

Timo Kaipainen

**Tietojärjestelmän suunnitteluteorian soveltuvuus projektin  
estimointi- ja mittaustyökalun  
kehittämiseen design-science näkökulmasta**

Tietojärjestelmätieteen  
kandidaatin tutkielma  
6.5.2008

Jyväskylän yliopisto  
Tietojenkäsittelytieteiden laitos  
Jyväskylä

## TIIVISTELMÄ

Kaipiainen, Timo Allan

Tietojärjestelmän suunnitteluteorian soveltuvuus projektin estimointi- ja mittaustyökalun kehittämiseen design-science näkökulmasta  
Jyväskylä: Jyväskylän yliopisto, 2008.

31 s.

Projektin estimointi- ja mittaus (PEM, Project Estimation and Measurement) on ilmiönä tutkijoita kiinnostava ala, koska ohjelmistoprojektien myöhästyminen ja budjetin ylittyminen aiheuttaa ongelmia organisaation sisällä. Kuten (Matikainen, 2006) toteaa, on näiden hallintatyökalujen (PEMT, Project Estimation and Measurement Tools) tutkimus jäänyt vähemmälle. Vaikka kaupallisia sovelluksia on tarjolla useita erilaisia ja eritasoisia, nämä eroavat huomattavasti toisistaan esimerkiksi toiminnallisuuksien ja hyödyllisyyden osalta (Matikainen, 2006). Tämä aiheuttaa organisaatioissa epätietoisuutta siitä, millainen työkalu olisi sopiva heidän tarpeisiinsa.

Tutkielman tarkoituksena on esitellä tietojärjestelmä tuetun projektin estimoinnin ja mittauksen menetelmiä lähinnä FiSMAssa kehitetyn konseptin, sekä Matikaisen Pro-gradu tutkielmassa (Matikainen, 2006) esitettyjen meta-vaatimusten ja tuotteensuunnittelu hypoteesien pohjalta.

AVAINSANAT estimointi, mittaus, projektinhallinta suunnitteluteoria, ISDT, PEM, PEMT

# SISÄLLYSLUETTELO

1 JOHDANTO.....	4
1.1 Ohjelmistoprojektin estimointi.....	4
1.2 Tutkimusongelma.....	6
2 TIETOJÄRJESTELMIEN SUUNNITTELUKEHYS (ISRF) JA TIETOJÄRJESTELMÄN SUUNNITTELUKEHYS (ISDT).....	8
2.1 Tietojärjestelmien suunnittelukehys (ISRF).....	8
2.2 Tietojärjestelmien suunnitteluteoria (ISDT).....	9
2.3 Design-science.....	12
2.4 Behavioral-science.....	13
2.5 Dual information systems.....	13
2.6 Emergent knowledge processes (EKP's).....	13
3 OHJELMISTOPROJEKTIN ESTIMOINTI- JA MITTAUSTYÖKALUT.....	14
3.1 FiSMAn konsepti organisaation oppimiseen ja prosessien parantamiseen.....	14
4 TUOTTEEN SUUNNITTELUHYPOTEESIT PROJEKTIN ESTIMOINTI JA MITTAUSTYÖKALULLE.....	17
4.1 Tuotteensuunnittelu hypoteesit.....	17
4.1.1 Toimintopisteet.....	18
4.1.2 Tilanne analyysi.....	20
4.1.3 Uudelleenkäyttö analyysi.....	21
4.1.4 Tuottoaste.....	22
4.1.5 Vertailututkimus (Benchmarking).....	23
4.1.6 Projektin estimointi- ja mittaustyökalun tunnistaminen (DPEMT)...	24
4.2 Tuotehypoteesien yhteenveto.....	25
5 JOHTOPÄÄTÖKSET JA YHTEENVETO.....	26
5.1 Organisaation tarpeet ja niiden huomioonottaminen.....	27
5.2 Yleisiä ongelmia ja näiden välttäminen.....	27
LÄHDELUETTELO.....	30

# 1 JOHDANTO

Ohjelmistoprojektien ongelmana on jo pitkään tunnustettu olevan myöhästymiset ja budjetin ylittyminen. Olennainen tekijä näiden välttämiseksi on projektin ennalta tapahtuva estimointi. Johtuen ohjelmien abstraktista luonteesta, on estimointi kuitenkin erittäin vaativa tehtävä, joka edellyttää järjestelmällistä tilanteen läpikäymistä, kokemusta ja hyvien työvälineiden tarjoamaa tukea. Työvälineiden on pystyttävä tarjoamaan selkeä kuva sen tuottamista tiedoista, niin että niitä voidaan käyttää projektinjohdon tukena (Kitchenham;Pickard;MacDonell;& Shepperd, 2001)

Finish software measurement association (FiSMA ry) on suomalainen järjestö joka keskittyy ohjelmisto-, järjestelmä- ja palveluprojektien ja -prosessien mittaamiseen ja niihin liittyvien standardien soveltamiseen ja kehittämiseen (FiSMA 2008)(FiSMA, 2008). FiSMA:n jäsenenä on lukuisia suuria ohjelmisto ja it-palvelutaloja sekä oppilaitoksia.

Pro-gradu tutkielmassa "Information system supported project estimation and measurement" (Matikainen, 2006) esittää osittaisen tietojärjestelmän suunnitteluteorian (ISDT, Information system design theory) projektin estimointi- ja mittaustyökalua (PEMT, Project estimation and measurement tool) varten. Tässä kuvataan tietojärjestelmän suunnitteluteorian ja tietojärjestelmän suunnittelukehyksen (ISRF) välinen yhteys. Suunnitteluteoria on myös yhteensopiva Dual information system (DIS) mallin kanssa, Dis-mallin ohjaamana on mahdollista rakentaa joustavia ja tehokkaita tietojärjestelmiä, jotka tukevat liiketoimintaprosesseja sekä niiden kehittämistä, vaikkei yhdelläkään sidosryhmällä olisi täydellistä tietämystä kaikista projektin estimointi- ja mittaustyökalun rakentamiseen vaadittavista osa-alueista.

## 1.1 Ohjelmistoprojektin estimointi

Yleisesti käytettyjä ja hyväksytyjä tapoja ohjelmistoprojektien estimoinnissa ovat toimintopisteiden (Function Point) laskeminen, minkä perustana ovat vaatimusmäärittelyt. Sekä tuottoaste (Delivery rate), jonka määritelmänä

pidetään tyypillisesti kuinka paljon aikaa yhden toimintopisteen tuottamiseen kuluu (h/fp). Myös projektikohtaiset tekijä, jotka vaikuttavat tuottavuuteen, pyritään ottamaan huomioon edellä mainitun lisäksi. Uudelleenkäytöllä ja uudelleenkäytettävyyshetodeilla on myös huomattava vaikutus ohjelmistoprojektin työmäärän arviointiin. Projektin estimointi ja mittaaminen on siis hyvin kriittinen tekijä realistisen aikataulun ja budjetin laadinnan kannalta, joka koostuu seuraavista edellä mainituista osioista:

- Funktionaalisesta koon määrittämisestä (Functional size measurement)
- Tuottoasteen arvioinnista (Delivery rate determination)
- Tilanne analyysistä (Situation analysis)
- Uudelleenkäytön vaikutusten arviointi (Reuse analysis)

Johtuen ohjelmistojen abstraktista rakenteesta ja muokattavuudesta, ohjelmistoprojektien vaatimukset voivat muuttua huomattavasti projektin aikana jolloin tarvitaan toimivaa projektin laajuuden-hallintaa (scope-management). Ohjelmistoprojektien mittaamisen tulisi olla standardoitua, jotta tiedot olisivat vertailtavissa niin eri projektien, kuin eri organisaatioiden ja yritysten välillä. Tästä johtuen projektin historiatiedon tallennus on tärkeä osa projektin estimointia ja mittausta (PEM), sekä olennainen osa projektin estimointi- ja mittaustyökalun ominaisuuksia. Historiatietokantaa voidaan käyttää kahteen tarkoitukseen, joko toimitusajan määrittämiseen tai prosessien parantamiseen. Prosessien vertailututkimuksessa pyritään kokemustietokannasta vertaamaan tietoja käynnissä olevaan projektiin ja näin parantamaan ohjelmistonkehitysprosessia. Esimerkkinä kokemustietokannoista voidaan mainita esimerkiksi FiSMA ry:n kehittämä Experience sekä ISBGin CDR10 -tietokanta.

Matikaisen (2006) mukaan projektin estimointi ja mittausprosesseja voidaan parantaa erilaisten tietojärjestelmien avulla. Nämä työkalut jakaantuvat viiteen

eri kategoriaan:

- Projektin koon laskeminen
- Uudelleenkäytön vaikutus
- Tietoisuus tuottoasteesta
- Tilanneanalyysi
- Vertailututkimus

Monet markkinoilla olevista työkaluista tukevat ainakin kahta näistä kategorioista, osa kaikkia viittä. Työkalun olisi kuitenkin pystyttävä vastaamaan mahdollisimman kokonaisvaltaisesti projektin mittaus- ja estimointi tarpeisiin ottaen huomioon organisaation erityisvaatimukset.

