtyvä kuvaaja seurattujen järjestelmien ja niiden komponenttien toteutuneesta laadusta. Graafi piirretään NetEye-valvontaan määriteltyjen valvontakohteiden, niiden tavoitearvojen ja palvelun kokonaiskäytettävyyteen vaikuttavien painokertoimien avulla. Mittaustavalla päästään aitoon päästä-päähän-mittaukseen, joka useimmissa tapauksessa kattaa usean eri toimittajan palvelukomponentit saumatomasti. BAM:in hierarkkisen rakenteen avulla havaittuun poikkeamaan voidaan pureutua syvemmälle. ICT-palvelun tapauksessa voi-

daan helposti selvittää vian ajankohta, kesto ja vikaantuneen järjestelmäosan tiedot. Samalla havaitaan minkä tahon vastuualueella vika on. Vian vaikutus kokonaiskäytettävyyteen riippuu sen painoarvosta, eli siitä kuinka paljon vikaantunut komponentti vaikuttaa koko palveluun.

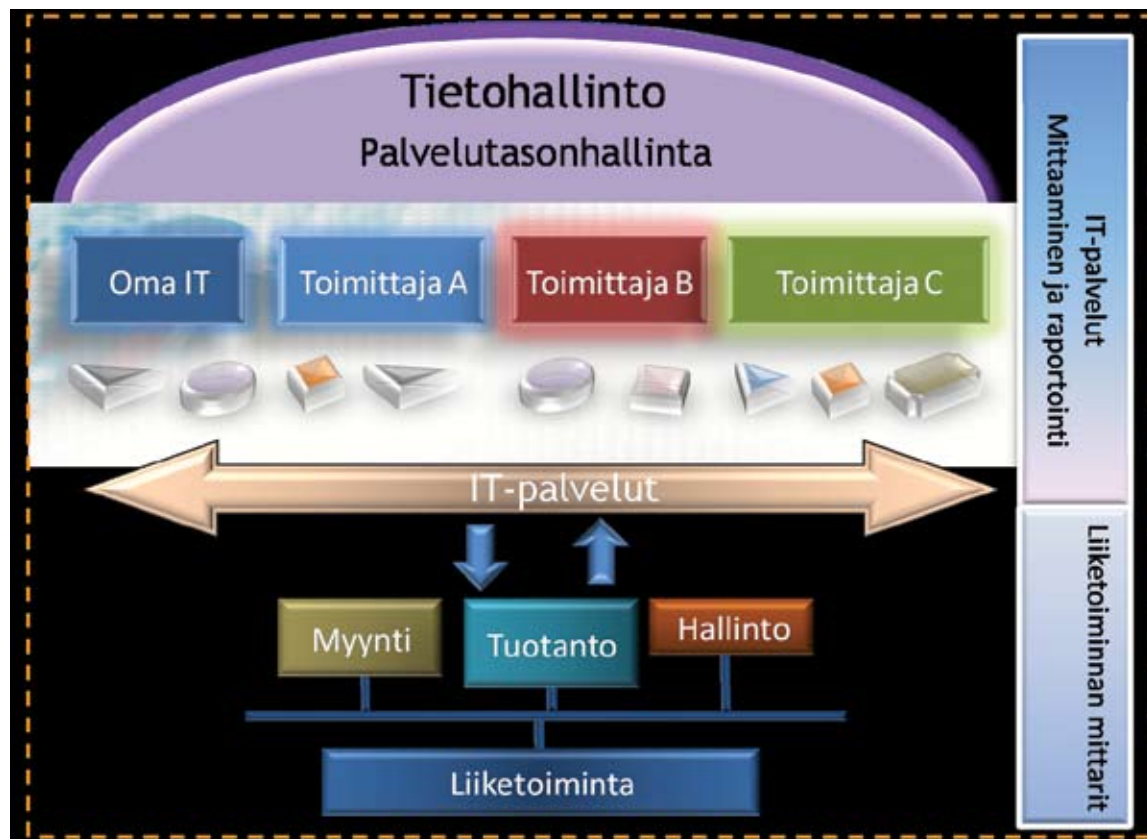
Palvelutasonhallinnan näkökulma

Palvelutasonhallinta on jatkuva prosessi, jonka tavoitteena on varmistaa palvelutasosopimukseen kirjattujen tavoitearvojen toteutuminen ja jatkuvasti kehittää palvelun laatua sen kaikilla osa-alueilla. SLM-prosessin tulee yksittäisten palvelutasosopimusten määrittelyn ja ylläpidon lisäksi varmistaa sopimusten yhteensopivuus järjestelmätasolla. Valvonta ja mittaaminen ovat oleellisia osia palveluiden kehittämisessä. On käytännössä mahdotonta kehittää palvelun suorituskykyä, jos käytössä on ainoastaan subjektiivisia käyttäjien kokemuksia.

ICT-palveluiden hallinta ja viilaaminen on siinä mielessä epäkiitollista puuhaa, ettei työ sen parissa koskaan lopu. Onnistunut palvelutasonhallintaprosessi, jonka osana soveltuvat työkalut, laadukkaat palvelutasosopimukset ja hyvät toimittajakumppanit tekevät siitä kuitenkin huomattavasti helpompaa, tehokkaampaa ja vaivattomampaa.

NetEye järjestelmän BAM on artikkelin kirjoitushetkellä patentoimisprosessissa.

Palvelutasonhallinnan periaate. SLM-prosessin tavoitteena on varmistaa eri palveluiden toimittajien palvelutasosopimusten soveltuvuudet keskenään ja erikseen. Liiketoiminnan seuraamisesta tutut mittarit on tuotu kuvaamaan ICT-palveluiden toimivuutta.





Paula Miinalainen,
yrittäjä ja konsultti
Arborvitae Oy

Valtionhallinnon ostotoimen ratkaiseminen

Valtionhallinto ratkaisee tulevaa työvoimapolua innovatiivisilla sähköistämiskäytännöillä.

"Valtiolta jää eläkkeelle tulevina vuosina niin paljon ihmisiä, että on pakko tehdä innovatiivisia ratkaisuja, jotta pystytään asiat hoitamaan. – Tämä ohjaa sähköistämään niitä toimintoja, joissa se on mahdollista toteuttaa. Työtä riittää." kertoo järjestelmäasiantunija Kristiina Seppälä Valtionkonttori Hallinnon ohjauksesta.

Valtiolla työssäkäyvien keski-ikä on yksityispuolta korkeampi. Valtio joutuu nyt ja entistä enemmän tulevaisuudessa taistelemaan pienestä, niukasta työvoimasta julkishallinnon ja yksityispuolen kanssa. Työt monipuolistuvat sähköistämisen myötä." jatkaa Kristiina Seppälä.

Valtion on sähköistämässä tilaustenhallintaa osaksi jo sähköistettyä laskujen välityspalvelua ja sähköistä taloushallintoa. Myös muita sähköistämishankkeita on meneillään.

1) Tulevaa henkilöresurssipulaa ennakoidaan valtiolla keskittämällä talous- ja henkilöstöasioiden hoitoa palvelukeskuksiin. Valtion talous- ja henkilöstöhallintoa hoidetaan viidessä palvelukeskuksessa.

Palvelukeskus

Sisäasiain palvelukeskus
Puolustushallinnon palvelukeskus
Oikeushallinnon palvelukeskus
Valtiokonttori
Yliopistojen palvelukeskus

Sijainti

Joensuu
Hämeenlinna
Hämeenlinna
Hämeenlinna
Vaasa

2) Prosessit on uudistettu jouheviksi ja kustannustehokkaiksi.

3) VM:n alaisesta, Valtiokonttorin toteuttamasta KIEKU-ohjelmasta saa tukea kehitystehtäviin ja sen puitteissa myös haetaan valtiolle keskitettyjä tietoteknisiä ratkaisuja ja yhteisiä prosesseja.

