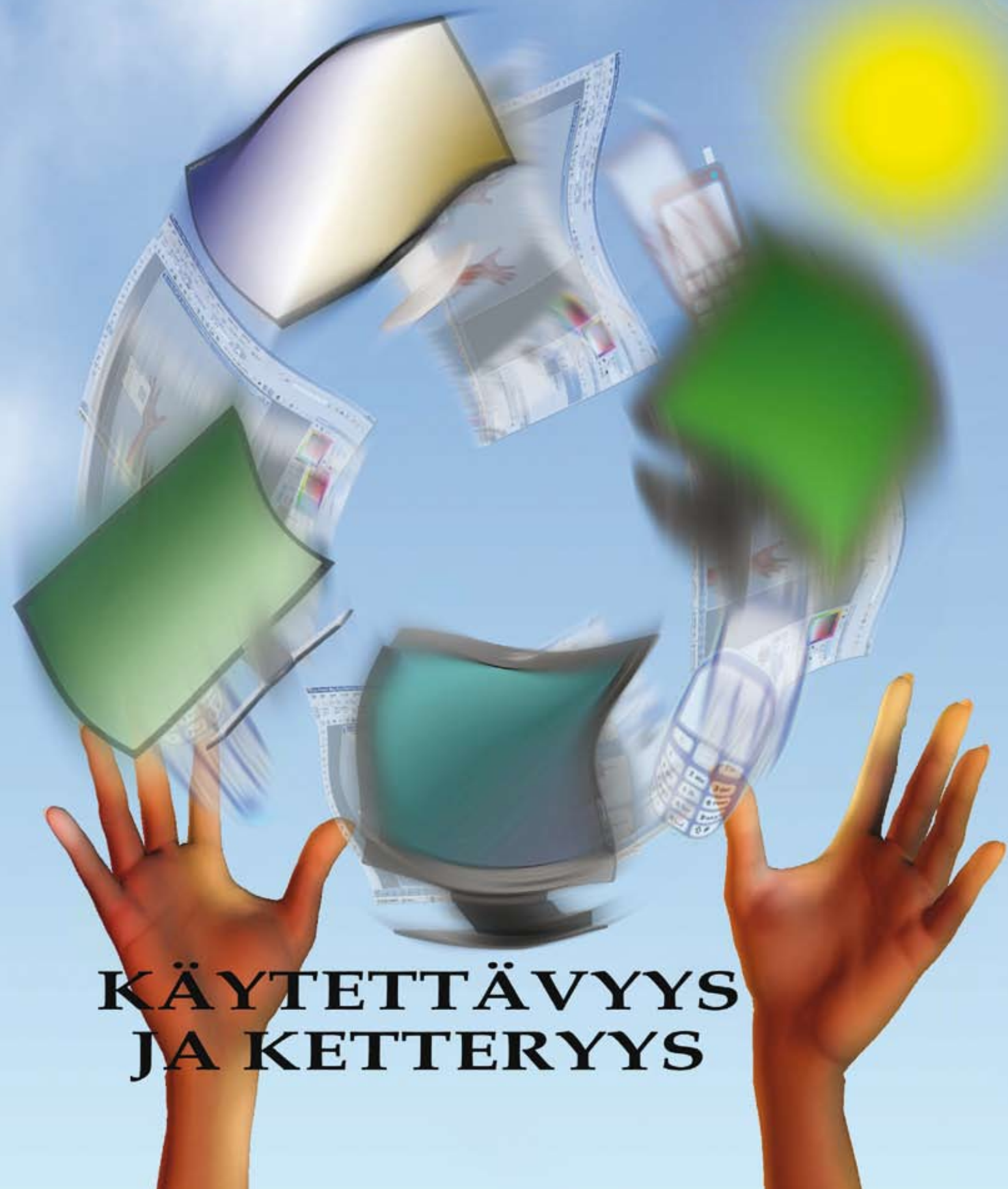


SYSTEMITYÖ

Systemityöyhdistys SYTYKE ry:n jäsenlehti N:o 4/2007



**KÄYTETTÄVYYS
JA KETTERYYS**

Oletko koskaan ajatellut mitä supersankarin työpäivään kuuluu?



Me olemme.

Supersankarit ovat fiktiivisiä hahmoja.
Siksi SYSOPENDIGIA tarjoaa monipuolisia ja
haastavia töitä todellisille ihmisille.

Etsimme nyt nuorempia sekä jo kokemusta hankkineita ohjelmistokehittäjiä ja
asiantuntijoita. Meillä on auki tehtäviä laajasti asiakasratkaisujen parissa ja
tuotekehityksessä. Meille on tärkeää, että viihdyt työssäsi. Työskentelet meillä alan
uusimpien teknologioiden ja ratkaisujen parissa yli tuhannen muun ammattilaisen kanssa.
Osaamisen jatkuva kehittäminen ja tiedon jakaminen sekä yhteistyö ovat meille tärkeitä
asioita. Tarjoamme monia etuja sekä palveluita työn ja perhe-elämän yhteensovittamiseksi.

www.sysopendigia.com/supersankari

SYSOPENDIGIA



Julkaisija

Systeemyöyhdistys 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 4/2007

Helena Venäläinen
Antti Martikainen
Matti Vuori
Eija Kaasinen

Lisätietoja lehdestä

www.sytyke.org/lehti

Tilaukset

Systeemyölehti 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

Matti Vuori

Seuraava numero

1/2008 Systeemyöläinenkin on ihminen
Toimituskunta:
Mitro Kivinen
Ilkka Pirttimaa
Eija Kalliala
Eija Hamina-Mäki

Ilmestyy: ma 10.3.2008

Painopaikka

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

Painos: 2500 kpl
ISSN 1237-0525
14 vuosikerta, nro. 4

Ilmoitushinnat

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

Arvonlisävero 0%
Vakiopaikan vähintään vuodeksi
varanneille 20% alennus.

Pääkirjoitus: Maailma muuttuu – ja meidän on muututtava sen mukana

Tässä lehdessä käsitellään kahta keskeistä systeemyön teemaa: käytettävyyttä ja ketteryyttä. Molemmat ovat merkki muuttuvasta maailmasta ja siitä, miten me suhtaudumme järjestelmiin ja niiden kehittämiseen ja samalla myös muutoksemme hitaudesta.

Käytettävyys on aina ollut systeemyötyössä tärkeää, vaikka se on käsitteenä suhteellisen nuori – totta kai järjestelmien pitää toimia erinomaisesti käytännön työprosesseissa ja käyttäjien arjessa. Kuitenkin, vasta jokin aika sitten on Systeemyöyhdistyksessäkin teema saanut niin paljon merkitystä, että aiheelle on perustettu oma osamisyhteisö ja otettu se Systeemyölehdessä keskeiseksi teemaksi.

Ketteryys on ilmiönä vielä uudempi. Koetaan, että maailma muuttuu niin nopeasti, että systeemyötä ja kaikkia organisaation toimintaprosesseja on toteutettava eri tavoilla kuin aiemmin – muutosherkästi, lyhyemmillä sykleillä. Tässä on kyse paradigmatason muutoksesta. Käytettävyyden ja ketteryyden

yhdistelmä on uusi haasteiden maailma, koska siinä yhdistyvät kaksi kehittyvää ajattelu- ja toimintamallien maailmaa, joiden molempien suhteen organisaatioiden käytännön valmiudet ovat usein puutteelliset.

Kaikkiin tämän tason muutoksiin liittyy suuria riskejä ja on vaarana, että lapsi menetetään prosessimuutosten pesuveden mukana, ellei uusia ajattelumalleja ymmärretä läpikotaisin ja osata soveltaa asianmukaisesti. Tämä lehti pyrkii herättämään tähän ymmärrykseen auttavia kysymyksiä, antamaan kenties jotain vastauksiakin, ja kertomaan siitä, millä tavalla käytettävyydenkin suunnittelu ja varmistaminen sopeutuu uuteen professionaaliseen ketterään toimintaan.

Toivomme kädessäsi lehden auttavan lukijoita uusien toimintamalliensa onnistuneessa, muutoksen riskit hallitsevassa ja uusia mahdollisuuksia löytävässä suunnittelussa ja soveltamisessa.

Hedelmällisiä lukuhetkiä!

Sisällys

- | | | | |
|----|---|----|--|
| 3 | Pääkirjoitus | 16 | Kokemuksia GUIDE:n ja Scrum:n yhdistämisestä
<i>Karri-Pekka Laakso</i> |
| 4 | Ketterät tuulet ohjelmistotuotannossa
<i>Pekka Abrahamsson
Outi Salo</i> | 20 | Sovelluksia verkkoon
<i>Irmeli Sinkkonen
Janne Tompuri</i> |
| 6 | Ketterän toiminnan filosofiaa ja periaatteita
<i>Matti Vuori</i> | 25 | Käytettävyyden ajankohtaiset haasteet
<i>Terhi Holappa</i> |
| 10 | Miten lunastaa ketteryyden arvolupaukset
<i>Mika Koski</i> | 28 | Protoiluvälineillä potkua vaatimusmäärittelyyn
<i>Antti Martikainen</i> |
| 13 | Kallialan kolumni
<i>Eija Kalliala</i> | 30 | WUD-seminaari
<i>Timo Jokela
Anneli Pekkarinen</i> |
| 14 | Käyttäjakeskeinen suunnittelu ja ketteryys
<i>Tomi Juhola</i> | 31 | Mikä mahtaa olla IN?
<i>Tarja Raussi</i> |
| | | 34 | Kuutamolla |



Pekka Abrahamsson, tutkimusprofessori, FLEXI-ITEA2-tutkimushankkeen konsortionvetäjä

Ketterät tuulet ohjelmistotuotannossa

Ketterät ohjelmistonkehityksen menetelmät rantautuivat Suomeen toden teolla vuoden 2002 aikana. Samana vuonna käynnistettiin myös VTT:ssä strateginen tutkimus ketterien menetelmien soveltamisen mahdollisuuksista ohjelmistoalan yrityksissä. Ketterät tuulet ovat puhaltaneet rajusti kuluneen viiden vuoden aikana.

Piloteista lupaavia tuloksia Euroopan laajuisesti

Erilaisten kokeilujen, pilottiprojektien ja tutkimustulosten kautta raportoidut positiiviset kokemukset ovat viime vuosina rohkaisseet entistä useampia yrityksiä muuttamaan toimintatapojaan ketterämpään suuntaan. Nykytulosten valossa ketteryys ei enää rajoitu pieniin ohjelmistotiimeihin ja ei-kriittisten sovellusten toteuttamiseen, vaan myös suuret, moniteknologiset, hajautettua tuotekehitystä ja alihankintaa soveltavat yritykset pyrkivät löytämään ratkaisuja ketterien menetelmien käyttöönottoon ja soveltamiseen. Puhtia tälle näkemykselle loi VTT:n vetämä Eurooppalainen 22 m€:n AGILE-ITEA tutkimusohjelma, jossa 68 pilottiprojektin ja yli 2000 ohjelmistosuunnittelijan ja insinöörin voimin ketteryyden sovellutuksia eri sovelluskentissä tutkittiin. 68 hankkeesta peräti 73 %:a päättyi hyvään tai erinomaiseen tulokseen. Kärkihankkeet suomalaisissa yrityksissä johtivat peräti 70 %:n aikataulu- ja kustannussäästöihin. Nämä tulokset lyötiin pöytään jo vuoden 2006 lopussa. AGILE palkittiin kansainvälisesti arvostetulla ITEA Achievement Awardilla 18.10.2007 Berliinissä.

Vaikutuksia koko organisaatioon

Organisaation koosta ja tavoiteltujen muutosten laajuudesta riippuen ketterien menetelmien omaksuminen vie 2-3 vuotta. Muutokset ovat kuitenkin välittömiä sovelluskehityskäytäntöjen tasolla tapahtuvien konkreettisten muutosten ansiosta. Välitön vaikutus konkreettiseen työhön onkin yksi

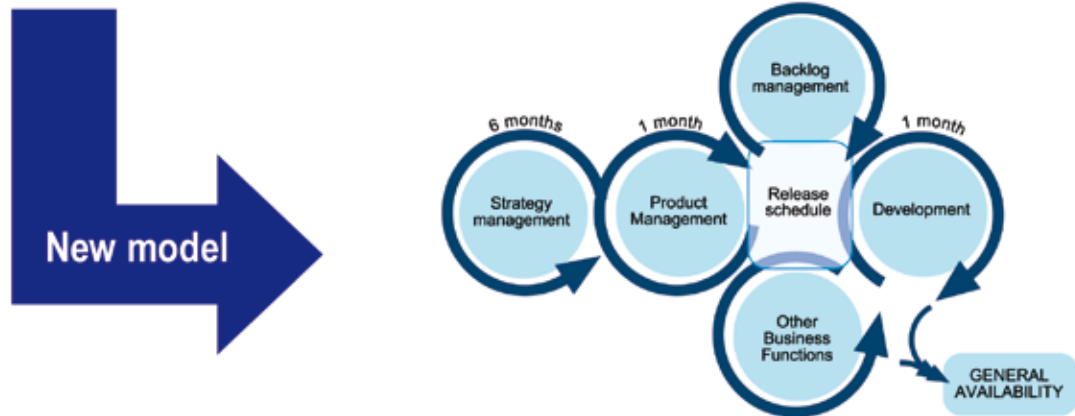
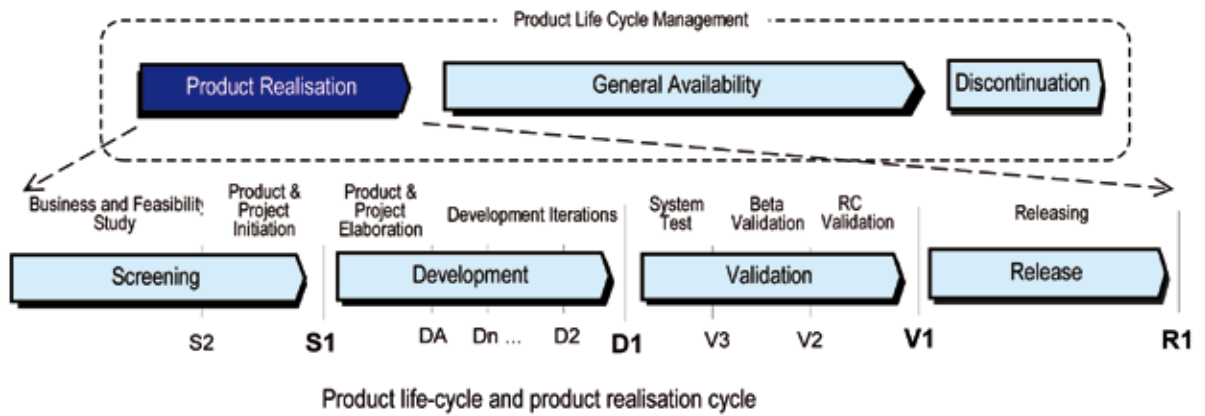
sy, miksi ketterät ohjelmistonkehityksen menetelmät ovat saaneet paljon huomiota kehittäjäyhteisössä. Tunnetuimpia ketteriä menetelmiä ovat Scrum ketterään projektinhallintaan ja Extreme Programming (XP) ohjelmistoteknisiin käytäntöihin.

Käyttöönottaessaan ketteriä menetelmiä monet yritykset ovat kuitenkin havainneet, että ohjelmistotuotantomenetelmien ja ohjelmistotiimien sisäisen toiminnan ketteröittäminen ei yksistään riitä varmistamaan kaikkia ketteryyteen liitettyjä hyötyjä. Paitsi sovelluskehittäjien, myös organisaation muiden toimintojen ja ohjelmistotuotannon sidosryhmien on muokattava toimintaansa tehostamaan ja tukemaan ketterien ohjelmistotiimien työtä ja poistamaan esteitä huippusuoritusta tavoiteltaessa. Yhtenä esimerkkinä voidaan mainita muuttuvien vaatimusten hallinta paitsi tuotekehityksen aikana myös niiden vaikutukset tuoteversioiden pidemmän aikajänteen suunnitteluun, kehitystyön aikataulutukseen ja kuluihin sekä hajautetun tuotekehityksen hallintaan. Ketterimmänkin ohjelmistotiimin toimintaa väistämättä rajoittaa ja hidastaa perinteisemmässä rytmisessä toimiva ja eri toiminta-ajatusta noudattava tukiorganisaatio.

Ketteryyden laajentaminen läpi organisaation toiminnan vaatii kokonaisvaltaista ajatustavan muutosta ja entisten toimintatapojen uudelleen harkintaa kaikilla yrityksen tasoilla. Ei myöskään ole lainkaan itsestään selvää, mikä ketteryyden aste tai mitkä ketterät menetelmät kuhunkin yritykseen soveltuvat. Yksi haaste on myös se, miten valitut ketterät menetelmät ja toimintamallit sovitetaan perinteisempien menetelmien kanssa yhteen toimivaksi, entistä ketterämmäksi toimintakokonaisuudeksi. Ketteryydelle ei ole kehitetty vielä standardia tai referenssimallia. Tästä syystä ketteryyttä ei sinänsä voi nähdä itseisarvona toimintojen kehittämisessä vaan etsiä kuhunkin yri-



Outi Salo, erikoistutkija, FLEXI-VTT-tutkimusprojektin projektipäällikkö



tykseen sopiva toimintamalli ja ketteryyden aste. Tuloksena on usein ns. hybridimalli, jossa parhaita vahvuuksia on yhdistelty molemmista lähestymistavoista.

Käyttöönotto vaatii suunnitelmallisuutta

Ennen kuin yritys lähtee ketteröittämään toimintatapojaan, tulisi tarkasti harkita mitkä ovat ne ensisijaiset tavoitteet ja lisäedut joihin ketterien menetelmien käyttöönotolla pyritään. Tutkimuksessa on jo luotu keinoja tämän tavoitehaarukoinnin ja nykytila-analyysin tekemiseen. On havaittu, että selkeillä tavoitteilla ja huolellisesti suunnitellulla ja toteutetulla käyttöönotolla onkin keskeinen rooli toimintatapojen muuttamisessa. Myös eri osapuolten – niin kehittäjien kuin johtoportaan-kin – sitoutuminen ja osallistuminen hankkeeseen vaikuttaa suuresti sen onnistumiseen. Todennäköisesti yleisin ja riskittömin tapa lähteä kokeilemaan

helposti keskittymään vain sovelluskehitystason toimintoihin. Pilottihankkeiden pitäisikin paremmin kattaa uuden toimintamallin muutokset läpi organisaation, jotta saadaan kokonaisvaltainen kuva ketteröitymisen vaikutuksista ja muutostarpeista.

Käyttöönottoa helpottamaan kannattaa tutustua AGILE projektin luoman palvelun (www.houseofagile.org) antimisiin. Palvelusta löytyy tietoa piloteista ja koulutusaineistoa viidellä Euroopan kielellä.

Tulevaisuudessa tuuli vain yltyy: Ideasta tuotteeksi kuudessa kuukaudessa!

Ketterät tuulet puhaltavat jatkossakin kun eurooppalainen tutkimus aiheen tiimoilta lyö lisää löylyä kiukaalle. FLEXI-ITEA2 (Flexible Global Product Development and Integration) on kolmi-

Mermit on nyt Ixonos Teknologiakonsultointi Oy

Ixonos Oyj:n tytäryhtiön, Mermit Business Applications Oy:n nimi muuttuu Ixonos Teknologiakonsultointi Oy:ksi. Sen palvelut kattavat järjestelmä- ja teknologiakonsultoinnin, arkkitehtuurikonsultoinnin sekä järjestelmäsuunnittelun ja –määrittelyn.

Ixonos Oyj on tieto- ja viestintäteknologia-alan palveluyritys, joka tuottaa asiakkailleen joustavia, näiden kilpailukykyä ja riskienhallintaa tukevia ohjelmistokehityksen ja ohjelmistoylläpidon sekä projektijohtamisen ratkaisuja.

Ixonos Teknologiakonsultointi Oy Lars Sonckin kaari 10, 02600 Espoo www.ixonos.com



ixonos
intelligent software. passionate people.

vuotinen VTT:n vetämä tutkimusprojekti (2007 - 2009), jonka konsortioon kuuluu kaiken kaikkiaan 38 tutkimus- ja teollisuuspartneria kahdeksasta Euroopan maasta (www.flexi-itea2.org). Projektin volyymi on merkittävät 40 m€. Projektissa tutkitaan ja sovelletaan ketteriä menetelmiä moniteknologisissa ohjelmistokehitysprojekteissa. Mottona projektille on "Ideasta tuotteeksi kuudessa kuukaudessa!". Tavoitteena on kehittää uusia ratkaisuja ohjelmistotuotantoteollisuuden keskeisiin haasteisiin, kuten globaalin ja hajautetun tuotekehityksen ja yrityksen koko liiketoimintamallin sekä verkottuneen tuotekehityksen ja tuotantoketjujen ketteröittämiseen. Suomesta ketteryyden edelläkävijää ovat tekemässä VTT:n lisäksi, Oulun yliopisto, Tampereen teknillisen yliopiston Porin yksikkö ja Reaktor Innovations.

Ketteryyden ytimeen pyrkivät tutkimuksen vahvistamana F-Secure, Elektrobit, Hantro, Nokia ja Nokia Siemens Networks. Suomen panostus lähentelee 100 henkilötyövuotta. Hanketta rahoittaa Suomen osalta Tekes.

Suomessa ketteriä menetelmiä ja niihin liittyviä ratkaisuja levittää FLEXI-ITEA2 projektiin liittyvät organisaatiot esimerkiksi Agile-Finland -yhteisön kautta. Koulutuksia ja tilaisuuksia järjestetään useita kertoja vuodessa. Suosittelemme yrityksiä osallistumaan alan merkkikonferenssiin, eli XP2008:aan (www.xp2008.org), joka järjestetään kesäkuussa 2008 Irlannin Limerickissä.