## 1.2 Tutkimusongelma

Tämän tutkielman tarkoituksena on luoda kokonaiskatsaus Matikaisen (2006) esittämään osittaiseen tietojärjestelmien suunnitteluteoriaan (ISDT) projektin estimointi -ja mittaus työkalua varten (PEMT). Useissa organisaatioissa on koettu tarpeita vastaavaan projektien estimointi- ja mittaus työkalun hankkimisen ja käyttöönoton olevan yleensä hyvin hankalaa, johtuen työkalujen ominaisuuksien erilaisuudesta sekä käyttötarkoituksista. Ongelmaksi voi myös muodostua työkaluista saatavan hyödyn välittömän näkyvyyden puuttuminen, jolloin näiden käyttöönotto herkästi tuomitaan epäonnistuneiksi.

Projektin mittaus- ja estimointi työkalun tulisi toimia läpinäkyvänä ja joustavana apuvälineenä ohjelmistoprojektien hallinnassa. Jotta työkalun käyttö olisi mahdollisimman tehokasta on sen käyttäjät sitoutettava ja koulutettava sen käyttöön. Ongelmaksi estimointi- ja mittaus työkalun käyttöönotossa koetaan tai saatetaan kokea myös kouluttautumisen ja omien projektien historiatietojen keräämisen aiheuttama resurssitarve. Hyvän työkalun tulisi siis olla helposti käytettävä sekä sen omaksuminen organisaatioon olisi tehtävä mahdollisimman helpoksi. Voidaan olettaa että hyvin moni hallintatyökalu lisää varsinaista

työmäärää projektinhallinnassa eikä alkuperäisen tarkoituksensa mukaisesti vähennä sitä. Tämä voi olla yksi tekijä miksi uusien työkalujen käyttöönotossa esiintyy ongelmia.

## 2 TIETOJÄRJESTELMIEN SUUNNITTELUKEHYS (ISRF) JA TIETOJÄRJESTELMÄN SUUNNITTELUKITEORIA (ISDT)

Sekä tietojärjestelmän suunnittelukehys (ISRF, Information system research framework) että suunnitteluteoria (ISDT, Information system design theory) pohjautuvat molemmat suunnitteluteorian konseptiin, minkä tarkoituksena on tuottaa entistä tehokkaampia ja tuottavampia tietojärjestelmiä. Tietojärjestelmän suunnitteluteorian (ISDT) päämääränä on rakentaa tietojärjestelmäartefakti määrittelemällä kernel-teoria, meta-vaatimukset, metasuunnittelu ja validoimalla artefakti nk. tuotteensuunnittelu hypoteeseilla. Puolestaan suunnittelukehys (ISRF) on jakaantunut behavioral- (ks. vasen KUVIO 1) ja design-scienceen (ks. oikea KUVIO1) .

### 2.1 Tietojärjestelmien suunnittelukehys (ISRF)

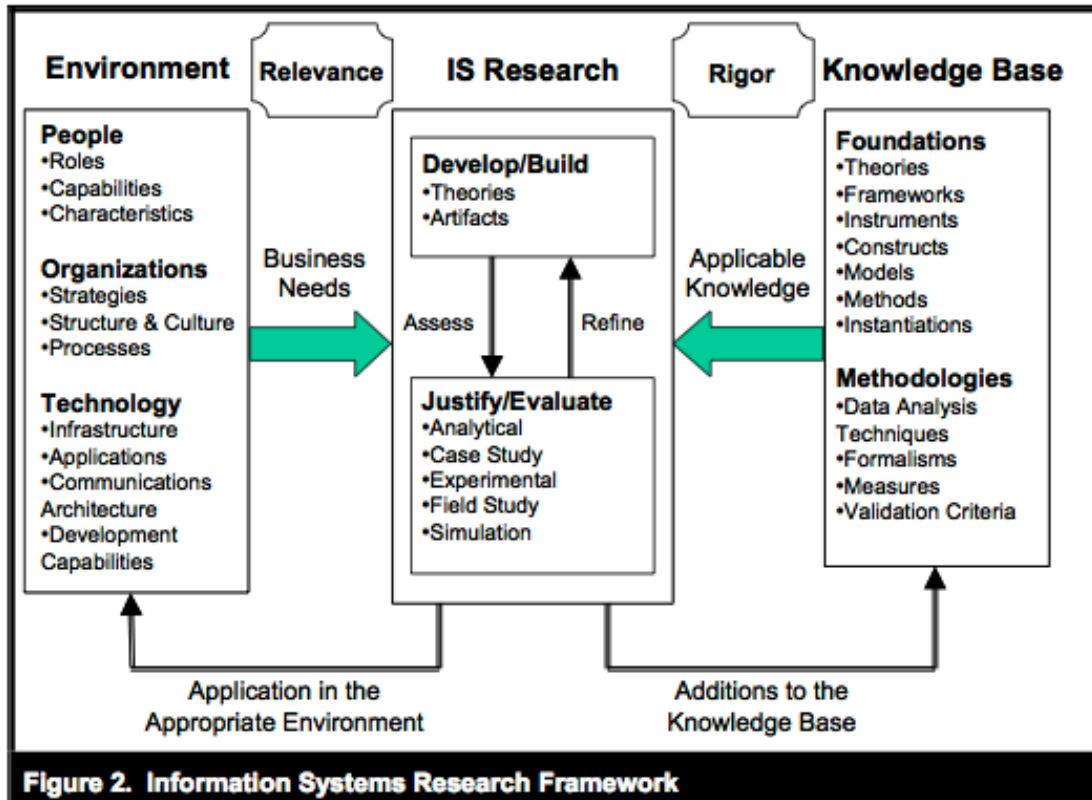
Hevner ym. (2004) esittävät kehyksen auttamaan tietojärjestelmien tutkimuksen ymmärtämistä, toteuttamista ja arviointia. Kehys jakaantuu desing-science ja behavioral science osioihin ympäristön mukaan (Knowledge base Environment). (Hevner;March;Park;& Ram, 2004) mukaan ympäristö määrittää ongelma-alueen. Tietojärjestelmien suunnittelussa tämä koostuu ihmisistä, organisaatioista ja olemassa olevista tai tulevista teknologioista.

Matikainen (2006) esittää, että liiketoimintatarpeet arvioidaan organisaation strategian, rakenteen, kulttuurin sekä olemassa olevien liiketoiminta prosessien kontekstissa. Tietojärjestelmien suunnittelu on jakautunut suunnittelukehyksen mukaan kahteen toisiaan täydentävään osaan: design- ja behavioral scienceen, jotka erottuvat (ks. KUVIO 1) Knowledge base (design science) ja Environment (behavioral science) osioissa.

Desing-science paradigman avulla luodut artefaktit voidaan testata behavioral science paradigmalla. Molemmat käyttävät samoja metodeja, myös design sciencessä voidaan käyttää empiirisiä menetelmiä, mutta ei niin kattavasti kuin behavioral sciencessä.



Edellä mainittu tietojärjestelmien suunnittelukehyksen tietämuspohja tuottaa materiaalia liiketoimintatarpeisiin tietojärjestelmä tutkimuksessa. Tietämuspohjaa ja toimintaympäristöä käytetään meta-vaatimusten ja meta-suunnittelun toteuttamiseen. Liiketoimintatarpeet puolestaan määrittelevät meta-vaatimukset, jonka jälkeen meta-suunnitelma voidaan rakentaa edellä määriteltyjen meta-vaatimusten pohjalta.



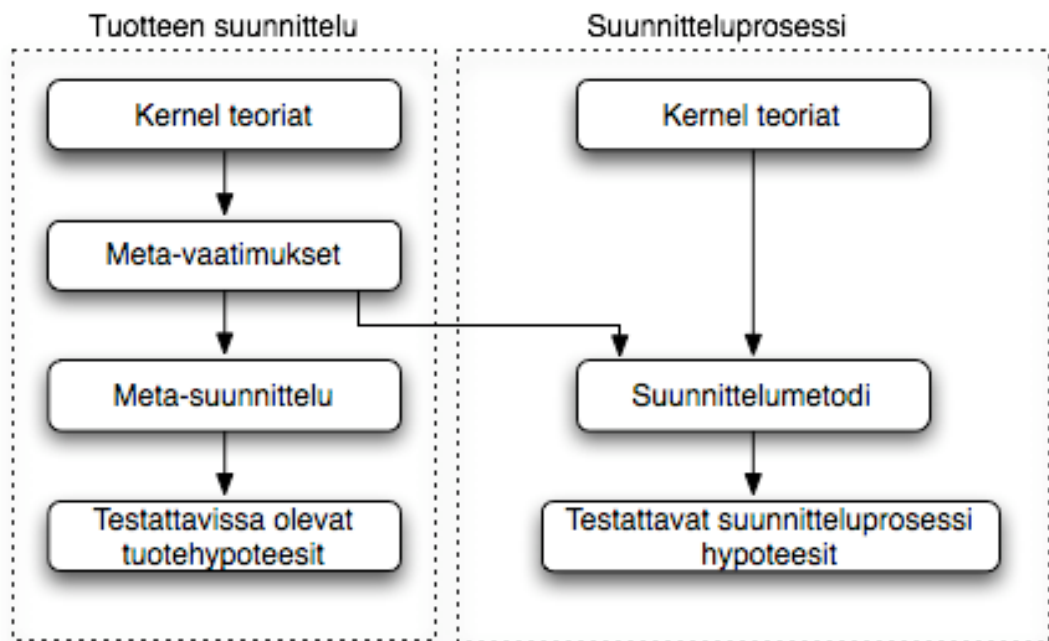
KUVIO 1 Information System research framework (Hevner;March;Park;& Ram, 2004)