4) Ensimmäinen talouspuolen järjestelmä uudistus oli ostolaskujen kierrätys- ja arkistointijärjestelmä RONDO. Tämän järjestelmän käyttöönotossa opittiin keskitetyn järjestelmän levityksen rutiinit ja havaittiin keskitetysti hankitun ratkaisun ja siihen liittyvän tuotteistuksen hyödyt.

Kaikki valtion ostolaskut käsitellään ja arkistoidaan tänä päivänä sähköisesti Rondoon. Osa laskuista tulee verkkolaskuina ja osa keskitetyn

skannauspalvelun kautta. Osa skannataan vielä virastoissa, mutta tästä toimintatavasta luovutaan vuoden 2008 loppuun mennessä.

Automaattinen ostolaskujen hyväksyntä

Toistuvien laskujen osalta ollaan siirtymässä laskujen automaattiseen täsmäytykseen siten, että käsittely tapahtuu RONDO-järjestelmässä. Järjestelmään luodaan sopimus, jolle annetaan maksu-aikataulu ja tiliöinti valmiiksi. Laskun saapuessa järjestelmä vertaa laskua sopimukseen. Mikäli lasku ja sopimus täsmäävät, voidaan lasku siirtää suoraan maksuun. Tällaisia toistuvia laskuja ovat esimerkiksi vuokrat, sähkö, vesi, lehdet, leasing, siivous, puhelin jne.

Tilaustenhallinnan sähköistämisen myötä myös tilaukset/tilausvahvistukset ja laskut tullaan täsmäyttämään automaattisesti.

Laskujen välityspalvelu kilpailutettiin kokonaispalveluna

Ostolaskujen vastaanotto

Ostolaskujen vastaanotossa keskeinen haaste oli valtiolle tulevien ostolaskujen kirjo. Niitä tulee lähes 200 000 toimittajalta ja verkkolaskujen osalta laskuttajilla on käytössä eri formaatteja. Tämä ongelma on ratkaistu sillä, että Valtiokonttori kilpailutti laskujen välityspalvelun kokonaispalveluna, jossa välittäjä tekee tarvittavat formaattimuunnokset ja valtiolle voi näin ollen lähettää laskuja eri formaateissa.

Myyntilaskujen sähköistäminen

Myyntilaskujen osalta kilpailutettu välittäjä tekee myös tarvittavat muunnokset laskuaineistoihin, joka helpottaa lähtevien laskujen sähköistämistä. Tämä on tärkeä asia, koska valtiolla on käytössä lukuisia erilaisia laskutusjärjestelmiä. Myös Pankit nähtiin selkein yhteistyökumppaneina myyntilaskujen sähköistämisessä, koska he pystyvät tarjoamaan kanavan kuluttajille vastaanottaa laskuja sähköisesti.

Laskujen välityspalvelun toimittajaksi valittiin Itella Information Oy, joka tuottaa skannauspalvelu, verkkolaskujen vastaanoton, verkkolaskujen lähetyksen ja tulostuspalvelun. Ratkaisuun ollaan oltu tyytyväisiä. Sähköistämistä tehdään operaati-



Kuvassa Kristiina Seppälä.

toreiden ja pankkien kanssa yhdessä. Pankit ovat tehneet mm. puhelinmarkkinointi kampanjoita omille asiakkailleen ja tarjonneet omaa sähköistä laskujen käsittelyjärjestelmäänsä.

Ostolaskuja käsitellään valtionhallinnossa 2,7 miljoonaa kappaletta vuodessa. Tämän vuoden loppuun mennessä niistä tulee olla 60 prosenttia verkkolaskuja. Kaikkien laskujen 100 prosenttisesti tulee olla verkkolaskuja vuoden 2009 loppuun mennessä. Kesäkuussa 2008 sähköisiä laskuja oli 30 prosenttia. Voidaan päätellä, että tavoitteen saavuttamiseksi täytyy tehdä paljon työtä. ”Yhdessä pankkien ja Itellan kanssa pystytään enemmän”, sanoo Kristiina Seppälä.

Myyntilaskujen käsittelyssä siirrytään yhte-näiseen sähköistettyyn järjestelmään

Myyntilaskuja valtion hallinnossa käsitellään vuosittain yli 13.2 miljoonaa kappaletta.

Laskut lähetetään Itellan välityspalvelun kautta. Tavoitteena on, että kaikki laskut ovat sähköistetty vuoden 2010 loppuun mennessä.

”On vanhanaikaista juuttua ole-massa olevaan lainsäädäntöön.”

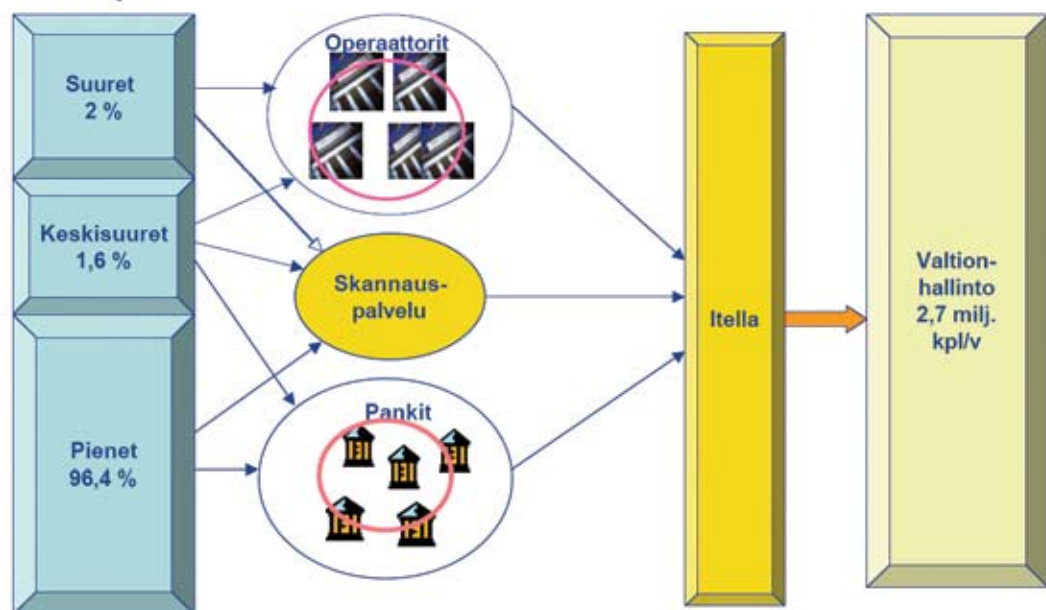
Monet sähköistämiseen liittyvät uudet toimintamallit vaativat lain-

uudistusta. Myyntilaskujen käsittelyn uusimisessa tarvitaan lainmuutosta ajoneuvoveron sähköistä-miseen kuluttajille. Näitä AKE:n ajoneuvoveron laskuja on vuodessa 6,7 miljoonaa kappaletta. Sähköistämiseen tarvittava lakiuudistus on käsit-telyssä.

Kristiina Seppälä kertoo, että heillä Valtiokont-torissa lakiuudistukset nähdään osana kehityspro-jektia. ”Lakimuutoksia käsitellään Arkadianmäellä joka perjantai”, sanoo Kristiina Seppälä pilke sil-mäkulmassa” .

