

Trendit

Vuosituhan ensimmäisen vuoden päätyessä on syytä tarkastella mihin systeemityö on menossa. Lyhyellä aikavälillä tarkasteltuna näyttää siltä että kaikki muuttuu koko ajan. Töiden organisointia muutetaan edestakaisin, välillä hajautetaan ja sitten taas keskitetään. Uusia, mullistavia työmenetelmiä tulee ja vanhoja – hetki sitten yhtä mullistavia – menee. Sama työvälineiden kanssa. Oliopohjainen systeemityö on muuttumassa komponenttipohjaiseksi vai onko vain nimi muuttunut. Oliotekniikan tosiasaajien joukossa Yhdysvalloissa jo lähes kultiksi muodostunut ääriohjelmointi (Extreme Programming) innoittaa jatkoja menetelmäpuolella. Ehkä ensimmäinen yleisempään käyttöön leviävä formaali systeemityöprosessi Rational Unified Process on poikinut vastateesinään kevyet tai ketterät menetelmät (Agile Modeling). Heiluri heilahtaa äärestä toiseen ja muistuttaa meitä pidempään alalla olleita että päivän ismeihin on syytä suhtautua kuten aina järjellä, ei tunteella ja poimia sieltä sopivimmat, elinkelpoisimmat ajatukset käyttöön. Projektin osaamistaseen riittävydestä ja kilpailukykyisyydestä on aina syytä olla huolissaan. Sopiva kombinaatio alalla pitkään toimineiden kokemusta ja näkemystä ja vastavalmistuneiden intoa yhdistettynä projektin kokoon ja organisaatioon sovitettuun systeemityömenetelmään tuo parhaita tuloksia. Konkareita ei joka hankkeeseen aina riitä, jolloin konkareiden osaaminen olisi syytä saada talteen nyrkkisäännöiksi, menetelmäohjeistoiksi ja tietämyskannoiksi. Mutta miten nämä kommunikoisi muille tiimin jäsenille ja tiimien välillä. Siinäpä on haastetta ja tasapainoilua yhä tiukempien projektiaikataulujen kera ja maantieteellisesti hajautuvien kehittäjäorganisaatioille. Osaamistaan voi pyrkiä osoittamaan toimittaja- kieli- ja menetelmäkohtaisten sertifioitumisten lisäksi jatkossa ehkä jopa sertifioitumalla systeemiammattilaiseksi. Sertifioitumisten haaste tulee olemaan oikean osaamisen erottaminen bittitason nippelitieto-osaamisesta ja sertifikaattien pitäminen relevantteina nopeasti muuttuvien tekniikoiden ja käytötapojen maailmassa.

Sytyke toivottaa yhdistykseen liittyvän JavaSig:in jäsenet tervetulleiksi. JavaSig toimii jatkossa kerhona Sytykkeessä.

Tämän lehden sisältämien artikkelien on tarkoitus antaa lukijalle silmäys muutamista systeemityön trendeistä, yrittäen näyttää kehitystä hieman pitemmällä aikavälillä, erottaen metsän puilta.

*Hannu Kokko,
Puh: 040 751 6232,
Email: Hannu.Kokko@sysopen.fi*

Hannu Kokko toimii johtavana konsulttina ja toimitusjohtajana olio- ja komponenttitekniikan alueella SysOpen Object Team Oy:ssä. Hän on Sytykkeen hallituksen jäsen.

Ketterä mallinnus

Merja Kajava,
SysOpen Object Team Oy
ja
Sebastian Nykopp,
Talentum Oyj

Ketterä projekti

Ketterässä projektissa varaudutaan, että vaatimukset muuttuvat projektin edistyessä. Myös suunnittelu, teknologia ja jopa ihmiset vaihtuvat projektin edistyessä. Vaatimusten muutoksista aiheutuu riskejä, joiden hallintaan tarvitaan uusia toimintatapoja.

Tällaisessa tilanteessa vaaditaan edelleen suunnittelua, mutta tällöin suunnittelun tulee lähteä erilaisista lähtökohdista verrattuna perinteisiin ohjelmistokehitysprosesseihin.

Näihin tarpeisiin on viime aikoina vastattu kehittämällä menetelmiä, jotka tähtäävät kevyempään prosessiin. Kevyitä menetelmiä ovat mm. Crystal / Highsmith, XP (Extreme Programming) ja ketterä mallinnus (Agile Modeling - AM). Tässä artikkelissa tarkastellaan lähemmin ketterää mallintamista.

Mitä on ketterä mallinnus?

Ketterä mallinnus on tehokkaiisiin käytäntöihin perustuva menetelmä mallinnukseen. Ketterä mallinnus perustuu arvoihin ja periaatteisiin sekä näitä tukeviin käytäntöihin. Mallinnuksessa keskitytään projektin aikana perinteisiä menetelmiä aktiivisemmin tarkastelemaan projektin todellisia mallinnustarpeita.

Ketterää lähestymistapaa voidaan soveltaa vaatimusmäärittelyyn, analyysivaiheeseen, suunnitteluun ja arkkitehtuurin määrittelyyn. Tavoitteena on saada aikaan tarpeeksi hyviä malleja, mikä johtaa tehokkaa-

seen mallintamiseen. Mallinnuksen tuloksia syntyy ehkä vähemmän, mutta projektissa ollaan tietoisempia siitä, mitä ja miksi mallinnetaan. Ketterässä lähestymistavassa otetaan projektikohtaisesti enemmän vastuuta asioista, joita 'raskaammissa' menetelmissä on kuvattu prosessidokumenteissa.

Ketterä mallinnus keskittyy järkevään ja tarkoituksenmukaiseen mallintamiseen ja dokumentointiin. Kyseessä ei ole siis prosessimalli ohjelmistokehitykseen. Ketterä mallinnus on osa ohjelmistokehitysprosessia ja täydentää omalta osaltaan XP:tä, RUP:ia ja muita vastaavia prosessimalleja.

Arvot ja periaatteet

Ketterä mallinnus perustuu seuraaviin arvoihin:

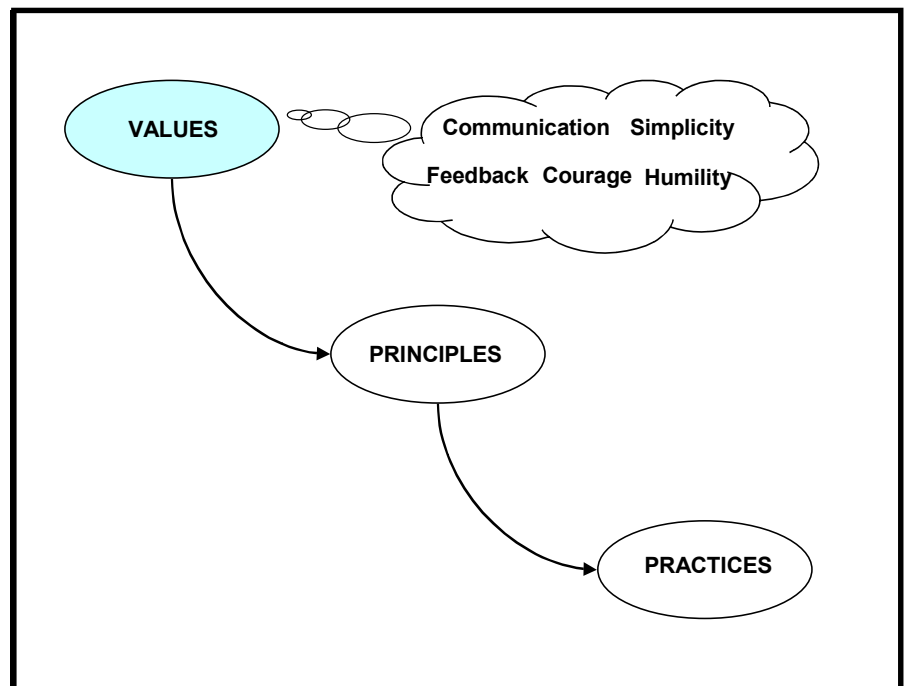
- Kommunikointi
- Helppotajuisuus
- Palaute
- Rohkeus
- Nöyryys

Neljä ensimmäistä arvoa pohjautuvat XP:ssä käytettyihin arvoihin. Ketterä mallinnus painottaa kommunikoinnin tärkeyttä mallinnuksessa. Ketterä mallinnus sopiikin parhaiten tilanteisiin, jossa projektin henkilöt toimivat samassa toimipisteessä, jopa samassa työtilassa.

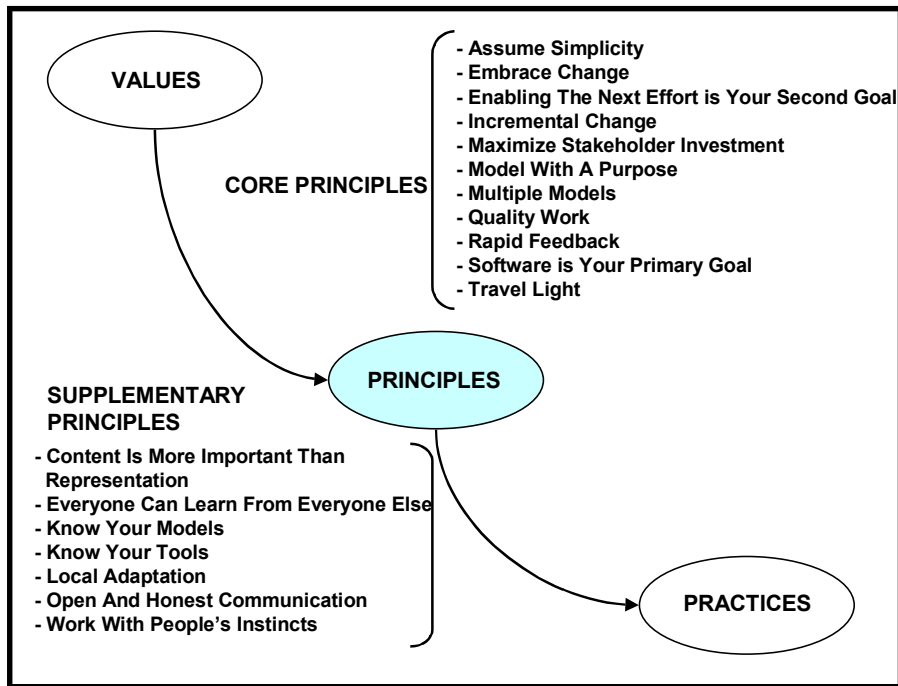
Periaatteet

Ketterään mallintamiseen liittyvät kiinteästi periaatteet, jotka jaetaan ydinperiaatteisiin ja täydentäviin periaatteisiin. Suurin osa periaatteista on samoja kuin XP:n toteuttamat periaatteet.

Dokumentoinnin 'keveydellä' ("Travel Light") pyritään siihen, että projektissa tuotetaan tarpeelliset dokumentit tarvittavalla tarkkuudella. Jokaista dokumenttia ja mallia on ylläpidettävä projektin kuluessa ja ylläpidolla on oma kustannuksensa. Projektikohtaisesti tulee harkita, mitä dokumentteja ja malleja tarvitaan ja millä tarkkuudella.



Kuva 1: Ketterän mallintamisen arvot



Kuva 2: Ketterän mallintamisen periaatteet

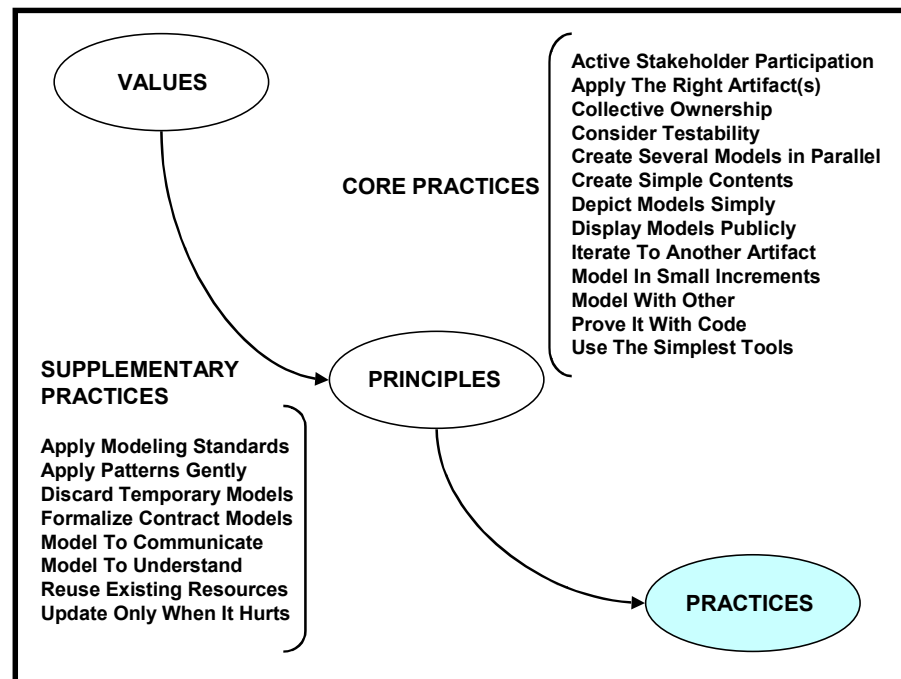
Käytännöt

Ketterä mallinnus ohjaa tiettyihin käytäntöihin, jotka jaetaan periaatteiden tavoin ydinkäytäntöihin ja täydentäviin käytäntöihin. XP:n mukaisia käytäntöjä on otettu mukaan mallinnusnäkökohdat huomioiden.

Ehkä tärkein ketterän mallinnuksen käytännöistä on asianomista-

jan aktiivinen osallistuminen projektiin. Asianomistaja laajennetaan tar koittamaan asiakkaan ja tilaajan lisäksi käyttäjiä, käyttäjien johtoa ja jopa tuotannon ja help-deskin edustajia. Käytännössä asianomistajan tulisi olla päivittäin yhteydessä projektin jäseniin, paikan päällä.

Vaatimusten muuttumisesta aiheutuneisiin riskeihin pyritään vas-



Kuva 3: Ketterän mallintamisen käytännöt

taamaan jatkuvalla integroinnilla. Mallinnuksen tuloksia voidaan varmentaa prototyyppisovelluksilla, jolloin saavutetaan konkreettisia tuloksia esimerkiksi sovelluksen suorituskyvystä ja käytettävyydestä.

Mallinnuksen tulokset voivat jäädä sisällöltään päivittämättä toteutuksen edetessä. Ketterässä mallinnuksessa pyritäänkin erottelemaan lopulliset tulokset ja väliaikaiset tulokset, joista voidaan luopua kokonaan projektin edistyessä. Mallinnuksen tuloksia tulee päivittää, mutta vasta tilanteessa, jossa päivityksen puuttuminen alkaa haittaamaan projektia ("Update Only When It Hurts").

Ketterässä mallinnuksessa korostetaan testattavuutta mallinnusvaiheessa. Jo mallinnusvaiheessa tulisi pohtia, miten kyseinen osuus voidaan testata.

Mihin ketterä mallinnus sopii?

Keskeinen edellytys ketterälle mallintamiselle on hyvin toimiva kommunikointi. Projektin jäsenien ja asianomistajien tulee olla 'samalla aaltopituudella', mitä päivittäinen kommunikointi luontevasti edistää. Ketterä mallinnus soveltuu hyvin projekteihin, joissa käytetään iteratiivista ja inkrementaalista prosessia. Iteratiivisuus onkin edellytys ketterän mallinnuksen vaatimalle kommunikoinnille ja palautteelle.

Ketterä mallinnus täydentää hyvin muita keveitä menetelmiä. Lähestymistapaa voidaan käyttää myös perinteisten menetelmien yhteydessä, mutta tällöin ketteryyden vaikutus voi jäädä vähäisemmäksi.

Ketterä mallinnus ei kuitenkaan sovellu jokaiseen tilanteeseen tai organisaatioon. Ketterää mallinnusta ei ole järkevää soveltaa, jos projektin asianomistajat eivät aktiivisesti osallistu projektiin.

Lopuksi

Scott Ambler, yksi ketterän mallinnuksen johtohahmoista, toteaa lähestymistavan käyttöönotosta seuraavaa:

"You can adopt as many of the principles and practices of AM as possible, you won't be truly doing AM, but you will likely be more productive as a developer."

Lähteet

The Official Agile Modeling (AM)
Site:
www.agilemodeling.com

Agile Alliance:
www.agilealliance.org

Alistair Cockburn:
The Agile Software Development.
Addison-Wesley. 2001

Merja Kajava

(merja.kajava@sysopen.fi) toimii johtavana konsulttina SysOpen Object Team Oy:ssä.

Sebastian Nykopp

(sebastian.nykopp@yahoo.fi) toimii asiantuntijana Talentum Oyj:ssä.