Ketterän toiminnan filosofiaa ja periaatteita



Ohjelmistoalalla tunnutaan nykyisin joka yhteydessä puhuttavan ketteryydestä. "Et ole ketterä" on lähes karkein kuviteltavissa oleva loukkaus. Ketteryydestä on tullut kaiken läpäisevä paradigma, jota sovelletaan mm. liiketoiminnassa, tuotekehityksessä, ohjelmistokehityksessä, testauksessa.

Millaista on ketterä toiminta?

Ketterää toimintaa on erilaisia tyyppejä. Tavoitetehtävinen ketteruus perustuu tilannekatsaukseen, jonka perusteella tunnistetaan uudet keinot, joilla päästään parhaiten tavoitteeseen. Tai tilannekatsauksen perusteella voidaan tunnistaa uusi tavoite, johon pyritään. Kokeileva ketteruus pyrkii tunnistamaan mahdollisuuksia ja niiden perusteella tekemään päätöksiä jatkosta.

Ketteruus on yleensä merkittävimmillään erilaisissa projekteissa tai toimintaprosesseissa. Niiden tyyppejä voivat olla esimerkiksi:

- Ketterä prosessi. Nopea, syklinen.
- Ketterä prosessin räätälöinti. Projektia käynnistettäessä räätälöidään nopeasti projektimallia valitsemalla siihen sopivat menettelytavat.
- Ketterästi tehtävät työt. Systemaattisen, ennakkosuunnitellun prosessimallin vaiheissa tehdään työtä ketterällä tavalla.
- Tutkivat prosessit ("exploration"-merkityksessä, ei "research"). Tehdään havaintoja ja edetään sen mukaan.

Näillä kaikilla on kuitenkin aina etukäteen päätetty missio.

Maailmalla on valmiita ketteriä malleja erilaisiin projekteihin ja prosesseihin. Niillä on sama ongelma kuin millä tahansa yksittäisten asiantuntijoiden tekemillä prosessimalleilla. Ne eivät ota huomioon kaikkia asioita – koska malleja kehittämässä ei ole ollut riittävän laajaa asiantuntemusta tai kehittämisen aikainen konteksti ei edellyttänyt prosessilta enempää. Mallit ovatkin luonteeltaan ideaalimallin sijaan filosofisia kärjistyksiä. Siksi niitä on sovellettaessa aina täydennettävä. On esimerkiksi varmistettava, että ketterässä ohjel-

Ammatinomainen ketteruus on oikeasti yksi vastaus muuttuneeseen maailmaan ja myös perinteisiin haasteisiin. Kuitenkaan sen ole-musta ei ole kylliksi jäsennetty eikä ole ilmi-ötasolla keskusteltu siitä, miten se suhtautuu yrityksen toiminnan kokonaisuuteen.

mistokehityksessä on mukana riittävä testaus ja riittävät käyttäjätutkimukset.

Ketterää toimintaa karakterisoivat mm. seuraavat piirteet; aikaan liittyvät: lyhyen tähtäimen tavoitteet, tekemisen rytmi, päätöksenteon nopeus, syklisesti toistuvat prosessit, oikean hetken tunnistaminen.

Kontekstiin liittyvät: todellisuus voittaa teorian, eläminen hetkessä, eläminen kontekstissa, menetelmien valinta tilanteen mukaan.

Ihmisiin liittyvät: usko yksilöihin, yksilöllinen osaaminen, vuorovaikutus, yksilöiden voimauttaminen, toimintatilan antaminen, dynaamiset roolit.

Tekemisen kohteeseen liittyvät: kohteen yhteinen omistajuus, asiakaslähtöisyys.

Toimintatyyliin liittyvät: tavoitetehtävyys, luovuus, suunnittelu asioiden edetessä, inkrementaalisuus, riskinotto, kokeileminen.

Ketterä toiminta on nopeaa ja luovaa. Ollaan psykologisesti valmiita muutokseen ja sitä jopa toivotaan. Toimintaa ohjataan tarvepohjaisesti luottamatta pitkiin ennakkosuunnitelmiin. Lähtökohtana on se, että maailmaa ei voida ennustaa kovin

Kirjoittaja toimii konsultointipalvelujen tuotepäällikkönä Plenware Oy:ssä ja konsultoi ja kouluttaa yrityksiä käytettävyyden varmistamisen lisäksi mm. laatuun, testaukseen ja riskienhallintaan liittyvissä asioissa. Hän on tehnyt erilaista toiminnan kehittämistä 1990-luvun alkupuolta lähtien. Vuori on mukana Käytettävyyden OSY:n ja Testaus-OSY:n hallituksissa.



pitkälle ja muutos on aina väistämätön. Tämä on arkista realismia.

Ketteryyteen assosioituu ajatus toimijoiden vapaudesta. Tämä on osin totta, osin myytti. Esimerkiksi ketterä ohjelmistokehitys on erittäin tiukasti prosessoitu. Se on ketterää päätöksenteon tasolla ja suunnittelun ja toteutuksen tasolla, mutta tiukasti sovellettava prosessi pitää kokonaisuuden kurissa. Keskeinen vapaus on kuitenkin se, että ei olla hirttäydytty vanhentuneisiin päätöksiin ja käsityksiin. Exploratiiviset prosessit ovat myös luonteeltaan itseohjautuvia, esim. tutkiva testaus.

Usein ajatellaan ketterän toiminnan olevan myös yhteisöllistä. Monet ketterän toiminnan peruselementeistä ovatkin sellaisia, joita tavataan yhteisöllisen toiminnan piirissä (yhteisöllinen media, verkkoyhteisöt, yhteisöllinen kehittäminen, Web 2.0). Yhteistä ovat mm. tilanneherkkyys, muutoksen nopeus, panostus psykologisiin tekijöihin, luottamus ihmisiin ja toisten voimauttaminen – kun ihmisille annetaan mahdollisuudet toimia, he tekevät hyviä asioita.

Tuntuu, että ketteryyteen liittyvät monenlaiset ideaalit. Onkin niin, että aina, kun kehitetään jokin uusi paradigma, siihen on tapana liittää kaikki samalla hetkellä oleelliset ihanteet, vaikka niillä ei olisi mitään tekemistä kyseisen paradigman kanssa. Esimerkiksi ketterää ohjelmistokehitystä tehdään asiakaslähtöisesti, mutta ihan yhtä hyvin voidaan ketterästi tehdä liiketoiminta- tai teknologialähtöistä ohjelmistokehitystä.

Ketteryys ei sinänsä ole suunnitelmallisen vastakohta. Ketterässä toiminnassa vain tehdään suunnitelmia tarpeen mukaan. Suunnitelmien määrä voi olla hyvinkin suuri. Olennaista on se, että suunnitelmia ei tehdä kovin aikaisessa vaiheessa.

Tarzan on klassinen esimerkki ketterästä toimijasta. Hänellä on kirkas kokonaistavoite tiedossa – missiona on prinsessan pelastaminen.

Hän hallitsee koko ajan tilanteen

- Tuntee ympäristön.
- Erittäin vahva osaaminen.
- Ymmärtää omat kykynsä, mihin pystyy.

Tarzan liikkuu vaihteittain, steppi kerrallaan.

- Kuitenkin riittävän pitkät liikkeet – ei saa pudota jokeen.
- Seuraava tarkistuspiste tiedossa. Aina uusi tilanearvio.
- Liikkuminen dynaaminen suunnittelu.
- Kaikista tulevista tapahtumista sisäinen malli.
- Kyky vaihdella rytmiä suunnittelun ja rajun toiminnan välillä.

Ennen seuraavaa liikettä tehdään aina sen ”riskianalyysi”. Tarzanilla on riskit tiedossa ja kyvykkyytensä ansiosta taito hallita ne. Hän tietää kaikki vaihtoehdot ja pystyy valmiutensa vuoksi nopeasti vaihtamaan suunnitelmaa.

Hänellä on apuna hyvät perusvälineet, jotka sopivat tilanteeseen kuin tilanteeseen – puukko, keihäs jne. Tarzan ei kuvittele pärjäävänsä yksin. Siksi hänellä on taustalla osaava tiimi eri asioiden eksperttejä – apina päästämään häkistä ja välittämään viestejä; raskaampiin operaatioihin norsu.

Ja keskeinen etu Tarzanilla on kuitenkin pitkä kokemus tästä työstä. Hänkään ei ollut seppä syntyessään.



Ketterän toiminnan suhde todellisuuteen

- Jos todellisuudessa ja teoriassa on ero, todellisuus voittaa.
- Jos suunnitelmissa ja todellisuudessa on ero, todellisuus voittaa.
- Maailmaa ei voi ennustaa pitkälle, joten sitä ei kannata yrittääkään.
- Mitään asiaa, ei sen vaatimuksia eikä optimaalista toteutusta, ei ymmärretä ennakoita riittävästi.
- Muutos on väistämätön.
- Kun vanhaan toimintaan tuodaan uusia elementtejä, toiminta muuttuu ennustamattomaksi ja sitä voidaan hallita vain ketteryydellä.
- On oltava koko ajan valmiina mihin tahansa muutoksiin.
- On oltava koko ajan valmiina tekemään mitä tahansa miten tahansa.

Ennakkosuunnittelu parantaa ketteryyttä

Ketteryyttä parantaa vahvasti, jos on valmiita toimintamalleja erilaisiin tilanteisiin. Näiden toimintamallien laadinta ei ole ketterää, vaan luonteeltaan hyvin systemaattista. Tehdään "entäpä jos..." -analyysyjä jokaiseen tilanteeseen. Kun niitä on valmiina, voidaan tilanneherkästi tunnistaa olosuhteet, valita nopeasti sopiva toimintamalli ja toteuttaa se tehokkaasti, koska se on pitkälti valmiiksi suunniteltu.

Esimerkki tästä on Ferrarin F1-tiimi. Tiimin on arveltu kykenevän salamannopeasti, Ross Brawnin nerokkaiden aivojen avulla, luomaan hetkessä uuden taktiikan kilpailun tilanteiden muuttuessa. Brown on kuitenkin kertonut, että taktiikkamuutokset ovat pitkälti perustuneet ennakkosuunnitteluun, eli analysointiin ja simulointiin ennakoita: Mitä tehdään, jos Häkkinen karkaakin lähdössä? Mitä tehdään, jos tulee radalle tulee kolarin vuoksi turva-auto? Entä sateen sattuessa? Mikä on eri varikkopysähdysmäärien vaikutus ja mitä vaikuttaa aikaan, jos pysähdysmääräsuunnitelmaa vaihdetaan tilanteen muuttuessa?

Sama periaate pätee yleiseen tiedonhankintaan. Esimerkiksi tuotekehityksessä on aina tärkeää mallintaa systemaattisesti käyttäjien maailmaa – sen jälkeen on hyvät mahdollisuudet tehdä kehitystä ketterästi. Kotiläksyt on siis aina tehtävä!

Mikä erottaa ketteryyden perinteisestä kaaoksesta

Ketterän toiminnan piirteet ovat usein lähellä "entisajan" huonosti organisoidun toiminnan piirteitä. Erona on kuitenkin monia tekijöitä, joilla

toiminta pidetään kasassa. Prosessit on mietitty ja niissä on sisäänrakennettuna laadunvarmistusmenettelyjä.

Ongelmana ovatkin heikosti toteutetut ketterät toimintamallit – jotka voivat toimia huomoin kuin heikosti toteutetut perinteiset mallit, koska niistä ei ole kokemusta.

Tasapainoinen ketteruus organisaatioissa

Organisaatioilla on aina tiettyjä peruslinjauksia. Ketteryydenkin on sellainen. Koska se edellyttää monia asioita, päätös ketteryyden tukemisesta on strateginen. Päätöksen teon jälkeen on organisaatiota johdettava eri tavalla kuin aiemmin ja panostettava erilaisiin asioihin kuin aiemmin. On muun muassa huolehdittava siitä, että erilaiset edellytykset ketterälle toiminnalle ovat olemassa.

Pelkkä ketteruus ei riitä organisaation toiminnan piirteeksi. Se tarvitsee tuekseen monenlaisia muita asioita mm. toiminnan isot linjaukset (politiikat ja roadmapit), osaamisen pitkäjänteisen kehittämisen, vahvat perusprosessit, menettelyt toimintaympäristön muutosten tunnistamiseen, luottamuksen, riskinoton ja riskienhallinnan tasapaino, kommunikatiivinen kulttuuri, kevyt byrokraatia ja vahva laatu-kulttuuria.

Ketteruus edellyttää monenlaisia asioita sekä yksilöiltä että organisaatioilta. Henkilöitä edellytetään monipuolista osaamista – oma "työkälypakki", jossa erilaisia toimintatyyplejä, menettelytapoja, menetelmiä, tekniikoita eli valmiita ajattel- ja toimintamalleja. Toimintatyyliä edellytetään nopeutta ja tilanneherkkyyttä.

Ketterän toiminnan ja ennakkosuunnittelevan toiminnan suhdetta demonstroi se, että ketteryyden ei voida siirtyä ketterästi. Esimerkiksi ketterän prosessin käyttöönottoprosessi on projekti, jossa tehdään mm. prosessin valintaa, suunnittelua ja räätälöintiä. Kaikkien osapuolten valmennusta. Pilotointia. Toiminnan vakiinnuttamista ja suunnitelmallista laajentamista.

Nämä kaikki ovat hyvin systemaattisia ja epä-ketteriä tehtäviä.

Ketterään toimintaan ei voida siirtyä, jos olemassa oleva toiminta on epäkypsää. Toiminta on ensin kasvatettava tietyille kypsyystasolle, jossa kaikki tarvittavat elementit ovat olemassa. Vasta silloin edellytykset laadukkaaseen ketterään toimintaan ovat olemassa.

Miten lunastaa ketteryyden arvolupaukset?

Mika Koski on Trinil Oy:n Senior Consultant, jolla on 13 vuoden kokemus ohjelmistokehityksestä eri rooleissa. Nykyisin hän toimii konsulttina ketterään kehityksen projekteissa. Trinil Oy on asiantuntijayritys, joka on keskittynyt projektinhallinnan ja soveluskehityksen asiantuntija-palveluihin.

Moni ketterä ohjelmistoprojekti on kohdannut loppunsa ennenaikaisesti, vaikka luvassa piti olla hyvää, halvalla ja nopeasti. Käyttäjakeskeiset suunnittelumenetelmät lupaavat parempaa yhteiseloä ohjelmistojen ja käyttäjien välille, mutta lopputuloksena saattaa olla näennäisen älykkäitä toimintoja epäyhtenäisesti toteutettuna. Miksi arvolupaukset jäävät lunastamatta?

Ohjelmistot ovat verkkopankkien, digiboxin ja kännykän myötä tulleet osaksi lähes jokaisen ihmisen arkipäivää. Muutos on huima niistä ajoista, jolloin ohjelmistojen käyttäjät olivat valkotakkisia miehiä ilmastoiduissa konesaleissa. Ohjelmistojen arkipäiväistyminen on tuonut myös uusia vaatimuksia ohjelmistojen tekijöille. Nykyään ohjelmiston käyttäjä voi olla kuka tahansa, missä tahansa ja käyttäjällä ei ole välttämättä minkäänlaista käyttökoulutusta ohjelmiston käyttöön. Kaikkien entisten laatuhyveiden lisäksi, tulee nykyaikaisen ohjelmiston olla helppokäyttöinen ja käyttökokeukseltaan miellyttävä. Miellyttävä verkkopalvelu tuottaa lisämyyntiä nykyisten asiakkaiden joukossa ja kaikkein parhaimmassa tapauksessa saa nämä suosittelemaan palvelua uusille käyttäjille.

Myös ohjelmistoteknologia on muuttunut sitten valkotakkisten miesten. Nykyiset ohjelmat eivät koostu enää pääohjelmasta ja aliohjelmista, jotka ladataan (kooltaan) suureen keskuskoneeseen, ajetaan ja sitten poistetaan uuden ajon tieltä. Nykyisissä ohjelmistoissa temmeltävät oliot, jotka heittelevät toisilleen poikkeuksia jopa tuhansia kilometrejä yhden ainoan transaktion aikana. Matka laskuaan maksavan verkkopankkiasiakkaan selaimesta, pankin oman tietojärjestelmäviidakon läpi perusjärjestelmään ja sieltä takaisin jonkin toisen pankin perusjärjestelmään saattaa mennä yli kahdenkymmenen arkkitehtuurikerroksen läpi.

Edelläkuvatussa toimintakentässä ohjelmistojen tekeminen on vaikeaa ja kallista. Vaikeaa siksi, että eri järjestelmiä ja integraatiopisteitä on paljon. Kallista siksi, että ohjelmistoprojektin

aikana ehtii muuttua niin tekninen- kuin liiketoimintaympäristö. Muutos maksaa aina.

Ketterät kehitysmenetelmät lupaavat testattuja, laadultaan parempia ohjelmistoja. Lisäksi nämä saadaan tuotantoon haukkumaan hintaansa takaisin ennätysajassa ja vielä budjetin puitteissa. Kun tähän toimintamalliin lisätään vielä käyttäjakeskeisen suunnittelun menetelmät, jolloin saadaan miellyttävä ja selkeä käyttöliittymä sekä järjestelmän ja käyttäjän välinen vuorovaikutus hiottua huippuunsa, niin mitä muuta voisi edes toivoa. Oletko kuitenkaan ollut mukana tai vieressä todistamassa helppokäyttöisten ohjelmistojen syntyä ketterien menetelmien avulla? Oletko kenties kohdannut projekteja, joissa on soudettu ja huovattu ilman tietoa siitä, mitä on saatu aikaan, missä nyt ollaan ja mikä on päämäärä? Jos haluat välttää tämän, käy läpi ennen seuraavaa projektia oheinen "tarkistuslista", jolloin sekä ketterän kehityksen että käyttäjäläheisen suunnittelun arvolupaukset toteutuvat todennäköisemmin.

Perusasiat

Perusasiat ovat aina yksinkertaisia. Ketterässä kehityksessä tulee julkaista säännöllisesti, lyhyin väliajoin ohjelmiston osia, jotka voidaan antaa jollekin käytettäväksi. Ohjelmiston käyttö voi olla oikea asiakasjulkaisu, sisäinen henkilökuntajulkaisu tai julkaisu kehitystiimeistä riippumattomille testi- tai laatuorganisaatiolle. Yhtä kaikki, jos projekti ei julkaise ensimmäisestä kuukaudesta lähtien säännöllisesti, tulee pysähtyä miettimään miksi. Karrikoiden voidaan sanoa, että tarkoitus pyhittää keinot. Vaikka prosessit ja menetelmät olisivat muuten vajavaisia, kertoo kyky julkaista palan ohjelmistoa siitä, että kyseisessä projektissa on jotkut asiat huomattavasti paremmin hallussa kuin sellaisessa projektissa, joka käyttää hienoa menetelmää, mutta ei pysty julkaisemaan mitään.

Harjoittelu ja toisto ovat ketterän kehityksen kulmakiviä, sillä näin saadaan nopealla aikataululla käytyä läpi pienimuotoinen ohjelmistopro-

jekti. Samalla toimintaan syntyy tietty rytmi ja aikaa myöden prosessissa mukanaolevat ihmiset oppivat mitä heidän tulee tehdä aina kussakin vaiheessa sykliä. Näin he pystyvät kehittämään itseohjaustaan sekä yksilöinä että tiiminä.

Lyhyin välein julkaistavat ohjelmistopalat tukevat myös käyttäjäläheisen kehityksen tarpeita. Virheiden korjaaminen ja parannusehdotusten toteuttaminen on halvinta silloin, kun näitä päätään tekemään mahdollisimman aikaisin. Parannuskohteita pystytään taas parhaiten löytämään antamalla oikea ohjelmisto oikeiden käyttäjien käyttöön. Tästä lisää seuraavassa tarkistuslistan kohdassa.

Sidosryhmät

Käyttäjäläheistä ohjelmistoa ketterin menetelmän ei voi tehdä ilman ns. maksavaa asiakasta, joka antaa projektille aina kulloisenkin suunnan ja sitoumuksen paljonko tiettyyn toiminnallisuuteen ollaan (vielä) valmiita investoimaan. Yli kvartaalirajan menevissä projekteissa on aina muutoslähteenä liiketoiminnalliset strategiamuutokset, jotka edellyttävät myös liiketoimintaa edustavien ihmisten osallistumista projektinohjaukseen. Jos todetaan, että ”meidän pankki hakee ensi kvartaalista lähtien kasvua talletuksilla”, niin tällöin kilpailevien liiketoimintasektorien toimintoihin ei aseteta seuraavissa iteraatioissa niitä kaikkein suurimpia panoksia, vaan saatetaan tyytyä aiemmassa iteraatiossa tehtyyn ”karvalakkimalliin”. Suunnainvaihto voidaan tehdä iteraation vaihtuessa, koska tavoite on saada tehtyä tuotantokelpoisia ja testattuja toimintoja jokaisessa iteraatiossa.

Liiketoiminnan lisäksi tarvitaan myös aito loppukäyttäjä, joka määrittää käytettävyyden minimiehdot kullekin toiminnolle ja verifioi sen, että toiminnallisuus kattaa ne toiminnot, jotka hän tarvitsee päästäkseen tavoitteisiinsa palvelun käyttäjänä.

Jos projektilla ei ole riittävää näkyvyyttä niin liiketoimintaan kuin käyttäjiin ihmisinä, joiden kanssa tulee voida käydä ajatustenvaihtoa ja jotka osallistuvat projektiin käyttäen koekaniineina vielä keskeneräistä palvelua, on vaarana, että projekti maalaa itsensä nurkkaan. Todellisen liiketoimintaja loppukäyttäjävaikeuden puute johtaa useimmiten siihen, että projekti alkaa itse luoda käsityksiä siitä, mitä heidän halutaan tekevän ja millainen valmiin palvelu tulisi olla. Lopputuloksena saattaa olla ketterästi metsään lipsahtanut ”turhake”, mutta vielä todennäköisemmin lopputuloksena on liian monimutkainen ja kallis palvelu, jonka jää kesken.

