

KARELIA-AMMATTIKORKEAKOULU
Teknologiaosaamisen johtamisen koulutusohjelma
Ylempi ammattikorkeakoulututkinto

Vesa Turunen

KETTERIEN JA LEAN-MENETELMIEN KÄYTTÖÖNOTTO IT-
PALVELUALAN YHTIÖSSÄ

Opinnäytetyö
Toukokuu 2019



OPINNÄYTETYÖ
Toukokuu 2019
Teknologiaosaamisen johtamisen
koulutusohjelma

Tikkarinne 9
80200 JOENSUU
+358 13 260 600 (vaihde)

Tekijä
Vesa Turunen

Nimeke
Ketterien ja Lean-menetelmien käyttöönotto IT-palvelualan yhtiössä

Toimeksiantaja
Pohjois-Karjalan tietotekniikkakeskus Oy

Tiivistelmä

Opinnäytetyön tavoitteena oli IT-palvelutuotannon osana toimivan tietojärjestelmäsuunnittelun toiminnan tutkiminen ja analysointi sekä toiminnan kehittäminen Leanin ja ketterän kehityksen periaatteisiin perustuen. Lähtökohta opinnäytetyölle oli laadukkaan ja tehokkaan toiminnan edelleen kehittäminen palvelemaan yhtiön asiakkaita.

Opinnäytetyössä käsitellään teoriapohjina Lean-periaatteita yleisestä ja informaatiotekniikan alan näkökulmasta sekä ohjelmistotuotannon ketterän kehityksen malleja. Mittaukset on tehty Lean-pohjaisen IT-palvelutuotannon näkökulmasta relevanteilla mittareilla ja läpikävelyllä. Analyseissä mittaustuloksia käsitellään Leanin teoriapohjan mukaan.

Lopputuloksena tuotettiin teorian, mittausten sekä analyysien perusteella kehitysehdotukset, joiden avulla on mahdollista kehittää toimintaa järjestelmällisemmäksi Lean-periaatteiden ja työkalujen sekä ketterän kehityksen mallin mukaisesti sovellettuna yhtiön tarpeisiin. Opinnäytetyössä esiteltyjä mittareita voidaan myös edelleen hyödyntää kehitystoimenpiteiden jälkeen toiminnan mittaamiseksi.

Kieli

Sivuja 65

Suomi

Asiasanat

informaatiotekniikka, palvelutuotanto, Lean, ketterä kehitys



THESIS
May 2019
Master's Thesis
Degree Programme in Technology
Competence Management

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600 (switchboard)

Author
Vesa Turunen

Title
Deployment of Agile and Lean Methods for IT Service Company

Commissioned by
Pohjois-Karjalan tietotekniikkakeskus Oy

Abstract

The target of thesis was to find development possibilities for information technology service production company's system design department based on Lean and agile methods. The baseline of thesis was to develop operative work further to higher quality and more efficient to serve the company's customers.

Base theories for the thesis were the Lean principles from general and information technology point of view and software development agile processes. The measurements and walkthrough were implemented to fit Lean based information technology service production. Measurements were analyzed from Lean viewpoint.

The result of thesis was a development proposal list which was created based on theory, measurements and analysis. By means of the proposal list it is possible to develop working to be more systematical with the help of Lean and agile principles adjusted to the company's operations. The measurements are also usable after the development work to measure the impact of changes in working activities.

Language

Finnish

Pages 65

Keywords

information technology, service production, Lean, agile

Sisältö

Lyhenteet

1	Johdanto.....	7
2	Pohjois-Karjalan tietotekniikkakeskus Oy.....	8
3	Leanin historia	9
4	Lean-ajattelu	10
4.1	Virtaustehokkuus ja resurssitehokkuus	10
4.2	Kapasiteetin mitoitus	11
4.3	Jatkuva kehitys	12
5	Lean-periaatteet ketterässä kehityksessä.....	13
5.1	Hukan poistaminen.....	13
5.2	Jatkuva oppiminen.....	14
5.3	Loppuvaiheen päätökset.....	14
5.4	Nopea toimitus.....	15
5.5	Työntekijöiden osallistaminen.....	16
5.6	Laadukkaan tuotteen rakentaminen.....	17
5.7	Kokonaisuuden näkeminen	18
6	IT-palvelutuotannon kehityskohteet	19
6.1	Hukan havaitseminen.....	19
6.2	Laadukas tuotanto	20
6.3	Kustannukset	21
6.4	Työnohjauksen prosessi	23
6.5	Tekoäly ja koneoppiminen työnohjauksessa.....	25
7	Ketterä kehittäminen	26
7.1	Ohjelmistotuotannon menetelmät palvelutuotannossa	26
7.2	Ketterän kehityksen menetelmät.....	26
7.3	Agile Manifesto	27
7.4	Scrum	28
7.5	Kanban	29
8	Nykytilan kartoitus	31
8.1	Suunnitelma toteutusprosessien selvittämiseksi	31
8.2	Läpikävely prosessissa - Gemba	32
9	IT-palvelutuotannon mittaaminen.....	33
9.1	Saapuneet ja toteutetut palvelupyynnöt	33
9.2	Keskeytykset.....	34
9.3	Gemba-läpikävelyn tulokset.....	35
10	IT-palvelutuotannon mittauksen analyysi	37
10.1	Keskeneräiset työt ja keskeytykset.....	37
10.2	Keskeneräiset työt pisteytettynä ja keskeytykset.....	39
10.3	Läpimenoajat ja Littlen laki	40
10.4	Sijointi tehokkuusmatriisissa.....	43
10.5	Tehokkuusparadoksi käyttöasteesta ja läpimenoajoista	44
10.6	Läpikävelyn analyysi	46
10.7	Mittausten kehitysajatuksia.....	48
11	IT-alan työskentely ja kehitysajatuksia	49
11.1	Kehitysajatuksia.....	49
11.2	Multitasking ja mindfulness.....	49
11.3	Ishikawa- eli kalanruotokaaviot	51
11.4	Viisi kertaa miksi.....	53

11.5	Obeya / War Room eli tilannehuone	55
11.6	Plan-Do-Study-Act (PDSA) eli suunnittele, toteuta, tutki ja päätä	56
11.7	Sprintit eli pyrähdykset	57
11.8	Daily Scrum eli Scrum-palaverit	58
11.9	Incident Management eli havaintojen hallinta	59
11.10	Toiminnan kehityslista nelikenttäanalyysinä	60
12	Pohdinta	61
13	Epilogi.....	63
	Lähteet	64

Lyhenteet

AI	Tekoäly, Artificial Intelligence.
FIFO	Käsittelyjärjestys, ensimmäisenä jonoon tullut käsitellään ensimmäisenä, First In First Out.
ICT	Tieto- ja viestintäteknikka, Information and Communications Technology.
IT	Informaatio- tai tietotekniikka, Information Technology.
ITIL	Hyvien käytäntöjen kokoelma informaatiotekniikan palveluiden hallintaan, Information Technology Infrastructure Library.
PDSA	Suunnittele, toteuta, tutki ja päätä, Leanin kehitysprosessimetodi, Plan-Do-Study-Act.
PTTK	Pohjois-Karjalan tietotekniikkakeskus Oy.

1 Johdanto

Nykyaikainen informaatiotekniikan palvelutuotanto tarvitsee ratkaisuja toteutukseen asiakkaiden tarpeet mahdollisimman laadukkaasti ja kustannustehokkaasti. Koska jokainen yhtiö ja jokainen asiakas ovat omanlaisiaan, niin myös käytettävät metodit on pyrittävä valitsemaan siten, että ne toimivat hyvin käytännön työssä. On myös tiedostettava, että toimintatapoja on tarkistettava ja kehitettävä säännöllisesti, koska yhtiön toiminta ja toimintaympäristö muuttuvat ajan kuluessa.

Toiminnan hallitsemiseksi on olemassa monenlaisia keinoja. Opinnäytetyössäni tutkin informaatiotekniikan palvelutuotannon menetelmiä ja pyrin löytämään menetelmistä ja tekniikoista sopivimmat kehittääkseni Pohjois-Karjalan tietotekniikkakeskus Oy:n tietojärjestelmien suunnittelutyön tekemistä. Tavoitteena on luoda teoriaosuuden, kartoitustyön, mittauksen ja analyysien perusteella kehitystyölistä, jonka avulla pystytään kehittämään toimintaa PTTK Oy:n tietojärjestelmien suunnittelutyön palvelutuotannossa. Kehityskohde on rajattu tietojärjestelmien suunnittelutyöhön PTTK Oy:n monialaisesta toiminnasta. Yhtiön toiminnan monimuotoisuudesta johtuen on opinnäytetyön laajuudessa järkevämpää keskittyä kapeaan sektoriin, kuin yrittää tehdä yleispätevää kehitystä koko yhtiön laajuisesti.

Opinnäytetyön rakenne käsittelee johdannon jälkeen toimeksiantajan historian lyhyesti. Tästä siirrytään varsinaiseen teoriaosuuteen, jossa käsitellään Leanin historiaa sekä Lean-ideologian perusteita yleisellä tasolla. Teoriaosuuden seuraavissa luvuissa käsitellään Lean-periaatteiden soveltamista ohjelmistotuotantoon ja siihen rinnastettavissa oleviin tehtäviin, jonka jälkeen käydään läpi Lean-periaatteita IT-palvelutuotantoon sovellettuna. Teoriaosuuden päättää ketterän kehittämisen ajatusmallin läpikäynti. Teoriaosuudesta siirrytään kehityskohteen nykytilan kartoitukseen, mittauksen suunnitteluun ja keräämiseen sekä mittauksen analyysiin. Analyysin jälkeen ovat vuorossa mittauksen tulokset yhdistettynä teoria-ajatuksiin, joista lopputuloksena on kerätty kehitysajatuksia sekä työkalu-

ja, joilla toimintaa pystyttäisiin kehittämään jatkossa. Opinnäytetyön päättää pohdintaosuus sekä loppusanat.

Kirjoittaja on työskennellyt informaatiotekniikan alalla noin 15 vuotta erilaisissa ohjelmistotuotannon kehitystehtävissä tuotekehityksen, räätälöityjen ohjelmistojen ja palvelutuotannon parissa yksityisen sektorin yhtiöissä sekä julkisella sektorilla. Käytän pitkää kokemuspohjaa analysoidessani teoriaa vasten käytännön tekemistä.

2 Pohjois-Karjalan tietotekniikkakeskus Oy

Pohjois-Karjalan tietotekniikkakeskus Oy on voittoa tavoittelematon ICT-palvelujen tuotantoyhtiö, jonka tavoitteena on tuottaa asiakkailleen kokonaisvaltaisesti ICT-palveluita. Tietotekniikkakeskuksen omistajia ja samalla myös asiakkaita ovat Pohjois-Karjalan kunnat ja kuntayhtymät. Tietotekniikkakeskus on perustettu 1980-luvulla, mutta varsinainen toiminta on alkanut vuonna 2009. (Pohjois-Karjalan tietotekniikkakeskus 2017, 2.)

Pohjois-Karjalan tietotekniikkakeskuksen toimialueita ovat projektipalvelut, hankintapalvelut, konesali- ja tietoliikennepalvelut, tukipalvelut sekä tietojärjestelmäpalvelut. Tietotekniikkakeskuksen toiminta perustuu asiakastuntemukseen, työntekijöiden asiantuntemukseen sekä sitoutumiseen tehdä asiat hyvin ja huolella. (Pohjois-Karjalan tietotekniikkakeskus 2017, 3.)

Maakunnallisena toimijana tietotekniikkakeskus on pystynyt tuottamaan laaja-alaista palvelua asiantuntemuksella koko maakunnalle ja kaikki asiakkaat ovat saaneet tästä hyötyä verrattuna tilanteeseen, että jokainen kunta ja kuntayhtymä olisivat järjestäneet palvelunsa itse. Tietojärjestelmien yhtenäisyyttä ja tietojen liikkumista maakunnassa on myös pystytty edistämään yhteisen tietotekniikkakeskuksen ansiosta edistyksellisellä tavalla, omalla toiminnallaan tietotekniikkakeskus on tehnyt Suomessa pioneerityötä edellä mainittujen asioiden osalta.

3 Leanin historia

Leanin perustana pidetään Toyota Motor Corporationissa kehitettyä Toyota Production System -johtamisjärjestelmää, jonka kehitys alkoi 1940-luvulla Japanissa. Toyota Production Systemin tarkoituksena on järjestelmällisesti tehostaa toimintaa, karsia turhat asiat pois, tehdä asioita oikeaan aikaan ja tuottaa mahdollisimman hyvälaatuisia tuotteita. Toyota Production Systemistä kehittyi termi Lean 1980-luvulla ja siitä lähtien Lean-järjestelmää on kehitetty toimimaan monissa erilaisissa tuotantoympäristöissä samoja pääperiaatteita noudattaen. (Bicheno & Holweg 2009, 279–281.)

Leania on pyritty laajentamaan myös teollisuusalojen ulkopuolelle. Informaatiotekniikan ala ja erityisesti ohjelmistotuotanto on tyypillisesti halunnut tutkia ja kehittää erilaisia työprosesseja, tästä johtuen Leanista on otettu ajatusmalleja myös ohjelmistotuotannon puolelle. Prosessit ovat muokkautuneet ohjelmistotalalle sopiviksi Leanin pääperiaatteita noudattaen. Ketterä kehitys on eräänlainen ilmentymä Leanin periaatteista ohjelmistotuotannossa. Asiakasohjautuvuus, laatu sekä asioiden tekeminen oikeaan aikaan ovat tärkeässä roolissa myös ohjelmistotuotannossa. (Poppendieck 2006 23–24, 34.)

Leanin monikymmenvuotinen historia elää edelleen ja Lean-ajattelumalli kehittyy yhä vaikka peruseriaatteet pysyvätkin samoina. Teknologiat ja tekniikat muuttuvat, poistuvat ja lisääntyvät ja Lean pystyy mukautumaan uusiin haasteisiin. IT-ala on tästä hyvä esimerkki, miten teollisuuden tarpeisiin tuotettu ajatusmalli on sovitettu hyvin toisenlaiselle alalle. Kun toiminnasta tehdään järjestelmällistä, niin Lean on ajatusten tukena erinomainen apuväline.

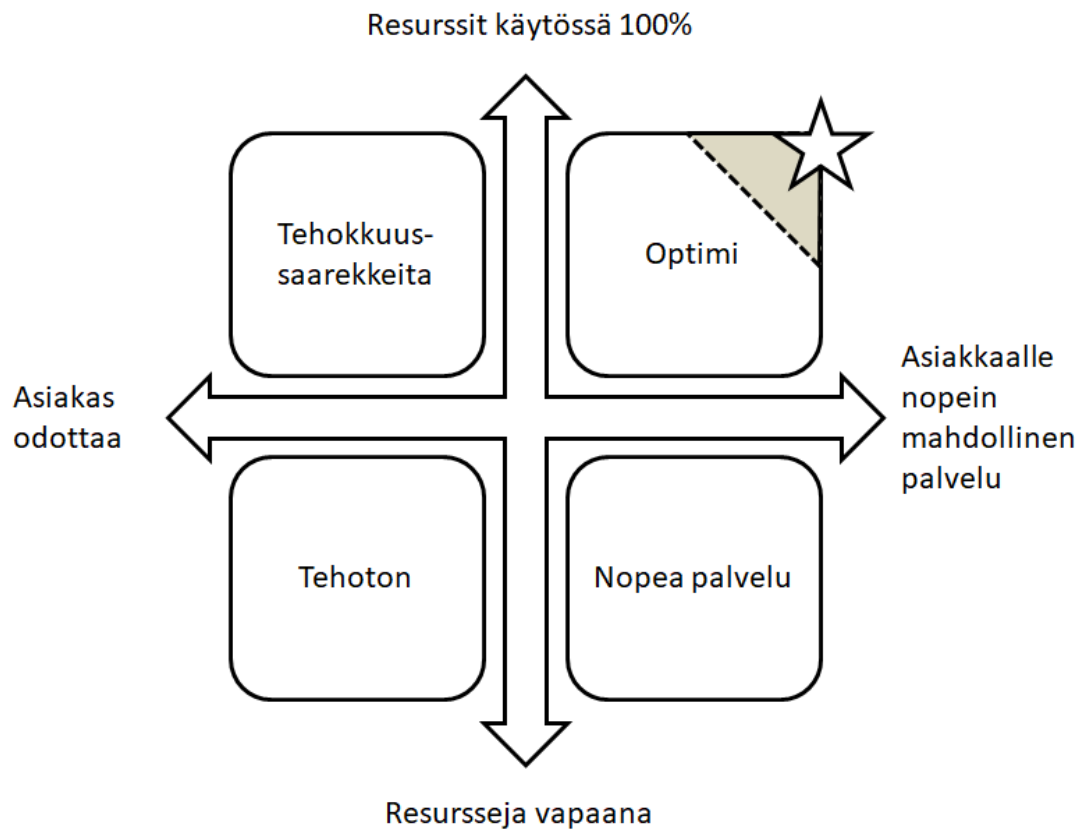
4 Lean-ajattelu

4.1 Virtaustehokkuus ja resurssitehokkuus

Lean jakaa työn tekemisen kahteen kategoriaan, on olemassa virtausta ja on olemassa resursseja, molempia tarvitaan, että työ saadaan tehtyä. Leanin periaatteisiin kuuluu, että organisaation virtaustehokkuus saataisiin hiottua mahdollisimman hyväksi eli pyritään saamaan asiakkaan kokema läpimenoaika mahdollisimman pieneksi ja pyritään saamaan tehtäviä valmiiksi mahdollisimman paljon. Läpimenoaika asiantuntijatyössä kuten IT-palvelutuotannossa tarkoittaa ajan kulumista siitä hetkestä kun asiakas tekee tilauksen siihen hetkeen kun asiakas saa toimituksen tilauksestaan. (Torkkola 2015, 57.)

Resurssitehokkuus eli resurssien käyttö eroaa virtaustehokkuudesta siten, että jos organisaatio on ainoastaan resurssitehokas, niin töitä on koko ajan jonossa ja resurssien käyttöaste on suuri. Resurssitehokkuuden seurauksena on usein kuormituksesta uupuneita työntekijöitä ja asiakkaat kokevat palvelun hitaana, koska töitä on paljon jonossa ja asiakkaan tilaukseen ei ehditä vastaamaan asiakkaan kannalta tarpeeksi nopeasti. (Torkkola 2015, 57–58.)

Kuvassa 1. on esitetty oikeassa yläalaidassa tähdellä pyrkimys mihin organisaation pitäisi pyrkiä, resurssitehokkuuden ja virtaustehokkuuden optimipisteeseen, jossa resursseja käytetään mahdollisimman paljon ja työt virtaavat mahdollisimman hyvin. Käytännössä asiantuntijaorganisaatiossa tehtävien työmäärän keskinäinen vaihtelu aiheuttaa sen, että optimiin ei koskaan päästä. Vaihtelun osuutta mallintaa harmaa alue. (Torkkola 2015, 57.)



Kuva 1. Virtaustehokkuus ja resurssitehokkuus tehokkuusmatriisissa (Torkkola 2015, 57).

Tehokkuusmatriisissa sekä tarjonnan, että kysynnän vaihtelu vaikuttaa mahdollisuuksiin päästä optimaaliseen resurssi- sekä virtaustehokkuuteen. Asiantuntijaorganisaatiossa vaihtelua tulee työtehtävistä ja ihmisistä, joihin voi olla haastavaa vaikuttaa. Vaihtelun pienentämiseksi on kuitenkin mahdollista tehdä toimenpiteitä myös asiantuntijaorganisaatioissa, kun prosesseja tutkitaan tarpeeksi. (Modig & Ahlström 2013, 106–107.)

4.2 Kapasiteetin mitoitus

Tehokkuusmatriisissa tarjontaa voidaan hallita oikeanlaisella kapasiteetin mitoituksella. Mikäli työjonoa alkaa muodostua, niin kapasiteettia on liian vähän ja toisinpäin ajateltuna, mikäli jokin resurssi on koko ajan ilman työtä, niin kapasiteettia on liikaa. (Torkkola 2015, 165.)

Kapasiteetin käyttöä voidaan määritellä Littlen lailla, siinä lasketaan keskimääräistä läpimenoaikaa kerrottuna tuotantoa tekevien yksiköiden määrällä eli kapasiteetilla. Läpimenoaika kasvaa, mikäli keskeneräisiä töitä on paljon ja tuotantokapasiteetti ei kasva. Tavoitteena on pyrkiä kohtuulliseen keskeneräisten töiden määrään suhteessa tuotantokapasiteettiin, kuitenkin siten, että keskeneräisten töiden määrä ei putoa nolnaan jolloin tuotanto pysähtyisi. (Torkkola 2015, 186–191.)