Systeemityölle yllättävää arvostusta

*Pekka Forselius,
STTF Oy*

Perjantaina 23.11.2001 järjestettiin Helsingissä Finlandia-talolla Kaartin Soittokunnan syyskonsertti Vaskivuoden marssit. Juhlallisen musiikin säestyksellä tilaisuudessa palkittiin myös muutamia ansioituneita sotilasmusiikin ystäviä, heidän joukossaan mm. rouva eduskunnan puhemies. Tärkein, seuraavan päivän Helsingin

Sanomienkin uutiskynnyksen ylittänyt tunnustus oli Savon Sotilassoittokunnan sotilasmestari Vesa Jähi valinta vuoden sotilasmuusikoksi. Vesa Jähi on ansioitunut sotilassoittajana, solistina ja soittokunnan vääpelinä. Erityisen maininnan ansaitsi kuitenkin hänen velvollisuutensa ylittävä aktiivisuus kehitettäessä atk-pohjais- ta sotilasmusiikkitietokantaa. Ainakin paikan päällä tilaisuutta seuranneelle tuli sellainen käsitys, että

ilman tätä systeemityöpanostusta monien hyvien soittajien rivistöstä olisi ollut vaikea nousta esiin. Vaikka ihmiset, jotka osaavat soittaa, ovatkin rouva puhemiehen mukaan pikkiriikkisen muita parempia. Siltikin sanon: Hyvä systeemityö! Se ratkaisee tiukoissa paikoissa!

*Pekka Forselius,
STTF Oy*

Kommunikaation merkitys ohjelmistokehityksessä

*Tuomo Kähkönen,
Nokia Research Center*

1. Ideoista ohjelmistoksi

Kevyet ohjelmistokehitysmenetelmät ovat nostaneet esille ihmisen tärkeimpänä yksittäisenä tekijänä ohjelmistotuotannossa. Ihmiset ja se kuinka he toimivat yhdessä, vaikuttavat lopputuloksena syntyvään ohjelmistoon laatuun merkittävästi. Tiimejä ja kommunikaatiota on toki tutkittu jo pitkään, mutta nyt näitä oppeja sovelletaan toden teolla myös ohjelmistotuotantoon. Uudet lähestymistavat täydentävät hyvin perinteisesti laadunvarmistukseen keskittynyttä prosessiajattelua.

Mitä ohjelmistokehitys oikeastaan on? Se koostuu kahdesta perustoiminnosta - uuden tiedon tuottamisesta ja kommunikoinnista. Näihin kahteen asiaan kiteytyy koko ohjelmistotuotteen elinkaari vaatimusmäärittelystä systeemitestaukseen. Ensin on ymmärrettävä ongelma. Sen jälkeen ongelmaan voidaan lähteä kehittämään ratkaisumalleja ja perimmäisenä tavoitteena on tuottaa lähdekoodi ohjelmistoon, joka ratkaisee kyseiset ongelmat. Ratkaisuja etsiessään ohjelmistokehittäjät luovat yksin ja yhdessä erilaisia abstrakteja malleja ongelmasta ja sen mahdollisista ratkaisutavoista. Ajatusten saaminen konkreettiseen muotoon siten, että ne voidaan kommunikoida muille ja jalostaa niitä yhdessä edelleen on suuri haaste.

Ajatuksista tulee konkreettisia silloin kun ne puhutaan, kirjoitetaan paperille tai syötetään tietokoneelle. Ajatusten konkretisoiminen on tärkeä vaihe kahdesta syystä. Ensinnäkin

ihmisen kyky käsitellä mielessään laajoja kokonaisuuksia on rajoittunut. Ohjelmistot ovat laajoja ja niiden toiminnallisuutta on tarkasteltava monesta toisiaan täydentävästä näkökulmasta. Mallit auttavat ohjelmistokehittäjää analysoimaan ja ymmärtämään paremmin omia ajatuksiaan ja ideoitaan. Toisaalta ajatusten saaminen konkreettiseen muotoon on edellytys niiden onnistuneelle kommunikoinnille toisille ihmisille. On myös helpompi palata aikaisempiin pohdiskeluihin kun edelliset tulokset on dokumentoitu.

Ajatukset jalostuvat matkan varrella kun niistä keskustellaan ja jopa väitellään. Esimerkiksi uusi ajatus voi alunperin olla keskustelussa ilmaan heitetty idea, josta laaditaan kahvipöydän ääressä luonnos kynällä ja ruutupaperilla. Sen jälkeen ajatus jalostuu sähköpostin välityksellä tapahtuvassa keskustelussa ja suunnittelupalavereissa. Lopuksi ratkaisu mallinnetaan ja sen käyttökelpoisuus todennetaan prototyypillä. Perinteinen prosessi virallisine dokumentteineen astuu mukaan vasta kun luova työ on jo suurelta osin tehty.

Tie ideasta suoritettavaan koodiin on pitkä ja siksi luovassa ohjelmistokehitysprosessissa tarvitaan välivaiheita. Aluksi luonnollista kieltä voidaan käyttää ideoiden kommunikointiin ja muistiin merkitsemiseen. Selkiintymättömien ajatusten ja laadullisen tiedon esittämiseen luonnollinen kieli on ylivoimainen ilmaisuväline. Luonnollinen kieli ei kuitenkaan ole ilmaisuvoimaltaan paras mahdollinen erilaisten rakenteiden ja toiminnalluuksien hahmottamiseen, vaan tarvitaan erilaisia mallinnusmenetelmiä. Koska kukin mallinnusmenetel-

mä kuvaa ongelmaa tai ratkaisua yhdestä näkökulmasta, tarvitaan useita näkökulmia kokonaisuuden hahmottamiseksi. Kaiken kattavaa yksittäistä mallia ei voida rakentaa.

Malli mahdollistaa ajatusten tehoakaan kommunikoimisen ihmisten välillä. Kun useampi ihminen tekee yhdessä luovaa työtä, tulee samaa asiaa tarkasteltua eri näkökulmista perustuen erilaisiin kokemuksiin ja tietotaitoihin. Tämä johtaa parempiin tuloksiin. Malli luo pohjaa ideoiden jakamiselle ja työskentelylle yhdessä, olettaen että mallinnusnotaatio ja semantiikka ovat molemmille tuttuja ja merkitykseltään samoja.

Mallin ei tarvitse olla alussa lähelläkään täydellistä - riittää että se on tarpeeksi hyvä keskustelun pohjaksi ja välittää olennaisen asian henkilöltä toiselle. Koska lopputuote on aina itse ohjelmisto, tulee mallin tarkkuus ja laatutaso aina suhteuttaa siihen, mitä tarpeita mallin käyttäjällä on. Systeemin toiminnallisuuden verifiointimisessa tarpeet ovat erilaiset kuin sen alustavassa suunnittelussa. Jos mallista tehdään liian aikaisessa vaiheessa täydellisempi kuin mikä on tarpeen, hukataan resursseja. Aikaa käytetään yksityiskohtien hiomiseen vaiheessa, jossa suuret kysymykset ovat vielä avoimina.

Myös ohjelmointikielen ilmaisuvoima on rajallinen. Käytettävissä on ainoastaan määritelty joukko käskyjä, joista voidaan muodostaa rakenteita ratkaisemaan kyseessä oleva ongelma. Käytettäessä malleja ihmisten väliseen kommunikaatioon ei mallien tarvitse olla täydellisiä, eikä käytetyn notaation täsmälleen oikeaa. Ohjelmointikielet sen sijaan ovat an-

teeksiantamattomia. Ohjelma suori-tetaan juuri sellaisena, kuin se on kirjoitettu. Ohjelmistokehitysproses-sissa ajatukset jalostuvat koodiksi, joka mahdollisimman hyvin vastaa al-kuperäiseen ongelmaan. Mallit ovat välivaiheita tässä prosessissa. Niiden avulla työstetään yksin ja yhdessä ajatuksia kohti lopullista muotoa.

2. Kahvipöytäkeskusteluja, kokouksia ja dokumentteja

Kommunikaatiota on muodolli-suudeltaan kahdenlaista - virallista ja epävirallista. Virallisena kommuni-kaationa voidaan pitää dokumentte-ja, tietokantoja ja kokouksia jotka on määritelty ohjelmistokehitysproses-sissa. Kaikkeaa muuta voidaan pitää epävirallisena. Monesti organisaati-ot suhtautuvat epäviralliseen kommu-nikaatioon ikään kuin sitä ei olisi-kaan. Kuitenkin käytännössä esimer-kiksi ohjelmistokehitysprojekteissa epävirallisella kommunikaatiolla on keskeinen rooli.

Virallista kommunikaatiota tar-vitaan erityisesti tapauksissa, joissa asianomaisten välille ei synny riittä-västi spontaania kanssakäymistä. He eivät ole fyysisesti tarpeeksi lähellä toisiaan tai he työskentelevät eri tii-meissä tai edustavat eri toimintoja (esim. markkinointi ja tuotekehitys). Virallisella kommunikaatiolla varmis-tetaan, että toiminnan kannalta riit-tävä määrä informaatiota liikkuu or-ganisaatiossa.

Viralliseen kommunikaatioon liittyy muutamia perusongelmia. En-sinnäkin tulee tietää etukäteen kuka tarvitsee mitäkin tietoa ja milloin. Koska tämä ei ole käytännössä aina mahdollista, voidaan päätyä tilantee-seen, jossa palaverit ja dokumenttien lukeminen vievät kaiken ajan varsi-naiselta työltä. Toisaalta voi käydä myös niin, että oman työn kannalta keskeistä tietoa jää saamatta. Toisek-si virallinen kommunikaatio tuo mu-

kanaan hitautta prosessiin. Käytössä voi olla aikaan sidotut raportointisyk-lit tai dokumentit toimitetaan eteen-päin vasta siinä vaiheessa kun ne ovat 'valmiita'.

Ohjelmistokehityksessä tiedon tarpeet muuttuvat tilanteiden mukaan ja ovat vastaavanlaisissakin tilanteis-sa usein erilaisia. Periaatteessa kos-ka tahansa voidaan tarvita mitä ta-hansa tietoa. Silloin on turvaudutta-va epäviralliseen kommunikaatioon ja etsittävä henkilö, joka tietää asiasta ja esitettävä kysymys hänelle.

Kommunikoinnin muodollisuu-den lisäksi voidaan tarkastella tapaa, jolla viesti toimitetaan henkilöltä toi-selle. Suullisella ja kirjallisella kom-munikaatiolla, samoin kuin erilaisil-la kommunikaatiomedioilla, on toi-sistaan eroavia ominaisuuksia. Ei voida sanoa, että jokin niistä olisi muita parempi, vaan erilaisiin tilan-teisiin tarvitaan erilaisia tapoja kom-munikoida.

Jos kommunikaatio on tiedotus-luontoista, ei kommunikaation koh-teena oleva tieto muutu, se vain ha-lutaan siirtää kohdeyleisön tietoon vieden mahdollisimman vähän hei-dän työaikaansa. Tiedotusluontoises-sa kommunikaatiossa esimerkiksi puhelinneuvottelu ja dokumentteihin pohjautuva viestintä ovat käyttökel-poisia. Usein kommunikaatio kuiten-kin tähtää uuden tiedon luomiseen, esimerkiksi jonkin ongelman ratkai-semiseen tai päätöksen tekemiseen. Silloin on olennaista osallistujien tie-don ja kokemuksen yhdistäminen uudella tavalla, joka on kokonaisuu-tena suurempi kuin osiensa summa. Tällöin tarvitaan myös tehokkaampia kommunikaatiomenetelmiä. Kasvok-kain tapahtuva keskustelu antaa mah-dollisuuden välittömään palauttee-seen ja kysymysten esittämiseen. Mu-kana on myös paljon sanatonta vies-tintää, joka jää pois puhelin- ja vi-deoneuvotteluissa. Kun kaksi ihmis-

tä on fyysisesti samassa huoneessa fläppitaulun ääressä, he ideoivat te-hokkaammin kuin puhelinpalaveris-sa.

Dokumentit ja tietokannat ovat hyviä suurten tietomäärien ja moni-mutkaisten ajatusten kommunikoimi-seen. Tiedon vastaanottaja voi tutus-tua tietoon oman vastaanottokykyn-sä mukaiseen tahtiin ja tarvittaessa palata aiempaan kohtaan lisäselvityk-sen saamiseksi. Ne myös luovat tie-don pysyvyyttä, joka mahdollistaa ajallisen ulottuvuuden kommunikaatiossa. Myöhemmin on mahdollista palata dokumenttiin ja katsoa mitä siellä sanottiin. Tämä koskee myös tiedon tuottajaa itseään - kukaan ei voi muistaa täysin tarkasti esimerkiksi mitä päätöksiä tehtiin puoli vuotta sitten ja minkä takia. Toisaalta doku-menttien huonona puolena on niiden kirjoittamiseen tarvittava aika, josta aiheutuu kommunikaatioon viivettä.

3. Kevyet menetelmät ja kommunikaatio

Prosessin tehokkuuteen vaikut-taa merkittävästi se, kuinka hyvin sii-hen osallistuvat henkilöt pystyvät koordinoimaan töitä keskenään ja kuinka hyvin he kommunikoivat kes-kenään prosessin tehokkaan suoritta-misen kannalta keskeisiä tietoja. Kun prosessi on monimutkainen, esimer-kiksi iteratiivinen tuotekehitysproses-si, voivat huonosta kommunikaatios-ta ja koordinaatiosta aiheutuvat tehok-kuushäviöt aiheutua pienessäkin tii-missä merkittäviksi.

Kevyet ohjelmistokehitysmene-telmät vähentävät tiimin sisällä tapah-tuvia tehokkuushäviöitä. Niissä kiin-nitetään paljon huomiota avoimeen, joustavaan ja tehokkaaseen kommu-nikointiin. Dokumenttien määrä py-ritään tietoisesti pitämään mahdolli-simman vähäisenä ja epävirallisen kommunikaation merkitys tunnuste-taan. Pyrkimyksenä on, että kommu-

nikointi tapahtuu osana päivittäisiä työtehtäviä jolloin viralliseen kommunikaation tarvittava aika ja ylimääräinen työ vähenevät. Pääasiallisena kommunikointitapana kevyissä menetelmissä on kasvokkain tapahtuva kommunikaatio, jolloin kommunikaatiosta ei aiheudu viiveitä ja toisaalta tarjoutuu mahdollisuus vuorovaikutteiseen asioiden tarkentamiseen ja ongelmanratkaisuun.

Kevyissä menetelmissä käytetään erilaisia tekniikoita riittävän epävirallisen kommunikaation varmistamiseksi. Esimerkiksi kun tiimi sijaitsee yhdessä tilassa ja toimisto on järjestelty siten, että tieto liikkuu mahdollisimman hyvin henkilöltä toiselle, epävirallinen kommunikaatio huolehtii suurelta osin tiedon välittämisen ongelmista. Pariohjelmointi sisältää jatkuvaa kommunikointia kahden ohjelmistokehittäjän välillä. Kun pareja vaihdetaan määräajoin, tehostetaan kommunikaatiota tiimin sisällä edelleen. Iteratiivinen prosessi puolestaan varmistaa sen, että tiedon vaihtoa tapahtuu mahdollisimman usein.

Kommunikaation keventäminen asettaa kuitenkin rajoja kevyiden menetelmien käyttökohteille. Isoissa projekteissa ei pelkällä epävirallisella kommunikaatiolla voida taata kaikkien osapuolten informoimista. Erietyisesti tilanteissa, jossa projektiin osallistuu kehitysryhmiä eri paikoissa, olkoonkin se sitten viereinen rakennus tai toinen manner, tämä on otettava huomioon. Yhtenä ratkaisuna on jakaa projekti tiimeihin, joiden sisällä kommunikoidaan epävirallisesti ja joiden väliseen kommunikointiin ja koordinointiin on käytössä muodollisemmat prosessit.

Myös tiimien välisestä epätäydellisestä kommunikaatiosta ja koordinaatio ongelmista aiheutuu prosessiin tehokkuushäviöitä. Yksittäisen tiimin ja koko organisaation työskentelyn tehostamiseksi työ tulisi jakaa eri tiimien kesken siten, että kukin tiimi voi tehdä omaa työtänsä mahdollisimman itsenäisesti ja kunkin tiimin vastuualue ja rajapinnat toisiin tiimeihin on selkeästi määritelty. Käytännössä tämä tarkoittaa sitä, että organisaatorakenteen tulee mukailla ohjelmiston arkkitehtuuria. Jos arkkitehtuuri on huonosti suunniteltu, heijastuvat siinä olevat puutteet organisaatioon tehtävien koordinoinnin vaikeutena ja kommunikaatio ongelmina.

Olipa arkkitehtuuri kuinka hyvä tahansa, ei kaikkia ongelmia ja niiden ratkaisuja pystytä koskaan eristämään yhden tiimin vaikutusalueelle. Silloin tarvitaan tiimien välistä yhteistyötä. Suuressa ja ketterässä organisaatiossa ei riitä, että tiimien välillä tapahtuu pelkästään virallista kommunikaatiota. Kommunikaatioverkoston tulee mukautua kulloiseenkin tilanteeseen ja pelkän tietojen vaihdon sijasta tulee pyrkiä aidosti yhteisten ongelmien tunnistamiseen ja niiden ratkaisemiseen yhteisvoimin. Tiimien tavoitteiden tulee olla keskenään yhdenmukaisia ja yhteisenä tavoitteena tulee olla mahdollisimman hyvin lopullisen asiakkaan tarpeita vastaavan ohjelmiston valmistaminen.