Muuta asiat numeroiksi

Liiketoiminnan ja käyttäjien mukanaolo ei yksistään riitä onnistuneeseen priorisointiin. Erilaisista painotuksista johtuen, saadaan pelkästään sub-

jektiivisillä tai kvalitatiivisilla mittareilla usein tuloksia, joissa saattaa olla sisällä merkittäviä virhelähteitä projektiohjauksen kannalta. Näitä virhelähteitä voidaan pienentää muuttamalla subjektiiviset arviot numeroiksi, jotka pohjautuvat joko liiketoiminnallisiin tunnuslukuihin tai muihin mitattaviin asioihin.

Useat ketterän kehityksen menetelmät painottavat liiketoiminnallisen lisäarvon tärkeyttä. Projektin olemassaolon edellytys on se, että projekti tuottaa mahdollisimman paljon ja tehokkaasti liiketoiminnalle välineitä liiketoiminnan harjoittamiseen. Joissain tapauksissa lisäarvo voidaan laskea helposti esim. tietyn toiminnallisuuden toteutuksesta saatavien hyötyjen ja toteuttamattomasta toiminnallisuudesta aiheutuvien kustannusten summasta. Tällöin saadaan konkreettiset ”kepit ja porkkanat”, jotka seuraavat eri priorisointiskenaa-riota.

Myös käytettävyyden tulee muuttua numeroiksi. Usein kuulee puhuvan ”hyvästä ja huonosta” käytettävyydestä ilman sen tarkempaa määrittelyä, mitä itse käytettävyys on. Käytettävyys ei ole jokin tietty kohta käyttöliittymässä, jokin hieno ratkaisu, vaan käytettävyys tulee määritellä käyttäjän näkökulmasta ja löytää jokin mitattava suure ilmentämään sitä. Käytettävyydelle voidaan löytää suureita, kuten opittavuus, muistettavuus, virheiden määrä ja transaktion läpimenoaika. Tämänkaltaisen operationalisoinnin jälkeen olemme aivan eri tasolla käyttäjäläheisessä suunnittelussa kuin perinteisellä ”ravista hihasta” –menetelmällä. Vaara siitä, että ”käytettävyys” alkaa elämään omaa elämää pienenee.

Kaikkein haastavin muunnos numeroiksi tehdään rajahyödyn kohdalla, kun yritetään vastata kysymykseen ”paljonko tästä kannattaa maksaa”. Useasti tämän tyyppinen tarkastelu voi olla niin vaativa ja luonteeltaan akateeminen, että on parasta pitäytyä siinä, että muistetaan panoksen ja siitä saadun hyödyn välisen suhteen olevan epälineaarinen. Käytetään siis tavallista heinänte-kojärkeä, jos investoinnit niin ominaisuuksiin kuin niiden käytettävyyteen alkavat tuntua kohtuuttomilta suhteessa havaittavaan hyötyyn.

Aloita yksinkertaisesti (ja jatka samaa rataa loppuun asti)

Jos mahdollista, aloita sekä ketterän kehityksen että käyttäjäläheisen toimintamallin harjoittelu pienessä projektissa. Näin saadut hyödyt ovat suuremmat, kun toimintaan osallistuvat henkilöt pääsevät lähemmäksi tapahtumien ydintä ja kaikki ovat tietoisia lähes kaikista asioista, joita ympärillä tapahtuu. Tärkeää on olla tietoinen myös niistä asioista, joita ei itse suorita, koska tämä tieto saattaa auttaa ymmärtämään paremmin oman osuuden merkityksen ”isossa kuvassa” ja siten edesauttaa itseohjautuvuuden kehittä-

*”Myös
käytettävyys
tulee muuttaa
numeroiksi.”*

tymistä. Vasta sen jälkeen, kun pieni ydin osaa toimia uudessa viitekehyksessä on aika skaalata projektia vastaamaan kokonaistarvetta.

Toinen asia, jossa kannattaa olla varovainen, on teknisten ratkaisujen toteutus. Mitä vaikeampi arkkitehtuuri ja kovempi pyrkimys yleiskäyttöiseen, laajennettavaan ja helposti ylläpidettävään ratkaisuun, sitä suurempi on riski epäonnistua. Tämä koskee erityisesti kunkin toiminnallisuuden ensimmäisiä iteraatioita. Nyrkkisääntönä voisi sanoa, että ensimmäisessä iteraatiossa kannattaa asettaa tavoitteeksi saada aikaan, jotain joka toimii ja vasta sitten murehtia siitä, että oliko ratkaisu tyylikäs. Refaktoriointi, ohjelman sisäisen rakenteen parantaminen toiminnan pysyessä samana, ei ole merkki epäonnistuneesta suunnittelusta, vaan siitä, että näkemys tarkentuu jokaisessa iteraatiossa.

Yksinkertaisuus on valttia myös ohjelmistotuotantoprosessissa ja siinä tehtävissä suunnittelukuvauksissa. Useimmiten paras ratkaisu on alkaa ns. puhtaalta pöydältä ts. tehdä jokin pieni kokonaisuus ensimmäisessä lyhyessä iteraatiossa ilman prosessia ja prosessin vaatimia artefakteja. Ainoa sääntö tällöin on, että julkaisu tai demo tapahtuu iteraation päätteeksi. Kehitystiimi tulee tunnistamaan viimeistään kolmannessa iteraatiossa ne prosessivaiheet ja suunnittelukuvaukset, joita se todella tarvitsee työssään. Näin vältymme pöyhötyneeltä toimintamallilta, jossa useimmat roolit, dokumentit ja prosessien vaiheet ovat kuolleita kirjaimia.

Lähtökohtaisesti ketterät ja käyttäjäläheiset menetelmät ovat haastavia projektin muutoshallinnan kannalta. Molemmat suosivat iteraatioita ja toivottavat tervetulleiksi käyttäjäpalautteesta

saavat muutokset. Näin ollen muutoshallinnan on oltava kunnossa. Tärkein väline muutoshallintaan on "product backlog", joka pitää sisällään kaiken tekemättömän työn, myös muutokset ja korjaukset aiemmin tehtyihin toimintoihin. Product backlog on pidettävä aina kunnossa, sillä se on projektin projektisuunnitelma ja muutoksenhallintaväline samanaikaisesti. Ilman product backlogia ei tiedetä mitä on saatu aikaan, missä nyt ollaan ja mitä tehdään seuraavaksi.

Pelkästään product backlog ei riitä muutoshallinnan onnistumiseen. Epäyhteneväisyyksistä johtuvia muutoksia, oli ne sitten peräisin erilaisista käyttöliittymäratkaisuksista tai useamman toteuttajan kirjoittamasta erilaisista koodirakenteista, tulee minimoida pitämällä säännöllisesti koodi- ja käyttöliittymäkatselmoitteja. Toinen tärkeä mekanismi turhien (=ei käyttäjälähtöisten) muutosten välttämiseen on ottaa projektissa käyttöön heti alusta alkaen riittävän tarkat tyyli- ja ohjelmointiohjeet. Tyyliohjeissa määritellään miltä jokin asia käyttöliittymässä näyttää ja millä tavoin se reagoi käyttäjän toimiin. Ohjelmointiohjeet taas määrittelevät, miten ohjelmat tulisi toteuttaa ja koodi dokumentoida. Näin vähennämme itse aiheutetun muutospaineen määrää.

Yhteenveto

Jos havaitisit, että tarkistuslistalla oli asioita, jotka projekteissasi eivät ole toteutuneet, niin prosessin kehittäminen todennäköisesti kannattaa. Pidä kehittämistyössä kuitenkin jalat maassa ja pää kylmänä. Terve järki vie pidemmälle kuin kirjaviisaus ja fundamentalismi.

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 www.ttlry.fi jasenasiat@ttlry.fi
02600 Espoo etunimi.sukunimi@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



Eija Kalliala

vaatimus jäätyy putoukseen, pyrähtää sulaen scrumiin

rup ruoppaa pienen palan valmiiksi, muuttaa lennossa suuntaa

Pyrähtelevä ohjelmistokehitys

Nykyaikaiset ohjelmistokehityksen mallit olivat aiheena Sytyken Mallinnus-Osyn jäsentilaisuudessa 17.9.07 SYSO-PENDIGIAlla.

Paula Salmi Osuuspankkikeskuksesta (OPK), nykyisestä OP-Pohjolasta esitteli heidän kehittämänsä V5-mallin, Allan Halme Accenturalta pureutui RUPin teoriaan ja Lasse Koskela Reaktor Innovationsista pohdiskeli Scrumin taustaa ja todellisuutta tänään.

Yleisössä istuneet kokeneet systeemyöläiset toisaalta kritisoivat Allan Halmen turhan teoreettista esitystä RUPista, toisaalta täydensivät Lasse Koskelan esitystä omilla kokemuksillaan scrumista.

V5-malli kehittää

OP-Pohjolan organisaatiouudistus pyrki eroon eri palvelualueilla toimivien järjestelmäkehittäjien erillisten ratkaisujen koordinaatioongelmista. Ict-palvelut ostetaan keskitetystä ict-organisaatiosta, vaikka liiketoiminnan tavoitteet määritetään edelleen palvelualueilla ja liiketoiminnan edustajat päättävät hankkeiden priorisoinnista ja etenemisestä.

Kehitysprosessin vaiheet noudattelevat perinteistä systeemyömallia idean kiteyttämisestä pilotointiin, jalkauttamiseen ja hyötyjen mittaukseen. Malli on ryhdistänyt ja selkeyttänyt prosessia ja vastuunjakoa, mutta haasteet eivät ole kadonneet: tehtävät ja dokumentit ovat vielä osin päällekkäisiä ja riittävän informaation takaaminen kapulanvaihtotilanteissa eri yksiköissä toimivien määrittelijöiden ja testaajien välillä vaatii ponnistelua.

Paula Salmi kevensi esitystään kuvalla

näppikseen nojaavasta pääkallosta, jota näytöltä tuijotti teksti "Waiting for reply". Hän kertoi ketterän kehittämisen käynnistyneen OP-Pohjolassa sertifoitujen projektipäälliköiden ja OWASP-tietoturvaan erikoistuneiden testaajien myötä.

RUP palastelee

Allan Halme vertasi RUPia vesiputousmalliin. RUPissa vaatimuksia ei jäädytetä, mutta koodaus kannattaa aloittaa vasta kun neljä viidesosaa järjestelmästä on määritelty. Kiinteähintainen tilaus edellyttää mahdollisimman hyvää vaatimusmäärittelyä.

Systeemyötä voi verrata Firenzen katedraalin kupolin rakentamiseen 1400-luvulla: järjestelmästä saa todellisen kuvan vasta, kun toteutus on jonkin matkaa edennyt ja nähdään, pysyykö toteutus sovitussa raameissa ja onko järjestelmän suorituskyky riittävä.

Kun tilaajalta tulee uusia tarpeita kesken projektin, RUPissa osa järjestelmästä on valmiina ja suuntaa voidaan muuttaa kohti uusia tavoitteita. Vesiputousmallissa taas kaikki olisi levällään eikä suunnan muuttamiseen löytyisi kiintopistettä.

Yleisö oli aktiivinen: Vastaava palasteleva suunnittelu-toteutus-idea kehitettiin jo 1990-luvun alussa Sytyken STST-työryhmässä, nykyisen Rela-yhteisön edeltäjässä. OP-Pohjolan V-malli ei tietenkään noudata vaihejakoa vesiputousmaisesti, kuten Allan Halme tulkitsi, vaan sisältää luonnollista iteraatiota, kun käyttäjät, ongelma, teknologia tai markkinat saattavat muuttua projektin aikana. Esitetyt RUPin työmääriä kuvaavat käyrät eivät toimi versioivassa tuotekehityksessä, jossa vaatimusmää-

rityksen työ määrää ei vähene työn edetessä.

Scrum pyrähtää

Lasse Koskela esitti Scrumin taustatiedot kuvaamalla Toytan toisen maailmansodan jälkeisestä innovoivaa tuotekehityksestä, jossa pienin askelin tehokkaasti tehtiin jotain konkreettista, mikä tuotti lyhimmän kehitysajan ideasta valmiiseen tuotteeseen. Tältä pohjalta kehitettiin Scrum 1990-luvun alussa.

Asiakkaan priorisoimat tärkeät kehittämistoimenpiteet kootaan Product Backlogiin, josta kahdesta neljään viikkoon kestävä pyrähdys työmäärä poimitaan Sprint Backlogiin. Pyrähdys jälkeä tuotos on valmis eikä vaadi hiomista.

Ict-ammattilaisten velvollisuus on selvittää asiakkaalle priorisointiin ja toteutusjärjestykseen vaikuttavat tekniset asiat. Yleisön joukosta todettiin, että Scrumissa käytetään jatkuvasti aikaa Backlogin kampaamiseen, ettei se happe.

Scrumia opitaan tekemällä, ei kursilla. Prosessi on yksinkertainen, mutta vaikea soveltaa. Scrumin toimijoita ovat tuotteen omistaja, moniosaava ja itseohjautuvat tiimi, johon kuuluu 5-9 henkilöä sekä Scrum-master. Scrum-master on kuin voi leivän välissä, poistaa esteitä ja ylimääräisiä välikäsiä sekä huolehtii prosessin järkevistä soveltamisesta.

Asiakkaiden kanssa keskustellaan heidän ymmärtämällä kielellä. Siihen ei kuulu pariohjelmointi eikä Scrum-master.

Käyttäjäkeskeinen suunnittelu ja ketteryys

– Ristiriitaiset lähestymistavat vai arvokas synergia

Käyttäjäkeskeinen suunnittelu (User-centered design) ja ketterä ohjelmistokehitys ovat yleisesti käytössä modernissa ohjelmistokehityksessä. Molemmat ovat iteratiivisia ja nostavat asiakkaan ja käyttäjät keskeiseen rooliin ohjelmistotuotantoprosessissa. Yhteisistä päämääristä huolimatta ketterien menetelmistojen ja käyttäjäkeskeisen suunnittelun yhdistäminen on harvinaista. Ovatko lähestymistavat siis keskenään ristiriidassa vai eikö niiden yhdistämisestä seuraa riittävä hyötyä?

Käyttäjäkeskeinen suunnittelu on 1980-luvun puolivälissä kehitetty kokoelma ohjelmistokehityskäytäntöjä ja myös oma ohjelmistokehitysfilosofiansa. Kantava ajatus on käyttäjän tuominen keskeiseksi osaksi ohjelmistokehitysprosessia. Käyttäjä yhdistetään prosessiin haastatteleamalla tulevia käyttäjiä, tarkkailemalla heitä työssään sekä tutkimalla ja analysoimalla käyttäjäryhmän tietoja. Käyttäjälle myös näytetään prototyyppejä järjestelmästä ja pyydetään häntä tekemään prototyypeillä toimintoja. Näistä tilanteista saatujen palautteiden, tietojen ja havaintojen avulla löydetään uusia vaatimuksia ohjelmistolle, parannuksia käyttöliittymään ja ymmärretään eri ominaisuuksien tärkeys paremmin.

Suurin osa tästä työstä tehdään ennen varsinaisen ohjelmistokehitystyön aloittamista, joka taas toteutetaan iteratiivisesti pitäen kohdekäyttäjät jatkuvasti ohjelmistokehittäjien mielissä. Tässä auttaa esim. persoonien määrittäminen, joka perustuu keskeisten käyttäjäryhmien tunnistamiseen ja niiden personoimiseen luomalla kuvitteellisia esimerkkikäyttäjiä. Esimerkkikäyttäjistä kehittäjät tietävät lähes kaiken: perhesuhteet, harrastukset, taidot ja rajoitukset. Esimerkiksi Yahoo!n web-sovellusten suunnittelussa nämä esimerkkihenkilöt ovat keskeisessä roolissa päätettäessä tuotteen ominaisuuksien tarpeellisuudesta, käyttöliittymän monipuolisuudesta ja jopa toteutustekniikasta.

Ketterä ohjelmistokehitys on yhteisnimitys kevyille ja muutokseen mukautuville ohjelmistokehitysprosesseille. Ketterille menetelmistöille tärkeimmät arvot on määritelty Ketterän ohjelmistokehityksen manifestissa (Agile manifesto).

Tärkeimpiä asioita ohjelmistokehityksessä ovat yksilöt ja näiden välinen vuorovaikutus, toimiva ohjelmisto, yhteistyö asiakkaan kanssa sekä muutokseen vastaaminen. Näiden arvojen pohjalta olisi hankala suoraan lähteä kehittämään ohjelmistojä, joten manifeston yhteydessä on myös määritelty 12 periaateetta, jotka ohjaavat tarkemmin työkentelyä ohjelmistokehityksiprojekteissa.

Ketteryys vs. käyttäjäkeskeisyys

Ketterät menetelmistöt, eteenkin Extreme programming painottavat usein etukäteissuunnittelun merkityksettömyyttä ja sen aiheuttamaa hukkatyötä ja muutosjäykkyyttä. Käyttäjäkeskeinen suunnittelu taas lähtee ajatuksesta, että ennenkuin lähdetään tekemään riviäkään toteutusta, tulisi olla tarkka kuva kohdekäyttäjistä ja jo ensimmäiset käyttöliittymäluonnokset valmiina. Kent Beck, Extreme programmingin kehittäjä, piti tätä etukäteissuunnittelua käyttäjäkeskeisen suunnittelun suurimpana ongelmakohtana, koska se tekee käytettävyyssuunnittelijoista projektien pullonkaulan ja samalla keskittää päätöksentekoa.

Ketterät menetelmistöt myös painottavat dokumentaation keveyttä, jopa niinkin paljon, että valloilla on yleinen harhakäsitys dokumenttien puuttumisesta kokonaan ketteristä ohjelmistokehitysprojekteista. Käyttäjäkeskeinen suunnittelu taas pohjaa pitkälti erilaisiin tutkimusdokumentteihin, käyttöliittymäsuunnitelmiin ja esimerkkihenkilöjulisteesiin.

Yhteneväisyksiäkin tuki löytyy näistä malleista. Molemmat haluavat luoda useita prototyyppejä, joista pyydetään palautetta ja tämän perusteella jatketaan kehitystyötä eteenpäin. Käyttäjäkeskeinen suunnittelu hakee tämän palautteen käyttäjiltä, Extreme programming pyytää palautetta asiakkaalta, joka vastaa myös käyttäjien tarpeista ja Scrum hakee tämän palautteen jokaisen sprintin eli iteraation jälkeisestä demo-tilaisuudesta, jossa voi olla paikalla sekä liiketoiminnan edustajia että varsinaisia käyttäjiä. Palaute tulee siis ketterissä menetelmistöissä useimmiten asiakkaan vastuulliselta edustajalta, joka toimii parhaansa mukaan välittääkseen käyttäjien tarpeet, kun taas käyttäjäkeskeinen suunnittelu hakee palautetta varsinaisilta loppukäyttäjiltä. Molempien mallien samanaikainen käyttö tuo luonnollisesti enemmän

Yhteneväisyksiäkin tuki löytyy näistä malleista. Molemmat haluavat luoda useita prototyyppejä, joista pyydetään palautetta ja tämän perusteella jatketaan kehitystyötä eteenpäin.

palautetta ja antaa paremman kuvan niin asiakkaan kuin loppukäyttäjien tarpeista.

Sekä käyttäjäkeskeisessä suunnittelussa että ketterissä menetelmistöissä on käyttäjä tai asiakas yhtenä keskeisimpänä roolina kehitysprosessissa. Heidän tulee ottaa osaa ohjelmistokehitykseen, jotta kehittäjät tuottavat ohjelmiston, joka vastaa heidän eksplisiittisiä sekä implisiittisiä tarpeitaan mahdollisimman hyvin. Jotta tätä ohjelmistoa voidaan tarkistaa tasaisin väliajoin, ovat molemmat lähestymistavat iteratiivisia.

Ohjelmistot sopiviksi palautteella

Palaute on siis molemmille hyvin keskeinen ohjelmistokehitystä ohjaava tekijä. Ketterät menetelmistöt aloittavat kehityksen tärkeimmistä korkean tason vaatimuksista, tulkitsevat niitä sekä kyselevät asiakkaalta lisätietoja ja luovat näiden perusteella jo heti ensimmäisessä iteraatiossa toimivan ohjelmiston, joka annetaan asiakkaalle testattavaksi. Asiakkaan palautteen mukaan ohjelmistoa muokataan ja rakennetaan edelleen, kunnes päästään asiakkaan tarpeet täyttävään sovellukseen. Erityisen tuskallista tämä saattaa olla käyttäjäliittymän osalta, jonka hiominen tyydyttävään muotoon voi kestää melko pitkään. Myös käyttäjäliittymän ja konseptien yhtenäisyyden kannalta tämä malli vaatii erityistä tarkkaavaisuutta. Tässä suhteessa käyttäjäkeskeinen suunnittelu voi avittaa ketteriä menetelmistöjä.

Käyttäjäkeskeisen suunnittelun käyttö projektin alussa mahdollistaa yhtenäisen käyttäjäliittymän suunnittelun, tekemättä kuitenkaan suuria ohjelmistokehitystä rajoittavia suunnittelupäätöksiä. Sen sijaan että iteroiden ja testaten etsitään käytettävyydeltään oikeanlaista käyttäjäliittymää, voidaan tämä saada suurinpiirtein kohdalleen heti ensimmäisestä iteraatiosta alkaen. Esimerkiksi käyttäen paperiprototyyppejä käyttäjäliittymien hahmottelemiseen ja käyttäjien kanssa kommunikointiin voidaan nopeasti saada selville käyttäjäliittymän yleinen rakenne. Paperiprototyypit eivät ole turhia ja ohjelmiston kannalta merkityksellisiä dokumentteja vaan arvokkaita ja testattavia käytettävyyksille, joiden pohjalta voidaan helposti luoda toimiva käyttäjäliittymä.

