

Mikko Kujanpää

Verohallinnon Valmis-hankkeen projektimalli ja sen analysointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

1.5.2016

Tekijä Otsikko Sivumäärä Aika	Mikko Kujanpää Verohallinnon Valmis-hankkeen projektimalli ja sen analysointi 40 sivua 1.5.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Yliopettaja Auvo Häkkinen Palvelupäällikkö Mika Kirveennummi
<p>Opinnäyteyön tavoitteena on esitellä ja analysoida Verohallinnon Valmis-hankkeen projektimallia sekä vertailla tätä yleisiin projektimalleihin. Työssä esitellään vertailussa käytettäviä projektimalleja ja projektinhallinnan menetelmiä.</p> <p>Vertailussa käytettäviä projektimalleja ovat perinteinen vesiputousmalli, Scrum ja RUP. Näiden lisäksi käydään läpi Lean-ajattelun perusidea, Gantt-kaavio ja yleisesti projektien riskienhallinnasta.</p> <p>Analysoinnissa keskitytään lähinnä Valmis-hankkeen aikatauluun, organisaatorakenteeseen, dokumentointi- ja tiedotuskäytäntöihin sekä riskienhallintaan.</p> <p>Analysoinnin ja vertailun pohjalta todettiin, että Valmis-hankkeessa on paljon piirteitä yleisistä projektimalleista. Yhtäläisyyksiä yleisiin projektimalleihin ja menetelmiin löydettiin. Myös useita ongelmakohtia ja mahdollisia kehityskohteita havaittiin analysoinnin yhteydessä. Hanke on kooltaan erittäin suuri, ja projektinhallinta on hankkeessa haastavaa. Paljon kehitettävää löytyy vielä, mutta jatkuvasti ollaan matkalla kohti toimivampaa kokonaisuutta.</p>	
Avainsanat	Projektinhallinta, projektimalli

Author Title	Mikko Kujanpää Finnish Tax Administration's Project Model and Its Analysis
Number of Pages Date	40 pages 1.5.2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Auvo Häkkinen, Principal Lecturer Mika Kirveenummi, Service Manager
<p>The objective of this thesis is to introduce and analyse the project model of Valmis programme and compare it to common project models. The thesis presents the project models and project management methods used in the comparison, i.e. waterfall model, Scrum and RUP. In addition, the basic ideas of Lean thinking, Gantt chart and general project risk management are reviewed.</p> <p>The analysis focuses mainly on the schedule, organizational structure, documentation and information methods and risk management of the Valmis programme.</p> <p>On the basis of the analysis and comparison it was stated that the Valmis programme shares a lot of features with common project models. Resemblances to common project models and methods were found. Also several problem and possible development areas were noticed during the analysis. The programme is very large and project management is challenging. There is still a lot of room for improvement but the journey towards a more functional entity is ongoing.</p>	
Keywords	Project management, project model

Sisällys

1	Johdanto	1
2	Projektit ja niiden hallinta	2
2.1	Projektinhallinta ja projektimallit	3
2.1.1	Vesiputousmalli	3
2.1.2	Scrum-malli	5
2.1.3	RUP	7
2.2	Lean-ajattelu	8
2.3	Gantt-kaavio	9
2.4	Ohjelmistoprojektit	10
2.4.1	Projektin osittaminen	10
2.4.2	Projektin organisaatio	12
2.4.3	Tyypillisimmät ongelmat projektityöskentelyssä	13
2.4.4	Riskien hallinta	13
3	Projektityöskentely Valmis-hankkeessa	16
3.1	Aikataulu ja projektin vaiheet	19
3.2	Organisaation rakenne, tehtävät ja kommunikointi	22
3.3	Palaveri- ja tiedotuskäytännöt sekä dokumentointi	26
3.4	Riskien hallinta	29
4	Johtopäätökset	33
	Lähteet	34

1 Johdanto

Valmis-hanke on Verohallinnon historian suurin yksittäinen muutoshanke, joka sisältää tietoteknisten muutosten lisäksi myös merkittäviä toiminnan muutoksia. Muutoshankkeen tarkoituksena on varmistaa, että verotuksen oikeellisuus, oikea-aikaisuus ja toimintavarmuus ovat turvattuja myös tulevaisuudessa. Hankkeen tarkoituksena on korvata tällä hetkellä käytössä olevat lukuisat ohjelmistot yhdellä verotuksen valmisohjelmalla. Uudistukselle koettiin tarvetta, koska nykyistä eri tietojärjestelmien muodostavaa kokonaisuutta on vaikea kehittää sen hajanaisuuden vuoksi, ja monet ohjelmista ovat ajaneet jo aikansa ohi.

Valmis-hankkeen toimittajina toimivat Fast Enterprises LLC ja Nortal Oy. Fast Enterprises on yhdysvaltalainen teknologiayritys, joka on erikoistunut valtion virastojen ohjelmistojen kehitykseen ja asennukseen. Verohallinnon hankkiman verotuksen valmisohjelmisto, GenTax, on Fast Enterprisesin kehittämä.

GenTaxin konfiguroinnista vastaa Fast Enterprises yhdessä Nortal Oy:n kanssa. Nortal Oy on kansainvälinen liiketoimintasovellusten ja sähköisten palveluratkaisujen toimittaja, joka palvelee sekä julkisen että yksityisen sektorin asiakkaita.

Hanke kokonaisuudessaan sisältää monta eri projektia. Varsinaisten toteutusprojektien lisäksi hanke on jaettu valmisohjelmiston hankinta-projektiin, valmisteluprojekteihin, kokonaisprojektsuunnitteluun ja tietohallinnon palvelumallin suunnitteluun ja käyttöönottoon. Tämän työn tavoitteena on pääasiallisesti keskittyä Valmis-hankkeen toteutusprojekteihin, eli varsinaiseen kehitystyöhön, ja esitellä sekä analysoida hankkeen projektityöskentelyä ja projektimallia. Samalla vertaillaan niitä perinteisten projektimallien sekä ketterien menetelmien käytäntöihin.

Vertailukohteena toimitusprojektimallille käytetään perinteistä vesiputousmallia sekä ketteristä menetelmistä Scrum-mallia ja RUP-kehystä. Työn ei ole tarkoitus olla kattava ohjeistus eri projektimalleille. Tarkoituksena on lyhyesti esitellä verrattavat projektimallit ja keskittyä pääasiassa hankkeeseen ja sen yhtäläisyyksiin ja eroavaisuuksiin muihin projektimalleihin nähden.

2 Projektit ja niiden hallinta

Projekteja syntyy monesta tarpeesta, ja ne voivat olla hyvin monimuotoisia. Projekti voi olla esimerkiksi tutkimusta, muutosta, investointia tai kehitystä. Tuskin on olemassa mitään liiketoimintaa, jonka elinkaaren aikana ei olisi minkäänlaista projektitoimintaa. Projektille on sen muodosta huolimatta kuitenkin aina tyypillistä, että sille on määritelty etukäteen aika- ja kustannustavoite sekä lopputavoitteet. (Projekti-instituutti 2016.)

Projektin päätyttyä sille asetettuja aikataulua, budjettia, resursseja sekä tavoitteita tarkastellaan ja todetaan projektin onnistuminen. Projektin onnistuessa ”pyhän kolminaisuuden” (kuva 1) kaikki elementit on saavutettu projektille asetetuissa puitteissa.



Kuva 1. Projektikolmio ja projektin laatu. (Vahtolammi 2010.)