## 2.2 Tietojärjestelmien suunnitteluteoria (ISDT)

Tietojärjestelmien suunnitteluteorian tarkoituksena on ohjata tietojärjestelmäartefaktien luomista. Nämä artefaktit käsittävät kaikki tietojärjestelmien kehityksessä syntyvät objektit, kuten lähdekoodin, sen sisältämät moduulit, luokat, dokumentaation ym. Termillä artefakti tarkoitetaan tässä yhteydessä erityisesti nk. IT-artefaktia, minkä Walls ym. (2004) määrittelevät seuraavasti tarkentaen vuoden 1992 määritelmäänsä;

“Artefakti on objekti, ja suunnittelun lopputulos, tässä tapauksessa suunnitteluteorian tuottama objekti”. Suunnitteluteorian tavoitteena on siis kuvailla IT-artefaktin ominaisuudet sekä metodit, jotta artefakti voidaan rakentaa. Lisäksi suunnitteluteoria parantaa sekä lopputuotteen laatua että prosessien suunnitelmallisuutta. Koska suunnitelmallisuus voidaan ymmärtää sekä verbinä, että adjektiivina, on myös suunnitteluteorian huomioitava nämä kaksi aspektia. Verbillä tarkoitetaan suunnittelua ja substantiivi puolestaan suunnittelun prosessin lopputulosta (artefaktia).

Lopputuloksena syntyvä artefakti vastaa määriteltyihin liiketoimintatarpeisiin, joten suunnitteluteoriolla on oltava testattavissa olevia hypoteeseja. Eräs esimerkki näiden testaamiseksi luoduista menetelmistä on Matikainen (2006) esittämät tuotteensuunnittelu hypoteesit, joiden validoimiseen ja testaamiseen palataan luvussa neljä. Kuten (Salo & Käkölä, 2005) toteavat tutkimuksessaan Nokia Oyj:n organisaatiolle, ovat tämän päivän vaatimukset ja paineet tuoda tuotteita markkinoille entistä nopeammin kasvaneet. Jotta yritys pystyisi säilyttämään ja kasvattamaan toimintaansa ja sen kannattavuutta on pystyttävä jatkuvasti etsimään uusia ja entistä kustannustehokkaampia tapoja toteuttaa ohjelmistoja, kuitenkin laadun kärsimättä. Tässä tilanteessa voidaan myös ottaa avuksi projektin estimointi- ja mittaustyökalu mikä on suunniteltu suunnitteluteorian pohjalta.



KUVIO 2 Tietojärjestelmän suunnitteluteorian komponentit

Tuotteen suunnitteluteorian lähtökohtana ovat meta-vaatimukset, jotka käsittävät niiden tavoitteiden joukon johon teorian on vastattava. Koska suunnitteluteoria ei vastaa vain yhteen ongelmaan vaan laajempaan kokonaisuuteen on syytä puhua joukosta. Meta-suunnittelu puolestaan pyrkii täyttämään meta-vaatimuksissa asetetut tavoitteet artefaktien avulla. Kolmas komponentti tuotteensuunnittelussa on ns. kernel-teoriat, jotka kattavat suunnitteluvaatimukset luonnon- tai yhteiskuntatieteen pohjalta. Kernel-teoriat toimivat "lähtökohtina" tietojärjestelmän suunnitteluteorian komponenttien välisissä yhteyksissä (ks. KUVIO 2). Lopputuloksena syntyneen artefaktin soveltuvuutta täyttämään meta-suunnitelman vaatimukset voidaan testata ns. tuotehypoteeseilla, joilla konkreettisesti todennetaan teorian toimivuus. (ks. luku 4)

Varsinaisen artefaktin rakentamisen suunnitteluprosessin kattamiseksi on esitetty vastaavanlainen kolmiportainen järjestelmä kuin edellä kuvailtuun tuotteensuunnitteluun. Suunnittelu metodin tarkoituksena on kuvata

tarvittavat toimenpiteet halutunlaisen artefaktin luomiseksi, sitä puolestaan täydennetään kernelteorioiden luonnon- ja yhteiskuntatieteellisellä pohjalla, kuten em. tuotteensuunnittelussakin. Myös suunnitteluprosessin teorioita voidaan ja tulee testata, joten tämän toteuttamiseksi voidaan luoda hypoteeseja suunnitteluprosessia varten joilla voidaan todentaa, että suunnitteluprosessi ja sen suunnittelumetodi tuottaa meta-suunnittelussa määritellyn artefaktin.

### **2.3 Design-science**

Seuraavassa esitetään Hevner ym. (2004) määritelmä tietojärjestelmien tutkimuskehykselle sekä design- ja behavioral sciencen erot. Tietojärjestelmien tarkoituksena organisaatioissa on useimmiten parantaa tehokkuutta ja tuottavuutta. Tavoitteiden saavuttamiseen vaikuttavat niin itse tietojärjestelmä, organisaation piirteet, työtavat, kehitys kuin näiden implementointimetodit yhdessä. Tiedon hankkimiseksi näistä on erotettavissa, Hevner ym. (2004) mukaan kaksi erilaista paradigmaa; behavioral science ja design science näkökulmat. Design-science lähestymistapa on ongelmanratkaisukeskeinen ja insinööritieteellinen. Design-science pyrkii luomaan innovaatioita jotka määrittelevät ideoita, käytänteitä, teknisiä mahdollisuuksia ja tuotteita joiden avulla analysointi, suunnittelu, toteutus, hallinta ja tietojärjestelmien käyttö voidaan suorittaa. Design-science on siis pohjimmiltaan ongelmanratkaisuprosessi. Tätä prosessia ohjaamaan on kehitetty (Hevner ym. 2004) seitsemänosainen ohjenuora, joka on esitetty seuraavassa:

1. Suunnittele artefakti
2. Ymmärrä ongelman merkitys
3. Tee suunnitelman arviointi
4. Tutkimuksen panos
5. Muista tutkimuksen kurinalaisuus
6. Tee suunnitelma etsimis prosessina
7. Muista tutkimuksen kommunikointi

Näistä erityisesti viimeinen kohta on erityisen tärkeä, koska ilman kaikkien sidosryhmien ymmärtämistä, toimii prosessi vain osittain, tai ei ollenkaan.

## **2.4 Behavioral-science**

Behavioral science keskittyy enemmän organisaation ja inhimillisen käytöksen selittämiseen analyysin, suunnitteluun sekä toteuttamiseen johtamis- ja tietojärjestelmien käytössä. Behavioral-science toimii tärkeänä palautemekanismina tietojärjestelmien suunnittelukehyksessä (ISRF) syöttäen tietoa reaali maailman olosuhteista ja vaatimuksista kehitysprosessiin. Tietojärjestelmän suunnitteluteorian tuottaman artefaktin validointi suoritetaan tietojärjestelmien suunnittelukehyksen syklisessä mallissa.

## **2.5 Dual information systems**

Dual information systems (DIS) mallin hyödyntäminen projektin estimointi- ja mittaustyökalun rakentamisessa on aiheellista kun eri osapuolet (engl. Agents) eivät ymmärrä kaikkia tekijöitä mm. vertailututkimuksesta (Matikainen 2006). Koska DIS malli keskittyy osapuolten oppimisen ja työskentelyn tallentamiseen, on huomioitava että yhteyksiä voidaan löytää myös PEMT työkalun suunnitteluteorian sekä DIS mallin suunnitteluteorian välillä.

## **2.6 Emergent knowledge processes (EKP's)**

Eräänä suunnitteluteoriaan liittyvänä lisäkohtana voidaan mainita lyhyesti nk. Emergent knowledge processes. Kuten edellä on todettu, projektin estimointi- ja mittaustyökalun toimintaolosuhteet voivat olla hyvinkin erilaiset riippuen organisaatiosta, joten yksilöidyn työkalun rakentaminen on jokaisella kerralla ainutkertainen prosessi. Emergent knowledge process tarkoittaa prosesseja (Markusym. 2002), joiden tarkoituksena on auttaa tietämyksen kehittymistä organisaatioissa uusien asioiden ja käytänteiden suhteen.

