

Hyvä systeemyön osaamisalueet -teemanumeron lukija!

Kädessäsi oleva Systeemyö-lehti käsittelee systeemyön osaamisalueita monesta näkökulmasta. Mukana on kirjoituksia systeemyön osaamisalueiden syvimmästä olemuksesta, eräitä osa-alueita käsitteleviä kirjoituksia sekä näkemyksiä systeemyöosaamisen parantamisen ja todentamisen menetelmistä. Enemmistö kirjoittajista on tällä kertaa systeemyön tutkijoita, kouluttajia, kehittäjiä ja muita systeemyöosaamisen broukeireita.

Juhani Iivari esittelee tieteellisissä artikkeleissaan käyttämäänsä viittä systeemyön tietämysperustaa: teknologia-, sovellusalue-, organisaatio-, sovellus- ja systeemyöprosessitietämys. Ainakin teknologia- ja systeemyöprosessitietämykseen meillä systeemyöoläisillä on yhteinen kieli. Mutta eriytykö kiinnostuksemme muilla tietämysalueilla, koska työskentelemme erilaisten toimialojen ratkaisujen parissa?

Työstä ja sen tekijöistä kuitenkin paljolti riippuu, miten kehittämämme ratkaisut käyttäjänsä hyödyttävät. Kuluvan syksyn risteilyseminaarimme pääpuhuja **Markku Nurminen** pohtii hyötykäyttöä ja sitä, miten hyöty saadaan tietojärjestelmään ja siitä ulos.

Asiakas- ja käyttäjätyöskentelyn läheistä asiaa on paljon myös **Markku Tukiaisen** artikkelissa. Jäävätkö vaatimusten keruu, analysointi ja hallinta liian vähälle huomiolle niin käytännön systeemyössä kuin sen koulutuksessakin?

Eri tahojen ja ihmisten huomion ottaminen systeemyössä on myös eettinen kysymys. Tätä aspektia valottaa Tietotekniikan liiton etiikan työryhmän jäsen **Kai Kimppa** artikkelissaan.

Teknologia- ja systeemyömenetelmäosaamiset eivät siis riitä tämän päivän



“Kukkaispojat” Esko ja Markku.

systeemyöhaasteissa. Selvän viestin tähän suuntaan antoi myös toukokuussa tehty **jäsenkysely**, jonka tuloksia esitellään tässä lehdessä. Suosituimmissa systeemyön osa-alueissa käyttäjien edustajien rooli on mitä keskeisin. Miten systeemyöoläisen ja käyttäjän vuorovaikutus onnistuu?

Systeemyöosaamisen kehittäjistä ja välittäjistä puheenvuoron lehdessämme saivat yliopistoväen lisäksi myös ammattikorkeakoulun (**Eija Kalliala**), osaamisverkoston (**Paula Männistö ja Timo Varkoi**), koulutusyrityksen (**Päivi Hietanen**) ja kattojärjestöme Tietotekniikan liiton (**Pentti Saastamoinen**) edustajat. Tarjolla on mielenkiintoisia näkemyksiä systeemyön verkko-opiskelusta, ohjelmistoprosessien ja -projektien standardointi- ja kehittämisverkostossa toimimisesta, osaamisen kehittämispalveluista ja systeemyöosaamisen sertifioinnista.

Toimituskunta jalkautui tällä kertaa itsekin toimittajan rooliin. Kävimme haastattamassa ammattitaidon MM-kisojen järjestäjiä. Millaisen kuvan MM-titteleistä kilvoittelevat tulevaisuutemme toivot saavat tietotekniikasta, millaista osaamista kisoissa heille korostetaan?

Jokaisessa lehdessä esittäytyy yksi yhdistyksemme kerhoista. Tällä kertaa vuoro on relaatiotietokantakerho RELAILA, jonka toimintaa esittelee kerhon vetäjä **Marja Kärmenniemi**. RELA-kerhon esittäytymistä jämäköittää **Ari Hovin** ja **Antoni Wolskin** raportti kerhon suorittamasta tutkimuksesta relaatiokantojen oliopiiirteistä.

Relaavan antoisia lukuhetkiä lehdessä parissa toivottavat

*Esko Marjomaa & Markku Niemi,
toimituskunta*

Akateemisia aivoituksia systeemityöstä, sen tutkimuksesta ja koulutuksesta

Juhani Iivari,
Oulun yliopisto

Johdanto

Olen koko 30-vuotisen akateemisen urani aikana ollut kiinnostunut ja tavalla tai toisella tutkinut systeemityötä. Jouduin menneenä kesänä miettimään useassa roolissa systeemityötä - yhden väitöskirjatutkimuksen esitarkastajana ja myös oman tutkimukseni vuoksi. Parhaillaan on myös menossa tutkinnonuudistus takaisin kaksiportaiseksi. Sen yhteydessä pitäisi pohtia uudestaan, mitä painottaa systeeminsuunnittelijoiden yliopistokoulutuksessa.

Useat tutkimukset väittävät, ettei systeemityömenetelmiä käytetä käytännössä. Kuitenkin törmään joka päivä ohjelmistoihin ja tietojärjestelmiin kaikkialla. Niiden kehittäminen on luonnollisesti vaatinut systeemityötä. Onko niiden kehittämisprosessi ollut täysin *ad hoc* vai onko siinä ollut jonkinlaista systematiikkaa ja metodiikkaa? Ihmiset käyttävät kehitettyjä ohjelmistoja ja järjestelmiä eri toiminnoissaan sekä työssä että vapaa-aikana. Järjestelmien laatu koskettaa entistä useampia ja yhteiskunnan toiminta on kriittisesti riippuvainen niiden toiminnasta. Tietystikään useimmat ohjelmistot ja tietojärjestelmät eivät ole täydellisiä, mutta iänikuinen puhe niiden huonosta laadusta, ohjelmistokriisistä ja valtaosin epäonnistuneista projekteista (The Standish Group, 1995) on harhaanjohtavaa (Glass, 2000), turhaa vaatimattomuutta ja itse asiassa jonkinsorttista sadomasokismia. Itse asi-

assa kun ajattelen ohjelmistojen luonnetta, niiden monimutkaisuutta, mielivaltaisten liityntöjen määrää, vaikeaa hahmotettavuutta ja kaikkia niihin tehtyjä muutoksia (Brooks, 1987) sekä ohjelmistojen diskreettiä luonnetta (Parnas, 1985) olen aina positiivisesti hämmästynyt, että ne pyörivät sittenkin – pääosin vähintäänkin tyydyttävästi.

Kokonaisuutena systeemityö on varsin vaativaa työtä. Nopeasti kehitetyn teknologian lisäksi systeeminsuunnittelijoiden on ymmärrettävä kehitettävän ohjelmiston tai järjestelmän toimintaympäristö. Järjestelmät voivat tehdä mahdolliseksi uudet tavat toimia. Systeeminsuunnittelijoiden olisi kyettävä näkemään teknologian luomat uudet toimintamahdollisuudet. Heidän on tunnistettava kehitettävän toiminnan ja sidosryhmien vaatimukset, luotava yhdessä sidosryhmien kanssa vaatimusmäärittely kehitettävälle systeemille, toteutettava järjestelmä, ja minusta heidän tulisi tuntea myös vastuuta järjestelmän käyttöönotosta. Systeemityön painopiste on myös siirtynyt organisaatioiden sisäisistä tietohallinto-osastoista markkinavetoisempaan toimintaan - ohjelmisto- ja järjestelmäkehitysprojekteja toimittaviin sekä ohjelmisto- ja ohjelmistointensiivisiä tuotteita kehittäviin yrityksiin. Entistä enemmän systeemityö, ainakin tietojärjestelmien kehitys, perustuu eritasoisin valmis- ja puolivalmisohjelmistoihin, vaatien niiden integrointia ja sovittamista. Suuret ohjelmistokehitysprojektit tehdään myös usein maantieteellisesti, jopa globaalisti hajautettuina, mikä luon-

nollisesti monimutkaistaa systeemityötä.

Quo vadis systeemityö?

Systeemityön hallitsemiseksi on kehitetty useita erilaisia lähestymistapoja, joiden kehityksessä voidaan havaita mielenkiintoinen dialektiikka järjestelmien mallintamista korostavien menetelmien ja erilaisten prosessimallien välillä. Järjestelmien mallintamisessa voidaan erottaa toimintosuuntautuneet rakenteelliset menetelmät, jotka ilmestyivät 1970-luvulla, tietosuuntautuneet menetelmät 1980-luvulla ja oliosuuntautuneet menetelmät 1990-luvulla. Prosessimallien kehitys voidaan tiivistää vesiputousmalliin 1950-luvulta, prototyyppiin ja evolutionaariseen kehittämiseen 1970-luvulta, spiraalimalleihin 1980-luvulta, samanaikaiseen kehittämiseen 1990-luvulta ja ketteriin menetelmiin 2000-luvulta.

Tämä historia herättää useita kysymyksiä. Ensinnäkin onko sattumaa, että kutakin keskeistä muutosta järjestelmien mallintamisessa on seurannut muutos prosessimalleissa? Jos ei, onko mallintamislähestymistapojen ja prosessimallien vuorovaikutus yhteensopivaa vai ristiriitaista? Itse olisin taipuvainen näkemään, että vesiputousmallin jälkeen vaihtoehtoiset prosessimallit ovat pyrkineet näkemään systeemityön nopeampi tempoisena ja joustavampana prosessina kuin mallintamislähestymistavat, jotka korostaessaan ohjelmistojen ja järjestelmien perusteellista mallintamista väistämättä pyrkivät olemaan suhteellisen raskaita. Tässä suhteessa uu-

det prosessimallien aallot herättävät *déjà vu*-tunteen, että olen nähnyt tuon jo aikaisemmin, vai onko kysymys vain iän mukanaan tuomasta kyynisyydestä?

Kysymystä, minne systeemytö on kehitymässä, voi lähestyä monesta näkökulmasta. Yksi tapa olisi koettaa ennakoita systeemytömenetelmien ja lähestymistapojen uusia aaltoja. Oliosuuntautuneet menetelmät ja ketterät menetelmät tuskin ovat lopullinen vastaus. Toinen mahdollisuus on arvioida, miten nähtävissä olevat muutokset teknologiassa ja liiketoimintaympäristössä vaikuttavat systeemytöön. Esimerkiksi Lyytinen ja Rose (2004) esittävät että Internet-ympäristö muuttaa oleellisesti systeemytötä ja raportoivat, että myös käytännön ihmiset kokevat näin. Itse näen, että on vielä liian aikaista arvioida esimerkiksi Internet-ympäristön vaikutusta systeemytöön, koska olemme vielä varsin varhaisessa oppimisvaiheessa. Tässä esityksessä otan lähtökohdaksi ajatuksen, että systeemin suunnittelijoiden yhteisö ei ole ainoastaan passiivinen sopeutuja ympäristön vaatimuksiin, vaan se voi, ja itse asiassa sen tulisi, myös vaikuttaa systeemytöikäntöihin.

Miten kehittää systeemytötä professionaalisemmaksi?

Viime vuosikymmen oli ohjelmistotuotantoprosessimallien aikakautta. Uskottiin, että kun ohjelmistotuotantoprosessia kehitetään "kypsemmäksi", myös ohjelmistotuotteiden laatu paranee. Tämä suuntaus unohti kuitenkin ihmisen, ohjelmistosuunnittelijan, ohjelmistotuotantoprosessin keskeisenä tekijänä. ACM:n ja IEEE:n käynnistämä ohjelmistotekniikan (Software Engineering) professionalisointihanke voidaan nähdä vasteena tähän, pyrkimyksenä panostaa ohjelmistosuunnittelijoiden valmiuksiin. Hankkeen yhteydessä luotiin eettinen koodi ohjelmistoammattilaisille (SWECOE, 2000) sekä mää-

riteltiin ohjelmistotekniikan tietämysperusta (body of knowledge) (SWE-BOK, 2000).

Hankkeen tarkoituksena oli kehittää ohjelmistotekniikasta professio, joka olisi verrattavissa perinteisiin professioihin kuten lääkärit, juristit ja insinöörit akkreditointi-, sertifiointi- ja lisensointikäytäntöineen. Kysymys on siis vallasta, oikeudesta harastaa kyseessä olevaa ammattia. Näin ollen professionalisointi on monimutkainen yhteiskuntapoliittinen prosessi, joka koskettaisi muun muassa työmarkkinasuhteita. Nyt näyttääkin siltä, etteivät edes hankkeen käynnistäjätahot pidä sitä realistisena. Tästä huolimatta voidaan kysyä, etteikö olisi syytä kehittää systeemytötä professionaalisempaan suuntaan määrittelemällä systeemytön tietämysperusta ja kehittämällä alan koulutusta vastaamaan kyseistä tietämysperustaa.

Viimeaikaisissa artikkeleissani olen pyrkinyt hahmottelemaan systeemytön, erityisesti tietojärjestelmä-asiantuntijoiden, tietämysperustaa (Iivari et al, 2004), joka on laajempi kuin SWE-BOK. Erottelimme siinä viisi tietämysaluetta: teknologiatietämys, sovellusalue-tietämys, organisaatiotietämys, sovellustietämys ja systeemytöprosessitietämys.

Teknologiatietämyksellä viit-taamme laitteisto-, tietoliikenne- ja ohjelmistoteknologian hallintaan. Se luonnollisesti on tärkeä systeemytös-sä. Toisaalta se on hyvin nopeasti muuttuvaa ja osittain vaikeasti enna-koitavissa. Tämä luonnollisesti tekee teknologiatietämyksen opettamisen ongelmalliseksi. Missä määrin paino-tetaan olemassa olevan teknologian hallintaa ja välittömiä käytännön val-miuksia tässä mielessä ja missä mää-rin pyritään opettamaan teknologiaa periaatteellisemmalla tasolla, minkä toivotaan edesauttavan uuden tekno-logiatietämyksen omaksumista tule-vaaisuudessa.

Sovellusalue-tietämys viittaa oh-jelmiston tai tietojärjestelmän sovel-lusaluetta koskevaan tietämykseen. Esimerkiksi kehitettäessä laskenta-toimen tietojärjestelmiä, systeemin-suunnittelijoiden on hallittava laskenta-toimen käsitteistöä ja periaatteita. Curtis et al. (1988) havaitsivat klas-sisessa tutkimuksessaan ohuen sovel-lusalue-tietämyksen yhdeksi systeemi-työn ongelma-alueeksi. Tietojärjestel-mien laajentuessa kattamaan lähes kaikki inhimillisen toiminnan alueet, sovellusalue-tietämyksen kattava ope-tus on mahdotonta. Käytännössä sen opiskelu on jätettävä kunkin opiske-lijan oman mielenkiinnon varaan ja usein työelämän yhteydessä tapahtu-vaksi. Toisaalta uuden kaksiportaisen tutkinnon tullessa yliopistoihin voi-daan olettaa, että tietojenkäsittely-tieteitä alkavat kasvavassa määrin opiskella sellaiset, joilla on alempi korkeakoulututkinto ja mahdollisesti myös työkokemusta aivan muulta alalta. Tämän systeemin suunnittelijoiden monitaustaisuuden voidaan toivoa myös lisäävän innovatiivisuutta kehitettäessä uusia sovelluksia ja avaavan aivan uusia ohjelmistoliike-toiminta-alueita.

Systeemytö tapahtuu yleensä jossakin organisaatioympäristössä ja ohjelmistoja kehitetään tukemaan or-ganisaatioiden ja organisaatiomuodostelmien toimintaa. Sen vuoksi *organisaatiotietämyksen*, mukaan lu-kien liiketoimintatietämyksen, voi-daan myös olettaa olevan hyödyllistä systeemin suunnittelijoille.

Sovellustietämys viittaa tietä-mykseen tyypillisistä sovelluksista annetulla sovellusalueella, niiden raken-teesta, funktionaalisuudesta ja käytöstä. Se sisältää tietämyksen kos-kien mahdollisuuksia tukea organi-saatioiden ja organisaatiomuodostel-mien toimintaa annetulla sovellusalu-eella.

Lopuksi, *systeemytöprosessitietämys* viittaa systeemytössä käytet-

täviin työväliseisiin, tekniikkoihin, menetelmiin, lähestymistapoihin ja periaatteisiin kattaen sekä projektienhallinta- että varsinaiset suoritusprosessit. Kehikon mukaan systeemityöprosessitietämys on siis vain yksi osa systeemin suunnittelijoiden tietämysperustaa. Kysymys on kuinka tärkeä ja keskeinen se on käytännössä ja kuinka paljon siihen tulisi panostaa systeemin suunnittelijoiden koulutuksessa, kun otetaan huomioon rajallinen aika, joka on käytettävissä koulutukseen esimerkiksi yliopistotasolla?

Systeemityöprosessitietämys ja systeemityö

Kansainvälisesti verrattuna useissa suomalaisissa yliopistoissa korostetaan systeemityöprosessitietämystä tietojenkäsittelytieteiden opinnoissa. Systeemityömenetelmien käyttöä koskevissa tutkimuksissa on kuitenkin havaittu, että monet organisaatiot väittävät, etteivät ne käytä mitään systeemityömenetelmiä (Chatzoglue ja Macaulay, 1996; Hardy et al., 1995) tai, jos käyttävät, käyttö on hyvin soveltavaa ja valikoivaa. Merkitseekö tämä, että systeemityöprosessitietämystä korostava opetus on virhepainotus?

Menetelmien käyttöä koskevia tutkimuksia voidaan osaltaan selittää menetelmäkäsittelyn ja niiden käytön moni-ilmeisyydellä. ter Hofstede ja van der Weide (1992) esittävät, että systeemityömenetelmät sisältävät ajattelutavan, mallintamistavan, työskentelytavan, projektinohjaustavan ja työkalutuen. Hyvin usein ainakin menetelmien opetuksessa ajatellaan, että menetelmien ydin on mallintamisformalismissa, kuten UML, yksityiskohtaisissa työskentelyaskeleissa ja niitä tukevissa työkaluissa? Jos ajatteleminen, että menetelmät sisältävät myös ajattelutavan, niiden käyttö voi olla paljon implisiit-

tisempää. Menetelmä voi sisältää ns. konstitutiivisia sääntöjä, esimerkiksi olioajattelun periaatteet, jotka määrittelevät milloin noudatetaan olioajattelua. Onhan tunnettua, ettei olio-ohjelmointikielen käyttökään takaa, että toteutuksessa todella noudatetaan olioajattelua saati sitten suunnittelussa. Menetelmä voi myös sisältää regulatiivisia sääntöjä, esimerkiksi, joilla vain säädellään toimintaa (vrt. liikennesäännöt). Monet systeemityöstandardit ovat esimerkkejä tällaisista regulatiivisista säännöistä. Menetelmät voivat myös toimia resursseina, esimerkiksi suunnitelmina tai vain muistilistoina, jota tukevat toimintaa, mutta eivät määrää eivätkä säätele sitä. Systeemityömenetelmä voi myös kuvata ainoastaan ideaalisen prosessin, jota ei oletetaakaan koskaan noudatettavan käytännössä. Siitä huolimatta se tarjoaa mielekkään tavan puhua systeemityöprosessista. Esimerkiksi systeemityöprosessi voidaan täysin mielekkäästi hahmottaa lineaarisen vesiputouksmallin mukaisesti – vaatimusmäärittely, suunnittelu, toteutus, käyttöönnotto – vaikka tuskin mikään systeemityöprosessi todella etenee sen mukaisesti. Lopuksi menetelmä voi toimia systeemityötä koskevan oppimisen apuna, “teorian” tai “hypoteesina” tehokkaasta systeemityöstä, johon systeemityökokemusta voidaan verrata ja sitä kautta oppia.

Edellisen perusteella suhtaudun moniin systeemityömenetelmien käyttöä koskeviin tutkimuksiin hienon skeptisesti, pohtien, että mahtavatko ne tavoittaa menetelmien ja niiden käytön moninaisuutta. Toisaalta ainakin yliopistotasolla minusta on myös tärkeää, ettei menetelmien opetus jää pelkäksi mallintamisformalismin, yksityiskohtaisten työskentelyaskelten ja työkalujen mekaaniseksi opettamiseksi, vaan menetelmien taustalla oleva ajattelutapa tulee myös esille.

Systeemityömenetelmien käyttöönnotto

Vaikka lähes kymmenen vuotta vanhoja, edellä mainitut systeemityömenetelmien käyttöä koskevat tutkimukset kuitenkin viittaavat, että menetelmien ja niihin liittyvän systeemityöprosessitietämyksen siirto käytäntöön on ongelmallista. Ne eivät kuitenkaan selitä menetelmien hyväksyntää, niiden käyttöä tai käyttämättömyyttä. Riemenschneider et al. (2002) havaitsivat yhteen yritykseen rajautuneessa tutkimuksessaan, että menetelmän hyödyllisyys työssä oli tärkein menetelmän käyttöä koskevia aikomuksia selittävä tekijä. Lisäksi menetelmän pakollisuus ja sosiaalinen paine, esimerkiksi työtoverit, vaikuttivat merkittävästi käyttöaikomuksiin. Oliomenetelmien käyttöönnottoa koskevassa tutkimuksessaan Hardgrave ja Johnson (2003) havaitsivat, että nimenomaan menetelmien organisaationaalinen hyödyllisyys, ts. missä määrin systeemin suunnittelijat kokevat menetelmän hyödylliseksi organisaatiolle, selittää tilastollisesti merkittävästi heidän aikomusta käyttää menetelmää, kun taas menetelmien henkilökohtainen hyödyllisyys ei osoittautunut merkittäväksi. Sosiaalinen paine myös vaikutti siihen, missä määrin vastaajat kokivat menetelmät organisaationaalisesti hyödyllisiksi, muttei suoraan käyttöaikomuksiin. Toisaalta menetelmien helppo opittavuus ja helppokäyttöisyys myös selittivät koettua organisaationaalista hyödyllisyyttä.