Ajallinen etäisyys tiedon tuottamisen ja sen tarvitsemisen välillä on yhtä merkittävä kommunikaatio este kuin fyysinen etäisyyskin. Kun ohjelmistokehitysprojekti ei ole kerta-luontoinen suoritus, vaan esimerkik-

si jatkuvaa tuotelinjan komponenttien kehittämistä tai uusien ohjelmistoversioiden tekemistä, ei voida luottaa pelkkään epäviralliseen kommunikaatioon. Kun kehitystiimin jäsenet vaihtuvat, vastuu ohjelmiston ylläpidosta siirtyy toiselle organisaatiolle tai on odotettavissa että ohjelmistoon tullaan tekemään muutoksia pitkän ajan kuluttua, on ohjelmiston dokumentoitava riittävästi nämä seikat huomioiden. Tämä ei ole ristiriidassa kevyiden menetelmien perusajatus-ten kanssa, mutta ei myöskään erityisesti niissä esille tuotu tekijä.

Kun suunnitellaan ja toteutetaan suuria taloudellisia tai ihmisiin kohdistuvia riskejä sisältäviä ohjelmistojä, asettaa laadunvarmistus dokumentoinnille omat vaatimuksensa. Katselmoinnit ja tuotteen verifiointi edellyttävät, että ohjelmisto ja sen suunnitteluvaiheet on tarkoin dokumentoitu. Joskus vaatimukset dokumentoinnille voivat tulla peräti lainsäädännöstä. Tällaisissa tapauksissa kevyitä menetelmiä ei voida aivan suoraan soveltaa, sillä ulkopuolisten dokumentointivaatimusten täyttäminen syö kevyiden menetelmien epävirallista kommunikoinnista saamaa hyötyä.

Kevyet ohjelmistokehitysmenetelmät ovat tulleet jäädäkseen. Menetelminä ne sopivat hyvin tietyn tyyppisiin tilanteisiin. Niiden taustalla olevilla ajatuksilla ja oivalluksilla on kuitenkin paljon annettavaa myös perinteisemmillekin projekteille. Ihmisten välisen vuorovaikutuksen ja kommunikaation kehittämisestä hyötyy jokainen projekti.

*Tuomo Kähkönen,
konsultti,
Nokia Research Center*

Trendit systeemityökoulutuksessa

*Päivi Hietanen,
KnowledgePool Tieturi Oy*

Systeemityö- ja projektinhallintakoulutus on Tieturin koulutuksen nopeimmin kasvava alue. Ennen trendit ohjasivat uusien työkalujen ja toteutusteknologioiden kouluttamiseen. Tänä päivänä trendejä säätelee yrityksissä läpivietävien projektien todelliset tarpeet – pelkän toteutustekniikan hallinta ei riitä.

Arvostettua räätälöintiä

Kun yritys päättää luoda oman menetelmänsä tai ottaa käyttöön olio- tai komponenttikeskeisiä työtapoja, tarvitaan kattava, useita osapuolia kosettava ja usein myös pitempikkestoinen koulutusohjelma. Usein yrityksellä on omia valmiita toimintamalleja ja menetelmistöjä, joihin koulutuksessa opetettavat menetelmät on sovitettava.

Koulutuksen on vastattava juuri niitä ominaisuuksia, mitä kullakin asiakkaalla on. Asiakkaat ostavat koulutuksen yhä useammin räätälöitynä, jolloin suunnittelu tehdään yhdessä asiakkaan kanssa. Räätälöidyn koulutuksen suunnittelussa onkin aina vuoropuhelu asiakkaan menetelmistötuntevan henkilön ja kouluttajan välillä. Parhaimmillaan vuoropuhelusta muotoutuu mentorointia muistuttava prosessi ja räätälöinnin tuloksia voidaan hyödyntää yrityksen menetelmäkehityksessä rakennuspalasina. Menetelmäkehitystä voidaan tukea myös koulutuksen jatkeena olevalla konsultointiprojektilla.

Työkalujen hallinnasta kokonaisvaltaiseen ajatteluun

1990-luvulla IT-alaa ovat mylertäneet monet uutuudet, joihin liittyvät työkalut ovat löytäneet tiensä toteuttajien käsiin. Systeemityön menetelmistöt ja työtavat ovat saaneet uusimmista trendeistä vaikutteita vasta, kun työkalut ovat jo käytössä. Oppirahoja joudutaan maksamaan, jos työkaluista tulee itsetarkoitus ja niitä käytetään ymmärtämättä taustalla vallitsevan ajattelutavan periaatteita.

Suomalaisissa yrityksissä on ymmärretty, että olio- tai komponenttilähestymistavan ylläpidettävyyss- ja uudelleenkäyttöetua ei saavuteta pelkästään ottamalla käyttöön ohjelmointikieleksi esim. Java. Hyödyt saavutetaan vasta, kun niihin pyritään tietoisesti suunnittelemalla ja sovittuja periaatteita noudattamalla. Myös järjestelmän toteutusarkkitehtuurin on saumattomasti tuettava näitä periaatteita.

Kuvaustapojen ja menetelmien hallinta on entistä tärkeämpää myös toteuttajille. Kuvauskielet, esim. UML tulee toteuttajien työpöydälle helposti nykyisten työkalujen kautta. Valitun lähestymistavan kouluttaminen ei keskity rajatun määrittelijäryhmän kouluttamiseen, vaan on muutumassa suuntaan, jossa lähestymistapa koulutetaan kaikille projektipäälliköistä toteuttajiin.

Perusasiat pysyy ja paranee

Järjestelmäkehitystä voisi verrata esim. kahvin tuotantoon, jossa voidaan erottaa perustuotantoon ja lopputuotantoon liittyvä prosessi. Perustuotantoprosessissa kahvin pavut korjataan, jauhetaan ja pakataan. Lopputuotanto tapahtuu lähellä asiakasta kahvibaarissa tai automaatissa ja siinä on eri nyanseja asiakkaan valitseman kahvityypin (café au lait, cappuccino) mukaan. Lopputuotanto antaa viimeisen silauksen ja hiukan eri makuvivahteet lopputuotteeseen.

Järjestelmäkehityksen vastaava perustuotantoprosessi sisältää mm. määrittely-, suunnittelu-, toteutus- ja testausvaiheet, jotka ovat olemassa riippumatta siitä missä muodossa järjestelmän palvelut loppukäyttäjälle tarjotaan. Lopputuotantoprosessin nyanssit tulevat näkyviksi erilaisten käyttöliittymäratkaisujen kautta (Windows, web, mobiili). Perustuotantoprosessin on toimittava virheettömästi, kun lopputuotantoprosessin halutaan onnistuvan. Perustuotantoprosessin avulla tuotetaan liiketoimintapalvelut takaava taustajärjestelmä (back-end) ja lopputuotantoprosessin avulla käyttöolosuhteita palveleva edustajärjestelmä (front-end).

Järjestelmän teknisen onnistumisen kulmakiviä ovat monitasoarkkitehtuuri sekä sen sisältämät olioista ja komponenteista muodostuvat rakennuspalaset. Kunnossa olevia järjestelmän tukipilareita ei horjuta hype-ilmiöt tai uusi toteutusväline, vaikka ne tuovatkin oman lisävärinsä

kehitystyöhön ja ovat ehkä osaltaan aiheuttaneet ammattitaidon kaventumista peruspilarien rakentamisessa. Koulutuksen kautta haetaan tänä päivänä tukea erityisesti peruspilareiden (olio- ja komponenttiarkkitehtuuri) vankistamiseen. Näihin liittyvien asioiden hallinta on tarpeellista kaikille projektipäälliköistä toteuttajiin.

Laatu ja liiketoiminta havahduttaa

IT:n rooli liiketoiminnassa on kiistatta melkoinen ja on vaikea löytää alaa, jota IT ei koskettaisi. Liiketoiminta tulee entistä riippuvaisemmaksi tietotekniikasta erityisesti kun liiketoiminnan osa-alueita aletaan siirtää verkkoon. Uhka liiketoiminnan haavoittuvuudesta on havahduttanut tietojärjestelmiä käyttävät ja tilaavat yritykset vaatimaan järjestelmiä, joiden arkkitehtuuri pystyy tarjoamaan liiketoimintaprosesseja tukevat palvelut ja muuntautumaan liiketoiminnan muuttuessa.

Kuvattu asetelma lisää järjestelmiä teettävän yrityksen halukkuutta olla vahvemmin kehitystyössä mukana ja ymmärtää sitä. Vastaavasti kehittäjältä vaaditaan oppimiskykyä ja kommunikointitaitoja, jotta järjestelmä saadaan kasaan asiakkaan toivomalla tavalla. Tieturin systeemyö- ja projektinhallintakursseilla tämä näkyy siten, että kuulijakunnassa on mukana sekä tietojärjestelmien tilaajia että kehittäjiä. Järjestelmäkehitystä tilaavien organisaatioiden kanssa yhteistyö on tiivistynyt entisestään ja asiakkaan tarpeisiin on toteutettu koulutusta mm. liiketoimintalähtöisestä vaatimusmäärittelystä, jonka oppien

pohjalta on helpompi kilpailuttaa kehitystyötä tarjoavia yrityksiä ja varmistaa tarpeellisen tiedon siirtyminen kehittäjille.

Lopputuotteen on siis oltava laadukas, jotta liiketoiminnan IT:ltä kaipaama tuki saavutetaan. Laatu pyritään varmistamaan monin eri tavoin. Systeemyön koulutuksessa tämä näkyy osallistujamäärän kasvuna erityisesti liiketoimintaan, järjestelmävaatimuksiin ja testaukseen keskittyvillä kursseilla. Järjestelmävaatimusten kartoitus- ja kuvaamisen menetelmät sekä testausmenetelmät ja -tekniikat ovat suuren suosion saavuttaneita osa-alueita. Satsaaminen näihin onkin viisasta, kun taloustilanne ei juuri suomahtollisuuksia virheistä oppimiseen.

Säätövaraa oltava

Liiketoiminnan ja järjestelmien rinnakkaiselosta seuraa, että liiketoiminnassa tapahtuvat muutokset aiheuttavat järjestelmien muutospainetta. Muutosvaatimukset on kyettävä käsittelemään nopeasti, jotta kilpailukyky myös tiukassa markkinatilanteessa säilyy. Asiakkaat asettavat erilaisia vaatimuksia tietojärjestelmien toimittajille. Myös toteutettavat projektit ovat luonteeltaan erilaisia. Kehittäjäorganisaatioon kohdistuvat paineet ovat moninaiset ja joustavuuteen tilanteen mukaan on sopeuduttava ja kuitenkin oleellisin tuotos – järjestelmä on aikaansaattava.

Ohjelmistotuotantoprosessit tuovat kehitystyöhön tehokkuutta ja jämäkkyyttä, jota ilman työ ei onnistu. Prosessia ja menetelmistöjen käyttöä on

kuitenkin pystyttävä säätämään valitsevien olosuhteiden ja projektin tarpeiden mukaan. Riskejäkin olisi pienennettävä toteuttamalla työ pienemmissä kehitysjaksoissa iteroiden. Aikaa ei ole myöskään hukattavaksi. Mukaan on sovittava muuttuvien vaatimusten käsittely ja testaus. Perinteisten raskaiden prosessien ja menetelmistöjen taivuttaminen nykytilanteen tarpeisiin ei ole helppoa. Kiinnostus joustavampia aikaa ja tuotosta korostavia menetelmistöjä kohtaan on kasvanut. Tieturin kautta on mahdollisuus tutustua suomalaisissa olosuhteissa kehitettyyn *Radit*-menetelmistöön. Kurssitarjonnasta löytyvät myös muualta maailmasta tuotettuja vaihtoehtoja, kuten *eXtreme Programming (XP)*, niitä janoaville.

Yhteenveto

Systeemyön koulutukselle asetetut vaatimukset heijastavat yritysten tarpeita pystyä tilaamaan ja toimittamaan juuri tarpeita palveleva järjestelmä. Painopistealueet ovat liiketoimintalähtöisyys, järjestelmävaatimukset, testaus, oliot, komponentit sekä nykyvaatimuksia vastaavat ohjelmistotuotantoprosessit ja -menetelmistöt.

Lisätietoja systeemyö- ja projektinhallintakoulutuksesta:
www.tieturi.fi/systeemyo
www.tieturi.fi/projektinhallinta
paivi.hietanen@tieturi.fi

*Päivi Hietanen,
Systeemyö ja projektinhallinta -
osaston osastopäälikkö
KnowledgePool Tieturi Oy*

Perinteinen SYTYKE-risteily

5.-7.9.2001

*Pirkko Kallaperä,
Pohjolan Systeempalvelu Oy*

Risteily alkoi ”pienellä” sählillä kun Heini huomasi 10 minuuttia ennen laivalle lähtöään, että laiva lähteenkin tänä vuonna satamasta jo kello viisi eikä kello kuusi kuten kaikille risteilylle lähtijöille oli ilmoitettu. Alkoi kiivas puhelinrumpa, jolla tavoitettiin ne puhujat, jotka olivat lähdössä laivalta pois sekä myös ne keiden kanssa sovittiin puheiden vaihdosta. Loppujen lopuksi aikataulua jouduttiin aikaistamaan siten, että ensimmäinen puhuja aloitti jo kello 14.00. Anne-Maj huolehti toimistosta käsin niille risteilijöille, jotka eivät olleet tulleet kello 14.00 mennessä, tiedon laivan lähtöajan muutoksesta, joten kukaan ei jäänyt rannalle riuuttamaan.

Agenda saatiin kuitenkin onnistuneesti uusittua ja ohjelma pääsi alkamaan. Päivi Hokkanen kellotti tiukasti puhujia etteivät nämä päässeet ylittämään aikaansa, ja näin pystyttiin pysymään aikataulussa.

Esityksistä päällimmäisenä jäi mieleen se, että systeemityön ammatilainen tarvitsee todella sanakirjaa. Jos ei meidän asiakkaat ymmärrä meitä, niin ei näitä asiantuntijoitakaan aina meinannut ymmärtää. Onneksi SYTYKE lehdessä aletaan julkaisemaan sanakirjaa, missä selvitetään uusien termien ja lyhenteiden sisältö.

Kun laiva lähti liikkeelle oli pakko ottaa tekniikka avuksi sillä muutoin puhujien ääni ei olisi kuulunut. Ihmisethän eivät vaihda paikkaa lä-

hemmäksi esiintyjä, vaikka näin olisi voitu taata kaikille hyvä kuuluvuus ilman mikrofoneja. Onneksi kuitenkin vatsat olivat täynnä hyviä sämpylöitä ja tuoretta kahvia tai teetä, niin työpäivän päätteeksi jaksettiin kuunnella mielenkiintoisia esityksiä.

En ole tarkoituksella kirjoittanut esityksistä täydellistä referaattia, vaan pienen raapaisun kustakin pinnalta, osoittamaan miten mielenkiintoisia aiheita olisi ollut mahdollisuus kuunnella risteilyllä.

Artikkelin loppuun olen koonnut koosteen risteilyllä annetusta palautteesta.

SWEBOK

Risto Nevalainen/STTF Oy

Ensimmäinen esitys oli SWEBOKista (SoftWare Engineering BOdy of Knollege) ja sen piti Risto Nevalainen STTF:stä.

SWEBOK eli Software Engineering Body of Knowledge on tulossa tärkeäksi dokumentiksi kuvaamaan ohjelmistotuotannon olennaisista sisältöä. Se sisältää 10 avainosaamisaluetta (Key Knowledge Areas). Dokumentin tasoa ja kattavuutta voi pitää erittäin hyvänä. Lähteenä on IEEE, mutta SC7 on mukana Liaison-järjestelyn kautta. Tavoitteena on antaa kommentteja ja huolehtia SWEBOKin vastaavuudesta ISO-standardeihin. Ensimmäinen kommentointikierros oli vuoden 2000 lopulla (dokumentti N2488). Lopullinen julkaisu lienee valmis vuoden 2001 loppupuolella.

SWEBOK dokumentti on referenssimalli, varsin tiivis, noin 200 sivuinen. Esimerkiksi testauksesta siitä on 20 sivun osuus, joka kertoo pääpiirteissään kaiken mitä testauksesta on hyödyllistä tietää.

SWEBOKin ydinalueita ovat:

- Software Requirements
- Software Design
- Software Construction
- Software Testing
- Software Maintenance
- Software Configuration Management
- Software Eng. Management
- Software Eng. Tools & Methods
- Software Engineering Process
- Software Quality

Lisätietoa SWEBOKista löytyy osoitteesta www.swebok.org.

Eri näkökulmat
arkkitehtuurin suunnittelussa
ja arkkitehtuuriratkaisujen
arviointi

Jari Isokallio/TietoEnator Oyj

Jari Isokallio toi esityksessään esiin sen, että on muistettava että ohjelmistokehityksessä on eri osapuolilla eri näkökulmat asioihin. Kun tunnistetaan eri näkökulmat, niin vältetään turhaa kuvaamista sovelluksen elinkaaren aikana.

Arkkitehtuurit / arviointimenetelmät

Laatuvaatimuksia on mm:

- suorituskkyky
- ylläpidettävyys
- luotettavuus
- tietoturva

Näitä osa-alueita ei pidä arvioida vain sanallisesti vaan kvantitatiivisesti.