Oikeankokoisen kapasiteetin määrittämiseksi on olemassa Kingmanin yhtälö. Kingmanin yhtälön peruseriaatteet ovat, että läpimenoaika kasvaa, mikäli käsittelyaika kasvaa, vaihtelu kasvaa tai resurssien käyttöaste kasvaa. Mikäli näitä tekijöitä löytyy prosessien toiminnasta, niin organisaatio siirtyy kohti ylikuormitustilaa. (Torkkola 2015, 191–192.)

4.3 Jatkuva kehitys

Leanin tarkoituksena ei ole tehdä kerralla muutosta, jonka jälkeen pysytään stabiilissa tilassa vaan tavoitteena on aloittaa prosessien jatkuva kehittäminen ja parantaminen. Organisaation muutoksia voi tutkia kahden ajankohdan välillä, mikäli muutoksia on tapahtunut, organisaatio on dynaamisessa muutosten tilassa. Jatkuvaa kehittämistä tekevä organisaatio oppii kokemuksistaan ja saa sitä kautta uutta osaamista ja ymmärrystä tekemiseensä. (Modig & Ahlström 2013, 152.)

5 Lean-periaatteet ketterässä kehityksessä

Mary ja Tom Poppendieck omaavat pitkän kokemuksen ohjelmistotuotannosta ja ovat keränneet teokseensa 'Lean Software Development – An Agile Toolkit' hyviä käytäntöjä Lean-prosessien yhdistämiseksi ohjelmistotuotantoon (Poppendieck 2007, 15–16). Ohjelmistotuotanto käsitteenä on laaja kokonaisuus ja pitää sisällään monenlaista toimintaa. Osa PTTK Oy:n tietojärjestelmien suunnittelun parissa työskentelevistä henkilöistä tekevät käytännössä ohjelmistotuotannon töitä ja ovat rinnastettavissa ohjelmistosuunnittelijoihin ja -kehittäjiin ohjelmistoalaan keskittyneissä yhtiöissä.

Poppendieckit ovat tuottaneet teokseensa (2007) listan Leanin periaatteista ja miten ne voidaan muuttaa ketterän kehityksen toimintatavoiksi. Lista käydään läpi alaluvuissa 5.1 – 5.7.

5.1 Hukan poistaminen

Hukka on kaikkea, mikä ei tuota lisäarvoa tuotteelle tai palvelulle asiakkaan näkökulmasta. Palvelun tuottajan näkökulmasta katsottuna hukka on este, joka hidastaa toimitusta asiakkaalle. Esimerkkejä hukasta ovat tuotannon siirtäminen toiseen tehtävään, joka aiheuttaa keskeytyksen työhön tai keskitytään tekemään asiaa, esimerkiksi ohjelmiston osaa, jota asiakas ei ole tilannut ja määritellyt. (Poppendieck 2007, 25.)

Hukkaa tapahtuu väistämättä normaalissa IT-alan työelämässä aina jonkin verran. Järjestelmällisellä ja onnistuneella työnjohdolla on kuitenkin mahdollista vähentää hukkaa ja myös työntekijä voi ohjata omaa tekemistänsä siten, että hukkamäärä pienenee.

5.2 Jatkuva oppiminen

Tuotannon kehittäminen ei välttämättä ole yksinkertaista. Tarvitaan useita kehityskierroksia, että prosesseja saadaan hiottua tarvetta vastaaviksi. Jokaisesta kierroksesta olisi hyvä havaita, miten asiat muuttuivat ja oppia havainnoista prosessin muuttamista oikeaan suuntaan. Vaikka ensimmäisellä kehityskierroksella päästäisiin jo hyviin tuloksiin, niin prosessin jatkokehitystä kannattaa silti tehdä ajoittain. (Poppendieck 2007, 26.)

IT-alalla jatkuva oppiminen on luontaista jo alan takia, usein kuitenkin keskitytään teknologioihin varsinaisten tuotantoprosessien sijaan. Kehitykselle pitäisi ohjata riittävästi resursseja, ettei kehitys pysähtyisi esimiesten ja työntekijöiden omien töiden kuormituksen vuoksi. Myös kehitysprosessin avoimuus on hyvä muistaa. Ajatuksia prosessien parantamiseksi pitää saada myös suorittavilta työntekijöiltä, eikä pelkästään päätäntävastuussa olevilta työntekijöiltä.

5.3 Loppuvaiheen päätökset

Prosessit, jotka mahdollistavat mahdollisimman myöhäisen päätöksen teon, ovat joustavampia kuin aikaisessa vaiheessa päätöksen teon lukitsevat prosessit. Kun mahdollisuuksia prosessin ohjaukseen jätetään prosessin loppuosaan, niin päätöksen tekeminen tukeutuu enemmän faktatietoon kuin ennusteisiin. Mikäli prosessin lopputuotteessa on epävarmuutta prosessia aloittaessa, niin päätöksen teon venyttäminen epävarmuustekijöiden osalta todennäköisesti auttaa lopputuloksen onnistumista. (Poppendieck 2007, 26.)

Loppuvaiheen päätöksiin auttava lähestymistapa on myös, että keskitytään ensin korkealle priorisoituihin asioihin toimituksessa ja toteutetaan pienemmän prioriteetin asiat myöhemmin. Ajatuksena tässä on se, että saadaan toteutettua mahdollisesti hankala ja kallis ominaisuus prosessin alkupäässä, jonka jälkeen prosessin myöhemmässä vaiheessa ei törmätä mahdolliseen ongelmaan, joka voisi aiheuttaa kriittisen pysähdysten tuotannolle. (Poppendieck 2007, 76–77.)

IT-alalla epävarmuutta on usein enemmän tai vähemmän erityisesti kehitystyön alkuvaiheessa. Järjestelmien vaatimukset ja määrittelyt voivat olla hyvin yleisluontoisia ja yleisluontoisten määritysten perusteella lopputulos voi olla hyvinkin jotain muuta, kuin mitä asiakas halusi. Päätösten teon loppuvaiheeseen jättämisellä voidaan välttyä näiltä ongelmilta. Tämä vaatii kuitenkin hyvää yhteistyötä ja säännöllistä katselmointia tilaajan ja tuottajan välillä, että ohjausta voidaan tehdä kehitysprosessin aikana oikeaan suuntaan. Toimittajalta periaatteen noudattaminen vaatii ammattitaitoa siten, että oletuksia ei lukita ennen kuin asioista on varmuutta. Myöhäisen päättämisen periaatteen noudattaminen osaltaan vähentää myös hukkaa, kun tehdään oikeita asioita, eikä resursseja kulu väärin asioiden tekemiseen.

5.4 Nopea toimitus

Nopean toimituksen periaatteen hyötyinä on, että asiakas saa nopeasti tilaamansa tuotteen tai palvelun ja kehityskierrosten pysyessä lyhyinä kehitysorganisaatio saa jatkuvan oppimisen periaatteeseen syötettä kehityskierroksilta. Nopea toimitus myös toteuttaa Leanin arvovirtamallin mukaista ajatusta mahdollisimman lyhyestä läpimenoajasta. (Poppendieck 2007, 26.)

Nopea toimitus ei kuitenkaan tarkoita kiireellä tehtyä ja mahdollisesti huolimattomasti tuotettua toimitusta. Nopea toimitus tarkoittaa myös nopeaa sisäistä toimitusta: kun toimitusta tehdään nopeasti, niin testaukselle jää periaatteessa enemmän aikaa jos aikataulua ei ole tarvetta kiristää. (Poppendieck 2007, 97–99.)

IT-alalla kehitysprosessit voidaan yrittää saada mahdollisimman nopeiksi ja lyhyiksi pilkkomalla asiat riittävän pieniksi osiksi, tämä vastaa ketterän kehityksen pilkontamallia. Erityisesti kehitysprosesseissa tämä auttaa prosessin läpimenoaikojen lyhyenä pitämistä. Ylläpitotyössä prosessit on myös mahdollista saada nopeiksi. Molemmissa tapauksissa työnohjaaminen ja resursointi ovat tärkeässä roolissa, myös hukka on saatava pieneksi, että päästään tavoitteisiin.

5.5 Työntekijöiden osallistaminen

Parhaaseen suorituskykyyn pyrkivässä prosessissa tulee pyrkiä saamaan niin suuret linjat kuin kaikki yksityiskohdatkin hiottua mahdollisimman sujuviksi. Suorittavassa portaassa toimivilla työntekijöillä on todennäköisesti paras tietämys yksityiskohdista ja niiden vaikutuksista prosessiin. Lean-periaatteet perustuvat virtauksen imuun ja aktiiviseen ohjaukseen. Suorittavan työn tekijät ovat virtauksen keskiössä ja prosessin sujuvuuden vuoksi työntekijöiden täytyy toimia aktiivisesti sekä viestittää prosessin toisille osapuolille omasta toiminnastaan, jotta prosessi pysyisi sujuvana koko ajan. Vaikutusmahdollisuudet työntekijätasolla auttavat prosessin imun sujuvuutta. (Poppendieck 2007, 26–27.)

Työntekijöiden osallistaminen on tärkeää myös IT-alalla ja erityisesti ryhmätyöskentelyssä. Jokaisen ryhmän jäsenen tulisi tietää prosessin tilanne, että hän osaisi omassa työssään ennakoida tulevan työn, arvioida siihen kuluvan ajan sekä varata työlle sopivan ajankohdan. Järjestelmäsuunnittelussa hyvä esimerkki tästä on järjestelmän käyttöönotto, karkeasti mallintaen, ensin tilataan palvelin, asennetaan palvelinympäristö sekä järjestetään tietoliikenneyhteydet, asennetaan ohjelmisto, konfiguroidaan ohjelmisto määritysten mukaisesti, testataan toimivuus ja luovutetaan käyttöön. Nopeasti laskien tarvittiin hankinta-, infra-, tietoliikenne ja järjestelmäasiantuntijoita ennen kuin järjestelmä saadaan luovutettua käyttäjälle. Koko ketjun on toimittava hyvin yhteen, että toimitus saadaan tehtyä mahdollisimman tehokkaasti ja mahdollisimman pienellä hukalla.

5.6 Laadukkaan tuotteen rakentaminen

Rakentamalla laadukas ja hyvin toimiva tuote tai toimitus vastataan useimmiten asiakkaan tilaukseen. On tärkeää varmistaa, että prosessi tai järjestelmä toimii kaikilta osa-alueiltaan, toimituksessa on tarkoitus huomioida muitakin asioita kuin nykyhetki ja oma osuus tai järjestelmä. Huomion kohteena tulisi olla kehittyminen ja vastaaminen tulevaisuuden haasteisiin sekä integroituminen muihin prosesseihin tai järjestelmiin. Laadukkaaseen kokonaisuuteen pääseminen vaatii huolellista suunnittelua, pitkäjänteistä toteutusta ja paljon testausta sekä mitauksia. Prosessin epäkohtien havainnointia ja niiden korjaamista tarvitaan niin kehitysprosessin aikana kuin ylläpitovaiheessa. (Poppendieck 2007, 27.)

Laadukkaan kokonaisuuden rakentaminen on IT-alalla kohde, johon tulisi pyrkiä aina. Tilaajan ja toimittajan välisessä työskentelyssä ongelmaksi muodostuu usein aika tai raha, koska kehityskohteelle määritellään aina aikataulu sekä budjetti ja joskus alkuperäinen arvio aikataulusta tai budjetista ei toteudu. IT-toimituksissa laadun varmistaminen on usein kohde, josta tingitään, kun aikataulu tai budjetti alkavat olemaan täynnä. Toimittajan kannalta ratkaisuna olisi usein se, että rajataan kohdetta siten, että rajauksen sisällä olevat ominaisuudet saadaan toteutettua ja testattua hyvin, mutta loppujen lopuksi asiakas on usein se, joka kertoo mitä haluaa rahoillaan tehtävän. Tästä saattaa joskus syntyä ongelmallinen kierre tilaajan ja toimittajan välillä, kun tilattu tuote ei täytäkään laadullisesti vaatimuksia, ja muutoksia sekä korjauksia joudutaan tekemään jo tuotannon käynnistymisen jälkeen. Alkuperäisessä kehitysprosessissa tingitty aika tai budjetti saattaa moninkertaistua johtuen ongelmista, jotka olisi pystytty havaitsemaan ja korjaamaan laadukkaassa kehitysprosessissa.

5.7 Kokonaisuuden näkeminen

Kokonaisuuden näkeminen ja onnistuneen kokonaisuuden rakentaminen vaatii osaamista monilta osa-alueilta. Erityisesti prosessin ohjauksesta vastaavan tulisi tuntea kokonaisuus siten, että hän pystyy ohjaamaan ja aikatauluttamaan työt oikeille henkilöille, että prosessi pysyy sujuvana. Yksittäisen työntekijän näkökulmasta myös työntekijöillä on vastuuta tehdä työnsä oikeaan aikaan, että prosessissa seuraavat osa-alueet pääsevät aloittamaan toimintansa suunnitellusti. Töiden oikea-aikainen rytmitys ja yhteistyö korostuvat kokonaisuuden rakentamisessa. (Poppendieck 2007, 27.)

Järjestelmien parissa työskentelevän täytyy nykypäivänä ymmärtää, että hyvin harva järjestelmä toimii vain yksin. Järjestelmät muodostavat ympäristön ja laajemmassa mittakaavassa kokonaisarkkitehtuurin. Kokonaisuuden ohjaamisen kannalta on nähtävä kokonaisuus, mutta usein ohjausroolissa toimiva ei voi olla asiantuntija jokaisessa järjestelmässä. Ohjaajalla tulee olla hyvä käsitys siitä, miten järjestelmien tulisi toimia yhdessä ja antaa järjestelmien erityisasiantuntijoiden hoitaa oma osuutensa. Yhteistyötä ja aktiivisuutta prosessiin osallistuvien välillä tarvitaan kokonaisuuden onnistumiseksi.

6 IT-palvelutuotannon kehityskohteet

6.1 Hukan havaitseminen

Jotta hukkaa voitaisiin alkaa poistamaan, on ensin havaittava hukan kohteet, mihin työaika kuluu ja mikä ei tuota loppuasiakkaalle lisäarvoa. Poppendieckien teoksessa kuvataan taulukkomuodossa (taulukko 1) seitsemän ohjelmistotuotannon hukkalajia, jotka pätevät myös IT-palvelutuotannossa. (Poppendieck 2007, 33.)

Taulukko 1. Lean ohjelmistotuotannossa, seitsemän hukkalajia.

Lean - Ohjelmistotuotannon seitsemän hukkalajia

Osittain tehty työ

Ylimääräiset prosessit

Ylimääräiset ominaisuudet

Töiden välillä vaihtaminen

Odottaminen

Turhat liikkeet

Virheet

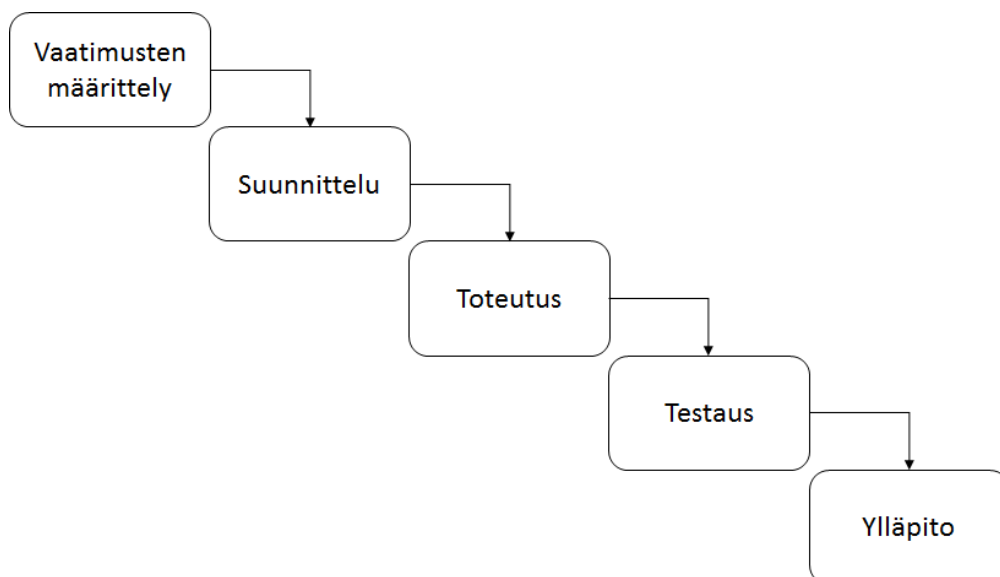
IT-palvelutuotannossa korostuvat hukan lajeista osittain tehty työ, töiden välillä vaihtaminen, turhat liikkeet ja virheet. Ylimääräiset prosessit ja ominaisuudet sekä odottaminen ovat usein pienemmässä roolissa. Osittain tehty työ ja töiden välillä vaihtaminen liittyvät vahvasti toisiinsa. Korkeamman prioriteetin omaava työ vaatii edellisen työn lopettamista hetkeksi tai pitemmäksi ajaksi ja hukka kasvaa tämän vuoksi. Turhat liikkeet eivät IT-palvelutuotannossa ole välttämättä ratkaisujen etsimiseen kuuluvia liikkeitä, vaan ne voivat olla myös laveita määri-tyksiä, jotka toteutuessaan eivät kelpaa asiakkaalle ja ovat tulkinnasta johtuvia virheitä. Varsinaiset virheet hukkaavat aikaa toisella tavoin, kun tehtyyn työhön palataan paikallistamaan ja korjaamaan virhettä.

6.2 Laadukas tuotanto

IT-palvelutuotannossa ja erityisesti järjestelmäsuunnittelun tehtävissä ratkaistavaksi tulee asioita, jotka ovat jollain tavalla uusia tai vastaavia tehtäviä ei ole aiemmin tehty. Tällöin hyvin toimiva prosessimalli korostuu, valmiina tulisi olla määrittelypohjat, jotka käydään yhdessä asiakkaan kanssa läpi ja vasta hyväksynnän jälkeen siirrytään toteutus- ja testausvaiheisiin. (Poppendieck 2007, 47.)

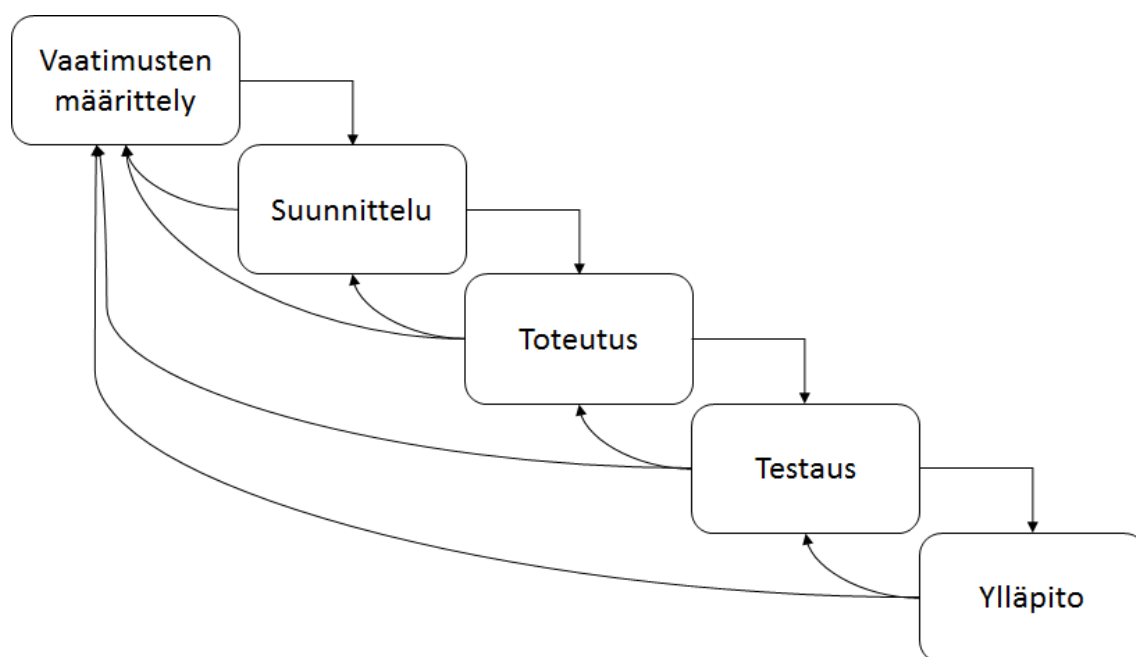
Kun järjestelmäsuunnittelun tekemään toteutukseen on suoritettu hyväksymistestaus ja tuotos siirretään tuotantoon, tulisi toimituksesta kerätä palautetta, jonka pohjalta voidaan toteuttaa Lean-periaatteena jatkuvaa oppimista. Palautteesta tulisi etsiä parannettavat asiat ja seuraavalla toteutuskierroksella pyrkiä toteuttamaan niitä. Jos prosessia ajatellaan virtauksena, niin virtausta saadaan sitä sujuvammaksi mitä vähemmän epäkohtia prosessista löytyy. (Poppendieck 2007, 50.)