Molemmat tutkimukset selittävät menetelmien käyttöä yksilötasolla, yksittäiset systeemin suunnittelijat havaintoyksikköinä. Kokonaisuutena ne molemmat korostavat kuitenkin menetelmien käytön sosiaalista luonnetta, menetelmien pakollisuutta, sosiaalista painetta ja organisaationaalista hyödyllisyyttä. Etelä-Afrikassa toteu-

tetussa tutkimuksessa (Iivari ja Huisman, 2001) tutkimme organisaatio-kulttuurin vaikutusta systeemityömenetelmien käyttöön ja hyväksyntään. Siinä organisaatiokulttuuria analysoitiin tietohallinto-osastojen tasolla "Competing Values" -mallin avulla. Malli erottelee neljä kulttuurityyppiä: ryhmäpainotteinen kulttuuri, hierarkkinen kulttuuri, innovatiivinen kulttuuri ja rationaalinen kulttuuri. Ryhmäpainotteinen kulttuuri korostaa työyhteisön hyvinvointia, hierarkkinen kulttuuri toiminnan määrämuotoisuutta ja järjestelmällisyyttä, innovatiivinen kulttuuri luovuutta ja rationaalinen kulttuuri tehokkuutta ja tuottavuutta. Tutkimuksessamme havaitsimme, että systeemityömenetelmien käyttö ja hyväksyntä liittyivät positiivisesti hierarkkiseen kulttuuriin, mutta rationaalisen kulttuurin organisaatioissa tietohallintopäälliköt suhtautuivat hyvin kriittisesti systeemityömenetelmiin.

Tulokset herättävät useita kysymyksiä. Edistääkö hierarkkinen kulttuuri systeemityömenetelmien hyväksyntää ja käyttöä? Voidaanko myös ajatella, että systeemityömenetelmät itsestään ovat hierarkkisen kulttuurin ilmentymiä niin, että systeemityömenetelmien käyttö merkitsee automaattisesti hierarkkisen kulttuurin vahvistumista? Jos näin on, onko niin välttämättä? Voidaanko tietohallintopäälliköiden kriittisyyttä rationaalisen kulttuurin organisaatioissa selittää sillä, että systeemityömenetelmät todella tukevat heikosti ainakin välitöntä tehokkuutta ja tuottavuutta? Useat tutkijat korostavat johdon merkitystä systeemityömenetelmien käyttöönotossa (Roberts et al., 1998). Jos näin on, mitkä ovat mahdollisuudet saada systeemityömenetelmiä hyväksytyiksi rationaalisen kulttuurin organisaa-

tioissa? Edelleen, korostaako systeemityön kasvava markkinavetoisuus nimenomaan rationaalisen kulttuurin arvoja, tehokkuutta ja tuottavuutta, tehden systeemityömenetelmien käytön ja hyväksynnän entistä vaikeammaksi? En yritä vastata esittämiini kysymyksiin. Tutkimus jatkuu, joskus löytyy joku vastauskin, mutta kysymykset eivät lopu. Sellaista on akateeminen elämä.

*Juhani Iivari,
professori,
Oulun yliopisto*

Viitteet

Brooks, F. Jr., No silver-bullet, Essence and accidents in software engineering, *Computer*, April 1987, ss. 10-19
Chatzoglou, P.D. and Macaulay, L.A., Requirements capture and IS methodologies, *Information Systems Journal*, 6, 1996, ss. 209-225

Curtis, B., Krasner, H. ja Iscoe, N., A field study of the software design process for large systems, *Communications of the ACM*, 31(11), 1988, ss. 1268-1287

Glass, R.L., Talk about a software crisis - not!, *The Journal of Systems and Software*, 55 2000, ss. 1-2

Hardy, C.J., Thompson, J.B. and Edwards, H.M., The use, limitations and customization of structured systems development methods in the United Kingdom, *Information and Software Technology*, 37(9), 1995, ss. 467-477

Iivari, J., Hirschheim, R. ja Klein, H.K., Towards a distinctive body of knowledge for information systems experts: Coding ISD process knowledge in two IS journals, *Information Systems Journal*, 2004 (painossa)

Iivari, J. ja Huisman, M., The relationship between organisational culture and the deployment of systems development methodologies, In Dittich, K., Geppert, A. and Norrie, M.C. (Eds.), *Advanced Information Systems Engineering*, Springer-Verlag, Berlin, 2001, ss. 234-250

Lyytinen K. ja Rose, G.M., The disruptive nature of information technology innovations: The case of Internet computing in systems development organizations, *MIS Quarterly*, 27(4), 2003, ss. 557-595

Parnas, D.L. and Clements, P.C., A rational design process: How and why to fake it?, *IEEE Transactions on Software Engineering*, SE-12(2), 1986, ss. 251-257

Roberts, R.K. Jr., Gibson, M.L., Fields, K.T. ja Rainer, R.K., Factors that impact implementing a systems development methodology, *IEEE Transactions on Software Engineering*, 24(8), 1998, ss. 640-649

The Standish Group, *The CHAOS Report*, The Standish Group, 1995, www.standishgroup.com

SWEBOK, Guide to the Software Engineering Body of Knowledge, A Stone Man Version (Version 0.6), February 2000 (www.swebok.org)

SWECO, Software Engineering Code of Ethics and Professional Practice, (www.cs.etsu.edu/seeri/secode.htm)

ter Hofstede A.H.M. and van der Weide, T.P., Formalization of techniques: chopping down the methodology jungle, *Information and Software Technology*, 34(1), 1992, ss. 57-65

Hyötykäytön osaaminen: hyöty tietojärjestelmässä, tietojärjestelmästä ja tietojärjestelmään

*Markku I. Nurminen,
Turun yliopisto, LABORIS*

Johdannoksi

Elämme jälleen hyödyn aikakautta vähän samaan tapaan kuin neljännesvuosituhat sitten. Vaikka valtiotalta ei tällä kertaa katso voivansa edistää hyvinvointia teollisuuslaitoksia ja yrityksiä perustamalla, edistää se toki elinkeinoelämää monin tavoin tutkimuksen ja erilaisten tukiaisten muodossa. Tutkimusrahoituksen painopiste on luonnollisesti hyötypainotteisessa tutkimuksessa ja tuotekehittelyssä. Teoreettisessa ajattelussamme olemme ensyklopedistien jälkeläisiä. Yrityksen sisällä kokoamme tiedot omaan erppiin ja verkossa näemme maailmanlaajuisen ensyklopedian vähitellen kasvavan.

Hyödyn aikakausi oli myös järjen aikakausi, jossa ihmisiä rohkaitiin ajattelemaan itsenäisesti, vapaina auktoriteeteista. Logiikan voima nähtiin suuremmaksi kuin perinteisen vallankäytön. Meidän aikanamme tietotekniikka edustaa tällaista järjen ja sen käytön symbolia. Tälläkin kerralla monet näkevät siinä ihmisten vapauden ja emansipoitumisen mahdollistajan, ovatpa monet olleet löytävinään tietotekniikasta keinotekoista älyä, järkeä, tietoa ja jopa keinotekoista todellisuutta.

Tietotekniikasta säteilevä hyöty saapuu useimpien ihmisten luokse

niin sanottujen tietojärjestelmien välittämänä. Vahvimpana tämä hyöty ilmenee heidän työpaikallaan, vaikka myös monien yksityiselämä näyttää olevan kasvavassa määrin vastaavan siunauksen alaisena. Kun olen jo hyvän aikaa harrastanut työtovereideni kanssa tuon hyödyn metsästämistä, olen kuitenkin keskittynyt ensisijaisesti työorganisaatioissa käytössä oleviin tietojärjestelmiin. Nehän ovat sillä tavalla hyödyn aikakauden lapsia, että niitä ainakin virallisesti pyritään perustelevaan nimenomaan niistä saatavalla hyödyllä.

Hyöty on kuitenkin sillä tavalla ovela asia, että se jotenkin pyrkii pakenevan katsojan silmää. Äskettäin esiin tullut tuottavuusparadoksi on tästä hyvä esimerkki. Yritän tässä kirjoituksessa katsoa missä mielessä hyöty tietojärjestelmässä on ja miten se sieltä voidaan saada houkuteltua ulos.

Taustaa

Työskentelen Turun yliopistossa, sen tietojärjestelmälaboratoriossa, jolla on nimenä LABORIS. Labor on latinaa ja tarkoittaa työtä, IS-lyhenteen taas tunnemme tarkoittavan (paitasi genetiivin päätettä, myös) tietojärjestelmää. Nimensä mukaisesti laboratorion ydinosamisalue sijoittuu sinne, missä tietotekniikka ja työ kohtaavat toisensa. Kohta parinkymmen vuoden ajan olemme tehneet tutkimusta tietojärjestelmistä ja niiden

hyötykäytöstä. Käyttötilanteen ja sen arvioinnin rinnalla tutkimusrepertuaarimme käsittää myös järjestelmien käyttöönottoon ja käytettävyyteen liittyviä teemoja.

Ensimmäinen laajempi projekti (jo ennen Laborista) oli nimeltään ”Tietotyöprojekti” (1985-89), josta kateelliset kollegat joskus unhottivat toisen tavun osoitetta kirjoittaessaan. Suomen Akatemian rahoittaman projektin loppuarviointitilaisuudessa eräs professori totesi, että ”kyllä tutkijoilla oli hirveän hyvä tuuri, kun onnistuivat löytämään niin heikkolaatuisen järjestelmän, että siitä löytyi niin paljon kriittistä sanottavaa”. Hyvä tuuri on jatkunut siitä lähtien lähes keskeytyksettä. Ei siis ole tullut vastaan sellaista järjestelmää, josta ei olisi osattu mitään kehittämiseen tähtäävää kriittistä lausua. Näyttää siltä, että hyötykäyttöongelmat ovat varsin tavanomaisia. Useammin kuin kerran olemme joutuneet puolustamaan ATK-päällikköä toteamalla, että vaikka analysoimamme tilanne johdon näkökulmasta näyttää katastrofilta, niin vastaavia pulmia on lähes kaikissa organisaatioissa.

Havaintomme eivät ole kuitenkaan kokonaan vastoin yleistä käsitystä. Viime syksynä Cap Gemini Ernst Young teki selvityksen, jonka mukaan huomattava osa (mahtoko se olla 8%) suomalaisten tietotekniikan käyttäjien työajasta kuluu hukkaan järjestelmien kanssa mähjätessä. La-

borikselle tämä ei tietenkään ollut mikään yllätys. Joitakin vihjeitä ongelman aiheuttajista voi saada tietojärjestelmätieteen tiedeyhteisön viimeaikaisesta identiteetikriisistä, jossa on suurella joukolla ja tohinalla pyritty määrittämään tieteenalan ydintä. Tämän keskustelun yhteydessä on tapahtunut mielestäni selvää ryhmitymistä organisatorisesti ja sosiaalisesti painottuneeseen itseymmärrykseen entisen teknisen orientaation kustannuksella.

Hyöty tietojärjestelmässä

Hyöty on siitä hankala käsite, että se ei seiso omilla jaloillaan, vaan vaatii rinnalleen vertailukohdan. Tämä on tarpeen ainakin jos pyritään vivahteikkaampaan keskusteluun kuin väittelemään siitä, onko tietojärjestelmästä hyötyä vai haittaa. Heti kun ryhdytään hyötyä mittaamaan, käsitteen takana piilottelee tämä verrokki. Tietojärjestelmän kohdalla vertailupariksi asettuvat luonnostaan tilanne ennen uutta järjestelmää ja sen jälkeen. Saadaanko asiat tehtyä nopeammin, laadukkaammin tai halvemmin vai onnistutaanko nyt tekemään uusia asioita, joihin ei ennen ole pystytty.

Hyötyä ei kuitenkaan ole, ellei joku sitä hyödynnä. Muutoin se on hyötypotentiaalia. Tietojärjestelmän hyöty perustuu luonnollisesti siitä saatavaan tietoon, mutta sen käyttäjät eivät tarvitse tietoa selviytyäkseen tietokilpailukysymyksistä vaan saadakseen työnsä tehtyä. Huono järjestelmä saa työt takkuamaan kun taas hyvä järjestelmä tukee ja sujuvoittaa

tekemistä. Yksi havainto on, että hyödyn ulossaamisen konsti on huonommin tunnettu kuin sen sisään saamisen. Ulosaaminen on kuitenkin mielestäni tärkeämpi, koska jos se jää puuttumaan, on myös sisään laittamisen investointi mennyt hukkaan.

Tuntuu äkkiseltään oudolta ajatella, että järjestelmästä voisi saada ulos hyötyä, vaikka sitä ei siellä alunperin olisikaan. Tähän voin todeta, että käyttäjät ovat kekseliäitä. Monesti he selviävät työstään hienosti juuri siksi että he käyttävät järjestelmää annettujen ohjeiden vastaisesti. Tavallisesti tietenkin tavoitellaan tilannetta, jossa olemassa oleva hyötypotentiaali saadaan täysimääräisenä ulos. Jos tuota potentiaalia ei ole, on tietenkin luonnollista että sitä on työlästä sieltä ulos saada. Päävastustajaksi jää kuitenkin tämän esityksen kannalta tilanne, jossa hyötypotentiaalia olisi, mutta sitä ei vain saada ulos.

Onko sitten tehtävissä jotakin asiantilan korjaamiseksi? On tietenkin. Opetellaan ottamaan paremmin hyötyä ulos ja huolehditaan myös siitä, että järjestelmän rakentajat saavat aikaan sen potentiaalın järjestelmän sisälle. Ensin mainittua olemme Turun yliopistossa pyrkineet kohentamaan antamalla meiltä valmistuneille maistereille sellaista koulutusta, että he saisivat ulosottamisen onnistumisastetta parannettua. Työpaikat ovat kuitenkin pääasiassa tuotantopuolella, missä järjestelmiä tehdään ja hyötypotentiaalia niihin yritetään ahata. Hivenen karrikoiden voidaan ohjelmistoyritysten urasuunnittelua luonnehtia siten, että pannaan uusi

tyyppi ensin viideksi vuodeksi koodaamaan. Jos hän siinä pärjää hyvin, siirretään sitten toisiin tehtäviin. Hyvällä onnella hän silloin voi vaikuttaa jotakin hyötykäyttöongelman ratkaisemiseen - ellei ole sillä välin menettänyt osaamistaan tai motivaatioitaan.

Jonkinasteisena ongelmana näen siten sen, että tietotekniikkaa hyväksikäyttävillä organisaatioilla ei ole tarjolla hyötykäyttöorientoituneita työpaikkoja siitä huolimatta, että sen luulisi olevan hyvinkin kannattavaa. Jos se mähjämisprosentti on vaikkapa vain kahdeksan ja jos siitä saataisiin hyvän osaamisen kanssa pois puolet, niin tietotekniikkaa käyttävän organisaation kannattaisi palkata yksi hyötykäyttöasiantuntija jokaista kahtakymmentäviittä työntekijää kohti. Mikrotukihenkilöt eivät useimmiten kelpaa, heidän osaamisensa sekä toimenkuvansa on kovasti liian kapea ja matala, eikä tätä puutetta korvaa sivutoimisten vastuukäyttäjien panos. Usein ainoat hyötykäyttöasioista kiinnostuneet ovat konsultteja, jotka tavallisesti lähtevät pois, kun alkavat päästä perille organisaation konkreettisesta toiminnasta. Konsulttipanoksessa on se etu, että sen määrä on joustavasti säädeltävissä, mutta se vaikuttaa usein jonkin verran kalliilta resursilta. Konsultoinnin osuus kehittämissuunnitelmien kustannusrakenteessa onkin monissa organisaatioissa viime vuosina kasvanut.

Osallistuva tietojärjestelmien kehittäminen

Tilanne ei ole viime vuosina saattavasti kohentunut. Tähän on osaltaan vaikuttanut ohjelmistotuotannon rakenteiden muuttuminen. Nostalgisesti voimme nimittäin muistella aikaa, jolloin käyttäjät tai ainakin heidän edustajansa osallistuivat järjestelmien kehitystyöhön. Tämä yhteistyö tarjosi yhdenlaisen platformin hyötykäyttönäkökohtien esillä pitämiseen.

Hyötypotentiaali	Saadaan ulos	Ei saada ulos
On sisällä	Tavoite toteutuu	Käytön ja käyttöönoton epäonnistuminen
Ei ole sisällä	Yllätys	Suunnittelun tai toteutuksen epäonnistuminen

Taulukko 1. Hyötypotentiaalın oleminen järjestelmässä ja sen ulossaaminen

On ehkä hyvä palauttaa mieleen osallistuvan suunnittelun kolme päätavoitetta:

1. Osallistumisen kautta pyrittiin saamaan käyttäjien asennoitumista uuteen tietojärjestelmään myönteisemmäksi. Muutosvastarinnan toivottiin tämän kautta loiventuvan.
2. Järjestelmän laadun toivottiin parantuvan siitä, että käyttäjät äänettömän tietämyksensä kanssa olivat mukana suunnittelutyössä. Aina ei äänetön tieto tuottanut uudentyypistä hyvää ratkaisua, mutta usein se onnistui tunnistamaan huonot ratkaisut.
3. Osallistumisen ajateltiin olevan yksi ihmisen demokraattisista perusoikeuksista, erityisesti kun on kyse häneen olennaisesti vaikuttavista asioista. Tutkimuksen kannalta tämä on hyvä reunaehto, mutta vaatii toimiakseen sisältöä ja osaamista, jotka ovat keskeisesti mukana kahdessa ensimmäisessä kohdassa.

Hyötykäytön kannalta lienee tarkoituksenmukaista painottaa tarkastelu kakkoskohtaan, ykköskohta usein nimittäin paranee siinä samalla. Vaikka osallistuminen on Suomessa ollut kenties jonkin verran vaatimattomampaa kuin muissa pohjoismaissa, ovat sen asema ja merkitys käyneet uhanalaisiksi viimeaikaisen kehityksen myötä. Tähän on vaikuttanut ainakin kaksi asiaa, järjestelmätuotannon ulkoistaminen ja valmisohjelmistot.

Ulkoistamisella on tavoiteltu ennen kaikkea kustannusten kuriin saamista. Oman organisaation yksikölle annetusta kustannuksista tai aikataulua koskevasta tavoitteesta oli helpompaa lipsua kuin kahden erillisen yrityksen välisestä sopimuksesta. Samalla suunnittelijoiden tuntuma asiakasorganisaation toimintaan heikkeni ja kanavat äänettömän tiedon välittymiselle ohenivat. Järjestelmän

laatu lepäsi aikaisempaa enemmän ulkoistettujen määrittelyjen varassa. Pitkälle tuotteistetut, usein sellofaaniin käärityt ohjelmat ovat osallistumisen kannalta vielä ongelmallisempia. Käyttäjien osallistuminen on looginen ja käytännöllinen mahdottoisuus. Käyttäjäorganisaatioita kun on (toivottavasti) suuri määrä. Suunnittelija ei voi tuntea tulevia käyttäjiä, eihän tuotetta suunniteltaessa tavallisesti edes tiedetä mihin organisaatioihin tuote onnistutaan myymään.

Tietojärjestelmäkäsitys ja sen merkitys

Vaikuttaa siltä, että hyötykäyttöä ja hyödyn ulosottamista ei pidetä kovin tärkeänä, ainakin aliarvioidaan sen vaatimaa työmäärää, kestoja ja kustannuksia. Laboriksen kokemukset vahvistavat tätä havaintoa. Tuntuu oudolta ajatella, että ensin nähdään runsaasti vaivaa järjestelmän suunnitteluun ja toteutukseen eikä sitten olla kiinnostuneita siitä, oliko tuosta vaivasta hyötyä. Tulee mieleen etsiä selitystä erilaisista käsityksistä siitä mikä tietojärjestelmä oikein on ja millainen on sen suhde työn tekemiseen ja toimintaan.

Tätä pohtiessa tulee mieleen Irmeli Sinkkosen taannoinen aforismi, jonka hän liitti sähköpostiansa signature-tiedostoon:

Käyttäjän työtä ei ole tietojärjestelmän käyttäminen

Se on siitä hyvä lausahdus, että se pakottaa muotoilemaan kaksi jatkokysymystä: mitä se käyttäjän työ sitten on? Sekä miten se työ sitten liittyy näihin järjestelmiin? Se antaa vihjeen myös siitä, että se muu – tai laajempi – työ on jollakin tavalla ensisijainen järjestelmän käyttöön verrattuna. Sieltä pilkottaa myös kannanotto siihen, että järjestelmää ei voi pitää työtoverina ja yhteistyön osapuolena, koska järjestelmä ei yksinkertaisesti

kelpaa toimijaksi. Jos nyt jossakin suunnittelijan oppikirjassa selitetään, miten tehtävät jaetaan järjestelmän ja käyttäjien kesken, onkin ristiriita jo valmis; käyttäjä joutuu vastaamaan tehtävistä, joita suunnittelija kuvitteli antaneensa koneen tehtäviksi.

Mistä sitten tiedetään, mitä kunkin tulisi tehdä työssään? Yksi paljon käytetty työn esitysmuoto on prosessikaaviot. Prosessit jaetaan nyt työntekijöiden kesken. Joustoa tulee siitä kun muistetaan, että yhdellä työntekijällä on tavallisesti vastuullaan monta prosessia ja useita prosesseja suoritettavaan tarvitaan useita henkilöitä. Tämän työnjakopasianessin hallitsemiseksi on käytännöllistä käyttää vaikkapa työroolin käsitettä, jossa työtehtäviä ja prosesseja niputetaan mielekkäiksi kokonaisuuksiksi. Prosessikaavioiden käyttö ei kuitenkaan vielä ratkaise kysymystä siitä, miten tietotekniikka kytkeytyy tähän toimintaan. Askel eteenpäin päästään, kun sijoitetaan tietojärjestelmän avulla suoritettavat tehtävät osiksi prosesseja, jolloin ne samalla jakautuvat eri roolien ja roolinhaltijoiden kesken.