Persoonien määrittämisen tehoa on myöskin kokeiltu ketterissä ohjelmistokehitysprojekteissa ja kokemukset ovat olleet lupaavia. Näkyvillä paikoilla sijaitsevien persoonien on nähty muistuttavan kehittäjiä keille he ovat ohjelmistoa tekemässä. Samalla päämäärä ja kokonaiskuva pysyy selkeämpänä. Nämä esimerkkikäyttäjät myöskin helpottavat päätöksentekoa, koska kehittäjillä on tarkempi kuva käyttäjistä ja heidän tarpeistaan. Toisaalta nämä persoonat on melko helppo unohtaa teknisten asioiden ollessa päällimmäisenä, joten persoonien määrittäminen ei yksinään varmista, että kehittäjillä on loppukäyttäjä mielessä päätöksiä tehdessään.

Sudenkuopista synergiaan

Kokemukset näiden menetelmien yhdistämisestä ovat olleet positiivisia, joskin luonnollisesti myös ongelmia on havaittu. Yksi yleisimmistä ongelmista näiden kahden menetelmän yhdistämisessä on ollut käytettävyyssuunnittelijoiden ja kehittäjien välinen kommunikointi. Molemmilla tahoilla on havaittu taipumusta suojautua toisen ryhmän tekemiä päätöksiä vastaan. Kehittäjien tehdessä päätöksiä käyttäjäliittymän tiimoilta vastustavat käytettävyyssuunnittelijat automaattisesti näitä. Sama pätee myös toinpäin ja tämä valtataistelu johtaa helposti haitalliseen "me ja ne muut" ajatteluun.

Jotta nämä menetelmät toimisivat mahdollisimman hyvin yhdessä, tarvitsee näihin muutamiin ongelmakohtiin kiinnittää huomiota. Ensinnäkin kehittäjien ja käytettävyyssuunnittelijoiden välistä eroa tulisi kaventaa ja parantaa näiden kommunikointia. Ihanteellista olisi luonnollisesti, jos kehittäjätkin hallitsisivat käytettävyyssuunnittelun periaatteet. Toiseksikin käyttäjien sitouttamiseen tulisi panostaa. Käyttäjät pitäisi saada mukaan lähes koko kehitysprojektin ajaksi ja heiltä täytyy saada arvokasta palautetta käytettävyydestä.

Erilaisilla projekteilla on erilaiset tarpeet. Tämä on hyvä ottaa huomioon käyttäjäkeskeistä suunnittelua ja ketteriä menetelmistöjä yhdistettäessä. Joissakin projekteissa käyttäjäkeskeisestä suunnittelusta ei ole hyötyä. Joissakin projekteissa ei ole aikaa tehdä kattavaa tutkimusta käyttäjistä, vaan ensisijaisena tarkoituksena on tuoda mahdollisimman nopeasti ohjelman ensimmäinen versio markkinoille. Joissakin projekteissa on hyvä suunnitella 90% käyttäjäliittymästä valmiiksi etukäteen, toisissa taas riittää karkea malli 60%:sta käyttäjäliittymästä. Joskus taas ei ole mahdollisuutta saada oikeita loppukäyttäjiä testaamaan ohjelmistoa ja paperiprototyyppejä. Käyttäjäkeskeisen suunnittelun käytölle on oltava selkeä tarve, jotta sitä kannattaa yrittää yhdistää ketteriin menetelmistöihin. Ilman tarpeiden ymmärtämistä voi tämä tekniikka hankaloittaa ohjelmistokehitystä.

Käyttäjäkeskeinen suunnittelu ja ketterät ohjelmistokehitysmenetelmät ovat siis yhteensovittavissa olevia tekniikoita. Niiden oikeanlainen yhdistäminen tuo usein lisäarvoa asiakkaalle, vähentää ominaisuuksien uudelleentyöstämistä ja helpottaa ohjelmistokehitystyötä. Eri tekniikoiden tuntemusta tosin vaaditaan, jotta niitä voidaan käyttää tehokkaasti yhdessä. Kaikenkaikkiaan jokaisessa ohjelmistokehitysprojektissa tulisi pääasiallisena tavoitteena olla asiakkaan ja käyttäjien tarpeiden täyttäminen sekä parhaan mahdollisen ohjelmiston kehittäminen -- ei puhtaiden kehittämisprosessien orjallinen noudattaminen, vaan prosessinkin ketterä räätälöinti kulloisenkin projektin tavoitteet parhaiten täyttäväksi.

Käyttäjäkeskeisen suunnittelun käyttö projektin alussa mahdollistaa yhtenäisen käyttäjäliittymän suunnittelun, tekemättä kuitenkaan suuria ohjelmistokehitystä rajoittavia suunnittelupäätöksiä.

Käyttäjäkeskeinen suunnittelu ja ketterät ohjelmistokehitysmenetelmät ovat siis yhteensovittavissa olevia tekniikoita.

Kokemuksia GUIDe-käyttöliittymäsuunnittelun ja Scrum-menetelmän yhdistämisestä



Karri-Pekka Laakso suunnittelee ja toteuttaa käyttöliittymiä Reaktor Innovationsissa.

Reaktor Innovationsissa on jo vuosien ajan käytetty ketterää Scrum-menetelmää käytännössä kaikissa ohjelmistoprojekteissa. Vuodesta 2005 lähtien ohjelmistojen käyttöliittymäratkaisuja on alettu parantaa laatimalla ne GUIDe-käyttöliittymäsuunnittelumenetelmää noudattaen. Näiden menetelmien yhdistäminen onnistuneesti ei ole ollut aivan itsestään selvää ja vaivatonta, ja siksi tässä artikkelissa jaetaan Reaktorissa keräytyjä kokemuksia.

Miksi ketterät menetelmät tarvitsevat systemaattista käyttöliittymäsuunnittelua?

Käyttöliittymäsuunnittelu on monitulkintainen termi, jolla joskus tarkoitetaan esimerkiksi pelkkää visuaalista suunnittelua tai vaatimusmäärittelyvaiheessa päätettyjen toimintojen asettelemista näytölle. GUIDe-mallin [GUIDe04] mukainen käyttöliittymäsuunnittelu on kuitenkin pikemmin-

kin vaatimusmäärittelyn työvaihe, joka tuottaa vastaukset seuraaviin kysymyksiin:

mitä tietoja järjestelmä käyttäjälleen näyttää (tietosisältö käyttäjän kannalta),

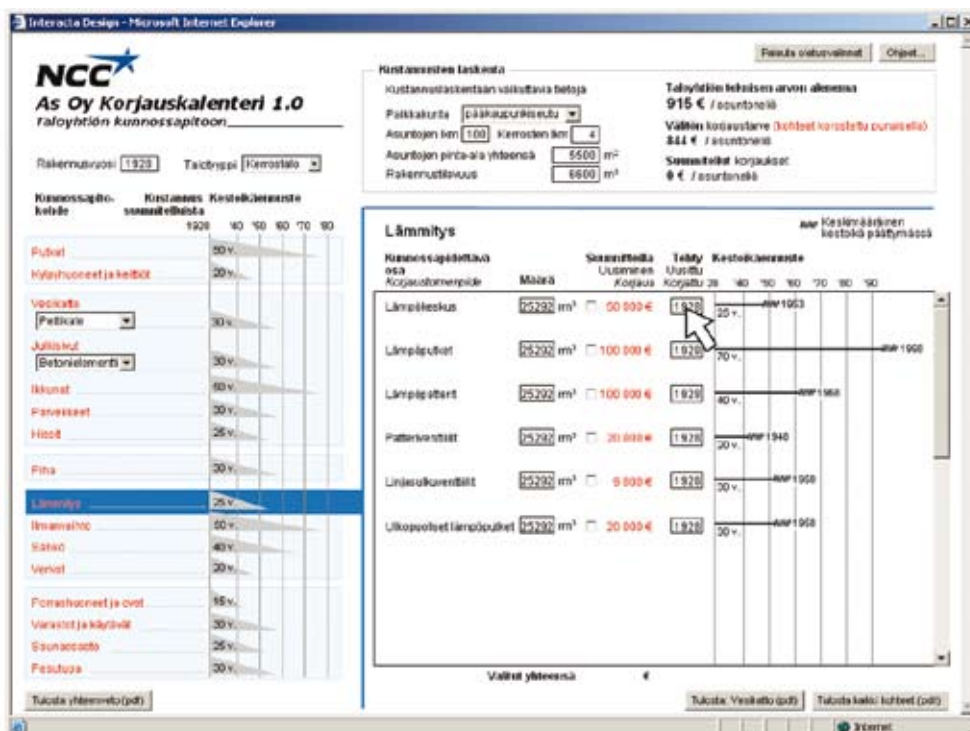
miten se näyttää nämä tiedot (tietojen organisointi ja visualisointi näytöllä),

mitä käyttäjä voi tehdä järjestelmällä (toiminnallisuus) ja

miten järjestelmä reagoi, kun käyttäjä tekee nämä asiat (toimintalogiikka).

Tällaisessa käyttöliittymäsuunnittelussa on siis kyse yksityiskohtaisesta tietosisällön ja toiminnallisuuden määrittelystä, joka tehdään käyttötilanteiden suorittamista simuloimalla. GUIDe-mallissa se kulminoituu yksityiskohtaiseen käyttöliittymäspeksiin, joka koostuu lopullisen ohjelmiston kaltaisista näyttökuvista (vrt. kuva 1, ensimmäinen näyttökuva). GUIDen käyttöliittymäsuunnittelua ja sen seurauksia on kuvattu Systemeemyö-lehden numerossa 2/2006 [Sytyke06].

Kuva 1. NCC:n Korjauskalenterin alkuperäinen käyttöliittymäkuva (Interacta).



Ketterissä menetelmissä vastaava työ hoidetaan yleensä kirjoittamalla käyttäjätarinoita, mutta niihin sisältyy käsitteenä paljon huonosti määriteltyjä kohtia: Kuka tarinat tekee, asiakas vai toimittaja? Kuka on vastuussa tarinoista? Kuvaako käyttäjätarina ongelman (mitä järjestelmällä pitäisi saada aikaan) vai ratkaisun (mitä toiminnallisuutta järjestelmään tulee)? Miten käyttäjätarinoista saadaan aikaan käyttöliittymäratkaisu (esimerkiksi tietojen organisointi ja toimintalogiikka) ja kuka sen laatii?

Käytännössä ketterää menetelmää käyttävä toimittaja työntää yleensä perimmäisen vastuun määrittelystä sekä sen kattavuudesta ja oikeellisuudesta asiakkaalle. Asiakas ohjataan kirjoittamaan käyttäjätarinaa ratkaisusta, jolloin itse asiassa asiakas yrittää suunnitella järjestelmän toiminnallisuuden itse. Tästä seuraa usein kömpelö järjestelmä, koska asiakas ei tyypillisesti osaa käyttöliittymäsuunnittelua.

GUIDen mukaisessa toimintatavassa asiakkaiden toiminnallisuustoiveita (ratkaisuja) ei oteta toteutettavaksi sellaisinaan, vaan ensin käyttöliittymäsuunnittelija selvittää, mikä on asiakkaan tarvetta vastaava käyttötilanne (ongelma), joka pitäisi ratkaista. Esimerkiksi ammattikorkeakoulun opetuksen suunnittelujärjestelmän asiakas taannoin pyysi, että järjestelmään tehtäisiin itsepalveluraportointi, jossa käyttäjä voisi hakea opetustapahtumia millä tahansa kentillä. Osoittautui, että konkreettisia hakutilanteita, joita tämä haku palvelisi, oli karkeasti ottaen kaksi: 1) opettaja kirjoittaa omista kursseistaan sisältökuvauksen ja aikataulun opiskelijoille, ja 2) hallinnon työntekijä tekee tilastokeskukselle raportin edellisen vuoden opetuksesta. Kumpaankaan näistä ongelmista itsepalveluraportointi ei ollut kovin hyvä ratkaisu: opettajalle kannattaa mieluummin näyttää lista hänen omista kursseistaan, ja hallinnon työntekijälle luoda suoraan tilastokeskuksen tarvitsema raportointimuoto.

GUIDe-mallin mukainen käyttöliittymäsuunnittelu täydentää hyvin ketterien menetelmien määrittelyvaiheen aukkoja, ks. laatikko.

Käyttöliittymästä toteutukseksi

Reaktorin projekteissa käyttöliittymäsuunnittelu sijoitetaan aivan projektin alkuun, mieluiten siten, että toteutustyö aloitetaan vasta käyttöliittymän valmistumisen jälkeen. Jos projektissa ollaan tekemässä kokonaan uutta järjestelmää, alku etenee GUIDeä noudattaen: ensin tehdään käyttötilanneselvityksiä käyttäjien luona, ja sitten suunnitellaan niiden pohjalta ratkaisut, jotka testataan käyttäjien kanssa.

Käyttöliittymäratkaisut kommunikoidaan toteutustiimille käyttöliittymäkuvina tai -kuvasarjoina. Tämän lisäksi käyttöliittymäsuunnittelija demoo ratkaisun toiminnan toteuttajille realistisilla käyttötilanteilla, jotta toteuttajat varmasti ymmärtäisivät ratkaisun toimintalogiikan oikein. Reaktorissa on pyritty kirjoittamaan mahdollisimman vähän dokumentaatiota ja panostamaan sen sijaan kommunikointiin: ideaalitulanteessa käyttöliittymäsuunnittelija istuu toteutustiimin kanssa koko projektin ajan saman pöydän ääressä tai samassa huoneessa, jolloin toteuttajat voivat milloin vain kysyä häneltä täsmennyksiä ja tarkennuksia. Käytännössä tähän päästään kuitenkin valitettavan harvoin, koska yhdellä suunnittelijalla on helpostikin käynnissä 2-5 projektia samaan aikaan.

Toteutusta varten käyttöliittymäsuunnittelija ja toteutustiimi purkavat käyttöliittymän toiminnoiksi tai käyttäjätarinoiksi, joista muodostuu projektin toimintolista (product backlog). Backlogin priorisoinnissa kiinnitetään erityisesti huomioita

GUIDe-mallin mukainen käyttöliittymäsuunnittelu täydentää hyvin ketterien menetelmien määrittelyvaiheen aukkoja:

- Käyttötilanteet kuvaavat käyttäjän ongelman eivätkä lainkaan ratkaisua (niissä ei esiinny toteutettavaa järjestelmää lainkaan).
- Käyttöliittymäratkaisu kuvaa yksikäsittisen ja suoraviivaisen ratkaisun em. ongelmiin.
- Ratkaisu esitetään kuvina ja kuvasarjoina, joita lukijan on helppo ymmärtää.
- Käyttöliittymäratkaisun laati käyttöliittymäsuunnittelija (vastuu ratkaisusta on siis toteuttajalla, ei asiakkaalla).
- Käyttötilanteet selvitetään ensisijaisesti järjestelmän tulevilta käyttäjiltä, ja asiakkaan rooli on ohjata suunnittelijat oikeiden käyttäjien luo tutkimaan heidän työtään.
- Ratkaisun oikeellisuutta ja kattavuutta voidaan testata jo ennen toteutusta.

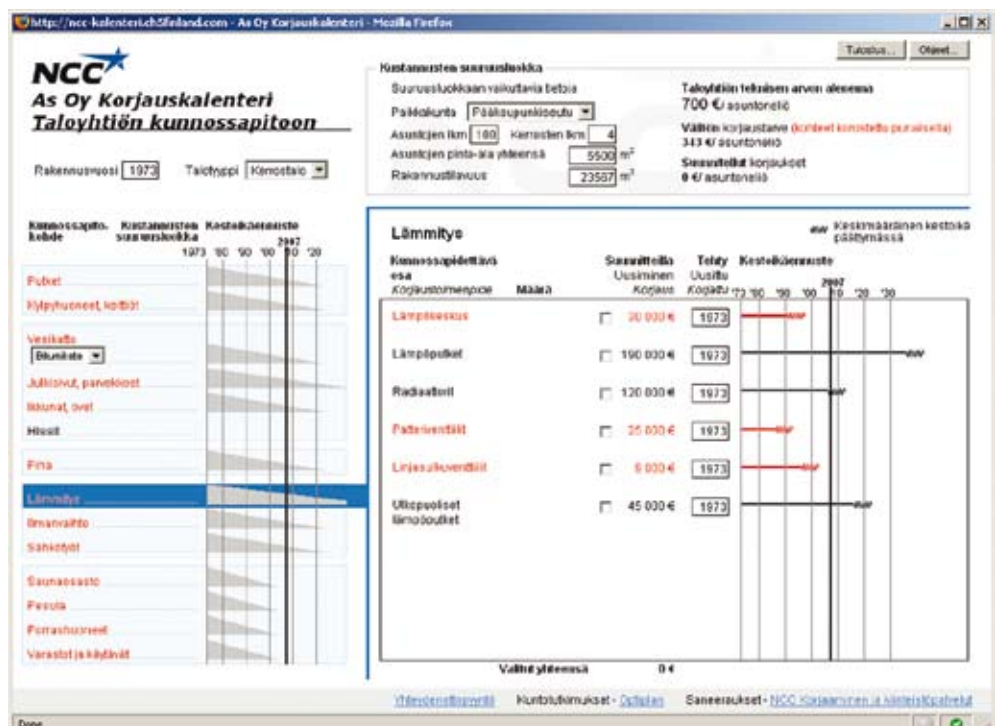
siihen, että käyttäjä voisi mahdollisimman varhaisessa vaiheessa suorittaa järjestelmällä kokonaisia käyttötilanteita loppuun asti.

Kun toteuttaja on koodannut omasta mielestään valmiiksi jonkin toiminnallisuuden, hän antaa sen käyttöliittymäsuunnittelijalle hyväksyttäväksi. Hyväksynnässä suunnittelija tarkistaa, että toteutus todella vastaa interaktiotavoiltaan käyttöliittymäratkaisua. Samalla hän myös testaa ratkaisua yksinkertaisilla poikkeustilanteilla. Asiakkaalle toiminnon kerrotaan olevan valmis vasta sen jälkeen, kun käyttöliittymäsuunnittelija on hyväksynyt toteutuksen.

Koettuja haasteita...

GUIDe-käyttöliittymäsuunnittelun ja Scrumin yhteenliittäminen ei ole sujunut kitkatta. Esimerkiksi projektin aloittaminen käyttöliittymäsuunnittelulla on näennäisesti ristiriidassa Agile

Kuva 2: Kuva lopullisesta toteutuksesta (Reaktor).



manifeston [Agile01] kanssa: toteutustyön pitäisi periaatteessa alkaa heti projektin ensimmäisistä päivistä lähtien. Tämän ajattelutavan vuoksi GUIDe-menetelmää käyttöönottaessa oli vaikea perustella, miksi koodaamista kannattaisi lykätä myöhemmäksi. Todellisuudessa järjestelmän käyttöliittymän määrittely koodaamalla on erittäin hidasta ja kallista verrattuna kynällä paperille piirrettyihin käyttöliittymäkuviin, ja lisäksi se tuottaa helposti huonoja ratkaisuja. Manifesto onkin tähdätty hyödyntä dokumentaatiota vastaan. GUIDen käyttöliittymäsuunnittelu ei ole manifeston tarkoittamaa ylimääräistä dokumentointia, vaan siinä järjestelmän käyttöliittymää kehitetään hyvin konkreettisesti piirtämällä kuvia ketteriä menetelmiä vastaavissa nopeissa suunnittelu-testaus-sykleissä – ainoastaan kymmeniä kertoja koodausta nopeammin.

Erityisesti jos projektin kustannukset halutaan pitää mahdollisimman pieninä eikä projektilla ole kova kiire, koodaaminen kannattaa aloittaa vasta käyttöliittymäsuunnittelun jälkeen. Näin ei tehty vielä ensimmäisissä projekteissa. Esimerkiksi NCC:lle tehty Korjaukskalenteri-sovellusta [NCC06] kehitettäessä Reaktor aloitti toteutustyöt samaan aikaan, kun Interacta Design Oy aloitti sovelluksen käyttöliittymäsuunnittelun. Käyttöliittymä-ratkaisun valmistuttua osoittautui, että kaikkea siihen mennessä tehtyä toteutustyötä ei olisikaan välttämättä tarvittu, vaan taustajärjestelmäksi olisi riittänyt yksinkertaisempikin toteutus. Tämä ei suinkaan ollut selvää projektin alkaessa, vaan se selvisi vasta käyttöliittymä-ratkaisun avulla. Toteutuksen työmäärässä voitaneen säästää kymmeniä prosentteja, jos käyttöliittymä-ratkaisu on etukäteen selvillä.

Yksinkertaisimmillaan Reaktorin ketterässä projektissa lähes kaikki työ on koodausta ja kaikki projektiin osallistuvat henkilöt osallistuvat vain tähän yhteen projektiin. Näin ei enää olekaan, kun projektiin sisältyy käyttöliittymäsuunnittelu. Silloin projektit pitäisi ensinnäkin miehittää niin, että toteuttajat tekisivät muita projekteja käyttöliittymäsuunnittelun ajan. Toiseksi projektipäällikön pitää huolehtia hyvissä ajoin siitä, että tiimillä riittää koko ajan mielekästä koodattavaa: toimintoja ei enää saadakaan toteutettavaksi ilman käyttöliittymäsuunnitteluvaihetta, vaan käyttöliittymäsuunnittelu on toteutuksen kriittisellä polulla. Jos käyttöliittymäsuunnittelijoiden muiden projektien työkuorma viivästyttää toteutusta, toteuttajat tyypillisesti kannattaa siirtää toisiin projekteihin siksi ajaksi, että käyttöliittymä-ratkaisu valmistuu.