Kolmio esittää kustannusten ja tavoitteiden välistä suhdetta. Tavoitteena on pitää kolmio tasasivuisena. Mikäli joku projektin osista suurenee, samalla myös muut osat joutuvat kauemmaksi keskustasta eli laadusta. Mitä isommat tavoitteet projektilla on, sitä laajemmaksi projekti muuttuu. Projektin kustannukset kasvavat, mikäli aikaa kasvataan. Kustannuksia tietoisesti lisäämällä on kuitenkin mahdollista kutistaa projektin aikataulua. Työtuntimäärää ei kuitenkaan voida merkittävästi vähentää kustannuksia lisäämällä. (Litke & Kunow 2004, 16.) Esimerkiksi lisätyövoiman hankinnalla voidaan vähentää projektiin käytettävää aikaa, koska voidaan tehdä enemmän asioita yhtä aikaa. Projektiin käytettävien henkilötyötuntien määrä säilyy kuitenkin samalla tasolla; työt on vain jaettu useammalle taholle.

Projektin tehokas, tuloksellinen ja onnistunut päätökseen saattaminen on erityisen tärkeää yritysmaailmassa. Projektityöskentely on ennen kaikkea tiimityötä, jossa osaaminen eri aloilta ja erilaiset ajatusmallit yhdistyvät. Vaikka projektin kokonaistavoite olisi selvä, tapa ratkaista tai lähestyä ongelmaa ei kuitenkaan aina ole niin yksiselitteinen. Projekteihin liittyy usein suuria investointeja ja on yleistä, että budjetin rajoissa ei aina pysytä. Epäonnistuneet projektit merkitsevät siis yleensä taloudellisia menetyksiä. Projektinhallinta on syntynyt vastauksena näihin ongelmiin. (Litke & Kunow 2004, 10-12.)

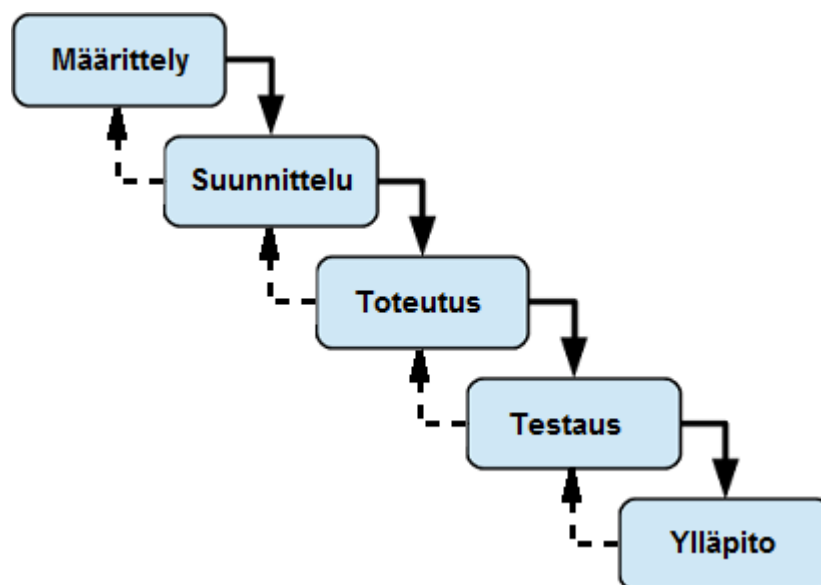
2.1 Projektinhallinta ja projektimallit

Projektinhallinnalla tarkoitetaan toimintoja, joilla projektille asetetut tavoitteet saavutetaan projektille asetettuja rajoja rikkomatta. Projektinhallinnan pääasialliset työkalut ovat projektinhallintamallit. Mallien avulla saadaan visuaalinen kokonaiskuva, joka auttaa yleensä projektiryhmän henkilöitä sisäistämään monimutkaisia kokonaisuuksia. Mikäli yhteistä mallia ei ole, voi se johtaa virheelliseen kuvaan todellisuudesta. (Forsberg, Mooz & Cotterman 2003, 15-19.)

Erilaisissa projektimalleissa on erilaisia työvaiheita ja niissä eteneminen tapahtuu eri tavoin. Tyypillisimpiä työvaiheita ovat kuitenkin projektin aloitus, suunnittelu, toteutus ja lopetus. Mallista riippuen aiempaan vaiheeseen voidaan palata milloin vain tai palaa-minen vain tietyissä vaiheissa.

2.1.1 Vesiputousmalli

Vesiputousmalli (kuva 2) on yleisesti käytetty projektimalli, ja se on alun perin kehitetty ohjelmistoprojekteja varten.



Kuva 2. Perinteinen vesiputousmalli. (Kasurinen 2015.)

Vesiputousmallissa yksi vaihe tulee toteuttaa täysin valmiiksi ennen seuraavan vaiheen aloittamista. Vesiputousmallissa korostuu huolellinen suunnittelu ohjelmiston elinkaaren alussa, joka johtaa merkittäviin säästöihin projektin myöhemmissä vaiheissa. Huonona puolena mallissa on, että edelliseen vaiheeseen palaaminen on yleensä todella haastavaa. Monesti, ainakin ohjelmistoprojekteissa, asiakkaan on vaikea esittää vaatimuksiaan ennen kuin on päästy testaamaan ohjelmiston toimintaa edes jonkinlaisen prototyypin avulla. Asiakas haluaa usein muuttaa vaatimuksiaan kesken projektin. Vesiputousmallissa tämä tarkoittaa, että suuri osa suunnitteluun käytetystä ajasta on ollut turhaa. (Wikipedia 2016.)

Peruseriaatteeltaan malli on hyvin yksinkertainen, mutta mallista löytyy useita variaatioita. Ohjelmistoprojekteissa työskentelevät henkilöt pitävät mallia huonosti soveltuvana suuriin ohjelmistoprojekteihin. Osan ohjelmistoprojekteista, joissa mallia on käytetty, on todettu myöhästyneen mallin joustamattomuuden vuoksi. Jos taas projekti on valmistunut aikataulussa, lopputulos ei ole vastannut käyttäjien toiveita. (Malmi & Åkerlund 2013, 13.)

Vesiputousmallin mukaisissa projekteissa tarvittavat roolit riippuvat projektin laajuudesta ja vaativuudesta. Yleensä mukana ovat ainakin määrittelijä, suunnittelija, ohjelmoija, testaaja ja projektipäällikkö. Projektipäällikköä lukuun ottamatta kussakin roolissa voi

työskennellä useampia henkilöitä. Yksi henkilö voi myös joissain tapauksissa kuulua useaan rooliin. (Sininen Meteoriitti 2013.)

Dokumentaatio on vesiputousmallissa yleensä kattava, koska eri vaiheissa saattavat työskennellä eri henkilöt. Dokumentaatio jäädytetään jokaisessa vaiheessa eikä sitä saa muuttaa ilman johtoryhmän lupaa. Kaikki vaatimukset on kerättävä kerralla, sillä niiden muuttuminen aiheuttaisi muutoksia dokumentteihin. (Taina 2009, 133.)

2.1.2 Scrum-malli

Scrumissa hyödynnetään iteraatiivis-inkrementaalista eli toistavaa ja lisäävää lähestymistapaa. Se mahdollistaa projektin vaiheistamisen ja jatkuvan kontrollin projektin etenemisestä. Tarkoituksena on hiljalleen kehittää ohjelmisto määrittelynsä mukaiseksi. Scrum-malli (kuva 3) on alun perin kehitetty erityisesti ketterien ohjelmistoprojektien hallintaan, mutta sitä voidaan soveltaa myös muissa projektoitavissa töissä. (Scrum Alliance 2016.)



Kuva 3. Scrum-malli. (suomennettu Scrum Alliance 2016.)