On todettava, että kaikista tietojärjestelmistä ei ole helppo nähdä, millaisia työn tekemisen ja työnjaon ratkaisuja niissä on ajateltu. Olen nähnyt sellaisiakin järjestelmiä, joissa koko kysymystä ei ilmeisimmin ole ollenkaan pohdittu. Eri näytöiltä voidaan tavallisesti suorittaa erilaisia tehtäviä, mikä vaikuttaa jonkinlaiselta työroolin vastuualueelta. Mutta voidaanko tältä pohjalta osoittaa järjestelmän kaikille osille vastuuhenkilöt tai –roolit? Miten määrittyvät roolikohtaiset tietovarastot ja millaiset apuvälineet käyttäjillä on niiden hallintaan? Näitä kysymyksiä hämärtää usein järjestelmän integroitu rakenne, joka myös vaikeuttaa näkemästä järjestelmää kommunikaation ja koordinaation apuvälineenä.

Suuri osa organisaation koordinoinnista tapahtuu tietokonepohjaisen järjestelmän välittämänä. Yksi kirjoittaa tietokantaan ja toinen lukee tämän tiedon myöhemmin. Näin syntyy kommunikaatiosuhde. On harvinaista, että käyttäjät ovat selvillä näin muodostuvasta viestiverkostosta ja sen rakenteesta. Yhteistyökumppanit järjestelmän takana ovat häipyneet näkymättömiin. Asia voisi olla toisin, jos järjestelmän perusyksikkö olisikin toimintaan perustuva, vaikka työroolikohtainen ja tietokantaan kirjautuvaa tietoa kutsuttaisiin viestiksi toisille toimijoille. Jos tietojärjestelmät kerran ovat apuvälineitä työn tekemistä varten, eivätkö ne voisi olla myös näyttää siltä ja olla tuohon tarkoitukseen sopivan kokoisia?

Ratkaisevassa osassa on käyttöönotto, koska sen tuloksena muodostuu ja vakiintuu tuleva käyttötilanne. Jos käyttäjä onnistuu tuottamaan itselleen sellaisen sosiaalisen tulkinnan, jossa ihmiset tekevät työtä järjestelmän avulla ja koordinoivat työssä niinkään sen avulla, niin hallittavuus kohenee nykytilanteen keskiarvoon verrattuna reilusti. Järjestelmätehtävät sulautuvat sisälle prosesseihin ja tietokannan laatu kohenee, kun sen eri osille on nimetyt vastuuhenkilöt. Vaikeus on siinä, että jokaisen käyttäjän on itse tuotettava itselleen tämä tulkinta, muut eivät voi tehdä sitä hänen puolestaan.

Hyötykäytön haasteita

Hyötykäyttö ei synny itsestään. Järjestelmän asentaminen ja testaami-

nen ei vielä riitä, eikä käyttäjien koulutukseen vielä vie maaliin saakka, koska sen sosiaalisen tulkinnan tuottaminen kuitenkin tapahtuu toimintaa tehtäessä ja ajan kanssa. Hyötykäytön tutkimisessa on erityisiä ongelmia. Hyötykäyttöä on esimerkiksi vaikeata havaita, kun kaikki sujuu saumattomasti. Paljon helpompi on havaita hyötykäytön puuttuminen. Ongelmat eivät ilmoita saapumisestaan etukäteen. Ja kun kovin monet asiat voivat mennä pieleen ja jokainen niistä vielä monella eri tavalla, eivät monet ongelmat tule lainkaan havaituiksi tai korjatuiksi. Ongelma on oikea ongelma vain silloin kun sille voidaan esittää jonkinlainen ratkaisu tai ainakin lievennys. Kaiken lisäksi voi löytyä useita ratkaisusuosituksia, jotka taas ovat keskenään ristiriidassa.

Hyötykäyttöä voidaan kuitenkin jäsentää kohtalaisesti. Laboriksella on ollut tässä asiassa käytössä seuraavankaltainen liikesalaisuus: Ensin esitetään organisaatioyksikön toiminta tavalla, joka on riippumaton kulloisestakin tietoteknisestä ratkaisusta. Silloinhan ollaan juuri muotoilemassa vastausta Sinkkosen aforismin kysymykseen käyttäjän työstä. Samalla kirjataan tuon toiminnan tavoitteet. Tämä on tärkeätä, koska ne ovat samat kuin järjestelmän onnistuneisuuden tavoitteet. Tarvitsee vain arvioida, miten hyvin tarkasteltava järjestelmä palvelee tätä toimintaa ja sen tavoitteita.

Sijamuotosulkeiset voidaan tehdä loogisessa aikajärjestyksessä: si-

sään – sisällä – sisältä. Siis hyöty pitää saada järjestelmään sisään ja se pitää saada siellä säilymään ennen kuin sitä voidaan saada sieltä ulos. Tärkeysjärjestys jo todettiinkin tämän kirjoituksen alussa, ja se on päinvastainen. Ulos saatu hyöty on tärkein ja muiden vaiheiden tulee palvella tätä. Nämä kolme sijamuotoa näyttävät usein myös jakavan systeemin ympärillä toimivat henkilöt luokkiin. Voidaan puhua input-käyttäjistä ja output-käyttäjistä. Kiusallisen usein output-käyttäjät ovat erilaisten raporttien vastaanottajia, joiden kautta he valvovat input-käyttäjien toimintaa ja vielä vaativat tunnontarkkaa tietojen kirjaamista. Usein on kuitenkin mahdollista järjestää työroolit sillä tavalla, että kaikki osapuolet pääsevät osallisiksi myös järjestelmän hyödyistä.

Ratkaisematta tämän kirjoituksen puitteissa kuitenkin jää, millä tavalla hyötykäyttöä saataisiin parannetuksi laajemmassa mittakaavassa. Missä saadaan koulutetuksi hyötykäyttöasiamiehiä ja mistä löydetään heille sellaisia työpaikkoja, että heidän työpanoksensa tosiaan parantaisi hyötykäytön tasoa organisaation ja valtakunnan laajuisesti. Toinen ratkaisematon kysymys on, miten näin koHoava hyötykäyttötietoisuus voisi vaikuttaa hyödyn sisään laittamisen puolelle. Voisivatko järjestelmät olla jossakin suhteessa erilaisia kuin tänä päivänä, jos haluttaisiin ja osattaisiin nostaa hyötykäytön tasoa.

*Markku I. Nurminen,
professori,
Turun yliopisto, LABORIS*

Mitä meidän tulisi hallita tulevaisuudessa

Markku Tukiainen,
tSoft - Ohjelmistotuotannon tieto-
keskus,
Tietojenkäsittelytieteen laitos,
Joensuun yliopisto

Ohjelmistotuotannon alalta on viime aikoina kerätty useampia kokoomia tietämystä, joiden tarkoituksena on määritellä alan ydinosaamisalueet. Osittain tämä kehitys on liittynyt tarpeisiin akkreditoida ohjelmistotuottajat ammatillisesti sekä koulutuslaitosten tarpeisiin määritellä (ohjelmistoteollisuuden tarpeisiin sopivia)

ohjelmistotuotannon opinto-ohjelmia. Tunnetuimpia näistä ovat ISO:n ja IEEE:n yhteisprojektina toteutettu SWEBOK (Software Engineering Body Of Knowledge, uusin versio 23.6.2004, www.swebok.org) ja ACM:n ja IEEE:n yhteisprojektina toteutettu, enemmän korkeakoulu- maailmaan suunnattu SEEK (Software Engineering Education Knowledge, uusin versio 21.5.2004, www.acm.org/education/curricula.html).

Kun tarkastelee kokoelmien sisällysluetteloita (kuva 1), niin huoo-

maa, että kokoelmat ovat hyvin samantyyppisiä ainakin sisällysluettelon tasolla. Syvempi tarkastelukin osoittaa, että ydinosaamisen sisältö on hyvin yhtenevä, pieniä painotuksellisia eroja lukuun ottamatta.

Lisäksi monet ohjelmistotyöhön liittyvät standardit ja *de facto* -standardit käsittelevät ohjelmien tekemisen kyvykkyyttä ja ohjelmistotuotannon organisaatioiden kypsyyttä. Tunnetuimmat näistä lienevät SEI:n CMMI (www.sei.cmu.edu/cmmi/) ja ISO:n ISO-15504 (SPICE, www.isospice.com). Ohjelmistotuotannon tietämyskokoelmiin verrattuna tekemisen hallinnan standardit kuvaavat vaatimuksia eri kypsyysoilla oleville organisaatioiden ohjelmistoprosesseille. Tietysti tämä johtuu osittain siitä, että standardeissa halutaan sanoa mitä pitää tehdä, ei miten se pitäisi tehdä. Joka tapauksessa standardeilla on suuri vaikutus alan kehitykseen, joten standardeja elävöittäville selitysteoksille jää tärkeä rooli. Kun yhdistämme ohjelmistotuotannon tietämyksen ja ohjelmistoprosessin hallinnan tietämyksen yhtenäisiksi sisältöaluemäärittelyiksi, meillä näyttäisi olevan melko yksimielinen käsitys siitä, mitä ohjelmistotuottajien tulee osata (henkilöstön kyvykkyys).

Kun tarkastelee ohjelmistotuotannon sisältöaluemäärittelyjä, huomaa selkeitä parannuksia aiempaan vuosiin. Määrittelyt keskittyvät yhä

CHAPTER 2 SOFTWARE REQUIREMENTS
CHAPTER 3 SOFTWARE DESIGN
CHAPTER 4 SOFTWARE CONSTRUCTION
CHAPTER 5 SOFTWARE TESTING
CHAPTER 6 SOFTWARE MAINTENANCE
CHAPTER 7 SOFTWARE CONFIGURATION MANAGEMENT
CHAPTER 8 SOFTWARE ENGINEERING MANAGEMENT
CHAPTER 9 SOFTWARE ENGINEERING PROCESS
CHAPTER 10 SOFTWARE ENGINEERING TOOLS AND METHODS
CHAPTER 11 SOFTWARE QUALITY

Chapter 4: Overview of Software Engineering Education Knowledge
4.8 Computing Essentials
4.9 Mathematical and Engineering Fundamentals
4.10 Professional Practice
4.11 Software Modeling and Analysis
4.12 Software Design
4.13 Software Verification and Validation
4.14 Software Evolution
4.15 Software Process
4.16 Software Quality
4.17 Software Management

Kuva 1. SWEBOK:in (ylempi) ja SEEK:n sisällysluetteloita

enemmän sovittamaan ohjelmiston tuotantoprosessia ympäröivän maailman (tekniseen) monimutkaisuuteen. Esimerkiksi ohjelmistojen elinkaarisista esitetään nykyisin useita malleja, kuten vesiputousmalli, spiraalimalli, erilaiset inkrementaalimallit, prototyyppeihin keskittyvät mallit sekä useita ketteriä malleja (XP, Scrum jne.). Lisäksi korostetaan elinkaari-mallin valinnan tärkeyttä, enää ei sanota yhden koon sopivan kaikille, vaan elinkaari-malli valitaan kehitystilanteen mukaan.

Ohjelmistotyön tukitoiminnot ovat kehittyneet avustamaan monimutkaisten järjestelmien tekemistä. Tuotteen hallinta ja versionhallinta avustavat erilaisten variaatioiden ja kiihtyvään tahtiin tuotettujen versioiden hallinnassa. Muutosten hallinta pitää kurissa vaatimusten muuttumisen kehitystyön aikana ja mahdollistaa järkevän kehitystyön. Projektin hallinta mahdollistaa tiukkojen aikataulujen ja niukkojen budjettien jonkinlaisen pitämisen. Laadunhallinta pyrkii takaamaan järjestelmien luotettavan toiminnan ja havaitsemaan järjestelmään tunkevat virheet mahdollisimman varhaisessa vaiheessa, jolloin niiden korjaaminen on tehokkainta ja edullisinta.

Paljon on siis tapahtunut, mutta mitä vielä puuttuu? Eikö asiat ole hyvin? Kun tarkastellaan julkaisuja, joissa pyritään hahmottamaan ohjelmistotyön ongelmakohtia tai syitä miksi useat kehitysprojektit edelleen myöhästyvät, ylittävät resurssinsa tai epäonnistuvat, eräs useimmin mainituista syistä on asiakasvaatimusten puuttellisuus tai virheellisyys.

Albert Endres ja Dieter Rombach ovat koonneet kirjaansa: "A Handbook of Software and Systems Engineering" joukon ohjelmistotyön empiirisiä, käytännöstä nousseita havaintoja lakien ja hypoteesien muotoon. Kirja on muuten hyödyllinen käsikirja kenen tahansa ohjelmistotuottajan hyllyyn, sillä se kokoaa suuren joukon ohjelmistotuotannossa havaittuja ilmiöitä helposti tavoitettavaan muotoon. Kirjan vaatimusten hallintaa käsittelevässä osassa esitetään mm. seuraavat lait:

1. Glassin laki: Vaatimusten puutteellisuus on tärkein projektien epäonnistumisen lähde.

2. Boehmin ensimmäinen laki: Virheitä tehdään eniten vaatimusten määrittelyn ja analysoinnin aikana ja ne ovat sitä kalliimpia mitä myöhemmin ne korjataan.

Glassin laki tuntuu intuitiivisesti oikealta, sen lisäksi lakia tukee joukko tapaustutkimuksia. Boehmin ensimmäistä lakia tukee useita tutkimuksia jo niinkin varhaiselta ajalta kuin 1970-luvun puolivälistä alkaen. Vaatimusten kerääminen on vaikea asia, koska sidosryhmien tarpeet kehitettävän järjestelmän suhteen vaihtelevat, keskenään ristiriitaiset vaatimukset ovat vaikeasti priorisoitavissa, kaikkien osallistuvien ihmisten tietämys on tavalla tai toisella rajoituttu, ihmiset omaavat paljon vaikeasti esplikoitavaa hiljaista tietämystä jne. Tämä on selkeästi tunnistettu tosiasia, mutta edelleen projekteissa ei tunnuta kiinnittävän riittävästi huomiota vaatimusten keruuseen. Eräässä tutkimuksessa, jossa haasteltiin 15 projektitiimiä yhdeksästä eri yrityksestä tietoliikenne- ja pankki-

sektoreilta, havaittiin parhaimpiin tuloksiin päästävän projekteissa, joissa oli vaatimusten hallinnassa oikea kombinaatio tietämystä, resursseja ja sopiva prosessi. Tietämyksellä tarkoitettiin asiakkaan riittävää osallistumista, kaikki mahdollisten tiedonlähteiden tutkimusta ja kyvykästä osallistujajoukkoa. Resurssilla tarkoitettiin 15 - 30 % projektin kaikista resurssista vaatimusmäärittelyyn. Oikea prosessi oli tavallisesti useamman syklin sisältävä, vaatimusten priorisointiin, jäljitettävyyteen ja validointiin keskittyvä prosessi (Hofmann, H.F., Lehner, F.: Requirements Engineering as a Success Factor in Software Projects. IEEE Software 18, 4, 2001, 58-66).

Samoin kun joukolta ohjelmistotuotannon johtavia tutkijoita (joukon koko 31 tutkijaa) ja joukolta asiantuntijoita (joukon koko 574 jäsentä IEEE:n Technical Council on Software Engineering-ryhmästä) kysyttiin, mitkä ovat tärkeimmät periaatteet (fundamental principles) ohjelmistotuotannossa, suurin ja yksimielisin kannatus oli periaatteella: Sijoita tarpeeksi resursseja ongelman ymmärtämiseen (Invest in the understanding of the problem). Periaatteita löydettiin kaikkiaan 15 kappaletta, muita esimerkkejä periaatteista ovat Rakenna ohjelmistoprosessisi niin, että se on joustava ja Koska muutokset ovat väistämättömiä, varaudu niiden hallintaan (Bourque et al.: Fundamental Principles of Software Engineering - a Journey. The Journal of Systems and Software 62, 2002, 59-70).

Tuntuisi siis selkeältä, että vaatimusten keruuseen ja hallintaan tulisi kiinnittää riittävästi huomiota ohjelmistotyössä. Samoin tuntuisi luonnolliselta, että ohjelmistotuottajien koulutuksessa kiinnitettäisiin paljon huomiota vaatimusten hallintaan.

Kun tarkastellaan miten SEEK-tietämyskokoelma suhtautuu ohjelmiston vaatimusten määrittelyyn ja hallintaan, niin huomataan, että melko vähäisellä panostuksella. SEEK:n Software Modeling and Analysis -osassa esitellään vaatimusten määrittelyä ja hallintaa 16 oppitunnin verran, kun koko ohjelmistotuotannon ydinosaaminen on tarkoitettu 494 oppitunnin kokoiseksi. Siis n. 3 % osaamisesta. Monet Suomen yliopistomaailmasta ohjelmistotuotannon palvelukseen valmistuvat ohjelmistotyön tekijät noudattavat opinnoissaan ACM:n yleistä tietojenkäsittelytieteen opintosuunnitelmaa. Se suhtautuu vaatimusten hallintaan käyttämällä 4 pakollista tuntia suunnitelman totaalista 280 pakollisesta tunnista, eli n. 1,5 %.

Ohjelmisto- ja järjestelmämarkkinat ovat globalisoituneet ja täällä Suomessakin yritykset kehittävät tuotteitaan ja palveluitaan tähtäimenä maailman markkinat. On tunnettu tosiasia, että eri maissa ja mantereilla elävät ihmiset omaavat erilaisen (erilaisia) kulttuuritaustan (taustoja). Silti ohjelmistotyön ydinalueanalyysit ja tekemisen referenssimallit eivät edes mainitse vaatimusmäärittelyosaamisessa ja -prosesseissa sanaa kulttuuri. Esimerkiksi CMMI-malli mainitsee kulttuurin puhuessaan toimittajaorganisaatiosta ja sen vallitsevasta organisaatiokulttuurista, ei asiakasorganisaatiosta ja sen vallitsevan kulttuurin huomioimisesta vaatimusmäärittelyssä.

Toisaalta voidaan tietenkin ruveta pohtimaan, mikä kuuluu ja mikä ei kuulu ohjelmistotuottajan ydinosaamiseen. Asiakasvaatimusten keruun ja analysoinnin, taikka ylipäätään sovelusalueen tietämyksen keräämisen menetelmien osaamisen ei välttämättä katsota kuuluvan ohjelmistotuottajalle. Toisaalta esimerkit joidenkin

opetusohjelmien yleisten runkojen sisältämistä pakollisista tunneista vaatimusten hallinnassa eivät kerro yksittäisten opetusohjelmien tarjoamista pakollisista tai (useammin) vapaaehtoisista vaatimusten hallinnan kurseista. Mutta varmaankin nämä mallit ja osaamisaluekartoitukset kertovat jotakin meidän yleisestä tavasta ajatella ohjelmistotuotantoa ja painotuksia alan sisällä.

Minun on vaikea kuvitella miksi vaatimusten keruun ja analyysin ei katsota olevan nykyistä tärkeämpi osa ohjelmistotuottajan ydinosaamista. Tilanne on samanlainen kuin sanoisimme, että ei lääkärin tarvitse osata tehdä diagnoosia, hänelle riittää parantamisessa tarvittavien välineiden ja menetelmien osaaminen.

*Markku Tukiainen,
tSoft - Ohjelmistotuotannon tietokeskus,
<http://cs.joensuu.fi/tSoft/>,
Tietojenkäsittelytieteen laitos,
Joensuun yliopisto,
markku.tukiainen@cs.joensuu.fi*

Tietojärjestelmätyön eettisiä kysymyksiä

*Kai K. Kimppa,
Informaatioteknologian laitos,
Turun yliopisto*

Miksi meidän tulisi olla kiinnostuneita etiikasta?

Informaatio- ja kommunikaatioteknologian (ICT, Information and Communication Technology) sovellukset kehittyvät yhä kiihtyvää vauhtia. Kaikissa sovelluksissa ei oteta tarpeellisella tavalla huomioon niitä ihmisiä, joiden elämään kyseisillä sovelluksilla on vaikutusta. ICT-kehityksen ongelmia yritetään ratkaista lainsäädännöllä. Kehitys alalla on niin nopeaa, että lainsäädäntö ei ehdi kyydissä. Silloinkin, kun lainsäädäntö saadaan ajan tasalle, se ei välttämättä ratkaise ongelmaa.

Tyypillinen esimerkki (olkoonkin, ettei suoranaisesti systemityön alalta) on elektroninen roskaposti eli spam, jota ainakin minun sähköpostilaatikkooni on tiputellut jo useita vuosia—USA ja EU ovat siitä huolimatta säätäneet lakeja ja direktiivejä asiasta vasta aivan viime aikoina. Sen lisäksi, että lainsäädäntö seuraa IT-alalla yleensä vuosia kehityksen jäljessä, se ei edes ole yhdenmukainen eri maissa. EU:ssa ratkaisuksi roskapostin suhteen valittiin ns. 'opt-in' järjestely, jossa roskapostin potentiaalisen kohteen täytyy erikseen antaa lupa roskapostin lähettämiseen, kun taas USA:ssa ratkaisuksi valittiin ns. 'opt-out' järjestelmä jossa roskapostin kohteen tulee erikseen kieltää roskapostin lähettäminen; ellei hän näin ole tehnyt, saa hänelle lähettää roskapostia kuinka paljon tahansa. Mahdollisuus kieltää lähettäminen täytyy

kuitenkin olla tarjolla jokaisessa roskapostiviestissä. Vaikka molemmat järjestelyt *periaatteessa* takaavat roskapostittomuuden (joskin USA:n valitsemassa järjestelmässä saa ainakin yhden roskapostiviestin), ongelmaksi USA:n ratkaisussa tulee laitton tai muista maista lähetetty roskaposti. Vastaaminen lain ulkopuolelle jääneeseen roskapostiin on yksi varmintapintoja saada lisää roskapostia—jolloin laki toimii roskapostin lähettäjän eikä sen kohteen hyväksi kun lähettäjä näin saa varmistettua että hänen lähettämiään roskaposteja oikeasti luetaan. Myöskään minkäänlaisia kansainvälisiä sopimuksia aiheesta ei vielä ole, jolloin muista maista lähetettyä roskapostia saattaa säädellä minkälainen lainsäädäntö tahansa; tai ei minkäänlainen lainsäädäntö.