... ja koettu onnistumisia

Hyväksymismenettely, jossa toteutettu toiminnallisuus palaa käyttöliittymäsuunnittelijan hyväksyttäväksi, on osoittautunut erittäin hyödylliseksi ja tärkeäksi vaiheeksi. Toteuttajat eivät ole yleensä varautuneet siihen, millaista laatua käyttöliittymätoteutuksen osalta heiltä odotetaan. Tuloksena on usein toteutus, joka pääsääntöisesti toimii jotenkuten mutta on interaktioiltaan puutteellinen tai virheellinen. On aivan tyypillistä, että uuden tiimin ensimmäisen sprintin kaikki toteutetut toiminnallisuudet palautetaan korjattaviksi. Tarvittaessa jopa kaikki ensimmäisen sprintin tehtävät saetaan siirtää sellaisenaan toiseen sprinttiin, koska uusia ominaisuuksia ei aleta toteuttaa, ennen kuin vanhat on korjattu ja hyväksytty. Tämän myötä käyttöliittymätoteutuksen laatu yleensä paranee ensimmäisten sprinttien aikana merkittävästi ja päästään tilanteeseen, jossa kaikki järjestelmään tähän mennessä toteutetut piirteet toimivat niin kuin pitääkin ja ne on testattu todellisia käyttötilanteita simuloimalla. Tämä kasvattaa tiimin ammattitilaa: järjestelmää voidaan esimerkiksi demota asiakkaalle vailla huolta ns. demoeffekteistä.

Koska toteutuksen työmääräarviot tehdään vasta valmiin käyttöliittymä-ratkaisun perusteella, ne ovat muuttuneet olennaisesti aiempaa osuvammiksi. Tämä on tuonut lisää realismia ja näkyvyyttä projektin tilaan: sekä asiakkaalla että toimittajalla on parempi käsitys siitä, mitä milloinkin tulee olemaan valmiina ja missä tilassa ollaan suhteessa tavoitetilaan.

Alun kompuroinnin ja roolien hahmottamisen jälkeen projektin tehtäväjako on muuttunut kaikkien osapuolten kannalta luontevammaksi. Toteuttajien ei tarvitse ottaa kantaa käyttöliittymä-ratkaisuun, vaan he voivat keskittyä koodaamiseen ja delegoida esiin tulleet käyttöliittymä-ongelmat eteenpäin. Projektipäällikön ei tarvitse enää neuvotella asiakkaan kanssa järjestelmän sisällöstä, vaan hän voi delegoida sen käyttöliittymäsuunnittelijalle ja keskittyä projektissa ilmenneiden esteiden raivaamiseen ja aikatauluihin.

Lopuksi

Käyttötilanteiden simulointiin perustuvan käyttöliittymäsuunnittelun lisääminen ketterän projektin alkuun saattaa tuoda ketterän herätyksen saaneille toteuttajille uhkakuvia paluusta kohden vesiputousmallia. Monet heistä kuitenkin näkevät viimeistään GUIDen ja Scrumin yhdistelmäprojektiin osallistumalla, miten paljon mielekkäämmäksi projekti muuttuu, kun käyttöliittymäkuvien myötä projektin tavoitetila on selvillä jo heti koodauksen alkaessa. Hyvien käyttöliittymä-ratkaisujen toteuttamisessa on sekin hieno puoli, että käyttäjiltä voi saada aika erilaista palautetta kuin ennen: "Tämä on todella mainio softa."

Lähteet
[Agile01]
Manifesto for Agile Software Development.
<http://agilemanifesto.org>
[GUIDe04]
Laakso Sari A., Laakso Karri-Pekka, Hyvän käyttöliittymän varmistaminen GUIDe-prosessimallilla. Helsingin yliopisto, tietojenkäsittelytieteen laitos, julkaisematon artikkeli, 2004.
<http://www.cs.helsinki.fi/u/salaakso/papers/GUIDe-suomeksi.pdf>
[NCC06]
NCC:n Korjaukskalenteri-sovellus.
<http://ncc-kalenteri.ch5finland.com>
[Sytyke06]
Laakso Sari A., Käyttöliittymälähtöinen vaatimusmäärittely. Systeemyö, nro 2, 2006, s. 19-21.

Mikä mahtaa olla...



IN

Ihminen

Lumikinokset

Joulurauha

2008

SYTYKE!

OUT



Kone

Syysateet

Kiire

2007



*Hyvää Joulua ja
syttyttävää vuotta
2008!*



Irmeli Sinkkonen on Adage Oy:n kehittämisjohtaja. Irmeli on käytettävyyden pioneereja Suomessa. Hän myös kouluttaa käyttäjakeskeisen suunnittelun käyttöä verkkoprojekteissa. Ennen käytettävyytutkimukseen siirtymistään hän on toiminut neljännesvuosisadan suunnittelijana ja suunnittelun esimiehenä.

Sovelluksia verkkoon – miten käytettävyyden?

World Wide Web on kokemassa historiallista muutosta siirryttäessä toisiinsa linkitetyistä hypertekstidokumenteista selainpohjaisiin sovelluksiin. Uudet teknologiat, kuten AJAX ja avoimet sovellusrajapinnat, mahdollistavat rikkaan vuorovaikutuksen käyttäjän ja sovelluksen välillä sekä useita tietolähteitä ja toimintoja yhdistäviä sovelluksia. Käytettävyyden kannalta muutos tuo haasteita. Uudet teknologiat rikkovat Webin perinteisen vuorovaikutusmallin. Toisaalta uusi teknologia mahdollistaa käytettävyyttä tukevia ratkaisuja, joiden avulla useita perinteisiä käytettävyysongelmia voidaan välttää. Verkkosovelluksen käytettävyys voidaan varmistaa hyödyntämällä käyttäjakeskeisiä suunnittelumenetelmiä.

Sivustoja ja sovelluksia

Verkkoon rakennettavat järjestelmät ovat perinteisesti olleet sivustoja, hypertekstijärjestelmien jälkeläisiä, joiden merkitys on tiedon esittämisessä ja rakennearkkitehtuuri perustuu tuotteen informaation arkkitehtuurille. Verkkosivustojen kehittämis- ja ylläpitovastuu on tyypillisesti josain muualla kuin yrityksen tai yhteisön tietohallinnolla. Tyypillinen ja luonteva verkkosivustosta vastaava osapuoli on viestintäyksikkö. Viestintävastuulliset ovat ymmärtäneet keskimäärin erittäin hyvin käytettävyyden tärkeyden ja käytettävyytutkimukseen osallistuvista on arviolta alle 5% IT-ammattilaisia, kun yli 90% on viestintävastuullisia, kuten tuotepäälliköitä ja graafikoita.

Varsin pitkään on osana suurelle yleisöllekin tarkoitettuja verkkosivustoja ollut lyhyehköjä toimintaprosessin pätkiä, suljetuissa verkoissa on vaativampia sovelluksia. Käytettävyytutkijat ovat olleet jo huolissaan siitä, että samalla kun

monessa yrityksessä tuotteiden toimintalogiikka monimutkaistuu, niin käyttöliittymät ”puuroutuvat” ja toimintaidea on kateissa, mikä on aina merkki puutteellisesta käyttöliittymän määrittelytyöstä.

Uusien teknologioiden ansiosta verkkosovellukset saadaan toimimaan lähes yhtä luontevasti kuin työpöytäsovellukset. Käytössä on esimerkiksi sivun osien dynaaminen päivitys, heti tehtävät syötteiden tarkistukset ja objektien raahaus. Verkkoympäristö taas antaa sovellukselle mahdollisuuksia, joita työpöytäsovelluksissa ei ole ollut. Ongelmatilanteisiin voidaan rakentaa parempi tuki, tuotteeseen monipuolisempi ja värikkäämpi käyttöliittymä. Käytettävissä on paremmat tiedon visualisointimahdollisuudet, paremmat yhteydet sovelluksesta toiseen ja monipuolisemmat yhteystavat käyttäjien välillä. Käyttöliittymistä voidaan myös tehdä helposti versiot mobiililaitteisiin ja toteuttaa erikieliset versiot.

Vaikka uusissa verkkosovelluksissa on kyse samasta asiasta kuin missä tahansa sovelluksessa, kuten tiedon säilyttämisestä, syötöstä, tulostamisesta ja päivityksistä sekä näyttämisestä, niin uudet sovellukset ovat entistä rikkaampia, monipuolisempia ja monimutkaisempia. Tämä tarkoittaa sitä, että jos käyttäjä ja käyttäjän tarpeet jätetään sivurooliin tuotekehityksessä, lopputulos voi olla lähes käyttökelvoton.

Uusea yritys mieltii paraikaa vanhojen työpöytäsovellusten siirtämistä verkkoon esimerkiksi vanhojen sovellusten huonon käytettävyyden takia. Käyttöliittymäsuunnitteluun uudet tekniikat antavat mahdollisuuksia, joita ei ole ennen ollut sivustoissa eikä työpöytäsovelluksissa. Vastassa on nyt kaksinkertainen haaste: miten saada käyt-



Janne Tompuri on Adage Oy:n toimitusjohtaja. Hänellä on yli kymmenen vuoden työkokemus verkkosivustojen ja selainpohjaisten sovellusten suunnittelusta ja projektijohtamisesta.

tettävyys kuntoon samalla kun sovellukset ja vuorovaikutustavat monipuolistuvat. Suunnittelijalta vaaditaan tämän kaltaisissa projekteissa paljon: asiantuntemusta niin informaatioarkkitehtuurin, käyttöliittymän kuin myös vuorovaikutuksen suunnittelussa.

Prosessi

Tuotekehitysprosessit eivät ole vakiintuneet vuosien saatossa. Viimeisen 40 vuoden kuluessa on kehitetty lukuisia menetelmiä kehittämistyön vakaannuttamiseksi ja teollistamiseksi. Useiden tuotekehitysmenetelmien ongelmat johtuvat puuttuvista lähtötiedoista: ei olla selvillä siitä, miksi tuotetta kehitetään; miksi joku käyttää tai ei käytä tuotetta; millainen tuote sopii tiettyyn työhön tai toimintatapaan. Saati ketkä tuotetta käyttävät tai keiden halutaan sitä käyttävän. Aikataulut eivät pidä, koska kukaan ei tarkkaan tiedä, mitä ja millaisia toimintoja tuotteeseen tarvitaan. Tuotteen toimittaja ei tiedä, paljonko kehitystyöhön kuluu aikaa, koska hänellä ei ole tarkkaa käsitystä tuotteen sisällöstä ja monimutkaisuudesta.

Ketterissä tuotekehitysmenetelmissä on nostettu kädet ylös ja luovuttu ajatuksesta, että tilaaja pysyisi tarkasti tietämään, mitä oikeasti haluaa, jolloin toimittaja vastaavasti tietää vielä vähemmän, mitä on myymässä. Menetelmä perustuu tekemiseen ja tehdyn työn toistuvaan korjaamiseen. Kyse on siis tuotteen rakentamisesta iteroimalla, mutta melko raskaalla ja kalliilla tavalla. Toisaalta ketterien menetelmien avulla hallitaan kehitystyön aikana muuttuvia vaatimuksia ja mahdollisesti nopeutetaan tuotteen valmistumista, mutta mitä vähemmän näitä muutoksia tulee toimintatapaan, sitä nopeampi prosessi on. Pahimmillaan ketterät menetelmät hidastavat tuotekehitystä.

Käyttäjakeskeisessä suunnittelussa ajetaan takaa samaa peikkaa, mutta menetelmä on osittain sama, osittain toinen. Tuote rakennetaan iteroimalla aivan kuten ketterissä menetelmissä, mutta iterointi on huomattavasti kevyempää ja tiheämpää sekä luotettavammin tuotetta eteenpäin vievää. Siinä iteroidaan mahdollisimman kevyin välinein käyttöliittymäprototyyppiä eli verkotuohteen käyttäjälle näkyvää osuutta. Tämä tarkoittaa sekä tuotteen toimintatapaan, rakennetta, terminologiaa että ikkunamalleja. Iterointikierron osana evaluoidaan ratkaisut käytettävyyksien tai asiantuntija-arvioin. Tällöin jatkotyö ei ole pelkästään tilaajan edustajan osaamisen varassa, vaan perustuu tietoon ja tutkimukseen.

Toinen tavallinen sovellusten kehittämistapa on muokattu vesiputousmalli, jossa tuote tehdään vaihe vaiheelta lähtien liiketoiminnallisista tarpeista. Vesiputousmalli on luonteva tapa edetä tuotekehityksessä. Kannattaa kuitenkin tiedostaa, että tämä edelleen mukana sinnittelevä prosessimalli on kehitetty aikana, jolloin käyttäjän panos

eli syötteet saatiin mukaan määrämuotoisten reikäkorttien avulla ja operaattorit pyörittivät heksalukuja päässään ja korjasivat ongelmatilanteissa suoraan koneen muistissa olevaa koodia. Malli vaatii käyttöliittymäsuunnittelun osalta suuria muutoksia toimiakseen.

Oli tuotekehitysmenetelmä mikä tahansa, tuotekehitysprosessi kannattaa käytännössä jakaa kahdeksi projektiksi. Ensimmäisessä määritellään ja toteutetaan käyttöliittymäprototyyppi. Toisessa projektissa toteutetaan tuote. Tuote siis ensin konseptoidaan ja varsinainen toteutus tehdään vasta kun kaikilla on selvä näkemys siitä, millainen tuotteesta tehdään. Toteutustyön kustannukset ja kesto kannattaa laskea vasta kun ensimmäinen projekti eli tuotteen käyttöliittymä on valmis, ja kun sekä tilaaja että toimittaja tietävät mistä kauppaa ollaan tekemässä.

Määrittelymenetelmät

Käyttäjakeskeisen sovelluskehityksen menetelmiä on useita. Menetelmät ovat perusrakenteeltaan hyvin samanlaisia, käsittäen seuraavat vaiheet.

1. Käyttäjätiedon keruu
2. Tiedon analysointi, mallintaminen ja tiivistäminen
3. Työn uudelleen suunnittelu
4. Prototyyppien teko ja testaus
 - a. rakennesuunnittelu
 - b. sivusuunnittelu
 - c. vuorovaikutussuunnittelu
5. Visuaalisen suunnittelun yhdistäminen
6. Käytettävyyksien testit
7. Käyttöliittymäohjeistojen ym. tuottaminen

Tunnetuimmat menetelmät eli *Contextual Design* (CD) ja *Scenario-Based Design* (SBD) eroavat lähinnä siinä, että CD muistuttaa analyysimenetelmässään perinteellisempiä määrittelymenetelmiä, kun taas SBD nojaa tekstimuotoisiin skenaarioihin ja niiden vaiheittaiseen kehittämiseen.

On tärkeää, että käyttöliittymän määrittelyn osana selvitetään liiketoiminnasta vastaavien kanssa, ketkä tulevat käyttämään tuotetta ja mitä heidän halutaan sillä tekevän. Seuraavaksi selvitetään millaisia käyttäjät ovat, miten uusi tuote voisi liittyä heidän toimintatapoihinsa, millainen on heidän toimintatapansa nyt ja miten sitä voidaan parantaa, millaisia ongelmia heillä on nyt työssään sekä millaisessa ympäristössä he toimivat ja haluavat toimia. Tämän vaiheen tärkeimmät menetelmät ovat haastattelut, havainnoinnit ja ryhmäkeskustelut.

Selvitykset analysoidaan, päällekkäisyydet puretaan ja asiat ryhmitellään samankaltaisuusanalyysillä, ja merkityksellisimmistä asioista

tehdään tarpeellisia lisäkuvauksia. Contextual Design -menetelmässä käytetään tähän flow-mallia, sekvenssimallia, fyysistä ympäristömallia, kulttuurimallia ja artefaktimallia, SBD:ssä käytetään eritasoisia täydentäviä skenaariokuvauksia (ongelmaskenaariot → aktiviteettiskenaariot → informaatio- ja vuorovaikutusskenaariot). Näitä malleja ja kuvauksia voi täydentää perinteisillä malleilla, mutta on huomattava, että nyt kuvataan käyttäjiä ja käyttäjien tekemisiä, ei rooleja.

Lopuksi

Käyttäjakeskeinen suunnittelu puree sekä tuotteen käytettävyyden parantamiseen, että tuotekehityksen suurimpien ongelmien karsimiseen. Tämä on johdonmukaista, koska kehitysprosessin viivästyksiset ja etenkin yllätykset ovat seurausta huonosti tehdystä pohjatyöstä. Jos halutaan käyttää ketteriä menetelmiä, on hänenkin, joka vastaa asiakkaan puolella tuotteen kehittämisestä, tehtävä nämä selvitykset ja tutkimukset – kunnolla.

Käytettävyyden huomioonottamista tuotekehityksessä ”syytetään” siitä, että projektit venyvät entistä pitemmiksi. Käyttäjakeskeisessä tuotekehityksessä käytetään kieltämättä enemmän aikaa määrittelyvaiheeseen kuin ei-käyttäjakeskeisissä. Toisaalta hyvin suunniteltu on jo melkein tehty. Vaihtoehtona on, että osa asioista vain jätetään kokonaan selvittämättä ja suunnittelematta.

Käytännössä projekteihin menee usein jopa vähemmän aikaa, kun tiedetään mitä ollaan tekemässä eikä tehdä liikaa tai liian vähän piirteitä tuotteeseen. Pois jää myös kalliita ja hyödyttömiä väittelyitä siitä, tarvitaanko jotain ominaisuutta vai ei, ja millainen sen olisi oltava. Samoin vähenee toimittajan resurssien hukkakäyttö ja turhat muutokset, kun tuotteen toiminta on valmiiksi mietitty. Suunnittelijan ja toteuttajan työn ikävimpiä tilanteita ovat tapaukset, joissa asiakkaan eri edustajat tai osastot antavat keskenään ristiriitaisia tai puutteellisia ohjeita suunnittelijalla ja toteuttajalle.

Käyttöliittymissä monipuoliset mahdollisuudet tarkoittavat käytännössä myös monipuolisia ongelmia. Mitä paremmat mahdollisuudet on rakentaa käyttäjälle hieno käyttäjäkokemus, sitä suuremmat mahdollisuudet on myös epäonnistua pahasti. Esimerkkinä erilaisten tehosteiden, korostusten ja huomiota kiinnittävien animointien ylitsepursuava hyödyntäminen verkkopalvelussa. Nämä tekevät kokonaisuuden sekavaksi ja vievät suunnitteluenergian helposti toisarvoiseen. Uusien tekniikoiden antaessa yhä monipuolisemmin uusia hyödyllisiä mahdollisuuksia alkaa käyttöliittymäsuunnittelu olla yhä enemmän erikoistuneiden suunnittelutiimien työtä.

Mikäli tuotteiden helppokäyttöisyydestä ei huolehdi, käyttäjien eriarvoisuus lisääntyy: kokeneet käyttäjät pystyvät hyödyntämään uusia ominaisuuksia entistä paremmin ja kokemattomilla voi olla entistä enemmän hankaluuksia.

Käyttäjakeskeisiin menetelmiin voidaan yrityksissä siirtyä vaiheittain ja ne voivat täydentää niin vesiputousmallia kuin ketteriä menetelmiä. Oleellista on kuitenkin tehdä käyttöliittymäsuunnittelu loppuun asti huolellisesti, jolloin projektista tulee ennustettavampi sekä lopputulokseltaan, kustannuksiltaan että aikataulultaan.

Lisätietoja artikkelissa viitatuista menetelmistä

Beyer, H., Holtzblatt, K.: Contextual Design: A Customer-Centered Approach to Systems Designs, John Wiley & Sons, New York 1996; Holtzblatt, K., Wendell, J. Wood, S.: Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design, Elsevier Science & Technology 2004

Carroll, J. Scenario-Based Design: Envisioning Work and Technology in System Development, John Wiley & Sons, 1995; Rosson, M.B.,

Carroll, J.: Usability Engineering, Academic Press 2002

CDstä löytyy napakka yhteenveto esimerkiksi http://en.wikipedia.org/wiki/Contextual_design#Work_modeling

Käyttäjien muuttuva rooli suunnittelussa



Teksti: Pirjo Näkki

Perinteiset käyttäjätutkimuksen menetelmät ja työkalut eivät välttämättä toimi Internet-maailmassa, jossa tuotteet ja palvelut halutaan julki nopealla tahdilla. Viimeiseen asti hiottujen tuotteiden sijaan Web 2.0 -sovellukset avataan usein beta-versioina ja annetaan innokkaiden käyttäjien testattavaksi ennen virallista julkistusta. Käyttäjät ovat siten mukana tuotekehityksessä, mutta usein varsin eri tavalla kuin käytettävyyden oppikirjoissa opastetaan. Internet-ympäristö haastaa käyttäjätutkimuksen uusiutumaan mutta tarjoaa myös uusia mahdollisuuksia vahvistaa ja muuttaa käyttäjien roolia pelkistä tutkimuskohteista aktiivisiksi suunnitteluosapuoliksi.

Perinteinen käyttäjäkeskeinen suunnittelu lähtee järjestelmän vaatimusten, käyttökontekstin ja käyttäjän sekä organisaation tarpeiden kartoituksesta. Vasta huolellisen taustaselvityksen jälkeen tuotetaan suunnitteluratkaisuja, joita myöhemmin arvioidaan yhdessä käyttäjien kanssa. Systemaattinen lähestymistapa on ollut tärkeää erityisesti suurten ja monimutkaisten järjestelmien toteuttamisessa, joissa alkuvaiheen käyttäjätutkimus tulee halvemmaksi kuin virheiden korjaaminen jälkikäteen. Olisi resurssien tuhlausta toteuttaa kalliita järjestelmiä, joiden valmistuessa huomataan, ettei niille ole käyttöä.