Valitaan laatuominaisuudet, joita arvioidaan ja valitaan ominaisuudelle jokin vertailuarvo johon mittauksia verrataan. Eri näkökulmat taas tuovat arkkitehtuuriratkaisuihin mukaan pehmeitä arvoja.

Muut kuin kokemuspohjaiset arviointimenetelmät ovat vasta tulos kun ei ole vielä vertailudataa. Vertailua voi todellisuudessa suorittaa vasta kun vertailudataa on ensin kerätty.

Java-mobiiliteknologia

Janne Kalliala Firsthop.com

Hyvä esittäjä joka sai yleisön mielenkiinnon todella syttymään.

Aiheena JavaOne

SMS gateway tarjoaa selkeitä rajapintoja esim visual basicille, HTTP, HTTPS, Java applivaatiot, TCP/IP, XML

Lanseerauksen yhteydessä pitäisi olla yhteydessä kaikkiin neljään operaattoriin ja gatewayn avulla voidaan olla kaikkiin yhteydessä.

Lisätietoja <http://www.ief.org/inet-draft/dfrat-koponen-sms-xml-oi.txt>.

WMP problematiikka

Puhelimet bugisia ja eri puhelimilla erilaiset näytöt ja fontit. Aiheesta on ollut artikkeli Tietokonelehdessä tänä vuonna. Artikkelin kannattaa tutustua jos haluaa lisätietoa WMP:stä

WAP:ia käytetään kun halutaan saada täsmätietoa välittömästi. Sisältö tuotetaan erikseen ja tyyllisivujen avulla voidaan esittää erilaisilla vempaimilla XSL on hidas.

Gateway busineksessä osataan rahastaa.

EAI Arkkitehtuurit

Jari Vainikka/SysOpen Oyj

70% järjestelmäkehityksestä liittyy integraatioon.

On mitä yhdistellä sillä koodia tehty vuosikausia, yhdistelyä lisää komponenttitekniikat, muutokset business malleissa, joustavuus -> kyky yhdistellä olemassa olevaa ja tulevaa, verkottuminen, internet ja uudet päätelaitteet. Myös toimittajat ovat havainneet tässä uuden uuden markkinaraon

ERP (hoitaa vain 30% yrityksen tarpeista, loppu integroidaan olemassa olevista), eli on integroitava olemassaolevat järjestelmät ERP -järjestelmään.

Asiakas hootuu valitsemaan hallitsematon vai hallittu integraation väliltä. Tällöin XML-avainasemassa esim. yritysten välisessä integraatiossa.

Mobiililaitteita on kohta enemmän kuin verkkoon kytkettyjä työasemia ja mobiililaitte on ”aina” mukana, joten se asettaa omat haasteensa integroitavuudelle.

Sovellusintegraatio ja tietoturva

Jari Pirhonen atbusiness.com

Ennen oli tärkeämpää että ensin tunnistettiin asiakas ja varmistettiin tiedon eheys ja kolmantena tuli vasta käytettävyys, mutta nykyisin internetin näkökulmasta on tärkeää että sovellus on käytettävä, tieto eheää ja ei ole enää niin paha asia vaikka saataisiin vahingossa toisen ihmisen tiedot näkyviin.

Edellä oleva huomioinen laatu-ihmisten pitäisi ottaa kanta myös tietoturvallisuuteen.

Ongelmat syntyy vahingossa, väärin annettu syöttötieto = Murphy's computer, kun taas hakkeria vastaan Satan's Computer, eli hakkeri yrittää tietoisesti murtaa koodia ja löytää tietoturva-aukkoja.

Yrityksissä pitäisi olla tietoturva-asiantuntija, joka palauttaa laatu-ihmiset ja sovelluskehittäjät ym maan pinnalle.

Tietoturvallisen sovelluksen teon hinnasta ei ole dataa, eli tuleeko se kalliimmaksi, mutta on tutkittu sitä että kun sovellus käydään läpi tietoturvamielessä, niin sovelluksen koodista tulee tehokkaampi ja lyhyempi.

Tietoturva on huomioitava systemityön jokaisessa vaiheessa.

Määrittely-suunnittelu-toteutus-käyttöönotto-ylläpito.

Toteutusvaiheessa koodikatselmointi on tehokas tapa. Jokainen, joka tietää että hänen koodiaan katselmoidaan, tekee koodistaan laadukkaampaa.

On myös olemassa tietoturvaohjelmistoja, millä tutkitaan koodia ja etsitään siitä tietoturva aukkoja.

Oracle 9i arkkitehtuuri

Oracle/Kari Rouvinen

Kari Ruovinen johdatti meidät rautaisella myyntimiehen ammattitaidolla Oraclen 9 i arkkitehtuuriin. Kuulimme puolessa tunnissa, miten Oracle on integroinut yhteen porttaalit, kehittämisvälineet, kannat, ajoympäristöt, valvonnan sekä skaalautuvuuden. Uusia termejäkin vilahteli: portletti on ilmeisesti porttaliassa oleva apletti? Kari Rouvinen pystyi omassa osuudessaan hienosti vetämään yhteen edellisten esiintyjien kohokohtia ja viittaamaan niihin omassa esityksessään.

Oliot, tietokanta ja transaktiot, Jaakko Rantanen

Jaakko Rantanen HAMK
Relaatiokanta –kerho

Jaakko Rantanen palautti varmaan monelle osallistujalle elävästi mieleen opiskeluajat. Itselleni tuli heti mieleen fysiikan tunnint ja mikkihii-renkorvat, millä Jaakko aikanaan selvitteli atomin rakennetta.

Relaatiokanta –kerhon asiantuntijat toivottavat olioasiantuntijat lämpimästi tervetulleiksi, jotta saataisiin lisää tietoa oliomenetelmistä.

Tietokanta-ajattelevan ihmisenä oli mukava kuulla toisen sanovan, voidaanko systeemityötä tehdä ilman

tietokantasuunnittelu. Eikö muuten lähes kaikkia asioita voitaisi lähestyä tietokantamielellä.

Oliokuvauksista, luokka aliluokka, voidaan erilaisia relaatiokanta ratkaisuja, ja onkin laskettava mikiä malli olisi kulloinkin tehokkain. Esimerkiksi suoraan luokamallista relaatiomalli, siten että luokka ja aliluokat muodostavat omat taulunsa, tai luokasta oma taulu ja aliluokista yksi yhteinen ja vielä vaihtoehto jossa kaikki luokan ja aliluokan tiedon on viety samaan tauluun.

Transaktioiden yhteydessä eheydestä tulisi muistaa pitää kiinni myös olopuolella ja tietorakenteet ja algoritmit ovat edelleen IN Ja kaiken tämän perustana.

Oliomallinnus

Hannu Kokko SysOpen

Hannu esitteli ensin lyhyesti SysOpenin oliopuolen.

Olioparadigman perusidean on, että jokainen malli kuvaa järjestelmää tietystä näkökulmasta. Malli on todellisuuden yksinkertaistamista ja niiden avulla voidaan havainnollistaa järjestelmää. Oliomallinnusta liiketoimintakin ymmärtää. Mallintaminen on myös osa dokumentointia.

Puhu oikealla kielellä oikeille ihmisille, eli tekniikkaa ja olioita teknisille ihmisille, toiminnallisuutta käyttäjille. Dokumentaatio on oltava oikealla tasolla huomioon ottaen lukija.

TIRA on SysOpenin oliopohjainen tietojärjestelmien rakentamisprosessi ja se perustuu RUP:piin (Rational Unified Process). Tässä perusta on iteroiden rakentamisessa, joka tarkentuu koko ajan. Käsitteet ovat samannimisiä joka paikassa.

Hannu kertoi myös kuinka ROSEa (Rational ROSE) voidaan käyttää hyödyksi ja kuinka ROSElla saadaan tuotettua samantien käyttäjien ja esimerkiksi johdon dokumentaatio.

Ennen ROSE:n tai muun vastaavan käyttöönottoa pitäisi kuitenkin opetella UML –menetelmä ja mallintaminen yleensä. Pelkkä välineen käyttöönotto ei riitä.

Aktorien (käyttäjäroolit) avulla löydetään käyttötapaukset ja esimerkiksi valtuudet

Sovellusintegraation haasteet komponenttiarkkitehtuurien myllerryksessä

JP Pirvola/Cap Gemini

Integraatiospagetista Integraatiolasagneen.

Suuri osa työstä on tällä hetkellä liittymien koodaamista. Ennen tätä pystyttiin tekemään isolla aikaikkunalla, mutta nykyisin kaikki on tapahtuttava nopeammin.

Integraatiospagetti on siis se missä kaikki protokollat ja tavat tehdä töitä on spagettimaisesti sekaisin. Integraatiolasagnessa pyritään keroksittaisuuteen.

MOM Message Oriented Middleware, jonojen hallinta järjestelmä

XML – pelastaako se maailman? XML:ssä on useita kymmeniä eri murteita.

Valmisohjelmistojen integrointi

Tapani Ranta/ Generum.fi

Haasteet

Integroinnin ensimmäinen haaste on saada asiakas kertomaan mitä

halutaan, ymmärtämään asiakkaan prosessit ja tuntea integroitavat järjestelmät. Pitää muistaa arvioida tilannekijät. Toimittajia on voi olla useampia silloin ihmiset on hajallaan ja vastauksia on vaikea saada ongelmatilanteissa. Ei kannata ajatella että homma päättyisi siihen kun on integroitu, vaan muutoksia tulee koko ajan ohjelmistojen muuttuessa esim versio päivitysten tai tietotarpeiden muuttuessa.

ExperiencePro auttaa kun se ”pakottaa” toiminnallisuudellaan käymään läpi määrittäykset.

Uudelleenkäytön arkkitehtuuriratkaisuja

Silja Räisänen
Pohjolan Systeemipalvelu Oy

Pohjolan Systeemipalvelu on IT - yritys jonka pääasiallisina asiakkaina ovat Pohjola- ja Suomi konsernin yritykset.

Pohjolassa uudelleenkäyttöä on mietitty sekä tekniseltä että loogiselta kannalta. Loogiselta kannalta tarkastelua on tehty sekä tieto- että toimintopohjaisesti ja siten saatu uudelleenkäyttöä tukeva komponentteihin perustuva sovellus- ja tietoarkkitehtuuri. Molempia arkkitehtuureja kehitettiin ja edelleenkehitetään liiketoiminnan ja IT-väen yhteistyössä.

Sovellusarkkitehtuuri perustuu kerrosajatteluun, jossa ylimpänä ovat varsinaista liiketoimintaa tukevat vakuutusjärjestelmät (esim. Suurasiakkaiden kuljetusvakuutusjärjestelmä). Seuraavassa tasossa yleiset järjestelmäkomponentit, joita kaikki järjestelmät käyttävät, jakaantuen omiksi järjestelmikseen (esim. Asiakashallinta, Laskutus ja reskontra, Myynti- ja palkkiot). Alimpana tasona sovellusarkkitehtuurissa ovat yleispalvelut (esim. valtuudet, tulostus, yleiset tietokantapalvelut, rajapinnat muihin

järjestelmiin). Sovellusarkkitehtuuri tukee myös valmisohjelmistojen hankintaa ja integrointia osaksi sovellusarkkitehtuuria.

Tietoarkkitehtuuri perustuu syvälliseen osaamiseen vakuutusliiketoiminnan tietotarpeista. Tietomalli on yhteinen kaikille järjestelmille (yleinen, yhteinen ja yleistetty). Se on suhteellisen stabiili: ei muutoksia, vain lisäyksiä. Tietoarkkitehtuurin perusteella on syntynyt myös joukko yleispalveluja sovellusarkkitehtuuriin.

Vakaan perustan näille loogisen tason arkkitehtuureille antavat standardisoitut ympäristöt ja menetelmät sekä yleinen infrastruktuuri (tekniset arkkitehtuurit). Myös WEB-kehittäminen on sopinut hyvin näihin arkkitehtuuriratkaisuihin ja on hyötynyt omalta osaltaan valmiista komponenteista.

Työtä ollaan näiden arkkitehtuuriden varaan tehty jo useita vuosia onnistuneesti. Pääasiallisina syinä onnistumiselle ovat olleet liiketoiminnan vahva tuki ja IT-väen syvälinen osaaminen vakuutusliiketoiminnan sovelluskehityksestä.

Sponsoreiden mietteitä siitä miksi ovat mukana risteilyllä.

Kaikki SysOpenilaiset kuuluvat SYTYKKEeseen. SYTYKKEen tilaisuudet ovat hyvä tapa verkostoitua ja SYTYKKEen tilaisuuksissa saadaan tietoa siitä missä mennään.

CapGeminin mielestä verkostoituminen on kivaa ja CapGeminissä tarvitaan osaavia työntekijöitä.

TietoX on lähettänyt ihmisiä tilaisuuksiin.

FD Finanssidata on runsaasti mukana, mutta ei ole haalimassa asi-

akkaita vaan verkostoitumista. SYTYKE on hyvä juttu.

Oraclen mielestä risteilyllä on selvästi pitkän kokemuksen omaavia IT vaikuttajia, joille Oracle haluaa tuoda Oracle 9i viestiä Suomessa enemmän.

TietoEnatorin mielestä verkostoituminen on parasta sekä arkkitehtuurit ja tämä risteily on hyvää.

STTF on SYTYKKEessä vahvasti mukana vaikka onkin pieni firma. Arvostaa verkostoitumista, parasta osaamista, IT projektinhallinta koulutusta ja kehottaa jäsenistöä tulemaan mukaan Fisman toimintaan.

Pohjolan Systeemipalvelu Oy oli risteilyllä tuomassa tietoa siitä, että yrityksessä on paikkoja osaaville IT-alan ihmisille.

WEB-toimikunnan osuus

Helena Venäläinen esitteli Webtoimikunnan ja kertoi sivuston uudistusprojektista. Sivustojen sisällön suunnittelun lähtökohdiksi otettiin it-alan tietotyöntekijöiden ammattitaidon ylläpitäminen ja kehittäminen, sekä yhteisten tiedonkehittelyprosessin tukeminen myös verkossa. Projekti tehtiin yhdessä Helsingin Yliopiston Mediakasvatuskeskuksen kanssa. Sivuston juuri verkkoon siirtymässä ollut versio sai ristelyjulkistuksen. Läsäolijat vastattasivat kiihkeästi kyselyyn, jonka tarkoituksena oli osaltaan varmistaa valitun profiloinnin oikellisuus sekä antaa suuntaa sivustojen jatkokehittämiselle. Tulos vahvisti linjan oikeellisuuden ja sivuistoille toivottiin seuraavaksi eniten kerhotoimintaa sekä tietopalvelua. Kyselyn vastausten yhteenveto on jo matkalla web-sivuillemme. Onnetar arpoi kaikkien vastanneiden kesken palkinnon - pullon kuohuviiniä voitti Aila Heinilä.

Risteilyesitysten arviointi

Risteilyn teema oli arkkitehtuuri. Osanottajat saivat esitysten lopuksi täytettäväkseen risteilypalautteen, jossa sai risteilyn lisäksi kertoa ideoita myös koko Sytyke ry:n toiminnan kehittämiseksi.

Risteilyn kokonaisarvosana asteikolla 1 (ihan mälsä) - 5 (hyvä) oli 3,27. Risteilytoimikunta oli hoitanut järjestelyt hienosti ja sai siitä kiitosta. Ohjelma kokonaisuutena koettiin hyväksi ja teemaa käsiteltiin arvioiden mielestä monipuolisesti sekä tasapainoisesti. Esiintyvät saivat kiitosta sekä asiantuntijuudestaan että esiintymistaidoistaan. Parhaaksi esitykseksi arvioitiin ehdottomasti Janne Kallialan Java-mobiiliteknologia.

Kehittämistäkin olisi. Eniten toivottiin esitysten pidentämistä, sillä ½ tunnissa ei ehdi paljon sanoa. Uusille ja mielenkiintoisille asioille pitäisi löytyä myös sulatteluaikaa. Keskustelua, paneleita ja seminaarityypistä työskentelyä kokemusten vaihdon ja käsitteistön selvittämisen merkeissä toivottiin seuraavien risteilyjen ohjelmaan. Aiheiksi (lehteenkin) toivottiin myös It-ammattilaisen huomioivia teemoja, esim. työssä jaksaminen, väsymyksen/stressin syyt, sovellusten käytettävyys, käyttöliittymät.

Risteilyn tarpeellisuutta ei kuitenkaan asettanut kyseenalaiseksi, joten ensi vuonna tavataan taas.

Mistä on Sytykkeen toiminnasta kysymys Pekka Forselius/SYTYKE

Lähtökohtana Sytykkeen toiminnassa on harrastus ja ammatillinen kiinnostus asiaan.

Jäseniä on tällä hetkellä 2193. Kun verrataan siihen, että Suomessa on noin 40000 ohjelmistoalan ammattilaista ja näistä noin 10000 systeemityön ammattilaista, niin voidaan todeta että neljännes heistä kuuluu SYTYKE ry:hyn.

Tärkeät asiat SYTYKKEessä

1. sijalla on tietenkin lehti, joka pyritään pitämään korkeatasoisena.