Edellä esitetty esimerkki on tyypillinen IT-alan tilanne jossa vanha lainsäädäntö ei säätele uutta toimintaa mitenkään. Joissain tilanteissa vanhaa lainsäädäntöä yritetään soveltaa uuteen tilanteeseen, mutta tästä seuraa helposti kummallisia tilanteita. Tyypillinen tapaus on tekijänoikeus- ja patenttilainsäädännön soveltaminen ohjelmistotuotantoon. Järjestelmien elinikä ei tyypillisesti ole yli 70 vuotta, jolloin niiden immateriaalioikeudet eivät raukea niiden eliniän aikana. Jos immateriaalilainsäädännön tarkoitus on edistää keksintöjen tekemistä, on kummallista, että lait efektiivisesti estävät järjestelmien vapaan edelleenkehityksen kokonaan.

Siitä, että lait eivät jotain kiellä ei (välttämättä) seuraa sitä että sen tekeminen olisi moraalisesti oikein. Uusien keksintöjen ja yhä kiihtyvän

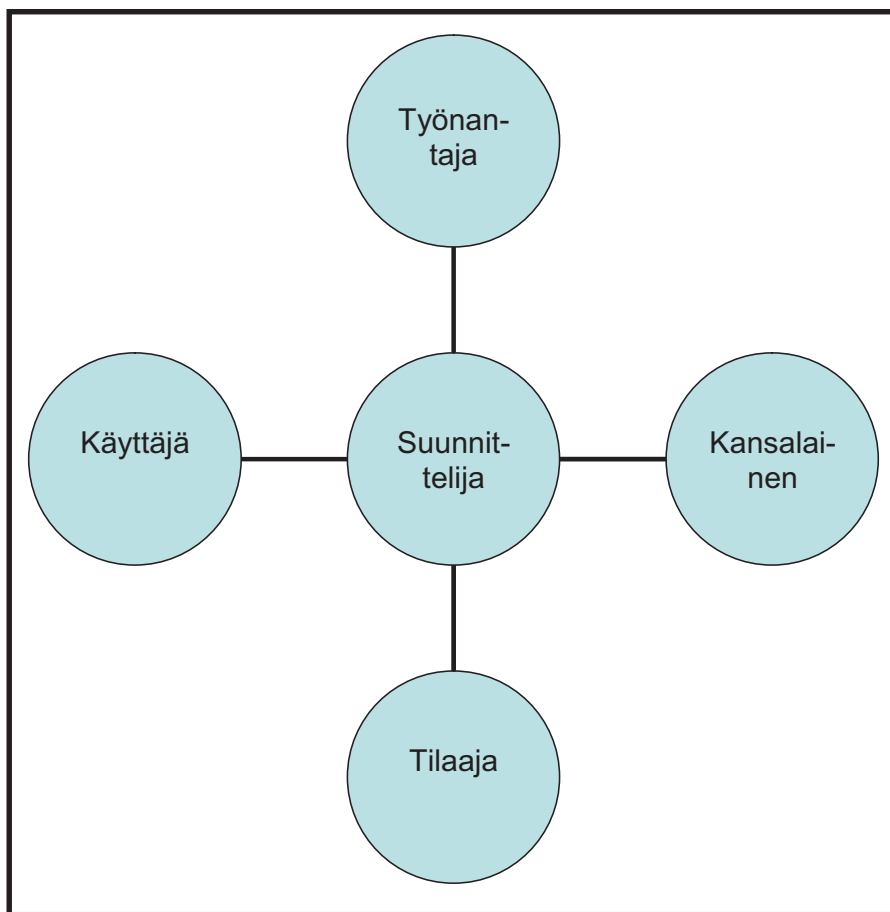
kehitystahdin vuoksi tulee ammattilaisen olla valmis vastaamaan eettiinkin haasteisiin suunnitellessaan ja toteuttaessaan järjestelmiä. Tästä syystä tarvitaan ohjenuora jota seurata kun lakeja ei ole. Tietotekniikan liiton (TTL ry) etiikan työryhmä on laatinut IT-alan ammattilaisille etiikan ohjeet auttamaan tällaisissa kysymyksissä (<http://www.tt-tori.fi/>, Etiikka).

Etiikka järjestelmän suunnittelussa ja toteutuksessa

Seuraavassa eritellään viisi ryhmää (ks. Kuva 1, seuraava sivu) joihin järjestelmäsuunnittelijan ja toteuttajan työ vaikuttaa ja joita kohtaan hänellä on moraalisia velvoitteita. Nämä ryhmät ovat suunnittelija itse, suunnittelijan työnantaja, järjestelmän tilaaja/-omistaja, järjestelmän käyttäjä ja muut ihmiset. Sisäisessä kehityksessä työnantaja ja tilaaja ovat usein sama taho, kun taas ulkoisessa järjestelmäkehityksessä työnantaja ja suunnittelija saattavat olla sama taho. Myös muut ryhmät voivat toistensa kanssa päällekkäisiä lähes millä tahansa yhdistelmällä. Kotikoodaaja voi olla kaikkia viittä samanaikaisesti.

Järjestelmäsuunnittelijalla on itsestään selvästi velvollisuuksia työnantajaansa kohtaan. Suunnittelijalla on eettinen velvoite tehdä työnsä parhaan kykynsä mukaan. Hänen tulee noudattaa toiminnassaan lakeja ja sopimuksia, joihin hän on sitoutunut. Suunnittelijalla on luonnollisesti velvoitteita myös muita ihmisiä kohtaan.

Suunnittelijan velvollisuus on myös tehdä toimivia järjestelmiä.



Kuva 1. Sidosryhmät joihin järjestelmäsuunnittelijan työ vaikuttaa ja joita kohtaan hänellä on moraalisia velvoitteita.

Suunnittelijan ei tule neuvotellessaan järjestelmän toimittamisesta luvata sellaista jonka ei usko onnistuvan tai jonka toteutuksesta sovitussa ajassa ei ole mitään varmuutta. Suunnittelijan täytyy usein tasapainoilla työnantajan ja järjestelmän tilaajan vaatimusten ja tarpeitten välillä. Vaikka työnantajalla on paljon valtaa työntekijänsä toimiin, tulee työntekijän kuitenkin muistaa myös velvoitteensa työn tilaajaa kohtaan. Suunnittelijan tulee parhaansa mukaan pyrkiä varmistamaan järjestelmän vastaavuus sovitun nähdn.

Sen lisäksi, että järjestelmän tulee toimia hyvin siinä tehtävässä johon sen omistaja sen haluaa, järjes-

telmän tulee toimia hyvin myös käyttäjän näkökulmasta katsottuna. Suunnittelijan tulee muistaa, että myös käyttäjä on *arvo sinänsä*, ei vain väline kohti jotain tavoitetta. Suunnittelussa tulee ottaa huomioon myös käyttäjän tarpeet työn sujuvuuden ja miellyttävyyden sekä käytön helpouden ja rasittamattomuuden kannalta.

Suunnittelijan tulee myös muistaa velvollisuutensa niitä ihmisiä kohtaan, jotka eivät ole suoraan järjestelmän kanssa tekemisissä, mutta joita järjestelmä silti koskee. Tällaisia ihmisiä ovat välittömästi esimerkiksi järjestelmän käyttöön ottavan yrityksen asiakkaat. Jos suunnittelija huo-

maa järjestelmän mahdollistavan laittomuuksia tai eettisesti arveluttavia toimia, tulee hänen ottaa asia esiin ja huomauttaa siitä jo suunnittelu- tai toteutusvaiheessa.

Järjestelmillä on tyypillisesti joko suoria tai epäsuoria vaikutuksia meidän kaikkien elämäämme. Esimerkiksi raideliikenteenohjausjärjestelmässä oleva virhe saattaa vaikuttaa kenen tahansa arjessa, oli hän sitten junalla matkustava tai radan poikki kulkeva tai vain kotonaan toista odottava. Onneksi virheet järjestelmissä eivät tyypillisesti aiheuta junaonnettomuuksia, mutta niillä saattaa olla yllättävänkin kauaskantoisia vaikutuksia.

Lopuksi

Muistetaanko kaikki nämä asiat järjestelmiä suunniteltaessa? Tuloksista päätellen vaikuttaisi siltä, että ei; miten muuten selittyy huonojen systeemien yleisyys? Eivätkö suunnittelijat tiedä tekevänsä huonoja systeemejä? Jos tietävät, miksi sitten edelleen tekevät niitä? Tahallaan? Tuskin. Järjestelmäsuunnittelijan tai -toteuttajan niskaan painavat suorituspainheet aiheuttavat usein ylenmääräistä kiirettä suoriutua kaikista tarvittavista toimituksista; eikä hän siksi aina pysty ottamaan kaikkia järjestelmän kohderyhmiä huomioon niin hyvin kuin hänen tulisi. Tulisiko siihen silti pyrkiä? On vaikea nähdä, miten nykyisessä, usein hektisessä kilpailutilanteessa yritys voisi taata suunnittelijalle tarvittavat toimintamahdollisuudet—ainakaan kaikissa tilanteissa. Tarkoitukseensa parhaan mahdollisen järjestelmän tavoittelua ei kuitenkaan tule unohtaa, ja sitä kohti voidaan edetä. Tulee pitää huolta siitä, että huolimattomuudesta ja välinpitämättömyydestä ei tule tapa—eihän vielä ole tullut?



*Kai K. Kimppa,
lehtori,
Informaatioteknologian laitos,
Turun yliopisto*

Kirjallisuutta

Feldman, Fred. (1978), Introductory Ethics. Prentice-Hall, Inc.

Johnson, D. G. (1994) Computer Ethics (2nd ed.), Prentice Hall, Upper Saddle River, NJ, USA.

Johnson, D. G. (2001) Computer Ethics (3rd ed.), Prentice Hall, Upper Saddle River, NJ, USA.

Kant I. (1970) On the Foundation of Morality (editoija ja kommentaarin kirjoittaja: Liddell), Indiana University Press, Bloomington and London. Original Grundlegung zur Metaphysik der Sitten, 1785.

Spinello, R. A. (1995) Ethical Aspects of Information Technology, Prentice Hall, Upper Saddle River, NJ, USA.

EXPERICON

www.expericon.fi

Expericon on kasvava kokoneiden IT-ammattilaisten yhteisyriitys.
Haemme ja tarjoamme asiantuntijoita ja projektipäälliköitä järjestelmähankkeisiin.

Ota yhteyttä!

Experientia omnia vincit

Expericon Oy Satamakatu 2 A 1, 00160 Helsinki puh. (09) 4540 0540 fax (09) 4540 0541 www.expericon.fi

Systemityön eri osaamisalueiden opiskelusta verkossa

*Eija Kalliala,
Helsingin liiketalouden ammatti-
korkeakoulu Helia*

Voiko systemityötä opiskella verkossa? Miten kaaviokuvia välittää verkkon kautta? Miten ”ruuvi opiskelijan päässä kääntyy”, miten ahaa-elämys syntyy ilman lähikontaktia? Entä miten seinätaulut ja palaverit käyttäjien kanssa taipuvat verkkoon tai miten tieto välittyy opiskelijaryhmän jäsenten kesken, jos he ovat eri paikkakunnilla?

Nuo kysymykset risteilivät päässäni, kun minulta kahdeksan vuotta sitten kysyttiin, voisiko oliolähestymistapaa tai systemityön selvitystä ja määrittystä opiskella verkossa.

Sain vastauksen kysymyksiini jo seuraavana syksynä, kun siirsin oliolähestymistavan opetukseni verkkoon. Ensimmäisessä videoneuvottelussa annoin tehtävän: ”Piirtäkää valitsemastanne kuvasta olioluokkakaa-vio UML-notaatiolla.” En ehtinyt kirjoittaa tehtävää ja siihen liittyvää ohjeistusta verkkoon, kun ensimmäiset opiskelijaryhmät jo lähettivät sähköpostin liitetiedostona skannatun kuvan ja siitä piirtämänsä olioluokkakaa-vion. Olin ihastuksesta ällistynyt! Ja tämä tapahtui vuonna 1997.

Verkko-opiskelun mahdollisuuksista

Verkko-opiskelusta on vaihtelevia käsityksiä aina tylsästä yksinpuurtamisesta täsmätietojen etsimiseen hakukoneella ja hedelmälliseen

ryhmätyöhön. Verkko toimii kanavana, joka mahdollistaa entistä monipuolisemman samanaikaisen ja eriaikaisen opiskelun samassa tai eri paikassa.

Verkko-opiskelussa käytetään yleensä verkkosivuja hyperlinkkeineen, hakukoneita, sähköpostia ja keskusteluareenoita, mutta usein myös samanaikaisen vuorovaikutuksen mahdollistavaa verkkojutustelua eli chattia. Nykyisin opiskelussa yleistyy mobiili tiedonvälitys kännyn tekstiviesteistä ja puheluista monipuolisiin kuvan ja liikkuvan kuvan välittämiseen muille opiskelijoille tai oppimismateriaalin tietokantoihin. Vastaavasti mobiililaitteen avulla voidaan etsiä täsmätietoa oppimismateriaalista tai verkon tietokannoista.

Verkko sisältää monipuolisia vuorovaikutusmahdollisuuksia sähköpostista keskusteluareenoihin ja ryhmätyötiloihin, jutustelukahviloihin sekä verkkoneuvotteluihin, jotka perustuvat ääneen, jaettuun sovellukseen ja liikkuvaan kuvaan. Näitä pitäisi käyttää verkko-opiskelussa entistä enemmän, jotta yksinäinen puurtaminen muuttuisi innostuneeksi yhdessä tekemiseksi.

Verkko on pohjaton tietovarasto, josta oikein etsimällä löytää ajantasaista täsmätietoa opinnoissa eteen tulevien tehtävien ja ongelmien ratkaisemiseen. Tietolähteistä voi helposti viestiä verkossa muille opiskelijoille, ja opettaja voi koota niistä jäsennellyn listan verkko-oppimismateriaaliinsa.

Systemityölehden artikkelit ovat erinomaista oppimismateriaalia. Sääli, että ne julkaistaan verkossa noin vuoden viiveellä. Artikkeleissa on usein verkko-osoitteita lisätietoihin sekä kirjoittajan sähköpostiosoitte, joten aiheesta kiinnostunut opiskelija voi halutessaan ottaa yhteyttä suoraan asiantuntijaan.

Systemityön yleisillä keskustelualueilla tapaa alan kansallisia ja kansainvälisiä asiantuntijoita. Heiltä voi saada parempia ja täsmällisempiä vastauksia ja vinkkejä ongelmiinsa kuin vain perehtymällä aiheeseen liittyvään verkon loputtomaan tietomäärään.

Sytyke-yhdistys tarjoaa jäsenilleen Fle3-tiedonrakentelu ympäristöä asiantuntijafoorumiksi ja Tietotekniikan liiton yhdistyspalvelimelle on syntymässä yhä monipuolisempia asiantuntijoiden vuorovaikutuskanavia. Nämä ovat myös liittoon tai Sytykeeseen kuuluvien opiskelijoiden ulottuvilla.

Verkossa materiaalia on helppo kopioida. Opiskelijat voivat muodostaa omia ratkaisujaan esimerkkien pohjalta. Nykyisin opiskelussa olennaista on oppia käyttämään ja soveltamaan tietolähteitä monipuolisesti eikä pelkästään oppia ulkoa faktatietoja, jotka ehtivät vanhentua ennen kuin opiskelija pääsee työelämään niitä soveltamaan.

Monet ajattelevat, että verkko-opiskelu tapahtuu vain verkossa, ja että verkossa viestitään kirjoittamal-

la sanoja näppäimistöllä. Verkko-opiskelussa voidaan verkkomateriaalin lisäksi käyttää perinteisiä oppikirjoja, kasvokkaisia tapaamisia ja perinteisiä tai kannettavia puhelimia. Verkkoviesti voidaan muodostaa paitasi kirjoittamalla näppäimistöllä myös puhumalla mikrofonin.

Systeemyön selvitys- ja määritysvaiheiden opiskelusta verkossa

Systeemyön selvitys- ja määritysvaiheiden opiskelu vaatii abstraktisten käsitteiden ymmärtämistä ja reaali maailman ilmiöiden olennaisten piirteiden kuvaamista sovitulla tavalla. Tutkittava ilmiö voidaan kiteyttää olioluokkamalliin tai käyttötapaumalliin monista eri näkökulmista.

Miten verkko-opiskelijat oivaltavat, että on olemassa muita mallintamiskäsitteitä kuin se, joka heille ensimmäiseksi tulee mieleen? Miten he saavat oppimisessa välttämättömät ahaa-elämykset? Nämä syntyvät parhaiten vuorovaikutuksessa muiden opiskelijoiden ja opettajan kanssa: perehtymällä muiden opiskelijoiden tuotoksiin, vertaamalla niitä omiinsa ja keskustelemalla niistä verkossa avoimesti.

Mallinnus ja sen eri vaiheet voidaan julkaista kaikille opiskelijaryhmille verkossa. Luokassa tämä tehdään seinätauluilla. Verkossa opiskelijaryhmien tuotokset ovat näkyvissä jatkuvasti, samoin kaikki niihin liittyvät kommentit ja kysymykset. Luokassa aika, joka käytetään keskusteluun, on käytännön syistä rajallinen, ja yksityiskohdat jäävät suuressa tilassa havaitsematta. Verkossa kaikki on näkyvissä ja opiskelijat voivat itse valita, kuinka pitkän ajan he käyttävät tuotoksiin perehtymiseen ja niistä keskustelemiseen. Kirjoitettuihin kommentteihin voidaan palata uudestaan, niistä voidaan keskustella ja esittää tarkentavia kysymyksiä verkossa.

Parhaimmillaan tällaisesta verkkovuorovaikutuksesta syntyy monipolvinen ja erilaisia näkökulmia ja ahaa-elämyksiä valottava keskustelupolku, joka voidaan jalostaa opiskelijaryhmän systeemyöprojektin loppuraporttiin.

Puhuen annetut kommentit eivät välttämättä tallennu, kuten ne eivät tallennu luokahuoneessa tai kokouksissa. Käytyjen verkkoneuvottelujen tai kasvokkaisten neuvottelujen lopputulos pitää siis dokumentoida muistioon. Se voidaan julkaista verkossa joko kaikille opiskelijoille näkyväksi tai oman projektiryhmän verkkoalueella, johon muiden opiskelijaryhmien pääsyä on rajoitettu.

Verkkoneuvottelut voivat perustua pelkkään puheeseen tai puheeseen ja kuvaan. Niitä voidaan opiskelijoiden systeemyöprojekteissa soveltaa käyttäjien haastatteluihin, aivoriiheen, tarkistuspisteisiin ja projektikokouksiin, mutta luonnollisesti myös toisten opiskelijaryhmien tuotosten kommentointeihin ja palautekeskusteluihin opettajan kanssa. Neuvottelijat voivat olla eri paikkakunnilla tai ympäri maailmaa, kunhan heillä on yhteys tietoverkkoon ja siellä olevaan yhteiseen neuvottelutilaan.

Puheen mahdollisuuksia verkko-opiskelussa pitäisi käyttää entistä enemmän. Opiskelijoiden selvitys- ja määritysprojekteissa on usein mukana ulkopuolisten yritysten kiireisiä edustajia. Heille olisi helpompaa paneutua mallinnukseen keskittymällä siihen sovittuun aikaan ja esittämällä kysymyksensä ja kommenttinsa suullisesti – sen sijaan, että he perehtyisivät yksinään raskaisiin, ehkä turhankin huolellisesti laadittuihin kuvauksiin ja joutuisivat esittämään kirjallisesti niihin liittyviä kysymyksiään ja kommenttejaan. Kun kysymykset usein poikisivat tarkentavia jatkokeskusteluita, niin kommentointi hajosi yritysten edustajien työkokousten

ja muiden työkiireiden lomaan. Tämä saattaisi heijastua ikävästi kommenttien laatuun.

Yksinkertaisesti verkkoneuvottelu toteutetaan jaetulla kuvaruudulla ja neuvottelupuhelimella. Käsiteltävien kuvausten näkeminen on yleensä tärkeämpää kuin neuvottelijoiden kasvojen näkeminen. Äänenlaatuun on hyvä kiinnittää huomiota: verkkoneuvottelussa ajatukset esitetään puhumalla. Kun perinteisessä opiskelussa ja neuvotteluissa ajatukset ilmaistaan puhumalla, niin miksei verkko-opiskelussa?

Yksinkertaiset verkkoneuvottelut ovat monien työpaikkojen virtuaalitiimeissä jo arkipäivää. Miksei niitä sovellettaisi opiskelijaprojekteissa, joissa opiskelijat valmentautuvat tekemään systeemyötä kansainvälisen työelämän virtuaalitiimeissä?

Kun systeemyön selvitys- ja määritysvaiheiden opiskelussa yhteistyö yritysten ja opiskelijaryhmien välillä toimii, päästään kaikkia osapuolia tyydyttävään lopputulokseen – sekä tietysti kannustaviin opiskelukokemuksiin.