Scrumin kolme tärkeintä tukijalkaa ovat läpinäkyvyys, tarkastelu ja sopeuttaminen. Läpinäkyvyys korostuu niin, että projektin kannalta oleelliset asiat, kuten määritelmät ja tavoitteet, on sovittu yhdessä kaikkien projektiin osallistuvien kanssa. Pyritään siihen, että kaikilla on yhteinen näkemys projektista. Projektijohto on vastuussa projektin tar-

kastelusta ja mahdollisesta sopeuttamisesta. Tarkasteluun kuuluu säännöllisin väliajoin suoritettava työn edistymisen tarkkailu, jotta mahdolliset poikkeamat havaittaisiin. Sopeuttamisella tarkoitetaan, että prosessia ja käytettäviä materiaaleja säädetään silloin, kun tarkastelija havaitsee jonkun osan olevan ennalta määritettyjen rajojen ulkopuolella. Sopeuttamista tarvitaan esimerkiksi uusien toiminnallisuuksien kanssa. (Schwaber & Sutherland 2013.)

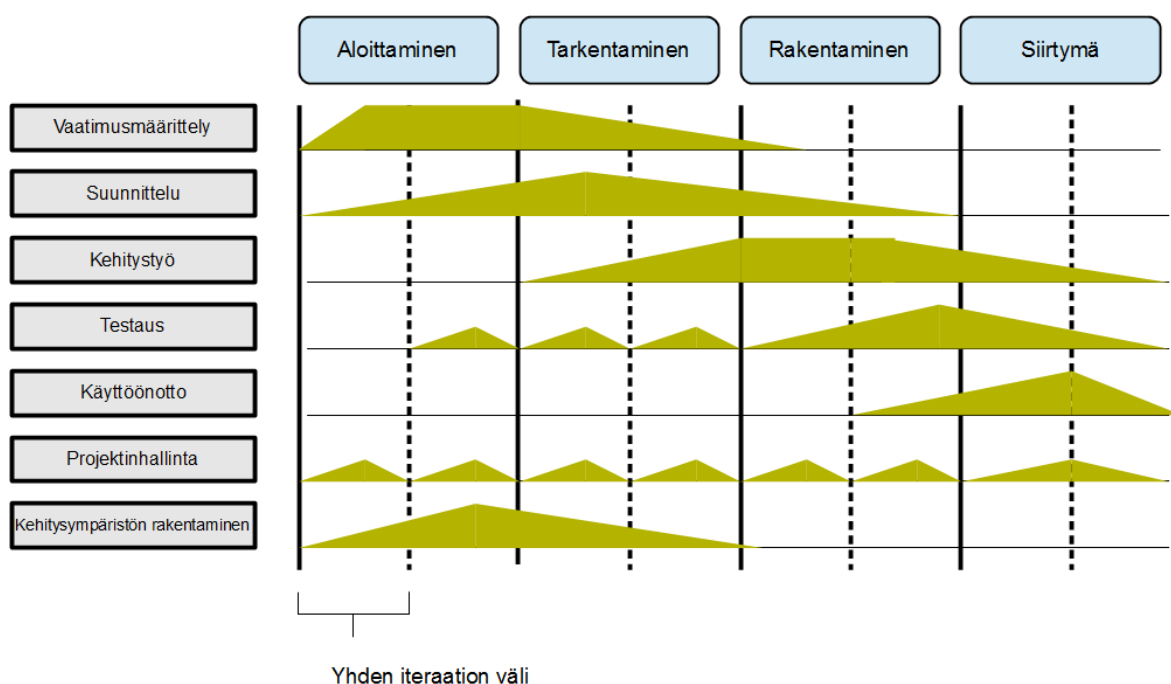
Scrumin ydin on 1-8 viikkoa kestävä kehitysvaiheet, joita kutsutaan sprinteiksi. Sprintin päätyttyä tulisi lopputuloksena olla toimiva tuote, joka sisältää tietyt ennalta määrätyt ominaisuudet. Sprintti aloitetaan aina suunnittelukokouksella ja sprintin aikana pidetään päivittäisiä Scrum-palavereita. Näissä jokapäiväisissä palavereissa tiimi kokoontuu lyhyeen tilannekatsaukseen, jossa jokainen jäsen kertoo mitä on päivän aikana tehnyt, mitä aikoo tehdä seuraavana päivänä ja mitkä tekijät estävät tai hidastavat saavuttamasta sprintin tavoitteita. (Sininen Meteoriiitti 2013.)

Scrumin tiimit ovat monesti itseohjautuvia ja monitaitoisia. Tiimin ovat harvoin riippuvaisia ulkopuolisista henkilöistä, sillä monitaitoisella tiimillä on kaikki työn tekemiseen vaadittava osaaminen. Tiimimalli on suunniteltu joustavuuden, luovuuden ja tuottavuuden optimoimiseksi. Scrum-tiimi koostuu tuoteomistajasta, kehitystiimistä ja scrummasterista. Tuoteomistajan vastuulla on tuotteen arvon ja kehitystiimin työn arvon maksimoiminen. Kehitystiimi koostuu osajista, joiden tarkoituksena on muuttaa tuotteen kehitysjonon sisältö julkaisukelpoiseksi tuoteversioksi jokaisen sprintin aikana. Scrummaster pitää huolen, että kaikki ymmärtävät ja käyttävät Scrumia. Scrummaster tukee kehitystiimiä ja varmistaa, että tiimin työskentely on tuottavaa. Scrummaster toimii myös tuoteomistajan apuna esimerkiksi ehdottamalla tekniikoita tuotteen kehitysjonon hallintaan tai auttamalla Scrum-tiimiä ymmärtämään tuotejonon tiettyjä kohtia. (Schwaber & Sutherland 2013.)

Dokumentointi on Scrumissa erittäin avointa. On tärkeää, että kaikki projektiin liittyvät tiedot ovat näkyviä kaikille osallisille. Projektin ja sprintin työlistat ovat julkisia kaikille projektissa työskenteleville ja jokaisen sprintin lopussa tulokset esitellään julkisesti projektihenkilöstölle. (Sininen Meteoriiitti 2013.)

2.1.3 RUP

RUP eli Rational Unified Process (suomeksi yhtenäistetty prosessi) juontaa juurensa UP:sta (Unified Process), joka on iteratiivisen ohjelmistokehityksen prosessikehitys. RUP on tarkoitettu räätälöitäväksi tilanteen mukaan, sillä se on sisällöltään hyvin laaja. Se kattaa useita modernin ohjelmistokehityksen parhaita käytäntöjä. RUP:n mukaan ohjelmistokehitykseen kuuluu neljä päävaihetta, aloittaminen, tarkentaminen, rakentaminen ja siirtymä (kuva 4). (Kruchten 2004, 18.)



Kuva 4. RUP:n vaiheistus.

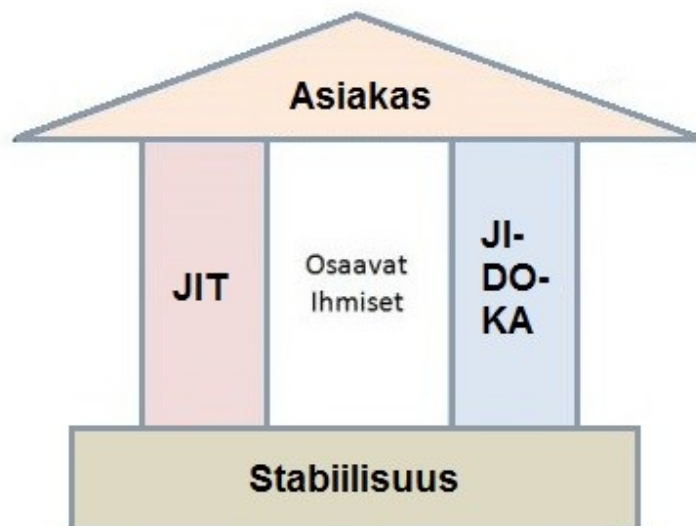
Jokainen päävaihe koostuu yhdestä tai useammasta iteraatiosta. Yhdessä iteraatiossa toteutuu kaikki vesiputousmallin vaiheet, mutta työvaiheiden painotus vaihtelee iteraatiosta riippuen. (Haikala & Märijärvi 2004, 45-47.)