- seuraavan lehden teema on TRENDIT, esim. menneisyyden paluu
- keväällä ilmestyy kaksi numeroa, joiden teemoista toivotaan ehdotuksia

2. Sijalla on SYTYKKEen ker-

hot

- relaatiotietokanta
- testauskerho, testaus voisi olla ensivuonna risteilyn teemana
- datanomi-tradenomi -kerho
- JavaSig -kerho kunhan JavaSig yhdistyy sytykkeeseen tämän syksyn aikana

lisäksi työryhmät WEB työryhmä ja Sujuva systeemityö työryhmä

Tulevia tapahtumia on mm keväällä kevätkokous, syksyllä 2002 viides perinteinen syysristeily.

Sytykkeen johtoryhmä esittänyt tehtävineen ja Helena Venäläinen esitteli WEB projektin työn tuloksia ja projektin etenemistä.

Risteilyn muut ohjelma

Alkuhäslinkien ja tiukan ohjelman jälkeen risteilyväki pääsi ostoksille laivan myymälöihin ja toiset menivät kostuttamaan kurkkuaan pubissa. Aikaa ei kyllä ollut paljon ennen illan ruokailua, joka tapahtui buffet ravintolassa. Ruokailun jälkeen porukka jotenkin hajosi ympäri laivaa, mutta sen mitä kenestäkin havaintoja tein, niin kaikki näyttivät nauttivan olostaan.

Torstai päivä meni omalla ohjelmalla Tukholmassa, kuka mitäkin tehden omien mieltymysten mukaan. Kun laiva sitten lähti kohti Helsinkiä, niin kaikki saapuivat kuuntelemaan toisen illan esityksiä.

Esitysten jälkeen saunottiin joulukolla siten, että aika meinasi loppua kesken, ainakin naisilta. Kaikki tälläntyivät kuitenkin iltaa varten, jolloin syötiin juhlallisesti á la Carte ravintolassa. Siinä ruokailun ja yhteislaulun lomassa kyseltiin kuinka moni oli käynyt kaikilla Sytykkeen risteilyillä, ja aika monta niitäkin löytyi, mutta löytyi myös noin kolmannes osallistujista jotka olivat mukana ensimmäistä kertaa. Yksi asia oli kaikilla yhteinen, kaikki halusivat tulla seuraavalla kerrallakin mukaan.

*Pirkko Kallaperä,
Pohjolan Systeemipalvelu Oy*



Talouden laskusuhdanteessa IT-investoinnit ja -hankkeet tarkempaan syyniin

Elina Mäkelä,
Market Visio Oy

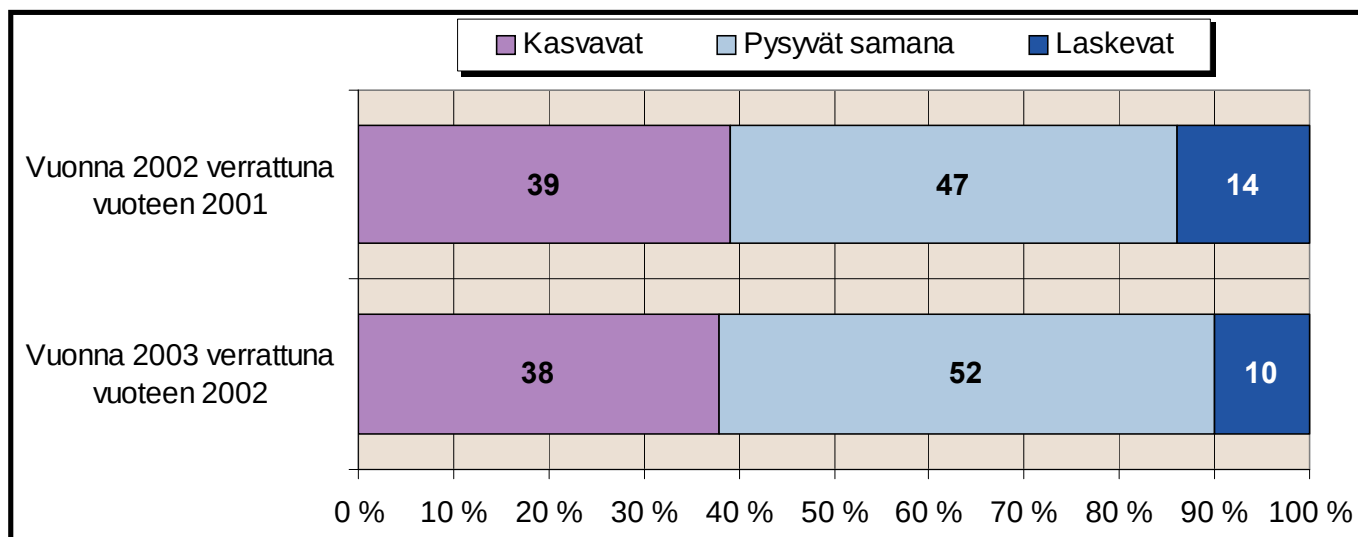
Uudella vuosituhannella yrityksissä on kaavailtu runsaasti uusia IT-hankkeita ja -investointeja. Termit kuten e-business, CRM ja knowledge management esiintyvät tiuhaan keskusteluissa ja suunnitelmissa. Nyt kuitenkin vaikuttaa siltä, että taloudellinen epävarmuus on saanut organisaatiot varpaalleen hankkeiden läpiviennin suhteen: toteutettavia hankkeita mietitään yhä tarkemmin, niitä saatetaan lykätä tai pistää kokonaan jäihin.

IT-kustannusten kasvu hidastuu

Market-Vision lokakuun 2001 alussa julkaistu tutkimus ”IT-investoinnit, -kustannukset ja -hankkeet Suomessa 2001 – 2003” kertoo yritysten kasvavasta varovaisuudesta IT-investointien ja -hankkeiden suhteen. Tutkituista 141 yrityksestä ja julkisesta organisaatiosta 39 prosentissa IT-kustannusten arvioidaan kasvavan ja 14 prosentissa laskevan vuonna 2002 edellisvuoteen verrattuna (ks. Kuvio 1). Kun tulosta vertaa vuoden 2000 syksyllä toteutettuun vastaavaan tutkimukseen, jonka mukaan 57 prosenttia vastaajista odotti IT-kustannusten kasvavan vuonna 2001, taloudel-

lisessä tilanteessa tapahtuneen negatiivisen käänteen vaikutukset ja organisaatioiden lisääntynyt varovaisuus ovat selvästi havaittavissa. Kuitenkaan mitään äkkijarrutusta IT-kustannuksissa ja -investoinneissa ei ole näköpiirissä, ja huomattavan suuret leikkaukset rajoittunevat yksittäisiin yrityksiin.

Tutkituissa yrityksissä ja organisaatioissa odotetaan IT-kustannusten kasvavan lähivuosina keskimäärin hyvin hitaasti: 1,9 prosenttia vuonna 2002 ja 1,6 prosenttia vuonna 2003. On kuitenkin oletettavaa, että jos taloudessa tapahtuu käänne myönteiseen suuntaan tai jos tilanne säilyy samanakin, kustannusten kas-



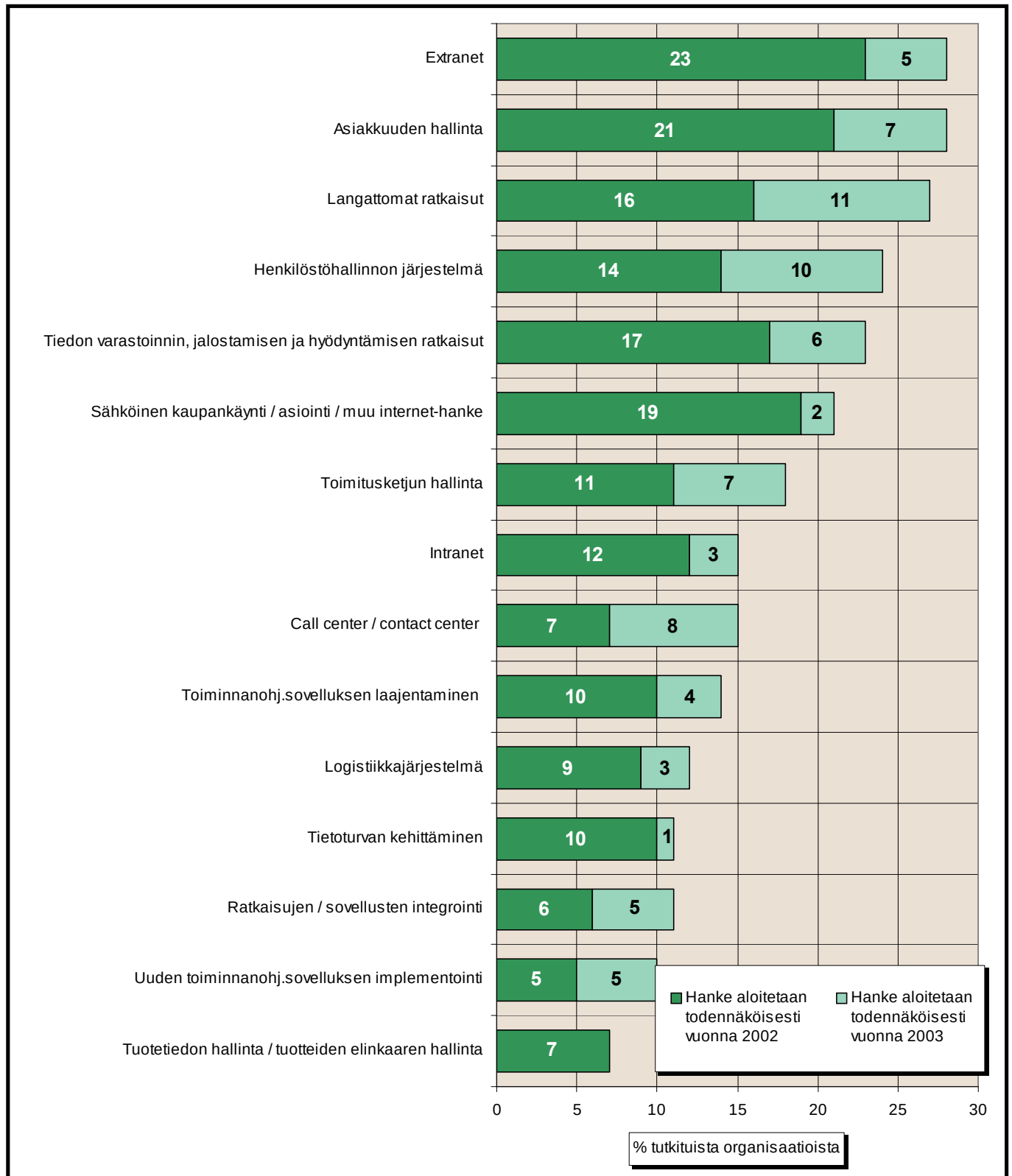
Kuvio 1. IT-kustannusten kehittyminen tutkituissa organisaatioissa vuosina 2002 ja 2003 edellisiin vuosiin verrattuna

vu tulee olemaan vastaajien ennakoimaa nopeampaa – kokemuksemme mukaan vastaajilla on tutkimustilanteessa taipumus aliarvioida kustannusten kasvua. Jos taas talouden negatiivinen kehitys vahvistuu, kustannusten kasvu jää hyvin hitaaksi tai tyssää kokonaan.

Lähipuosille kaavaillut IT-hankkeet

Lähipuosille kaavaillut IT-hankkeet liittyvät tutkituissa organisaatioissa useimmin extranet-ratkaisuihin, asiakkuuden hallintaan sekä langattomiin ratkaisuihin (ks. Kuvio 2).

Verrattuna vuonna 2001 toteutettaviin hankkeisiin extranet-hankkeet sekä muut sähköisen kaupan ja asiointin hankkeet ovat pinnalla myös vuoden 2002 uusien hankkeiden joukossa. Asiakkuuden hallinnan hankkeet nousivat yleisimpien hankkeiden joukkoon, mikä heijastaa asiakassuhteiden



Kuvio 2. Vuosina 2002 ja 2003 aloitettavaksi kaavaillut hankkeet

den kehittämisen ja asiakaskannattavuuden parantamisen kasvavaa merkitystä organisaatioille. Vuoden 2001 kärkihankkeista intranet-hankkeiden yleisyys näyttää puolestaan olevan laskusuunnassa, mikä kertoo siitä, että intranetit alkavat useimmissa organisaatioissa olla kunnossa (tosin jatkokehityshankkeita tullaan varmasti toteuttamaan).

Erityyppisistä hankkeista *langattomien ratkaisujen kehittämiseen liittyvät hankkeet* yleistyvät voimakaimmin: vuonna 2003 niitä kaavailaan aloitettavan hankkeista useimmin. Langaton tiedonsiirto on liiketoiminnassa suhteellisen uusi asia, ja toistaiseksi suurin osa toteutetuista hankkeista on ollut kokeiluluonteisia. Teknologian kehittyessä ja ratkaisujen kypsyessä hankkeisiin satsataan enemmän, ja aikaisten kokeilijoiden ohella myös muut lähtevät liikkeelle.

Laskusuhdanteen vaikutukset toteutettaviin IT-hankkeisiin

Haastattelujen suorittamisajankohdan (elo-syyskuu 2001) jälkeen organisaatioissa on saatettu muuttaa hankesuunnitelmia talouden negatiivisen kehityksen vahvistuessa. Sen lisäksi, että hankkeita on saatettu karsia ja lykätä, toteutettavissa hanketyypeissä on myös saattanut tapahtua muutosta. Laskusuhdanteessa painopiste siirtyy sellaisten hankkeiden

toteuttamiseen, joiden avulla *pyritään aikaansaamaan kustannushyötyjä tehostamalla ja rationalisoimalla yrityksen toimintaa*. Hankkeiden tavoitteissa korostuvat silloin ”kovat” tavoitteet (esim. investoinnin tuotto ja tuottavuuden parantuminen) ”pehmeitä” tavoitteita (esim. käyttäjätyytyväisyyden parantuminen) enemmän.

Aikana, jolloin kustannussäästöjen ja tehokkuuden tavoittelu korostuu, sellaiset hankkeet, joilla pyritään *uuden liiketoiminnan luomiseen, tuottojen lisäämiseen ja/tai kilpailukyvyyn parantamiseen*, jäävät usein taka-alalle. Tämänäyttävät hankkeet perustuvat usein uusien teknologioiden tai ratkaisujen hyödyntämiseen, ja siten projektien toteuttamiseen ja tavoiteltavien hyötyjen toteutumiseen liittyy enemmän epävarmuutta ja riskiä. Kuitenkin voi olla lyhytnäköistä jädäyttää tällaiset projektit kokonaan, sillä jos ne toteutetaan ”huonoina aikoina”, ratkaisut ovat hyödynnettävissä heti ”hyvien aikojen” palatessa, jolloin niistä saadaan maksimaalinen hyöty toimitaan.

Yhteenveto

Yleistaloudellinen tilanne ja yritysten menestyminen vaikuttavat oleellisesti yritysten mahdollisuuksiin investoida IT-ympäristöönsä ja toteut-

taa uusia hankkeita. Pidemmällä tähtäimellä – kun taloudellisten suhdanteiden vaikutukset jätetään huomiota – on kuitenkin todennäköistä, että organisaatioiden IT-panostukset jatkavat tasaista kasvua. Tähän vaikuttaa mm. se, että IT:sta tulee yrityksille ja organisaatioille yhä kriittisempi resurssi: IT siirtyy taustan tukifunktiosta eturintaman toiminnoksi, jonka avulla on mahdollista saavuttaa tuntuvaa liiketoiminnallista hyötyä edellyttäen, että järjestelmien kehittämiseen satsataan jatkuvasti.

Myös e-business on tässä suhteessa merkittävä tekijä: useimmat yritykset ja organisaatiot hyödyntävät tai ryhtyvät hyödyntämään jollain tavoin sähköisen liiketoiminnan tarjoamia mahdollisuuksia, ja monet yritykset panostavat tuntuvasti tälle alueelle. Eräinä tekijöinä voi vielä mainita yrityskaupat ja -fuusiot sekä yritysten kansainvälistymisen: yhteisten tietojärjestelmien, verkkojen ja prosessien luominen yksiköiden välille vaatii tuntuvia rahallisia panostuksia.

*Elina Mäkelä,
analyttikko (Sourcing Strategies),
Market-Visio Oy*

Kannattavuuden ja kustannusten hallinnan uudet tuulet

Pekka Forselius,
STTF Oy

Ohjelmistokehityksen kustannusten hallinnasta on tehty pilkkaa pitkän aikaa, eikä aivan syyttä. Ovat-han kustannusarviot ylittyneet usein moninkertaisesti ja hankkeiden aikataulut venyneet niin, ettei lopulta valmistunut ohjelmisto olekaan vastannut enää sen hetkisten markkinoiden tarvetta. Kyyniset tarkkailijat väittävät, että näin on aina ollut ja näin tulee aina olemaan. Asiaan liittyviä ilmiöitä kannattaa kuitenkin tarkastella pitemmän aikavälin valossa. Voimme erottaa ohjelmistoprojektien hallinnan historiassa kolme erillistä aikakautta: ennen 1980-lukua vallinnut ”jura-kausi”, sitä seurannut ”pimeä keskiaika” ja muutama vuosi sitten alkanut ”nykyaika”. Kausien tar-koissa alkamisvuosissa on toki suurta vaihtelua toimialoittain ja yrityk-sittäin, mutta jaottelu kelvannee kansallisella tasolla. Seuraavilla sivuilla tarkastellaan ohjelmistotyön kannattavuuden, tehokkuuden ja kustannusten hallinnan kehittymistä Suomessa näiden kausien aikana. Jotain parannusta on tapahtunut jo viime vuosikymmenellä ja juuri nyt on il-massa vahvoja merkkejä todellisesta muutoksesta.