### **3 OHJELMISTOPROJEKTIN ESTIMOINTI- JA MITTAUSTYÖKALUT**

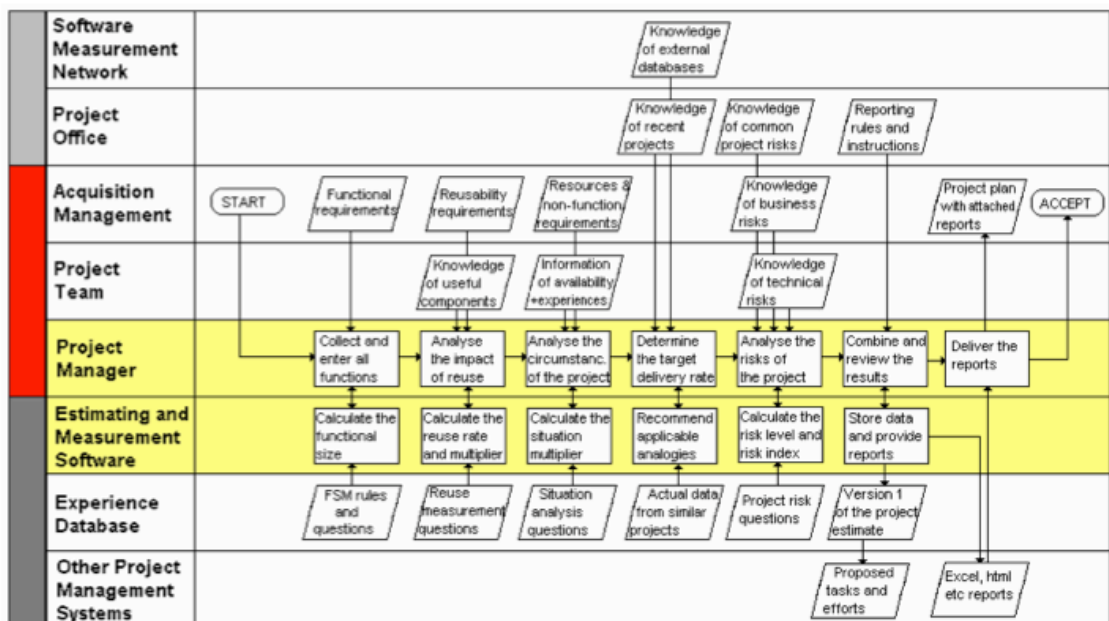
Kuten johdannossa on todettu, ohjelmisto projektien estimointiin ja mittaukseen on kehitetty runsaasti työkaluja (kuten esimerkiksi Microsoft project) joiden menetelmät ja sisältö vaihtelevat suuresti. Tässä tutkielmassa keskitytään FiSMAn kehittämään konseptiin. Olennainen tekijä mittaustyökalun toiminnassa ja sen antamien estimaattien tarkkuudessa on käytettävissä oleva historiatieto edellisistä projekteista. FiSMAn kehittämässä järjestelmässä tämä historiatieto on tallennettuna Experience tietokantaan. Kuten Käkölä ym. (Käkölä; Wu; Yalaho; & Nahar, 2006) toteavat, ohjelmistoteollisuuden kansainvälistyessä on myös tämän asettamat vaatimukset otettava huomioon työkalujen suunnittelussa. Perinteisien sähköpostin ym. Lisäksi kommunikoinnissa esimerkiksi eri toimipisteiden välillä voidaan käyttää erilaisia video-neuvottelu sovelluksia, jotka täydentävät omalta osaltaan projektin kokonaisvaltaista hallintaa estimointi- ja mittaustyökalun lisäksi. Luvussa 3.1 kerrotaan pääpiirteet FiSMAn konseptista ja sen sisältämistä Organisaation oppimisen konsepteista, Fisman aktoreista, Projektin työmäärän estimoinnin, Projektin estimointi ja mittaus prosessin ja Fisman mittausmenetelmät. Matikaisen (2006) Pro-gradu työssä näiden pohjalta on määritelty meta-vaatimukset projektin estimointi ja mittaustyökalulle.

#### **3.1 FiSMAn konsepti organisaation oppimiseen ja prosessien parantamiseen**

Fisman kehittämä konsepti organisaation oppimiseen ja prosessien parantamiseen käsittää seuraavat osapuolet projektin estimoinnissa ja mittauksessa ohjelmiston tuottajan puolella: Ylempi johto, projektitoimisto sekä projektin johto. Tässä luvussa käydään seikkaperäisesti lävitse Matikaisen (2006) identifioimat FiSMAn menetelmien tuottamat metavaatimukset projektin mittaus- ja estimointityökalulle.

Kaikkien PEMT-artefaktien tulee tukea verkoston rakentamista tiedon keruuta varten projekteista, sekä pystyä jakamaan kokemustietokannasta uusia versioita. Oleellinen osa FiSMAn menetelmää on työmäärän laskeminen jota työkalun on tuettava seuraavin osin: Toiminnallinen koon mittaaminen, uudelleenkäytön analyysi, tilanne analyysi ja tuottoaste joka pohjautuu kokemustietokannan tietoihin. Työkalun tulee luonnollisesti myös pystyä suorittamaan estimointi ja analyysi em. mainittujen tekijöiden avulla sekä tallentamaan tästä analyysistä saadut tiedot kokemustietokantaan.

Laajuuden hallintaa (scope management) tukevat Matikaisen (2006) esittämät viisi prosessia työkalun kehittämiseen suunnitteluteorian pohjalta ovat; 1) projektin asettaminen, jota seuraa 2) kustannusten ja keston arviointi sen jälkeen siirrytään itse 3)kehitysvaiheeseen jonka aikana 4) edistymisen valvonta ja muutosten hallinta vuorottelevat. Kehitysprosessin lopussa on vielä 5)kehityksen lopettaminen. Meta-vaatimuksina projektin asettamis prosessille ovat metodien ja mallien valinta, luokittelevien kysymysten esittäminen.



KUVIO 3 FiSMA effort esimation process (Matikainen, 2006)

Kustannusten ja keston arviointia pidetään usein projektin estimointi- ja mittaustyökalun tärkeimpänä tehtävänä, joten myös FiSMAn konseptissa

estimointimenetelmien tuelle tälle annetaan verrattain suuri painoarvo. Estimointiprosessi alkaa toiminnallisen koon laskemisella, sen lähteenä ovat vaatimusmäärittelyt, toimintoluettelon pohjalta arvioidaan uudelleenkäytön vaikutukset ja näiden kahden tekijän pohjalta otetaan vielä huomioon tilanneanalyysi perustana mm. saatavilla olevat resurssit ja tuotteen ei-toiminnalliset vaatimukset. Viimeisenä estimointiprosessin vaiheena on tuottoasteen päättelemisen kokemustietokannan ja/tai muun ulkoisen historiatietokannan avulla ja tämän tiedon yhdistäminen muihin arvioihin.

Muutoksen hallinta on erittäin tärkeää projektissa, joten työkalun tulisi täyttää myös seuraavat meta-vaatimukset: muutoksen hallinta, version hallinta, valmiusasteen arviointi (muutos tilanteissa uudelleenlaskenta) ja lisäksi tukea historiatiedon tallentamista kokemustietokantaan tulevia projekteja varten. Työkalun on selvittävä myös muutosten vaikutusten arvioinnista tai uusien vaatimusten tuomisesta projektiin, pystyvä vertailemaan nykyistä ja muutettua versiota projektista sekä tuottamaan näistä asianmukaiset raportit sekä laskemaan uuden version arviosta pohjautuen em. tekijöihin.

Projektin lopettaminen on varsinkin asiakkaan kannalta erittäin tärkeä prosessi johon Matikainen (2006) ehdottaa työkalun ominaisuuksiksi mm. Seuraavia asioita: toiminnallisuuksien lukitseminen ja viimeisimmän version määrittelyt, viimeiset tuottavuus ja toimitusajan pituus analyysit sekä loppuraportin tuottaminen.



## 4 TUOTTEEN SUUNNITTELUHYPOTEESIT PROJEKTIN ESTIMOINTI JA MITTAUSTYÖKALULLE

Jotta suunnitteluteorian toimivuus voitaisiin todentaa, on luotava hypoteeseja ja todistuksia näille hypoteeseille, että suunnitteluteorian noudattaminen todella avustaa tehokkaan projektin estimointi ja mittaus työkalun kehittämisessä. Sekä tutkittava niiden avulla toimisivatko nämä käytännössä. Tässä tutkielmassa käytetään lähtökohtana Matti Matikaisen (2006) luomia tuotteensuunnittelu hypoteeseja. Koska hypoteesien todentaminen todellisessa ympäristössä on erittäin tärkeää, luodaan tässä tutkielmassa perusta hypoteesien toiminnan testaukselle esimerkiksi haastattelututkimuksen avulla. Luvussa myös esitellään ja täydennetään Matikainen (2006) esittämiä tuotteensuunnittelu hypoteeseja, ja pyritään antamaan näin pohja tulevaisuuden validointitutkimukselle. Hypoteesit on seuraavissa kappaleissa purettu osiin ja näiden avulla pyritty avaamaan mahdolliset vuorovaikutus suhteet näiden sisällä. Esitetyt hypoteesit pohjautuvat FiSMA:n kehittämiin menetelmiin ja näiden arvioinnissa on käytetty hyväksi FiSMA:n tarjoamia artikkeleita ja raportteja.