Valtiokonttori
Statskontoret
State Treasury
Toimittajat

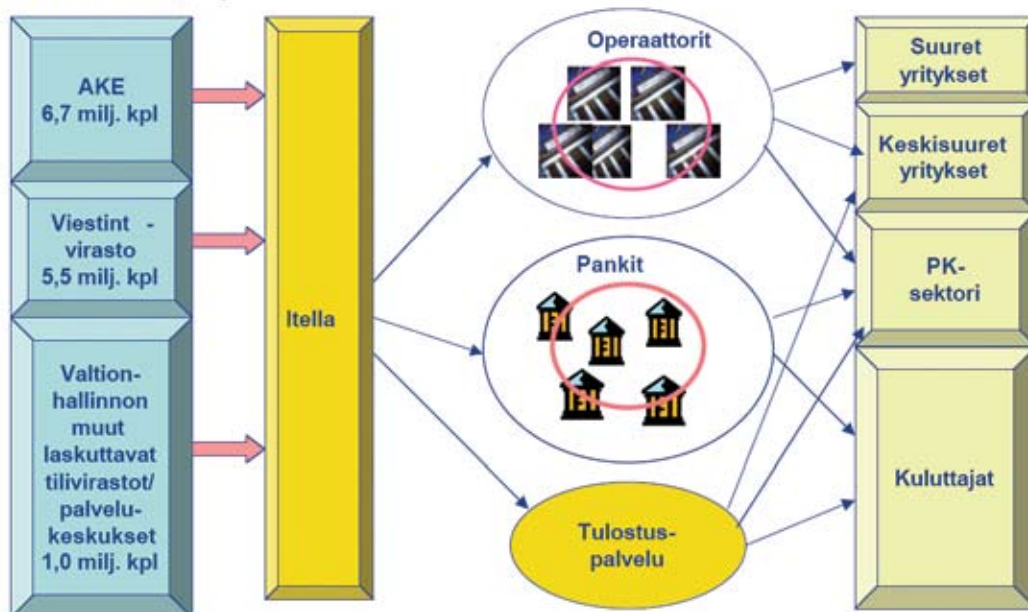
Laskujen vastaanotto



237 600 companies in Finland (2005)

Valtiokonttori
Statskontoret
State Treasury

Laskujen lähetys





Mikko Holmberg
CodeBakers Oy
ICT-arkkitehti
Technology Lead – Java



Perttu Karvinen
CodeBakers Oy
ICT-arkkitehti
Technology Lead – Menetelmät



Joni Moilanen
CodeBakers Oy
IT-konsultti
Technology Lead – Microsoft

Kollaboraatioalustat sovelluskehityksessä – mikä muuttuu?

Perinteisesti sovelluskehitysprosessissa työkalut ovat vastanneet yksittäisistä tehtävistä, kuten projektihallinnasta, ohjelmoinnista, mallinnuksesta, kommunikaatiosta, vaatimusten hallinnasta jne. Ongelmana ei pidetä näiden työkalujen toimintaa vaan niiden yhteistoimintaa. Ohjelmistojen hyvä yhteistoimintakaan ei nykypäivänä riitä vaan mukaan pitää saada tuki prosesseille ja kommunikoinnille. Näihin haasteisiin vastaavat uudenlaiset sovelluskehityksen kollaboraatioalustat.

Johdanto

Kollaboraatioalustoja on ollut tarjolla ja käytössä jo pitkän aikaa. Yleisimmin niillä tarkoitetaan alustoja, joissa toimistosovellukset on integroitu yhdeksi käyttäjäkokemukseksi ja jotka tarjoavat yhteistoimintaan sekä viestintään liittyviä ominaisuuksia kuten sähköpostit, kalenterit, tiedostojen jako, tehtävien hallinta, tavoitettavuustiedot, pika-viestit, blogit, rss-syötteet jne. Tällaisia ovat mm. IBM Lotus - ja MS Exchange/SharePoint -alustoihin pohjautuvat ratkaisut ja ne ovat nykyisin laajassa käytössä yrityksissä. Tällaiset alustat helpottavat ja tehostavat ihmisten jokapäiväistä työtä, mutta eivät tarjoa sellaisenaan tarpeeksi ominaisuuksia sovelluskehitysprojektien tarpeisiin.

Viime vuosina on puhuttu paljon ALM:sta eli Application Lifecycle Managementista. Yleisesti voidaan sanoa, että ALM-järjestelmä:

- koordinoi sovelluskehitykseen liittyviä toimintoja, mm. vaatimusten keräämistä ja analysointia, mallintamista, toteutusta ja testausta
- hallitsee sovelluskehityksen eri toiminnoissa tuotettavia väli- ja lopputuloksia sekä niiden välisiä suhteita, mm. jäljitettävyyttä
- ohjaa käyttäjän toimintaa järjestelmään konfiguroiduilla prosesseilla
- tuottaa raportteja tehtävien edistymisestä ja projektin tilasta

Nykytilanne

Nykyisin yleisesti käytössä olevat ohjelmistot toteuttavat vain pienen osan edellä mainituista kollaboraatioalustojen ominaisuuksista. Joillakin pienemmillä toimittajilla on ratkaisuja, jotka vastaavat artikkelin kollaboraatioalustoja, mutta niiden suosio on vielä ollut marginaalista.

Ohjelmistojen yhteistoiminta on tyypillisesti saatu aikaan löyhällä integraatioilla eli toisesta ohjelmasta päästään ehkä näkemään toisilla ohjelmilla tehtyjä lopputuloksia, mutta ei kenties muuttamaan niitä. Prosessituki on yleensä vain kunkin ohjelmiston sisäistä, ohjelmistojen välille

se ei yllä. Projektinhallintaan liittyvät integroinnit ovat ehkä kaikkein huonoiten hoidettu. Projektin tehtävien hallintaa ja toteutustyötä ei ole yleensä integroitu millään tavalla.

Nykyisten järjestelmien suurin ongelma on se, että ne eivät tue jokapäiväistä työtä. Eri rooleilla ei ole näkyvyyttä toistensa tekemisiin, joten perinteisen kommunikointiin (face-to-face, sähköposti) menee paljon aikaa. Kaikille rooleille, esim. liike-toiminnan edustajalle, ei ole tarjota muuta kuin perinteisiä dokumentteja ja ehkä jotain generoituja web-sivuja yksittäisistä lopputuloksista. Eri vaiheiden lopputulosten linkitystä toisiinsa ei tueta riittävän hyvin.

Ketterien menetelmien ja toimintatapojen käyttöönotto on entisestään lisännyt tarvetta uudentaisille ohjelmistoille, jotka tukevat valittuja menetelmiä ja ohjaavat alusta alkaen tekemistä ja avointa kommunikointia. Tällaisia ohjelmistoja onkin tullut tarjolle viime vuosina ketterien menetelmien suosion kasvaessa, mutta ne ovat tyypillisesti tarkoitettu pienille yksittäisille tiimeille ja keskittyvät lähinnä ketterien tiimien toiminnan tehostamiseen.

Agile-prosessien avuksi löytyy tuotteita, jotka panostavat yhteistyön ja kommunikoinnin helpouteen, mutta eivät varsinaisesti ole sovelluskehitysvälineitä. Joitain näistä sovelluksista tarjotaan myös SaaS-palveluina. Tällaisia tuotteita ovat mm. ThoughtWorksin Mingle, Danuben Scrum Works ja Zimbra Collaboration Suite.

Seuraavaksi esittelemme kaksi erilaista ohjelmistoa, joissa edellä mainittuja ongelmakohtia on ratkaistu: IBM Rational Team Concertin (RTC) ja Microsoft Team Foundation Serverin (TFS). Molemmat ovat uusia tuotteita ja tarjoavat jo aika paljon tavoiteltuja ominaisuuksia. Toki muitakin ratkaisuja on olemassa ja tulossa markkinoille. Esimerkkinä voidaan mainita TechExcelin DevSuite.

Ratkaisuvaihtoehtoja

Rational Team Concert

IBM Rational julkaisu toukokuussa uuden Jazz-ohjelmistoalustansa ja tähän ensimmäisen sovelluksen Rational Team Concertin. RTC tarjoaa työkaluja pääasiassa sovelluskehitysprosessin implementaatiovaiheeseen. Se on lähinnä sovelluskehittäjän ja projektipäällikön tai esimerkiksi Scrum Masterin työkalu.