Systeemyön toteutusvaiheen opiskelusta verkossa

Verkko tarjoaa monipuolisia mahdollisuuksia systeemyön toteutusvaiheessa ohjelmoinnin, tietokantojen ja käyttöliittymien opiskeluun. Tukimateriaalina tarvittavat käsikirjat ovat verkossa ja monipuoliset hakumahdollisuudet auttavat täsmätietojen etsinnässä. Verkossa on paljon ohjelmointiesimerkkejä, joihin opiskelijat voivat tutustua. Opettaja voi laatia omaan verkko-oppimismateriaaliinsa esimerkkiohjelmia, joita opiskelijat voivat kopioida, ja perehtyä niiden toimintaan suorittamalla niitä työasemassaan. Luonnollisesti kopioinnissa ja ideoiden soveltamisessa pitää kunnioittaa tekijänoikeuslakia.

Opettaja voi liittää kurssin verkkosivuille monenlaisia teknisiä opiskelua tukevia ohjelmistoja. Ohjelmoinnin opiskelussa voidaan käyttää ohjelmistoja, jotka automaattisesti tarkistavat opiskelijoiden laatiman koodin ja ilmoittavat havaitsemistaan virheistä.

Tietokantojen opiskeluun voidaan liittää verkkolomake, jolla opiskelijat lähettävät SQL-lauseensa tietokantapalvelimelle. Opiskelijoilla ei tarvitse olla tietokantaohjelmistoa omilla työasemissaan, mutta he näkevät välittömästi SQL-lauseidensa tuottamat tulokset. He voivat vertailla saamiaan tuloksia ja keskustella niistä keskenään tai kysellä opettajalta neuvoja ja vinkkejä verkon keskustelualueella.

Käyttöliittymien opiskelussa opiskelijat voivat arvioida verkossa

olevia käyttöliittymiä, keskustella niistä ja soveltaa niistä saamiaan ideoita omiin ratkaisuihinsa.

Kaikessa opiskelussa verkko voi toimia opiskelijaryhmien tuotosten julkaisupaikkana ja keskustelualueena. Ohjelmointitehtävät ovat usein niin monimutkaisia, että opiskelijoiden ratkaisujen julkaisemista varten pitää kehittää parempia vaihtoehtoja kuin verkon keskustelualueet tai perinteiset oppimisalustat.

Muiden opiskelijaryhmien tuotoksiin perehtyminen, niiden vertaaminen omiin ja niistä keskusteleminen syventää oppimisprosessia. Parhaimmillaan tällainen hedelmällinen ja avoin verkkovuorovaikutus johtaa tutkivaan, yhteistoiminnalliseen ja syvälliseen oppimiseen.



*Eija Kalliala,
systeemityön lehtori,
Helsingin liiketalouden ammatti-
korkeakoulu Helia,
Sähköposti:
etunimi.sukunimi@helia.fi,
Kotisivu: <http://myy.helia.fi/~kalei/>*

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

www.sytyke.org

Systemityön osaamiskysyntä muuttuu

Päivi Hietanen,
Tieturi Oy

Vuonna 2002 on tehty tutkimus ICT-alan toimintatapojen muutoksista. Tutkimuksesta ilmenee, että nopeasti muuttuva toimintaympäristö asettaa uusia haasteita alan osaamistarpeille ja niiden ennakoinnille. Liiketoiminnan ohella myös osaamisen kehittämistä on tullut tasapainoilua muuttuvissa olosuhteissa.

Muuttuvassa toimintaympäristössä ei ohjelmistotuotantokaan ole entisensä. Kilpailuetua tavoitellessaan ICT-organisaatiot pyrkivät entistä parempaan kustannustehokkuuteen, asiakaslähtöisyyteen ja jatkuvaan kehittämiseen. Ydinliiketoimintaan keskittyminen, ulkoistus ja liiketoimintastrategiat auttavat tässä. Yksilön kyvykkyyden kehittäminen on osa organisaation keinoja saavuttaa kilpailuetua.

Kilpailu alalla kiristyy

Tilastokeskuksen vuonna 2002 tekemän tutkimuksen [2] mukaan Suomessa ICT-alalla palvelutuotannossa työskenteli lähes 48000 henkilöä. Palvelutuotantoon kuuluvat mm. ohjelmistot ja tietojenkäsittelypalvelut. ICT-alan työntekijöiden määrä ei enää merkittävästi kasva, joten asiantuntijan on koulutauduttava entistä määrätietoisemmin kilpailukykyä säilyttääkseen.

Organisaatiot rekrytoivat mahdollisimman 'valmiita' asiantuntijoita. Arvokasta on kokemus, taito sitoa oppimansa uusi aiemmin opittuun ja kyky soveltaa asioita uuteen ympäristöön. Muuttuneessa markkinatilanteessa talous- ja markkinointitietämys on entistä tärkeämpiä. ICT-alan yritykset tasapainottavat tekniikan osaamista rekrytoimalla kaupallisen alan asiantuntijoita. Kilpailuetua haetaan nyt enemmän yksilöiden osaamisesta, ei niinkään teknologioista kuten joskus oli tapana.

Osaamistarpeiden ennakointi – ad hoc vai strategia?

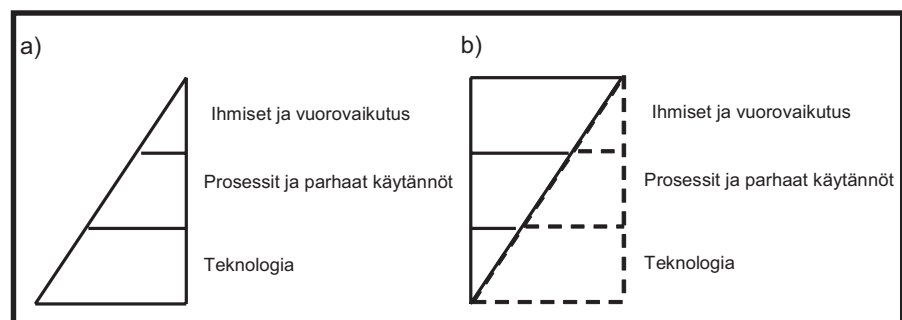
Teknologiaosaaminen yhdistettynä hyvään liiketoiminnan tuntemukseen ja yhteistyötaitoihin on haaste ICT-alalla. Liki 80 % organisaatioista tiedostaa jo liiketoimintastrategian ja pitkäjänteisen osaamisen kehittämisen merkityksen. Liiketoimintastrategiasta osaamistarpeet siirtyvät ihmisten osaamisprofiileihin.

Liiketoimintastrategian perusteella myös rekrytoidaan - vai sanoisiko ostetaan - 'puuttuvia' resursseja.

Turhan usein kuitenkin on niin, että osaamisen kehittämistä ei viedä toimenpiteiksi asti. Tällöin esim. muutokoulutusta tarvitsevat ryhmät jäävät usein huomaamatta. Pitkäjänteistä muutokoulutusta tarvitsevat hankkivat sirpaletietoa eivätkä kykene aina yhdistämään sitä jo opittuun. Oma tarinansa on myös viime hetkellä kurssille ilmoittautuneilla ja niillä, jotka tarvitsevat pikaisesti uutta tietoutta jo alkaneessa projektissa.

Osaaminen levenee ja syvenee

Yksilön kyvykkyydellä on suuri merkitys ohjelmistotuotannossa. Yksilöiden välinen tehokkuus voi vaihdella 1 – 10-kertaisesti. Tehokkaan osaajan profiili on muuttumassa teknologiaosaajasta pehmeämpään suuntaan, mikä korostaa systemityön osaamista entisestään.



Kuva 1. Taitojen tarve a) ennen, b) nyt ja tulevaisuudessa.

Yhden teknologian syvälinen osaaminen ei riitä menestymiseen. Nyt yksilöllä tulee olla laaja perustietämys ja syvällisiä taitoja yhdestä tai useammasta osaamisalueesta konkreettisia projektitehtäviä suorittaakseen. Ohjelmistokehityksen vaikuttajan Scott Amblerin [3] mukaan tällaiset yleisosaaaja-asiantuntijat (generalizing specialist) ovat parhaita tiimin jäseniä.

Tyypillisesti yleisosaaaja-asiantuntijalla on hyvä näkemys ohjelmistoprosessista. Hän pystyy myös tekemään perustellumpia päätöksiä ja kommunikoimaan tehokkaammin kuin erikoisasiantuntijat. Yleisosaaaja-asiantuntija on innokas laajentamaan taitojaan esim. mentoroinnilla.

Ulkoistus tuo mukanaan laajan kumppaniverkoston, jonka toimiminen vaatii hyvää yhteistyökykyä, usein useamman kielen taitoa ja projektityökokemusta. Yhteistyötaitojen, ohjelmistoprosessien ja systeemyön parhaiden käytäntöjen osaamista tulee lisätä (Kuva 1). Tämän allekirjoittavat myös toimittajat, joiden ohjelmistokehitysvälineille MDA (Model Driven Architecture) ja SOA (Service Oriented Architecture) ovat tuttuja.

Asiakaslähtöistä osaamista kysytään

Tietojärjestelmien tulee tukea organisaation liiketoimintaa ja prosesseja. ICT-organisaatioiden onkin osattava tuottaa asiakkaalleen sovitettavia palveluja, ei niinkään standardituotteita, joihin asiakkaan on mukauduttava. Oma lukunsa on toiminnanoh-

jausjärjestelmät, joiden parametroida ympäristöön sopivaksi on rajallista.

Määrittelijöiden on tunnettava riittävästi liiketoimintaa mallintaakseen prosessit. On oltava taitoa selvittää tarpeet liiketoiminnan edustajilta. Määrittelijän on oltava hyvä kuuntelija ja mallintaja. Vuosien saatossa tietojärjestelmäkehittäjien taito tarvekartoittajina on parantunut, vaikka oppimista riittää edelleen.

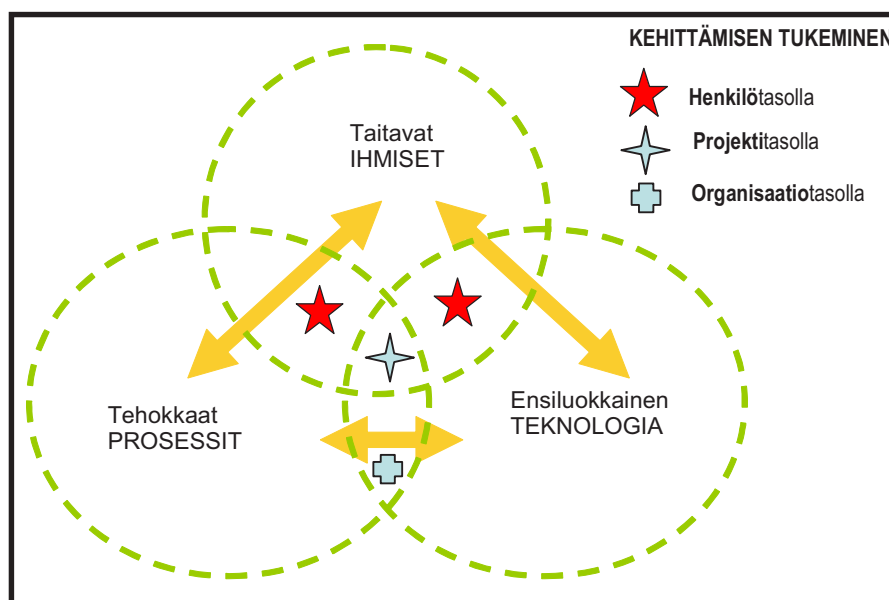
Kauan on Chaos-raporteista [4] voitu lukea, että eräs suurimmista tietojärjestelmäprojektien epäonnistumisen tekijöistä on käyttäjäosallistumisen puute. Asiakkaalle oman roolin ymmärtäminen ei aina ole helppoa. IT-alan ammattilaiset sanovat, että asiakkaan taito ostaa on ratkaisevan tärkeä (ITviikko 15.8.2003). Osaava asiakas tietää mitä tahtoo, osaa arvioida tuotoksia ja testata. Osaava asi-

akas sitoutuu ja tuntee vastuunsa tietojärjestelmäprojektissa. Systeemyön valmistus ulottuu tänä päivänä myös tietojärjestelmän tilaajiin. Oppimiskohteina ovat prosessien kehittäminen ja mallintaminen, tietojärjestelmän hankinta, vaatimusmäärittely ja testaus. Entisajan loppukäyttäjistä onkin tullut aktiivinen vaikuttaja.

Kehittäjät hopealuotia etsimässä

Kursseilla törmää seuraavankaltaisiin kysymyksiin: ”Mikä on paras teknologia, Microsoft vai Java?” tai ”Ovatko oliomenetelmät perinteisiä parempia?”. Kysymyksissä piilee odotus kaiken ratkaisevasta tavasta, hopealuodista.

Jostakin syystä kehittäjien on vaikea hyväksyä, että hopealuotia ei ole eikä tule. Oman päättelyn merkitys muuttuvassa ohjelmistotuotannos-



Kuva 2. Osaamisen kehittämisen tukeminen on monitasoista.

sa unohtuu helposti. Oikein asetettu kysymys onkin: ”Mikä sopii parhaiten näihin olosuhteisiin?”. Tarpeellista onkin kehittää olosuhteiden ymmärtämiskykyä. Myös omien työ- ja kommunikointikäytäntöjen tehokkuutta on punnittava.

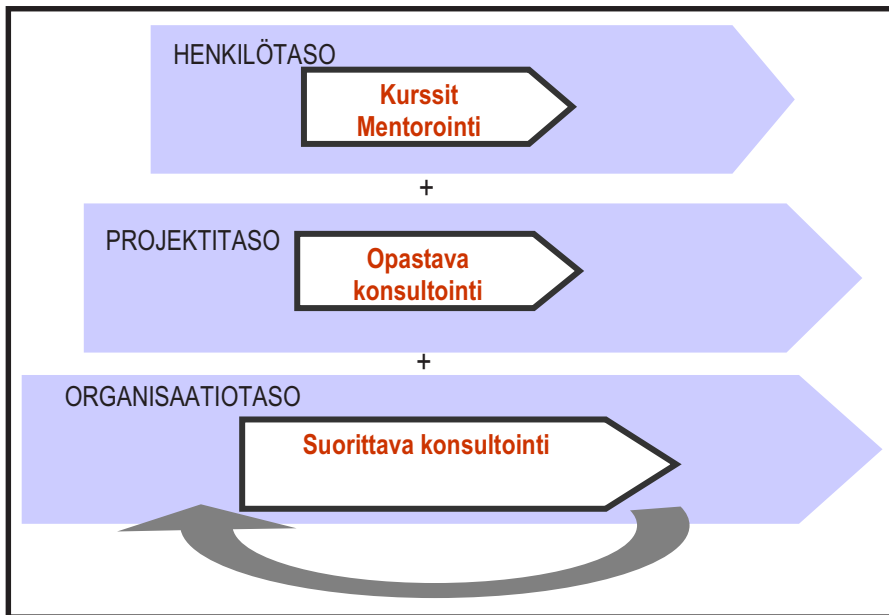
Kehittäjien tulee tuntea paremmin menetelmiä liiketoimintaa tukevan järjestelmän rakentamiseksi. Ratkaisuja on etsittävä muutoksia sietävän ja luotettavan järjestelmän kehittämiseksi. Muutossietoisuuteen keinoja saadaan iteratiivisesta ohjelmistoprosessista ja tilanteeseen sopivasta arkkitehtuuriratkaisusta. Luotettavuutta korostavat mm. aikainen testaus, käytettävyys ja tietoturva. Ohjelmistotuotantoon tuo tehokkuutta mm. yhteinen UML-kuvauskieli ja yhteiset työtavat.

Ketterä ohjelmistotuotanto koetaan usein hopealuodiksi. Ketteryy-

dessä kaiken a ja o on kuitenkin asi-
antuntijan kyvykkyys päätellä tilan-
nekohtaisesti olosuhteista toimintata-
pa. Ketteryys ei siis olekaan yksittäi-
nen tekniikka vaan kokonainen, hyö-
dyllinen ajattelutapa. Edut saavuttaa
vain asioiden luonteen sisäistämällä.

Haasteita kouluttajan näkökulmasta

Muuttuva toimintaympäristö on haasteellinen myös kouluttajalle. Miten antaa parhaat mahdolliset eväät osaamisen kehittämiseen? Joskus tarvitaan apua laajempien ajatusmallien jalkauttamiseksi organisaatioon, toisinaan tukea on saatava esim. prosessien ja teknologian soveltamiseksi projektissa. Kaikissa tilanteissa yksilöiden aiempi kokemus on otettava huomioon. Yksi ratkaisu ei sovi kaikille ja kaikki eivät tarvitse kaikkea (Kuva 2).



Kuva 3. Osaamisen kehittämisessä ja muutoksessa tuki kurssin jälkeiselle työlle on merkittävä.

Osaamisen kehittämisen on tuettava organisaation toimintaa ja erilaisia tilanteita monipuolisesti. Osaamista tuleekin kehittää henkilö-, projekti- ja organisaatiotasolla eri tavoin [5].

Kyky palvella asiakasta kouluttajana, mentoroijana ja oman systeemyöalueensa huippuasiantuntijana ei ole yksinkertainen kaikille sopiva tehtävä. Tästä huolimatta monet organisaatiot ovat taloudellisen taantuman johdosta ajautuneet hoitamaan tehtävää liian pienin resurssein. Tämä ei yleensä palvele riittävästi oppimistavoitteita ja opetettu jää irralliseksi.

Kurssi on oppimisessa jäävuoren huippu. Opin vieminen käytäntöön vaatii yksilöltä sisäistämistä ja toimintatapojen kehittämistä. Osaamisen kehittäminen tulisikin ymmärtää kokonaisvaltaisena ja jatkuvana muutosprosessina, ei pelkästään kurssinä. Muutoksessa tavoite on saada toimintatavat paremmin tarkoitukseen sopiviksi. Tässä olennaista on kussin jälkeiselle työlle saatava jatkuva tuki (Kuva 3).

Roolimuutoksen tukena henkilökohtainen valmennus

Nykyajan ohjelmistotuotanto vaatii muutoksia työrooleihin. Suunnittelijasta tulee arkkitehti, ohjelmiojasta integraattori ja projektipäälliköistä muutosten hallitsija. Kaikille avoimet kurssit palvelevat yksilön perusosaamisen kehittämistä. Kurseilla tutustutaan erilaisiin haasteiden ratkaisuvaihtoehtoihin, verkostoituminen on luontevaa ja kokemusten vaihto tuo uutta perspektiiviä oppimiseen. Varsinainen haaste oppimisessa tulee

kuitenkin vasta kurssin jälkeen, kun oppeja tulisi soveltaa käytäntöön. Tukea tarvitaan myös kurssin jälkeen.

Luokkaopiskelua voidaan täydentää virtuaalisessa oppimisympäristössä. Ennakkotehtävillä ja tukiaineistolla saadaan aiheeseen pidempi orientaatio- tai sisäistämisaika. Samalla yksilön sitoutuminen kehittämiseen kasvaa, opiskelusta tulee monimuotoista ja myös vuorovaikutteista. Motivaatio omaehtoiseen kehittämiseen on osoitettavissa myös kansainvälisillä sertifiointitesteillä.

Työroolin muuttuessa esim. testausvastaavaksi tarvitaan kokeneen asiantuntijan henkilökohtaista tukea. Sopiva työtapana on mentorointi. Lähtökohtana on mentoroitavan omasta työympäristöstään määrittelemä oppimistavoite. Mentorointia ei ratkaise ongelmia vaan auttaa mentoroitavaa löytämään ratkaisun ja tukee uuden oppimisessa ja soveltamisessa. Parhaimmillaan mentorointi on kahden asiantuntijan välistä vuorovaikutusta, josta kumpikin oppii.

Projekteille opastavaa konsultointia

Onnistunut projekti on saavutus sinänsä. Tasapainoilu vaatimusten ja rajoitteiden tikkailu on vaativaa keeneellekin konkarille. Vaikeuskerrotoimia tulee lisää uusista käytäntöön sovellettavista asioista. Varmuutta soveltamiseen voidaan saada työpajassa, jossa riippumaton asiantuntija opastaa uuden soveltamisessa käytännön tilanteisiin.

Alkuun päästyään projektiryhmä jatkaa itsenäisesti, mutta voi tarvittaessa pyytää esim. katselmointiapua tuotoksilleen varmistuakseen

opin perillemenosta ja tuotostensa laadusta. Tyypillinen käytännön esimerkki on vaatimusmäärittelyn työpajan ohjaus ja vaatimuskäsittelyn katselmointi. Myös uutta työvälinettä tai teknologiaa käyttöönotettaessa on systeemyön asiantuntijalta saatu tuki merkittävä, jotta myös varsinainen työprosessi olisi toimiva myös käyttöönoton jälkeen.

Räätälöityä osaamisen kehittämistä

Yksinkertaisimmillaan osaamisen kehittäminen on standardikurssin toteuttamista organisaatiossa sopivana ajankohtana. Toisessa ääripäässä ovat laajamittaiset valmennusohjelmat, perustekniikoiden kouluttamisesta asenteisiin vaikuttamiseen. Väliin mahtuu mentorointia, opastavaa ja suorittavaa konsultointia. Suorittava konsultointi voi olla esim. ohjelmistoprosessin ja käytäntöjen arviointia, menetelmien soveltamisohjeistojen laadintaa tai vaikkapa kuormitustestien tekemistä. Palvelun sisältö ja muodot sovitaan yhdessä asiakkaan kanssa. Usein räätälöity palvelu koostuu erilaisista osista.