Yleiskehitys viime vuosikymmeninä

Automaattisen tietojenkäsittelyn ja ohjelmistokehittämisen alkuvuosikymmeninä - aina 1970-luvun loppuun asti projektityö oli pääasiassa erilaisten organisaatioiden sisäistä tuotekehitystä. Projektien kannattavuuden kanssa ei ollut erityisiä ongelmia, koska automatisoinnin avul-

la pystyttiin saamaan yksinkertaisiin perusliiketoiminnan prosesseihin valtavia parannuksia, eikä toimittajien välistä kilpailua ollut. Jokaisella ohjelmistojen teettäjällä oli oma ”hovi-toimittajansa”. Ulkopuolisia ohjelmistotoimittajia ei juurikaan ollut. Kustannuksista saati sitten tehokkuudesta ei edes puhuttu, korkeintaan ihmeteltiin noiden kummallisten atk-suunnittelijoiden korkeita palkkoja. Tavallisen kadunmiehen näkökulmas-ta heille maksettiin mystisestä mutta täysin näkymättömästä työstä kadeh-dittavan suuria summia. Hyvä palkka ja helpot hommat.

1980- ja 1990-luvut toivat atk-alalle kilpailun. Tuotantorakenteet muuttuivat ilmeisen pysyvästi. Ohjelmistotyössä siirryttiin valtaosaltaan aitoon tilaaja-toimittaja-asetelmaan. Erityisesti tämä koskee ”perinteisiä” kaupallishallinnollista tietojenkäsittelyä harjoittavia organisaatioita. Vasta tämän kauden aikana nousunsa alkaneessa sulautettujen ohjelmistojen tuotannossa kehitys laahaa luonnollisesti jonkin verran jäljessä. Kokonaisuuden kannalta tarkasteltaessa voidaan kuitenkin sanoa, että kustannuksista, aikatauluista ja niiden pitävydestä tuli viimeisten kahden vuosikymmenen aikana jokapäiväinen puheenaihe. Kannattavuuskin on ruvennut kiinnostamaan entistä enemmän, koska yhä monimutkaisempia ja vaikeammin hallittavia prosesseja yritetään kehittää automaation avulla. Kilpailukykyistä myös tehokkuuteen alettiin kiinnittää huomiota, ainakin teoriassa.

Kolmannen vuosituhatosen alkaessa maailmamme on muuttunut ja muuttuu yhä kiihtyvällä vauhdilla

kansainväliseksi. Globalisoituminen tuntuu meidän jokaisen työssä, kun oman tai läheisten yhteistyöyritysten ensimmäiseksi kieleksi on tullut englanti. Jopa ruotsia, ranskaa tai saksaa täytyy jossakin määrin hallita, vaikka niitä ei sentään dokumentoinnissa tai käyttöliittymissä vielä vaadita. Myös toimittajien välinen kilpailu on muuttunut suuressa määrin kansainväliseksi. Oman suolaisen ripauksensa tämän päivän ja lähitulevaisuuden hankkeisiin tarjoaa hajauttaminen. Projektit muodostuvat monella paikkakunnalla, monessa maassa ja monella kielellä toimivista soluista, joiden kuvitellaan ilmeisesti tehostavan tuloksellisuutta, nopeuttavan läpimenoaikoja ja parantavan laatua verrattuna entiseen. Kannattavuuteen ilmeisesti uskotaan vahvasti, mutta ainakin kustannusten kurissa pysymistä sopii vahvasti epäillä. Ainakin hajautetut hankkeet ovat tähän mennessä lisänneet sekä matkustelua että erilaisia sähköisen kommunikoinnin välineiden kokeilemista. Sen voi varmasti luvata, että monimutkaisesta tuotantorakenteesta ja kaikille systeemyön osa-alueille ulottuvasta kilpailusta johtuen tulevaisuuden hankkeissa kannattavuuden ja kustannusten hallinta ei ole ainakaan helpompaa kuin menneissä. Pieneen pelkoonkin voisi olla aihetta, kun muistaa aikaisempien aikakausien ”onnistumiset”.

Menetelmien ja menettelytapojen kehittyminen

Ohjelmistotyössä kannattavuuden, tehokkuuden ja kustannusten hallinnan keskeiset elementit ovat työ-määrä, työn hinta ja tietojärjestelmän suuruus. Myös muilla kehittämis-hankkeisiin liittyvillä hankinnoilla on

vaikutusta kokonaiskannattavuuteen, mutta niiden kustannusvaikutuksen arviointi ei ole läheskään yhtä vaikeata kuin tarvittavien työmäärien ennustaminen. Siitä on eniten kenkä puristanut - silloin kun on puristanut.

(Kuva 1.)

”Jura-kauden” projekteissa – ylläpitotöistä puhumattakaan – ei kerätty toteutuneita työtunteja lainkaan. ”Ihmisethän ovat täällä joka tapauksessa töissä.” Erilaisia töitä tehtiin ja kiireellisiä tehtäviä hoidettiin sen mukaan mitä päivän mittaan eteen tuli. Koska kustannustietojen kerääminen laiminlyötiin, ei kannattavuuteen tai tehokkuuteenkaan voinut ottaa objektiivisesti kantaa. Yleisesti ottaen jokainen piti omia hankkeitaan erittäin kannattavina ja keskitasoa parempina. Niin onkin helppo väittää, jos kukaan ei mittaa mitään. Käytännössä ei mitattu sen enempää käytettyjä panoksia kuin tuotettuja tuloksiakaan. Ohjelmistojen suuruutta tosin yritettiin silloin tällöin mitata rivimäärien avulla, jotta olisi voitu asettaa ohjelmoijille normitavoitteita. Rivimäärä on kuitenkin melko huono ohjelmistojen koon mittaamiseen. Se on suunnilleen yhtä hyvä kuin ”poron kusema” matkojen mittaamiseen – aika paljon kiinni porosta ja sen syömisistä, ja lisäksi täysin useimpien ihmisten kokemuspiirin ulkopuolella. Lisäksi kaikki asiantuntijat tietävät, ja tiesivät jo silloin, että ohjelmistojen tuottaminen on erittäin helppoa jos ohjelmilta ei muuta vaadita kuin pituutta. Kirjallisuudessa ja alan lehdissä rivimääriin perustuvia tehokkuuslukuja kuitenkin esitettiin – paremman puutteessa.

(Kuva 2.)

”Pimeällä keskiajalla” projektityöstä ruvettiin laskuttamaan, joka johti väistämättä siihen että työtuntit piti alkaa kirjata projekteittain. To-



Kuva 1.

sin tämä käytäntö ei kosketa vieläkään kovin kattavasti tuotekehitysprojekteja, mutta valtaosa tietojärjestelmätyöstä tuli kuitenkin tämän ajanjakson kuluessa kohtalaisen tarkan raportoinnin piiriin. Työmäärien kerääminen ja kohdistaminen hankkeille avasi käytännössä mahdollisuuden sekä kannattavuuden että tuottavuuden arviointiin. Hanke- ja tuotekohtaisen kannattavuuden osalta suomalaiset yritykset ovat kovin vaatimat-

tomia, tai sitten ne eivät halua kilpailukykyisistä julkaista tuloksiaan. Ulkopuoliset tarkkailijat voivat kyllä aavistaa, että muutamat vaikkapa Nokian ohjelmistokehitysprojekteista ovat täytyneet olla erittäin kannattavia, mutta tarkka tieto siitä on korkeintaan yhtiön sisällä. Monissa itseään ammattimaisena pitävässä organisaatiossa käynnistettiin 80- ja 90-lukujen aikana tehokkuuden mittaamiseen, vertailemiseen ja parantami-



Kuva 2.

seen tähtäviä sisäisiä kehityshankkeita. Muutamissa näin tehtiin jopa useita kertoja. Tulokset olivat kuitenkin muutamaa poikkeusta lukuun ottamatta lähinnä masentavia.

”Pimeän keskiajan” kaikki projektit olivat aluksi hyvin väljästi sovitut, laskutuksen perustuessa raportoituihin työmääriin. Ohjelmistojen toiminnallisia vaatimuksia ei määritelty etukäteen millään tavalla – paitsi ehkä ihan tärkeimpiä suunta- viivoja. Lukematon määrä ”pikkujuttuja” lisättiin sitten järjestelmäkehityksen aikana ilman sen kummempaa byrokratiaa. Joustavaa ja mukavaa monien mielestä. Hankkeet olivat kestoltaan niin pitkiä, että laskutus jakautui asiakasyrityksessä usean tilikauden ajalle. Kokonaisuus hämärtyi tämän ansiosta sopivasti, eikä kukaan laskenut käytettyä kokonaissummaa, mikä olisikin varmasti herättänyt vain turhaa melua asiaa tuntemattomissa sidosryhmissä (esim. omistajissa). Kaikki olivat tyytyväisiä, kunhan sovitussa vuosibudjeteissa pysyttiin edes suurin piirtein. Kun tällaiset mammuttihankkeet sitten aikanaan lopetettiin, ne yleensä ”vaiettiin kuoliaaksi”, yrittämättäkään mitata kuinka paljon saatiin aikaan ja millä panoksilla. Kukaan ei varmastikaan olisi saanut todellisista mitaustuloksista kiitosta, ja jokaisella takoi mielessä että ”tämä oli sitten viimeinen kerta tällä tavalla”.

Jokin tunti laskutukseen perustuvassa työskentelyssä kuitenkin rupe si hiertämään, erityisesti tilaajaorganisaatioissa, koska vähitellen tämän kauden aikana nousi esiin voimakas vaatimus ”kiinteähintaisiin” ja ”nopeisiin” projekteihin siirtymisestä. Näin monissa paikoissa tehtiinkin, mutta kaikki ei vielä kukaan lokahtanut kohdalleen. Asiakkaan arvostama joustavuus niiden ihanien pikkujuttujen toteuttamisessa katosi. Kiinteä hintaisuus kasvatti toimittajien riskejä huomattavasti, mutta ei sinänsä

taannut tilaajalle lisää kannattavuutta. Tämä johtui erityisesti siitä, että toimittajan alun perin alakanttiin arvioimien työmäärien täytyttyä heihin iski hillitön halu tinkiä laadusta kaikkialla, missä se suinkin oli tilaajan huomaamatta mahdollista – ja kyllä hän se monessa paikassa oli. Ainoastaan jos toimittajat osasivat reilusti yliarvioida tarvittavan työmäärän, heitä pidettiin ammattilaisina ja toimituksiin oltiin tyytyväisiä. Reilu pelivara takasi jopa mahdollisuuden joustaa siellä täällä pikkuriikkisen ja tuottaa tilaajan edustajille ylimääräistä mielihyvää. Tilajaorganisaatiot eivät osanneet edes epäillä mitään, kun ei vertailukelpoista tietoa vastavista projekteista ollut yleisesti saatavilla. Ainoa mihin toteutunutta työ määrää tai hintaa voitiin verrata, oli saman projektin arvioitu työ määrä ja hinta. Ei lopultakaan kovin loistavaa mittaamista, mutta parempaa sentään kuin ”jura-kaudella”.

”Pimeällä keski-ajalla” päästiin myös tehokkuuden mittaamisessa aimo askelia eteenpäin. Allan J. Albrecht IBM:ltä kehitti 1980-luvun vaihteessa idean ohjelmistojen toiminnallisen koon mittaamisesta asiakkaan esittämien vaatimusten perusteella. Syntyi ensimmäinen ns. toimintopistemenetelmä. Sittemmin siitä on syntynyt eri puolilla maailmaa noin 40 lisää. Suomessa aloitettiin alkuperäisen menetelmän pohjalta kokeilut 80-luvun puolivälissä, mutta sen lukuisien puutteiden ja epämu kavuuksien takia päädyttiin kuitenkin kehittämään oma parannettu versio. Sitä onkin viime vuosikymmenen alusta käytetty melko monessa yrityksessä – paremmalla tai huonommalla menestyksellä. Heti suomalaisen toimintopistemenetelmän käyttöönoton yhteydessä aloitettiin myös yksityiskohtainen projektitietojen kerääminen. Näiden tietojen analysoinnin perusteella menetelmään tehtiin sittemmin pieniä korjauksia, kunnes se vakiintui nykyiseen Experience 2.0

muotoonsa. Kaikki erilaiset toimintopistemenetelmät ovat rivimäärien laskemiseen verrattuna ylivoimaisia ohjelmistojen koon mittaamiseen, joskin vain muutamat tärkeimmät ovat monelle toimialalle ja erityyppisiin ohjelmistohankkeisiin soveltuvia. Suomalainen menetelmä kuuluu näihin tärkeimpiin, kuten myös amerikkalaisten IFPUG 4.0, englantilaisten Mk II ja hollantilaisten Nesma-muunnos. Näennäisestä erinomaisuudesta huolimatta nämäkään menetelmät eivät kuitenkaan ole onnistuneet vakiinnuttamaan asemaansa – aivan parhaita yrityksiä lukuun ottamatta. Syy on niiden monimutkaisuudessa, korkeassa käsitteellisessä terminologiassa ja alun perin tieteellisessä lähestymistavassa. Pystyäkseen laske maan luotettavasti ohjelmiston koon jollakin yllä mainituilla menetelmillä täytyy laskijan ensin koulutautua menetelmän käyttöön parin päivän kurssilla ja sitten vielä hankkia valvonnan alaisena useiden projektien kokemus soveltamisesta. Huh-huh. Kukapa moiseen haluaisi ryhtyä? Varsinkin kun muutakin tekemistä tuntuu aina riittävän.

Parhaat ohjelmistotyötä tekevät organisaatiot onnistuivat jo ”pimeän keskiajan” kuluessa keräämään merkittävän määrän omaa projektitietoa, mutta pelkkä oman tehokkuuden tunteminen ei ole johdon mielestä täysin tyydyttävää. Aina olisi mukava tietää, millainen se on verrattuna muihin – ettei tarvitsisi luulla olevansa muita parempi, vaan voisi todella tietää sen. Kunnollisia yleisiä koke mustietokantoja ei onnistuttu hyvistä yrityksistä huolimatta kokoamaan vielä viime vuosituhannella, joten vertailumahdollisuudet avoimesti julkaistuun tietoon olivat hyvin rajalliset. Moni yritys kyllä lähti tiedonkeruuseen mukaan, mutta koon laskennan monimutkaisuudesta ja edelleen melko vaihtelevista työraportoinnin käytännöistä johtuen korkeinta johtoa ei saatu luottamaan ensimmäisiin

mittaustuloksiin. Johdon todellinen sitoutuminen jäi ensimmäisten epäonnistuneiden kokeilujen ja projekti-päälliköiden ennakkoluulojen vuoksi saavuttamatta. Kaikki olivat kuulleet ”joltakin entiseltä kollegaltaan”, etteivät toimintopisteet sovi ”sellaisen ja sellaisten” ohjelmistojen mitaamiseen, joten turha niitä on enää kokeilla. Samalla tavalla jotkut uskoivat hyvin pitkään, että maapallo on pannukakun muotoinen.

Ohjelmistoprojektien hallinnan ”pimeän keskiajan” päättyessä esitettiin projektien onnistumisesta aivan yhtä karmaisevia lukuja kuin sen alkuvaiheissakin. Kustannus- ja aika-tauluarviot ylitettiin edelleen jopa sadoilla prosenteilla. Suomesta tosin ei ole pitkään aikaan tietoja kerätty, mutta amerikkalaisen Standish Groupin laajaa CHAOS raporttia on siteerattu joka puolella alan nykytilan todellisuena osoittajana. Kuten jo aiemmin todettiin, pelkkä arvion ja toteuman vertaaminen ei kuitenkaan anna täyttä kuvaa. On hyvä muistaa, että toimintopisteillä (fp) mitattuna ohjelmistoprojektin keskikoko on kasvanut viimeisten 15 vuoden aikana selvästi, tuottavuuden (fp/h) parantuessa samanaikaisesti keskimäärin 8 % vuodessa. Jotakin todellista edistymistä siis saavutettiin tälläkin kaudella. Samalla työmäärällä ja samassa ajassa pystytään nykyisin toimittamaan ohjelmistoihin moninkertainen määrä erilaisia toiminnallisia piirteitä verrattuna 80-luvun puoliväliin.

(Kuva 3.)