RTC:iin kuuluvat Eclipse-pohjainen asiakasohjelma, rajoitetumpi web-käyttöliittymä sekä itse Jazz-palvelinohjelmisto. Siihen voidaan integroida

chat-asiakasohjelmia lisäämään kollaboraatio-ominaisuuksia. Se ei itsessään tarjoa työkaluja erillisten dokumenttien hallintaan. Se sisältää myös build-moottorin ja tämän hallintatyökalut, joilla buildeja voidaan hallita ja niiden tuloksia raportoida.

RTC:n toiminta perustuu tehtävien hallintaan. Niitä voidaan osoittaa tietyille käyttäjälle ja aikatauluttaa sitomalla ne hankkeen iteraatioihin. Käyttäjät pitävät yllä tehtävien tilaa ja projektipäällikkö voi kätevästi seurata tehtävien edistymistä. Tehtävien tyypit ja iteraation rakenne riippuu projektin käytössä olevasta menetelmästä. Välineessä tuettuja menetelmiä ovat mm. Scrum ja "Eclipse way", mutta niitä voi toki luoda myös itse. Tavallisimpia tehtävien tyyppejä ovat yleinen työtehtävä, virhe, muutospyyntö ja käyttäjätarina.

Tehtävien käsittely on varsin monipuolista. Käyttäjä voi esimerkiksi liittää tehtävään koodipätkän tai dokumentin ja siirtää sen toiselle käyttäjälle. Lisäksi tehtävien etsintä on helppoa. Käyttäjä esimerkiksi näkee yhdellä silmäyksellä kaikki hänelle määrättyt tehtävät.

Monet toiminnot ovat roolipohjaisia. Esimerkiksi projektipäällikkö voi muokata iteraatiosuunnitelmaa, mutta tavallisella käyttäjältä onnistuu vain sen katsominen. Roolit riippuvat taas käytetystä menetelmästä. Tavallisimmat roolit ovat kehittäjä, projektipäällikkö sekä build-vastaava.

RTC:n käyttöönotto on suhteellisen suoraviivaista. Alkuun pääsee kolmen käyttäjän ilmaisversiolla. RTC on paketoitu ilmaisen Express-C:n lisäksi Express- ja Standard-paketteihin. Lisenssit ovat käyttäjälisenssejä, eikä ohjelmisto vaadi

Kuva 1:
Esimerkki iteraatio-
suunnitelmasta.

The screenshot displays the 'Iteration 1' view in the RTC application. The top section shows the iteration details: 'Team Area: Development Project Team | Iteration: 1.12 (13.8.2008 - 27.8.2008) | 0 Closed | 7 Open'. Below this, there are two main sections: 'Top Items' and 'Defects & Enhancements'.

Top Items: This section lists two items: 'Muuta myyntiartikkelin hintaa' (Medium priority, assigned to Kari Kehittäjä, 40 hours) and 'Lisää myyntiartikkeli' (Medium priority, assigned to Timo Tunari, 37 hours). The progress is 0 / 0 h and estimated at 100%.

Defects & Enhancements: This section lists five items: 'Hintojen tarkistus' (Medium priority, assigned to Timo Tunari, 39 hours), 'Punaisen boksen pitää vilkkua vihreänä' (Medium priority, assigned to Pauli Päällikkö, 34 hours), 'Päivitä tietokanta' (Medium priority, assigned to Kari Kehittäjä, 38 hours), 'Suunnitteluitemi' (Unassigned, assigned to Pauli Päällikkö, 36 hours), and 'Joo, ei mikään taas toimi' (High priority, assigned to Kari Kehittäjä, 35 hours). The progress is 0 / 116 h and estimated at 80%.

At the bottom, there is a table titled 'Found 4 work items - Open items for Iteration 1'.

I	Status	P	S	Summary	Owned By
39	New	Medium	Yellow	Hintojen tarkistus	Developer
38	New	Medium	Yellow	Päivitä tietokanta	Other Developer
35	In Progr...	High	Red	Joo, ei mikään taas toimi	Other Developer
34	New	Medium	Yellow	Punaisen boksen pitää vilkkua vihreänä	Project Manager

muuta maksullisia sovelluksia kuten tietokantoja. Palvelinsovellus toimii Windows- ja Linux-ympäristöissä ja sen lisensointi on Open Source -pohjainen. Express- ja Standard-versiot voidaan asentaa myös WebSphere sovelluspalvelimelle ilmaisen Tomcat-alustan sijasta. Lisäksi ne voidaan konfiguroida käyttämään DB2- ja Oracle-tietokantoja.

Microsoft Team Foundation Server

Microsoft Visual Studio Team System on Microsoftin vastaus ALM-markkinoille. Team System koostuu useasta osasta, jotka ovat sekä palvelin-, että asiakaspäässä. Kaiken ytimenä toimii juuri SP1-versioon päässyt Team Foundation Server 2008, johon työasemat ovat roolista riippuen yhteydessä eri asiakasohjelmilla.

Microsoftin tyyliin kokonaisuus on koottu joukosta valmistajan omia tuotteita, jotka myös integroituvat hyvin yhteen. Tästä syntyvän vahvan riippuvuuden vastapainoksi saadaan toimiva kokonaisuus, jossa projektin eri osapuolet voivat käyttää itselleen tuttuja välineitä.

Team Systemissä prosessi alkaa projektin luonnilla, mikä perustaa SharePointiin automaattisesti projektisivuston, jonne voi koostaa keskitetysti kaikki dokumentit ja muun tarvittavan oheismateriaalin. Mukana tulee myös joukko raportteja, joista näkee silmäyksellä muun muassa projektin etenemisen, bugien määrän ja korjausasteen.

Projektipäälliköt voivat määritellä tehtävät ja vastuuhenkilöt tutulla Excelillä tai Projectilla. Sovelluskehittäjät saavat kyselyä tehtävänsä

lisenssit SharePointiin, SQL Serveriin ja minimivaatimukset täyttäviin Visual Studioihin. Palvelinsovellukset voidaan hajauttaa usealle palvelimelle tai keskittää yhteen, mikäli käyttäjämäärät ovat pieniä. Gold Certified-tason Microsoft-kumppanit tosin säästävät lisenssikustannuksissa huomattavasti.

Uudet haasteet ja mahdollisuudet

Uusien työkalujen myötä saadaan vanhoja haasteita: miten suuri joukko käyttäjiä saadaan sujuvasti käyttämään uutta työkalua tai ympäristöä? Miten koulutus järjestetään ja kuka sen hoitaa? Miten käytössä oleva prosessi taipuu uusille välineille? Kuinka kivulias siirtymävaihe ylipäättensä on? Miten päivitykset hoidetaan? Kollaboraatioalustan valinta johtaa myös jonkinlaiseen vendor-lock-tilaan.

Java-puolella alustat ovat yleensä avoimia ja laajennettavissa. Varsinkin Eclipse-pohjaiset ratkaisut kuten RTC mahdollistavat hyvät kustomointimahdollisuudet ja kolmansien osapuolien ratkaisut. Toisaalta Microsoft-maailman TFS on suljettu ympäristö, mutta siihen on olemassa rajapinnat, joiden avulla voidaan myös tarjota kolmansien osapuolien tuotteita. Edellä esitellyt ratkaisut ovat isojen ohjelmistotalojen tuotteita, joten voidaan olettaa, että niille on olemassa hyvä enterprise-tasoinen tuki.