RUP:ssa ei ole varsinaista organisaatiomallia, vaan yksittäisten työntekijöiden ja tiimien käyttäytymismallin ja vastuut määrittää rooli. Rooli on RUP:n yksi keskeisimpiä käsitteitä. Rooleja ei käsitellä yksittäisinä henkilöinä eikä myöskään työnimikkeinä. Roolit tulisi jakaa työntekijöiden ominaisuuksien mukaan. Projektipäällikkö huolehtii roolituksesta projektin aikana. (Kruchten 2004, 36-37.)

Dokumentointi RUP:ssa on vaihtelevaa sen joustavuuden vuoksi. Malli kuitenkin suosittelee, että tuotosten kirjaamisessa ja ylläpitämisessä käytettäisiin paperidokumenttien sijaan sopivia välineitä. Tämä pitää huolen siitä, että tiedot on helpompi pitää ajan tasalla ja ne ovat aina saatavilla. (Kruchten 2004, 40-42.)

2.2 Lean-ajattelu

Leanin pääidea perustuu kaiken turhan tekemisen poistamiseen prosessista eli hukan minimoimiseen. Samalla on tarkoitus maksimoida asiakkaan saama hyöty ja tuoda lisäarvoa asiakkaalle. Tiettyjen pisteiden sijaan turhaa tekemistä poistetaan pitkin projektia. Tämä luo prosesseja, jotka vaativat vähemmän työtä, tilaa, pääomaa ja aikaa. Tämän ansiosta asiakkaan tarpeisiin ja vaatimuksiin pystytään vastaamaan tehokkaammin. Myös dokumentointi ja tiedonhallinta muuttuvat yksinkertaisemmaksi ja tarkemmaksi. Lean-ajattelua kuvataan yleensä Lean-talolla (kuva 5). (Lean Enterprise Institute 2016.)



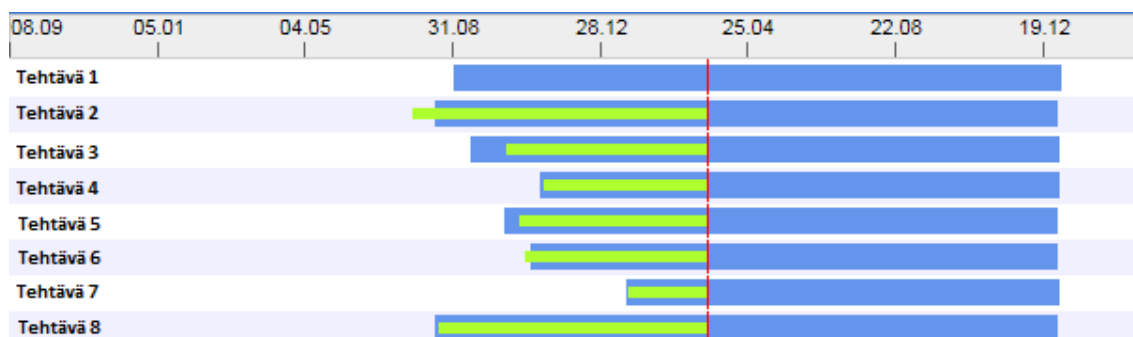
Kuva 5. Lean-talo.

Lean-talo on visuaalinen kuvaus Leanin peruselementeistä, johon koko Lean-filosofia perustuu. Talon sisusta koostuu tiimeistä. Talon perustus koostuu filosofiasta, visuaalisesta johtamisesta, vakaasta ja standardoidusta prosessista sekä kysynnän ymmärtä-

misestä ja siirtämisestä arvon tuottoon. Vankan perustuksen päälle tulevista tukipila-reista vasemmanpuoleinen JIT eli Just-in-time pilari edustaa kehitystoiminnan proses-sia, saatetaan oikeat ihmiset tekemään oikeita asioita oikeaan aikaan. Oikeanpuolei-nen pilari, Jidoka (suomeksi autonomaatio eli ihmisavusteinen automaatio), kuvastaa toiminnan kiihdyttämistä poistamalla esteitä. Jidoka viittaa kykyyn keskeyttää tuotanto ihmisvoimin tai koneellisesti mikäli ilmenee koneen toimintahäiriö, laatuvirhe tai muu vastaava. Katto sisältää tavoitteiden saavuttamisen matalammilla kustannuksilla ja lyhyemmällä läpimenoajalla laatua unohtamatta. Mikään näistä elementeistä ei johda toivottuun lopputulokseen yksistään, vaan täydentävät toisiaan ja niiden sisältämiä työ-kaluja tulee käyttää tilanteen mukaan. (Jokela 2011.)

2.3 Gantt-kaavio

Gantt-kaavio (kuva 6) on yksi vanhimmista projektihallinnan menetelmistä, ja se on hyvin suosittu työkalu. Sen tarkoituksena on esittää projektin ja sen eri työvaiheiden edistymistä suhteessa aikaan. (Hughes & Cotterell 2005, 175-176.)



Kuva 6. Valmis-hankkeen toteutusprojekteissa käytettävä Gantt-kaavio.

Gantt-kaaviossa sijoitetaan yleensä ylimmäiseksi vaakatasoon aikajana ja työvaiheet allekkain kaavion vasempaan laitaan. Kutakin työvaihetta kuvaa yksi jana, ja janat sijoitetaan työtehtävää vastaavalle riville aikajanan siihen kohtaan, jolloin työ on tarkoitus aloittaa. Janaa venytetään aina siihen pisteeseen asti, jolloin työvaiheen on määrä olla valmis. Kaavioon sijoitettujen janojen on tarkoitus selkeyttää työvaiheiden toteutusjär-jestystä ja niiden vaatimaa työmäärää. Janojen värityksellä voidaan havainnollistaa työvaiheen edistymistä. Yleensä kaaviossa on lisäksi kyseisen päivän kohdalla jonkin-

lainen merkki, johon vertaamalla voi helposti nähdä, onko työvaihe aikataulussa. (Hughes & Cotterell 2005, 176-177.)

Gantt-kaavion etuna pidetään sen visuaalisuutta, joka auttaa hahmottamaan projektin aikataulun kokonaisuuden. Selkeällä kaaviolla on helppo pitää työntekijät perillä projektin etenemisestä sekä tehdä selvitystä projektin tilanteesta. Gantt-kaavio on kuitenkin hyvin yksinkertainen. Se esittää työmäärän ainoastaan suhteessa aikaan eikä ota huomioon muita työn vaativuuteen liittyviä tekijöitä. Työvaiheiden väliset riippuvuudet eivät ole aina selvästi tunnistettavissa ja jos työvaiheita on lukumääräisesti monia, saattaa kaavion visuaalinen selkeys kärsiä. (Hughes & Cotterell 2005, 175-176.)

2.4 Ohjelmistoprojektit

Ohjelmistoprojekteille tyypillistä ovat jatkuvasti muuttuvat vaatimukset. Tämän vuoksi perinteisiä projekti- tai vesiputousmalleja on vaikea hyödyntää. Muutoksista aiheutuvat kustannukset ovat liian suuria ja aiheuttavat aikataulun venymistä. Vesiputousmallin teoria tunnetaan kuitenkin hyvin ja sen vaihtoehtoisia muotoja, joissa iteratiivisuus sallitaan, käytetään suurimmassa osassa ohjelmistoyrityksiä. (Taina 2009, 130-134.)

Ohjelmistoprojekteissa yleensä ennen projektin aloittamista on tunnistettu jokin tarve, jonka seurauksena syntyy projektialoite tai projektiehdotus. Aloite on saatettu tehdä vain suullisena toteamuksena siitä, että jonkinlaiseen ongelmaan pitäisi löytää ratkaisu. Etenkin suurissa ohjelmistoprojekteissa on joskus helpompaa jakaa projekti osaprojekteiksi. (Haikala & Mikkonen 2013, 21.)