Yhteenveto

Liiketoiminta ja sen muutoksia tukeva kouluttaminen on muuttunut kokonaisvaltaiseksi osaamisen kehittämiseksi. Organisaatiot ja yksilöt panostavat aiempaa enemmän osaamisen kehittämiseen kilpailuetua saavuttaakseen. Kapea-alaisesta teknologiaosaamisesta on pyrittävä laajempaan osaamiseen vuorovaikutuksen, ohjelmistoprosessien ja parhaiden käytäntöjen alueilla. Onnistunut osaamisen kehittäminen vaatii monimuotoista tukea kurssin jälkeen.



*Päivi Hietanen,
osastopäällikkö, systeemyö,
Tieturi Oy,
www.tieturi.fi/systeemyo,
paivi.hietanen@tieturi.fi*

Lähteitä:

- [1] Manninen Anneli, Meristö Tarja, Tulevaisuuden ICT-osaaminen – yritysten ja yksilöiden strateginen haaste. Åbo Akademi University, CoFi Report No 1/2004.
- [2] Informaatiosektori toimialoitain 2002. Tilastokeskus 2004.
- [3] Ambler Scott, Isn't that special? Software Development, January 2003.
- [4] Chaos-raportit. <http://www.standishgroup.com>
- [5] Tieturi Frame –osaamisen kehittämisen viitekehys, lisätietoja kirjoittajalta.

FiSMA ry, prosessien ja projektien johtamisen standardointi- ja kehittämisverkosto

FiSMA-verkostossa on mukana nelisenkymmentä suomalaista ohjelmistotyötä tekevää ja palveluja ostavaa yritystä, sekä yliopistoja ja muita julkisen hallinnon organisaatioita. Verkosto on toiminut vuodesta 1992 lähtien ja 8.12.2003 alkaen se on ollut rekisteröity yhdistys. FiSMA ry:n toiminta jakaantuu kahteen jaostoon: Projektijohtaminen sekä Prosessijohtaminen.

Toimintopistelaskennasta projektijohtamista

*Paula Männistö,
TKP Tieto Oy,*

Aloitetaanpa aivan alusta; siis siitä, minkälainen oli ensitapaamiseni FiSMA-verkoston kanssa. Oli vuoden 1998 loppupuolisko ja aloittelin opinnäytetyötäni Heliassa aiheenani ExperiencePro-laskentaohjelmiston ja toimintopistelaskennan käyttö oli projektien työmääränarvioinnissa.

Ajattelin, että kun aiheeseen oikkein perin juurin perehdyn, niin luonnollisesti osallistun myös käyttäjäkerhon toimintaan. Ensimmäiseen koko-

ukseen menoa taisin hieman jännittääkin; huone oli täynnä tummapukuisia keski-ikäisiä miehiä. 'Missä ovat kaikki tämän alan naiset?' ajattelin silloin. Ensimmäisestä kokouksesta minulla ei ole muita muistikuvia. Kovasti paljon en tainnut silloin aiheesta ymmärtää, olinhan vasta FiSMA-urani alkumetreillä.

FiSMA Projektijohtamisen jaostoon kuuluu noin 20 yritystä. Jaosto kokoontuu 4-5 kertaa vuodessa ja kokouksien asialistalla on aina myös ExperiencePron käyttöä ja laskentäsääntöjen tulkintaa käsittelevät kohdat. Nämä keskustelut koen erittäin tärkeäksi anniksi, mitä kokouksista saa. Silloin on mahdollisuus saada apua muilta käyttäjiltä ja asiantuntijoilta, vaihtaa kokemuksia ja mielipiteitä.

Joskus on intouduttu jopa syvälisiinkin keskusteluihin laskennan periaatteista ja tarkoituksesta. Ymmärtääkseni kaikilla meillä syy toimintopistelaskennan käyttöön on se, että pyrimme aina vain parempaan ja parempaan projektien työmäärien enustettavuuteen, jolloin kaikki vinkit työmäärän arvioinnissa ovat tervetulleita.

Vuosien pitkäjänteisellä työllä omassa työpaikassani TKP Tiedossa on päästy lähes 100 prosentin työmääräpitoon. Tähän on tarvittu kaikkien toteutusprojektien järjestelmällinen laskenta ennen projektin aloitusta, sekä projektin päätyttyä tehtävä jälkilaskenta ja toteumatietojen keruu. Oma kokemuskantamme sisältää yli sadan projektin tiedot: sen pohjalta on hyvä tehdä alkavan projektin työmäärän arviointi.

FiSMA Projektijohtamisen jaostossa toimii useita työryhmiä:

- FSM-työryhmä: työstetty uusi koonlaskentamenetelmä
- SCOPE Manager ryhmä: mitoitusvastaavien oma ryhmä
- Valmisohjelmistoryhmä: ohjeistusta valmisohjelmistojen käyttöönnoton laskentaan
- Cosmic -ryhmä: nimensä mukaisesti Cosmic -käyttäjien ryhmä.

Olen osallistunut FSM-työryhmän ja SCOPE Manager ryhmän, sekä nyt jo toimintansa lopettaneen Uudet teknologiat ryhmän toimintaan. Työryhmiin osallistumisen suurimmaksi anniksi olen kokenut sen, että niissä voi itse konkreettisesti vaikut-

taa siihen, millaisiksi arviointikäytännöt ja työtavat muotoutuvat.

Tärkeä verkostosta saatava anti on myös se, että saa tavata ihmisiä eri organisaatioista ja kuulee, millaisia toimintopistelaskennan toimintatapoja muissa organisaatioissa on. Tämä antaa uusia näkökulmia myös omaan työhön.

Vuosien varrella olen kehittänyt omaa osaamistani ja kokemusta on tullut rutkasti lisää: satoja projekteja, joiden toimintopisteitä olen laskenut, tai ainakin avustanut laskennassa. Ja kymmeniä kokouksia, joihin olen osallistunut, ja joissa on nykyisin tummapukuisten herrojen lisäksi myös useita naisia, joten tasa-arvo on toteutunut tälläkin alalla. Olen kokenut FiSMA-verkoston hyödylliseksi kokemustenvaihtoforumiksi, jossa on mielenkiintoista olla mukana.

*Paula Männistö,
TKP Tieto Oy,
Finnish Software Measurement Association ry:n ja sen Projektijohtaminen-jaoston puheenjohtaja*

Koitetaan saada prosessia paremmaksi

*Timo Varkoi,
Tampereen teknillinen yliopisto,
Porin yksikkö*

Näin vastasi entinen teekkari tenttikysymykseen: *Mikä on ohjelmistoprosessien parantamisen tarkoitus?* Myös FiSMA:n prosessijohtamisen jaosto hakee vastausta tuohon moniulotteiseen kysymykseen.

Jaoston viime vuosien toiminnassa on keskitytty prosessien parantamisen apuvälineiden kehittämiseen sekä menossa olevaan kansainvälisen



Paula Männistö

ISO/IEC 15504 standardin uudistamiseen. Jaoston ohjauksella on kehitteillä kaksi ohjelmistotyökalua: monipuolinen prosessi-arvioinnin ja arviointitulosten käsittelyn järjestelmä sekä ohjelmistoprosessien parantamisen (software process improvement - SPI) asiantuntijatietämyksen hallintajärjestelmä. Jaoston jäsenet ovat myös mukana ISON standardointityöryhmissä osallistuen standardien tekemiseen. Standardien kehitysversioita on myös kokeiltu ja kommentoitu jaostossa. Toinen kansainvälinen tiedon välittämiskanava on prosessien parantamista käsittelevät konferenssit, kuten EuroSPI ja SPICE, joissa jaostolaiset ovat usein mukana sekä kuuntelijoina että puhujina.

Jaoston toiminnan runko muodostuu kokouksista, joita pidetään lähes kuukausittain. Yritysten edustajat ovat kiitettävän aktiivisesti osallistuneet jaoston kokouksiin ja sitä kautta ohjanneet FiSMA:n toimintaa ohjelmistoliiketoimintaa tukevana verkostona. Kokouksissa välitetään

yrityksille myös runsaasti tietoa maailman kehittymisestä prosessien parantamisessa. FiSMA:n toiminnan kautta sen jäsenillä on käytettävissään uusin tieto ohjelmistoteollisuuden tuottavuuteen vaikuttavista tekijöistä.

FiSMA mukana tutkimuksessa

Prosessijohtamisen jaoston kautta FiSMA osallistuu merkittävään, SPI-tietämystä kokoavaan hankkeeseen, jonka toteutuksesta vastaavat yhteistyössä Joensuun yliopiston tietojenkäsittelytieteen laitos ja Tampereen teknillisen yliopiston Porin yksikkö. Hankkeen päärahoittajana on Tekes. Hankkeessa rakennetaan asiantuntijatietämyksen tietokanta ja siihen älykkäillä hakumekanismeilla varustettu käyttöliittymä. Gnosis-järjestelmä on parhaillaan toisessa koekäytövaiheessa Joensuun ja Porin seutujen ohjelmistoyrityksissä. Projektin päättyessä tämän vuoden lopussa Gnosis on jaoston jäsenten käytettävissä kokeiluversiona. Prosessien parantamisen malleja ja menetelmiä ke-

hittämällä myös ohjelmistoteollisuuden toimintaedellytykset paranevat.

Merkittävää ohjelmistoprosesseihin liittyvää tutkimustoimintaa Suomessa harrastetaan Joensuun ja Porin lisäksi Oulussa (Oulun yliopisto ja VTT), Turussa (Åbo akademi, TUCS) ja Espoossa (TKK:n Sobe-riIT). Mielestäni FiSMAn toiminnassa voisimme tulevaisuudessa enemmänkin painottaa yhteistyötä ohjelmistoteollisuuden ja yliopistojen välillä. Viime vuosina Tekesin ja alueellisten TE-keskusten rahoitus myös ohjelmistosektorin prosessien parantamisen tutkimus- ja yritysyrityshankkeille on kasvanut, hyvänä esimerkkinä Tekesin SWENG-ohjelma. FiSMAlla on nyt hyvä tilaisuus syventää ohjelmistotuotannon tutkimuksen ja liike-elämän yhteistoimintaa.

Maailmanlaajuinen yhteistyö

FiSMA toimii monella tapaa linkkinä kansainväliseen yhteistyöhön ja alan parhaaseen osaamiseen. Pitkäjänteinen osallistuminen standardointityöhön on tuonut Suomelle näkyvyyttä ja arvostusta, josta yhtenä osoituksena on ensivuonna Helsingissä järjestettävä ISON ohjelmisto- ja järjestelmätuotannon standardointi-

kokous. Kokoukseen odotetaan noin 150 osallistujaa ympäri maailmaa, joten silloin meillä on mahdollisuus tavata useita maailman eturivin asiantuntijoita.

Yliopistojen mukaantulo FiSMAn toimintaan on myös lisännyt kansainvälisiä kontakteja ja aktiivisuutta konferenssiesitysten tuottamisessa. Samalla saadaan uusinta tietoa muun maailman tutkimushankkeista ja kehitystrendeistä. FiSMA osallistuu näin kansainvälisellä yhteistyöllä hankitun ohjelmistoprosessien parantamiseen tarvittavan tietämyksen muokkaamiseen kotimaisten ohjelmistoyritysten toimintaympäristöön sopivaksi.

Tulevaisuudessakin FiSMAn tavoitteena on kehittää ohjelmistoteollisuuden kilpailukykyä parantamiseksi tarvittavia toimintamalleja, tukipalveluja ja apuvälineitä.

Lopuksi, prosessien parantamisen tarpeellisuutta voisi perustella seuraavasti:

Ohjelmistoja tuottavan yrityksen toiminnan kehittämiseksi ohjelmistoprosessien merkitys on keskeinen. Ohjelmistoprosessien tehokkuudella ja ennustettavuudella voidaan vaikuttaa olennaisesti yrityksen asia-



Timo Varkoi

kastytyväisyyteen ja kilpailukykyyn sekä lopputuotteiden laatuun.

Lisätietoa FiSMasta ja sen jaostoista: www.fisma-network.org

*Timo Varkoi,
projektipäällikkö,
Tampereen teknillinen yliopisto,
Porin yksikkö,
FiSMAn prosessijohtamisen jaoston
puheenjohtaja*

Systemityöyhdistyksen jäsenkyselyn tulokset: mitä jäsenet odottavat yhdistykseltä?

Markku Niemi,
STTF Oy

Yhteenveto

Toukokuussa 2004 toteutetun sähköisen jäsenkyselyn keskeisimmät tulokset: Systemityö-lehti osoittautui toimintamodoista ylivoimaisesti arvostetuimmaksi ja parhaiten tunnetuksi. Systemityöalueista eniten kiinnostusta on tällä hetkellä projektinhallintaan ja määrittelymenetelmiin. Johtopäätökset: yhdistys turvaa lehden laadun tehden siihen pieniä parannuksia ja satsaa lisää kerhotoimintaan ja jäsentiedottamiseen.

Jäsenkyselyyn vastasi 415 sytykeläistä, mikä oli noin viidennes kaikista jäsenistä ja hieman vajaa kol-

mannes niistä, joilla on liiton jäsenrekisteristä saatu toimiva sähköpostiosoite. Jäsenkyselyyn vastanneiden kesken suoritettussa arvonnassa ilmaisen laivaseminaarin voitti Jarmo Saarinen ja samppanjapullon Janne Kattaja. Kiitokset kaikille vastanneille ja onnea voittajille!

Jäsenkunnan profiili

38% ilmoitti olevansa asiantuntijoita/konsultteja. Seuraavaksi suurin ryhmä oli projektipäällikkö 15 prosentilla. Johtajia/päälliköitä ilmoitti olevansa 14,3%, suunnittelijoita 12,8%. 80% vastaajista kuului siihen näihin kategorioihin. Ohjelmoija, opiskelija, tutkija, menetelmäkehittäjä ja muut esillä olleet nimikkeet saivat lähinnä hajaääniä.

Kiinnostuksen kohteet

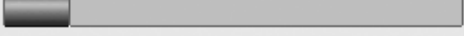
Projektinhallinta ja määrittelymenetelmät tuntuvat nyt olevan jäsenkuntaa eniten kiinnostavaa asiaa, sillä kummankin ruksasi yli puolet kaikista vastaajista. Toisena ryhmänä osa-alueista erottuvat arkkitehtuurit, vaatimusten hallinta, liiketoimintaprosessien kehittäminen ja ohjelmistotyön laatu ja tuottavuus 46-47% kiinnostusasteella. Ehkä hieman yllättävää oli testauksen ja käytettävyyssioiden melko alhainen kiinnostavuus (noin kolmannes vastaajista) suhteessa muihin osa-alueisiin. Niin ikään teknologia-asiat eivät vaikuta olevan ainakaan jäsenistön enemmistön päällimmäisiä murheita tällä hetkellä.

2. Systemityön osa-alueet: ruksaa Sinua eniten kiinnostavat osa-alueet (voit halutessasi tarkentaa vapaamuotoisesti kiinnostuksesi kohdetta)




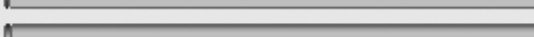
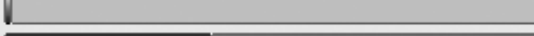

Kysymykseen vastanneet: 414

(2.1) Liiketoimintaprosessien kehittäminen		46.4%	192
(2.2) Vaatimusmäärittely ja vaatimusten hallinta		47.3%	196
(2.3) Määrittelymenetelmät		50.7%	210
(2.4) Arkkitehtuurit		47.8%	198
(2.5) Ohjelmointi ja toteutusteknologiat		36.5%	151
(2.6) Testaus		31.9%	132
(2.7) Tiedonhallinta ja tietokannat		38.9%	161
(2.8) Käytettävyys ja käyttöliittymät		36.0%	149
(2.9) Projektinhallinta		51.0%	211
(2.10) Ohjelmistotyön laatu ja tuottavuus		46.1%	191
(2.11) Jokin muu yllä mainitsematon - mikä:		5.6%	23

3. Yhdistyksen toimintamuodot: ruksaa sinua eniten kiinnostavat (voit halutessasi tarkentaa vapaamuotoisesti):
Kysymykseen vastanneet: 391

(3.1) Kerhotoiminta		16.6%	65
(3.2) Yhdistyksen kevät- ja syysseminaarit		36.3%	142
(3.3) Laivaseminaari		32.0%	125
(3.4) Muut jäsentilaisuudet		18.2%	71
(3.5) Systeemityölehti		88.0%	344
(3.6) Työskentely työryhmässä		8.2%	32
(3.7) Keskusteluihin yhdistyksen verkkosivuilla		14.1%	55
(3.8) Jokin muu - mikä:		1.5%	6

4. Valitse merkittävin syy, jonka vuoksi kuulut Systeemityöyhdistykseen:
Kysymykseen vastanneet: 405

(4.1) JavaSIG-kerho		2.5%	10
(4.2) RELA-kerho		0.5%	2
(4.3) Testaus-kerho		4.0%	16
(4.4) vvv-kerho		0.7%	3
(4.5) Yhdistyksen kevät- ja syysseminaarit		1.2%	5
(4.6) Laivaseminaari		1.2%	5
(4.7) Systeemityölehti		39.5%	160
(4.8) Tietotekniikan liiton jäsenetulehdet ja muut edut		34.3%	139
(4.9) Haluan jäsenyydelläni vain tukea systeemityön kehittämistä		7.9%	32
(4.10) Muu syy - mikä:		8.1%	33

Yhdistyksen toimintamuodot

Systeemityö-lehti oli lähes mursertavan ylivoimainen: 88% kysymykseen vastanneista ilmoitti sen eniten kiinnostavien joukkoon. Tämä on toki ymmärrettävää - tulehan se jäsenelle postissa, minkä ansiosta siihen voi syventyä omana tahtiin. Kakokseksi ja kolmoseksi sijoittuivat yhdistyksen sääntömääräiset kevät- ja syyskokoukset sekä jo perinteeksi tulut laivaseminaari.

Erilaiset jäsenetulehdet ovat ylivoimaisesti tärkeimmäksi äänestetty syy kuulua yhdistykseen. Systeemityö-lehden lisäksi suosiota saivat Tietotekniikan liiton yleiset jäsenetulehdet.

Yhdistyksen toiminnan laadun arviointi

Peräti kolme neljännestä vastaajista piti Systeemityö-lehden laatua hyvänä tai peräti erinomaisena. Se myös tunnettiin parhaiten ”en osaa sanoa” vastausten vähäisyyden vuoksi. Erot eivät ole kovin suuria, mutta eniten parannettavaa jäsenistö löysi tiedottamisesta ja yhdistyksen verkkosivuista www.sytyke.org.

Vähiten tunnetuksi osoittautui kyselyn perusteella työryhmätoiminta, josta 82% ei osannut antaa arviota. Työryhmät eivät liene olleet kovin näkyviä viime vuosina. Toisaalta sytykeläiset ovat olleet varsin aktiivisia liittotason työryhmissä, esimerkiksi Tietojärjestelmän hankintaopasprojektissa.

visia liittotason työryhmissä, esimerkiksi Tietojärjestelmän hankintaopasprojektissa.

Johtopäätöksiä

Miten yhdistys pyrkii turvaamaan tunnistetut vahvuutensa ja kehittämään todettuja heikkouksia?

Systeemityö-lehti on saavuttanut aseman, joka kannattaa turvata. Lehdestä kiitokset ja onnittelut Lauri Laitiselle, monivuotiselle päätoimittajalle! Kehitystä tehdään evoluutio-omaisesti, pienin askelin. Lehden pyritään jatkossa saamaan lisää ilma- vuutta esimerkiksi kuvituksella ja haastattelutyypillisillä pikku jutuilla.

**7. Arvioi Systeemityöyhdistyksen toiminnan laatua:
Kysymykseen vastanneet: 410**

	(7.1) 5=erinomainen (arvo: 1)	(7.2) 4=hyvä (arvo: 2)	(7.3) 3=kohtalainen (arvo: 3)	(7.4) 2=välttävä (arvo: 4)	(7.5) 1=heikko (arvo: 5)	(7.6) En osaa sanoa (arvo: 6)
Kerhotoiminta (ka: 4.8, yht: 399)	0.5% 2	18.5% 74	14.0% 56	0.8% 3	0.3% 1	65.9% 263
Yhdistyksen kevät- ja syysseminaarit (ka: 5.0, yht: 396)	0.5% 2	16.9% 67	9.1% 36	0.5% 2	0.3% 1	72.7% 288
Laivaseminaari (ka: 4.8, yht: 397)	7.1% 28	15.6% 62	5.8% 23	0.5% 2	0% 0	71.0% 282
Systeemityölehti (ka: 2.2, yht: 407)	17.0% 69	57.5% 234	18.9% 77	3.2% 13	0% 0	3.4% 14
Työryhmätoiminta (ka: 5.4, yht: 384)	0.5% 2	5.5% 21	10.7% 41	0.8% 3	0.3% 1	82.3% 316
Verkkosivut (www.sytyke.org) (ka: 3.9, yht: 399)	0.5% 2	25.8% 103	29.1% 116	7.8% 31	0.8% 3	36.1% 144
Systeemityön menetelmien ja apuvälineiden kehittäminen (ka: 4.5, yht: 391)	0.5% 2	18.7% 73	21.5% 84	2.8% 11	1.0% 4	55.5% 217
Tiedottaminen jäsenistölle (ka: 3.1, yht: 400)	2.5% 10	37.2% 149	35.5% 142	9.0% 36	1.2% 5	14.5% 58
Johtokunnan toiminta (ka: 5.0, yht: 395)	1.0% 4	15.4% 61	11.6% 46	0.8% 3	0.5% 2	70.6% 279
	3.4% 121	23.7% 844	17.4% 621	2.9% 104	0.5% 17	52.2% 1861

Systeemityöyhdistyksen toiminnan laadun arviot. Huom: Vastauskeskiarvot eivät pidäpaikkansa.