Perinteistä järjestelmätuotetta tehtäessä ohjelmistotuotannon vesiputousmalli (Kuva 2) ja Lean voisivat toimia erittäin hyvin. Asiat tehtäisiin järjestelmällisesti ja huolellisesti alusta alkaen, prosessi etenisi järjestelmällisesti kohti maalia ja lopputuote olisi tasalaatuisesti tehty.



Kuva 2. Vesiputousmalli (Royce 1970, 2).

IT-palvelutuotannossa taas ei useimmiten rakenneta tuotetta ja on normaalia, että alemmista prosessivaiheista palataan usein ylemmän tason vaiheisiin (Kuva 3). Tämä johtuu usein monestakin seikasta. Usein toistuvia asioita ovat, että asiakkaalta saadaan liian yleisellä tasolla olevat määrittymiset, toteutus aloitetaan ilman riittävää suunnittelua tai validointia ei tehdä kunnolla. Aina pitäisi kuitenkin ottaa huomioon, että kun palataan vesiputouksessa taaksepäin, menetetään aikaa ja aikataulutusta pitäisi tehdä sen mukaan uusiksi mihin vaiheeseen asti palataan. Kun palataan vesiputousmallissa taaksepäin, kustannukset lisääntyvät ja toimitusaika pitenee.



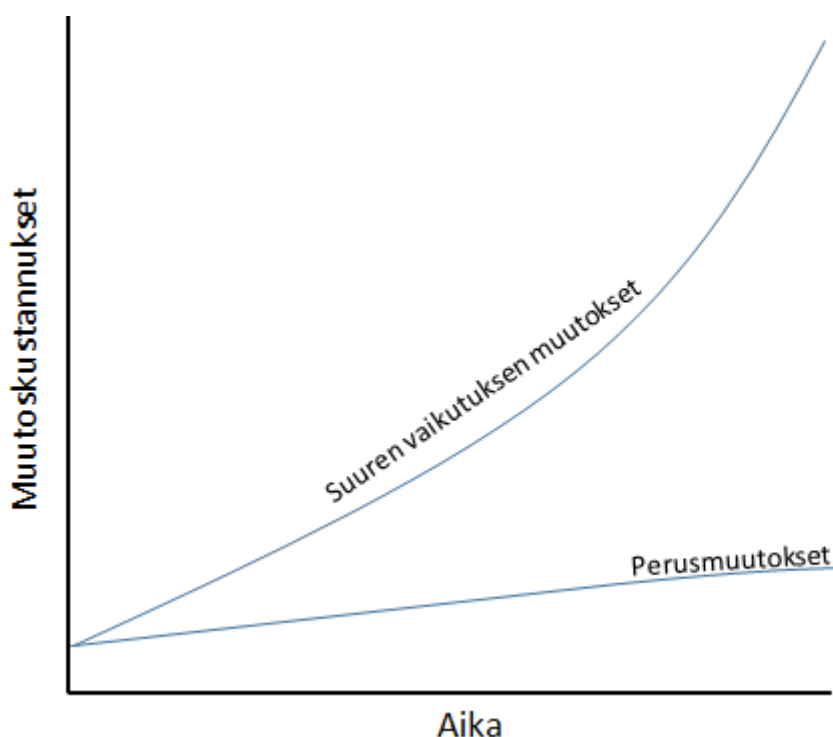
Kuva 3. Vesiputousmalli IT-palvelutuotannossa.

6.3 Kustannukset

IT-palvelutuotannosta syntyy aina kustannuksia, mutta palvelutuotantosopimuksesta riippuen joko asiakkaan tai toimittajan näkökulmasta on taloudellisesti järkevää pyrkiä kustannusten minimointiin. Yleishyödyllisesti ajatellen kustannusten minimoiminen on aina järkevää.

IT-palvelutuotannossa perustoiminnasta syntyy kustannuksia ja niiltä ei voi välttyä, mutta hyvällä suunnittelutyöllä voidaan pienentää muutoksista aiheutuvia

kustannuksia. Muutosten kustannuksiin vaikuttavat muutoksen vaikutuksen koko ja aika, missä vaiheessa muutos tehdään. Mitä myöhemmäksi muutoksen tekoa venytetään ja mitä suurempi sen vaikutus on, niin pääsääntöisesti sitä suurempi kustannuskerroin muutokselle syntyy (Kuva 4). (Poppendieck 2007, 77–79.)



Kuva 4. Muutosvaikutukset kustannuksiin (Poppendieck 2007, 79).

IT-palvelutuotannossa asiakkaan ja toimittajan järjestelmällisellä yhteistyöllä voidaan välttää suuren vaikutuksen muutosten aiheuttamia kustannuksia. Hyvänä esimerkkinä suurista kustannusvaikutuksista IT-palvelutuotannossa ovat vanhat laitteet ja järjestelmät. Järjestelmien ja laitteiden elinkaaria tulisi käydä läpi säännöllisesti, että välttyttäisiin vanhentuneen teknologian ongelmilta. Vanhojen järjestelmien ylläpito muodostuu jossain vaiheessa hankalaksi ja usein myös kalliiksi ja vanhat laitteet muodostavat rikkoutuessaan riskin varaosien tai kokonaisten laitteiden saamisessa tuotantoon, mikäli tuotteen valmistus on lopetettu jo kauan sitten.

Euroopan unionin sähköisen hallinnon toimintasuunnitelmassa vuosille 2016-2020 on määritelty vanhoille teknologioille poistosuositus, kun käyttönotosta

on kulunut 15 vuotta tai enemmän (European Commission, 3). Fyysisten laitteiden osalta suosituksen noudattaminen on melko yksinkertaista, mutta ohjelmistojen osalta tulkinta muuttuu hankalammaksi, etenkin jos järjestelmää on päivitetty vuosien saatossa. Ohjelmistoteknisesti ajatellen 15 vuoden takaisia ohjelmointikieliä ei sellaisenaan käytetä, vaan jos ne ovat eläviä ohjelmointikieliä, niin niihin on tullut paljon päivityksiä. Ohjelmistotoimittajilta olisikin ehkä pyydettävä selvitystä, kuinka paljon järjestelmässä on alkuperäistä ohjelmaa jäljellä, jos käyttönotosta on kulunut pitempi aika, kuten 10 - 15 vuotta.

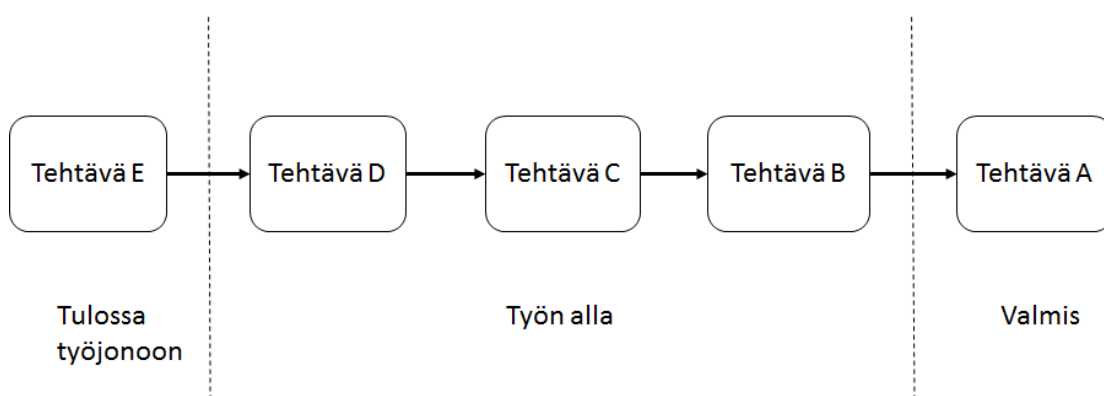
Järjestelmällisellä työllä ja prosesseilla on mahdollisuus vaikuttaa kustannusten syntymiseen. Esimerkkejä ja tapauskuvauksia löytyy moneen asiaan ja niitä kannattaisi hyödyntää oman toiminnan ennakoinnissa.

6.4 Työnohjauksen prosessi

Tasapuolisesti asiakkaitaan kohtelevan yhtiön työnohjaus nousee tärkeään rooliin, kun palvelutuotanto on jatkuva. Mikäli tekijät on kiinnitetty kokonaan tietylle asiakkaalle ja tiettyyn projektiin, työnohjaus voi tulla asiakkaan suunnasta, asiakas voi määritellä mitkä tehtävät ovat suurimmalla prioriteetilla ja pitää tehdä ensiksi. Asiakkaan tulisi kuitenkin huomioida toimittajan asiantuntijoiden asiantuntemusta, että työjono pysyisi järkevänä priorisoinniltaan. Työnohjaus voidaan tällaisessa tapauksissa tehdä mahdollisimman itseohjautuvaksi asiakkaan ja toimittajan asiantuntijoiden välillä. (Poppendieck 2007, 102–103.)

Toimittajan puolesta työnohjausta tarvitaan enemmän, kun toimittajan asiantuntijat työskentelevät usealle asiakkaalle ja palvelua pyritään tuottamaan tasapuolisesti kaikille. Tällöin toimittajan edun mukaista on selvittää asiakkaalle, että asiantuntijat työskentelevät usealle asiakkaalle ja asiakkaan tehtäviä ratkotaan tietystä aikaikkunassa. Työnohjauksen palaverieja kannattaa pitää toimittajan asiantuntijoiden kesken säännöllisesti, lyhyessäkin palaverissa saadaan selvyttä työntekoon ja mahdollisiin ongelmiin. (Poppendieck 2007, 103–104.)

Useaa asiakasta palvelevan IT-palvelutuotannon työohjauksen tulisi toimia First In First Out (FIFO) -periaatteen mukaisesti (Kuva 5). Ensimmäisenä työhön tullut asia pitäisi saada ensimmäisenä tehtyä. IT-palvelujen tarjoajan pitää kuitenkin huomioida tässä prioriteetit, korkeamman tason tehtävät pitää tehdä matalamman tason prioriteetin tehtäviä aiemmin. FIFO-periaatteen toiminnassa tulisi huomioida, että resurssia suhteessa tehtäviin on liian vähän jos työjono ei koskaan tyhjene. Riskinä on silloin, että matalan prioriteetin tehtävät eivät tule koskaan tehdyksi. (Bicheno & Holweg 2009, 143–144.)



Kuva 5. FIFO-prosessi.

IT-palvelutuotannossa, jossa asiantuntijat työskentelevät usealle asiakkaalle, olisi työohjauksen kannalta suositeltavaa käyttää IT Infrastructure Libraryn (ITIL) mukaista ohjausta, jossa voi olla useampia tasoja. Pääsääntönä on se, että mikäli asiakkaan ongelmaa ei pystytä ratkaisemaan ad hoc -ratkaisuna, niin ongelma siirtyy toimittajan sisällä prosessin mukaisesti seuraavalle tasolle. Helpdesk-tyyppisessä toiminnassa ensimmäinen porras voi olla puhelinpalvelu ja suunnittelua enemmän vaativissa tehtävissä asiakkaalle vastaava porras voi olla IT-palvelupäällikkö. Aloitusportaasta jatketaan toimittajan portaikossa syvemmille tasoille. (ITIL Roles 2011.)

Lean-periaatteisiin verraten IT-palvelutuotannossa osittain tehdystä työstä, töiden välillä vaihtamisesta sekä turhista liikkeistä muodostuvaa hukkaa saadaan vähennettyä laadukkaalla työohjauksella. Työohjaukseen käytetty resurssi on hyvin tärkeää monelta osin, asiakkaalla on nimetty yhteys, johon voi ottaa aina yhteyttä, työntekijöiden töiden teko on selkeämpää ja todennäköisesti asiakas

saa laadukkaamman lopputuloksen nopeammin, kuin huonolla työnohjauksella tehdyssä prosessissa.

6.5 Tekoäly ja koneoppiminen työnohjauksessa

Lean-metodeissa työnohjaus on tärkeässä roolissa. IT-alalle sovitettuna apuvälineenä voitaisiin käyttää työkaluina tekoälyä ja koneoppimista. Järjestelmät voisivat tehdä työnohjausta sekä raportoida tapahtumia. Vaiheet, jotka ovat vielä nykyisin manuaalityötä, voisivat tulevaisuudessa olla järjestelmien tekemiä.

Tekoälyn eli AI:n (Artificial Intelligence) eräs määritelmä on itsenäinen ja muokautuva järjestelmä, joka pystyy ratkaisemaan ongelmia. Koneoppiminen on yksi tekoälyn osa-alue, jota tekoälyratkaisuihin voidaan käyttää. Koneoppimisen menetelmät voidaan jakaa karkeasti kolmeen eri osaan, ohjattuun oppimiseen, ohjaamattomaan oppimiseen sekä vahvistettuun oppimiseen. Ohjatussa oppimisessa järjestelmälle annetaan syötteitä ja niihin liittyviä palautearvoja ja järjestelmän tulee oppimisen jälkeen palauttaa oikeita vastauksia oppimisen pohjalta. Ohjaamattomassa oppimisessa järjestelmän tulee päätellä syötteistä rakenteita, joiden pohjalta lopputulos muodostuu. Vahvistetussa oppimisessa järjestelmän päätöksille annetaan palautetta ja järjestelmä vahvistaa omia päätöksiään palautteiden perusteella. (Reaktori Oy & Helsingin yliopisto 2018.)

Työnohjauksessa voidaan käyttää tulevaisuudessa tekoälyä ja koneoppimista, ohjatun ja vahvistetun oppimisen malleilla järjestelmä pystyy ohjaamaan tehtäviä työntekijöille, arvioimaan prioriteetteja, aikatauluttamaan tehtäviä ja ehkä jopa tekemään rakenteellisen tehtäväkuvauksen esimerkiksi asiakkaan kanssa käydyn chatbot-keskustelun pohjalta. Tämä tulee olemaan kustannustehokasta, vähentämään rutiininomaisten työnohjaustehtävien määrää sekä parantamaan laatua niissä kohteissa, joissa työnohjauksen kanssa on kehitystarpeita. IT-palvelutuotanto ei eroa tältä osin muiden alojen palvelutuotannosta.

7 Ketterä kehittäminen

7.1 Ohjelmistotuotannon menetelmät palvelutuotannossa

Ohjelmistotuotannon menetelmät ovat usein suunniteltuja projektiluontoisen tehtävän tekemiseksi, jonka jälkeen tuotos siirtyy ylläpitovaiheeseen. Palvelutuotannossa tehdään projekteja sekä ylläpitoa ja ylläpito ei tarkoita vain olemassa olevan järjestelmän virheiden korjaamista ja käynnissä pitoa, vaan myös muutosten tekemistä sekä uusien ominaisuuksien kehittämistä.

Palvelutuotannossa ohjelmistotuotantoa tehdään pääasiassa toimittajanäkökulmasta eli toteutetaan asiakkaalta saatujen vaatimusten perusteella järjestelmiä. Palvelutuotannossa myös useimmiten räätälöidään järjestelmät asiakkaan toiveiden mukaisesti, että asiakkaan toiveet tulisivat mahdollisimman hyvin toteutetuiksi. Järjestelmien elinkaari tulisi suunnitella myös ylläpitovaiheessa siten, että osattaisiin määritellä missä vaiheessa järjestelmää kannattaa vielä päivittää ja panostaa ylläpitovaiheen kehitykseen ja missä vaiheessa uuden järjestelmän suunnittelu ja käyttöönotto pitäisi tehdä ja vanha järjestelmä ajaa alas. (Haikala & Mikkonen 2011, 19–23.)

7.2 Ketterän kehityksen menetelmät

2000-luvun alussa perinteisten ja suunnitelmaperustaisesti raskaiden järjestelmäkehityksen prosessien vastapainoksi oli alkanut kehittyä kevyempiä menetelmiä, joita alettiin kutsua ketteriksi (Agile) menetelmiksi. Perusajatuksena on, että päämääränä ovat korkea asiakastyytyväisyys sekä vaatimukset täyttävä laadukas ohjelmisto. Raskaampiin menetelmiin verrattuna ajatuksena on, että prosessia ei tehdä prosessin vuoksi, dokumentaatiota tehdään siksi, että siinä dokumentoidaan olennaiset asiat ja etsitään käyttöön työhön parhaiten sopivat työkalut. (Haikala & Mikkonen 2011, 43–44.)

7.3 Agile Manifesto

Ketterän kehityksen menetelmiä kehittää ja tukee voittoa tavoittelematon Agile Alliance -järjestö, jonka tarkoitus on edesauttaa parempaa ohjelmistojen kehitystä nyt ja tulevaisuudessa. Järjestö tarjoaa tukeaan ja tietoaan kaikkien ohjelmistokehityksestä kiinnostuneiden käyttöön. Agile Alliance sai alkunsa vuonna 2001 julkaistessaan Agile Manifeston (taulukko 2), johon kirjatut periaatteet toimivat ketterien kehitysmenetelmien perustana. (Agile Alliance 2018.)

Taulukko 2. Agile Manifesto (Agile Manifesto 2001).

Agile Manifeston periaatteet

Tärkein päämäärämme on tyydyttää asiakastarpeet hyvissä ajoin alkavilla ja jatkuvilla ohjelmistotoimituksilla.

Muuttuvat vaatimukset on hyväksyttävä missä vaiheessa toimitusta tahansa. Ketteryys on asiakkaan etu.

Toimita ohjelmistoa jatkuvasti lyhyisiin aikaväleihin tähdäten.

Liiketoiminnan ja kehittäjien tulee työskennellä yhdessä koko projektin ajan.

Rakenna projektit motivoituneiden ihmisten kanssa. Luo ympäristö ja tuki, jota tarvitaan työn tekemiseksi.

Tärkein ja tehokkain metodi kommunikaatiokeinoksi ovat keskustelut ihmisten välillä.

Toimiva ohjelmisto on edistymisen tärkein mittari.

Ketterät prosessit edistävät tasaista kehitystä. Kaikkien pitäisi edetä tasaisella tahdilla.

Huomioid teknisesti hyviin ratkaisuihin ja hyvään suunnittelun edistävät ketteryys.

Yksinkertaista, on olennaista osata keskittyä oikeaan tekemiseen.

Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseohjautuvissa yksiköissä.

Yksikön pitäisi selvittää säännöllisesti miten se tulisi entistä tehokkaammaksi ja toimia selvityksen jälkeen sen mukaisesti.