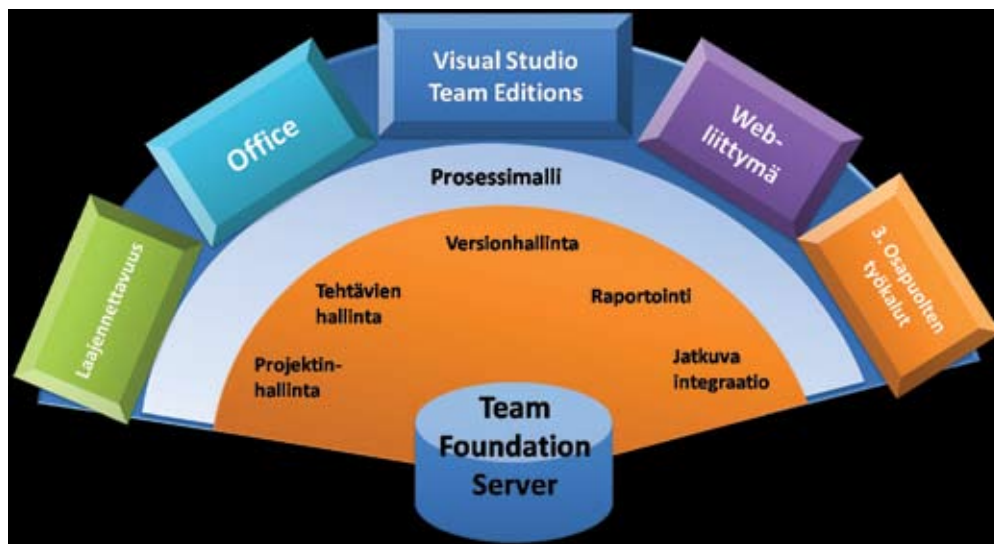
Yhteenveto

Onko aika kypsä tällaisten kollaboraatioalustojen käyttöönotolle? Mielestämme on, teknologiamielessä ainakin. Monipuolisten kommunikointitekniikoiden ja ryhmätyöominaisuuksien lisääminen ALM-tyyppisiin sovelluskehitysympäristöihin tekee ympäristöistä niin paljon mielekkäämpiä käyttää, että on ihme ettei näitä ominaisuuksia ole yhdistetty jo aikaisemmin.

Se että kaikki sovelluskehityksessä mukana olevat henkilöt näkevät koko ajan projektin tilan ja sen kehittymisen sekä se että kommunikointi on helppoa, tehostaa tiimin toimintaa ja parantaa laatua. Tällaisia ohjelmistoja voidaan pitää myös riskienhallinnan välineinä, koska virheet huomataan nopeasti ja niihin voidaan puuttua välittömästi.

Ketterien menetelmien käyttöönotto on lisännyt ymmärrystä siitä, minkälaisia kehitysympäristöjen tulisi olla ja mitä toimintoja niiden pitäisi tukea. Kollaboraatioalustat vastaavat hyvin tähän huutoon. Yhtä asiaa kannattaa kuitenkin korostaa: varsinainen toteutustyö ja testaus eivät muutu mitenkään, vaikka käytettäisiin näitä uusia ohjelmistoja.

Lopuksi voisi todeta että mikä tahansa ympäristö on juuri niin hyvä kuin sen käyttäjät ovat eli ympäristöt eivät itsessään ratkaise yhtään ongelmaa. Toisaalta huono ympäristö voi pilata paljon.



Kuva 2.
Team Foundation
Server

suoraan kehitysympäristöön, josta ne on helppo löytää ja kuitata tehdyiksi. Team Foundation Serverissä on lisäksi versionhallinta sekä tuki jatkuvalla integraatiolla ja automaattisille testeille. Palvelimessa on myös paljon avoimia web service -rajapintoja, joiden avulla myös omia tai muiden ratkaisuja on helppo integroida osaksi kokonaisuutta. Tästä huolimatta kokonaisuus toimii helpoiten Microsoft-alustaan panostavilla tiimeillä.

Team Systemin käyttöönotto ei tule halvaksi, sillä se vaatii oman lisenssin lisäksi palvelimen ja



CodeBakers - Uunituoretta IT-osaamista

Testaus

- kokonaistestauspalvelu
- asiantuntijapalvelut
- suorituskykytestaus
- testausvälinekonsultointi
- testauksen menetelmäkehitys
- testauksen automatisointi

CRM

- asiakkuuksien ja palveluiden hallintajärjestelmät
- asennus, käyttöönotto ja koulutus
- migraatiot ja integraatiot
- MS Dynamics CRM

Ohjelmistokehitys

- asiakaskohtainen ohjelmistokehitys
- arkkitehtuurikonsultointi
- teknologiakonsultointi
- asiakaskohtainen tuotekehitys
- vaatimusmäärittelyt
- projektipalvelut

Tule meille tuottamaan uunituoreita IT-ratkaisuja!

Tarjoamme asiakkaillemme palveluita niin konsultointina, projekteina kuin kokonaisratkaisuinakin.

Tapamme toimia on tutustumisen arvoinen, ota yhteyttä!

www.codebakers.fi

Testaus on innovatiivista

Teksti: Risto Paavola

Ensiaskelelta voisi olla vaikeaa kuvitella, että innovoinnilla ja testauksella olisi jotain yhteistä tai yleensäkin mitään tekemistä toistensa kanssa. Onhan testaus jonkin olemassa olevan koittamista, sen testaamista, kun taas innovointi jonkin uuden asian keksimistä. Onneksi kuitenkin innovatiivisuutta pääsee harjoittamaan myös testaamisen saralla, olisihan testaus muuten kovin tylsää ja itseään toistavaa työtä. Mahdollisuus innovatiivisuuteen piilee testausmenetelmissä, -tavoissa ja työkaluissa; kuinka testata jotain olemassa olevaa paremmin kuin ennen.

Miellän innovaation uuden asian keksimiseksi. Harvemmin innovaatio kuitenkaan syntyy tyhjästä, sanokoot omena ja vetovoima -tarinan kannattajat mitä tahansa. Innovaatio vaatii tietyn alkutietämyksen asiasta, mieluiten mahdollisimman laajalta alueelta asian ympäriltä. Tästä esitiedoista innovatiivinen, fiksu ihminen luo sitten jotain aivan uutta ja - toivottavasti - parempaa kuin entinen. Analogia tieteelliseen tutkimukseen? Kyllä – siitä myös toiminimi innovaatioyliopistolle!

Innovointi on joskus vanhan ajattelutavan hylkäämistä

Yleensä innovaatio vaatii, että asiaa tarkastelee juuri siltä kantilta, josta sitä ei yleensä tarkistella

tai jopa siltä kantilta, joka yleensä mielletään vääräksi. Tämän takia innovaatioon kykenevä ihminen on kiinnostunut erilaisista näkökulmista ja myös kunnioittaa niitä. Innovaatioon kuuluu vanhan ajattelutavan parantaminen tai joissain tapauksissa kokonaan sen hylkääminen. Kannattaa siis olla avoin eriäville mielipiteille ja eikä tule pelätä koittaa hulluimmaltakaan tuntuvaa ideaa.

Järkevä testaus on optimointiongelmaksi

Mitä innovatiivisuus sitten tarkoittaa ohjelmistotestauksessa? Kaiken mahdollisen testaaminen ohjelmistossa takaa voiton (lue: voidaan todeta, että ohjelma toimii kuten vaaditaan), mutta se ei ole järkevää. Resurssien määrä, joka kaiken testaamiseen vaaditaan, on miltei ääretön. Kyseessä on siis optimointitehtävä, kuinka testata ohjelmisto mahdollisimman täydellisesti, kuitenkin minimoiden resurssimäärät?