2.4.1 Projektin osittaminen

Perinteisen projektisuunnittelun yksi tärkeimmistä ja keskeisimmistä tehtävistä on projektin osittaminen, jossa projekti jaetaan sopiviin osiin tai osaprojekteihin. Osaprojekteja voivat olla esimerkiksi määrittely-, suunnittelu- ja toteutusprojekti. Osittamiseen ja vaiheistukseen vaikuttaa suuresti työn luonne ja laajuus. Ei ole olemassa yhtä oikeaa tapaa projektin osittamiseen. Suuria tehtäväkokonaisuuksia tulisi välttää, sillä niiden

todellista aikaa on vaikea arvioida. Niissä on myös vaarana, että poikkeamia ei huomata ajoissa. (Haikala & Mikkonen 2013, 157-159.)

Ohjelmistoprojekteissa osittaminen pitkälle tulevaisuuteen on hankalaa jatkuvasti muuttuvien vaatimuksien takia. Pitkissä projekteissa ositusta tarkennetaan työn edetessä eikä koko projektia ole tarkoitus osittaa aluksi kovinkaan tarkasti. Ositusta on tarkasteltava aikataulun ja tehtävien osalta säännöllisesti. (Haikala & Mikkonen 2013, 157-159.)

Yleensä suuret ohjelmistoprojektit alkavat määrittelyprojektilla, jossa pyritään määrittelemään mahdollisimman tarkasti, millaisen ohjelmiston tilaaja tarvitsee. Tässä vaiheessa ohjelmiston tilaaja esittää tarpeensa ja toiveensa ohjelmistosta ja mahdollisesti näyttää tai antaa määrittelijälle pääsyn olemassa olevaan ohjelmistoon. Määrittelyssä tarkoituksena on luoda pohja ja kehykset sille, millainen ohjelmistosta tulee. Määrittelyprojektin jälkeen saadaan määrittelydokumentaatio eli speksi siitä, millainen ohjelmisto tarvitaan. Suuressa ohjelmistoprojektissa voidaan eriyttää monet eri kokonaisuudet pienemmiksi ketteriksi projekteiksi, joita voidaan kehittää vaikka limittäin toistensa kanssa. Myös olemassa olevan järjestelmän korvaaminen uudella voidaan toteuttaa ketterästi. Pieni osa nykyistä järjestelmää voidaan korvata uudella ja hiljalleen siirtyä uudempaan, aina osan kerrallaan. (Kärenaho 2016.)

Määrittelyä seuraa suunnitteluvaihe, jossa ideaa työstetään eteenpäin ja siihen liittyviä yksityiskohtia täsmennetään. Suunnitteluvaiheen aikana tuotetaan projektisuunnitelma, jota käytetään suunnittelun, seurannan ja ohjauksen apuvälineenä. Tästä suunnitelmasta käyvät ilmi muun muassa projektin lähtökohdat, tavoitteet ja rajaukset, tehtävien määrittely, työmääräarvio, kustannusten arviointi ja budjetointi, projektiorganisaatio ja sen tehtävät sekä lopullisen tuotteen arvioinnin perusteet. (Haikala & Mikkonen 2013, 166.)

Toteutusvaihe sisältää varsinaisen kehitystyön ja testauksen. Toteutusvaiheessa on tärkeää suorittaa jatkuvaa raportointia ja seuranta projektin edistymisestä. Seuranta suoritetaan projektin ohjausryhmän palavereissa, joissa projektin edistymistä verrataan suunnitteluvaiheessa tehtyyn projektisuunnitelmaan. Tällä pyritään huomaamaan mahdolliset poikkeamat tarpeeksi ajoissa, jotta korjauksille voidaan varata enemmän aikaa.

Toteutusvaihe päättyy, kun katsotaan, että asetetut tavoitteet on saavutettu. (Haikala & Mikkonen 2013, 164.)

2.4.2 Projektin organisaatio

Projektin toteuttamisesta ja siihen liittyvästä päätöksenteosta vastaavat projektiin osallistuvat ihmiset eli projektin organisaatio. Kaikkien työskentelevien henkilöiden vastuut, velvollisuudet ja valtuudet on ennalta määritelty ja sovittu. Projektionnisaatiota voidaan kuvata tavoiteorientoituneeksi ja kertakäyttöiseksi, sillä se on väliaikainen tiimi, jonka päämääränä on sille asetettujen tavoitteiden saavuttaminen. Projektionnisaatiolle kaikista tärkeintä tavoitteiden saavuttaminen, ja siksi organisaation on tarvittaessa kyettävä nopeisiin muutoksiin, vaihtuvaan työskentelytahtiin ja kestävä stressiä. (Ruuska 2007, 130-132.)

Tyypillinen projektionnisaatio koostuu yleensä projektipäälliköstä, ohjausryhmästä ja projektitiimistä. Projektipäällikkö on projektin operatiivinen johtaja, jolle kuuluu projektin päivittäisjohtaminen. Projektipäällikön päävastuuna on projektille asetettujen tavoitteiden saavuttaminen ennalta määrättyillä resursseilla. Projektipäällikön tulee osata johtaa ja ohjata projektissa työskenteleviä ihmisiä ja varmistaa, että projekti etenee laaditun aikataulun mukaan. (Ruuska 2007, 137-150.)

Projektitiimissä työskentelevät henkilöt vastaavat projektin varsinaisesta toteutuksesta. Projektipäällikkö voi itse vastata projektitiimin nimeämisestä tai se voidaan tehdä hänen puolestaan. Tiimin valinnassa on tärkeää huomioida henkilöiden osaaminen ja ammattitaito. Projektitiimin jäsenet raportoivat projektin etenemisestä projektipäällikölle, joka taas puolestaan raportoi asiasta ohjausryhmälle. (Lester 2007, 296.)

Ohjausryhmällä on ylin päätösvalta, ja se sisältää jäseniä kaikilta projektin vaikutusalueilta. Ohjausryhmän tehtävänä on suunnitelman vahvistaminen ja hyväksyminen, resurssien jakaminen, projektin etenemisen seuraaminen ja projektipäällikön tukeminen asetettujen tavoitteiden saavuttamiseksi. Ohjausryhmä myös päättää, milloin projektissa voidaan siirtyä vaiheesta toiseen ja milloin projekti päätetään. (Lester 2007, 35-36.)

2.4.3 Tyypillisimmät ongelmat projektityöskentelyssä

Projekteilla on tapana ajautua kohti epäjärjestystä, sillä ne sisältävät yleensä paljon vaikeasti ennakoitavia epävarmuustekijöitä. Epäonnistuminen johtuu harvoin tekniikasta tai työvälineistä. Suurin osa projektien epäonnistumisista johtuu projektinhallinnassa olevista puutoksista. (MAMK 2013, 52.)

Yksi tyypillisistä projektin ongelmista on puutteellisesti suoritettu määrittely. Projekti aloitetaan liian helposti eikä kustannuksia ja hyötyjä ole selvitetty tarpeeksi laajasti. Määrittelyn tarkoituksena on karsia pois kannattamattomat projektit. Huolellisella määrittelyllä projektin tavoitteet ja rajaukset ovat selkeät eikä jatkuville muutosehdotuksille ole tarvetta. (MAMK 2013, 53-55.)

Projektin onnistumisessa olennaista on myös henkilöstö ja sen johtaminen. Mikäli projektissa työskentelevien henkilöiden osaaminen ei ole tarvittavalla tasolla tai he eivät vain sovi projektiryhmään, voi projektin eteneminen vaikeutua. Myös projektipäällikön rooliin pitää valita oikea henkilö. Projektipäällikön tulee olla tarpeeksi vaativa ja pitää huoli aikataulussa pysymisestä. (MAMK 2013, 56.)