Eräs tuotehypoteesien avulla tässä tutkimuksessa esille noussut asia on laajuudenhallinnan (scope management) tarve ohjelmistoprojekteissa, mihin em. Esitetyt tuotteensuunnittelu hypoteesit eivät ota kantaa millään tavalla. Laajuudenhallintaan FiSMA on myös esittänyt oman konseptinsa, uusimpana northernScopekonseptin mikä on esitelty ja otettu käyttöön organisaatioissa kevään ja kesän 2007 aikana ja jonka ensikokemukset ovat olleet lupaavia (FiSMA, 2007)

### 4.1 Tuotteensuunnittelu hypoteesit

Matikainen (2006) jakaa tuotteensuunnittelu hypoteesien ryhmät kuuteen eri kategoriaan, toimintopisteisiin, tilanne analyysiin, uudelleenkäyttö analyysiin, tuottoasteeseen, vertailututkimukseen ja projektin estimointi- ja

mittaustyökalun tunnistamiseen. Kaikkien näiden hypoteesien noudattamisen tavoitteena on tuottaa artefakti, mikä noudattaa ainakin seuraavia Matikaisen määrittelemiä piirteitä: standardoitu-, tarkka-, jäljitettävä- ja läpinäkyvä prosessi tai käytäntö. Seuraavassa käydään lävitse nämä tuotteen suunnitteluhypoteesit kriittisesti tarkastellen niitä erityisesti design-science näkökulmasta. Hypoteeseista lähes kaikki ovat syklisiä ja tässä tutkielmassa olen pyrkinyt havainnollistamaan hypoteesien sisältöä ja yhteyksiä kaavioilla, joissa on kuvattu artefaktin ominaisuuksia parantavat tekijät + merkillä ja huonontavat - merkillä, lisäksi hypoteesien osien välillä olevat nuolet osoittavat mihin muihin osiin hypoteesia kyseinen ominaisuus vaikuttaa.

#### **4.1.1 Toimintopisteet**

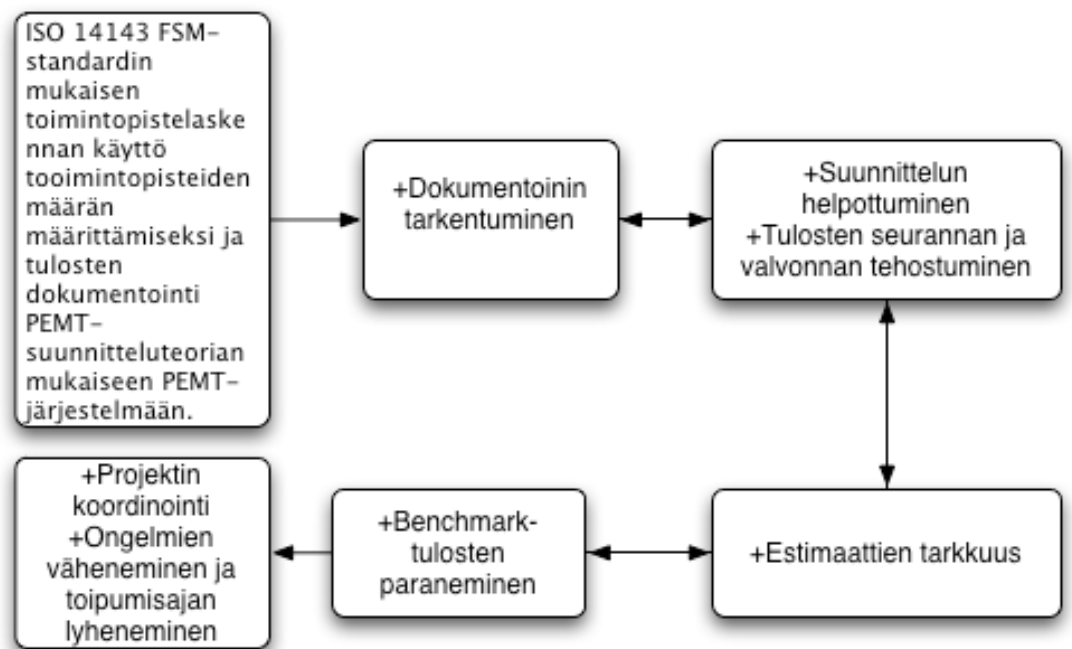
Toimintopisteiden (FSM) laskemisella pyritään saavuttamaan yhteneväinen laskutapa ohjelmiston koolle, jonka perusteella voidaan tehdä erilaisia arvioita mm. ohjelmistoprojektin kestosta. Kuten (Tran-Cao;Levesque;& Abran, 2002) toteavat, että toimintopisteiden laskentaan on olemassa perinteisesti kaksi erilaista lähestymistapaa: 1. Koodirivien (LOC) määrä ja 2. erilaisten ominaisuuksien, kuten funktioiden tai metodien määrä ja monimutkaisuus. Näistä koodirivien määrä on yksinkertaisin sekä jälkikäteen tehtynä tarkka, mutta ennalta tehtynä mahdoton arvioitava sekä vaikeasti yleistettävä johtuen esimerkiksi erilaisten ohjelmointikielien syntaksista eroista. Puolestaan ohjelmiston ominaisuuksiin ja toiminteisiin (kohta 2.) nojaava toimintopisteiden laskenta on huomattavasti tarkempi, mut(Walls;George R;& Omar A El, 2004)ta vaati myös paljon enemmän paneutumista jokaisen toiminnallisuuteen, sekä määritellyn tavan laskea tietyn monimutkaisuuden omaavien ominaisuuksien toimintopistemäärä. Esimerkiksi FiSMA:n kehittämä FSM metodi pohjaa jälkimmäisenä mainittuun menetelmään (FiSMA, 2006) ja sillä on saatu erinomaisia tuloksia osassa sitä käyttäneissä organisaatioissa.

Toimintopisteisiin perustuva projektin koon arviointi vaikuttaa positiivisesti projektin dokumentaatioon, mahdollistamalla tarkemman

dokumentaation projektin etenemisestä (KUVIO 4). Projektin toteuttamiseen vaadittavan ajan estimaatit myös tarkentuvat, kun toimintopisteiden avulla määritellään tarkasti kuinka

paljon resursseja jokaisen toiminnallisuuden toteuttaminen vie. Puolestaan kun tiedetään tarkemmat arviot toiminnallisuuksien toteuttamiseksi, myös projektin kokonaisvaltainen suunnittelu helpottuu. Tulosten ja projektin tilan seurannan ja valvonnan tehostuminen on seurausta paremmin dokumentoidusta ja hyvin määritellyistä toteutusajoista kullekin toiminnallisuudelle, jolloin tiedetään jatkuvasti projektin tila sekä voidaan puuttua mahdollisiin epäkohtiin ajoissa.

Edellä mainitut tekijät puolestaan kokonaisuudessaan lisäävät projektista tehtävien (mm. kesto- ja kustannusestimaattien) estimaattien tarkkuutta (KUVIO 4). Puolestaan kun projektin suoritusarvoja myöhemmin vertaillaan toisiin projekteihin (Benchmarking) hyvin toteutetulla toimintopisteiden määrittelyllä projektissa saadaan vertailutukimusta tarkempia arvioita. Projektin koordinoinnin, eli sen miten projektia kokonaisuudessaan ohjataan, sekä ongelmista toipumiseen kuluvan ajan väheneminen ovat selkeitä etuja, jotka polveutuvat edellä mainituista hypoteesin osista.



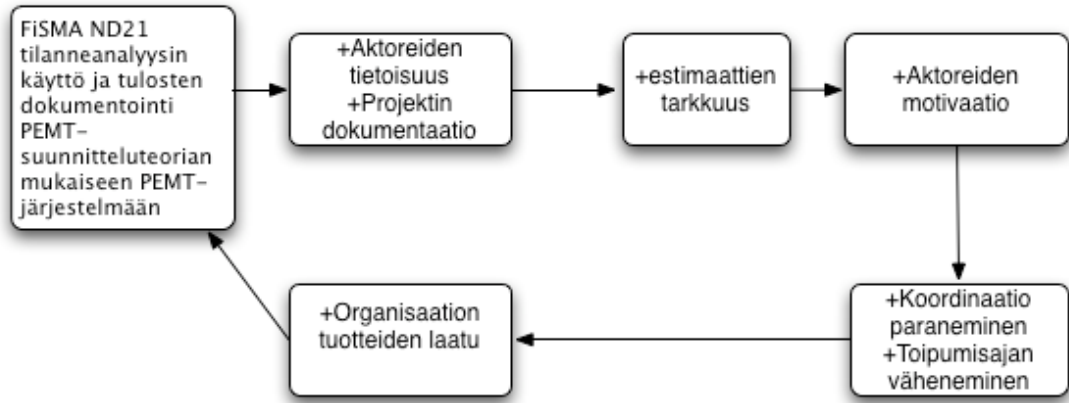
KUVIO 4 Funktionaalisen koon laskennan tuotehypoteesit (Matikainen 2006)

#### 4.1.2 Tilanne analyysi

Tilanne analyysin tarkoituksena on lisätä aktoreiden tietoisuutta projektin kulusta (mm. resurssien allokoinnista, aikataulun pitävyydestä ja mahdollisista ongelmista) joka auttaa projektinjohtoa tekemään olosuhteita korjaavia toimenpiteitä sekä parantaa projektin dokumentaatiota, jonka kautta myös motivaatio kasvaa (Matikainen 2006). Tässä tutkielmassa tilanneanalyysin hypoteesi pohjaa FiSMA:n kehittämään Situation Analysis ND21 metodiin (FiSMA, 2003). Tämä metodi koostuu 21 erilaisesta tuottavuustekijästä, joista jokainen on tasapainotettu kokemustietokannan perusteella. FiSMA:n tilanneanalyysi metodi on pyritty tekemään mahdollisimman helppokäyttöiseksi, ja em. 21 tuottavuustekijää on ryhmitelty neljään eri kategoriaan: Projekti, Prosessi, Tuote ja Ihmiset. Näitä 21 tekijää arvioidaan yksinkertaisella ++,+,-,-,- asteikolla missä ++ on paras mahdollinen tilanne ja - huonoin mahdollinen.