Agile Manifeston kirjoittamisen aikaan etätyöskentelymahdollisuudet ja tietoliikenneyhteydet eivät olleet samaa tasoa kuin nykyisin. Kehitysyhteisösovellukset niin yhtiöiden sisäisissä tietoverkoissa kuin pilvipalveluissakaan eivät tarjonneet tuolloin samaa käytettävyyttä kuin nykypäivänä, joten osa listasta voitaisiin

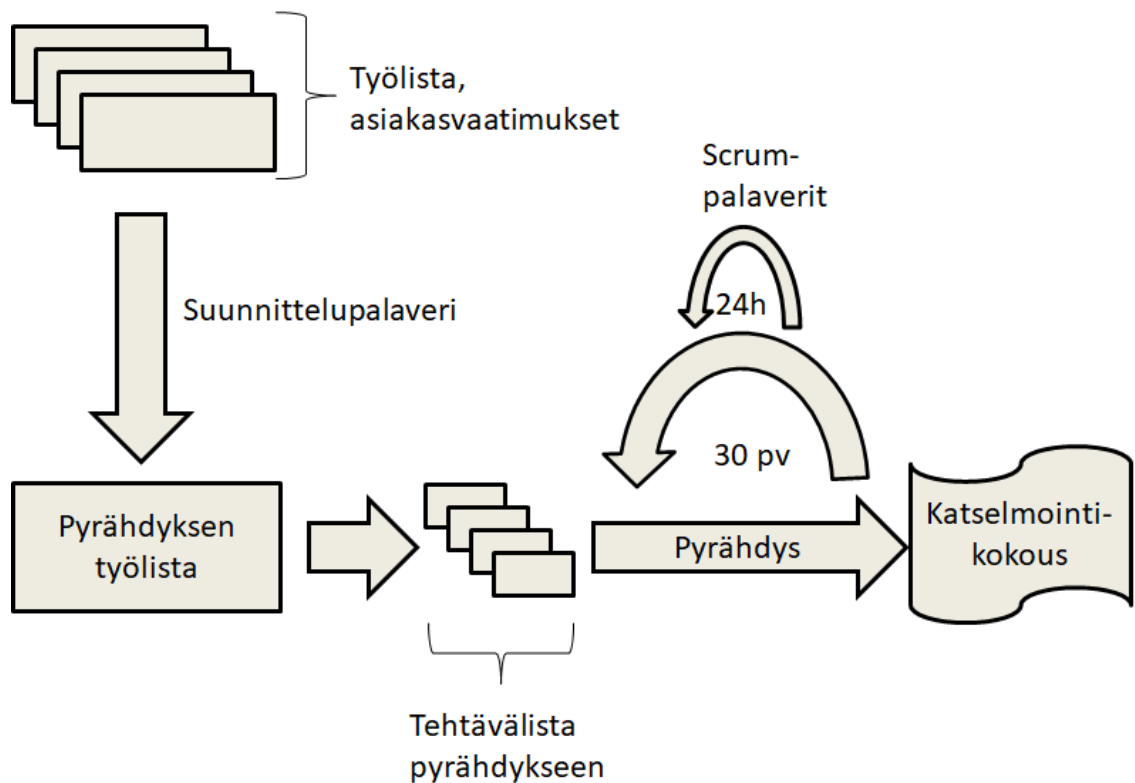
päivittää vastaamaan nykyhetkeä vähintään kommunikaatioon liittyvien periaatteiden osalta. Toisaalta Agile Manifesto on omalta osaltaan ollut vaikuttamassa sovellusten kehitykseen ja ilman sitä tänä päivänä ei välttämättä olisi loistavia sovelluksia helpottamaan yhteistyötä ja työnohjausta ja siten parantamaan ketterän kehityksen malleja työkalujen osalta.

7.4 Scrum

Scrum on ketterä menetelmä, joka on alun perin kehitetty tuotekehitykseen Japanissa 1980-luvulla. Ohjelmistokehityksen tarpeisiin Scrum-menetelmää jatkokehitettiin Yhdysvalloissa 1990-luvulla. Scrum on teoriassa yksinkertainen iteratiivinen eli kehityskierroksia toistava metodi, joka antaa raamit prosessille, mutta ei ota kantaa miten ja millä järjestelmää tai ohjelmistoa kehitetään. (Haikala & Mikkonen 2011, 46–47.)

Puhdasoppisessa Scrum-prosessissa on kolme roolia, tuotteen omistaja, scrum-mestari (Scrum Master) sekä toteuttajien tiimi. Tuotteen omistaja ylläpitää työlistaa ja priorisoi työlistan alkioita. Vaatimustenhallinta on tuotteen omistajan vastuulla. Scrum-mestari on eräänlainen iteraation projektipäällikkö: hän vastaa siitä, että toimitaan Scrumin mukaisesti, vastaa lopputuloksista, tiimin työkuormasta ja tiimin etenemistä estävien asioiden poistamisesta. Tiimin tulisi koostua kokoonpanosta, jossa on useita erilaisia henkilöitä, kuten ohjelmistoarkkitehti, ohjelmoijia ja testaajia. (Haikala & Mikkonen 2011, 48–49.)

Tiimi vastaa omasta työskentelystään, pilkkoo vaaditut tehtävät sopiviksi paloiksi, aikatauluttaa tekemistään sekä määrittää mitkä tehtävät otetaan seuraavaksi työn alle. Kehitystä tehdään pyrähdyksissä (Sprint), jotka kestävät ennalta määritetyn ajan, esimerkiksi 30 kalenteripäivää, ja joihin tehdään työlista, joka pysyy muuttumattomana pyrähdysten ajan. Päivittäin käydään läpi tilannetta lyhytkestoisissa Scrum-palavereissa. Pyrähdysten lopussa käydään läpi siihen asti tehty toteutus katselmointitilaisuudessa, kerätään palaute ja jatketaan seuraavalle pyrähdyskierrokselle. 30 päivän kestoinen Scrum-prosessi piirrettynä kuvassa 6. (Haikala & Mikkonen 2011, 48–51.)



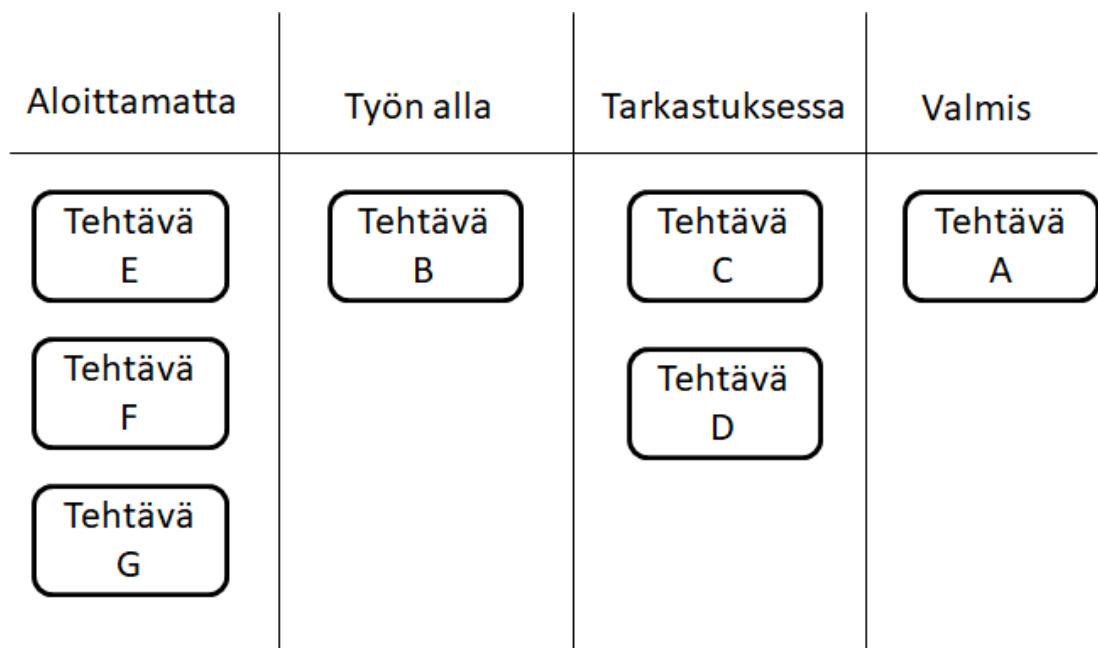
Kuva 6. Scrum-prosessi (Haikala & Mikkonen 2011, 48).

Ajatuksena Scrum-prosessissa on, että tuotteen omistaja ja mahdolliset liiketoiminnan edustajat saavat lyhyehköllä aikavälillä mielikuvan siitä, mitä tiimi on tehnyt ja kohtasiko toteutus vaatimuksen. Perinteisessä järjestelmäkehityksen metodeissa tuotosta on saatettu työstää pitkäkin aika ja tilaajan nähdessä lopputuloksen se ei olekaan vastannut sitä, mitä tilaaja ajatteli ja minkä pohjalta vaatimukset oli tehty. Järjestelmäkehityksen vaatimusten ymmärtämisessä sattuu usein tulkintavirheitä, siksi niiden läpikäymistä asiakkaan ja toimittajan välillä ei voi liikaa korostaa.

7.5 Kanban

Kanban on Lean-metodi, joka on sovitettu mukaan ketteriin menetelmiin. Se on työnhallintatapa, jonka periaatteita ovat työn kulun visualisointi, kaistanrajoittimet sekä läpimenoajan mittaaminen. Työn kulun visualisoinnissa jokaiselle tehtävälle on oma sarakkeensa esimerkiksi *aloittamatta*, *työn alla*, *tarkastuksessa*

ja *valmis*. Jokainen työ on omalla Kanban-kortilla ja siirtyy taululla omaan sarakkeeseensa työvaiheen mukaan. Kaistanrajoitin tarkoittaa, että yhdessä vaiheessa voi olla vain tietty määrä töitä. Läpimenoajan mittaamisella on mahdollista mitata prosessin pullonkauloja ja kehittää prosessia sitä kautta. Ohjelmistotuotannossa läpimenoaikojen mittaaminen voi olla haastavaa, mikäli tehtävät eivät ole samanlaisia keskenään. Alun perin Kanban on kehitetty teollisuusprosesseihin kuten Leankin, joten kaikki hyödyt eivät välttämättä toteudu samoin ohjelmistotuotannon prosesseissa kuin ne toteutuvat valmistavan teollisuuden prosesseissa. Esimerkki Kanban-tilasta on esitettyä kuvassa 7. (Haikala & Mikkonen 2011, 54–55).



Kuva 7. Kanban-tila.

8 Nykytilan kartoitus

8.1 Suunnitelma toteutusprosessien selvittämiseksi

Pohjois-Karjalan tietotekniikkakeskuksen järjestelmäsuunnittelun parissa työskentelevien henkilöiden palvelutuotantotoiminnan kartoitus on tärkeässä osassa, jotta kehityskohteita löytyisi mittausten ja tämän jälkeisen analyysin avulla. Järjestelmäsuunnittelun tehtävät poikkeavat henkilöiden välillä hyvinkin paljon toisistaan, tehtävät voivat kohdistua täysin sovellustasolla substanssiohjelmistoihin kuten talousohjelmistoihin, tai sitten järjestelmätasolla esimerkiksi tietokantoihin. Tästä johtuen mittarit on valittava sen mukaan, että ne soveltuvat kaikille henkilöille, eikä yksittäisten henkilöiden työnkuva vaikuta mittauksiin vääristävästi.

Mittarit, joilla nykytilaa lähdetään analysoimaan, ovat kartoitus työkuormasta tehtävien määränä mittausten alkaessa, saapuvien tehtävien määrä ja tulosuunta, valmistuneiden tehtävien määrä sekä keskeytysten määrä ja aiheuttaja. Koska tehtävät eivät ole vertailukelpoisia keskenään, tehtäville tehdään karkea jako työmääräarviolta alle 1 h:n tehtäviin, 1 – 4 h:n tehtäviin, 4 – 8 h:n tehtäviin sekä määrittelemättömiin tehtäviin. Määrittelemätön tehtävä on suurempi tehtävä, jota ei ole pilkottu pienempiin osiin. Määrittelemätöntä tehtävää ei oteta työn alle ilman pilkontaa pienempiin osiin, jossa maksimi on 4 - 8 h tehtävä.

Mittarien tulosten perusteella on tarkoitus analysoida, onko toiminta prosessivai virtaustehokasta ja miten keskeytykset vaikuttavat virtaustehokkuuteen. Analyysiin pohjautuen tuotetaan kehitysajatuksia Leanin ja ketterän kehityksen mallien mukaisesti.

8.2 Läpikävely prosessissa - Gemba

Kaikki järjestelmäsunnittelun työpyynnöt eivät tule suoraan asiakkaalta toteutukseen vaan osa työpyynnöistä tulee Pohjois-Karjalan tietotekniikkakeskuksen palvelupyynnöitä vastaanottavan service desk -palvelun kautta järjestelmäsunnittelulle. Kokonaisprosessiin perehtymiseksi tehdään service deskin työpyyntöjen vastaanottoon tutustuminen Leanin mukaisena Gemba-läpikävelynä.

Gemba-läpikävelyn tarkoituksena on saada selville suorittavalta tasolta, miten asiat todellisuudessa tapahtuvat eli yhtiön johto tai prosessin tarkkailija jalkautuu prosessiin. Prosessia seurataan tarkkailemalla ja kysymyksiä esittämällä suorittavalle tasolle, havainnot kirjataan ylös ja analyysi tehdään läpikävelyn jälkeen. Läpikävely tehdään asiakkaan näkökulmasta ja siinä pyritään ymmärtämään prosessia. Asiakasnäkökulma voi olla niin sisäinen kuin ulkoinenkin. (Torkkola 2015, 125–126.)

Sari Torkkola on teoksessaan 'Lean asiantuntijatyön johtamisessa' valinnut Gemba-läpikävelyn toteuttamiseen sopivia kysymyksiä, jotka on listattu taulukkoon 3.

Taulukko 3. Gemba-läpikävelyn esimerkkikysymykset asiantuntijaorganisaatiossa (Torkkola 2015, 237–239).

Gemba-läpikävelyn kysymykset

1. Mitkä ovat tämän palvelun/toiminnan ongelmat asiakkaan tai oman organisaation kannalta?
2. Kuka on vastuussa tästä palvelusta asiakkaan näkökulmasta päästä päähän?
3. Miten työpyynnöt saadaan asiakkaalta?
4. Kuka aikatauluttaa ja priorisoi toiminta asiakastilausten perusteella? Minkälaisia priorisointisääntöjä on käytössä?
5. Kuinka sujuvaa työ on?
6. Miten työt siirretään priorisoinnin jälkeen tiimistä toiseen?
7. Miten tarvittavat tiedot toimitetaan tähän työvaiheeseen?
8. Miten kehittäisit tätä työvaihetta?

9 IT-palvelutuotannon mittaaminen

9.1 Saapuneet ja toteutetut palvelupyynnöt

Järjestelmäsuunnittelun palvelupyyntöjen mittaamiset suoritettiin joulukuussa 2018 kahden työviikon aikana. Mittaamisen kirjaamisen tueksi toteutettiin excel- taulukko, joka oli mahdollisuus tulostaa tai täyttää sähköisesti. Taulukko lähetettiin saatesanoilla ja ohjeistuksella kaikille mittauksiin pyydytyille henkilöille, mittauksiin vastasi 1 henkilö yhdestätoista mittauksiin pyydytystä henkilöstä.

Taulukossa 4 on esitetty mittausten alussa keskeneräisenä olleet työt suunnitelman mukaisella jaolla eli alle 1 h:n tehtävät, 1 – 4 h:n tehtävät sekä 4 – 8 h:n tehtävät. Määrittelemättömiä tehtäviä ei ollut mittausjakson alkaessa tai ne oli pilkottu määriteltyihin rajoihin.

Taulukko 4. Keskeneräiset palvelupyynnöt mittausjakson alussa.

Keskeneräiset työt	Alle 1h	1 - 4h	4 - 8h
1. päivä	5	3	5

Taulukossa 5 on esitetty päivätasolla saapuneiden palvelupyyntöjen lukumäärät. Huomioitavaa on, että mittausjakson aikana ei saapunut töitä, jotka olisivat olleet työmääräarvioltaan yli 8 tuntia kestäviä.

Taulukko 5. Saapuneet palvelupyynnöt.

Saapuneet työt	Alle 1 h	1 – 4 h	4 – 8 h
1. päivä	4	1	0
2. päivä	3	1	0
3. päivä	4	2	0
4. päivä	2	2	0
5. päivä	1	2	1
6. päivä	6	0	0
7. päivä	0	0	1
8. päivä	2	1	0
9. päivä	2	0	2
10. päivä	6	1	0

Taulukossa 6 on esitetty toteutettujen palvelupyyntöjen määrät päivätasolla mittausjakson aikana.

Taulukko 6. Toteutetut palvelupyynnöt.

Toteutetut työt	Alle 1 h	1 – 4 h	4 – 8 h
1. päivä	4	1	
2. päivä	2	1	1
3. päivä	3	1	
4. päivä	5	1	
5. päivä		1	
6. päivä	3		
7. päivä	3	1	1
8. päivä	4	1	
9. päivä	3	1	1
10. päivä	5	1	

9.2 Keskeytykset

Keskeytysten mittaamiseksi tuotettiin taulukko kymmenen työpäivää kestävä mittaamisen kirjaamisen avuksi. Valitut keskeytystyypit olivat sähköpostin aiheuttama keskeytys, kollegan aiheuttama suullinen keskeytys, pikaviestimen eli tässä tapauksessa yhtiössä käytetyn Skype-ohjelman aiheuttama keskeytys, puhelimen aiheuttama keskeytys puhelulla tai tekstiviestillä sekä kokouksen aiheuttama keskeytys. Näiden lisäksi mittauksiin valittiin tehtäviin liittyvät keskeytystyypit tilanteista, joissa tehtävät eivät pääse jonkin itsestä riippumattoman

asian vuoksi etenemään tai tehtävä keskeytyy korkeamman prioriteetin eli tärkeämmän tehtävän vuoksi. Mittauksen vastaaja ohjeistettiin lisäämään taulukkoon keskeytystyyppejä, mikäli jokin keskeytystyyppi puuttuisi. Vastaajia ohjeistettiin myös, että lounas- ja kahvitaukoja sekä WC-käyntejä ei tarvitsisi merkitä taulukkoon, lopullisen analyysin kannalta nämä ovat merkityksettömiä mittauskohteita, koska niihin ei voida vaikuttaa lakisääteisyytensä sekä ihmisen luonnollisen toiminnan kannalta. Keskeytysten mittaustulokset on esitelty taulukossa 7.

Taulukko 7. Keskeytysten mittaustulokset.

Keskeytykset työhön	Sähköposti	Kollega	Pika-viestin / Skype	Puhelin	Kokous	Tehtävä ei etene ja tehtävä vaihtuu	Tehtävä keskeytyy korkeamman prioriteetin tehtävän vuoksi
1. päivä	15	3	6		1	1	2
2. päivä	10	4			1		1
3. päivä	12	7	12	2	2		
4. päivä	20	2	2	1	1	2	1
5. päivä	7	6	7		1		
6. päivä	12	8	4	2			1
7. päivä	8	3	4	1	1	1	
8. päivä	9	2	2	1	2	1	2
9. päivä	8	6	2	1	2		
10. päivä	15	5	1		1	2	

9.3 Gemba-läpikävelyn tulokset

Gemba-läpikävelyn esimerkkikysymykset ovat esitetty taulukossa 3 (Torkkola 2015, 237 - 239). Esimerkkikysymykset soveltuivat sellaisenaan hyvin IT-palvelutuotantoon ja Gemba-läpikävely toteutettiin perjantaina 28.12.2018 Pohjois-Karjalan tietotekniikkakeskuksen tiloissa ICT-arkkitehtina toimivan opinnäytetyön tekijän ja service desk palveluista vastaavan esimiehen kesken. Läpikävelyn tulokset on esitetty taulukossa 8.

Taulukko 8. Gemba-läpikävelyn tulokset.

Kysymys 1: Mitkä ovat tämän palvelun/toiminnan ongelmat asiakkaan tai oman organisaation kannalta?
Vastaus 1: Järjestelmäsuunnittelulle service deskin kautta saapuvien palvelupyynnöiden ongelmia ovat, että service deskiin saapuu ongelmakuvauksia, joiden tarkkuus on puutteellinen tai asiakas ei osaa selittää ongelmaa ymmärrettävästi. Tästä johtuen palvelupyynnön kuvaus saattaa jäädä vaikeasti tulkittavaksi ja lisätietoja joudutaan pyytämään asiakkaalta. Myös osa suoraan asiantuntijoille tulevista palvelupyynnöistä ovat tiedoiltaan puutteellisia.
Kysymys 2: Kuka on vastuussa tästä palvelusta asiakkaan näkökulmasta päästä päähän?
Vastaus 2: Koko organisaatio on vastuussa palvelupyynnöiden vastaanotosta ja ongelman ratkaisusta.
Kysymys 3: Miten työpyynnöt saadaan asiakkaalta?
Vastaus 3: Palvelupyynnöt saadaan service deskiin tikettijärjestelmän, sähköpostin ja puheluiden kautta. Suoraan asiantuntijoille palvelupyynnöt voivat tulla myös pikaviestimen kautta tai suoraan asiakkaalta. Muut kuin tikettijärjestelmään suoraan tulleet palvelupyynnöt kirjoitetaan service deskissä tikettijärjestelmään. Asiantuntijoille suorat yhteydenotot pitäisi myös kirjata tikettijärjestelmään.
Kysymys 4: Kuka aikatauluttaa ja priorisoi toimintaa asiakastilausten perusteella? Minkälaisia priorisointisääntöjä on käytössä?
Vastaus 4: Asiakas voi pyytää palvelupyynnölleen nopeaa ratkaisua, service desk kuitenkin tekee päätöksen prioriteetista. Priorisointisäännöt, jotka koostuvat vaikutusalueesta ja kiireellisyydestä ovat määriteltyjä, mutta asiakkaan pyynnöt vaikuttavat näihin korottavasti. Korotetun prioriteetin tiketeille on omat käsittelysääntönsä, että tehtävä otettaisiin työn alle mahdollisimman nopeasti. Asiantuntijat määrittelevät itse kirjaamilleen tiketeille prioriteetit.
Kysymys 5: Kuinka sujuvaa työ on?
Vastaus 5: Palvelupyynnöiden vastaanottaminen sujuu hyvin, harvemmin käsiteltyjen järjestelmien osalta vastuullisen henkilön selvittäminen on haastavaa service deskissä.
Kysymys 6: Miten työt siirretään priorisoinnin jälkeen tiimistä toiseen?
Vastaus 6: Järjestelmä mahdollistaa kenen tahansa tukiorganisaation jäsenen siirtää tikettiä henkilöltä toiselle. Vastuuhenkilöitä selvitetään järjestelmäluettelon kautta.
Kysymys 7: Miten tarvittavat tiedot toimitetaan tähän työvaiheeseen?
Vastaus 7: Tiedot kerätään palvelupyynnön vastaanottamisen yhteydessä. Palvelupyynnöille kysytään lisätietoja tarvittaessa.
Kysymys 8: Miten kehittäisit tätä työvaihetta?
Vastaus 8: Knowledge basen eli tietämyskannan kehittäminen ja käyttöönotto voisivat auttaa työnohjausta ja mahdollisesti aikaisemman vaiheen ratkaisua. Myös tarkempi palvelupyynnön kuvauspohja voisi auttaa ongelman ratkaisussa tai ainakin vähentäisi tarkentavien tietojen kyselyä.