Projektin yhtenä tärkeimpänä osana pidetään suunnittelua. Projektin suunnittelu on tärkeää hoitaa kunnolla. Liian optimiset aikataulut ja työarviot vaikeuttavat projektin seuraamista ja projektin kulun ennakoimista. Huono suunnittelu aiheuttaa epävarmuutta ja epävarmuus puolestaan johtaa riskeihin. Suunnittelun merkitystä ei voi ylikorostaa. Koko projektinhallinta perustuu projektisuunnitelmaan, ja se määrää lopulta projektin onnistumisen. Jos projektia ei ole suunniteltu, ei tiedetä tavoitteita ja onnistumista ei voida todeta. (MAMK 2013, 59-63.)

2.4.4 Riskien hallinta

Riskienhallinnan tavoitteena on tunnistaa riskit ajoissa ja pyrkiä estämään tai ainakin minimoimaan niiden vaikutus. Riskien tunnistamiseen ja arvioimiseen käytetään muodollista ja toistettavaa hallintaprosessia (kuva 7). Sen avulla voidaan myös määrittää riskienvähennystoimet sekä seurata ja raportoida riskienhallinnan onnistumista ja edistymistä. (Murch 2002, 163.)



Kuva 7. Riskienhallintaprosessi. (Pöyry 2016.)

Riskienhallintaprosessissa riskit tunnistetaan kirjaamalla ne alustavaan riskilistaan, jonka jälkeen riskit analysoidaan. Riskiin liitetään todennäköisyys ja seurauksen vakavuus ja samalla mietitään riskin ennusmerkkejä ja mahdollisia seurauksia sidosryhmissä. Lopuksi selvitetään, voidaanko riskiltä välttyä kokonaan tai miten sen seuraukset minimoidaan. (Haikala & Mikkonen 2013, 166.)

Yleisellä tasolla riskit jaetaan yleensä viiteen luokkaan, joita ovat ulkoiset riskit, kustannusriskit, aikatauluriskit, tekniikkariskit ja toiminnan riskit. Ohjelmistoprojekteissa on kuitenkin tapana luokitella riskit projektiriskeihin, tuoteriskeihin ja liiketoimintariskeihin. Projektiriskin sattuessa aikataulut ovat vaarassa viivästyä tai koko projektin valmistuminen vaarantuu. Tuoteriskit ovat riskejä, jotka vaikuttavat tuotteen laatuun. Liiketoimintariskit liittyvät markkinoilla tapahtuviin muutoksiin, jotka tapahtuessaan aiheuttavat projektin alun perin suunnitellun tuoton merkittävän pienenemisen. (Haikala & Mikkonen 2013, 165.)

Useissa perinteisissä menetelmissä on hyvin tavallista, että mahdollisiin ongelmiin ei juurikaan varauduta etukäteen. Ongelmia korjataan sitä mukaan kuin niitä ilmenee. Varsinaista riskienhallintaa ei siis ole. Pitkällä aikavälillä tämä ei ole kovinkaan järkevää

tai tehokasta, sillä se saattaa aiheuttaa lisäkuluja projektille tai vaarantaa projektin aikataulun. Tämän vuoksi on erittäin vaikeaa löytää kattavia ja yksityiskohtaisia riskienhallinnan oppaita perinteisille ohjelmistokehityksen menetelmille.

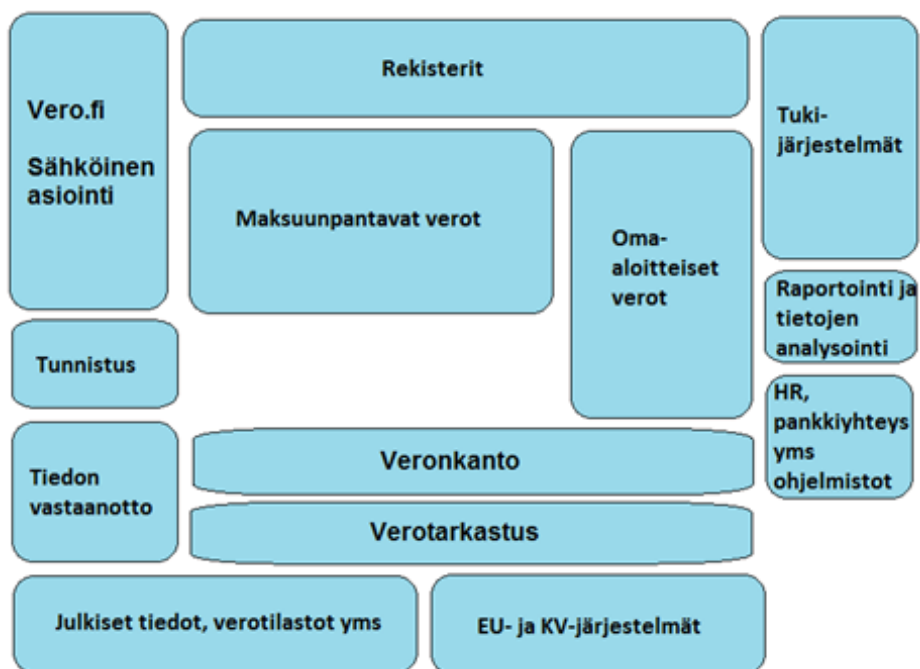
Ketterissä menetelmissä riskienhallinta on osittain sisäänrakennettuna. Riskejä seurataan jatkuvasti toistuvissa palavereissa. Varsinaisen riskienhallintasuunnitelman tarpeellisuutta ketterät menetelmät eivät kuitenkaan poista. (Haikala & Mikkonen 2013, 166.)

3 Projektityöskentely Valmis-hankkeessa

Verohallinnon olemassa oleviin tietojärjestelmiin ja järjestelmäalustaan on tunnistettu liittyvän merkittäviä elinkaaririskejä. Vanha järjestelmärakenne on pirstaleinen, hankalasti ylläpidettävä ja kehitettävä ja sen ylläpito vie vuosittain yhä enemmän voimavaroja. Se ei myöskään tue tehokkaasti verotusprosesseja ja virkailijatyötä. Virkailijoilta puuttuu esimerkiksi kokonaisnäkyä asiakkaan tilanteesta.

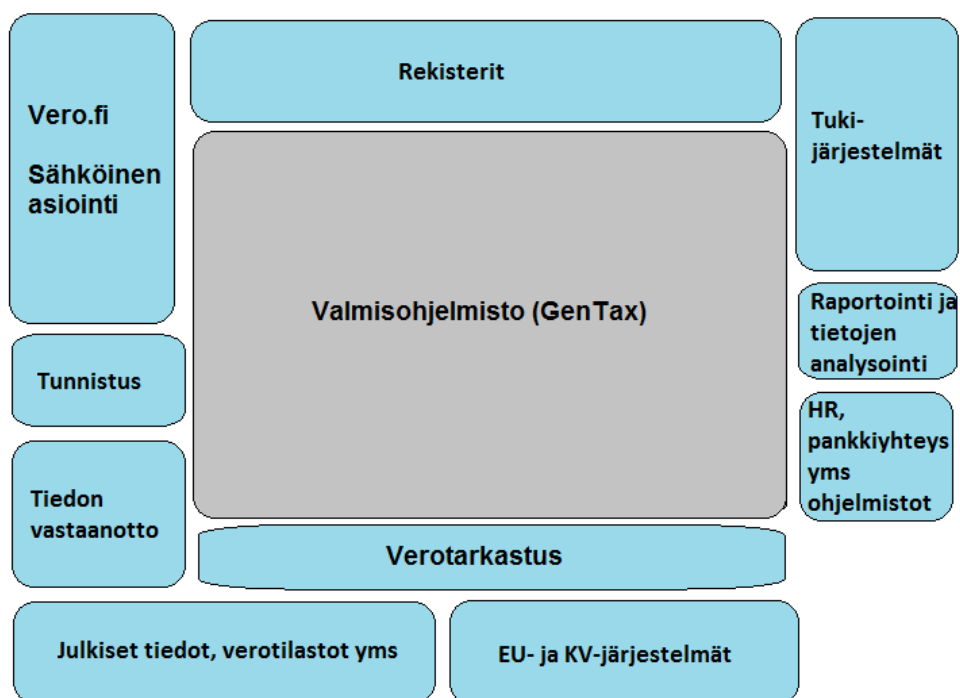
Valmis-hankkeen tavoitteena on uudistaa verotusjärjestelmä ja sen lomassa parantaa verotuksen toimintaprosesseja sekä yksinkertaistaa ja yhtenäistää vanhaa menettelylainsäädäntöä vastaamaan nykyisiä tarpeita. Hankkeen tehtävät on ositettu useaan eri projektiin. Näitä projekteja ovat mm. valmisohjelmiston hankinta, kokonaissuunnittelu-projekti, ennen jokaista toteutusprojektia suoritettavat valmisteluprojektit sekä varsinaiset toteutusprojektit. Kukin projekti tuottaa oman projektisuunnitelmansa, jossa on mainittu keskeiset tuotokset sekä aikataulut. (Hankesuunnitelma 2013.)