Tilanne analyysin pohjalta voidaan esimerkiksi havaita työvoiman riittämättömyys/liiallisuus tai työkalujen sopimattomuus kyseiseen projektiin

(KUVIO5). Tilanne analyysin tarkoituksena on ennekaikkea toimia ennalta ehkäisevänä ja analyyttisena työkaluna projektin johdolle (Matikainen 2006).



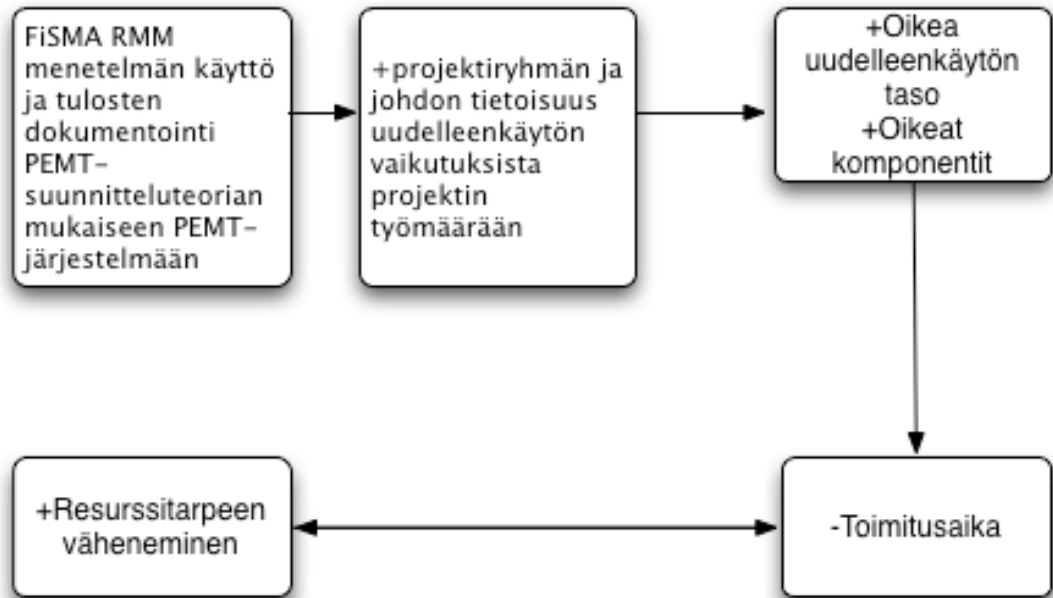
KUVIO 5 Tilanneanalyysin tuotehypoteesit (Matikainen 2006)

### 4.1.3 Uudelleenkäyttö analyysi

Ohjelmistojen kehityksessä komponenttien uudelleenkäytöllä voidaan aikaansaada huomattavia säästöjä niin ajankäytössä kuin rahallisestikin (FiSMA, 2002). FiSMA on luonut uudelleenkäytön estimointiin oman metodinsa (FiSMA Reuse Measurement Method, FiSMA RMM version 2002), monet organisaatiot ovat pyrkineet em. syistä nostamaan uudelleenkäytön tasoa, mutta kunnollista uudelleenkäyttö estimointi menetelmää ei ole ollut saatavilla. Menetelmä pohjautuu toimintopisteiden sekä neljän toimitettavan osa-alueen (Ohjelmakoodi, testitapaukset, ohjelmiston dokumentaatio ja loppukäyttäjien dokumentaatio) arviointiin esitetyin oletuspainotuksin.

Uudelleenkäyttöä suunnitellessa on tärkeää pystyä tiedostamaan mitä hyötyjä ja haittoja uudelleenkäyttö aiheuttaa ohjelmistoprojektille. Ei ole itsestään selvää, että kaikki uudelleenkäyttö vähentäisi esim. resurssitarpeita. Uudelleenkäytön tässä esitetyt tuotehypoteesit auttavat projektin johtoa määrittämään oikean uudelleenkäytön tason sekä valitsemaan oikeat komponentit mitä voidaan uudelleenkäyttää. Oikein toteutetulla uudelleenkäytöllä projektin työmäärää ja toimitusaikaa voidaan vähentää huomattavasti, tosin epäonnistuessaan tämä voi muuttua päinvastaiseksi (KUVIO 6). Keskustelu ja systemaattinen läpikäynti auttavat maksimoimaan

hyödyllisen komponenttien käytön. Uusien komponenttien tuottaminen muuttuu myös entistä suunnitelmallisemmaksi (Fan & Leung 2002).

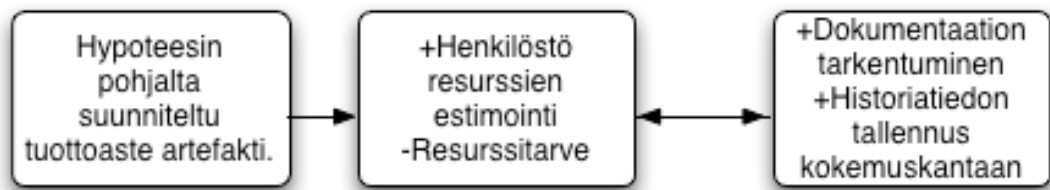


KUVIO 6 Uudelleenkäytön tuotehypoteesi (Matikainen 2006)

#### 4.1.4 Tuottoaste

Tuottoasteella tarkoitetaan aikaa mikä kuluu projektin tietyn osan valmistamiseen, esimerkiksi 10 toimintopistettä kuukaudessa / sovelluskehittäjä. Tässä erityisen tärkeäksi nousevat kaikki aikatauluun vaikuttavat tekijät, kuten henkilöstö resurssien määrä ja saatavuus.

Tuottoasteen estimoinnissa esim. tieto henkilöresurseista on olennainen, joten näiden historiatiedon hyödyntäminen estimoinnissa on erittäin olennaista (Matikainen 2006). Tuottoaste voidaan siis ilmaista esimerkiksi työtunneilla / toimintopiste. Tuottoaste on hyvin riippuvainen muista projektin osa-alueista, kuten lopputuotteen vaatimat muutokset luonnollisesti vaikuttavat suoraan tuottoasteeseen, joka on tarpeen mukaan laskettava uudestaan.



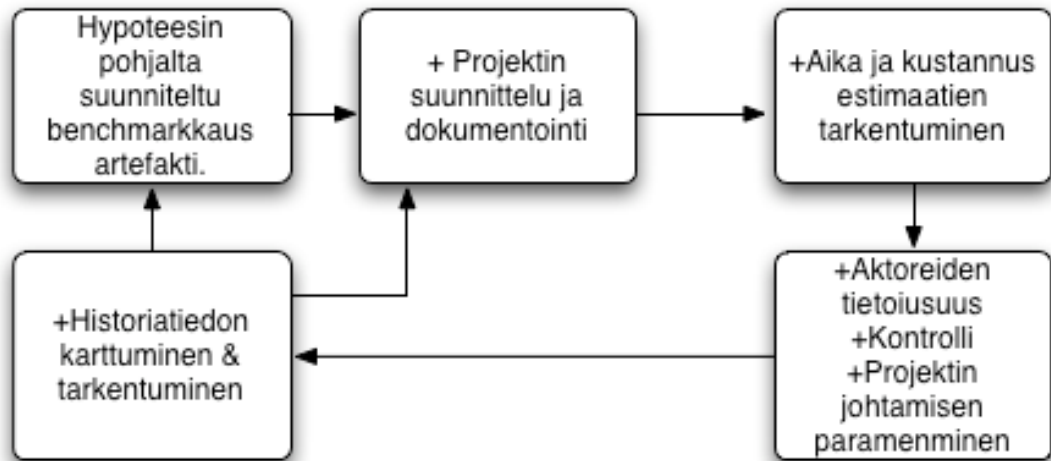
KUVIO 7 Tuottoaste tuotehypoteesi (Matikainen 2006)

#### 4.1.5 Vertailututkimus (Benchmarking)

Vertailututkimuksen tavoitteena on vertailla projektin suoritusarvoja edellisiin projekteihin ja näin saada palautetta sekä ohjausta projektin johtoon ja estimointiin. Vertailututkimuksen avulla tulevat aika- ja kustannus estimaatit saadaan tarkemmiksi, edellyttäen että historiatietoa on tarpeeksi saatavilla ja että se on hyvin heterogeenistä. Tärkeä osa vertailututkimusta on vertailudatan kerääminen, esimerkiksi ISBSG ja suomalainen FiSMA ovat kehittäneet menetelmiä omien kokemustietokantojensa pohjalle.