Uusille Internet-palveluille on puolestaan tyypillistä yksinkertainen ja rajattu toiminnallisuus. Monet Web 2.0 -palvelut rakentuvat yhden ominaisuuden ympärille tai ovat yhdistelmiä eri sovellusten toiminnoista (mash-ups). Toimiva palvelu voidaan tehdä lomapuhteena kaveriporukalla ilman suuria kehityssponsisteluja, eikä mittavaa käyttäjätutkimustakaan siten koeta tarpeelliseksi.

Tarvitaanko käyttäjätutkimusta enää?

Käyttäjä- ja tarvekartoitusten sijaan sosiaalisen median palveluiden alkusysäyksenä on usein kehittäjien oma tarve tietylle palvelulle. Jos käytännöllistä elokuva-arvosteluohjelmaa ei löydy valmiina, se koodataan itse. Käytettävyydestä puolestaan ajatellaan hoituvan itsestään, kun keskeneräinen sovellus laitetaan verkkoon kaikkien vapaasti kokeiltavaksi ja kommentoitavaksi. Beta-testaajat voidaan myös rekrytoida verkon välityksellä esimerkiksi blogeja ja verkostopalveluja hyödyntäen.

Pienelle kohderyhmälle suunnattuja palveluja voidaan suunnitella kevyesti ja toteuttaa ilman laajamittaisia käyttäjätestejä – varsinkin, jos kehittäjät kuuluvat itse kohderyhmään. Vaikka sovelluskehitys ei nykyään vaadi välttämättä ohjelmointia, on kuitenkin turha olettaa kaikkien ihmisten voivan tai haluavan itse kehittää tarvitsemansa palvelut. Suuren yleisön toiveiden ja tarpeiden selvittämiseen tarvitaankin edelleen käyttäjätutkimusta.

Verkon haasteet ja mahdollisuudet

Sosiaalisen median sovelluksia kehitettäessä luotetaan helposti yhteisölliseen prosessiin ja tuotteiden hioutumiseen ajan myötä. Todellisuudessa puolihuolimattomasti toteutettu ja kerran julkaistu toiminnallisuus voi jäädä ohjelmaan pysyväksi piirteeksi, kun käyttäjien oletetaan jo oppineen käyttämään siitä. Käytettävyyttä ikuisessa beta-vaiheessa olevat tuotteet eivät siten edistä. Monille vanhoille käytettävyysopeille olisi kuitenkin käyttöä myös Web 2.0 -palveluja kehitettäessä.

Vaikka käyttäjäpalautteen keruu Internetin kautta on näennäisesti helppoa, siihen liit-

Pirjo Näkki on informaatioverkostojen diplomi-insinööri ja toimii VTT:llä sosiaalisen ja semanttisen median tutkijana erityisalanaan käyttäjäkeskeinen tuotekehitys.

tyy joukko haasteita. Vapaamuotoista palautetta kertyy helposti paljon, ja ison tekstiaineiston analysointi on työlästä. Vastausten tai kohderyhmän kattavuudesta ei ole takeita, ja kasvoton vastaaaminen voi innostaa myös häiriköintiin.

Sosiaalinen media haastaa käyttäjätutkimuksen soveltamaan vanhoja menetelmiä uuteen ympäristöön. Toisaalta se tarjoaa myös joukon uusia menetelmiä, jotka voivat parantaa käyttäjätutkimuksen joustavuutta ja tehokkuutta. Haastatteluja ja ryhmäkeskusteluja voidaan järjestää verkon välityksellä, jolloin on helpompi tavoittaa myös maantieteellisesti hajallaan olevia käyttäjiä. Kiireisten käyttäjien osallistumiskynnystä madallaa se, että verkkokeskusteluihin voi osallistua omien aikataulujen mukaan.

Owela yhdistää kehittäjät ja käyttäjät

VTT:llä on käynnissä sosiaalisen median SOMED-hanke, jossa tutkitaan sosiaalista mediaa ilmiönä ja kehitetään teknologiaa sen tarpeisiin. Lisäksi kehitetään uusia tapoja hyödyntää Internetiä käyttäjakeskeisessä suunnittelussa. Projektin puitteissa kehitetty Owela (Open Web Lab, <http://owela.vtt.fi>) on avoimen innovaation verkkolabo-

ratorio, jossa käyttäjät, kehittäjät ja tutkijat voivat yhdessä ideoida, kehittää ja testata tulevaisuuden tuotteita ja palveluja. Owela toimii samalla uusien tutkimusmenetelmien testausalustana.

Owela tarjoaa työkaluja innovaatioprosessin eri vaiheisiin. Ideatuubissa käyttäjät voivat arvioida ja kommentoida erilaisia tilannekuvauksia, ideoita ja skenaarioita. Testilabrassa kuka tahansa pääsee kokeilemaan kehitteillä olevia web- ja mobiilisovelluksia ja antamaan niistä palautetta. Käyttäjät voivat kertoa ideoitaan ja antaa palautetta myös chatin kautta, jossa tutkijat päivystävät tiettyinä kellonaikoina. Lisäksi käytössä on verkkokyselyjä ja tutkimusblogeja, joiden kautta tutkijat ja koekäyttäjät voivat kommunikoida testijakson aikana.

Myös Owelaa itsessään kehitetään jatkuvasti käyttäjien ja tutkijoiden kokemusten pohjalta. Tulevaisuudessa mukaan tulee muun muassa mobiiliominaisuuksia, joiden avulla käyttäjätutkimusta voidaan tehdä paikasta riippumatta siellä, missä käyttäjät ovat. Avointen ideointi- ja testausosioiden lisäksi Owelaan voidaan räätälöidä myös rajattuja tutkimusosioita yritysten käyttöön.

Toisiaan täydentävät menetelmät

Verkossa tapahtuva käyttäjätutkimus ei voi kokonaan korvata perinteistä haastattelua, havainnointia ja testausta, vaan niillä on edelleen paikkansa tietyissä tuotekehityksen vaiheissa. Uudenlaiset verkkotyökalut voidaankin nähdä kasvokkain tapahtuvien tutkimusmenetelmien täydentäjinä.

Verkkokeskusteluihin ja blogikommentointiin tottuneille käyttäjille on luontevaa osallistua myös avoimeen käyttäjätutkimukseen. Kehittäjien kynnys osallistua keskusteluun voikin olla suurempi kuin käyttäjien. Käyttäjien kannalta on kuitenkin motivoivaa, jos myös yritysten edustajat ottavat osaa keskusteluun ja osoittavat, miten käyttäjien mielipiteet on huomioitu kehitystyössä.

Verkossa avoimesti tehtävä tutkimus luo haasteita tutkimuksen luottamuksellisuudelle mutta tarjoaa toisaalta yksittäishaastatteluja tai suljettuja kyselyjä hedelmällisemmän maaperän uusille ideoille. Perinteisen käyttäjätutkimuksen lisäksi käyttäjät voidaan ottaa aktiivisemmin mukaan tuotteiden suunnitteluun, johon myös sosiaalisen median työkalut tarjoavat hyviä mahdollisuuksia.

Sisältö: Owela on käyttäjakeskeisen tuotekehityksen avoin verkkolaboratorio.

Käytettävyyden ajankohtaiset haasteet hyvinvointiteknologiassa

Hyvinvointiteknologian kehitykseen vaikuttaa sosiaali- ja terveyspalvelujen tuottamiseen liittyvät muutostekijät, taloudelliset paineet sekä teknologian kehitys. Nämä tekijät yhdessä luovat haasteita alan tuotekehitykselle, liiketoiminnalle sekä käyttäjäyhteistyölle osana näitä prosesseja. Nopea kehitys sekä kiristynyt kilpailu saattavat vaikuttaa olennaisesti siihen miten käytettävyyssuunnittelua toteutetaan yrityk-

terveydenhuollon teknologian tuotekehitys on monimutkainen prosessi, jossa toimittajan on huolehdittava säädösten ja lakien noudattamisesta sekä standardien hyödyntämisestä. Koska tuotekehitysprosessi vaatii useiden tekijöiden huomioon ottamista, on aika ideasta markkinoille usein pitempi. Valmistajalta edellytetään tiivistä yhteistyötä kliinisen käyttöympäristön sekä valvojen viranomaisten kanssa. Nämä tekijät lisäävät tuotekehityksen kustannuksia ja tuovat haasteita myös käyttäjäkeskeisen suunnittelun kehittämiseen hyvinvointiteknologiassa.

Hyvinvointiteknologian liiketoiminta-alueet ovat muutoksessa sosiaali- ja terveyspalveluiden kehittyessä. Teknologinen kehitys ja taloudelliset paineet siirtävät lääketieteellisiä toimintoja sairaaloista koteihin sekä pyrkivät edistämään kansalaisten valtauttamista omasta terveydestään ja sairauden hallinnasta Hyvinvointiteknologiaa käytetään yhä kasvavassa määrin ei-ammattilaisen toimesta ja vakioimattomissa käyttöympäristöissä. Yritykset ovat joutuneet uudistamaan liiketoimintamallejaan kytetäkseen tarjoamaan terveydenhuolto-organisaatioille näiden muutosprosesseissaan tarvitsemia ratkaisuja (kuva 1).

Terveydenhuollon muutosprosessit luovat yhteiskehittämisen tarpeen

Terveydenhuollon innovaatioita on yleensä tarkasteltu teknologialähtöisesti.

Tarkastelunäkökulma on muuttumassa siten, että terveyspalvelujen ja teknologian innovointi

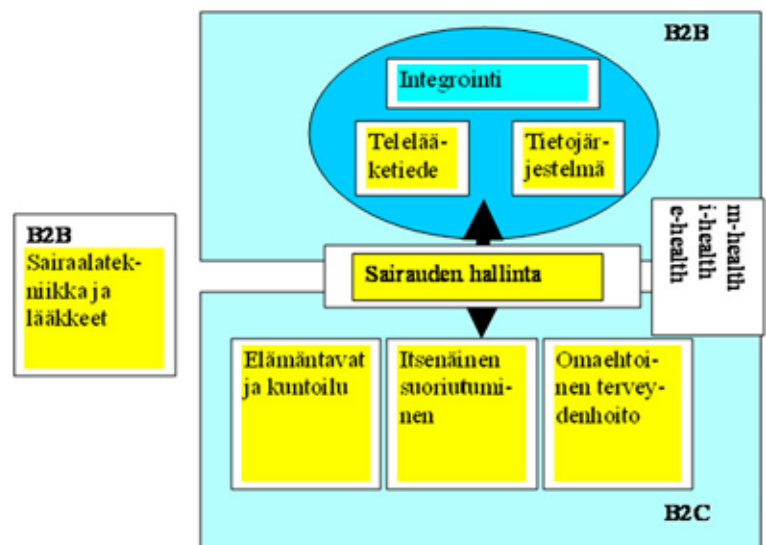
sessä. Käytettävyyttä on tutkittu lääketieteen tekniikan sekä hyvinvointi-teknologian kentässä yllättävän vähän. Maailmanlaajuisen käytettävyysspäivän (World Usability Day) teemana oli tänä vuonna juuri terveydenhuollon tuotteet. Hyvinvointiteknologian käytettävyys koskettaa meitä jokaista, olipa kyse sitten kuluttajatuotteesta tai terveydenhuollon ammattilaisten käyttöön tarkoitettua tuotteesta.

nähdään systeemisenä kokonaisuutena, tavoitteena innovaatioiden yhteiskunnallisen laadun kehittäminen. Eri toimijoiden välisen vuoropuhelun merkitys korostuu, kun huomataan, että palveluiden ja teknologioiden kehittäminen on riippuvuussuhteessa toistensa kanssa (kuva 2).

Terveydenhuollon innovaatiotutkimusta koskehtavan kirjallisuuden perusteella vaikuttaa kuitenkin siltä, että vuoropuhelu on usein vaikeaa terveydenhuollon toimijoiden sekä yritysten välisessä yhteistyössä. Käyttäjäyhteistyö ja käyttö-

Kirjoittaja on toiminut lääketieteen tekniikan alan tutkimus-, kehittämis- ja koulutustehtävissä 12 vuotta. Parhailaan hän toimii erityisasiantuntijana ja työryhmän koordinaattorina Oulun teknologiaterveyskeskus hankeosiossa sekä käytettävyysyksikön vetäjänä Oulun ammattikorkeakoulun tekniikan yksikössä sijaitsevassa Hyvinvointi-teknologian tutkimus- ja tuotekehityssuhteissa.

Kuva 1. Hyvinvointiteknologian laajentuneet sovellusalueet (Sarvanummi 2001, 18)



toiminnan tuntemisen merkitys ovat innovaation onnistumisen ja teknologiasta saatavan yhteiskunnallisen hyödyn merkittäviä osatekijöitä terveydenhuollossa.

Yritykset toimivat usein varsin teknologiakeskeisesti ja omaavat puutteellisen tietämyksen kliinisen käyttötoiminnan vaatimuksista. Käyttäjän voi olla mahdotonta esittää uudelle teknologialle etukäteen vaatimuksia, jos aikaisempaa kokemusta vastaavasta teknologiasta ei ole. Usein vasta käyttöönoton aikana tuotteen käytettävyyden ja siitä saatavat hyödyt tulevat esiin ja arvioinnin kohteeksi.

Hyvinvointiteknologian käytettävyys luo myös turvallisuutta

Lääkintälaitteiden ja terveydenhuollon tietojärjestelmien käytön aikaiset ongelmat ovat suurenuslasin alla niiden aiheuttamien riskitilanteiden johdosta. Inhimillisiä tekijöitä on tutkittu verrattavan vähän lääketieteellisten laitteiden ja järjestelmien osalta. Käyttövirheet, jotka aiheutuvat puutteista järjestelmien käytettävyydessä, ovat nousseet keskeisiksi tarkastelukohteiksi. Usein tutkimuksissa on korostettu käyttökoulutuksen merkitystä sen sijasta että painotettaisiin ihmisen-kone vuorovaikutuksen suunnitteluun liittyvän osaamisen hyödyntämistä. Käyttöliittymän laadun puutteista johtuen kärsii paitsi käytettävyys, samalla usein myös turvallisuus.

Sähköisiä lääkelaitteita koskevan standardisarjan osassa 60601-1-6 kuvataanriskianalyysi käytettävyyden painopisteen suunnassa, käytettävyyssuunnitteluun integroituna (SFS-EN 60601-1-6, 2004).. Lääketieteellinen laite, kun sitä käytetään käyttötarkoituksen mukaisella tavalla, tulee saavuttaa riittävä käytettävyyden taso siten, että riskit jotka seuraavat normaalista käytöstä sekä käyttövirheistä, ovat hyväksyttäviä.

Liljegen (2004) toi esille lääketieteen tekniikan käytettävyyttä koskeavassa väitöskirjassaan itsellenikin vuosien saatossa yhä tärkeämmäksi nousseen ymmärryksen siitä, että paitsi että käytettävyyssuunnittelun menetelmiä pyritään hyödyntämään valmistajan kannalta tuotesuunnittelussa, on tärkeää myös lisätä asiakkaiden tai ostajien käytettävyydestä ja -ymmärrystä. Käytettävyys on nostettava tärkeäksi kriteeriksi ostopäätöstä tehtäessä. Asiakkaiden kriittisyyden kohottaminen on tärkeää siksi, että tuotteen käytettävyydellä on usein merkittävä rooli turvallisuustekijöiden kannalta – erityisesti turvallisuus-kriittisissä toimintaympäristöissä.

Käytettävyyssuunnittelu vaatii vielä kehittämistä ja tukea yrityksissä

Hyvinvointiteknologian yritykset ovat tyypillisesti pieniä ja menestymisen kriittisiä avaintekijöitä ovat usein tehokas tuotteistus ja

kansainvälistyminen. Käytettävyys-suunnittelu on pk -yrityksissä tyypillisesti vielä varsin vähäistä.

Oulun yliopiston teknillisen tiedekunnan työtehteen laboratorio toteutti kyselyn ICT -alan suunnittelijoille ja tutkijoille alkuvuonna 2000. Kyselyn tavoitteena oli selvittää Oulun seudun käytettävyystekniikan T&K -toimintaa. Käytettävyys katsottiin hyvin yksimielisesti kuuluvan olennaisesti tuotekehitykseen ja sen mahdollisimman aikaista huomioonottamista tuotekehityksessä arvostettiin. Kehittämisehdotusten osalta esille tuli esille mm. että käytettävyysasioiden muotoilu vaatimuslistaksi on vaikeaa. Todettiin myös, että käytettävyyssuunnittelu on mennyt eteenpäin, mutta systemaattisuutta menettelyyn olisi saatava lisää. Tärkeimmäksi menetelmäksi koettiin käytettävyydesti, jossa käyttötilannetta havainnoidaan ja mitataan objektiivisesti (Värynen et al, 2001).

Oulun aluetyöterveyslaitokselle toteutetussa opinnäytetyössä kartoitettiin Oulun alueen pk -yritysten käytettävyyssosaamista ja tarvetta osana ESR rahoitteista -hanketta. Käytettävyyssosaamista kartoittava kysely toimitettiin Oulun seudun yritysten johtohenkilöstölle tai tuotekehityksestä vastaavalle henkilölle. Eniten tukea ja käytettävyyssuunnitteluvaiheen konseptien ja prototyyppien käytettävyyden asiantuntija-arvioinnissa ja pienimuotoisessa testauksessa, joihin ei selvityksen mukaan suuresti panostettu. Myös käyttötilanteiden selvittämisessä ja käyttäjryhmien tunnistamisessa yritykset tuntuivat kaipaavan apua. Selvityksen perusteella vaikutti siltä, että käytettävyys menetelmiksi mielletään varsin perinteisesti juuri käytettävyydestit todellisilla loppukäyttäjillä (Törmänen, 2004).

Hyvinvointiteknologian tutkimus- ja tuotekehityskeskus toteutti Oulun seudun hyvinvointiteknologian yrityksille vuonna 2006 kyselyn, jonka tavoitteena oli selvittää käytettävyyssuunnittelun tilaa alan yrityksissä. Selvitys toteutettiin insinöörityönä allekirjoittaneen ohjauksessa. Kuusi kyselyyn osallistuneista yrityksistä toimii pelkästään ohjelmistoalalla ja kahdeksan yritystä valmistaa myös laitteita. Hyvinvointiteknologian tai lääketieteen tekniikan alalla toimii kahdeksan kyselyyn osallistuneista yrityksistä. Muut yritykset kehittävät hyvinvointiteknologian tai lääketieteen tekniikan tuotteita tai tarjoavat aloille palveluita muiden toimialojensa ohella. (Turunen, 2006).

Vastaajat olivat yhtä mieltä siitä, että hyvä käytettävyys takaa laadukkaamman ja kilpailukykyisemmän tuotteen. Samalla käyttäjät ovat tyytyväisempiä ja heidän työskentelynsä on tehokkaampaa. Sen sijaan tuotekehitykseen kuluvaan aikaan ja kustannuksiin ei käytettävyydellä ole kyselyn mukaan vaikutusta (kuva 3) (Turunen, 2006).

Yrityksiltä tiedusteltiin myös heidän systemaattisesti käyttämiään käytettävyyden toimenpiteitä. Käytettävyydestä, käyttäjäpalautteen keruu sekä käyttöohjeen tarkastaminen tuntui hiukan korostuvan näissä vastauksissa. Yritykset hakivat ulkopuolista tukea eniten käytettävyyden asiantuntija-arviointiin ja tuotteen käytettävyydestä tulkintaan. (Turunen, 2006).

Tutkimuksen mukaan hyvinvointitekniikan tai lääketieteen tekniikan alalla toimivat yritykset olivat pääpiirteittäin tehneet tuotteiden hyvän käytettävyyden eteen jonkin verran enemmän toimenpiteitä verrattuna muihin alalla mukana oleviin yrityksiin. Samaten tuloksista ilmeni, että käytettävyyssuunnittelussa laitevalmistajat ovat pääpiirteittäin hieman ohjelmistoyrityksiä aktiivisempia.

Yhteistyöllä käytettävyyssuunnittelun edistämiseen

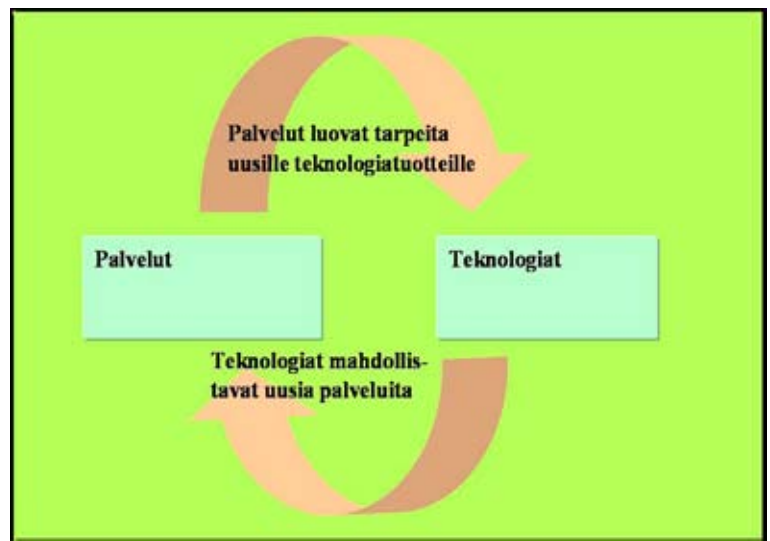
Oulun seudulla on viime vuosien aikana tapahtunut asteittaista verkottumista käytettävyyssalalla. OAMK Tekniikan yksikön hyvinvointitekniikan tutkimus- ja tuotekehityskeskukseen (HYTKE) on kahdessa vuodessa rakennettu modernia teknologiaa edustava käytettävyyssuunnitteluohjelma ja laitteistoinen (www.oamk.fi/hytke). Käytettävyyssuunnitteluohjelma tarjoaa yritysten ja muiden yhteisöjen käytettävyyssuunnittelijoille tukea asiantuntija-arviointiin sekä kehitys- ja hyväksymisvaiheen käytettävyyssuunnitteluihin.