10 IT-palvelutuotannon mittauksen analyysi

10.1 Keskenkäydyt työt ja keskeytykset

Perusanalyysinä tutkitaan keskenkäydyt palvelupyynnöt määrää verrattuna keskeytyksiin. Keskenkäydyt palvelupyynnöt on laskettu alkutilanteessa kartoitettujen palvelupyynnöt määräästä taulukosta 4, joihin on lisätty saapuneet palvelupyynnöt taulukosta 5 ja joista on vähennetty toteutettujen palvelupyynnöt määrää taulukosta 6. Keskenkäydyt palvelupyynnöt päivätasolla on esitetty taulukossa 9.

Taulukko 9. Keskenkäydyt työt.

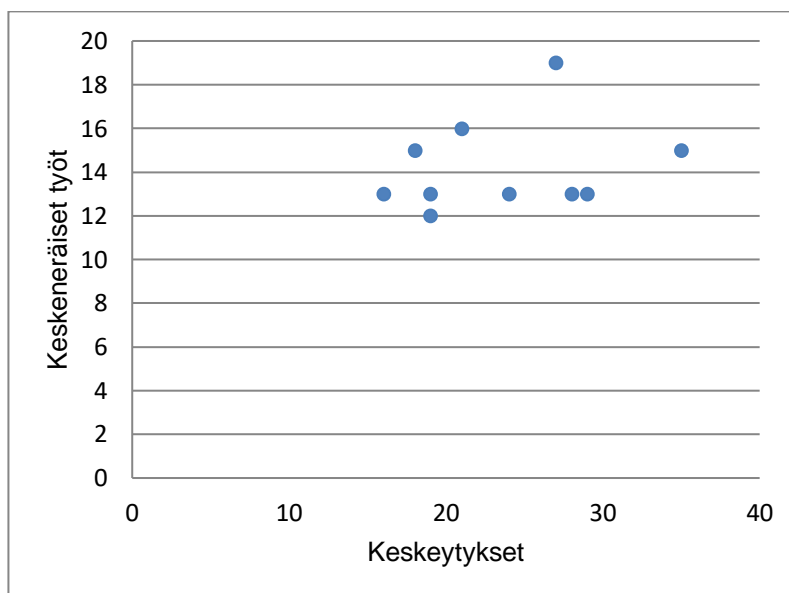
Keskenkäydyt työt	Alle 1 h	1 – 4 h	4 – 8 h
1. päivä	5	3	5
2. päivä	6	3	4
3. päivä	7	4	4
4. päivä	4	5	4
5. päivä	5	6	5
6. päivä	8	6	5
7. päivä	5	5	5
8. päivä	3	5	5
9. päivä	2	4	6
10. päivä	3	4	6

Keskenkäydyt palvelupyynnöt laskennan jälkeen voidaan verrata lukemia taulukon 7 keskeytyksien summiin ja tutkia vallitseeko asioiden välillä korrelaatiota eli ovatko asiat riippuvaisia toisistaan. Vertailussa on käytetty Pearsonin korrelaatiokertoimen laskennan tunnuslukua.

Tampereen yliopiston menetelmäopetuksen internet-sivuilla korrelaatio on esitetty seuraavalla tavalla: "Kahden muuttujan välisen riippuvuuden astetta voidaan nimittää yleisessä merkityksessä korrelaatioksi. Jos korrelaatio on voimakasta, voidaan toisen muuttujan arvoista päätellä toisen muuttujan arvot melko täsmällisesti. Jos korrelaatio on heikko, ei muuttujien välillä ole yhteisvaihtelua." (Tampereen yliopisto, 2004). Samalla internet-sivulla Pearsonin korrelaatiokerroin on esitetty näin: "Yleisin käytetty korrelaatiota kuvaava tunnusluku on Pear-

sonin tulomomenttikorrelaatiokerroin (r). Se on vähintään kahden intervallias- teikollisen muuttujan keskinäisen lineaarisen riippuvuuden voimakkuutta kuvaava tilastollinen tunnusluku.” (Tampereen yliopisto 2004.)

Microsoft Excel taulukkolaskentaohjelman PEARSON-funktio toteuttaa Pearsonin korrelaatiokertoimen laskennan (Office.com 2019). Funktiolla laskettuna keskeneräisten palvelupyyntöjen ja keskeytysten korrelaatioluvun arvoksi saadaan pyöristettynä 0,25. Laskentapisteen on esitetty graafisesti kuviossa 1.



Kuvio 1. Keskeneräisten töiden määrä verrattuna keskeytyksiin.

Korrelaatiokertoimen analyysin määritelmä on seuraavanlainen: ”Tulomomenttikorrelaatiokertoimen arvo vaihtelee välillä -1 - $+1$. Korrelaatiokertoimen ollessa 0 , ei muuttujien välillä ole lineaarista riippuvuutta. Vastaavasti arvoilla $(+/-) 1$ muuttujien välillä on täydellinen positiivinen / negatiivinen lineaarinen riippuvuus. Täydellisen lineaarisen riippuvuuden tapauksessa muuttujien kaikki arvot sijoittuvat hajontakuviassa samalle suoralle viivalle. Yleensä muuttujien välinen korrelaatiokerroin poikkeaa nolasta. Tämä voi johtua myös sattumasta. Korrelaatiokertoimen merkitsevyytensä avulla voidaan arvioida kertoimen tilastollista merkitsevyyttä.” (Tampereen yliopisto 2004.)

Pearsonin korrelaatiokertoimen tunnusluku 0,25 on positiivinen, mutta hyvin lähellä nollaa oleva, joten johtopäätöksenä tästä mittauksesta on, että asioiden välillä ei ole lineaarista riippuvuutta tai riippuvuus on hyvin vähäinen.

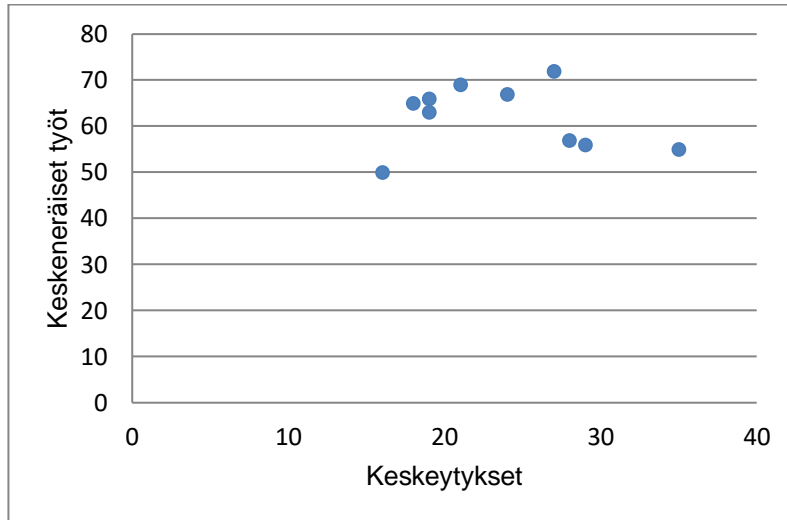
10.2 Keskenenäiset työt pisteytettynä ja keskeytykset

Kehittämällä luvussa 10.1 esitettyä perusanalyysiä, mittaustuloksista keskenenäisille palvelupyynnöille asetettiin kertoimet, koska kaikki työt eivät ole keskenään samanlaisia. Keskenenäiset työt olivat jaettuna kolmeen eri kategoriaan eli alle 1 h:n tehtävät, 1 – 4 h:n tehtävät sekä 4 – 8 h:n tehtävät. Tämän pohjalta palvelupyynnöille asetettiin vastaavat kertoimet alle 1 h:n tehtävät saavat arvoksi 1, 1 – 4 h:n tehtävät saavat arvoksi 4 ja 4 – 8 h:n tehtävät saavat arvoksi 8. Arvot on mietitty maksimituntiarvion mukaisesti. Pisteytetyt arvot ovat esitettyinä taulukossa 10.

Taulukko 10. Keskenenäiset työt pisteytettynä.

Keskenenäiset työt pisteytettynä	Alle 1 h =1	1 – 4 h =4	4 – 8 h =8
1. päivä	5	12	40
2. päivä	6	12	32
3. päivä	7	16	32
4. päivä	4	20	32
5. päivä	5	24	40
6. päivä	8	24	40
7. päivä	5	20	40
8. päivä	3	20	40
9. päivä	2	16	48
10. päivä	3	16	48

Laskentapistteet ovat graafisesti esitettyinä kuviossa 2.



Kuvio 2. Keskeneräisten töiden pisteytetyt arvot verrattuna keskeytyksiin.

Pearsonin korrelaatiokertoimella laskettuna keskeneräisten töiden pisteytettyjen arvojen ja keskeytysten tunnusluvuksi saadaan $-0,17$. Tunnusluku muuttuu negatiiviseksi, mutta lähenee nollaa joten johtopäätöksenä tästäkin analyysistä on, että keskeneräisten töiden ja keskeytysten välillä ei ole suoraa korrelaatiota.

10.3 Läpimenoajat ja Littlen laki

Keskimääräinen palvelupyynnön valmistumisnopeus on kappalemäärä per jaksoaika (Torkkola 2015, 189). Taulukossa 11 on esitetty mittausten valmistumisnopeudet.

Taulukko 11. Valmistumisnopeudet.

Työ	Valmistunut (kpl)	Jaksoaika (pv)	Valmistumisnopeus (kpl/pv)
Alle 1 h	32	10	3,2
1 – 4 h	9	10	0,9
4 – 8 h	3	10	0,3

”Keskimääräinen yhden tehtävän valmistumisaika on valmistumisnopeuden käänteisluku.” (Torkkola 2015, 189). Valmistumisaikat on esitetty taulukossa 12.

Taulukko 12. Valmistumisajat.

Työ	Tunteja päivässä	Valmistumisnopeus (kpl/pv)	Valmistumisaika (h)
Alle 1 h	8	3,2	2,5
1 – 4 h	8	0,9	8,9
4 – 8 h	8	0,3	26,7

Taulukon 9 pohjalta laskettujen keskeneräisten töiden 10 päivän keskiarvon perusteella ja taulukon 11 valmistumisnopeuksien perusteella voidaan laskea läpimenoajat eli keskeneräiset työt jaettuna valmistumisnopeudella (Torkkola 2015, 190). Tulokset on esitetty taulukossa 13.

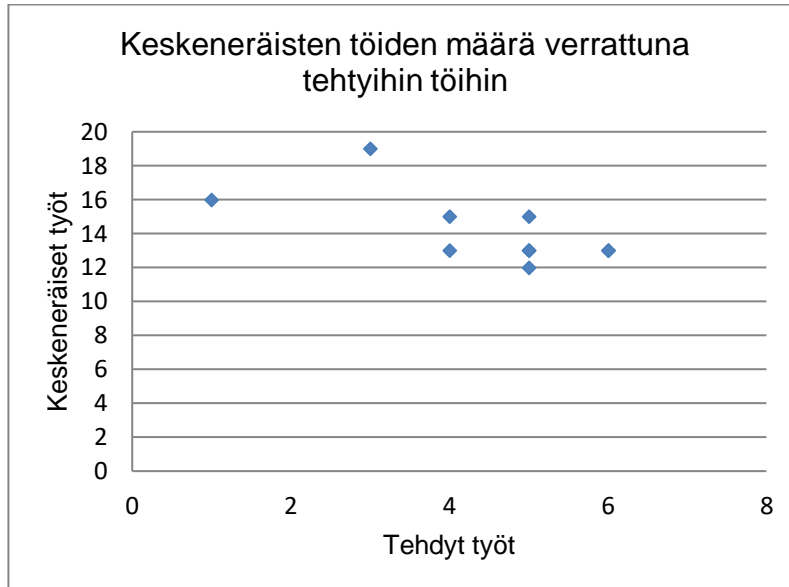
Taulukko 13. Läpimenoajat.

Työ	Keskiarvo keskeneräisille 10 pv	Valmistumisnopeus (kpl/pv)	Läpimenoaika (pv)
Alle 1 h	4,8	3,2	1,5
1 – 4 h	4,5	0,9	5,0
4 – 8 h	4,9	0,3	16,3

Littlen lain mukaan keskeneräisen työn määrä vaikuttaa läpimenoaikaan ja keskeneräisten töiden määrää tulisi saada pienemmäksi, mikäli läpimenoaikaa halutaan nopeammaksi (Torkkola 2015, 189–190).

Littlen lakiin pohjautuen taulukossa 9 esitettyjen keskeneräisten töiden ja taulukossa 6 esitettyjen valmistuneiden töiden välillä Pearsonin korrelaatiokertoimeksi saatiin 0,44. Laskentapistet on esitetty graafisesti kuviossa 3.

Pearsonin korrelaatiokertoimen tunnusluku 0,44 on positiivinen ja lähes puoleen väliin 0:n ja maksimi 1:n välille sijoittuva, joten johtopäätöksenä tästä mittauksesta on, että asioiden välillä on riippuvuussuhde. Mittauksena tästä voi hieman kriittisesti ajatella, että näin riippuvuuden pitääkin olla, kun töitä valmistuu, niin sama määrä töitä pitäisi myös poistua keskeneräisistä töistä, riippuvuussuhde on kuitenkin nyt todettu laskennalla. Mikäli riippuvuussuhdetta ei olisi, niin asiaa tulisi varmasti tutkia, mistä tulos voisi johtua.



Kuvio 3. Keskeneräisten töiden arvot verrattuna tehtyihin töihin.

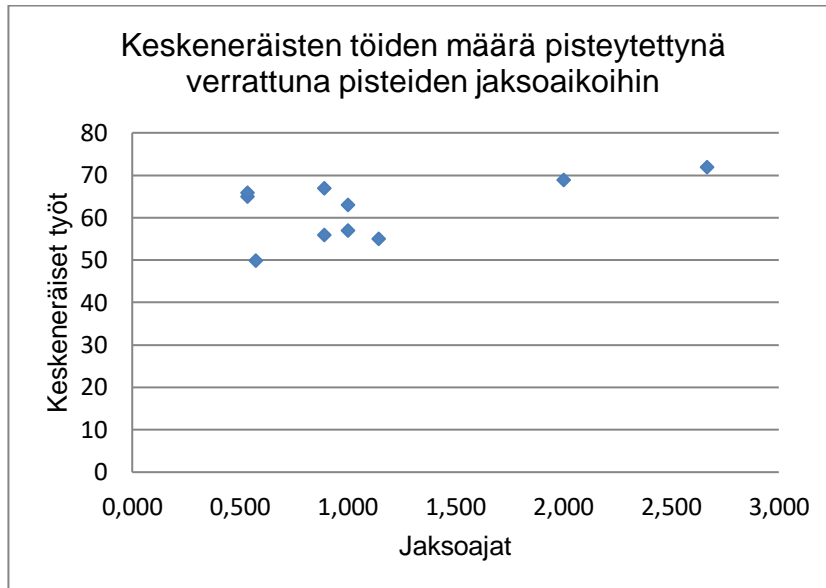
Littlen lakiin pohjautuen mitattiin myös läpimenoaikojen riippuvuutta keskeneräisiin töihin. Koska kaikkentyyppisiä 1 h:n, 1 – 4 h:n ja 4 – 8 h:n tehtäviä ei toteutunut joka päivä, mittaus toteutettiin päivittäin tehtyjen pisteiden perusteella jaksoaika laskien. Laskennan koontitaulukko on esitetty taulukossa 14.

Taulukko 14. Tehtyt ja keskeneräiset työt pisteytettynä sekä päiväkohtaiset jaksoajat laskettuna.

Tehtyt työt	Alle 1 h = 1	1 – 4 h = 4	4 – 8 h = 8	Tehtyjä pisteitä	Keskeneräisiä pisteitä	Jaksoaika per piste (8h/pistemäärä)
1. päivä	4	1		8	57	1
2. päivä	2	1	1	14	50	0,571
3. päivä	3	1		7	55	1,143
4. päivä	5	1		9	56	0,889
5. päivä		1		4	69	2
6. päivä	3			3	72	2,667
7. päivä	3	1	1	15	65	0,533
8. päivä	4	1		8	63	1
9. päivä	3	1	1	15	66	0,533
10. päivä	5	1		9	67	0,889

Jaksoaikojen ja keskeneräisten töiden välillä Pearsonin korrelaatiokertoimeksi saatiin 0,31. Laskentapistet on esitetty graafisesti kuviossa 4.

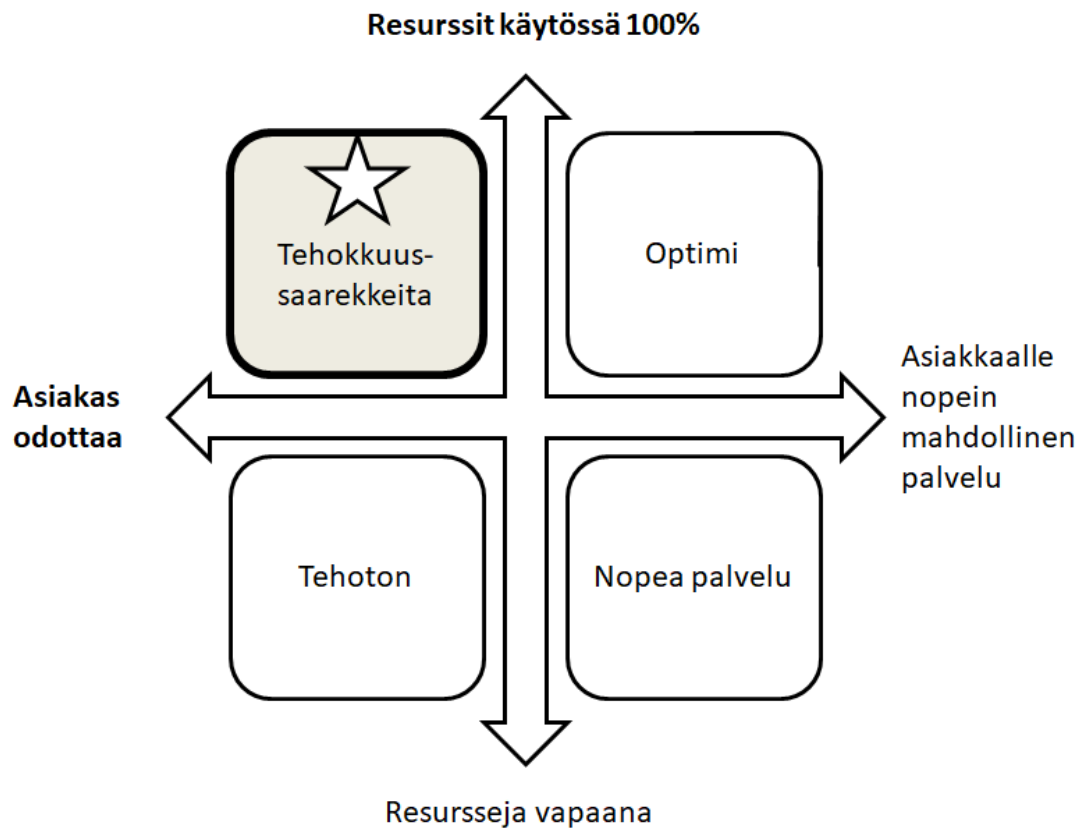
Pearsonin korrelaatiokertoimen tunnusluku 0,31 on positiivinen ja osoittaa heikkoa riippuvuussuhdetta asioiden välillä. Auki kirjoitettuna tulkintana voidaan siis todeta, että keskeneräisten töiden määrällä on riippuvuussuhde läpimenoaikojen kasvuun ja tämä todistaa Littlen lainmukaisuuden.