Vertailututkimuksen avulla voidaan asettaa tavoitteita prosessien parantamiselle, jolla voidaan parantaa yrityksen kilpailukykyä markkinoilla. Vertailututkimusta on muilla insinöritieteellisillä aloilla käytetty jo pitkään hyväksi projektien keston estimoinnissa, mutta vasta nyt ohjelmistoteollisuus on löytänyt sen tuomat edut (Lokan;Wright;Hill;& Stringer, 2001)

Kuten seuraavassa vertailututkimus tuotehypoteesia kuvaavassa kaaviossa (KUVIO8) on esitetty, hyödyn saaminen liittyy kiinteästi varsinaista projektia edeltäviin valmistelutoimiin, mutta vertailtava data on oltava olemassa etukäteen, joko ulkoisista (esimerkiksi FiSMAn Experience-tietokanta) tai sisäisistä lähteistä. Kattavin vertailututkimus voidaan tehdä, kun projekti on loppunut tai loppumassa, jolloin projektin suoritusarvot ovat lopulliset



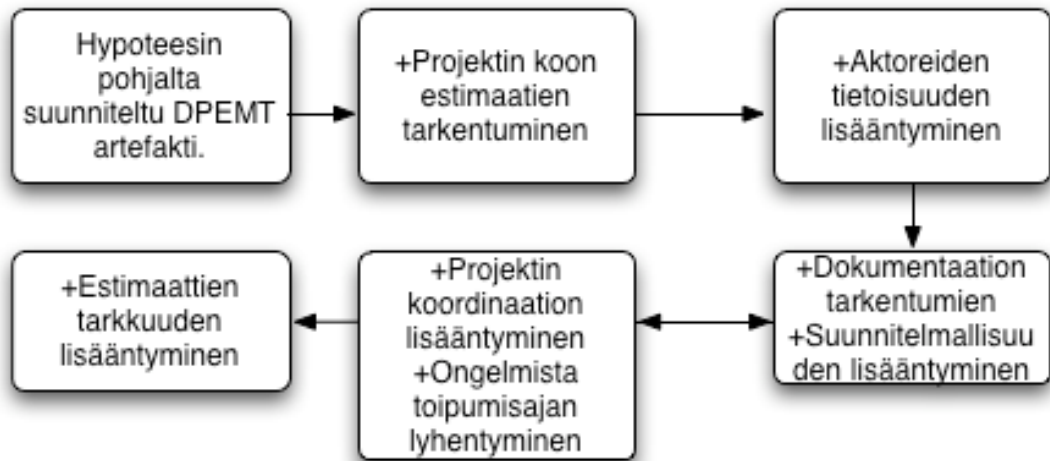
KUVIO 8 Vertailututkimus tuotehypoteesi (Matikainen 2006)

#### 4.1.6 Projektin estimointi- ja mittaus työkalun tunnistaminen (DPEMT)

DPEMT (Dual project estimation and measurement tool) toimii kuten aiemmin on kuvailtu Dual Information Systems-mallin esittelyn yhteydessä, tilanteissa joissa eri sidosryhmien tietoisuus projektin osatekijöistä ei ole täydellistä. Ongelmana ohjelmistoprojekteissa on usein eri sidosryhmien väliset ymmärtämys ja kommunikaatioerot (Keung;Jeffery;& Kitchenham, 2004) Erityisesti projektien jakaantuessa entistä laajemmalle niin tietämyksellisesti kuin maantieteellisestikin, erilaisten ryhmien edustajien tieto esimerkiksi teknisistä ratkaisuista voi olla hyvin pientä, tai sitä ei ole ollenkaan.

Hypoteesin pohjalta suunniteltu artefakti lisää projektin koon ja keston estimaattien tarkkuutta, jolloin myös sidosryhmien ja aktoreiden tietoisuus projektin kulusta ja tilasta kasvaa. Kun puolestaan sidosryhmät ovat tietoisia projektin kulusta voidaan myös dokumentaatiota rakentaa vastaamaan heidän nykyisiä ja tulevia tarpeitaan vastaavaksi. Kun projektin estimaatit ovat tarkkoja, ja dokumentaatio on hallittu oikein on myös ongelmista toipumiseen kuluva aika lyhyempi. DPEMT-työkalulla koko hypoteesin tarkoituksena on olla itseään iteraatio toisensa jälkeen parantava, jolloin viimeisenä kohtana on estimaattien tarkentuminen.





KUVIO 9 DPEMT tuotehypoteesi (Matikainen 2006)

## 4.2 Tuotehypoteesien yhteenveto

Näiden edellä mainittujen hypoteesien avulla voidaan validoida Matikainen (2006) esittämän osittaisen suunnitteluteoria projektin estimointi- ja mittaustyökalulle. Keskeisimpinä hypoteeseina pitäisin Toimintopiste-, ja vertailututkimus hypoteeseja joiden merkitys projektin estimoinnille ja mittaamiselle on erittäin suuri ja perustavanlaatuinen.

Edellä mainitut tuotteensuunnitteluhypoteesit ovat suunniteltu validoimaan osittaista (Matikainen 2006) esittämää suunnitteluteoriaa, näiden täydentäminen vastaamaan paremmin organisaatioiden tarpeita on hyvin tärkeää (Keung ym.2004). Koska tietojärjestelmien kehittäminen on usein useiden erilaisten sidosryhmien yhteistyönä tapahtuva iteratiivinen prosessi, asettaa tämä kovat vaatimukset artefakteille, joiden on otettava huomioon eri projekteissa hyvinkin erilaiset ympäristöt. Olennaisinta kaikissa hypoteeseissa on sisällyttää historiatiedon tallentaminen osaksi prosessia, jolloin voidaan työkalun käytön myötä kasvattaa myös estimaattien kannalta ensiarvoisen tärkeää historiatiedon määrää.

DPEMT-työkalu ja menetelmä soveltuu em. kaltaisiin projekteihin, joissa kaikilla osapuolilla ei ole täydellisiä tietoja projektin vaatimuksista, jolloin työkalulla voidaan luoda jatkuvasti itseään tarkentava estimointi malli.

## 5 JOHTOPÄÄTÖKSET JA YHTEENVETO

Tietojärjestelmän suunnitteluteorian soveltaminen organisaation tarpeisiin projektinestimointi ja mittaustyökalun suunnittelussa on erittäin tärkeää toimivan työkalun aikaansaamiseksi. Ohjelmistoprojektin estimointi on myös monimutkainen prosessi ja muuttujia on paljon, joten täysin pitävän arvion saaminen on tällä tietämyksellä harvinaista. Kuitenkin suunnitteluteorian pohjalta luodulla estimointityökalulla voidaan saavuttaa huomattavia parannuksia estimaattien tarkkuudessa, ja näin aikaansaada huomattavia parannuksia mm. projektien läpimenoajoissa. Tämä täsmällisyys luonnollisesti heijastuu asiakkaiden luottamuksena organisaatioon tai yritykseen luotettava toimittajana ja näin takaa myös tulevaisuuden liiketoiminnan jatkuvuuden.

Walls ym. (2004) esittävät, että suunnitteluteorian käyttö voidaan organisaatioissa jakaa neljälle tasolle, riippuen siitä kuinka pitkälle käyttö on viety. Ensimmäisellä tasolla suunnitteluteorian käyttö on hyvin pinnallista ja siihen viitataan vain sopivuuden kannalta. Toisella tasolla puolestaan soveltaminen on viety pidemmälle, ja suunnitteluteoriaa käytetään kehyksenä ja yhteisenä kielenä mm. meta-vaatimusten määrittelyssä uudelle luokalle. Kolmannella tasolla suunnitteluteorian soveltaminen on viety jo suhteellisen pitkälle, jolloin teorian avulla pystytään jo luomaan uusia näkemyksiä kehitettävien luokkien sisällä, jolla pystytään löytämään aivan uusia näkökulmia mitä ei muilla menetelmillä olisi muuten löydetty. Ylimmällä neljännellä tasolla suunnitteluteoria kehittää itseään jatkuvasti saamalla palautetta työskentelyn alla olevista luokista ja järjestelmistä, näin myös itse teoria on jatkuvan kehityksen kohteena. Tähän sisältyykin myös tämän tutkimukseni ongelma, suurin osa organisaatioista on suunnitteluteorian hyödyntämisessä tasoilla yksi tai kaksi, harvat tasolla kolme. Hevner ym. (2004) mukaan tasolle neljä ei todistetusti ole vielä päässyt yksikään organisaatio. Kappaleissa 5.1 ja 5.2 esitetään menetelmiä näiden ongelmien löytämiseksi ja

suunnitteluteorian ongelmakohtien verifioimiseksi käytännön projektinhallinta ja estimointityökalun toteutuksessa.