Tervetuloa vihdoinkin ”nyky aikaan”. Kukapa osaisi sanoa ehdottoman varmasti, ovatko tällä kaudella rakennettavat tietojärjestelmät kuniimpia ja tarkoituksenmukaisempia kuin ennen vanhaan rakennetut, mutta se on melko selvää että ne ovat entistä suurempia ja toiminnoiltaan monimutkaisempia. Projektien hallinnassa täytyy olla entistä ammattitai-

toisempi, sillä suurissa järjestelmä-hankkeissa on myös suuret riskit. Onnistumisesta on tultava nyt sääntö, eikä poikkeus kuten aiemmilla aikakausilla. Siihen on myös loistavat mahdollisuudet – monestakin syystä. Onnistumisista on nyt myös näyttöä! Maapallon eri puolilla on pistetty hihat heilumaan, ja hyvät uutiset tulevatkin pääasiassa Australiasta ja Suomesta. Ne ovat jo pitkään olleet väkilukuun suhteutettuina ylivoimaisesti parhaat maat kokemusista oppimisessa, joka johtaa nyt vääjäämättä haluttuun lopputulokseen eli parempaan ohjelmistoprojektien kannattavuuden ja kustannusten hallintaan.

Kansainvälisen yhteisön pitkäjänteisen työskentelyn tuloksista voidaan nostaa esiin neljä keskeistä saavutusta, joiden ansiosta kaikki aikaisempina kausina suuria ongelmia aiheuttaneet esteet nyt ovat helposti kaadettavissa. Nämä neljä menestystekijää ovat:

- Suuret yleiset kokemustietokannat ISBSG ja Experience
- ISO/IEC 14143 Functional Size Measurement standardi
- KISS FSM menetelmä toimintopisteiden laskemiseen
- southernSCOPE projektienhallintakonsepti



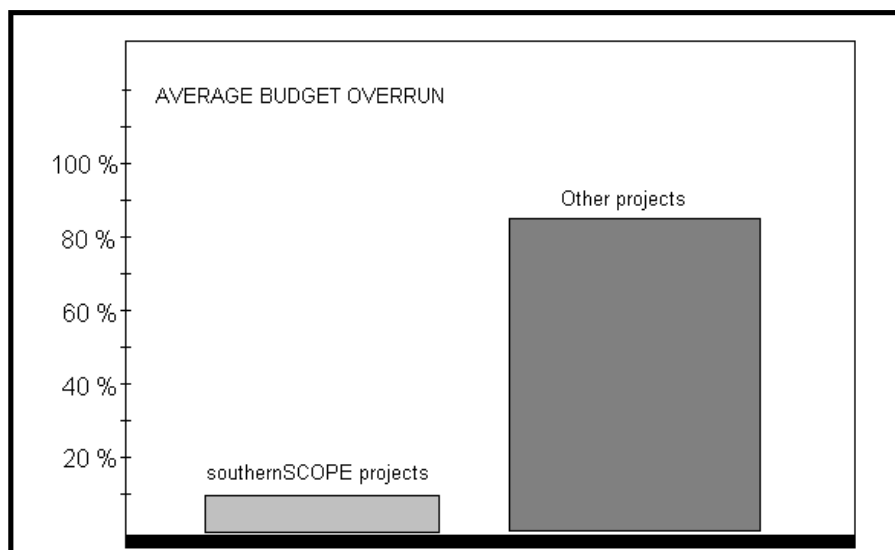
Kuva 3.

Ohjelmistoteollisuutta on puolusteltu takavuosina sen ”nuoruuden” takia, mutta mikäs nuori se viisikymppinen muka on. Hyvässä keski-iässä, sanon minä! Nuorilla ei ole kokemuspohjaa, johon perustaa ammattimaiset työmääräarvionsa, mutta tähän on vihdoinkin tullut muutos. Sekä Australiassa Melbournessa majaansa pitävä ISBSG (International Software Benchmarking Standards Group) että suomalainen FiSMA-yhteisö (Finnish Software Metrics Association) aloittivat kokemustietojen keräämisen jo yli 10 vuotta sitten, mutta viimeisten parin vuoden aikana on molempien tiedonkeruu lähtenyt uuteen nousukiitoon. Australialaisilla on kerättyä jo 1250 projektia eri puolilta maailmaa, yli 20 maasta. Meillä suomalaisilla niitä on tällä hetkellä lähes 500, kaikki kotimaasta paria poikkeusta lukuun ottamatta. Näitä molempia maailman suurimpia tietokantoja on tutkittu ja analysoitu maailman parhaiden ammattilaisten ja tiedemiesten voimin. Tutkimusten tuloksena on löydetty tärkeimmät projektien tuottavuuteen vaikuttavat tekijät, ja mikä rohkaisevinta, molempien tietokantojen perusteella esiin ovat nousseet lähes samat tekijät. Yksittäisten ohjelmointikielten ja mallintamisteknologioiden yli on päästy vihdoinkin. Ne eivät olekaan maailman napa! Ne harvoin ratkaisevat projektityön tehokkuutta enempää kuin muutaman prosenttiyksikön verran. Kaikissa näiden tietokantojen projekteissa ohjelmistojen koko on laskettu toimintopisteinä ja projektien toteutuneet työmäärät kerätty samoja tiukkoja sääntöjä noudattaen. Siksi niitä voi nyt todella käyttää myös arvioinnin apuna! Useimpiin lähtöihin löytyy ainakin jonkinlaista kokemuspohjaista tietoa tuottavuustasosta. Tulevaisuudessa näiden tietokantojen kasvu vain kiihtyy ja hyödyllisyys paranee eksponentiaalisesti. Kun ihminen oppii lukemaan, hänen mahdollisuutensa omaksumaa uutta tietoa moninkertaistuvat.

Koko ”pimeä keskiaika” hapuiliin erilaisten toimintopistemien viidakossa, eikä mittaamista aloittelevien organisaatioiden ollut ollenkaan helppo tehdä valintaa eri menetelmien välillä. Niinpä useimmat jäivät odotelemaan aina näihin päiviin asti. Kansainvälisessä standardointijärjestössä ISO/IEC saatiin vihdoinkin aikaan ja julkaistuksi toiminnallisen koon mittaamisen standardi numero 14143-1 kesäkuun puolivälissä vuonna 1998. Sen jälkeen ovat tärkeimpien käytettyjen toimintopistemien ”omistajat” täydentäneet ja hioneet omia menetelmiään, jotta nämä voitaisiin osoittaa standardin mukaisiksi. Meillä suomalaisilla ei ole tässä suhteessa syytä huoleen, sillä edustajamme ovat olleet standardintyössä mukana alusta alkaen, huolehtien siinä sivussa myös oman Experience 2.0 menetelmämme laadusta ja FiSMA-yhteisön ajan tasalla pysymisestä. Standardointi on vihdoinkin tuonut toimintopisteet samalla tavalla käyttö- ja vertailukelpoisiksi kuin ovat metrit ja jaardit pituuden mittaamisessa. Standardin julkaisemisen ansiosta myös menetelmien soveltaminen on tullut hieman aikaisempaa helpommaksi, kun menetelmäkuvausten terminologiaa on siivottu ja yhdenmukaistettu. Lähes jokaiselle ohjelmistotyypille löytyy jo oma soveltuva toimintopistemelmä, useimmille montakin. Uusien menetelmien kehittämistä ei kuitenkaan ole lopetettu vielä, eikä ole syytäkään. Muiden muassa COSMIC Full Function Point ja FiSMA FSM ovat kiinnostavia lähitulevaisuuden tulokkaita, ja varmasti niidenkin jälkeen tulee uusia.

Suomessa kehitetty ja tänä syksynä kansainvälisillä foorumeilla esitelty KISS-menetelmä on aivan uudenlainen lähestymistapa ohjelmiston koon laskentaan. Siinä ei ole omaa mittayksikköä, mutta sen avulla voidaan laskea kaikkia yleisimpiä toimintopisteitä (Experience 2.0, IF-

PUG 4.0 ja Mk II). Kuten arvata saatetaan, nimi KISS tulee englannin lauseesta ”Keep it simple, stupid”. Sanalla ”stupid” ei viitata tässä menetelmän käyttäjään vaan muiden menetelmien kehittäjiin. On käsittämättöntä, miten vaikeita sääntöjä niinkin yksinkertaisen asian kuin toiminnallisen koon laskemisen ympärille on onnistuttu vuosien varrella kehittämään. Asioita ei vain ole haluttu nimittää niiden oikeilla nimillä. Me suomalaiset emme ole tässä olleet yhtään muita parempia, ennen KISS-menetelmän kehittämistä. Sitä on kehitelty Suomessa useissa eri tilaisuuksissa tammikuusta 2001 alkaen ja tulokset ovat hätkähdyttävän hyviä. Kuka tahansa jolla on ohjelmistomääritykset ja kyky ymmärtää niitä, pystyy laskemaan ohjelmiston koon nopeasti ja tarkasti. Tähänastisten kokemusten mukaan koon laskenta on 4-5 kertaa nopeampaa ja kolme kertaa tarkempaa kuin perinteisillä toimintopistemenetelmillä. On lisäksi muistettava, että kuka tahansa ei perinteisillä menetelmillä saisi mitään järkevää edes viikossa aikaiseksi, vaikka ymmärtäisikin määrittäisi. ”Kolme kertaa tarkempaa” tarkoittaa tässä yhteydessä sitä, että aloittelija tekee helposti ensimmäisissä kokeiluissaan perinteisillä menetelmillä +/- 30 % virheen, kun taas KISS-menetelmällä virhe pysyy yleensä +/- 10 %:ssa. Nämä kokemukset ja menetelmän todellinen helppous ovat Suomen lisäksi herättäneet jo kohtalaista kiinnostusta ainakin Englannissa, Ranskassa, Australiassa, Ruotsissa ja Hollannissa, joihin on tähän mennessä lähetetty pyynnöstä KISS-menetelmäkuvaus. KISS Quick –pikalaskentamenetelmä perustuu 28-kohtaiseen kysymyssarjaan ja kertosikä yhteenlaskuun. Menetelmä kehitetty tulevaisuudessa jonkin verran sen mukaisesti, miten sen avulla laskettavat ”emo”-menetelmät kehittyvät.



Kuva 4.

Sokerina pohjalla ”nykyajan” saavutuksista on vielä southernSCOPE-konsepti, joka on kehitetty ja otettu käyttöön muutaman viimevuoden aikana Australiassa, Victorian osavaltion julkisissa tietojärjestelmähankkeissa. Sen kokeileminen aloitettiin vuonna 1996 ja tähän mennessä konseptin mukaisesti läpivietyjä projekteja on kymmenkunta. Useita projekteja on myös parasta aikaa menossa, sillä Victorian osavaltion hallitus on päättänyt että mallia noudatetaan tästä lähtien kaikissa tietojärjestelmähankinnoissa. Tulokset ovat erittäin merkittäviä. On tunnettua, mm. Standish Groupin tutkimusten mukaan, että keskimääräinen budjetin ylitys tietojärjestelmähankkeissa on useita kymmeniä prosentteja, julkisissa hankkeissa keskimäärin 87 %. Tähän mennessä päättyneissä southernSCOPE-projekteissa keskimääräinen ylitys on ollut vain 10 % ja haastateltuani Victorian tietohallintojohtajaa Terry Wrightia hän kertoi, että tulokset liikkeellä olevista projekteista tulevat noudattamaan samaa linjaa. Oheinen Terry Wrightin australialaisessa Software quality & process improvement lehdessä julkaistusta artikkelista poimittu kuva havainnollistaa saavutettua kustannusten hallinnan tasoa.

(Kuva 4.)

Että mitenkö se tehdään? Se on loppujen lopuksi aika yksinkertaista. southernSCOPE sai alkunsa kahdesta keskeisestä havainnosta. Victoriasa huomattiin vuonna 1996 että heillä onkin nyt käytettävissään kohtuullinen kokemustietokanta (ISBSG) ja että toimintopisteiden laskeminen on vakiintunut siedettävän hyvälle tarkkuustasolle. Australiassa on paljon sertifioituja toimintopisteiden laskijoita. Koska näin oli, päätettiin että hankkeiden hinnoittelussa siirrytään muilta teollisuudenaloilta tuttuun hinta/tuotettu yksikkö-malliin. Rakenusallahan tarjoukset annetaan useimmiten mk/m² ja moottoritiehankkeissa mk/km. Niinpä southernSCOPE-projekteissa tarjoukset pyydetään A\$/fp pohjaisina. Tällä on onnistuttu välttämään sekä tunti-laskutuksen että kiinteähintaisen mallin pahimmat haitat ja saatu sekä toimitaja että varsinkin tilaaja kiinnittämään huomiota skoopin muutoksiin. Uudet toiminnalliset piirteet kasvattavat aina ohjelmiston kokoa ja sen myötä kokonaishintaa. Yksinkertaista, mutta nerokasta. Hinnoittelumallin ohella toinen merkittävä yksityiskohta konseptissa on Scope Managerin palkkaaminen projektiin heti tarjouspyyntövaiheessa. Tämä skoopin hallinnan ammattilainen on sitten projektin aikana projektipäällikön apuna muutamaa eri otteeseen, yhteen-

sä ehkä muutamia kymmeniä tai korkeintaan satoja tunteja, mutta ilmeisesti tällainen investointi on osoittautunut kannattavaksi. Kokonaisuudessaan southernSCOPE-konsepti muodostuu kahdeksasta askelmasta, jotka vaikuttavat ensi näkemältä hyvin loogisilta ja selkeiltä. Tätä konseptia kannattaa varmasti laajentaa muuallekin, vaikkapa Suomeen heti ensi tilassa. Meillä asia on sikälikin ajankohtainen, että tämän vuoden loppuun mennessä valmistuvat sekä tilaajille että toimittajille tarkoitetut yhteiset Tietotekniikan liiton ohjeet/suositukset tietojärjestelmätyön hankintaa varten. Niihin pyritään parhaan mukaan adoptoimaan myös kultajyvät southernSCOPE-konseptista. Lisäksi vuonna 2002 tullaan järjestämään Suomessa ensimmäiset Scope Manager –koulutukset. Sen sortin ammattilaisia tulevaisuudessa varmasti tarvitaan!

Summa summarum, pakko on sanoa että kehitys kehittyy. Me suomalaiset emme ole taaskaan jääneet tuleen makaamaan, vaan rynnistämme etulinjassa tai heti sen takana aina kulloisenkin tilanteen mukaan. Jos ja kun haluamme säilyä eräänä maailman kilpailukykyisimmistä kansantalouksista, meidän on osattava kannattavuuden arviointi ja kustannusten hallinta vähintään yhtä hyvin kuin kilpailijamaamme. Mieluummin vähän paremmin!

*Pekka Forselius,
STTF Oy,
pekkaf@sttf.fi*

Systemityön trendit taktisella aikavälillä

Per Silfver,
Finncompass Analysis Oy

Compass Consulting Group on kansainvälinen IT:n tehokkuuden analyysiin ja kehittämisen konsultointiin erikoistunut yritys. Seuraavassa on tarkasteltu systemityön trendejä vuosien 1996-2000 aikana tehtyjen tutkimusten pohjalta.

Aineisto käsittää noin 60 asiakkaan 200 tutkimusta. Tarkastelussa on käytetty yritysten keskiarvoja. Vaikka hajonta on yleensä suuri, keskiarvoja pidetään hyvin suuntaa antavina. Compass laskee yrityskohdattaiset luvut painotettuina keskiarvoina. Systemityössä se on yleensä sovellusten tai projektien koolla painotettu keskiarvo. Tutkimusten luvut pohjautuvat vuoden aikana valmistuneisiin projekteihinsa ja kaikkiin käytössä oleviin sovelluksiin.

Projektityön tuottavuus

Projektityön ja projektien kustannukset ovat kohonneet vuoden

1996 lukemasta noin 3000 markasta toimintopisteitä kohden 5900 markkaan vuonna 2000. Kustannusten nousun syynä on työvoiman kustannusten kohoaminen, tuottavuuden alentuminen uuden teknologian käyttöönottovaiheessa ja muiden kustannusten kuten menetelmä- ja työvälinekustannusten määrän ja osuuden kasvu. (Kuva 1.)