Kuvio 4. Keskeneräisten töiden arvoitetut määrät verrattuina jaksoaikoihin.

10.4 Sijoitus tehokkuusmatriisissa

Mittausten perusteella keskeneräisiä töitä oli koko ajan jonossa, jolloin organisaatio toimii Lean-ajatusmallin määrittymen mukaan resurssitehokkaasti. Tilanne on kuvattuna tehokkuusmatriisissa kuvassa 8.



Kuva 8. Resurssitehokas organisaatio tehokkuusmatriisissa.

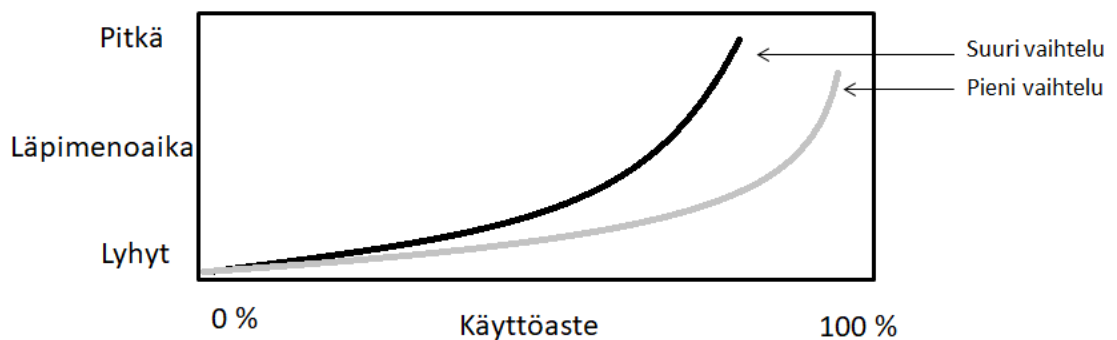
Analyysinä tästä on, että resurssit ovat täysin kuormitettuna ja asiakas odottaa töiden valmistumista. Jotta päästäisiin Lean-mallin mukaiseen virtaustehokkuuteen resursseja tulisi lisätä tai vastaanotettavien töiden määrää karsia.

10.5 Tehokkuusparadoksi käyttöasteesta ja läpimenoajoista

Mittausten perusteella resurssien käyttöaste on täydet 100 %, keskeneräiset työt eivät lopu koskaan ja tämä vaikuttaa läpimenoaikoihin Littlen lain mukaisesti (Torkkola 2015, 189). Tilanne on siis sama kuin kuvassa 8, mutta yhtiön toiminnan kannalta esitettynä. Keskeneräisen työn määrää pitäisi vähentää tai valmistumiseen vaikuttavaa resurssia tulisi lisätä, että päästäisiin lyhyempään läpimenoaikaan ja käyttöasteen virtaustehokkaaseen käyttöön.

Käyttöasteita ja läpimenoaikoja on esitelty Kingmanin yhtälössä, kun käyttöaste nousee niin läpimenoaika kasvaa eksponentiaalisesti. Kingmanin yhtälössä

myös töiden vaihtelu tuo oman merkityksensä, suuri vaihtelu työtehtävissä korostaa käyttöasteen ja läpimenoajan välistä riippuvuutta. Kingmanin yhtälöä havainnollistava graafi kuvassa 9. Kun käyttöaste nousee käyttöasteen lähentyessä 100 prosenttia, niin läpimenoaika kasvaa nopeammin kuin käyttöasteen ollessa lähempänä 0:n prosentin kuormitusta. Tämä johtuu monista syistä, teoreettisesti mm. keskeytyksiä, töiden järjestämistä ja saman työn moneen kertaan aloittamista tapahtuu enemmän käyttöasteen noustessa ja tämä vaikuttaa läpimenoajan kasvamiseen. (Modig & Ahlström 2013, 42–43, 50–55.)



Kuva 9. Kingmanin yhtälö graafina (Modig & Ahlström 2013, 42).

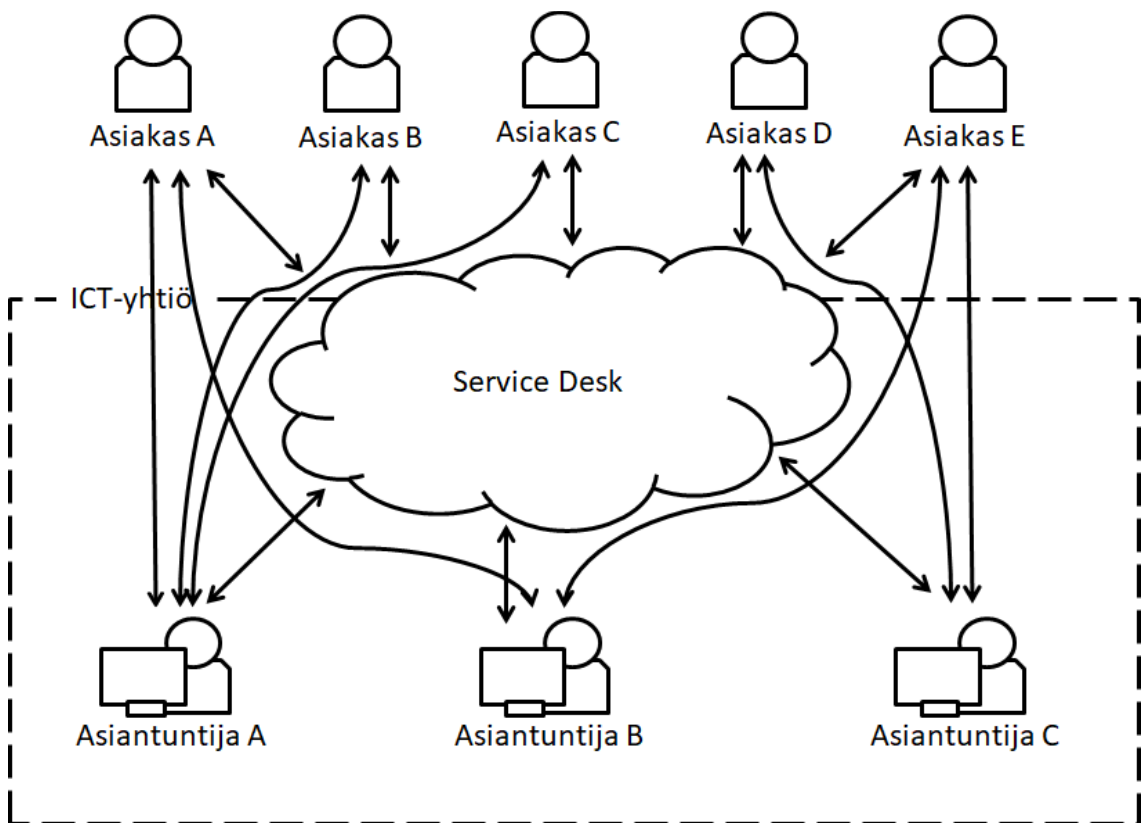
Tehokkuusparadoksi muodostuu siitä, että vaikka toimitaan resurssitehokkaasti, niin virtaustehokkuus on heikkoa. Paradoksi selittyy sillä, että resurssit tekevät paljon hukkaa eli töitä, jotka eivät lisää tuottavaa työtä. Ratkaisu tehokkuusparadoksiin on virtaustehokkuuden lisääminen ja käyttöasteen pienentäminen, tällöin tehdään arvoa tuottavia asioita ja hukkaa syntyy vähemmän pienemmällä käyttöasteella eli tehdään vähemmän mutta oikeita asioita. (Modig & Ahlström 2013, 65–67.)

10.6 Läpikävelyn analyysi

Läpikävelyn perusteella kehittämiskohteiksi löytyivät palvelupyynnöiden vastaanotto, palvelupyynnöiden kirjaaminen sekä palvelupyynnöiden ohjaus eteenpäin ratkaisuportaaseen.

Palvelupyynnöiden vastaanotossa ja käsittelyssä ei pitäydytä tiukassa prosessimallissa ja tämä saattaa kuormittaa työntekijöitä.

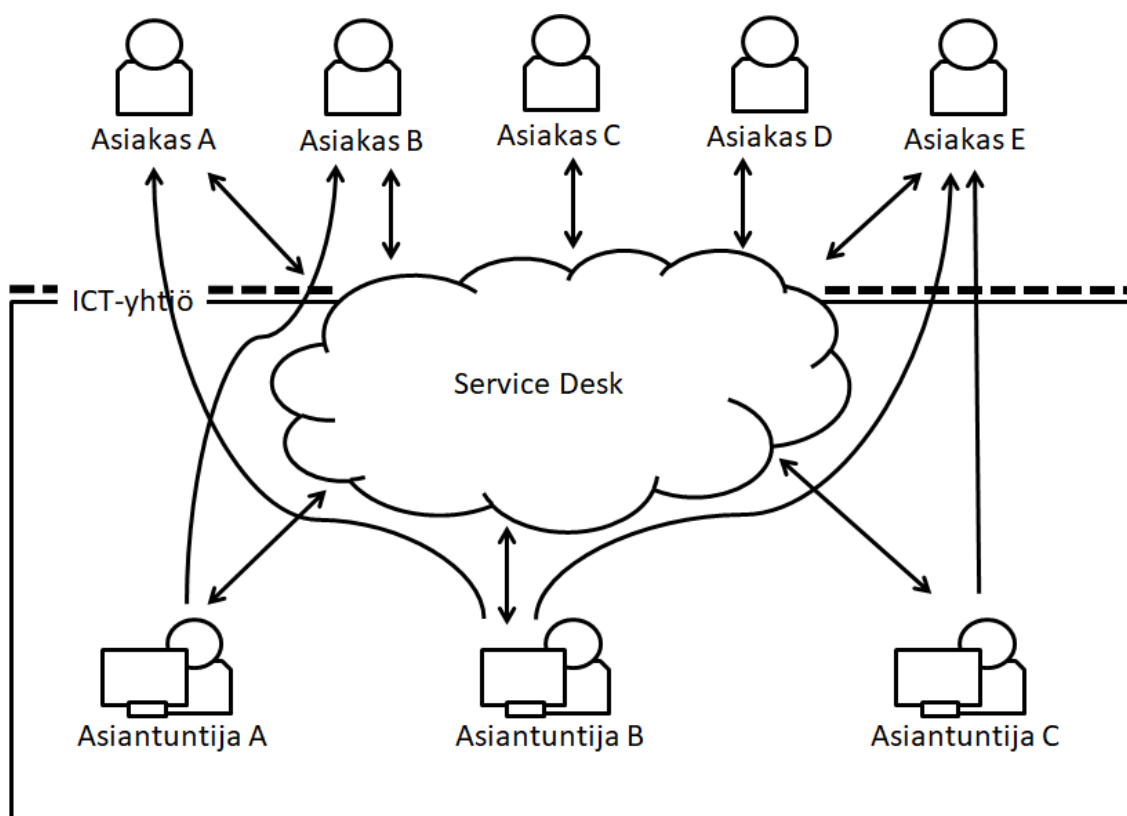
Palvelupyynnöiden käsittelyprosessin nykytila on kuvattu kuvassa 10.



Kuva 10. Palvelupyynnökäsittelyn nykytila.

Suorilta yhteydenotoilta asiakkailta suoraan asiantuntijoille tulisi välttyä ja jos palvelupyynnöt tulisivat yhden rajapinnan kautta, niin työn ohjaus vakiintuisi ja palvelupyynnökäsittelyn prosessoitumisaste nousisi. Havainnekuva uudesta tilasta kuvassa 11. Service desk toimii kuvassa suodattimena sekä työnohjauksen järjestelijänä. ICT-yhtiö on muuttunut läpipäästävästä tilasta läpipääsemättömäksi ja asiakas tekee palvelupyynnön aina service deskiin. Service desk ohjaa palvelupyynnön asiantuntijalle, joka selvittää asian service deskin välityk-

sellä tai suoraan asiakkaan kautta. Tärkeää kuitenkin on, että palvelupyynnön asiantuntijalle ei tule suoraan asiakkaalta, jolloin käsittelyn aloittamiseen ja sitä kautta asiantuntijan työkuorman tasaisuuteen on helpompi vaikuttaa.



Kuva 11. Palvelupyynnön käsittely korkeammalle prosessoituna.

Palvelupyynnön kirjaamisen kirjavuutta voitaisiin välttää esitäyttölomakkeella. Nykyiseen lomakkeeseen palvelupyynnön tekijä täyttää omat tietonsa sekä kuvauksen ongelmasta. Kehitysjatoksena tähän voisi olla lomakkeen kehittäminen tarkentavilla tiedoilla sekä palvelupyynnön vastaanottovaiheessa validointi joko asiantuntijapohjaisesti tai jopa tekoälypohjaisesti.

Läpikävelyssä tuli ilmi myös, että service deskissä vastaanotettujen palvelupyynnön edelleen ohjaaminen asiantuntijoille on joskus haasteellista. Hyvänä ehdotuksena edelleen ohjaamisen kehittämiseen tuli jo läpikävelyssä esille tietämiskannan kehittäminen. Tietämiskanta voisi kehittyä ICT-palveluyhtiön tapauksessa ratkaistujen palvelupyynnön perusteella ja siitä saisi myös selville eri järjestelmistä tietävien henkilöiden nimet.

10.7 Mittausten kehitysajatuksia

Mittaukset onnistuivat muutoin suunnitellulla tavalla, mutta mittausten luotettavuutta heikensi vastaajakato. Analyysien perusteella mittauksissa työkuormien ja keskeytysten välillä ei löydetty suoraa riippuvuussuhdetta, joten mittausten perusteella ei voida tehdä päätelmää, että keskeytykset suoraan vaikuttaisivat virtaustehokkuuteen. Keskeneneräisen työn ja tehtyjen töiden välillä sekä keskeneneräisten töiden ja läpimenoaikojen löydettiin riippuvuussuhde, jonka perusteella keskeneneräisten töiden määrän kasvaessa toteutuneiden töiden suhteellinen osuus laskee.

ICT-palvelualan työskentelyä ajatellen työn teon kannalta keskeytyksiltä ei pystytä välttymään, mutta toisaalta lukumäärä ei näytä korreloivan suoraan työn tekemiseen ja virtaustehokkuuteen. Mittausteknisesti ajatellen vastaavaa mitausta tehtäessä kannattaisi yrittää mitata, minkä verran aikaa keskeytyksien käsittelyyn kului, analyysin lopputulos voisi olla erilainen jos keskeytyksille saataisiin aikakerroin. Myös töiden vaihtelua olisi hyvä mitata tarkemmin, miten paljon tehtävät vaihtelevat keskenään, nyt toteutus vaihtelusta oli melko karkealla asteikolla ja mittaustuloksiakin syntyi melko vähän.

Keskeneneräisten töiden määrä kuitenkin vaikuttaa tehtyjen töiden määrään ja töiden läpimenoaikaan, tästä voidaan tehdä johtopäätös Littlen lain mukaisesti, että jos keskeneneräisten töiden määrää saataisiin lähemmäksi nolaa, niin läpimenoajat nopeutuisivat. Keskeneneräisten töiden analysoinnissa olisi myös järkevää tutkia, onko osa keskeneneräisistä töistä ollut pitkään jonossa ja mikäli jokin tehtävä on jäänyt pitkäksi aikaa keskeneneräiseen tilaan, niin se olisi yksi selvitetävä asia. Mittauksiin liittyvänä havaintona myös mittauksiin osallistujien kattavuutta olisi hyvä saada korkeammaksi luotettavampien tulosten saamiseksi, joten motivaatiota vastaamiseen tulisi saada paremmaksi joko vastaajalähtöisesti tai toimeksiantajan puolelta kannustamalla.

Gemba-läpikävely kannattaisi myös toteuttaa uudestaan, kun sen pohjalta luotua kehitystä on testattu jonkin aikaa. Olisivatko havainnot yhä samoja vai olisiko huomio kehityskohteissa keskittynyt muihin asioihin.

11 IT-alan työskentely ja kehitysajatuksia

11.1 Kehitysajatuksia

Teoriaosuuksissa ja mittaus- sekä analyysiosuuksissa on tutkittu Lean-periaatteiden ja ketterän kehityksen malleja sekä tehty konkreettiset mittaukset ja analyysit tuloksista. Analyyseissä selvimmin esille päässyt kehityskohde oli keskeneräisen työn minimoiminen, tähän päästäkseen toiminnan kehittämistä voidaan tehdä monella tavalla. Vaihtoehtoja kehitystoimenpiteille on useita, seuraavissa luvuissa esitellään kehitystekniikoita ja toimenpiteitä, joiden avulla olisi mahdollista päästä sujuvampiin prosesseihin ja sitä kautta keskeneräisen työn minimointiin. Toiminnan kehittäminen sujuvammaksi johtaisi mahdollisesti myös parempaan tyytyväisyyteen niin asiakkaan kuin henkilöstönkin näkökulmasta. Toisin sanoen näillä tekniikoilla päästään kohti Lean-ajatusmallia ketterän kehityksen metodeilla lisättynä. Tekniikat eivät ole ehdottomia vaan ennemminkin suosituksia toiminnan kehittämiseksi, testaamatta ja mittaamatta ei voida päätellä, autoiko tekniikka vai ei.

Ajatusmaailman kannalta teoriaosuuksissa esitellyt Lean-ajatusmallin mukaiset toimintatavat sekä ketterän kehityksen ajatusmalli auttavat omaksumaan uusien tekniikoiden käyttöönottoa IT-palvelutuotannossa. On hyvä ymmärtää suurempi kokonaisuus ennen menemistä yksityiskohtaiselle tasolle, Lean-ideologian mukaisesti tiivistettynä ”näe kokonaisuus”.

11.2 Multitasking ja mindfulness

Keskittymiskykyä tarvitaan nykyaikaisessa tietotyössä vähintään yhtä paljon kuin aiemminkin. Häiriötekijöiden määrä on kuitenkin lisääntynyt moninkertaiseksi mukana kannettavien laitteiden ja työelämän viestinnän sekä viestintäsovellusten myötä. Ihmisen ominaisuuksiin kuuluu ärsykkeiden huomioiminen ja samalla tekemisen yrittäminen yhtäaikaaisesti. Neurologisesta näkökulmasta katsottuna ihminen ei kuitenkaan pysty tekemään useampaa asiaa yhtä aikaa vaan usean tehtävän suorittaminen yhtä aikaa vaatii tehtävästä toiseen vaihte-

lua nopealla tahdilla. Tällaisesta syntyy näkemys siitä, että ihminen pystyisi multitaskingiin eli moniajoon. Toisin kuin tietokoneisiin, ihmiseen ei saada lisää kapasiteettia lisäämällä suoritusydinten määrää vaikka nykyaikainen tietotyö sitä joskus vaatisikin. Koska kapasiteetin lisääminen ihmiseen ei ole mahdollista, on käytettävä tehokkaasti sitä mitä on käytettävissä. (Hougaard & Kalajo 2016, 26–27.)

Koska multitasking on jatkuvasti tehtynä kehoa ja mieltä rasittavaa, on mietittävä myös ratkaisuja multitaskingin hallitsemiseksi. Yksi ratkaisumalli on mindfulness, joka löyhästi suomennettuna tarkoittaa tietoisuustaitoa tai tietoista läsnäoloa. Ajatuksena on keskittymiskyvyn hallinnointi ja keskittäminen siihen asiaan mitä on tekemässä, suodattamaan ärsykkeet oman mielensä mukaan ja tehdä valinta tarttumisesta uuteen tehtävään tai jatkamaan käynnissä olevaa tehtävää. Periaate on kuitenkin se, että ihminen kohdistaa keskittymisensä vain yhteen tehtävään kerrallaan ja säätelee näin omaa tekemistään pois multitaskingista. (Hougaard & Kalajo 2016, 30–36.)