On olemassa menetelmiä ja tapoja, jotka auttavat testaajaa testaamaan mahdollisimman viisaasti – siis vähällä vaivalla ja mahdollisimman kattavasti. Tämänlaisia menetelmiä ovat muun muassa raja-arvoanalyysi tai vaikkapa automatisoitu regressiotestaus. Raja-arvoanalyysin oletusarvona on se, että yleensä virheet ohjelmistossa tuppavat kasaantua kohtiin, joissa muuttuja saavuttaa raja-arvon, missä sen käsittelyyn käy-



Risto Paavola
Senior Quality
Consultant,
Qentinel Oy

”Täysin uuden asian testaaminen vaatii innovaatiota, koska aiempia kokemuksia sen testaamisesta ei välttämättä ole tai ei ainakaan ole helposti saatavilla.”

tetty ehto muuttuu. Ei ole järkevää testata kaikkia arvoja, joita muuttuja voi saada, vaan ne, missä riski virheeseen on suurin. Automaattisen regressiotestauksen oletuksina ovat ne, että joskus uudet tehdyt virheet tuppaavat vaikuttamaan jo kertaalleen testattuun alueeseen ja aina uudelleen ja uudelleen samojen asioiden testaaminen ei ole järkevää. Kummassakin tavassa tai menetelmässä on siis käytetty esitietoja hyväkseen helpottamaan testaajan arkea, riskejä ja resurssien käyttöä minimoiden. Nekin siis ovat olleet innovaatioita, kun ne on ensimmäisen kerran keksitty!

Raja-arvoanalyysillä ja automaattisella regressiotestauksella on kuitenkin eräs fundamentaalinen ero toisiinsa nähden. Raja-arvoanalyysillä pyritään löytämään virheitä, kun automaattisella regressiotestauksella pyritään varmistamaan, että virheitä ei löydy. Molemmat kuitenkin tähtäävät virheettömyyden ohjelmistoon. Eri lähestymistavat samaan ongelma-alueeseen voidaan nähdä myös innovaatioina, kuten vaikkapa negatiivinen vs. positiivinen testaus ja toiminnallinen vs. ei-toiminnallinen testaus. Innovaatio testaustavassa tarkoittaa siis jonkin uuden ja järkevät käytännön koettamista tietyssä tilanteessa.

Innovointi ohjelmistotestauksessa

Innovatiivista tapaa testata on mahdotonta opettaa tai neuvoa, koska innovaatio lakkaa olemasta innovaatio, kun se on keksitty. Kuinka opettaa jotakin, jota ei ole vielä keksitty? Onneksi vinkkejä innovatiiviseen lähestymistapaan, tai innovoimiseen voi antaa.

Täysin uuden asian testaaminen vaatii innovaatiota, koska aiempia kokemuksia sen testaamisesta ei välttämättä ole tai ei ainakaan ole helposti saatavilla. Kuinka sitten aloittaa täysin uuden asian testaaminen, jos vanhat menetelmät eivät sovi tilanteeseen?

Esimerkeistä raja-arvoanalyysistä ja automaattisesta regressiotestauksesta voi löytää joi-takin yhteisiä asioita. Näitä ovat muun muassa: kannattaa testata sitä, minne virheet tuppaavat kasaantumaan ja ei ole järkevää testata kaikkea. Kysymys on siis pohjimmitaan riskeihin perustuvaasta testaamisesta ja ajattelutavasta, että kaikkea ei kannata testata. Samoja ajattelumalleja pystyy yleensä käyttämään tilanteessa kuin tilanteessa.

Ajattelemalla pääsee yleensä helpommalla

Jos testaajan käsissä on uusi tuote, jonka rajapinnat tai toteutus ovat sellaisia, mihin nykyinen työkalupakki ei riitä, kannattaa pysähtyä hetkeksi ajattelemaan. Millä kielellä tuote on tehty? Mitkä ovat kyseisen kielen heikkouksia? Miten käyttäjä käyttää tuotetta? Minkälaisissa tilanteissa käyttäjien käyttötapa ei välttämättä ole yksiselitteistä? Kuka on kehittänyt tuotetta? Minkälaisia virheitä tämä kehittäjä on tehnyt mahdollisesti aiemmissa

tuotteissa? Tämänlaiset kysymykset auttavat keskittämään testauksesi alueihin, missä mahdollisesti – tai keskiwertoa suuremmalla riskillä on virheitä. Jos pystyy vetämään johtopäätöksen ja yleistyksen, että yleensä tapauksessa X virheet kasaantuvat paikkaan Y johtuen syystä Z ja niitä kannattaa testata tyyliin T, niin voilà!, innovaatio on syntynyt.

Yhtä lailla uuden tuotteen testausta kannattaa puntaroida myös siltä kannalta, miten todetaan mahdollisimman kattavasti ja käytettävissä olevin resurssein, että tuote toimii sillä tavalla kuin halutaan (positiivinen testaus). Innovointi tässä tapauksessa piilee myös resurssien minimoimisessa ja kattavuuden maksimoimisessa. Innovointi testauksessa tarkoittaa ongelman tarkastelua eri puolilta ja eri näkökulmista.

Innovaatio voi olla menetelmän tai tavan lisäksi myös työkalu. Työkalut helpottavat usein testaajaa työssängä, mutta eivät – ainakaan vielä – pysty täysin korvaamaan häntä. Yksinkertaisia ja itseään toistavia testejä voi jopa täysin automatisoida, mutta kertaluontoisempia testejä ja dynaamisia ohjelmistoja ei ole järkevää ainakaan toistaiseksi automatisoida. Tänä päivänä vaikuttaa siltä, että innovointi keskittyy testauksessa lähinnä työkalurintamalle. Tämä ei ole huono asia, mutta kannattaa muistaa, että automaattiset testitkin pohjautuvat algoritmile, joka pohjautuu testausmetodille. Testausmetodien kehittäminen on siksi tärkeää myös automaattisen testauksen kannalta.

Muita innovoinnin alueita ohjelmistotestauksessa

Jos hyppää käytännön testauksesta askelta – paria ylemmälle tasolle, niin innovoinnin mahdollisuudet kasvavat. Eräs kovasti muokkautuva alue laaduntakauksessa on testauksen näkyvyys ja raportointi. Innovoinnit tällä alueella keskittyvät mittareihin, tiedonvälitysmenetelmiin ja johtopäätösten tekemiseen tiedon pohjalta. Tämä ei suinkaan ole ainoa innovoinnin alue, koska tietotekniikka on vielä suhteellisen nuori ja kehittyvä ala. Innovaatioita syntyy tietotekniikan taholla jos toisellakin.

Ajan käyttäminen innovointiin kannattaa

Innovointi uuden asian testauksessa tarkoittaa päähkimistä, mikä olisi mahdollisimman järkevä tapa testata uutta ohjelmistoa. Innovaatio taas on menetelmä tai työkalu, jolla vastaavanlaista ohjelmistoa voidaan tulevaisuudessa testata. Molemmat ovat tärkeitä asioita testaajan arjessa – puhumattakaan siitä, että ne ovat rahanarvoisia keksintöjä. Innovointi tai innovaatio luo lisäksi onnistumisen tunnetta, joka motivoi tekemään niitä lisää. Näiden asioiden takia kannattaa uhrata hieman omaa aikaansa ”linjatyöstä”, miettiäkseen kuinka asian saisi toteutettua paremmin!

Virheettömän ohjelmistotoimituksen haasteet



Matti Matikainen
Projektipäällikkö Diglassa,
Systeemyöhydistyksen
hallituksen jäsen.

Lehdissä on keskusteltu vuoden aikana ohjelmistoista, jotka eivät toimineet luotetavasti ja projekteista ja joiden käyttöönotto epäonnistui. Virheelliset laskut, hukkuvat laboratoriotulokset ja toimimattomat luotokortit ovat nousseet keskustelun aiheeksi. Ongelmiin ei ole löydettävissä yhtä ratkaisua, koska virheisiin johtaneet tekijät ovat moniulotteisia. Toimivan ohjelmiston lisäksi tulee varmistaa myös toimivat laitteistot. Esimerkiksi virheet varmennusyhteyksissä saattavat lamaannuttaa käyttöönotetun järjestelmän.