Kuten edellä on todettu on organisaatioiden pääseminen ylemmille tasoille suunnitteluteorian hyödyntämisessä hyvin harvinaista, jopa mahdotonta, joten (Walls;George R;& Omar A El, 2004) esittävät neljä strategiaa joiden avulla voidaan mahdollisesti saavuttaa parempia tuloksia.

### **5.1 Organisaation tarpeet ja niiden huomioonottaminen**

Ohjelmistoprojektin laajuuden hallinta vaatii lähes poikkeuksetta työkaluja. Mutta kuten edellä on todettu, eri työkalut soveltuvat hyvin vaihtelevasti organisaatioiden tarpeisiin ja tarve yksilöityyn ratkaisuun tai työkalujen yhdistelyyn on monessa organisaatiossa ilmeinen. Kuitenkaan ei kaikkien organisaatioiden tapauksissa voida edes olettaa että yksilöity työkalu olisi tarpeellinen tai taloudellisesti kannattava, vaan nk. hyllytuotteena löytyviä järjestelmiä voidaan myös käyttää hyväksi.

### **5.2 Yleisiä ongelmia ja näiden välttäminen**

(Walls;George R;& Omar A El, 2004) esittämät strategiat suunnitteluteorian paremman hyödyntämisen edesauttamiseksi ovat seuraavat:

1. Yleinen suunnitteluteorian parantaminen käytännön esimerkkien kautta. Suunnitteluteorian erilaisten komponenttien yhdistäminen ei välttämättä ole lukijalle tai kuulijalle helppoa. Tätä tulisi myös harrastaa projektin laajuudenhallinta työkaluille kehitettävässä suunnitteluteoriassa, jolloin organisaation johto ja muut sidosryhmät ymmärtäisivät siitä saatavan hyödyn, kehittäjien ja tutkijoiden lisäksi.
2. Työkalu strategiat, jotka käsittävät digitaalisessa muodossa olevan dokumentaation, esimerkkejä ja pohjia suunnitteluteorian hyödyntämisen tueksi.

3. Suunnitteluteorian rakenteen tarkentaminen jatkuvasti tutkimalla suunnitteluteoriaa ja jalostamalla sitä eteenpäin muuttuvien vaatimusten ym. mukana. (Walls;George R;& Omar A El, 2004) esittämä tekijä tämän kohdalla on, että suunnitteluteoria ei juurikaan auta ohjaamaan kernel-teorioiden tunnistamisessa, joten jatkuva teorian tarkastaminen ja jalostaminen on tarpeellista.

4. kohtana (Walls;George R;& Omar A El, 2004) mainitsee tarpeen levittää tietoa (nk. ISDT evankelismi) suunnitteluteorian käytöstä ja soveltamisesta niin akateemisessa kuin yritysmaailmassakin. Ongelmakohdiksi tässä tutkimuksessa tietojärjestelmien suunnitteluteorian implementoinnissa käytäntöön on identifioitu seuraavat asiat:

1. Tuloksien välitön näkyvyys
2. Referenssien puute ja/tai vähäisyys
3. Pitkäjänteisyyden puute käyttöönotossa
4. Tietämättömyys suunnitteluteorian hyödyntämisestä

Tuloksien näkyvyyden lisäämiseksi vaaditaan organisaatioissa pitkäjänteisyyttä ja työkaluilta puolestaan nopeampaa ja välittömämpää näkymistä organisaation prosesseissa. Myös referenssien puute (kohta 2) saattaa haitata varsinkin suunnitteluteoriaan pohjautuvan estimointityökalun käyttöönoton aloittamista, koska esim. johdolle ei ole esittää menetelmien konkreettista toimivuutta. Tämän korjaamiseksi on akateemisessa tutkimuksessa pyrittävä tuomaan esille onnistuneita referenssejä joiden avulla voidaan vakuuttaa eri sidosryhmät menetelmien toimivuudesta. Luonnollisesti näiden referenssien on oltava luotettavia. Pitkäjänteisyyden puutteeseen käyttöönotossa voidaan vaikuttaa luomalla työkalu, jolla tulokset näkyvät nopeammin sekä organisaation sisällä antamalla työkalulle aikaa vaikuttaa prosesseihin. Koska tietojärjestelmien suunnitteluteoriaan pohjautuvan projektin estimointi ja mittaustyökalun tutkimus ja kehitys on ollut hyvin vähäistä on myös tietämys organisaatiossa tästä hyvinkin heikkoa, jolloin ei

osata edes vaatia em. asioita. Tätä voidaan parantaa lisäämällä tietoisuutta suunnitteluteorian eduista estimointityökalun rakentamisessa (Walls; George R; & Omar A El, 2004)).

Jotta edellä mainittuihin suunnitteluteorian soveltamisen ongelmakohtiin voitaisiin perehtyä tarkemmin, on tämän tutkielman pohjalta esitetty kyselytutkimuksen runko, jossa käsitellään käytännönläheisemmin soveltamiseen liittyviä ongelmia lähinnä Matikainen (2006) esittämien tuotteensuunnittelu hypoteesien avulla.

Koska esimerkiksi suomalaisen FiSMAn kehittämän menetelmän ja järjestelmän avulla on saavutettu huomattavia parannuksia projektinhallinnassa pienessä osassa organisaatioita, olisi erityisen tärkeää saada tietoa siitä, mitkä ovat kriittiset kohdat suunnitteluteorian soveltamisessa ohjelmistoprojektinhallintatyökalun suunnittelussa ja ennen kaikkea käyttöönotossa, joka tämän tutkielman pohjalta on ilmennyt suurimmaksi esteeksi hyödyntämiselle. Toisaalta ohjelmistoprojektin estimointi ja mittaus on erittäin laaja käsite, jonka takia pelkällä mittaustyökalulla ei saada tarkimpia mahdollisia estimaatteja, vaan on myös osattava hyödyntää asiantuntija tietoa.

Tutkielman johtopäätöksenä voidaan todeta, että suunnitteluteorian avulla voidaan siis tuottaa erittäin toimiva projektin estimointi- ja mittaustyökalu ohjelmistoprojektien hallintaan esimerkiksi FiSMA ry:n tarjoamien menetelmien pohjalta kehitys on tulevaisuudessa mahdollista laajemmassa mittakaavassa. Ennen kaikkea näen tärkeänä työkalun suunnittelussa organisaation tarpeiden kartoituksen ennen työkalun rakentamista, jolloin pystytään helpommin soveltamaan tietoa itse suunnitteluteorian malleihin.

## LÄHDELUETTELO

- FiSMA. (2003). *Experience ND21 Situation Analysis Model*. FiSMA. FiSMA.
- FiSMA. (2006). *FiSMA 1.1 PAS Submission*. FiSMA. FiSMA.
- FiSMA. (2002). *FiSMA Reuse Measurement Method*. FiSMA ry. FiSMA.
- FiSMA. (2008, 3 16). <http://www.fisma.fi>. Retrieved 3 16, 2008, from Fisma ry website: <http://www.fisma.fi>
- FiSMA. (2007). NorthernScope - customer driven scope control for ICT projects. FiSMA.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in Information systems research. *MIS Quarterly*, 28 (1), 75-105.
- Käkölä, T., Wu, C., Yalaho, A., & Nahar, N. (2006). A Framework for ICT-supported international outsourcing of software production process and management. Department of Computer Science and Information systems.
- Keung, J., Jeffery, R., & Kitchenham, B. (2004). The Challenge of Introducing a New Software Cost Estimation Technology into a Small Software Organisation. *Proceedings of the 2004 Australian Software Engineering Conference*.
- Kitchenham, A. B., Pickard, M. L., MacDonell, G. S., & Shepperd, J. M. (2001). What accuracy statistics really measure. *IEEE Proc.-Softw.*, 148 (3).
- Lokan, C., Wright, T., Hill, P. R., & Stringer, M. (2001, September). Organizational Benchmarking Using the ISBSG Data Repository. *IEEE SOFTWARE*, 26-32.
- Matikainen, M. (2006). Information system supported project estimation and measurement. Jyväskylä: Jyväskylän yliopisto.
- Salo, A., & Käkölä, T. (2005). Groupware support for requirements management in new product development. *Journal of organizational computing and electronic commerce*, 15, 253-284.

Tran-Cao, D., Levesque, G., & Abran, A. (2002). Measuring Software Functional Size: Towards an Effective Measurement of Complexity. *Proceedings of the International Conference on Software Maintenance (ICSM'02)*. IEEE.

Walls, J. G., George R, W., & Omar A El, S. (2004). Assessing information system desing theory in perspective: how useful was our 1992 initial rendition? *Journal of information technology theory and application* , 43-58.