Hyvinvointitekniikan tutkimus- ja tuotekehityskeskukseen toimesta järjestettiin vuonna 2005 aivoriivi, jonka tavoitteena oli kartoittaa tilannetta käytettävyyssuunnittelun, yritys yhteistyön sekä käytettävyyssuunnittelun kentässä Oulun seudulla. Oulun yliopiston, työterveyslaitoksen, yritysten sekä Oulun ammattikorkeakoulun kesken järjestetyn tilaisuuden lopputulemana oli havainto, että alueelliselle verkottumiselle sekä osaamis- ja kehittämiselle olisi tilaus olemassa. Verkottumisen kautta voidaan löytää uutta näkökulmaa tutkimus-, kehittämis- ja opetustyöhön sekä edistää käytettävyyteen liittyvien tilojen ja laitteistojen yhteiskäyttöisyyttä.

Vuonna 2005 eri puolilla maailmaa järjestetty maailmanlaajuinen ensimmäinen käytettävyyden -aiheinen WUD -teemapäivä (www.worldusabilityday.org) toteutettiin myös Oulussa työryhmän toimesta. Päivä toteutui Oulun yliopiston koordinoimana ja teemana 3.11.2005 oli: 'Miten teknologiasta ihmisläheisempää - Käytettävyyssuunnittelu osana tuote- ja ohjelmistokehitystä'. Vastaava tapahtuma toteutettiin 14.11.2006 HYTKE -hankkeen koordinoimana. Seminaarin teemana oli: 'Käytettävyyssuunnittelun arki - menestystekijät ja pullonkaulat'. Kumpaankin seminaariin osallistui n. 180 henkilöä, mikä kertoo osaltaan kasvaneesta kiinnostuksesta käytettävyyden kohtaan.

Käytettävyyssuunnittelun tehokkaille integroimiselle tuotekehitykseen on selkeä tarve olemassa ja tuotteeseen liittyvät riskit ja tuotteen käyttöliittymän laatu ovat merkittäviä ajankohtaisia tekijöitä hyvinvointitekniikassa. Ei siis ollenkaan ihme että Usability Professionals Association (UPA) päätti nostaa vuoden 2007 maailmanlaajuisesti teemaksi juuri terveydenhuollon tietojärjestelmät.

Kuva 2. Palvelujen ja teknologian kehittämisen vuorovaikutus (Saranummi et al, 2005, 2)



Lähteet:

- Saranummi, N. (2001). Hyvinvointi- ja terveysalan teknologia- ja palvelutuotteet. Tekniikkatiedustelu 103/2001. TEKES
- Saranummi, N.; Kivisaari, S., Väyrynen, E & Hyppö, H. (2005). Terveystieteiden uudistaminen. Systematit innovaatiot ja asiantuntijapalvelut muutoksen ajureina. Tekniikkatiedustelu 180/2005. TEKES
- SFS-EN 60601 1-6 (2004). Medical electrical equipment part 1-6: General requirements for safety, Collateral standard: Usability. Helsinki: Suomen Standardoimisyhdistys SFS ry.
- Turunen, N. (2006). Käytettävyyssuunnittelun tila Oulun seudun hyvinvointitekniikan yrityksissä. Insinööri. Oulun Ammattikorkeakoulu.
- Törmänen, M. (2004). Käytettävyyden osaaminen PK -yrityksissä. Diplomityö. Oulun yliopisto.
- Väyrynen, S.; Kalliojärvi, T.; Törmänen, M.; Autio, T. & Backman, K. (2001). Käytettävyystekniikan kärkeasiat: luonnostelu kirjallisuustutkimuksen ja empirian pohjalta. Työtieteen laboratorion hankeraportteja No:12. Oulun yliopisto.



Antti Martikainen on käytettävyyttä ja sovelluskehitysmenetelmiä päivittäisessä työssään pohtiva Realizer by SYSOPENDIGIA -tuotteen tuotepäällikkö. Realizer on mallinnuspohjainen suunnittelu- ja toteutusympäristö selainpohjaisten sovellusten rakentamiseksi.

Pätevillä protoiluvälineillä potkua vaatimusmäärittelyyn

Oikeiden vaatimusten löytäminen IT-hankkeen budjetoinnin ja aikataulutuksen pohjaksi on vaikeaa. Väärät vaatimukset puolestaan johtavat usein vakaviin laatuongelmiin erityisesti käyttäjien tarpeiden näkökulmasta. Näihin ongelmiin voidaan pureutua projektointia työtä edeltävällä, kattavasti käytettävyyteen ja käyttöliittymään keskittyvällä vaatimusmäärittelyvaiheella, jossa hyödynnetään tarkoituksenmukaisia välineitä.

Tutkimukset osoittavat, että vaiheistettuja sovelluskehitysmenetelmiä käytettäessä jopa 40 prosenttia IT-hankkeiden elinkaarikustannuksista aiheutuu vääristä vaatimuksista. Voitaneen siis todeta, että alkuvaiheen tekemiseen pitäisi löytää uusia lääkkeitä; oikeat vaatimukset pitää saada esiin aikaisemmassa vaiheessa kehityshankkeen elinkaarta, jotta projektit saadaan suunnattua oikein. Ketterät menetelmät lupaavat parannusta, mutta niiden soveltaminen on usein sopimusteknisesti haastavaa niihin liittyvän jaetun riskin vuoksi. Tässä tekstissä pohditaan mahdollisuuksia luoda vaiheistetuille hankkeille laadullisesti parempi lähtökohta, tehokkaan käyttöliittymälähtöisen vaatimusmäärittelyvaiheen avulla.

Prototyypointityön apuna on usein muihin tarkoituksiin suunnitellut välineet

Käyttöliittymäprototyyppejä on jo pitkään hyödynnetty vaatimusten löytämisen välineenä. Esimerkiksi paperiprototyypit ovat nopeasti tuotettavissa, ja niiden avulla voidaan tehdä myös käyttäjätestausta. Muita usein käytettäviä ei-tekniisiä välineitä ovat muiden muassa MS PowerPoint, sekä Visio. Mitään edellä kuvatuista välineistä ei olla kuitenkaan lähtökohtaisesti suunniteltu käyttöliittymäprototyypin rakentamiseen, mikä näkyy niiden käyttömahdollisuuksien rajallisuutena. Erityisinä ongelmina voidaan mainita niillä tuotettujen prototyypin tarjoama rajallinen laatu sidosryhmäkommunikaation ja todellisen käyttäjäkokemuksen testausten kannalta, sekä puuttellinen kytkentä sovelluskehitysprosessin muihin tuotoksiin.

Teknisiä välineitä hyödynnettäessä ongelmaksi puolestaan muodostuu – niissä ketterämmissäkin tapauksissa – heikko tuottavuus ja vastaavasti kyvyttömyys skaalautua laajojen hankkeiden käyttöliittymien kattavaan (esimerkiksi poikkeustapaukset huomioivaan) ja interaktiiviseen prototyypointiin siten, että kehitystyö ja kustannukset pysyisivät aisoissa.

Entäpä jos otetaan kuvaan mukaan välineet, jotka on lähtökohtaisesti suunniteltu nimenomaan laadukkaiden ja interaktiivisten käyttöliittymäprototyypin kustannustehokasta tuottamista varten. Mikä voisi olla tällaisten – käyttöliittymien nykyistä tehokkaamman simuloinnin mahdollistavien – välineiden tuottama lisäarvo?

Käyttöliittymien simulointivälineillä oikeat vaatimukset esiin

Tarkennetaanpa simulointivälineiden tarjoama hyötypotentiaali hypoteesin muotoon: sovelluskehityksen riskejä voidaan nykyisestä pienentää siten, että palvelun täsmälliset vaatimukset – käytön onnistumisen ja siten asiakasorganisaation liiketoiminnallisen hyödyn kannalta – haetaan esiin koko järjestelmän toiminnot kattavan



Hyvän prototyyppivälineen ominaisuuksia

- Nopeus ja ylläpidettävyyys todellisen kaltaisten ja interaktiivisten prototyyppien tuottamisessa
- Prototyypit käytettävissä oikeilla laitteilla (esim. puhelin, PC)
- Monimutkaistenkin käyttötapausten simulointi poikkeustapauksineen mahdollista
- Helppo jaeltavuus, ei kynnystä kokeiluun esim. asennusvaatimusten muodossa

interaktiivisen ja todellisen kaltaisen prototyypin kautta, ennen varsinaisen määrittelytyön, teknisen suunnittelun ja toteutuksen aloitusta. Puhutaan siis prototyypistä, joka mahdollistaa hankkeesta keskustelemisen tehokkaasti niin sisäisesti kuin kumppaneiden kanssa, ja myös mahdollistaa – ainakin soveltuvin osin - lopullisen kaltaisen käyttäjäkokemuksen testaamisen. Tällainen malli tuntuisi melko luontevasti mahdollistavan vaatimusten – ja budjetin ja aikataulun - perustelumman kiinnittämisen teknistä suunnittelua ja toteutusta varten. Ai niin, ja asiakaskin tietäisi jo etukäteen mitä tulee saamaan.

Entäpä sitten tällaiset simulointivälineet, onko niitä? Maailmalla – vähemmän Suomessa - on tosiaankin etenkin selainpohjaisten sovellusten kehityksessä alettu hyödyntämään välineitä, joiden ideana on luoda kehitysvaiheen kommunikaatiolle ja käyttäjäpalautteelle mahdollisimman konkreettinen pohja, nopeasti tuotettavan ja lopullisen tuntuisen käyttöliittymän käytön kautta. 100M\$ vuodessa Web-projekteihin sijoittava Sprint säästi ensimmäisenä "simulaatiovuotenaan" 50% (17M\$) vuosittaisiin muutospyyntöihin varatusta budjetistaan (Forrester research: Show, Don't Tell: How High-Fidelity Prototyping Tools Improve Requirements Gathering, 2005). Uudenlaisilla menetelmillä ja välineillä ollaan järjestelmien käyttäjilleen tarjoaman laadun suhteen päästy - ainakin paikoin - huikesiin tuloksiin.

Toimintamallit ja välineet murrosvaiheessa

Käyttöliittymäsimuloinnin mahdollistavien välineiden markkina on nyt elinkaarensa alkuvaiheessa. Esimerkkinä markkinoilla olevasta tuotteesta voi mainita amerikkalaisen iRise-välineen. Omassa Realizer by SYSOPENDIGIA -tuotteessamme käyt-

töliittymien tehtävämallinnuspohjaisuus kytkee prototyyppivaiheessa tuotetut ja simuloidut selainkäyttöliittymät suoraan tekniseen toteutusvaiheeseen. Prototyyppi ei siis ole poisheitettävä asia, vaan lähtökohtaisesti osa teknistä projektitoimitusta. Näin pyritään nopeuttamaan kokonaisprosessien läpivientiaikoja ja siirtämään käytettävyyden eteen alkumetreillä tehtävä työ muuttumattomana hankkeiden lopputuloksiin.

Käyttäjälähtöisyys on joka tapauksessa nousussa kurssissa. Ja niin on myös tarve tehostaa käytettävyytyötä ja sen vaikutusta sovellusten laatuun. Tarvitaan paitsi menetelmiä ja ajatuksia, myös entistä tehokkaampia työvälineitä, jotka mahdollistavat kattavien ja laadukkaiden prototyyppien kustannustehokkaan kehityksen ja käytettävyysspanostusten nykyistä konkreettisemmän integroinnin ohjelmistokehitystyöhön. Lisäksi kavaataan uskallusta muuttaa vanhoja toimintamalleja, monilla toimialoilla myös tietojärjestelmien ostajan näkökulmasta.



World Usability Day – seminaari

Oulussa järjestettiin tänä syksynä, 8.11.07, kolmannen kerran World Usability Day (WUD) –seminaari vetäen yleisöä n 140 henkilöä. Seminaarin teemana oli ”Käytettävyys kehitty – kehittykö hyvinvointi?”

Timo Jokela, Joticon Oy

Joticon Oy auttaa asiakkaitaan varmistamaan, että kehitettävistä järjestelmistä tulee helppokäyttöisiä. Timo Jokela on käytettävyysalan kokeneimpia ammattilaisia Suomessa. Hän on työskennellyt aiemmin mm. VTT:llä, Nokialla ja Oulun yliopistossa, jossa hänellä on edelleen sivutoimi.

Anneli Pekkarinen, Työterveyslaitos, Ergonomia ja käytettävyys Ergonomia ja käytettävyys tiimin tavoitteena on kehittää työprosesseja ja työmenetelmiä turvallisemmiksi ja työvälineitä käytettävimmiksi tutkimus- ja kehittämishankkeiden, palvelun, koulutuksen ja tiedotuksen avulla. Anneli Pekkarinen on valtakunnallisen tiimin oululainen erikoistutkija.

Käytettävyyspäivän juuret Oulussa

Vuonna 2005 perustettiin kansainvälinen World Usability Day, jonka tarkoituksena on kiinnittää ihmisten huomiota siihen, että niin vapaa-ajan kuin työelämän tärkeät järjestelmät ja tuotteet ovat helposti lähestyttäviä ja yksinkertaisia käyttää. Päivän ideana on, että eri puolilla maailmaa järjestetään tapahtumia käytettävyysasian edistämiseksi.

Oulussa tartuttiin heti haasteeseen - Oulu oli syksyllä 2005 Suomessa ainoa paikkakunta, jossa järjestettiin yleinen World Usability Day -tapahtuma. Yliopistolle saapui noin 200 käytettävyystä kiinnostunutta tuotepäällikköä, tutkijaa, asiantuntijaa, suunnittelijaa ja opiskelijaa myös muualta Suomesta. Tapahtuman ympärille muodostui verkosto, jossa on mukana Oulun yliopisto, Oulun ammattikorkeakoulu, Työterveyslaitos sekä useita käytettävyysuunnittelun edelläkävijäyrityksiä Oulun seudulta. Seuraavana vuonna WUD järjestettiin ammattikorkeakoululla ja tänä syksynä jälleen yliopistolla. Idea on levinnyt muutamille muillekin paikkakunnille.

Terveydenhuollon järjestelmät käyttöön - ilman kurssituksia

Tämän vuoden käytettävyyspäivän kansainvälinen teema oli terveydenhuoltojärjestelmien käytettävyys. Lääkäreiden ja hoitohenkilökunnan käyttöön tarkoitetut järjestelmät ovat erityisen haasteellisia ja vaativia, koska niitä käytettäessä kyseessä on potilaiden terveys ja ääritapauksessa henki. Ammattilaiset ovat kiireisiä eivätkä aina ehdi osallistumaan uusien järjestelmien koulutukseen. Terveydenhuoltojärjestelmien suunnittelun yhdeksi käytettävyystavoitteeksi pitäisikin selkeästi asettaa, että lääkäreiden tulisi pystyä ottamaan käyttöön uusi järjestelmä yksinkertaisesti ”alkamalla vain käyttää sitä” - ilman erillisiä kurssitusten tarpeita.

Terveydenhuollossa on lisäksi nähtävissä suuntaus, että potilaiden tulisi ottaa entistä enemmän vastuuta omasta hoidostaan mielellään vielä kotiympäristössä.

Esimerkiksi tieto kotona m i t a t u s t a verenpaineesta tai verensokerista siirtyy terveydenhuollon tie-

tojärjestelmiin ammattilaisten tarkasteltaviksi. Koska hoitolaitteen ja järjestelmän käyttäjänä onkin potilas itse hoitotyön ammattilaisten sijasta, käytettävyys ja käytön turvallisuuteen on panostettava vieläkin enemmän.

Nykyisin terveystietoa etsitään entistä enemmän internetistä lääkärikirjojen sijaan. Nettilääkäriltä saa nopeasti vastauksen kysymykseen, mutta ovatko ohjeet ymmärrettäviä ja kannattaako niiden perustella ryhtyä toimeen. Toimivan järjestelmän lisäksi sisällön luotettavuudella on suuri merkitys.

Tutkimustuloksia ja uutisia

Oulun seminaarissa pidettiin kolme esitystä. Tutkija Leena Arhippainen Oulun yliopistolta kertoi käyttäjätyypeistä teknologiatuotteiden kannalta: mitkä ovat neljä pääkäyttäjätyyppiä, ja miten ne tulisi ottaa huomioon teknologiatuotteiden ja -järjestelmien suunnittelussa. Dosentti Risto Näsänen, Työterveyslaitos, aiheena oli visuaalinen käytettävyys. Aivotutkimukseen pohjautuen hän esitti, millä tavalla tieto kannattaisi esittää käyttäjille tietojärjestelmien käyttöliittymissä.

FT Timo Jokela, Joticon Oy, kertoi käyttäjäkeskeisen suunnittelun keskeisimmän standardin ISO 13407 (”Human-centred design processes for interactive systems”) uudistustyöstä: mitkä asiat ovat muuttumassa ja mitkä pysyvät ennallaan. Samoin hän kertoi siitä, miten standardin uusi sisältö syntyy. Tämän hetken näkymä on se, että standardin sisältämät käyttäjäkeskeisen suunnittelun periaatteet päivittyvät: aiemman neljän periaatteen sijalle on ehkä tulossa kuusi. Käyttäjakeskeisen suunnittelun prosessien lukumäärä näyttää säilyvän entisellään (neljä), mutta niiden kuvaukset muuttuvat.

Seminaarin päätti paneeli, jonka aihe oli ”käytettävyiden totuudet ja harhaluulot”. Paneeli perustui yleisön kokemuksiin ja mukanaoloon: tätä varten ilmoittautumislomakkeessa oli kohta, jonka osallistujat voivat täyttää. Paneelin vetäjät, Jarmo Vorne (Työterveyslaitos) ja Timo Jokela esittivät sekä yleisön tuottamia että etukäteen valmisteluja väittämiä, joiden paikkaansa pitävyys aluksi yleisö otti kantaa ja sen jälkeen panelistit. Panelisteina olivat Eeva Kangas (Sysopen-Digia), Leena Arhippainen, Pertti Puolakanaho (Polar Electro), Pirjo Ritokangas-Huttunen (Bellevues) Esimerkkejä väittämistä olivat ”Ikonit ovat parempi ratkaisu kuin tekstipohjaiset symbolit” ja ”Käytettävyysmenetelmiä voi soveltaa omatyylisesti, eikä tarvitse noudattaa kirjojen ohjeita tarkasti”.



Mikä mahtaa olla In?

10 vuotta Sytykeen laivasemiaareja! Mitä on tapahtunut kymmenessä vuodessa? Mikä on nyt ja mikä silloin IN? Tästä oli ottamassa selvää reilut 70 sytykeläistä.

Iloisia moikkauksia ja tervehdymiä kuului Viking Linen terminaalissa, kun tutut tapasivat. Ihan ensimmäisen moikkauksen kirjoittaja kuuli jo tullessaan, mutta ei suinkaan sytykeläiseltä, vaan toisen seminaariryhmän järjestelijöiltä. Olimme viime vuonnakin samassa kopissa jakamassa lippuja. Aika ja laiva olivat molemmilla ryhmillä samat: 5.-7.9.2007 ja Viking Mariella.

Valtaosa osallistujista oli hyvissä ajoin paikalla, ja niinpä semiaari saatiin alkamaan aikataulussa. Osaaminen ja verkostoituminen ovat ehdottomasti IN, niinpä puheenjohtajamme Mitro Kivinen avasi seminaarin kertomalla lyhyesti osaamisyhteisöistä Osyistä. Uusimpina osyinä ovat KäytettävyysOsy ja SOA SIG. Laivaseminaarityöryhmä esittäytyi ja kävi läpi muutamia käytännön asioita ja esitteli seminaarin sponsorit. Sponsoreiden avulla seminaarin kustannukset saadaan pidettyä sytykeläisille edullisina.

Osaaminen hukkuu isojen hankkeiden välillä

Seminaarin puheenjohtaja Ilkka Pirttimaa esitteli ensimmäisen puhujan Timo Kaislan Ixonos Projektipalveluista ja sitten päästiinkin asiaan. Timo kertoi projektijohtamisesta isoissa hankkeissa ja isolla hankkeella hän tarkoitti projektia, jonka kesto on yli kolme vuotta. Hän toi hyvin esiin asioita, mitkä aiheuttavat kuilun syntymistä suunnitelman ja toteuman välille. Kokeneet projektipäälliköt saattoivat vain nyökytellä tutuille asioille, kuten sille että liian usein työmäärä annetaan ylhäältä, jolloin se on ristiriidassa todellisuuden kanssa. Ajattelun aihettakin tuli: miten saataisiin ihmiset tajumaan, että näin iso hanke tulee vaikuttamaan kaikkien työnkuvaan ja elämään projektin aikana ja sen jälkeen.

Suurin haaste on kokonaisuusien hallinta

Claus Günther IDS Scheer Finlandista aloitti puheenvuoronsa kertomalla, että hänen muuttaman vuoden takaiset laivaseminaarikalvonsa olisivat olleet ihan toimivia nykyäänkin. Business Process Excellence oli hänen ja Sakari Lehtosen aiheena - työnimenä esityksellä oli Open BPM.

Suurihaasteon kokonaisuusien hallinta. Liiketoimintaprosessien automatisointi on yksi tämän hetken trendeistä. Kun nykyään prosesseista on automatisoitu noin neljännes, niin tulevaisuudessa luku olisi 80%. Tähän suuntaan ollaan menossa, sillä Business-Driven SOA ja Process Intelligence & Performance on tämän hetken IN-juttuja.