Valmisohjelmiston käyttöönotto mahdollistaa nykyistä nopeamman ja joustavamman verotusohjelmistojen jatkokehittämisen sekä tarjoaa virkailijoille työn tehokkuutta, laatua ja työtyytyväisyyttä lisäävät työkalut. Hankkeessa on tarkoitus ottaa vaiheittain käyttöön eri verolajeille yhteinen ja korkeaa automaatioastetta tukeva verotuksen valmisohjelmistokokonaisuus (kuvat 8 ja 9). (Hankesuunnitelma 2013.)



Kuva 8. Yksinkertaistettu versio Verohallinnon järjestelmäkartasta ennen valmisohjelmiston käyttöönottoa.

Suurin osa nykyisistä järjestelmistä korvataan joko kokonaan valmisohjelmistolla tai kehitetään liittymä vanhan järjestelmän ja uuden välille. Muista järjestelmistä korvataan osa toiminnoista ja jäljelle jääviin toimintoihin kehitetään liittymät. GenTaxin on tarkoitus korvata kokonaan maksuunpantavien verojen, oma-aloitteisten verojen ja veronkannon järjestelmät. Kokonaan liittymien avulla hoidettavia osia ovat mm. sähköinen asiointi, tietojen analysointi ja pankkiyhteydet. Ainoastaan tunnistusjärjestelmä säilyy täysin ennallaan. (Hankesuunnitelma 2013.)



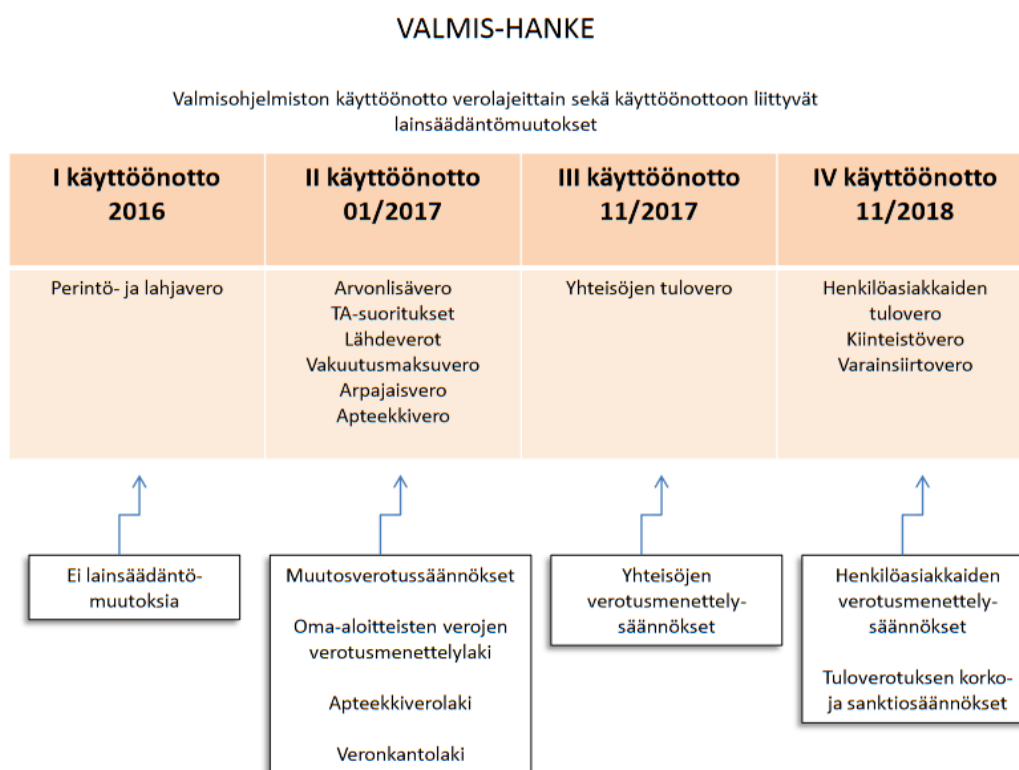
Kuva 9. Verohallinnon järjestelmäkarta valmisohjelmiston lopullisen käyttöönoton jälkeen.

Hankkeen lopputuloksena Verohallinnon järjestelmäkarta näyttää yksinkertaistettuna yllä olevan kuvion kaltaiselta. Tällöin valmisohjelmisto, GenTax, on korvannut suurimman osan nykyisistä järjestelmistä.

Valmis-hankkeen projektimalli poikkeaa jonkin verran yleisistä projektinhallinnan malleista. Suurimpana syynä tähän on se, että hankkeen molemmat pääosapuolet, Verohallinto ja Fast, suosivat eri menetelmiä projektinhallinnassa. Tämän vuoksi Valmis-hankkeessa vallitsee projektinhallinnan kannalta tietynlainen kahden näkemyksen verkko. Perinteisiä ja ketteriä menetelmiä on käytössä enemmän tai vähemmän lähes jokaisessa vaiheessa ja jokaisella projektinhallinnan osa-alueella. Projektinhallinta on tästä syystä jatkuvasti pintaan nouseva keskustelun aihe Valmis-hankkeessa. Menetelmiä ja käytäntöjä on jatkuvasti muokattava molempia osapuolia miellyttäväksi. Seuraavissa kappaleissa on tarkoitus keskittyä näihin menetelmiin eri projektinhallinnan osa-alueilla.