Ohjelmistojen tuottamista verrataan usein rakennusteollisuuteen. Vaikka talojen rakennuksessa tapahtuukin virheitä, materiaalien käyttäytyminen, lujuslaskelmat ja työmäärien arvioiminen onnistuvat suhteellisen tarkasti. Menetelmät ovat myös kehittyneet alan pitkän historian varrella. Abstraktien ohjelmistojen rakentaminen vaatii erityisen paljon ammattitaitoa tietojärjestelmän toimittavalta organisaatiolta. Ohjelmistojen monimutkaisuutta ei aina hahmoteta projektin tarjous- tai alkuvaiheessa. Osittain se johtuukin ohjelmisto-alan nuoresta iästä.

Mitä virheetön ohjelmisto maksaa? Ollaanko virheettömyydestä valmiit maksamaan?

Toimittajan kannalta yksi kriittinen tekijä on toimituksen kiinteä budjetti. Toinen ongelmakohta on epäselvä vaatimusmäärittely. Mikäli ei tiedetä tarkkaan, mitä ollaan tekemässä, mikä on asiakkaan tarve tai miten ohjelmistoa käytetään, tuloksin on helposti huono.

Suurten projektien onnistumisen edellytyksiä ovat huolellinen valmistelu ja toimiva ohjelmistokehitysmenettely. Kehitystyössä on huomioitava pitkin ohjelmistokehitystä tapahtuva testaus ja laadun hallinta. Testaus ei siis saa olla projektin tai hankkeen viimeinen toimenpide ennen ohjelmiston luovuttamista asiakkaalle.

Laadun hallinnasta tinkimisen seurauksena ohjelmiston käyttöönotto saattaa epäonnistua tai kriittisen ohjelmiston toiminta saattaa aiheuttaa liiketoiminnallisia tappioita. Yksi varteenotettava vaihtoehto onkin ottaa testaus toimittavan organisaation ulkopuolelta. Tällöin on kuitenkin var-

mistettava testauksen liiketoimintaosaaminen. Testaajan on ymmärrettävä se, mitä testataan.

Ongelmana on myös ohjelmistoprojektien laajuuden hallinta. DeFacto standardina pidetään, että suurissa ohjelmistoprojekteissa koodauksen osuus on 20 % ohjelmistoprojektista.

Vaatiiko ohjelmistoprojektin ostaminen ammattitaitoa?

Onnistunut projekti tarvitsee muutakin kuin toimivat laadunhallinta prosessit, testauksen ja koulutetun henkilöstön.

Ohjelmistoprojektien ostaminen vaatii myös ammattitaitoa. Suuren ohjelmiston ostaminen on monesti asiakkaalle ainutkertainen ja kertaluontoinen hankinta.

Tällaisessa tilanteessa, on luonnollista, että hankkivan organisaation kokemus ohjelmistoista tai järjestelmätoimituksista on vähäistä. Ostavalta osapuolelta ei voi odottaa pitkälle hiottuja hankintaprosesseja. Toimittaja ei myöskään voi olettaa, että asiakas olisi ohjelmistoalan ammattilainen. Prosessiosaamisen tulee lähtökohtaisesti tulla ohjelmiston toimittavalta organisaatiolta.

Ohjelmiston ostotilanteessa ja suunnitteluvaiheessa on hyvä ymmärtää ohjelmistoliiketoiminnan realiteetit. Suurissa tai vaativissa ohjelmistoprojekteissa hyvää ei saa halvalla. Mikäli laadusta ja projektin hallinnasta tingitään, riskit kasvavat. Yksi epäonnistumisen ja tyytymättömyyden kriteeri on tarjouskilpailutilanteessa liiallinen hinnan korostaminen. Kustannuspaineiden johdosta monet ohjelmistot siirretään tuotantoon liian heikoin perustein. Juuri laadunhallinta ja testaus ovat useimmiten tekijöitä, joista tingitään ensimmäisenä. Tämä puolestaan altistaa ohjelmistojen virheille ja lisää mahdollisia liiketoimintariskejä. Tarjouskilpailussa kannattaa kyseenalaistaa, onko halvin tarjous paras? Hankinnan hinnan lisäksi kannattaa myös arvioida, mitä epäonnistuminen maksaa?

Projektin ammattitaitoinen johtaminen korostuu monimutkaisissa toimitusprojekteissa. Suuri osa julkisuuden epäonnistuneista projekteista onkin kaatunut tarjousvaiheessa tai projektien johtamisessa tapahtuneisiin virheisiin. Menetelmistö toimii vain silloin, jos toimituksen johdossa on ammattilainen.

”Suurissa tai vaativissa ohjelmistoprojekteissa hyvää ei saa halvalla. Mikäli laadusta ja projektin hallinnasta tingitään, riskit kasvavat.”

Innomotivoivaa ja liukuhihnatyötä

Kaikentietävästä Wikipediasta löytyy, että innovaatio eli uudennus on jokin uutuus. Innovaatiohan sanana on jotain sellaista, josta me suomalaiset saamme olla ylpeitä. Innovaatio-sanana juuri on lähtöisin suomen kielen into-sanasta. Innostunut, innokas, innovaatio. Eli innovaatio on jokin uusi asia, joka on syntynyt keksijän ollessa suuressa innostuneisuuden flow-tilassa. Kerran joku väitti, että sana juontaa juurensa englannin kielestä, tai jopa jostain latinan novus-sanasta, mutta sehän on täyttä soopaa. Innostunut on englanniksi enthusiastic, joten innovaatio olisi niin muodoin entusiasiaatio. Ruotsiksi se taas on ivrig, joten innovaatio olisi ivrigaatio.

Innovaation ja massatuotannon vastakkainasettelu on runsaasti ajatuksia herättävä. Nimittäin, alalla kuin alalla, yritykset hyvin mielellään pyrkivät kehittämään jotain, jolla erottua kilpailijoista. Esimerkiksi Alko on erottautunut innovoimalla itselleen alkoholituotteiden vähittäismyyntimonopolin. Sitä ei voida enää toistamiseen keksiä, joten Alkolla on kyllä siinä mielessä hyvin pullat uunissa. Joku eduskunnassa töissä oleva ihminen voisi tietenkin innovoida, että murretaan tuo monopoli. Erottumista saa aikaiseksi myös jollain keksinnöllä, joita myös siis innovaatioiksi kutsutaan. Tarkoituksena saada siitä innovaatiosta massatuote, jotta saisivat myyntituottoina niin paljon rahaa, että omistajat voisivat kuluttaa päivänsä yrittäessään innovoida, mitä niillä rahoilla tekisivät. Niin simppeleä se on.

Tietotekniikassa innovaatioita vilisee. Eikä niitä varten tarvitse keksiä digitaalisessa viestinnässä käytettävää Viterbi-algoritmia tai tiedonsiirron nopeutuksessa käytettävää optista valokaapelivahvistinta. Näillä aikansa innovaatioilla - Viterbi-algoritmi kehitettiin 60-luvulla - kamppailtiin tämän vuoden Millennium-palkinnosta. Ihan tavallinen pulliainenkin voi innovoida. Ei siihen tarvita kuin kekseliäisyyttä. Otetaan esimerkiksi Alex Tew. Muutama vuosi sitten tuo parikymppinen brittipiskelija teki miljoonan pikselin alueen kotisivulleen, jakoi sen sadan pikselin alueisiin ja alkoi myydä niitä sadan dollarin hintaan. Siis dollari per pikseli. Eli miljoona dollaria! Ja kuinka ollakaan, idea lumivyörähti ja kaveri sai miljoonansa. Tuo kertoo jo varsin hyvin, että kellä tahansa on mahdollisuus mihin tahansa, varsinkin internetissä.