Sakari Lehtonen korosti, että liiketoimintaprosessien hallinta (BPM) on osa yritysarkkitehtuuria, jonka muita näkökulmia ovat sovellus-, informaatio- ja tekninen arkkitehtuurit. BPMS-ratkaisukerros pitää huolen liiketoimintaratkaisujen hallinnasta, vaikka sen alla ovat BPEL-toteutuskerrokset muuttuisivatkin. BPEL-generointi tehdään "alkuperäisestä" liiketoimintamallista kullekin toteutusalueelle erikseen. Tämä mahdollistaa sen, että toteutusalueista voidaan tarvittaessa vaihtaa.

Tietoturvatarkastuksesta vain 5% läpi puhtain paperein

Joakim Sandström nSensestä sai kaikki totaalisesti hereille kertomalla haavoittuvuuksista. Tietoturvan testaamisen pitää olla osa systeemiä, ei vasta jälkeensä suunniteltua. Haavoittuvuudet eivät enää ole verkossa, vaan 60-80 % kaikista haavoittuvuuksista on sovelluksissa. Kuitenkin 95% haavoittuvuuksista voidaan välttää kurinalaisella suunnittelulla ja toteutuksella - tietoturallinen sovellus ei synny itsestään.

Joakim lupasi esitellä demojaan myöhemmin seminaarin aikana, koska videoprojektori ei jostain syystä suostunut toistamaan hänen koneensa kuvaa.

Esityksen jälkeen suuntasimme pidemmälle tauolle majoittumaan ja nauttimaan kokoussämpylöistä. Juttua riitti niin esityksistä kuin kuulumisistakin - osa porukasta tapaa toisiaan vain täällä seminaarissa.



*Kokki lauloi
O sole mio'n
illallisella
Pavarottin
muistoksi.*

Seminaaria tukivat

A-sponsoreina:

CodeBakers

IDS Scheer Finland

Ixonos

nSense

Qentinel

SYSOPENDIGIA

Tieturi

B-sponsoreina:

Endero

Samcom

Vaatus valmiiksi viikossa?

Onko mahdollista saada vaatimus valmiiksi viikossa? Siis alkumäärittelystä toteutetuksi? Pentti Virtanen Tieturista valotti ketterien menetelmien periaatteita, mm. yksinkertaisuus on oleellista. Kyseessä on empiirinen prosessi. Eihän perunoitaakaan keitetä siten, että ensin mitataan 2,1 l vettä kattilaan, sijoitetaan sinne 10 perunaa. Lämmitetään vesi vähintään 100 asteiseksi ja sitten perunat ovat valmiita 42 minuutin kuluttua. Homma menee pieleen jos perunat ovatkin isoja; ne jäävät raaoksi, tai pieniä; ne hajoavat. Kyllä kypsyyssaste kokeillaan haarukalla! Mutta tuolla suunnitellun prosessin tavalla me kuitenkin yritämme rakentaa tietojärjestelmiä.

Kyllä, on mahdollista saada vaatimus idusta toteutetuksi viikossa, mutta siihen on luotava edellytykset. Näitä ovat mm. moniosajajohdannot, jotka tekevät työtä yhdessä päästä päähän, ja vaatimusten pilkkominen itsenäisiin, riittävän pieniin osiin.

Ei kun koko BI-ratkaisu 100 päivässä!

Minna Oksanen SYSOPENDIGIALta vastasi Pentin haasteeseen lupaamalla koko Business Intelligence -ratkaisun valmiiksi alle puolessa vuodessa.

Onko liiketoimintasi pulloposti (CMM 1), purjehvene (CMM 2-3) vai laiva (CMM 4-5)? Mikä auttaa ohjaamista? Ruori (sisäinen vaikuttaja) ja majakka (ulkoinen vaikuttaja). Business Intelligence on liiketoimintatiedon hallintaa sekä sisäisesti että ulkoisesti. BI-ratkaisun tehtävänä on tarjota oikeaa informaatiota oikeille ihmisille oikeassa paikassa.

Toisenlainen BI

Sitten olikin aika siirtyä vapaalle. Pienen tauon jälkeen kokoontuimme tutkimaan toisenlaista BI:tä, nimittäin Buffet Intelligenceä – miten syödä monenlaisia herkkuja tulematta ähkyyn kuntoon. Siinä onkin ihmettelemistä, kun tarjolla on alkupaloja kotoisesta kalapöydästä välimerellisten herkkujen kautta itämaisiin erikoisuuksiin. Ja lämpimät ruuat ja jälkiruuat päälle! Ruokailun lomassa keskustelu sujui pöydissä vilkkaana. Viihdyimme keskustellen niin hyvin, että tarjoilijat tulivat antamaan valomerkkejä ja kehoittamaan illan jatkamista muualla - olimme näet viimeisessä kattauksessa.

Geokätköilyä ja shoppailua

Torstaina kukin kävi omaan tahtiin aamiaisella ja Tukholmaan saapuessamme porukka hajaantui kuka minnekin. Helena ja Silja saivat mukaansa joitakuuta perinteeksi muodostuneelle antiikkikirjutorikierrokselle - uhkasivat tuoda mukanaan vanhan kiikkustuolin, mutta ei tainnut tällä kertaa löytyä. Ilkka oli löytänyt sytykeläisten joukosta pari muutakin innokasta geokätköilyn harrastajaa ja he keräsivät pienen joukon, joka kävi etsimässä pari Tukholmassa olevaa geokätköä ja jättämässä

niihin oman "aarteensa". Nämä ovat vain pari esimerkkiä päivän aktiviteeteista.

Java-kehityksen uudet tuulet

Klo 15 Ruotsin aikaa jatkoimme seminaaria CodeBakersin Perttu Karvisen ja Mikko Holmbergin esityksellä Javan uusista tuulista. Mikko kertoi, että Java7 on tulossa ehkä 2008-9 ja siinä on mm. modulaarisuuteen liittyviä parannuksia ja dynaamisten kielten tuki. Javalla on erittäin vahva open source -yhteisö. Hän painotti, että jos valitut kehitysvälineet ovat huonot, niin jopa 30% ajasta voi mennä välineen huoltoon ja sen kanssa tapeluun.

Perttu esitteli eri kehitysympäristöjen plussia ja miinuksia. Kehitysympäristöjä on sekä ilmaisia (Eclipse, Sun NetBeans), partnereille ilmaisia (BEA Workshop) että maksullisia (MyEclipse, CodeGear JBuilder, IBM RAD, IntelliJ IDEA).

Testaus auttaa liiketoimintariskien hallinnassa

Juhani Snellman Qentinelistä kertoi testauksen trendeistä. Testauksen liiketoiminnallisena tehtävänä on tuottaa tietoa, joka tuo riskit ja mahdollisuudet näkyviksi. Se ei siis suoraan lisää laatua. Testaus auttaa kustannustehokkaasti liiketoimintariskien hallintaa. Kyse on tärkeästä tukiprosessista. Tämän hetken trendejä ovat testauksen halpatuotanto, erikoistuminen ja ketterä testaus.

Juhani painotti, että on liian myöhäistä, jos hyväksymistestaus tehdään vasta lopuksi. Sitä pitää tehdä koko ajan. (Iteratiivinen ohjelmistokehitys mahdollistaa tämän, kirjoittaja huomauttaa.)

Mikä on ollutkaan IN kautta aikain?

Pienen virkistystauon jälkeen päästimme Silja Räisäsen Osuuspankkikeskuksesta ääneen. Hän oli erityisesti kutsuttuna puhujana, sillä Silja on ollut jokaisella risteilyllä mukana.

Silja pisti ihmiset keskustelmaan vierustoveriensä kanssa ja arvuuttelemaan, mikä on ollut IN aikasempina vuosina. Osa arvauksista meni kohdalleen, osa pieleen. Jotkut pieleen menneistä arvauksista jäivät ainakin kirjoittajaa mietityttämään, että olisikohan tuosta aiheesta pitänyt järjestää seminaari.

Silja esitti hyvän koosteen seminaarivuosisista omin havainnoin höystettyinä, ja sai ihmiset monin paikoin kommentoimaan ja nauramaan. Systemityö-lehti todettiin oivaksi tietolähteeksi seminaarien suhteen, sillä melkeiin kaikista on kirjoitettu juttu. Juttujen tyyli on vaihdelleet - tietysti kirjoittajansa mukaan.

Laivaseminaari 10 vuotta!

Mitro piti lyhyen yhteenvedon ja antoi lyhyen puheenvuoron seminaarin sponsoreille. Tällä kertaa mukana oli myös B-sponsoreita, jotka saivat hieman pidemmän ajan, koska heillä ei ollut varsinaista puheenvuoroa. Samcom ilmoitettiin omassa puheenvuorossaan jo ensi vuodelle

Laivaseminaarin konkari Silja.



A-sponsoriksi, mikä otettiin ilolla vastaan.

Mutta ei juhlaa ilman kuohuvaa! Osallistujat hakivat lasillisen käteensä ja Mitro piti juhlapuheen. Virallinen seminaariohjelma oli ohi ja illan päättäisi a la carte -illallinen hyvässä seurassa.

Rahat veks! Autsi!

Koska Joakimin läppäri ei ollut suostunut 1. päivänä yhteistyöhön tykin kanssa, hän ja Ilkka olivat viritelleet demoympäristöä Ilkan koneeseen. Saimme nähdä virallisen ohjelman päätteeksi pari tosi hiljaiseksi porukan vienyttä demoa. Tämä oli vapaa-ajalla ollut sessio, mutta porukka istui kuin liimattuina penkkeihin. Joakim näytti mm. miten helppoa on kadottaa rahat nettipokerissa, vaikka ei edes pelaisi tai häviäisi.

Ilkka totesi demojen aikana, että tämän jälkeen häneen koneensa menee kyllä formatointiin, sen verran monta tietoturva-asetusta otettiin pois päältä. Ja myöhemmin illalla parikin miestä totesi, että tältäköhän kastrointi tuntuu.

Viihteelle juhlaillallisen kautta

Kokoonnuimme Food Gardeniin juhlaillalliselle keskustelemaan ja nauttimaan ruuasta. Pääruean jälkeen oli "pakollinen" tietokilpailu, jossa mitattiin pöytäkuntien muistia seminaarin antimista. Edellisvuosista oli opittu, että porukka on hyvämuistista, niinpä joukkoon oli ututettu muutama tosi haastava kysymys hajonnan aikaansaamiseksi.

"Pienet konjakit, pienet konjakit ne lystikkäitä on..." viritimme pienen yhteislaulun. Ja saimme kahvin kanssa avecit. Aika lensi kuin siivillä ja taas meitä hätyyteltiin pois ravintolasta. Aika iso osa porukasta siirtyi keskustelemaan erilaisista aiheista lasillisen ääreen. Oli todella mukava nähdä osallistujien joukossa myös ensikertalaisia.

Katsotaan mitä ensi vuosi tuo tullessaan!

Laivaseminaarien teemat:

- 1998 Komponentit, olioteknologia
- 1999 Systemityön välineet ja menetelmät sekä koulutus
- 2000 Internet ja WEB-sovelluskehityksestä
- 2001 Arkkitehtuurit (työnimi?: Oliot ja komponentit)
- 2002 Testaus ja laatu
- 2003 Mallinnus ja menetelmät
- 2004 Systemityö ennen, nyt ja tulevaisuudessa
- 2005 Arkkitehtuurit - tekniikkaa ja taidetta
- 2006 Tuottavat ja laadukkaat prosessit
- 2007 Mikä mahtakaan olla IN?



Mitro avaa seminaarin.



Minna vastaa Pentin haasteeseen.

Pentti lupaa vaatimukset valmiiksi viikossa.





Ketteryys ja käytettävyys ja auta armias

Pitäisiköhän tässä oikein loukkaantua? Ymmärän kyllä viimekertaisen kahvihuone-episodin jälkeen, että päätoimittaja ehdotti välttämään kollegoitteni haastattelua. Tai no, itse asiassa termi taisi olla "lähestymiskielto". Mutta viis oikeusmurhasta, enemmän ihmettelin päätoimittajan pikkutarkkaa ohjeistusta, joka lähenteli sanan- ja yksilönvapauden riistoa puhumattakaan luovuuden kahlitsemisesta.

"Se on nyt sillä lailla, että tyylin on muututtava kertakaikkisesti. Enää et kirjoita mitään puoliseksistisiä ja kaksinaismielisen moraalittomia tarinoita, joilla ei ole mitään tekemistä systeemyön minkään aspektin kanssa! Seuraavan lehden teema on käytettävyys ja ketteryys ja auta armias, jos jutussa on yhtään mitään jostain povenpidikkeiden hakasten aukeavuudesta tai ketterästi keikkuvista minihameisista kesäheitukoista."

"Ja käytettävyys on myös kiellettyjen sanojen listalla!", karjaisi päätoimittaja vielä, kun olin jo sulkemassa ovea. Uskomaton tyyppi. Kirjoita tässä nyt sitten jotain järkevää, kun laitetaan tuollaiset rajoittimet päälle. Meno on kuin Myanmarissa.

Mikä ihmeen teema tuo nyt sitten on? Käytettävyyshän on lopputulos ja ketteryys tapa, jolla se voidaan saavuttaa tai sitten ei. Molemmat termit voidaan käsittää täysin väärin tai ainakin toisella tavalla kuin tarkoitus. Käytettävyys sekoitetaan saatavuuteen ja vastausaikoihin ja ketteryydellä mainostetaan niin moottoripyöriä kuin imatralaista urheiluakin. Tai siis itse asiassa käytettävyys oikeasti tarkoittaa myös tuota englannin "availability" -termiä. Kumpiakohan päätoimittaja nyt sitten itse asiassa tarkoitti, usabilityä vai availabilityä?

Entäs yhdistely? Ketterä käytettävyys? Tarkoittaisiko se mahdollisesti mahdollisimman joustavaa ja helposti omiin tarkoituksiin personoitavaa portaalia (jossa ohjeistus eli dokumentointi on unohdettuhah hah ... öh ... no, vanha vitsi). Sillekin varmaan on olemassa jo joku terminsä, jonka saisin varmaan selville Mutikaeselta, tiedä vaikka olisi hänen kehittämänsäkin. Mutta kun kerran päätoimittaja on kieltänyt lähes kaikenlaisten lähteiden käytön, niin jätätte rakkaat lukijani

nyt sitten ilman tätäkin tietoa. Lehden toimituskunta tosin yrittää kaikkensa saada tämän lehden täyteen asia-artikkeleita, joten ehkäpä muilta sivuilta löytyy sitten vastaus.

Yhdistämisen puolesta tietyllä tavalla puhuu mahtipontinen Utahissa after ski -tunnelmissa syntynyt ketterä manifesti. Se varmaan löytyy lehden muilta sivuilta, lukekaapas sieltä (tai jos ei löydy, niin lähettäkää palautetta päätoimittajalle, samalla voitte tietty ihmetellä, miksi Kuutamolla-kolumni juuri tällä kertaa oli jotenkin tukahdutetun tuntuinen). Laitan tähän vain viimeisen lauseen: "Vaikka oikeallakin puolella on arvoa, me arvotamme vasemmalla olevia asioita enemmän."

Kummastuttaa hieman tuo lause. Onko tuossa jokin poliittinen salamerkitys? Oliko tuo Utahin kokous 2001 jonkinlainen hajonnut itä-blokin masinoima maailmanvalloituksen aloitus? Jos mieltii oikein syvästi noita arvostuslauseita, niin eikös niissä olekin anarkian ja sekamelskan siemen? Mitä tästä hommasta tulisi, jos jätettäisiin dokumentointi tekemättä, ylenkatsottaisiin sopimuksia ja suunnitelmia? Jos individualistisen impulsiivista, mielihaluihin perustuvaa projektipoukkoilua alettaisiin arvostaa enemmän kuin pedanttis-preussilaismaisen projektisuunnittelun parhaita puolia, niin äärimmäisenä tuloksena voisi kyllä olla nopeasti ja kivasti tehtyjä toimivia sovelluksia. Mutta kuinka kauan tietäisimme miten ne toimivat? Maailma valuisi pikkuhiljaa kohti kaaosta, ohjelmistot ja koneet ottaisivat vallan, Skynet perustettaisiin ja se saavuttaisi tietoisuuden ... tule apuun, Arnold! Miettikäpä vielä, että heti tuon Utahin kokouksen jälkeen alkoi Lahdessa hiihdon MM-kisat, viikko sen jälkeen alkoi suu- ja sorkkatauti-epidemia Englannissa. Ja mitä tapahtui syyskuussa ...

Kerronpa vielä yhdestä tuttavastani, Rainier Ahvenvuosta. Hänellä on vaihtelevan menestyksenkäs yhden hengen konsulttiyritys. Viimeiset vuodet hän on kehittänyt mullistavaa uutta ketterän kehityksen mallia, joka hänen sanojensa mukaan saa ketterän manifestin kirjoittajat hyppäämään häpeästä mereen. Itse en ole saanut kunniaa tutustua tuon mallin saloihin, mutta eräänä heikkona hetkenään Rainier kertoi minulle nimenneensä menetelmän. Sen myötä hänen nimikirjaimensakin tulisivat loistamaan ohjelmistokehityksen maailmanhistorian ketjussa kirikkaana lenkinä. Ja siinä lenkissä lukisi: RAAGILE.

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

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

Lisätietoja SYTYKE ry:stä: www.sytyke.org

TOIMISTO

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

Johtokunta 2007

Puheenjohtaja Mitro Kivinen

p. +358 40 589 2724
mitro.kivinen@qentinel.com

Paula Miinalainen

p. +358 50 500 2363
paula.miiinalainen@arborvitae.fi

Seppo Takanen

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

Ilkka Pirttimaa

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

Tarja Raussi

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

Jori Rätty

p. +358 50 551 5152
joriraty@hotmail.com

Kari Uusi-Äijö

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

Varajäsenet

Lauri Laitinen

lauri.laitinen@nokia.com

Lea Virtanen

lea.virtanen@jobit.fi

Liittokokous- edustajat

Silja Räisänen

silja.raisanen@op.fi

Markku Niemi

make.niemi@kolumbus.fi

Mitro Kivinen

mitro.kivinen@qentinel.com

SYTYKE ry:n johtokunnan sähköpostilista:

hallitus@sytyke.ttlry.fi

Sytyttääkö? - Liity jäseneksi



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

Tietotekniikan liiton henkilöjäsenmaksu vuonna 2007 on alkaen 52€, erityisryhmien hinnoittelusta lisätietoja Tietotekniikan liitosta. Lisjäsenyys maksaa 11€/yhdistys.

Osaamisyhteisöt

Systeemyöyhdistyksessä 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 intresiryhmä, vetäjänä Simo Vuorinen.
simo.vuorinen@tietoenator.com

ProjektiOSY - PrOSY pyrkii yhdistämään Systeemyön 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. Suomen osaston johtoryhmän vetäjänä Pekka Valta, yhteyshenkilönä Minna Oksanen.
minna.oksanen@gmail.com

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

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

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

KäytettävyyOSY vaihtaa kokemuksia käytettävyyden kehittämiskeinoista sekä kehittää omaa käytettävyyssosaamistaan. Vetäjänä ja yhteyshenkilönä Helena Venäläinen,
helena.venalainen@op.fi.

RYHDY
LAUMASI
KEHITTYNEIMMÄKSI
YKSILOKSI

Tieturi
WWW.TIETURI.FI

www.tieturi.fi

KETTERÄSTI UUTEEN VUOTEEN!

Liiketoiminnassa tapahtuu muutoksia hurjaa vauhtia ja ohjelmistotuotannon pitäisi pystyä mukautumaan tuohon vauhtiin. Perinteiset tavat ovat lähes aina liian kankeita, niinpä päivän sana onkin ketteruus. Ketterät menetelmät kuten Scrum ja XP ovat vallanneet jo pysyvän jalansijan Suomessa. Koska ketteryydessä palautetta saadaan asiakkaalta jatkuvasti, myös käytettävyys on esillä jatkuvasti. Joko sinä hallitset ketteryyden? Tule syventämään tietämystäsi ja osaamistasi.

Ketterä ohjelmistotuotanto Scrumilla	10.1. ^{Tku} • 22.1. • 11.3.
Ketteryyden kulmakivet (0,5 pv) Uutuus!	9.1. ^{Tku} • 21.1. • 10.3.
Ketterän ohjelmistoprojektin läpivienti	28.-29.2.
Ketterä ohjelmistotuotanto ohjelmoijille	7.3.
Extreme Programming (XP) -menetelmistö	4.-5.2.
Agile Methods Java Workshop Uutuus!	18.-19.2.
Iteratiivinen ohjelmistokehitys (RUP) Uutuus!	28.-29.1.
Web-sivuston käytettävyys ja esteettömyys	9.4.
Ohjelmiston käytettävyyden testaus ja arviointi	14.4.
Prosessien mallintaminen	21.-22.1. • 13.-14.3.
BPMN prosessien kuvaamisessa Uutuus!	18.2.
Tietojärjestelmän käyttöönotto	4.-5.2.
SOA ja Web Services -yleiskatsaus	23.1. • 10.3.
SOA-palveluiden määrittely ja suunnittelu	4.-5.2.
Määrittely ja suunnittelu (UML 2.1) Uudistettu!	13.-15.2.
Vaatimusten määrittely ja hallinta	5.-7.2. • 17.-19.3.
Sovellusten elinkaaren hallinta (ASL Foundations)	25.-27.2.
ISEB ISTQB Foundation Certificate -valmennusohjelma	28.-30.1.
Testauksen valmennusohjelma	4.-6.2. • 10.-12.3. ^{Te}
Järjestelmäintegraation valmennusohjelma	3.3. alkaen
Tietojärjestelmäprojektin suunnittelu ja läpivienti	14.-16.1. • 20.-22.2.
ITIL Foundations (V3)	16.-18.1. • 5.-7.2.

Ilmoittautumiset ja lisätiedot: puh. 09 4315 5333
kurssit@tieturi.fi | www.tieturi.fi

Liiketoiminta kehittyy - kehity sinäkin.