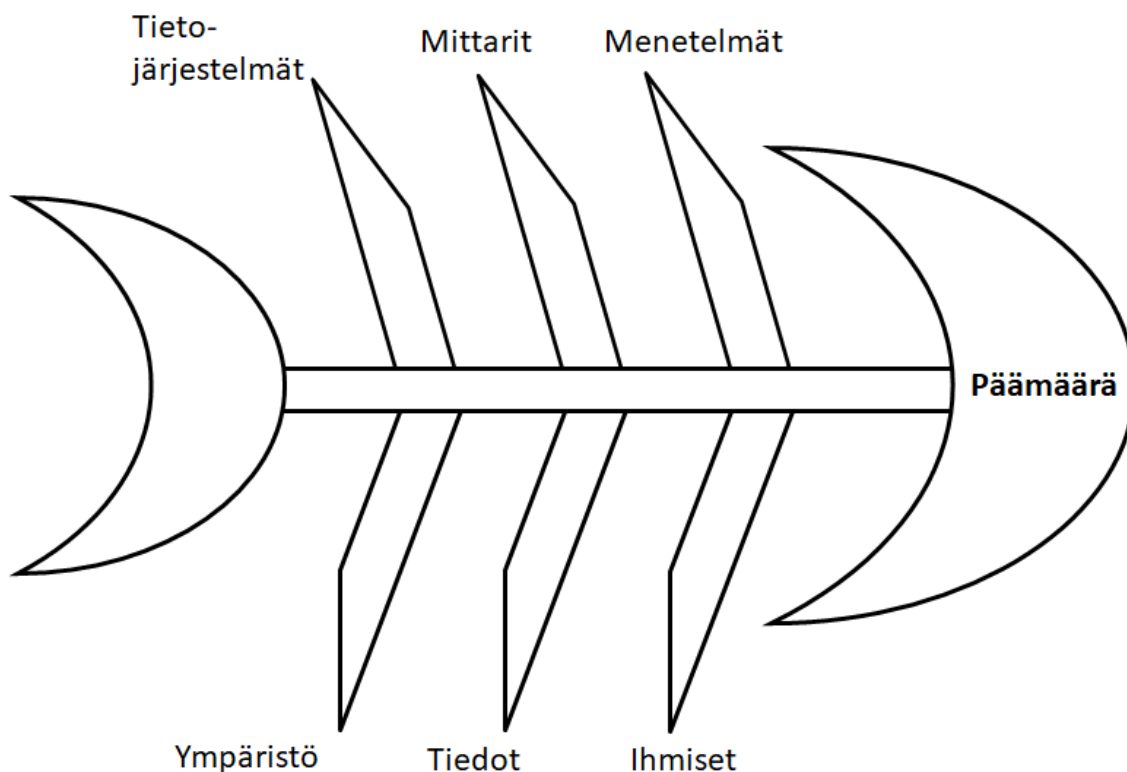
IT-alan näkökulmasta multitasking on yksi suurista ongelmista nykypäivänä, ihmisten töissä jaksamiseen voi vaikuttaa henkilökohtaisella tasolla omalla asenteellaan ja esimerkiksi mindfulness tyyppisten ratkaisujen käytöllä. Myös työnohjauksessa tulisi huomioida mahdollisimman hyvin se, että henkilöstö pystyy keskittymään omaan tekemiseensä ja minimoida häiriötekijät niin hyvin kuin se on mahdollista.

Lean-ajatusmallin mukaisesti keskittyminen kohteeseen ja multitaskingin karsiminen auttaa läpimenoaikoihin sekä tukee FIFO-periaatetta. Ensiksi saapunut työ valmistuu asiakkaalle ensimmäisenä. Tehtäviin keskittyminen auttaa mahdollisesti myös pitämään keskeneräisten töiden jonoa mahdollisimman pienenä käytettävissä olevilla resursseilla.

11.3 Ishikawa- eli kalanruotokaaviot

Leanin mukaisessa systeemiajattelussa systeemi muodostuu osista, joiden tavoite on toimia yhdessä systeemille määriteltyn toimintatapojen mukaan kohti päämäärää. Systeemiajattelun tavoitteena on johtaa systeemiä siten, että kokonaisuus toimii mahdollisimman sujuvasti. Tiivistettynä ajatus on, että systeemin kokonaisuus on enemmän kuin osiensa summa. (Torkkola 2015, 96–97.)

Systeemin ominaisuuksia voidaan mallintaa Ishikawa- eli kalanruotokaaviolla, johon on määriteltä päämäärä ja joissa osina ovat *ihmiset*, *menetelmät*, *mittarit*, *laitteistot*, *ympäristö* sekä *materiaalit*. Asiantuntijatyössä tai palvelutuotannossa materiaalit voidaan korvata tiedolla. Periaatteellinen Ishikawa-kaavio kuvassa 12.

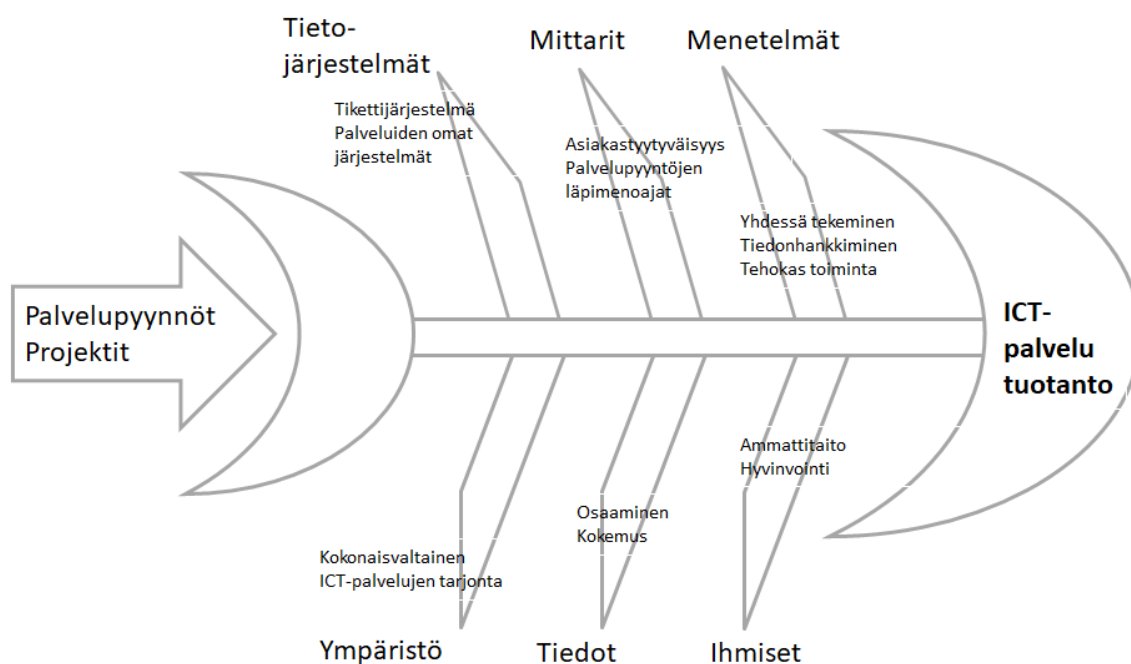


Kuva 12. Ishikawa- eli kalanruotokaavio (Torkkola 2015, 98).

Systeemiajatteluun liittyy myös pullonkaulateoria, pullonkaulateorian mukaan systeemin kokonaisvirtaus on niin suuri kuin sen osat antavat myöten. Mikäli jokin osa kuristaa virtausta enemmän kuin muut, niin se on pullonkaula ja joh-

don pitäisi toimia sen poistamiseksi. Loppujen lopuksi kun pullonkaulan löytää ja korjaa, niin seuraavaksi jostakin toisesta osasta tulee pullonkaula, jota lähde-tään korjaamaan, tästä seuraa siis jatkuvan parantamisen ketju. Ishikawa-kaaviolla hahmotetaan osat, joita seurataan ja pyritään löytämään pullonkaulat. (Torkkola 2015, 99.)

ICT-palvelutuotanto-organisaatiolle tehty Ishikawa-kaavio voisi olla kuvassa 13 esitetyn mukainen. Johto voisi tutkia kaavion perusteella mitä pitäisi ottaa huomioon, että palvelutuotanto olisi mahdollisimman sujuvaa ja mihin osa-alueisiin tulisi kiinnittää erityistä huomiota.



Kuva 13. ICT-palvelutuotannon Ishikawa-kaavio.

Auki kirjoitettuna kuvan 13 kaavio tarkoittaa, että tietojärjestelmistä yleisiä palvelutuotannossa ovat tikettijärjestelmät, johon palvelupyynnöt kirjataan sekä palvelukohtaisesti jokainen järjestelmä. Huomioitavana asiana on, ovatko järjestelmät ja järjestelmien käyttö niin hyvällä tasolla kuin pitäisi olla. Mittareista palvelutuotannossa ovat tärkeitä tehokas toiminta eli toisin sanoen läpimenoajat ja toinen tärkeä mittari on asiakastytyväisyys, ovatko asiakkaat tyytyväisiä toimintaan. Menetelmiä ovat yleensä tiimityöskentely, hyvällä yhteistyöllä saa ratkaistua ongelmia usein paremmin kuin jos jokainen tekisi asioita yksin. Tiedonhank-

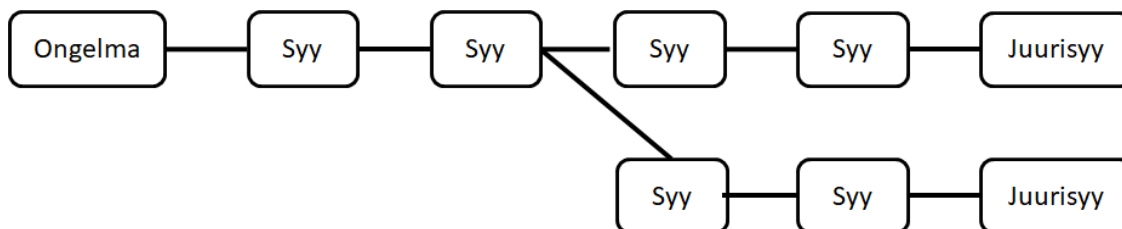
kiminen on myös tärkeässä roolissa, kuten jo mittariinkin suoraan viittaava tehokas toiminta, asiat on pyrittävä ratkaisemaan mahdollisimman nopeasti tai saatava eteenpäin mikäli asiaa ei itse pysty ratkaisemaan.

Ympäristöä on myös hyvä tutkia, mikäli tarkoituksena on ylläpitää kokonaisvaltaista palvelua, niin laitekannan tulee vastata järjestelmien vaatimuksiin. Tietojen osalta osaamista on ylläpidettävä ja kokemuksen kautta hankittua tietoa jaettava. Ihmisten osalta ammattitaito on tärkeää ja henkilöstön hyvinvointi vaikuttaa paljon systeemin toimintakykyyn. Lean-ajatteluun perustaen kaavion kokonaisuuden hahmottaminen auttaa toimintaprosessin sujumista mahdollisimman hyvin.

Ishikawa-kaavio toimii erityisesti johdon muistilistana ja työvälineenä, jota voi säännöllisesti käydä läpi ja päivittää tarpeen vaatiessa. Kaaviolla on helppo esittää tiivistettynä mitä ollaan tekemässä ja mistä asioista kokonaisuus muodostuu. Muutoksen aikaansaaminen vaatii usein myös johdon sitoutumista toiminnan muutokseen.

11.4 Viisi kertaa miksi

Viisi kertaa miksi -ongelmanratkaisussa on tarkoitus löytää ongelman juurisyy esille. Tarkoituksena on paikallistaa ongelma ja etsiä syytä mistä ongelma on lähtöisin, kun syytä ei enää löydy, niin juurisyy on paikallistettu. Ongelma voi haarautua myös useampaan juurisyyhyn. Kun juurisyy on paikallistettu, niin ongelma on mahdollista korjata tätä kautta. Kuvassa 14 on esitetty kaavio miten periaatteellinen viisi kertaa miksi -ongelmanratkaisu voi edetä. (Bicheno & Holweg 2009, 185–186.)



Kuva 14. Viisi kertaa miksi -kaavio.

Taulukkoon 15 on kuvattu viisi kertaa miksi ongelmanratkaisua potentiaaliseen ongelmaan miksi palvelupyyntöjen toimitus kestää pitempään kuin asiakas on toivonut. Ongelmana siis on, että resurssit eivät ehdi tekemään pyyntöjä sovitussa aikataulussa.

Taulukko 15. Viisi kertaa miksi -taulukko.

Ongelma: Resurssit eivät ehdi tekemään pyyntöjä sovitussa aikataulussa

Syylinja 1:

SL1.1 Henkilöstö itseohjautuvaa

SL1.2 Luotto siihen, että kaikki sujuu vähäisellä ohjauksella

SL1.3 Useasta suunnasta tuleva kuormitus aiheuttaa ongelmia

SL1.4 Työnohjausresurssia ei ole saata

Juurisyys 1: Työnohjausresurssia on liian vähän

Syylinja 2:

SL2.1 Palvelupyyntö saapuu epätarkalla tasolla

SL2.2 Palvelupyyntö on otettu edellisessä portaassa vastaan epätarkalla tasolla

SL2.3 Palvelupyynnön tekijä ei ole osannut määritellä palvelupyyntöä

SL2.4 Palvelupyynnön tekoa ja vastaanottoa ei ole ohjeistettu riittävästi

Juurisyys 2: Ohjeistus on puutteellisella tasolla palvelupyynnön tekoon ja vastaanottotarkistuksiin

Syylinja 3:

SL3.1 Tekijäresurssilla on liian paljon tekemistä

SL3.1.1 Työtä ei saa jaettua muille työntekijöille

**Juurisyys 3a: Toiminta-alueen työntekijöitä on liian vähän suhteessa tarvit-
tuun työhön**

SL3.1.2 Työ keskeytyy liian usein

Juurisyys 3b: Keskeytyksiä tulee liian paljon

Juurisyiden löytymisen perusteella ongelmiin voitaisiin lähteä selvittämään potentiaalisia muutosratkaisuja juurisyiden ratkaisemiseksi. Toimenpiteiden ja mitausten jälkeen ongelma voi olla juurisyys voi olla poistunut tai muuttunut ja viisi kertaa miksi -ongelmanratkaisulla voidaan selvittää uudestaan muuttunutta tilannetta. Säännöllisesti läpikäytynä juurisyys selvitys auttaa löytämään kehityskohteita työn sujuvuuden lisäämiseksi.

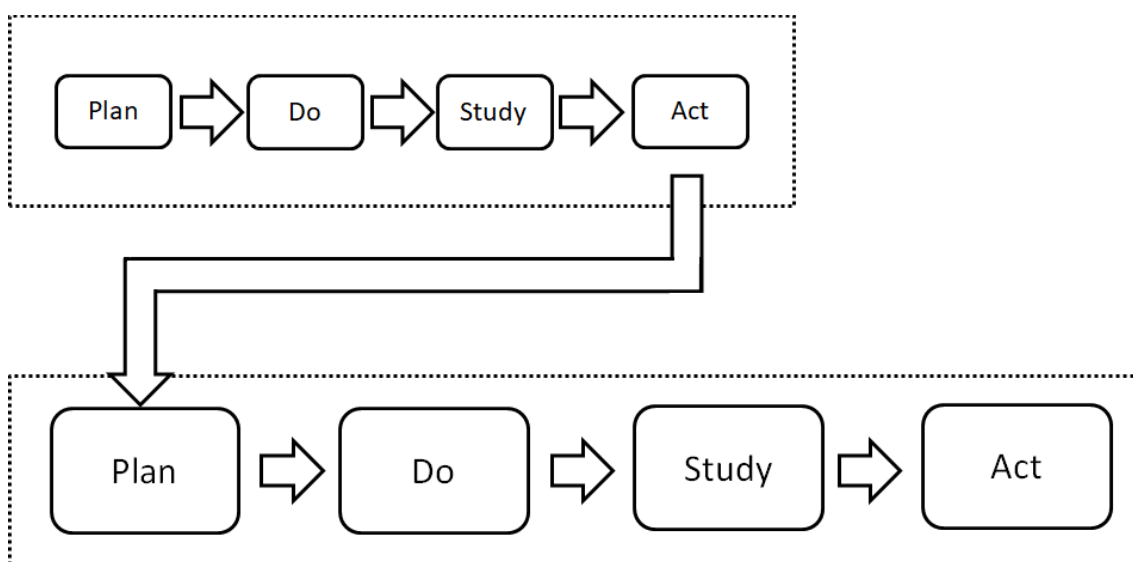
11.5 Obeya / War Room eli tilannehuone

Obeya / War Room eli tilannehuone on toimintatapa, jossa useamman toimialan vastuuhenkilöt kokoontuvat sovituksi ajaksi selvittämään projektin tai tuotannon tilannetta yhteen huoneeseen. Tilanne on nähtävissä huoneessa visuaalisesti ja ongelmat pyritään ratkaisemaan nopeasti tilannehuonetapaamisen aikana. Tilannehuoneessa kaikki kriittinen tieto on saatavissa päätöksenteon tueksi. (Weber 2017.)

Leanin tilannehuone konsepti muistuttaa jonkin verran ketterän kehityksen scrum-tapaamista. Ehkä suurimpana periaatteellisena erona on, että tilannehuoneeseen on tarkoitus kutsua myös muiden kuin toteuttajaryhmän jäseniä. IT-palvelutuotantoon vietyinä tilannehuone voisi toimia päivittäisen työnhjauksen metodina, johon voitaisiin kutsua myös tilaajien edustajia esimerkiksi tarkentamaan tilausta. Nykyisillä etätyöyhteisillä ajattelu fyysisestä huoneesta on syytä ajatella uudestaan, voisiko tilannehuone olla mittaristoja ja palvelutuotannon ohjausjärjestelmiä yhdistettynä ja kokous järjestettäisiin etäyhteisöjen kautta säännöllistä keskustelua ja tiedonvaihtoa varten. Oikein resursoituna tilannehuone voisi vähentää keskeytysten määrää työntekijöiltä jos ajatellaan, että tilannehuoneessa ratkaistaan tai otetaan vastaan tehtäviä ja tilannehuoneen ulkopuolinen aika rauhoitetaan työnteolle.

11.6 Plan-Do-Study-Act (PDSA) eli suunnittele, toteuta, tutki ja päättä

Plan-Do-Study-Act eli PDSA on malli, jolla voidaan kokeilla ja testata toimintaa parantavia muutoksia prosesseihin ja töiden tekemiseen. Ajatuksena on, että muutosta kokeillaan ensin pienemmässä ympäristössä ja päätetään vasta analyysin jälkeen voidaanko muutos laajentaa suurempaan ympäristöön. Laajentaminen suurempaan ympäristöön esimerkiksi tiimi-tasolta yksikkö-tasolle tehdään iteraatiolla eli samat vaiheet käydään uudestaan läpi uudella kierroksella (Torkkola 2015, 39–42.)



Kuva 15. PDSA-malli.

Plan eli suunnitteluvaiheessa määritellään muutoksen tavoite, mittaustapa sekä odotetut tulokset, joiden perusteella voidaan päätellä, toimiko muutos kuten ajateltiin. *Do* eli toteutusvaiheessa pyritään toteuttamaan muutos mahdollisimman pienessä erässä, tällöin kustannukset sekä työpanos pysyvät muutoksen testaamisen kannalta mahdollisimman taloudellisena. *Study* eli tutkintavaiheessa analysoidaan mittaukset, mitä ja miten tapahtui sekä saavutettiin odotetut tulokset. Jos tulosten pohjalta on mahdollista tehdä tilastollista analyysia, niin mahdollisuus kannattaa hyödyntää varmistuakseen, että tulokset ovat toistettavissa mahdollisella seuraavalla kierroksella. *Act* eli päätösvaiheessa päätetään lähdetäänkö muutosta viemään suurempaan ympäristöön, muutetaanko testiä ja ajetaan sykli uudestaan tai lopetetaanko muutoksen läpivienti tähän kierrok-

seen ja jätetään muutos tekemättä suurempaan ympäristöön. (Torkkola 2015, 41–42.)

IT-palvelutuotannossa prosessien kehittämistä voitaisiin testata PDSA-mallilla, esimerkkinä scrumin ja sprinttipyrähdysten käyttöönotto voitaisiin testata PDSA-mallilla. Karkealla ajattelulla muutoksessa voitaisiin mitata yhdellä tiimillä käytettyä aikaa, dokumentaation laadun muutosta sekä henkilöstön tyytyväisyyttä prosessiin. Mikäli muutoksesta saataisiin hyviä kokemuksia, niin muutosta voitaisiin jalkauttaa useammalle tiimille ja viedä sama PDSA-malli läpi isommalla kohteella. Näin toteutettuna PDSA-malli toteuttaisi Leanin jatkuvaa parannusta sekä ketterien kehitysmenetelmien sopivuuden testaamista.