Usein meillä raskaan työn raatajilla, normaalin palkkatyön parissa puuhastelevilla pienillä ihmisillä ei riitä energiaa moisten kuningasajatusten tuottamiseen. Tai ainakin näin luulemme. Kaikkeenhan riittää aikaa ja energiaa, jos vain löytyy riittävästi motivaatiota, tuota luovuuden jumalten kunnioittamaa ahkeruuden airueiden nauttimaa nektaria, joka usein innovaatioiden syntyyyn niin saumattomasti liitetään. Minäkin olen käyttänyt lukuisia motivaatio-yksiköitä sen miettimiseen, miten saisin päätoimittajan maksamaan itselleni parempaa palkkaa. Ahneus, laiskuus ja mukavuudenhalu ovat niitä parhaita motivaattoreita vallan ja seksin lisäksi. Tai no, parhaathan löytyvät sieltä Maslowin portaikon alapäästä. Mutta tuohon palkka-asiaan, olen monasti sanonut päätoimittajalle, että olen juuri tekemässä suurta kirjallista innovaatiota, minulle pitäisi nyt ehdottomasti maksaa lisää palkkaa, vähentää työtehtäviä ja antaa ylimääräisiä lomaviikkoja, koska tunnetustihan luovuus vaatii kiireetöntä ja stressitöntä joutenoloa - ilman mitään sitoumuksia mihinkään suuntaan. Kirjoitinkin esimiehelleni sähköpostia, että Pulitzer-palkinnon saamiseksi tarvitsee ehdottomasti olla riippumaton. Harmikseni vain innostuneisuuden tilassani olin unohtanut kirjoittaa tuon "olla"-sanana siihen lauseeseen...

PS.

Valtakunnallinen Systeemyölehdessä kilpailukysymyskilpailukin julistetaan nyt ratkaistuksi. Molemmat vastaukset olivat erittäin hyviä, toinen niistä jopa oikein. Jutun jujunahan oli lähteä purkamaan arvoitusta takaperin ja kääntämällä laskutoimitukset vastakkaisiksi, kertolaskut jakolaskuiksi ja niin edelleen. Kompana siinä voidaan pitää sitä, että 3/4:lla kasvattaminen tarkoittaa kertomista 7/4:llä, sen käänteistoimitus on kertominen 4/7:llä ja 1/3:aan vähentäminen taas tarkoittaa jakamista kolmella, ei kertomista 2/3:lla. Vai miten se nyt oli? Vastaus siis muodostuu näin: 2 kerrotaan 10:llä, saadaan 20. Tästä vähennetään 8, saadaan 12. 12:n neliöjuuri on 144, johon lisätään 52, saadaan 196. Tämän neliöjuuri on 14, joka kerrotaan 3:lla, saadaan 42. 42 kerrotaan 7:llä, saadaan 294. 4/7 294:stä on 168, joka lopuksi jaetaan 3:lla, vastaus on siis 56. Tämä siis oli oikea vastaus. Muttamutta. Harvinaisen paljon toimitusta ja varsinkin lehtitalon atk-osastoa työllisti lähteessä oleva virhe! Lähteenä käytetty Matematiikan käsikirja nimittäin sekoitti kolmasosaan vähentämisen ja kolmasosalla vähentämisen. Lähteen mukaan oikea vastaus olisi ollut 28. Mitäpä tästä opimme? Ei kannata luottaa yhtään mihinkään ikinä. Kaikki ei ole totta, ei edes Internetissä. Mutta joka tapauksessa, palkintotehtävä ei ollut ollenkaan helpoimmasta päästä, mutta tokihan palkinnon upeuskin jo vaatii vaativaa ja eritoten innovoitunutta suoritusta. Nopeimmin oikea vastaus tuli Jaana Siukolalta. Onneksi olkoon!

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

Teemayhdistyksen jäseneksi voivat liittyä kaikki systeemytyöstä 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
Systeemytyöyhdistys Sytyke ry
Talvikkitie 40 A 33
01300 Vantaa
p. 09 56075363
f. 09 56075365
sytyke@hennax.fi

Hallitus 2008

Puheenjohtaja Mitro Kivinen

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

Mika Hynä

p. +358 50 368 4277
mika.hynna@ake.fi

Seppo Takanen

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

Ilkka Pirttimaa

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

Marianne Malila

marianne.malila@digia.com

Matti Matikainen

p. +358 40 578 0434
matti.matikainen@digia.com

Kari Uusi-Aijö

p. +358 40 835 6541
kari.uusi-aijo@pohjola.fi

Varajäsenet

Juha Jääskinen

juha.jaaskinen@digia.com

Tarja Raussi

p. +358 50 548 1823
tarja.raussi@tieturi.fi

Liittokokous- edustajat

Mitro Kivinen

mitro.kivinen@kolumbus.fi

Minna Oksanen

minna.oksanen@gmail.com

Seppo Takanen

seppo.takanen@codebakers.fi

SYTYKE ry:n hallituksen sähköpostilista:

hallitus@sytyke.ttlry.fi

Sytyttääkö? - Liity jäseneksi



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

Tietotekniikan liiton henkilöjäsenmaksu vuonna 2008 on alkaen 49€ ilman lehtiä, 1 lehti 61€. Eri-tyisryhmien hinnoittelusta lisätietoja Tietotekniikan liitosta. Lisäjäsenyys maksaa 12€/yhdistys.

Osaamisyhteisöt

Systeemytyöyhdistyksessä 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 laittamalla postia vetäjälle.

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

ProjektiOSY - ProSY pyrkii yhdistämään Systeemytyön projektitoiminnasta ja sen kehittämisestä kiinnostuneet, vetäjänä Markku Niemi.
make.niemi@kolumbus.fi

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 vetäjänä Lauri Pietarinen.
lauri.pietarinen@relational-consulting.com

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

KäytettävyysOSY 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

JÄRJESTYKSEN JA LUOVUUDEN VUOROPUHELU

Innovatiivisuus edellyttää vapautta ja luovuutta. Vuorovaikutus-, tiimi- ja leadership-aidot nousevat merkittäviksi kansalaistaidoiksi. Massatuotanto tarvitsee hiotut prosessit, tehokkaat toimintatavat ja arkkitehtuurin. Ohjelmistotuotannossa tarvitaan innovatiivisuuden ja järjestyksen tasapainoa. Tähän tasapainoiiluun tarvitaan ketteryyttä. Liiketoiminta kehittyy - kehity sinäkin!

Tue innovatiivisuutta leadership-osaamisella

Taitavan projektipäällikön leadership-aidot	10.-11.11. • 16.-17.12.
Muutosjohtamisen valmennusohjelma	3.11. alkaen
Change Leadership - muutoksen menestyksessä johtaminen	15.12.
Tuloksekas tiimidynamiikka	29.10.
Itseohjautuva organisaatio	3.11.
Monikulttuurinen viestintä	10.11.
Onnistunut vuorovaikutus ja toimintatyyli (DiSC)	5.12.
Johda itseäsi ja hallitse ajankäyttöäsi	20.11.
Työodotukset käytännön työvälineeksi	16.12.

Ryhdistä prosessit, projektit ja arkkitehtuuri

Prosessien mallintaminen	26.-28.11.
Liiketoimintaprosessien kehittäminen	19.-20.11.
Liiketoimintaprosessien mittaaminen	28.11.
Päivässä sinuiksi projektityön kanssa	15.10. • 15.12.
Projektin suunnittelu ja läpivienti	20.-22.10. • 10.-12.12.
Vaatimusten määrittely ja hallinta	22.-24.10. • 10.-12.12.
Yritysarkkitehtuurin suunnittelu	6.-7.11.
Tietojärjestelmäarkkitehdin valmennusohjelma	17.11. alkaen

Tasapainoile ketterästi ohjelmistotuotannossa

Laiha ohjelmistotuotanto (0,5 pv)	21.10.
Ketteryyden kulmakivet (0,5 pv)	21.10. • 3.12.
Ketterä ohjelmistotuotanto Scrumilla	22.10. • 4.12.
Ketterän ohjelmistoprojektin läpivienti	4.11.

Ole ajan hermolla - osallistu seminaariin

ICT Hype - kehityksen kärjessä	20.11.
Palvelujen tuotteistaminen ICT-alalla	26.-27.11.

Jotain ihan uutta www.tieturi.fi. Tule kokeilemaan!

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