Kirjoittajaksikin ilmoitettiin kyselyssä ilahduttavan moni jäsen.

Jäsentiedotuksen ja -viestinnän foorumeita parannetaan sekä Sytykeen että liittoyhteisön toimesta. Parannuksia on luvassa sekä liiton yhteisen www-sivuston (tt-tori) jäsenasiointiin että sen rinnalle kehitettävään keskitetysti tarjottuihin viestintäpalveluihin. FLE on jo osoittautunut toimivaksi foorumiksi johtokunnalle ja muille aktiiviryhmille.

Yhdistys ottaa ensi vuoden alusta käyttöön sähköisen jäsenkirjeen, joka sai kyselyssä lähes varauksetoman kannatuksen. Sähköpostikirjeellä voidaan sekä alentaa jakelukustan-

nuksia että saada tietoa jäsenille nykyistä ajantasaisemmin. Jotta sähköinen kirje tavoittaisi jäsenistön, ovat sähköpostiosoitteiden ja muiden yhteystietojen päivitystalkoot edessä ennen vuoden vaihdetta. Jäsenten panosta kaivataan tähän, esimerkiksi www.tt-tori.fi -sivuston jäsentietojen päivityslomakkeen kautta (www.tt-tori.fi).

Lehdet ovat suosituin toimintamuoto, mutta yhdistys satsaa myös **kerhotoimintaan**: Systeemityöalueista suosituimmaksi osoittautuneelle projektitoiminnalle on jo suunnitteilla oma kerho, joka toimii todennäköisesti yhteistyössä projektitoiminnan muiden yhdistysten kanssa.

JavaSIG-kerho jatkaa arkkitehtuurialueen aktiivista toimintaa. Kerhotoimintaa puuhataan myös tietovarastoinnin alueelle, relaatiotietokantakerhon (RELA) rinnalle. Testauskerho FAST jatkaa vireää toimintaansa.

Toki muillekin uusille systeemityön osa-alueiden kerhoille on tilaa. Tartu ihmeessä toimeen ja ota yhteyttä olemassa olevien kerhojen vetäjiin tai yhdistyksen johtokuntaan, joilta saat käytännön vinkkejä ja muuta apua kerhotoiminnan virittämiseen.

*Markku Niemi,
STTF Oy,
Sytykeen johtokunnan jäsen*

Miten tulevaisuutemme toivoja arvioidaan ammattitaidon MM-kilpailuissa?

Markku Niemi,
STTF Oy

Olympialaiset ovat juuri päättyneet. Yleisurheilufanien ajatukset lienevät jo ensi vuoden MM-kisoissa, jotka pidetään Helsingin olympiastadionilla. Kisajärjestelyt käyvät jo kuumana myös hieman toisenlaisia kilpailuja varten, jotka pidetään Helsingissä ensi vuonna.

Suomi on saanut järjestettäväkseen vuoden 2005 **WorldSkills** eli Ammattitaidon MM-kilpailut. Järjestelyistä vastaa **Skills Finland ry**, joka pyrkii edistämään kotimaisen ammattillisen koulutuksen ja osaamisen, erityisesti kädentaitojen arvostusta. Kisat ovat suurin Helsingin Messukeskuksessa koskaan järjestetty tapahtuma, johon odotetaan jopa 150 000 katsojaa.

Joka toinen vuosi järjestettävissä MM-kilpailuissa alle 23-vuotiaat nuoret ympäri maailmaa kilpailevat omaa ammattialaansa vastaavassa lajissa. Virallisia kilpailulajeja on 34, joiden lisäksi kilpaillaan noin viidessä näytöslajissa. Lajit edustavat tekniikan, teollisuuden ja palvelujen aloja. Kilpailijoita saapuu lähes 40 maasta yhteensä noin 800.

Kilpailulajit vaihtelevat floristikaasta mekatronikkaan, laatoituksesta

ta CAD-suunnitteluun, kokkaamisesta elektroniikkaan, muurauksesta pukuompeluun – jos haluaa oikein etsimällä etsiä lajikirjon ääripäitä. Suomen edustajat valikoituvat lajikohtaisissa SM-kisoissa, jotka pidetään lähikuukausina eri puolella Suomea.

IT:n osuus kilpailulajeissa

Painotetusti IT-osaamista mitattavia lajeja ovat *Informaatioteknologia (IT/Software Applications)*, *Web design* sekä *Tietokone ja verkot (IT PC/Network Support)*. ”Kilpailulajeista kuitenkin jopa kolmanneksessa IT-osaaminen on hyvin keskeistä”, sanoo **Tjerk ”Jack” Dusseldorp**, kansainvälisen WorldSkills-järjestön australialainen puheenjohtaja. Jack kävi Suomessa kesäkuussa 2004.

Jackin mukaan julkinen koulutussysteemi ei yksin pysty vastaamaan osaajakysyntään oikea-aikaisesti. Pitää etsiä jatkuvasti uusia tapoja välittää ja hyödyntää kokeneiden osaajien taitoja ja osaamista. Tätä WorldSkills pyrkii omalta osaltaan edistämään.

Jack haluaa kehittää kilpailua nykyistä enemmän sosiaalisten taitojen (”soft skills”) arvioimisen suuntaan. Avainsana on aikaansaaminen tiimien yhteistyönä, mitä eräissä la-



Yllä: **Jack Dusseldorp, WorldSkills-organisaation puheenjohtaja**

jeissa jo arvioidaankin. Esimerkiksi ”Manufacturing Team Challenge” eli Konepajatekniikan lajissa ei mitata pelkästään tekemisen ja loppusuorituksen teknistä laatua vaan myös aikaansaadun ratkaisun kustannustehokkuutta.

Jackin ajama suuntaus on varmasti oikea meidän systeemityöammattilaistenkin mielestä. Tiimityö- ja johtamistaitojen arviointi vaatii tietysti enemmän kilpailun tuomareilta. Toivottavasti kisat sujuvat ilman Ateenan olympialaisten kaltaisia arvosteluskandaaleja.

Kisojen informaatiojärjestelmä kehitteillä

Järjestäjien IT-osaaminen on koetuksella jo hyvän aikaa ennen kilpailuja. Kisaorganisaatio on päättänyt tilata uuden atk-pohjaisen informaatiojärjestelmän aiemmissa kisoissa käytetyn sveitsiläisvalmisteen ohjelmiston tilalle. Uusi CIS – Competition Information System - hoitaa kisojen tuloslaskennan internet-pohjaisesti, avoimella lähdekoodilla toteutettuna.

Projektin luonteeseen sopii, että informaatiojärjestelmän on annettu alan nuorten virtuoosien rakennettavaksi. Hämeen ammattikorkeakoulun opiskelijatiimin rakentama systeemi on testausvaiheessa. Sitä testataan ennen varsinaista koitosta mm. Turun Taitaja 2005 tapahtumassa tammi-kuussa. ”Aloitimme CIS-kehitystyön täysipäiväisesti maaliskuussa 2004. Yhteistyö kansainvälisen ohjausryh-

män kanssa on toiminut kohtalaisen hyvin ja järjestelmän kehitysprosessi on aikataulussa.”, kertoo CIS-kehitystiimin jäsen Tommi Saarinen (WorldSkills 2005 Newsletter nro 6).

Onnea loppusuoralle

Koko tapahtuman järjestely on ilmeisen vaativa ponnistus. Uusi business-kriittinen informaatiojärjestelmäkin on vielä kehitettävänä. Me systeemityöläiset toivotamme onnea ja tsemppiä kisajärjestelijöille niin järjestelmäkehitykseen kuin muihinkin valmisteluihin sekä tietysti itse tapahtuman läpivientiin.

Miksemme toivottaisi onnea samalla myös Suomen kisajoukkueelle – alakanttiin sujuneista olympialaisista kun taisi jäädä mitalinälkää - ainakin meille penkkiurheilukärpäsen puremille.

Jack Dusseldorp toivotti Systeemityö-yhdistyksen ja muut IT-alan

aktiivit tervetulleiksi seuraamaan tapahtumaa toukokuussa 26.-29.5.2005 - vaikkapa osaksi tapahtuman ”Global Village” -konseptia. Pääsyliput tulevat myyntiin lähiaikoina. Mahtaisimmeko saada Sytykeestä kokoon porukan paikan päälle kannustamaan omia?

Lähteet

WorldSkills-organisaation presidentin Tjerk ”Jack” Dusseldorpin haastattelu 17.6.2004.

Skills Finland ry:n www-sivut: <http://www.skillsfinland.com>

WorldSkills 2005 Helsinki kisojen www-sivut: <http://www.wsc2005helsinki.com/>

WorldSkills 2005 Helsinki Newsletter nro 6, elokuu 2004: http://www.wsc2005helsinki.com/attachments/newsletter_nro6.pdf

SYTYKEEn 25v-juhlat

perjantaina 12.11.2004

Ravintola Tarmossa.

**Ohjelmasta ja teemasta tarkemmin
jäsenkirjeessä**

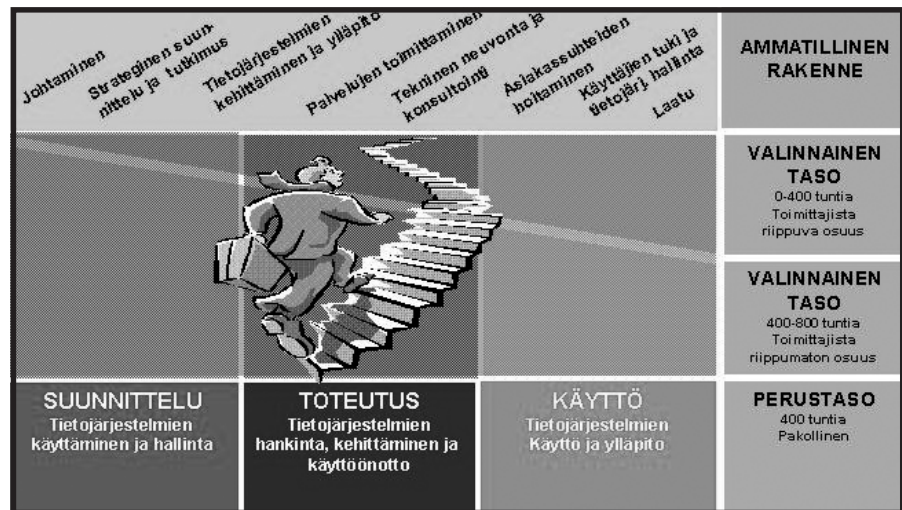
Tervetuloa!

Miten systeemytö on esillä EUCIPissa, TTL:n tulevassa “ammattilaisten ajokorttikonseptissa”?

Pentti Saastamoinen,
Tietotekniikan liitto

Systeemytöläisenkin maailmaan on ilmaantunut kasvava joukko sertifikaatteja, jollaisen suorittamalla voi osoittaa ammatillista osaamistaan. Ehkä tunnetuimpia esimerkkejä ovat Microsoftin ja monien muiden valmistajien teknologioihin liittyvät (ja niiden myyntiä tukevat) sertifikaatit, sekä tietoturvaosaajien CISSP.

IT-ammattilaisten ja IT-ammattilista koulutusta tarjoavien oppilaitosten lukumäärän raju kasvu vuosittain vaihteen molemmin puolin synnytti tarpeen luoda yhteismitallisuutta värikkääksi käyneeseen nimikkeiden ja tutkintosisältöjen kirjoon. Globalisaatio on tuomassa tilanteeseen vielä omat lisähaasteensa. Kauan jatkuneesta alikoulutuksesta johtuen alalla on myös paljon ilman muodollista pohjakoulutusta olevia ammattilaisia ja tarvitaan keino verrata kokemuksen tuomaa osaamista tuoreeseen koulutietoon



Euroopan tietotekniikan liittojen yhteisö CEPIS on kehittänyt vuodesta 2000 alkaen mm. EU:n tuella EUCIP (European Certification of Informatics Professionals) –nimistä sertifiointiohjelmia.

EUCIP:in ensimmäiset osat ovat valmiina ja tuotantokäyttö käynnistyi syksyllä 2003 muutamissa Euroopan maissa. TTL on ollut aktiivisesti mukana EUCIPia kehittämissä ryhmissä, Suomessa EUCIP:ia on pilotoitu ja

varsinainen toiminta on tarkoitus käynnistää vuoden 2005 alkupuolella. Sitä ennen on mm. suomennettava useiden satojen kysymysten laajuisen kysymyspankki.

EUCIP toimii kahdella tasolla: IT-ammattilainen voi suorittaa tutkinnon ja saada EUCIP-sertifikaatin ja oppilaitos voi verrata omia tutkintosisältöjään EUCIP-tutkintovaatimuksiin ja varmistaa opiskelijoidensa kilpailukyvyyn työmarkkinoilla.

Yllä oleva kuva esittää EUCIP-sertifiointijärjestelmän periaatteita ja sisältöä. Tutkinto sisältää aineksia tietotekniikan soveltamisesta, tietojärjestelmien toteuttamisesta ja käytöstä mukaanlukien laitteisto- ja tietoliikennetekniikka. Vaakasunnassa tutkinto koostuu kaikille yhteisestä, pakollisesta perustasosta (ammattilaisen yleistiedot) sekä valinnaisesta tasosta, jonka sisältö määräytyy ammatillisen profiilin mukaisesti.

Opintoviikkoina mitattuna koko tutkinnon laajuus on noin 40 ov.

Ammatillisten profiilien kokonaismääräksi on arvioitu 20-30 ja niistä on valmiina neljä: Business Analyst, Information Systems Analyst, Software Developer ja Network Manager.

Systeemyön toimenkuvia ovat erityisesti Information Systems Analyst (suunnittelija) sekä Software Developer (ohjelmoija).

Suunnittelijalta edellytetään hyvää osaamista mm. seuraavissa aiheissa:

- vaatimusten hallinta
- tietomallinnuksen tekniikat ja välineet
- kehittämisen ja testauksen tekniikat ja välineet
- oliokeskeiset suunnittelu- ja toteutusmenetelmät mukaanluetuna UML-tekniikat.

Yleistietotason osaamista suunnittelijalta edellytetään mm. seuraavissa:

- liiketoimintaprosessien mallintaminen
- tietojärjestelmien liiketoimintayhteydet
- projektin hallinta
- liiketoimintariskit ja tietoturva.

Ohjelmoijan osaamisvaatimuksissa painottuvat enemmän ohjelmointikielet ja muut kehittämisvälineet, käyttöjärjestelmät ja tietokannat. Yleistietoja edellytetään mm. arkkitehtuurisuunnittelussa, tietosuunnittelussa ja eräissä muissa suunnittelun osa-alueissa. Internet-tekniikat ovat keskeisesti mukana.



Systeemyöhön liittyvien ammattiprofiilien määrittely on aloitettu yleisimmistä ja joukko täydentyy myöhemmin. Kehittämiskohteiden listalla ovat esimerkiksi testaus ja järjestelmäintegrointi.

*Pentti Saastamoinen,
Suunnittelujohtaja,
Tietotekniikan liitto*

Relaatiokantojen oliopiiirteiden käyttö Suomessa

RELA-kerhon tutkimus

Antoni Wolski ja
Ari Hovi,
SYTYKE:n RELA-kerho

Ensin oliotietokantojen piti syrjäyttää relaatiotietokannat. Niin ei ole vielääkään käynyt. Sen sijaan relaatiotietokantatoimittajat alkoivat lisätä olio-ominaisuuksia tuotteisiinsa. On syntynyt uusi käsite: oliorelaatiotietokanta.

RELA-kerho on suorittanut kyselyn suomalaisten tietokantakäyttäjien keskuudessa oliorelaatiolaajennusten mielekkyydestä ja tosiasiallisesta käytöstä. Tulokset tukevat käsitystä, että aikoinaan tehtiin paljon melua tyhjistä.

Suuret visiot

Vuosina 1995-97 tietokanta-ala koki pienen ravistuksen, kun suuret tietokantavalmistajat toivat uuden oliorelaatiotekniikan käyttäjien ulottuville. Uutta oli poikkeaminen puhtaasta relaatiomallista ja oliomaailmasta tunnettujen käsitteiden muukaanotto. Laajennettujen relaatiotietokantojen tekniikkaa on markkinoitu lupauksilla helpommasta mallintamisesta, paremmasta sopivuudesta ohjelmointikieliin ja paremmasta suorituskyvystä [1, 3]. Uuden tekniikan piti alentaa kehityskustannuksia

ja pidentää järjestelmien elinkaarta. Sinä aikana markkinoille tulleeisiin tuotteisiin kuuluivat Informixin (ja aiemmin Illustran) Universal Server, IBM:n Universal Database ja Oraclen Object Option. Oliosanojen runsaus tuotekuvauksissa heijasti taistelua oliotietokantoja vastaan käyttäjien sielusta. Oliotietokannat olivat tuolloin julkisuudessa vahvasti esillä, vaikkei kaupallisesta menestyksestä ollut merkkejä.

Todellisuudessa oliorelaatiotietokannat tukivat olioajattelua vain viitteellisesti. Oliomaailman tärkeimmät ominaisuudet kuten periytyminen ja monimuotoisuus usein puuttuivat toteutuksista. Tätä voidaan selittää sillä, että ennen markkinoille tuloa tätä suuntausta kutsuttiin nimellä "laajennettu relaatiomalli". Lopulta ajan oliohenkisyys voitti ja nimeksi tuli "oliorelaatiomalli".

Systeemityöyhdistys SYTYKE ry:n Relaatiotietokantakerho (RELA-kerho) päätti kartoittaa oliolaajennuksilla laajennettujen relaatiotietokantojen tekniikan vastaanottoa systeemi ammattilaisten keskuudessa nyt, kun tekniikan kaupallistamisesta on kulunut jo useita vuosia. Tarkoituksenamme oli selvittää, miten annetut lupaukset ovat toteutuneet ja min-käläisiä kokemuksia uudesta tek-

niikasta on saatu. Jos näytti siltä, että oliolaajennuksista ei oltu innostuttu, halusimme myös selvittää miksi.

Tämä artikkeli sisältää yhteenvedon tästä selvityksestä. Pienen teknisen johdannon jälkeen esitämme kyselyn ja sen tulokset

Myös standardi

Vuonna 1999 hyväksytty standardi SQL-99 sisälsi paljon oliorelaatiolaajennuksia [2]. Niihin kuului laaja tietotyypinvalikoima, periytyminen, tyyppien metodit ja niiden monimuotoisuus, ja taulutyyppiin pohjautuvat taulut. Vain osa näistä ominaisuuksista on toteutettu tuotteissa.

Mikä on oliorelaatiotietokanta?

Tietokantaratkaisun voidaan sanoa pohjautuvan oliorelaatiomalliin, jos yhtäkin alla olevista ominaisuuksista on käytetty:

Johdetut tyypit (distinct types)

Olemassa olevia tietotyyppejä voidaan käyttää uudennimisen tietotyypin luomiseen. Kaikki perustyyppien ominaisuudet (sallitut arvot, operaatiot) säilyvät. Johdettujen tyyppien

avulla voidaan nostaa tyyppityksen vahvuutta: järjestelmä estää eri tyyppinä olevien arvojen (sarakkeiden) käyttöä samantyyppisinä lausekkeissa.

Esimerkki:

Kokoelmatyypit (collection types)

Taulujen rakennetta voidaan syventää jakamalla sarakkeen arvot pienempiin osiin, vaikka haittana on, että tämä rikkoo ensimmäisen normaali-muodon. Kokoelmatyyppejä ovat ARRAY (jossa on samantyyppisiä arvoja) ja ROW (joka kokoaa eri tyyppisiä).

Esimerkki:

Rakenteelliset tyypit (structured types, abstract data types)

Rakenteelliset tyypit ovat varsinaisten SQL-oliolaajennusten perusta. Niihin nojautuvat tyyppimetodit, tyyppipohjaiset taulut, tyyppiperiytyminen ja viitetyyppi. Tällaisen tyyppin avulla käyttäjä määrittelee rakenteen, jolla ei ole esiintymää.

Esimerkki:

Tyyppien metodit (type methods)

Rakenteelliseen tyyppiin voidaan liittää metodeja, joilla rakennetta käsitellään yhtenä kokonaisuutena (oliona). Tyyppikohtaiset metodit voidaan kirjoittaa joko SQL-kielen laajennuksilla tai toisella ohjelmointikielillä (Java, C).

Tyyppipohjaiset taulut (type-based tables)

Rakenteellisen tyyppin pohjalta voidaan luoda taulu (tyyppipohjainen taulu).

Esimerkki:

Viitetyyppi (reference type)

Tyyppipohjaisessa taulussa voidaan riveihin viitata käyttämällä rivit yksilöivää tunnistetta, joka ei ole sarakkeen arvo (eikä voi muuttua). Tällaiselle tunnisteelle voidaan antaa jopa sarakkeen nimi, kuten yllä olevassa esimerkissä. Viiteavainten sijasta tauluihin voidaan viitata REF-tyypisellä viittauksella.

Esimerkki:

Tyyppiperiytyminen (type inheritance)

Rakenteellinen tyyppi voi pohjautua toiseen tyyppiin.

Olionäkymät (object views)

Perinteiset taulut voidaan saada näkymään tyyppipohjaisina tauluina, jolloin ne saavat oliopohjaisia piirteitä.

Laajennustietotyypit (extensible types, opaque types)

Nämä tietotyypit ovat monimutkaisia laajennuksia, jotka ovat saatavilla tietokantatuotteessa tai kolmannen osapuolen kehittäminä ja niitä voidaan liittää tuotteeseen. Nämä tyyppit on tarkoitettu esim. videon, kuvan, äänen, paikkatiedon tai tekstitiedon käsittelyyn. Laajennustyyppien sisäinen rakenne on käyttäjältä piilossa ja sisältö käsitellään tyyppikohtaisten metodien avulla.