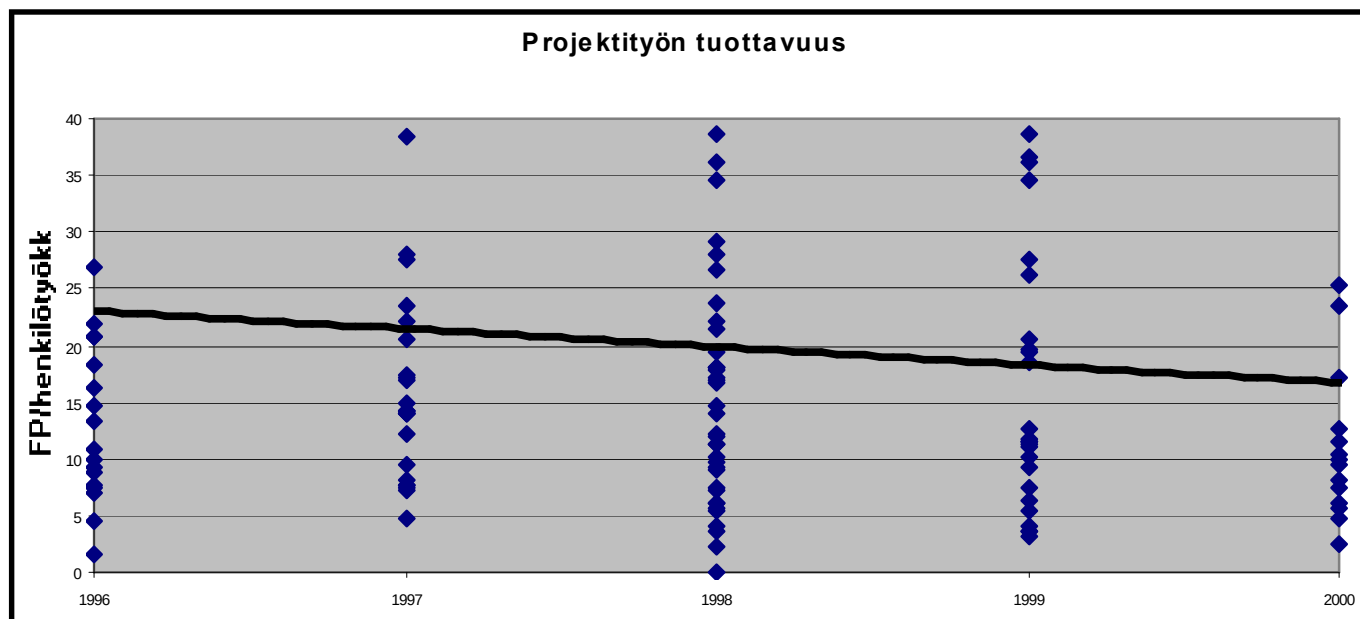
Tuottavuuden kehittymisen suunta tuotettujen toimintopisteiden määränä henkilötyökuukautta kohden on ollut lievästi laskeva vuoteen 1999 saakka. Vuoden 2000 ja 2001 tutkimusten mukaan lasku näyttää päättyneen ja tuottavuus on kääntynyt nousuun.

Hyvä tuottavuus korreloi systemaattisen toiminnan ja vakioitujen menettelytapojen noudattamisen kanssa. Näissä taloissa luovuus käytetään ongelman ratkaisuun eikä menetelmän uudelleen keksimiseen. Projektityön prosessit ovat noin 90 %:ssa tapauksista vahvistettujen menettelytapojen mukaisia. Näistä va-

jaassa 10 %:ssa on projektin ulkopuolinen auditointi ja/tai kytkeä menetelmän kehittämiseen. Valtaosassa nämä menetelmät on kehitetty perinteisten kekkonejärjestelmien rakentamista varten. Uuden teknologian osalta menetelmäkehitys on keskittynyt työvälineiden ja ohjelmistojen tutkimiseen ja valintaan. Prosessin suunnittelu ja ohjeistus sekä kokemusten huomioiminen niissä on resurssiniukkuuden vuoksi jäänyt vähäisemmälle huomiolle. Seuraavat projektit samassa uudessa teknologia-ympäristössä ovat näin joutuneet pohtimaan työprosessiinsa liittyviä ehkä jo kerran aikaisemmin ratkaistuja asioita.

Projektien suunnittelu ja resursointi

Projektin hyvä suunnittelu on oleellinen osa tuottavaa ja kustannustehokasta projektia. Hyvä kommunikointi tilaajan ja käyttäjän kanssa on edellytys realistisen tavoitteen asettamiseksi heti projektin alkuvaiheessa. Usein ryhdytään liian haastavaan



Kuva 1.

hankkeeseen, kun halutaan palvella asiakasta hyvin, vaikka ei ole kokemuseräistä tietoa siitä, mitä hanke todella edellyttää. Seurauksena on hankkeen venyminen tai rajaaminen kesken projektin. Molemmat vaihtoehdot voivat olla asiakkaalle huomattavasti hankalampia hyväksyä kuin alunperin vaatimattomampi tavoite. IT-hankkeeseen liittyy yleensä liiketoiminnan toimenpiteitä, markkinointia yms., jotka on tällöin järjestettävä uudelleen.

Projektin suunnitteluun liittyy myös projektin koon ja työmäärän sekä aikataulun määrittely. Näissä onnistumiseen tarvitaan tavoitteen ja prosessin tuntemisen lisäksi historia-tietoa, mitä on mahdollista tehdä missäkin ajassa. Myöskään projektin riskien analyysiä ei voi liikaa korostaa. Keskimäärin projektit vaativat henkilötyöaikaan 40 % yli suunnitellun ja kalenteriaikaan 20 % yli suunnitellun. Aikataulupoikkeama on keskimäärin 25 % suunnitellusta. Vajaat 10 % projekteista valmistuu etuajassa. Osassa syynä on tehtävän muuttuminen pienemmäksi tai helpommaksi. Pääasiallinen syy on arviointivirhe. Työmääräarvion ylittyminen selittyy tehtävän muuttumisella 10-15 %-yksikön osalta.

Pätevät arviot mahdollistavat osaltaan oikean resursoinnin. Arvioiden epäonnistuminen hankaloittaa toimintaa asianomaisen projektin lisäksi muissa hankkeissa. Aikataulutilanteen selvittäessä tavoiteajankohdan pitämiseen tähtäävät toimenpiteet, lisäresurssien järjestäminen, tehtävien tai tavoitteiden tarkistaminen, lisätuki- ja seuranta-toimenpiteet, muuttavat projektin sisältöä ja rutii-neja, ja ne voivat siten lisätä viivästy-mistä. Resurssien siirtely vaikeut-taa muiden hankkeiden toteutusta ja viivästymisen on pois seuraavista hankkeista.

Projektien kesto on lyhentynyt. Kun se vuonna 1996 oli keskimäärin

360 päivää, niin vuonna 2000 se oli 256 päivää. Tehokkuuden on havaittu parantuvan projektiajan lyhenty-essä. Käytännössä sen tulisi olla enintään 6-8 kk. Pitemmissä hank-keissa riski ympäristön muutoksista kasvaa. Myös projektihenkilöstön motivaatio heikkenee. Usein myös projektipäivää kohden tehtyjen tun-tien määrä on alhainen. Tilanne kor-jaantuu tekemällä projektit mielui-min peräkkäin ja riittävällä panostuk-sella kuin rinnan jakaen niukkoja re-sursseja.

Uusiokäyttö ja ohjelmistogeneraattorit

Tuottavuuden ja kustannuste-hokkuuden parantumista on odotettu ns. integroiduista kehitysmenetelmis-tä ja ohjelmiston generaattoreista ja toisaalta ohjelmakoodin ja kompo-nenttien uudelleenkäytöstä. Yksit-täisten lupaavien hankkeiden läpi-viennin lisäksi muutamat yritykset ovat onnistuneet tässä. Ne ovat mer-kittävien ponnistelujen jälkeen saa-neet menettelyt riittävän laajaan käyt-töön. Osassa yrityksiä prosessi on kesken. Osassa kehittämisohjelmaa on jouduttu tarkistamaan ja tavoite on rajattu pieneen osaan järjestelmiä tai toimintoja. Kokonaistuottavuus ja kustannustehokkuus ei ole parantu-nut.

Ohjelmistokehittimillä, jotka eivät vaadi pitkälle menevää mene-telmistön integrointia, on yleensä saa-vutettu suhteellisen nopeasti tuotta-vuuden paranemista. Eri syistä joh-tuen ne eivät useinkaan riitä pääasi-alliseksi kehittämisvälineeksi.

Time to Market

Kustannustehokkuutta tärkeäm-mäksi on muodostumassa toimitus-nopeus. Tuotteen tai palvelun saa-minen markkinoille ensimmäisenä tai aikaisemmin antaa kilpailuetua.

Keskimäärin projektit ovat tuot-taneet noin 2 toimintopistettä projek-tipäivää kohden.

Ympäristöjen diversiteetti

Teknologian kehittyminen mo-nipuolistanut projektien kehittämis-ympäristöjä. Hankkeissa käytetään useampia laitealustoja, käyttöjärjes-telmiä, ohjelmointikieliä, tietokantoja ja kehittämisohjelmistoja. Yhä har-veimmalla hankkeella on täysin sa-manlainen ympäristötekijöiden ko-koonpano. Myös tuki- ja ylläpito-palvelut pitää varmistaa useamman-laiselle ympäristölle, mikä vaatii re-sursseja ja tuo kustannuksia. Nope-an teknologiakehityksen johdosta käyttöympäristö saattaa vaatia väli-neitä ja ohjelmistoja, jotka käyttöön-sä nähden muodostuvat kustannusra-sitteeksi. Kohdistetaanko näiden kus-tannukset aiheuttamisperiaatteella, harkitaanko niiden uusimista? Vuon-na 1996 projektien lukumäärä ym-päristökokoonpanoa kohden oli noin 3,8, vuonna 2000 noin 3,5. Ylläpi-dettävien ja tuettavien sovellusten määrä vuonna 1996 oli 4 ympäristö-kombinaatiota kohden, vuonna 2000 enää 2,5. Tämä kehitys vaikuttaa työn tuottavuuteen ja kustannustehokkuuteen heikentävästi. Kaikki ym-päristön komponentit vaativat omat kustannuksensa ja asiantuntemuksen ylläpidon.

Sovellusten tuki- ja ylläpito

Ylläpitotyön tuottavuutta mita-taan tarvittavana vuotuisena henkilöt-yönä 10 000 toimintopistettä kohden. Tässä ylläpitotyöhön luetaan korjaa-va ylläpito, tuki ja neuvonta sekä tek-ninen, teknisestä ympäristöstä johtu-va kehittäminen. Kun sovellusten ylläpito vuonna 1996 vaati 10 000 toi-mintopistettä kohden 3 henkilöä, tar-vittiin vuonna 2000 jo yli viisi hen-keä. Tämä johtune osittain tarkaste-luajankohtaan osuneista mittavista Y2K- ja Euro- hankkeista. Niiden työmäärä on pyritty luvuista puhdis-

tamaan. Välillisesti ne ovat kuitenkin lisänneet tuki- ja korjaustarvetta.

Vuosi 2000 aiheutti myös sovelluskannassa merkittävää nuorentumista. Kolmen ensimmäisen käyttövuoden aikana sovellusten vaatima ylläpitotyö on suurimmillaan.

Sovellusten ja ylläpitotyön laatu on kehittynyt suotuisaan suuntaan. Vuonna 1996 laskettiin ilmoitettujen virheiden määräksi 38 tuhatta toimituspistettä kohden. Vuoden 2000 luku oli 32. Näistä noin 40 % johtaa koodin korjaamiseen, muut selvitetään ohjeistuksella ja menettelytapamuutoksilla. Ilman Y2K- ja Euro- muutoshankkeita kehitys olisi ollut vielä positiivisempaa.

Laadun parantamiseksi tehokain tapa on ylläpidon versiointi. Uudet versiot tuotetaan suunnitelmal-

lisesti. Toteutukseen ja testaukseen sekä käyttöönoton edellyttämään ohjeistukseen, tiedotukseen ja koulutukseen varataan riittävä aika. Myös käyttäjät ovat todenneet menettelytavan hyväksi, vaikka uudistukset saatavat näin vähän viipyä.

Systeemityö organisaatiossa

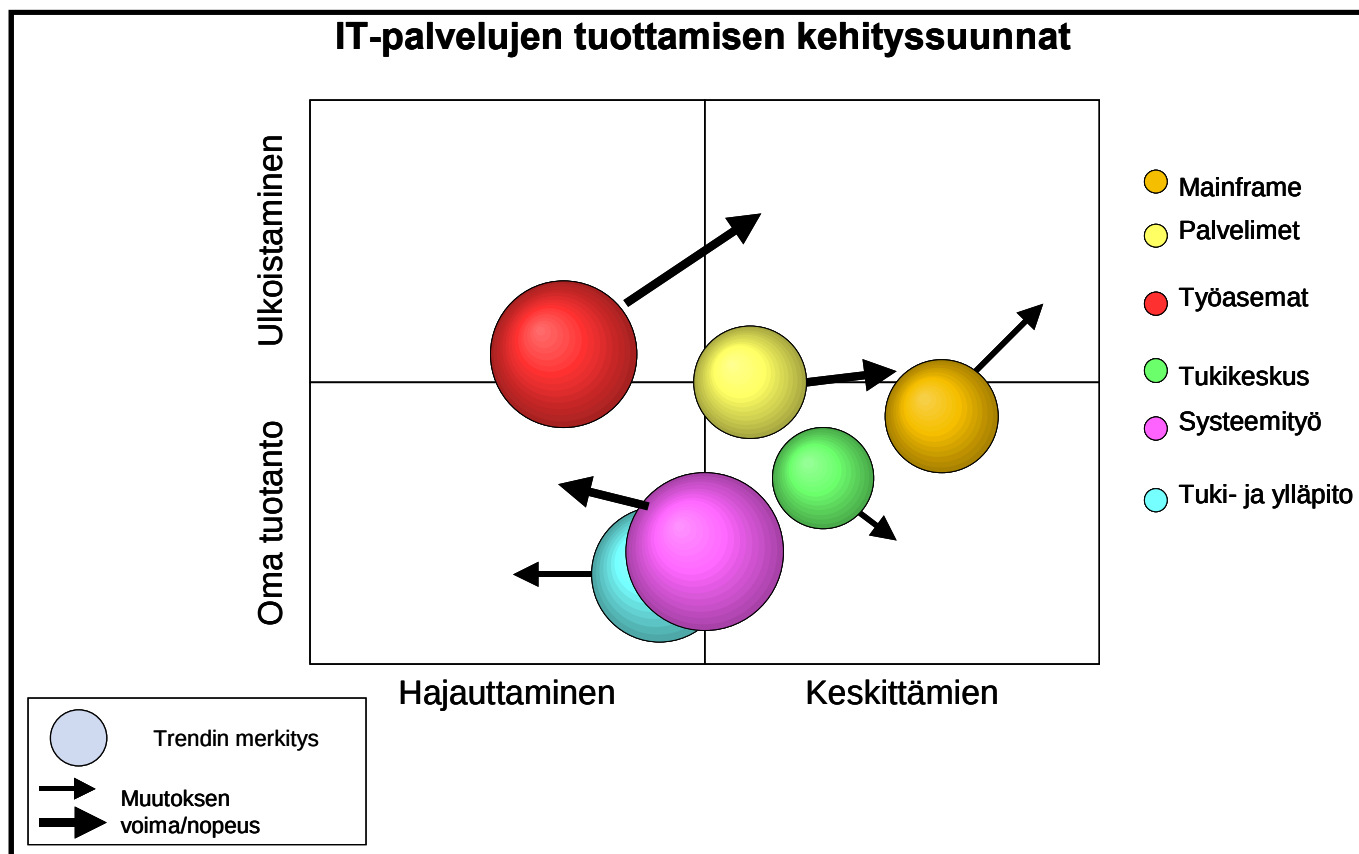
Varsin yleisenä suuntauksen systeemityö on muodostanut oman organisatorisen kokonaisuutensa tavallisesti tuotantoyksikön rinnalla. Näistä IT-yksiköistä on muodostunut omia yhtiöitä konserniin tai sen ulkopuolelle. Niitä on ryhdytty hoitamaan kuten liiketoimintaa. Tuottavuuden ja kustannustehokkuuden merkitys on kasvanut.

Systeemityön hinnoittelu on ollut pääasiassa tuntipohjaista. Tehokkuuden yksiselitteinen arviointi on

sen pohjalta osoittautunut vaikeaksi. Kokolaskelmia on eräissä tapauksissa ryhdytty käyttämään palvelun hinnoittelussa tai työmäärän muutosten selvittämisessä.

Varsin laajamittaisesti on kuitenkin nähtävissä toisensuuntaista kehitystä. IT-kehitys halutaan kytkeä tiiviimmin liiketoimintaan. Systeemityö hajautuu liiketoimintoihin. Liiketoimintaorganisaation rooli kasvaa kehittämisprosessin alkupään tehtävissä ja ylläpitotoiminnoissa. Liiketoimintojen IT-yksiköt lisäävät omaa kehittämistoimintaa ja järjestelmähankintaa.

*Per Silfver,
Finncompass Analysis Oy,
per.silfver@compass.mc,
www.compassmc.com*



Kuva 2.

Tule mukaan vauhdin hurmaan...

**DT-kerho järjestää
jäsenilleen jo perinteiset
Karting-kisat**

Paikka: FormulaKeidas, Juvanmalmi,
Juvan Teollisuuskatu 12,
02920 Espoo



Aika: Sunnuntai 16.12.2001 klo 14-16
Paikalla oltava viimeistään **15 minuuttia ennen.**

Mukaan ajamaan mahtuu 20 nopeinta ja innokkainta, jos paikkoja jää
voidaan mukaan ottaa ulkopuolisiakin.

Osallistuminen ajoihin on jäsenille 50 mk, muille 150 mk.

Ilmoittautuminen:

Olli Laukkaselle 10.12.2001 mennessä puh. 050-5202 313

(Mielellään tekstiviestillä, jossa oltava osallistujan nimi!!!)

Tarkemmin ko. paikasta löytyy www-osoitteesta:

[HTTP://WWW.FORMULAKEIDAS.FI](http://www.formulakeidas.fi)



**Muistathan käydä tutustumassa
Systeemityöyhdistys SYTYKE ry:n kotisivuihin:**

www.pcuf.fi/sytyke