PDSA-mallia voitaisiin raportoida esimerkiksi A3-raportilla, joka on myös yksi Lean-työkalu. A3-raportoinnissa ajatuksena on, että prosessi tai tässä tapauksessa kehitysprosessi pystytään kuvaamaan yhdelle A3-kokoiselle paperille. Kun tilaa on rajoitetusti käytössä, niin asioita on pakko tiivistää ja tietoa saadaan välitettyä raportin avulla tehokkaasti. A3-raportille voidaan esimerkiksi kirjata tavoitteet, selvittää nykytilaa, selvittää mahdolliset kehityskohteet tavoitteiden toteuttamiseksi ja kirjata kehitysehdotukset, joilla tavoitteisiin päästään. (Poppendieck 2006, 157–158.)

11.7 Sprintit eli pyrähdykset

Luvussa 7.4. esiteltyt sprintit eli pyrähdykset voisivat auttaa järjestelmällisemmän työn toteuttamisessa ja keskeytysten hallinnassa, tämä vaikuttaisi omalta osaltaan virtaustehokkuuteen. Palvelutuotantoon, jossa samat henkilöt kehittävät ja ylläpitävät järjestelmiä ja ohjelmistoja, sprintit voitaisiin ottaa myös sovellettuna käyttöön eli pyrittäisiin tekemään kehitystä järjestelmällisesti, mutta kriittiset vikatilanteet ajaisivat ohi työlistalla, jota käytäisiin läpi päivittäisessä Scrum-palaverissa. Järjestelmällisyyttä saataisiin kuitenkin aikaiseksi siten, että kiireettömät työpyynnöt käsiteltäisiin prioriteetin mukaisessa tai FIFO-metodin mukaisessa järjestelmässä. Varsinaisen sprintin pituutta palvelutuotannossa voitaisiin miettiä siten, että jos työn alla ei ole varsinaista suurempaa kokonai-

suutta, jolla olisi määritelty aikaraja, niin sprintin pituuden kannattaa olla vähintään 4 viikkoa. Työtilanteen muuttuessa sprintin pituutta voi ketterästi säätää tarpeen mukaan.

Sprintin tehtävälista voidaan esittää listana ja operatiivinen tehtävälista Kanban-tauluna, jota käsiteltiin luvussa 7.5. Sprintin alussa määritelty lista tuodaan Kanban-tauluun, jonka edistymistä voidaan seurata päivittäisissä Scrum-palaverieissa. Tietojärjestelmäpohjainen Kanban-taulu on myös helppo jakaa koko organisaatiolle tehtävälistan läpinäkyvyyden mahdollistamiseksi.

11.8 Daily Scrum eli Scrum-palaverit

Luvussa 7.4 esiteltyjen päivittäisten Scrum-palaverien käyttöönotto toimisi sprinttipyrähdysten yhteydessä hyvin. Palaverissa käytäisiin läpi päivän tilanne ja toimittaisiin sen mukaan. Scrum Masterin johdolla pidetty päivittäinen palaveri kestäisi mahdollisimman lyhyen aikaa ja tiedon jaon hyöty olisi merkittävä.

Scrum Master-rooliin pystyy kouluttautumaan joko itse tai kurssilla opiskellen. Sertifioidun Scrum Masterin koulutus kestää yleensä 2 työpäivää, koulutuksen jälkeen kouluttautujalla on valmiudet suorittaa sertifiointi sekä aloittaa Scrum-tiimin johtaminen sekä Scrum-palaverien pitäminen. (Scrum Alliance 2019a.)

Myös muille Scrum-tiimin jäsenelle on olemassa sertifiointikoulutuksia, kuten tuoteomistajalle sekä kehittäjille (Scrum Alliance 2019b). Ketterän kehityksen ja Scrum-periaatteiden yleiskoulutuksia on myös tarjolla räätälöitynä, tällaisen koulutuksen järjestäminen organisaatiolle ketterään kehitykseen siirtymisen yhteydessä voisi olla myös hyödyllinen tiedon jaon ja onnistuneen toiminnan kehittämisen kannalta. (Tieturi 2019.)

11.9 Incident Management eli havaintojen hallinta

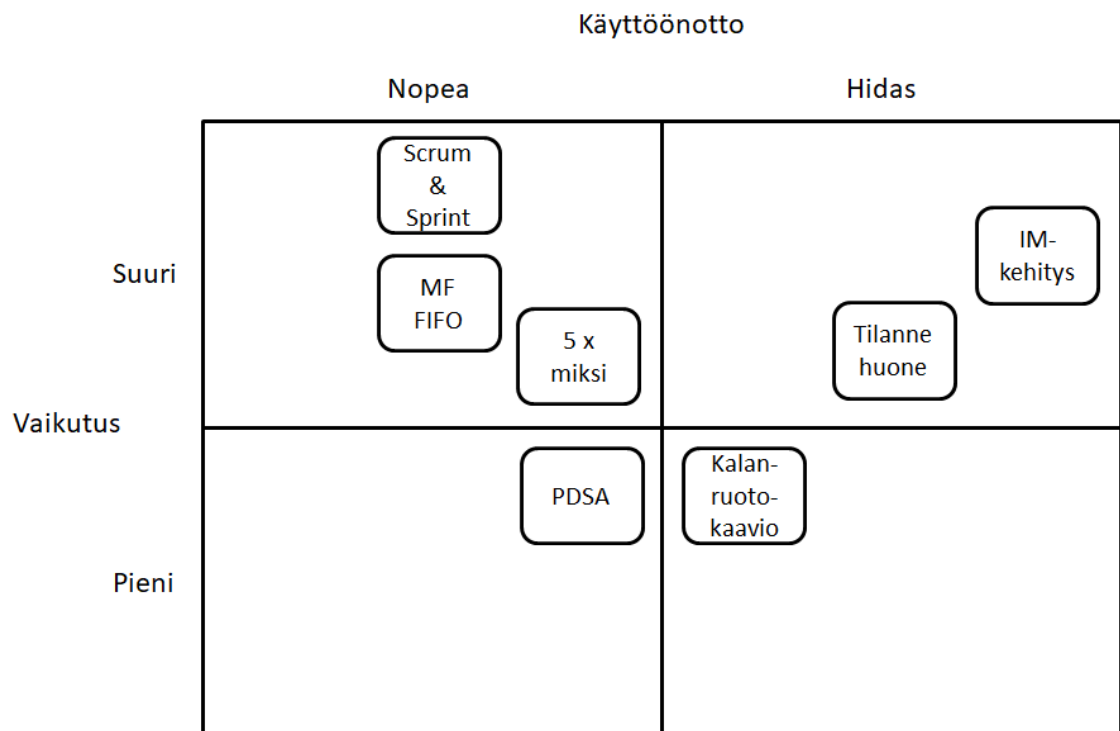
Järjestelmällinen havaintojen hallinta on tärkeä osa IT-palvelutuotantoa. Ketterästi toteutettuna havaintojen hallinta on läpinäkyvää ja asiakas pystyy seuraamaan havaintonsa tilannetta reaaliajassa. Tämän mahdollistavia järjestelmiä on useita, mutta niiden käyttöön tulee suunnitella organisaation ja asiakkaan tarpeita vastaava prosessi. (Winter 2018.)

Gemba-läpikävelystä havaitut kehityskohteet kohdistuvat palvelupyynnöiden hallintaan. Palvelutuotantoon soveltuva havaintojen vastaanottoprosessi voisi toimia palvelupyynnöitä vastaanottaessa yhden pisteen periaatteella eli palvelupyynnön vastaanotto ohjattaisiin aina yhteen kohteeseen eli havaintojen vastaanottoon. Havainnoille voisi kehittää oman lomakkeen vastaanottamiseen, jossa olisi listattuna seuraavia asioita: havainnon tekijä, tekijän yhteystiedot, järjestelmä, kuvaus ongelmasta, oliko ongelma toistettavissa, mahdollinen kuvaus miten järjestelmän olisi pitänyt toimia tilanteessa sekä kriittisyysaste. Kriittisyysasteen jälkeen havainnolle arvioitaisiin prioriteetti palvelutuotannon sisällä. Mikäli ongelma on työt pysäyttävä, niin se on otettava korjauksen alle mahdollisimman pian ja mikäli ongelma ei pysäytä järjestelmän käyttöä tai ongelma on kierrettävissä toisella toiminnalla, niin prioriteetti ei ole niin korkea.

Joissakin palvelutuotannon organisaatioissa on määritelty resurssi Incident Manager, joka on joskus suomennettu häiriöhallinnasta vastaavaksi henkilöksi, jonka päätehtävä on havaintojen vastaanottamisen ohella priorisoida sekä ohjata niitä toteutuksen kannalta oikeille henkilöille. Resurssin tarkoitus on tehostaa prosessin kulkua ja havaintojen hallinnan virtaustehokkuutta, omalta osaltaan havaintojen hallinnan resurssi myös vähentää keskeytyskuormitusta ottamalla vastaan ja esivalmistelemalla havaintoa. Organisaation toiminnan ja henkilöstön tuntemus on eduksi havaintojen hallinnassa ja kokeneella resurssilla onkin edellytykset huomattavaan toiminnan parantamiseen. NykYTEknologian näkökulmasta AI-pohjainen Incident Management on jo joissain organisaatioissa käytössä ja tekoäly tulee todennäköisesti lisääntymään tällä sektorilla.

11.10 Toiminnan kehityslista nelikenttäanalyysinä

Kehityslista sellaisenaan ei välttämättä luo konkretiaa, joten kehitysehdotukse-
na toteutettiin nelikenttäanalyysi, jossa on mietitty kehitysajatusten käyttöönoton
mahdollisuuksia käyttöönoton nopeuden ja vaikutuksen näkökulmasta. Ajatuk-
sena on, että kehitystä voisi lähteä rakentamaan siitä näkökulmasta, millä uu-
distuksilla olisi vaikutusta nykyiseen toimintaan ja jotka olisi toteutettavissa no-
peasti. Kuvassa 16 on esitetty nelikenttäanalyysi.



Kuva 16. Nelikenttäanalyysi kehitystoimista.

Analyysiä avaamalla Scrum & Sprint sijoitettuna yhdeksi kehitystoimeksi nousi-
vat arvion mukaan nopeasti käyttöönotettaviksi ja vaikutukseltaan suureksi. Ket-
terät kehitystavat tukisivat mindfulness-ajattelua ja FIFO-periaatetta, joten nämä
yhdessä toisivat suuren uudistuksen toimintaan nopeasti. Lisäksi mukaan otettu
5 x miksi juurisyiden selvitys on nopeaa ja useimmiten tehokasta toimintaa. Juu-
risyys selvitysten vaikuttavuus todennäköisesti pienenee ajan myötä, mutta alus-
sa vaikutukset voivat olla suuriakin. Myös PDSA-toiminnan aloittaminen on no-
peaa, mutta vaikuttavuus ei välttämättä ole niin suurta kuin ketterän kehityksen
käyttöönotolla.

Hitaammin käyttöönotettavien puolella on vaikutusarvioltaan merkityksellisiä kohteita tilanhuoneen käyttöönotto ja havaintojen hallinnan kehittäminen. Pitkällä aikavälillä näillä voi olla suuri vaikutus toimintaan, mutta ne vaativat huolellista suunnittelua käyttöönottamiseksi ja mahdollisesti myös suunnitelmallista toiminnanmuutosta. Kalanruotokaavioiden käyttöönotto on hitaan ja nopean välillä, periaatteellinen kaavio on tehty, mutta ajatusmallin sisällyttäminen ja kalanruotokaavion tarkentaminen vaatinee pidemmän aikavälin. Kalanruotokaavion vaikuttavuus voi aluksi olla vähäistä, mutta ajan myötä vaikuttavuus voi mahdollisesti nousta suuremmaksi. Enemmän taustalla kuin toiminnassa vaikuttava kaaviomalli voi lopulta ollakin toiminnan ohjauksen kannalta tärkeä.

Kehityslista nopean ja näkyvän toiminnan kehittämisen kannalta suunnata nopeisiin ja vaikutukseltaan suuriksi arvioituihin kohteisiin ja laajentaa siitä muihin toimenpiteisiin. Alkuun pääsyn myötä ja Leaniin sekä ketteriin kehitysmalleihin tutustumalla enemmän lista voi myös elää niin sisällöllisesti kuin vaikuttavuusanalyysin osalta.

12 Pohdinta

Opinnäytetyön tavoitteena oli tutustua Lean-ideologiaan sekä ketterän kehityksen teoriaan, kartoittaa Pohjois-Karjalan tietotekniikkakeskus Oy:n tietojärjestelmäsuunnittelun nykytilaa mittausten ja analyysien perusteella, sekä tuottaa kehitysohjelmaa, jonka avulla pystyttäisiin kehittämään toimintaa tietojärjestelmien suunnittelutyön palvelutuotannossa. Opinnäytetyö oli rajattu tietojärjestelmäsuunnitteluun kehityskohteen ja metodien hallinnan kurissa pitämiseksi. Aikataulua opinnäytetyön tekemiseen ei määritelty ja toteutus kesti noin vuoden verran kevästä 2018 kevääseen 2019.

Tavoitteet saavutettiin suunnitellulla tavalla, hyviä teorioita ja käytäntöjä on kerätty opinnäytetyöhön kehityksen ja ajatusten tueksi sekä suunnitelma kehityskohteista ja työkaluista on muodostettu. Mittaukset olivat relevantteja ja niiden analyysien pohjalta saatiin tärkeitä tietoja kehityskohteiden muodostamiseksi.

Teoriaosuuden lähdeaineistojen tietoja yhdistelemällä pystyttiin poimimaan kattavasti käytäntöön sopivia malleja ja työkaluja toiminnan kehittämisen tueksi. Laajasta teoria-aineistosta on poimittava sopivat lähteet, osa teoriasta sopi kehityskohteen toimintaan paremmin kuin toiset.

Parannettaviakin asioita löytyi, opinnäytetyön aikana tutkittiin Lean-kirjallisuutta lähtökohdasta, jossa tekijällä ei ollut syvällisempää aikaisempaa kokemusta Lean-periaatteista, alkuinnostuksen jälkeen mittauskohteiden valinta toi erityisiä haasteita. Opinnäytetyöprosessin aikana suoritettu Lean Green Belt -kurssi auttoi kuitenkin osaltaan monessa asiassa ja aivan erityisesti mittauskohteiden valinnassa.

Mittausten kattavuus muodostui myös ongelmaksi, vaikka mittauskohteina oli 11 henkilöä, niin vapaaehtoisuuteen perustuva mittaus ei tuottanut tuloksia kuin yhden henkilön kohdalta. Kriittisesti ajatellen tämä vaikuttaa tutkimuksen luotettavuuteen, tutkimuksen tulokset ovat niin luotettavat, kuin siitä lähtöaineiston perusteella on ollut mahdollista tehdä. Uusintamittauksissa tutkimuksen luotettavuuden varmistamiseksi tulisi käyttää erityistä huomiota siihen, että otoskoko saataisiin riittävän kattavaksi.

Positiivinen onnistuminen oli Lean-periaatteiden ja ohjelmistotuotannossa laajalle levinneen ketterän kehityksen samankaltaisuuksien löytäminen. Toisiaan täydentämällä periaatteet ovat hyvä lähtökohta toiminnan kehittämiseksi IT-alan palvelutuotannossa. Useampien yhtiöiden pitäisi tutkia oman sektorinsa ulkopuolelta ajatusmalleja, ohjelmistoyhtiöissä ei välttämättä ole ajateltu Leania ohjelmistotuotannon metodina ja palvelutuotantoyhtiöissä ei välttämättä ajatella ketterän kehityksen auttavan toimintaa millään tavoin. Tutkimuksen perusteella näin ei pitäisi olla, vaan teoriapohjasta kannattaisi poimia omaan toimintaan sopivat ajatusmallit ja työkalut ja aloittaa toiminnan kehittäminen uudesta näkökulmasta.

13 Epilogi

Informaatiotekniikan ala on elänyt aina suuressa muutoksessa, niin myös opinnäytetyöni toteuttamisen aikana. Pohjois-Karjalan tietotekniikkakeskus luovutti liiketoimintansa 1.1.2019 uudelle yhtiölle Meidän IT ja talous Oy:lle, johon liittyivät samanaikaisesti myös Siun talous Oy sekä Saimaan talous ja tieto Oy. (Meidän IT ja talous 2019.) Toimintatavat tulevat todennäköisesti muuttumaan jo yhdistymisen johdosta, kun useamman yhtiön yrityskulttuurit kohtaavat.

Opinnäytetyössäni esittelemät mittaukset ja analyysien kautta saadut kehitysjatukset ovat kuitenkin valideja ja mittausten kannalta ainakin osittain toistettavissa uudestaan myös uudessa yhtiössä. Toivonkin, että opinnäytetyöstä ja siinä esitetyistä kehitysjatuksista olisi apua IT-palvelutuotannon kehittämisessä uudessa yhtiössä sekä julkisena opinnäytetyönä myös muille IT-palveluja tuottaville yhtiöille toiminnan kehittämiseksi. Pienillä muutoksilla voi saada aikaiseksi suuria parannuksia toimintaan.

Lähteet

- Agile Alliance. 2018. What Agile Alliance Is and What it Does. <https://www.agilealliance.org/the-alliance/>. 20.10.2018.
- Agile Manifesto. 2001. Principles behind the Agile Manifesto. <http://agilemanifesto.org/principles.html>. 20.10.2018.
- Bicheno, J., & Holweg, M. 2009. The Lean Toolbox 4th Edition: The Essential Guide to Lean Transformation.
- European Commission. 2015. eGovernment Action Plan 2016-2020. http://ec.europa.eu/smart-regulation/roadmaps/docs/2016_cnect_006_e_government_action_plan_en.pdf 8.9.2018.
- Haikala, I., & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Helsinki: Talentum.
- Hougaard, R., & Kalajo, T. 2016. Tilaa ajatella: Tehosta työtäsi mindfulnessin avulla. Jyväskylä: PS-kustannus.
- ITIL Roles. 2011. ITIL roles and boards - Service Operation. https://wiki.en.it-processmaps.com/index.php/ITIL_Roles#ITIL_roles_and_boards_-_Service_Operation 9.9.2018.
- Meidän IT ja talous Oy. 2019. Meidän IT ja talous Oy on syntynyt 1.1.2019. <http://www.meita.fi/> 3.2.2019.
- Modig, N., & Åhlström, P. 2013. Tätä on lean: ratkaisu tehokkuusparadoksiin. Rheologica publishing.
- Office.com. 2019. Pearson-funktio. <https://support.office.com/fi-fi/article/pearson-pearson-funktio-0c3e30fc-e5af-49c4-808a-3ef66e034c18> 2.2.2019.
- Pohjois-Karjalan tietotekniikkakeskus. 2017. PTTK Oy:n esite. http://www.esitteemme.fi/Pohjois-Karjalan_Tietotekniikkakeskus_Oy/WebView/ 7.10.2018.
- Poppendieck, M., & Poppendieck, T. 2006. Implementing Lean Software Development: From Concept To Cash. Addison-Wesley.
- Poppendieck, M., & Poppendieck, T. 2007. Lean Software Development: An Agile Toolkit: An Agile Toolkit. Addison-Wesley.
- Reaktori Oy & Helsingin yliopisto. 2018. Elements of AI. <https://www.elementsofai.com/> 7.10.2018.
- Royce, W. 1970. Managing the Development of Large Software Systems <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf> 8.9.2018.
- Scrum Alliance. 2019a. Certified Scrum Master. <https://www.scrumalliance.org/get-certified/scrums-master-track/certified-scrummaster>. 3.3.2019.
- Scrum Alliance. 2019b. Certifications by Scrum Team Role. <https://www.scrumalliance.org/get-certified>. 3.3.2019.
- Tampereen yliopisto. 2004. Korrelaatio ja riippuvuusluvut. <https://www.fsd.uta.fi/menetelmaopetus/korrelaatio/korrelaatio.html> 2.2.2019.
- Tieturi. 2019. Agile-, Lean- ja Scrum-koulutukset. <https://www.tieturi.fi/koulutukset/agile-lean-scrum> 3.3.2019.
- Torkkola, S. 2015. Lean asiantuntijatyön johtamisessa. Helsinki: Alma Talent.

- Weber, A. 2017. Make Room for Obeya.
<https://www.assemblymag.com/articles/93826-make-room-for-obeya>.
24.2.2019.
- Winter, S. 2018. Agile for when things go wrong: the missing piece of your incident response plan. <https://www.atlassian.com/agile/software-development/incident-response>. 3.3.2019.