Kysely ja sen tulokset

Kysely on jaettu suomalaisten tietokantakäyttäjien kokouksissa ja koulutustilaisuuksissa. Kyselyyn on saatu yhteensä vain 21 vastausta. Kyselyn alkuosassa kyseltiin onko tietty ominaisuus tuttu, onko sitä kehitetty tai sovellettu tuotannossa. Toisessa osassa pyydettiin vapaamuotoisia vastauksia lisäkysymyksiin. Alla on kysymykset ja niiden kohdalla yhteenveto vastauksista.

Osa 1: ominaisuudet

Johdetut tyypit (distinct types)

Olen kuullut: 18

Olen kokeillut: 3

Olen soveltanut: 0

Kokoelmatyypit (collection types)

Olen kuullut: 16

Olen kokeillut: 2

Olen soveltanut: 2

Rakenteelliset tyypit (structured types)

Olen kuullut: 14
Olen kokeillut: 3
Olen soveltanut: 1

Tyyppien metodit (type methods)

Olen kuullut: 12
Olen kokeillut: 1
Olen soveltanut: 0

Tyyppipohjaiset taulut (type-based tables)

Olen kuullut: 9
Olen kokeillut: 3
Olen soveltanut: 1

Viitetyyppi (reference type)

Olen kuullut: 10
Olen kokeillut: 3
Olen soveltanut: 1

Tyyppiperiytyminen (type inheritance)

Olen kuullut: 9
Olen kokeillut: 3
Olen soveltanut: 0

Olionäkymät (object views)

Olen kuullut: 11
Olen kokeillut: 2
Olen soveltanut: 0

Laajennustietotyypit (extensible types)

Olen kuullut: 11
Olen kokeillut: 2
Olen soveltanut: 2

Osa 2: lisäkysymykset

(Vastaukset ovat numeroituja vastaajan mukaan)

Mikä oli motiivisi tutustua näihin ominaisuuksiin?

- 1) Kiinnostus selvittää missä tilanteissa niistä olisi hyötyä.
- 2) Seurata kehitystä.
- 3) Ohjelmointityön laatu & tehokkuus.

- 4) Mielenkiinto.
- 5) Yleissivistys, jos joskus niitä tarvitsisi.
- 6) Oli kurssilla.
- 7) Olisi avuksi, tuli ostetun mukana ja haluttiin lisää.
- 8) Oliomallin ja relaatiotietokannan yhdistämisen tarve, impedance mismatch.
- 9) Projekti, jossa käytettiin oliomallinnusta ja olio-ohjelmointia ja käsiteltiin tekstidokumentteja ja AV-dataa.
- 10) 90-luvun puolivälissä tietokanta-alustan valitseminen sisällönhallintajärjestelmälle.
- 11) Paikkatieto-ohjelmistot tukevat Spatial Objekti-tyyppiä paikkatietojen tallennusformaattina.
- 12) Kurssi.
- 13) Olen kuullut joitakin esityksiä ja selannut muutamia artikkeleita.
- 20) Olisko niistä hyötyä.

Olivatko kokemuksesi hyviä vai huonoja (ja miksi)?

- 1) Auttoi käyttämään uudestaan, kun tuli vastaan ongelma, joka oli hyvä ratkaista objektityypillä. Kokeilu auttoi löytämään pari asiaa, jotka pitää huomioida, jos käyttää objektiominaisuuksia.
- 2) Oracle8 toteutuksessa oli liikaa rajoituksia, eikä työkaluohjelmissa kunnan tukea.
- 3) Nojoo, ei huonoa eikä hyvää.
- 7) Monimutkaista. Jos osaa, hyvä.
- 14) En lopulta luottanut tietokantamoottorin virheettömään toimintaan näiden epätyypillisten ominaisuuksien suhteen. Tekstitiedolle ja AV-datalle kehitettiin tässä tapauksessa toinen ratkaisu.
- 16) Oliorelaatiokannat olivat silloin kovin tuoreita ja bugisia, eli osittain huonoja.
- 17) Hyvät kokemukset. Hyvä suoriutuskyky suurien aineistojen käsittelyssä.

- 20) Huonoja. Liikaa monimutkaisuutta hyötyyn nähden. Kokoelmatyypit järjettömiä. Relatiomaailma ja oliomaailma pitäisi ”yhdistää” jollakin muulla tavalla.

Mikä on merkittävä tekijä, joka edistää tai estää näiden ominaisuuksien käyttöä?

- 1) Estää: vain harvat työkalut tukevat objektiominaisuuksia.
- 2) Ei standardin mukaisia toteutuksia. Liikaa puutteita ja rajoituksia. Turhan monimutkainen malli, sotkee relaatiomallin.
- 3) Oppimiskäyrä, alkupanostus estävät.
- 6) Välineet eivät tue, ihmiset eivät tunne.
- 7) Monimutkaista.
- 10) Liiketoiminnan vanhollisuus estää uusien ominaisuuksien käyttöönottoa ja niihin tutustumista. Jos väline (ohjelmointi-) ei tue, ei ominaisuuksia voi käyttää.
- 12) Tehokkuus/tehottomuus
- 15) Merkittävät estäjät ovat oliominaisuuksien oletettu kehittymättömyys ja tarpeen puuttuminen.
- 16) Edistää: erityisesti kannan toiminnallisuuden laajentaminen uusilla tyypeillä ja upotetulla koodilla (C ja Java).
- 17) Useat eri ohjelmistotoimittajien tuotteet lukevat suoraan Oracle Spatial tietokantaobjektia, jolloin ei tarvita erillisiä tiedostopohjaisia tiedonsiirtoja järjestelmien välillä. Spatiaaliset hakurutiinit rakennettu valmiiksi tietokantaproseduureiksi.
- 19) Maksajan puute estää. Kukaan ei maksa ominaisuuksiin tutustumista, jos ei voida osoittaa niille löytyvän hyötykäyttöä & sätöjä.

20) Monimutkaisuus. Ei hyötyä.

Aiotko jatkaa näiden ominaisuuksien soveltamista ja miten?

- 1) Tarvittaessa.
- 2) Seurata kehitystä.
- 3) Kyllä.
- 4) En, toimenkuva muuttunut.
- 6) Jos tulee tarve.
- 7) Ehkä
- 9) Ei vielä
- 14) Tarpeen tullen testataan uudemmilla versioilla. Seuraan näiden ominaisuuksien yleistymistä.
- 16) Tuotannossa todennäköisesti vaihdetaan alustaa toiseen, tilanne tutkitaan sitten
- 17) Kyllä jatkan. Tarkoituksena käyttää Spatial-objekteja mahdollisimman paljon eri ohjelmistojen kesken yhteisenä tiedontal-
lenusmuotona.
- 19) Yritän tutustua & keksiä käyttöä.
- 20) En, jos ei ole pakko.

Aiotko suositella näiden ominaisuuksien käyttöä muille?

- 1) Yksi käyttökelpoinen vaihtoehto.
- 2) En ainakaan tällä hetkellä.
- 3) Kyllä.
- 4) Kyllä.
- 6) En.
- 7) Kyllä.
- 9) Ei vielä.
- 14) Korkeintaan mainita. Jokainen arvioikoon tilanteen omasta näkökulmastaan.
- 16) Miksei.
- 17) Kyllä.
- 18) En osaa sanoa.
- 20) Mitäs luulisit (toim. huom. EI)

Käytetäänkö yrityksessäsi oliosysteemityötä ja -ohjelmointia?

- 1) Satunnaisesti, jatkossa enemmän.

- 2) Kyllä.
- 3) Joissakin paikoissa.
- 4) Kyllä.
- 5) Kyllä.
- 7) Vain 1 projekti
- 8) Ei.
- 9) Ei aktiivisesti.
- 10) Tulossa.
- 11) käytetään – Java, C++, mallin-
nus UML:llä.
- 12) Kyllä.
- 13) Ei.
- 14) Kyllä.
- 15) Kyllä ja kyllä.
- 16) Jonkin verran. suuret
ohjelmointityöt pyritään alihan-
kittamaan.
- 17) Jonkin verran.
- 18) Kyllä.
- 19) Ei varsinaisesti.
- 20) Kyllähän ne yrittää??

Mitkä ohjelmointikielät ja kehitysympäristöt ovat pääasiallisesti käytössä työympäristössäsi?

- 1) PL/SQL, Oracle iAS, Oracle
iDS, Oracle 9i-kanta.
- 2) Java, PL/SQL.
- 4) Oracle, COBOL, VB,
- 5) Oracle Developer, SQL, PL/
SQL, C ja C++.
- 6) Java, PL/SQL.
- 7) Java.
- 8) Oracle.
- 9) Java, C++, WebSphere, Cobol,
CICS.
- 10) PL/1, Cobol, Java on tulossa,
keskuskone, C/S (Advantage
tier).
- 11) Java (J2EE), C++.
- 12) Java, Visual Basic, PL/1, Cobol.
- 14) Kaikki yleisimmät.
- 15) Java, C++, JBuilder, MSVS.
- 16) Java, C, C++, Netbeans, Visual
C++, J Builder.
- 17) Microsoft Visual Studio (VB,
C++), J2EE, Oracle (PL/SQL,
Forms).

- 18) Oracle, DB2, PL/SQL, Java, C,
C++.
- 19) PL/1, Cobol Advantage Gen.

Oletko kiinnostunut ns. puhtaista oliokannoista (ei siis relaatiokannan laajennuksista)?

- 1) En.
- 2) En oikeastaan.
- 3) En.
- 4) En.
- 5) En.
- 6) En hirveästi.
- 7) En.
- 8) Enpä juuri.
- 10) En.
- 11) Kyllä.
- 12) En.
- 14) Aika vähän toistaiseksi.
- 16) Käsitteellisesti kyllä, tuotannol-
lisesti en...
- 17) Hiukan.
- 18) En.
- 19) En tällä hetkellä.
- 20) No, joo.

Jos sinulla on kokemusta ns. puhtaista oliokannoista, kuvaa lyhyesti kokemuksesi.

- 2) Ei käytännön kokemuksia.
- 5) Ei ole.
- 7) Ei kokemusta
- 11) Versant, In-memory database,
Toplink, erilaisiin käyttötarkoi-
tuksiin.
- 14) Ei ole.
- 16) Ei ole.
- 17) Ei ole.
- 20) Vähän tutustunut Yasmineen.
Kaupallis-hallinnollisissa
sovelluksissa vähän hyötyä. Re-
laatiomaailman ongelma on tie-
don toistaminen. Sama ongelma
joudutaan miettimään myös
oliomaailmassa ja -kannoissa.

Muita näkemyksiä?

- 2) SQL-standardi on jo liian monimutkainen (muistuttaa ADAa). Tarvittaisiin riittävän yksinkertainen standardi ja toteutus (olio-ohjelmointi yleistyi vasta Javan myötä, C++ ja Smalltalk olivat liian monimutkaisia). Lähitulevaisuudessa kehitystä tapahtuu varmaan EJB entity bean muiden oliomallien määppäämisessä relaatiokannan päälle.
- 6) Välineongelma.
- 11) UML <=> relaatiotietokanta 'mapping tool' eli esimerkiksi generointivälineiden käyttö relaatiotietokannan luomiseksi oliomallista
- 16) Mielipiteeni eivät välttämättä ole enää kovin valideja, koska en ole ko. asioiden kanssa enää juurikaan tekemisissä.
- 20) Ainakin Oracle on pilattu näillä "olio-ominaisuuksilla". TKHJ:stä on tullut tolkuttoman raskas. Luulen, että open source jyrää kaupalliset tietokannat.

Onko tämä selvitys tarpeellinen?

- 1) Mielelläni vastasin.
- 2) Ainakin mielenkiintoinen.
- 3) Kyllä.
- 4) En tiedä, riippuu tuloksista
- 7) En tiedä.
- 8) Ainakin mielenkiintoinen.
- 9) Mielenkiintoinen, ajankohtainen.
- 11) Juu ...
- 12) Kyllä.
- 15) Mielenkiintoinen.

- 17) Kyllä.
- 18) On.
- 20) Joo.

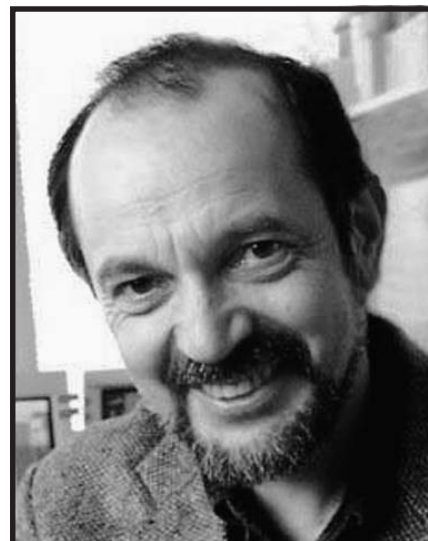
Yhteenveto

Tutkimuksen päätulos oli, että relaatiotietokantatuotteiden olio-ominaisuuksia ei juurikaan ole tarvittu tai osattu käyttää. Vaikka olio-ohjelmointi on levinnyt laajalle, tehdään tietokantojen käsittelyosuus perinteisesti SQL:n peruskäskyillä.

Toimittajat eivät myöskään nykyisin korosta tai markkinoi tietokantojensa oliopiirteitä, joskin ne käyttävät niitä sisäisesti omissa tuotelaajennuksissaan. Onko niin, että yksinkertaisesti vain vie aikaa omak-sua SQL:n oliopiirteitä vai onko nykyinen olio- ja relaatiomaailmojen jako pysyvä, jää nähtäväksi. Jos "puh-taat" oliotietokannat jossakin tulevaisuudessa yleistyvät, tilanne muuttuu.

Lähteet

- [1] Grimes, S.: "Modeling Object/Relational Databases", DBMS, April 1998. <http://www.dbmsmag.com/9804d13.html>
- [2] Melton, J.: "Advanced SQL:1999 - Understanding Object-relational and Other Advanced Features", Morgan Kaufmann Publishers, 2003.
- [3] Stonebraker, M.: "Object-Relational DBMSs—the Next Great Wave", Morgan Kaufmann Publishers, 1996.



Antoni Wolski toimii johtavana tutkijana Solid Information Technology Oy:ssä. antoni.wolski@iki.fi



Ari Hovi on tietokantoihin erikoistunut konsultti yrityksessään Ari Hovi Oy:ssä, www.arihovi.com. ari@hovi.pp.fi

RELA-kerhon kuulumisia

RELA-kerho on kaikkien relaatiotietokannoista kiinnostuneiden epämuodollinen keskustelu- ja opintopiiri, joka kokoontuu noin kerran kuussa.

Valitsemme jonkin teeman, jota käsittelemme eri näkökulmista useammassa kokoontumisessa. Tässä lehdessä on artikkeli relaatiotietokantojen oliolaajennusten käytöstä. Se syntyi koosterakenteet teemasta, johon keskityimme vuoden 2003 aikana. Nyt aiheenamme on ISO SQL:2003 standardi, jonka puitteissa olemme pohtineet mm. SQL-standardien kehittymistä, SQL/XML laajen-

nusta, DATE/TIME -tietotyyppien toteutusratkaisuja.

Systeemyölehden numerossa 4-2002 oli joukko artikkeleja teemasta ”oliosuunnittelusta hyvä relaatiotietokanta” ja Data Warehouse teemanumero 4-1999 oli tuotoksena ”tietovarastojen suorituskyky” teemasta.

Kerhon sähköpostijakelulistalle ilmoittautuminen tai kokouksiin osallistuminen ei velvoita mihinkään. Joukkomme koostuu sekä nuorista että vanhemmista asian harrastajista, jotka toimivat hyvin erilaisissa työ-

tehtävissä relaatiotietokantojen parissa. Ilmoittaudu sähköpostijakelulistalle lähettämällä osoitteella

`'sytyke-rela-kerho-request@pcuf.fi'`

viesti, jonka otsikkokentässä on sana 'subscribe' (ilman hipsuja). Listalta saat tiedot tulevista tapaamisista. Jos haluat tietää enemmän kerhon toiminnasta, niin lähetä viesti kerhon yhteyshenkilölle:

`marja.karmeniemi@oracle.com`.

*Marja Kärmeniemi,
Oracle Finland Oy*



-
-
n
-
-
n
n



Systemityö-yhdistys

Johtokunta 2004

sytyke-hallitus@pcuf.fi
http://www.sytyke.org

Helena Venäläinen

(puheenjohtaja)
FD Finanssidata Oy
Teollisuuskatu 1 B,
PL 308, 00101 Helsinki
Puh. 09-404 3690, 050-568 6690
helena.venalainen@osuuspankki.fi

Simo Vuorinen

(varapuheenjohtaja)
TietoEnator Oy
Kumpulantie 11, 00521 Helsinki
Puh. 040-524 8936
simo.vuorinen@tietoenator.com

Kati Ahlgrén

Nokia Oyj
Karaportti 2, 02610 Espoo
Puh. 050-486 0036
ext-kati.ahlgren@nokia.com

Petteri Holländer

SysOpen Oyj
Hiomotie 19, 00380 Helsinki
Puh. 0424 2020 457
petteri.hollander@sysopen.fi

Esko Marjomaa

Joensuun Yliopisto
Tietojenkäsittelytieteen laitos
PL 111, 80101 Joensuu
Puh. 013-251 7957
marjomaa@cs.joensuu.fi

Markku Niemi

STTF Oy
Tekniikantie 14, 02150 Espoo
Puh. 050-512 4687
markku.niemi@sttf.fi

Minna Oksanen

Cap Gemini Ernst & Young
Puh. 040-577 6640
minna.oksanen@cgey.com

Varajäsenet

Pirkko Leivo

Pohjolan Systemipalvelu Oy
00013 Pohjola
puh. 010 559 2786, 050-567 9352
pirkko.leivo@pohjola.fi

Erkki Pöyhönen

Nokia Research Center,
PL 407, 00045 NOKIA GROUP
puh. 0718 037 595, 050-483 7595
erkki.poyhonen@nokia.com

Web-toimikunta

sytyke-web-toimikunta@pcuf.fi

- Agnieszka Nummi (pj)
- Petteri Holländer
- Heini Holopainen
- Eija Kalliala
- Marzena Ksiazkiewics
- Lauri Laitinen
- Esko Marjomaa
- Juha Mäkeläinen
- Erkki Pöyhönen
- Anne Tilli

Liittokokousedustajat

- **Lauri Laitinen**
(lauri.laitinen@nokia.com)
- **Silja Räisänen** (silja.raisanen@pohjola.fi)
- **Simo Vuorinen**
(simo.vuorinen@tietoenator.com)
- **Helena Venäläinen**
helena.venalainen@osuuspankki.fi

SYTYKE-kerhot

- **Testauskerho (FAST)**
maaret.pyhajarvi@conformiq.com
- **JavaSIG**
simo.vuorinen@tietoenator.com
- **VVV-kerho (virkistys)**
marko.pudas@nordea.com
- **Relaatiokerho (RELA)**
marja.karmeniemi@oracle.com

Toimisto, sihteeri, Systemityö-lehden tilaukset ja talouden hoito

Susanna Koskinen (sytyke@hennax.fi)
Henrikintie 7 A, 00370 Helsinki
puh. 09-5607 5363, fax. 09-5607 5365

Jäsenyysasiat

Jäsenneksi liittymis- tai jäsentietojen
muutoslomakkeiden tilaus:
Tietotekniikan liitto ry./Piia Heiniö
Lastenodinkuja 1 A (5.krs),
PL 325, 00181 Helsinki
puh. 09-476 5800, http://www.tt-tori.fi

Systemityö-lehti

Julkaisija

Systemityöyhdistys SYTYKE ry
ISSN 1237-0525
11. vuosikerta no 3.
Painos 2500 kpl.
Paino: Tuokinprint Oy
Sirrikuja 4 C, 00940 Helsinki
puh. 09-342 4150
tuokinprint@co.inet.fi

Toimitussihteeri

Susanna Koskinen,
Toimistopalvelu Hennax Tmi
Tikkurikuja 10 T, 00750 Helsinki
puh. 09-5617 830
fax. 09-5607 5365
sytyke@hennax.fi

Päätoimittaja

Lauri Laitinen
Nokia Tutkimuskeskus
PL 407, 00045 Nokia Group
Itämerenkatu 11-13, 00211 Hki,
puh. 050-483 6551 (NRC)
Tornitasa 3 A 37, 02120 Espoo
lauri.laitinen@nokia.com

Toimituskunta 2/2004

Helena Venäläinen ja Erkki
Pöyhönen

Seuraava numero:

4/2004 SYTYKE 25-v.
Toimituskunta:
minna.oksanen@cgey.com ja
petteri.hollander@sysopen.fi

Tilaukset

Systemityölehti sisältyy yhdistyksen
Tietotekniikan liiton suositusten mu-
kaiseen yhdistyksen jäsenmaksuun,
joka vuonna 2004 on 62, 42, 25, 10
euroa/vuosi. Vuositilaus 20 euroa.
Irtonumerot 5 euroa. Hyvässä ajoin
ennen lehden painatusta tehty vähin-
tään 50 kpl:een lisätilaus 2 euroa/kpl.
Tilaukset yhdistyksen toimistosta.

Ilmoitushinnat

	mv	4 väriä
Takakansi	-----	1200 euroa
Sisäkannet (K1, K2)	-----	1000 euroa
Sisäsivut A4	400 euroa	800 euroa
Sisäsivut A5	200 euroa	600 euroa
Sisäsivut A6	100 euroa	-----

Vakiopaikan vähintään vuodeksi varanneille 20 % alennus.

Internetin WWW-sivut

http://www.sytyke.org
Systemityö-lehden sisällysluettelo
1988- ja useimmat artikkelit PDF-
muodossa.