3.1 Aikataulu ja projektin vaiheet

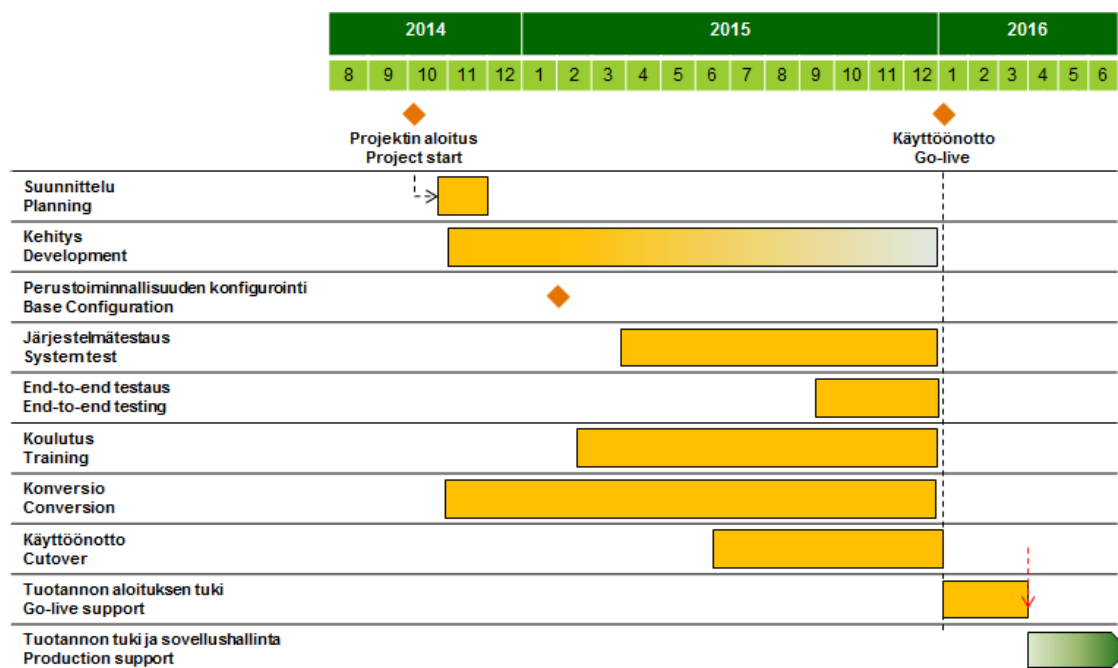
Koko hankkeen kesto on yhteensä kuusi vuotta, josta uuden järjestelmän konfigurointi ja käyttöönotto vie noin neljä vuotta. Projektia toteutetaan vaiheittain jakaen projekti vuosittaisiksi jaksoiksi. Näistä jaksoista puhuttaessa käytetään yleensä termiä rollout. Joka vuosi toteutetaan tietyt, ennalta määritellyt verolajit järjestelmään (kuva 10).



Kuva 10. Uuden verotusjärjestelmän käyttöönoton alkuperäinen suunnitelma. (Valtiovarainministeriö 2015.)

Ensimmäisenä vuonna on tarkoituksenaan toteuttaa perintö- ja lahjavero sekä joitain ohjelmiston perustoiminnallisuuksia. Lainsäädäntömuutoksia ei ensimmäisenä vuonna ole. Toiseen käyttöönottoon olisi tarkoitus saada toteutettua arvonlisävero, työnantaja-suoritukset, lähdeverot, vakuutusmaksuvero, arpajaisvero sekä apteekkivero. Lisäksi on hoidettava näihin liittyvät lakimuutokset ja muutosverotussäännökset. Kolmannen ja neljännen vuoden aikana hoidetaan jäljellä olevat verolajit: yhteisöjen tulovero, henkilöasiakkaiden tulovero, kiinteistö vero ja varainsiirtovero. Yhteisöjen verotusmenettelysäännökset hoidetaan kolmantena vuonna ja henkilöasiakkaiden verotusmenettely-

säännökset neljäntenä vuonna. Koko järjestelmän pitäisi alkuperäisen suunnitelman mukaan olla valmiina marraskuussa 2018. Tästä alkuperäisestä suunnitelmasta on kuitenkin jouduttu jo joustamaan ja tämänhetkinen päättymispäivä olisi vuoden 2019 puolella.



Kuva 11. Ensimmäisen toteutusprojektin aikataulu. (Hankekäsikirja 2016.)

Jokainen vuosi on myös tavallaan jaettu eri vaiheisiin. Kehitystyö on suunniteltu alkavaksi varsinaisesti helmikuussa ja sitä jatketaan aina marraskuun (ensimmäinen ja toinen vuosi) loppuun asti. Tietojen konversiota vanhoista järjestelmistä suoritetaan rinnakkain kehitystyön kanssa. Kehitystyön alussa suoritetaan perustoiminnallisuuden kehitys (base configuration) eli tehdään suuntaa antava pohja toiminnallisuuksille. Tämä pohja testataan ja sen hyväksyttämisen jälkeen voidaan toiminnallisuus kehittää valmiiksi. Varsinainen testaus on tarkoitus aloittaa yleensä siinä vaiheessa, kun isommat kokonaisuudet on todettu toimiviksi. Järjestelmätestausta jatketaan aina rolloutin loppuun asti. Samoihin aikoihin testauksen kanssa aloitetaan myös loppukäyttäjien koulutus. Rolloutin loppupuoliskolla suoritetaan vielä kokonaisvaltainen järjestelmän testaus. Viimeinen kuukausi ennen käyttöönottoa on niin sanottua ”development freeze”-aikaa, uusia toiminnallisuuksia tai ohjelmiston osia ei enää kehitetä, ainoastaan korjataan testauksessa ilmeneviä virheitä. Uusien toiminnallisuuksien käyttöönottoon

valmistaudutaan jo hyvissä ajoin kuukausia ennen, mutta varsinaisesti toiminnallisuudet otetaan käyttöön rolloutin lopulla muutaman päivän ajan kestävän siirtymän aikana, josta käytetään termiä ”go-live”.

Verrattaessa kappaleessa 2 käsiteltyihin yleisiin menetelmiin ja malleihin, rolloutit muistuttavat sisällöltään ja vaiheiltaan jokseenkin vesiputousmallia ja RUP:ta. Alkuperäisenä suunnitelmana oli, että kaikki vaiheet voitaisiin suorittaa loppuun asti tietyssä järjestyksessä ilman palaamista aikaisempiin vaiheisiin. Haluttiin siis lähes puhdas vesiputousmalli. Rolloutin vaiheilla ilmenee kuitenkin päällekkäisyyksiä. Esimerkiksi kehitystyötä ei aina voida tehdä täysin valmiiksi ennen testauksen aloittamista. Myös jotkin yksittäiset työtehtävät ovat riippuvaisia tietyistä toiminnallisuuksista. Lopputulos on siis lähempänä RUP:n iteratiivista mallia kuin perinteistä vesiputousmallia. Vesiputousmallin voi kuitenkin havaita yksittäisten työtehtävien ja toiminnallisuuksien kehittämisessä. Alueen tai toiminnon kehitystyötä ei aloiteta ilman määrittelyä ja suunnittelua. Kehitystyö pyritään myös saamaan loppuun asti, tai ainakin mahdollisimman pitkälle, ennen testauksen aloittamista.

Lean-ajattelusta ja sitä kuvaavasta Lean-talosta (ks. sivu 8) voidaan kehitystyötä ajatellen ottaa toinen tukipilari. Projektissa on suhteellisen toimiva tapa panna jokin asia hetkeksi jäihin, mikäli suurempia ongelmia ilmenee. Tämä tekee kehittämisprosessista myös hieman sulavamman. Projektissa pyritään myös vahvasti tekemään asioita lyhyemmällä ajalla ja pienemmillä kustannuksilla, jotta Lean-talolle saataisiin katto.

Jos katsotaan kaikkia rollouteja yhdessä, projektin aikataulutus ja vaiheistus omaa samoja piirteitä kuin Scrum-projektimalli. Rolloutit vastaavat tavallaan Scrumin sprinttejä, jotka ovat huomattavasti pitempiketoisia ja sisällöltään laajempia. Kuten sprinttien lopussa, myös rolloutien loputtua pitäisi saatavilla olla toimiva tuote. Ennen kehittämisen aloittamista luodaan suunnitelma rolloutille, joka vastaa Scrumin sprinttisuunnitelmaa. Tästä suunnitelmasta selviää myös rolloutin aikana suoritettavat tehtävät.

Yksittäisten tehtävien ja järjestelmän toiminnallisuuksien kehittämisen aikataulutuksessa hyödynnetään pääasiassa Gantt-kaaviota (ks. sivu 9). Työtehtävien edistymisen seuranta on välillä hankalaa. Joskus edistymistä seurataan ehkä liiankin sokeasti projektin Gantt-kaavion pohjalta eikä tiettyjä toiminnallisuuksien ja tehtävien välisiä riippuvuuksia huomioida. Tämä saattaa aiheuttaa yhdelle kehittäjälle tai kokonaisuudelle tiimille

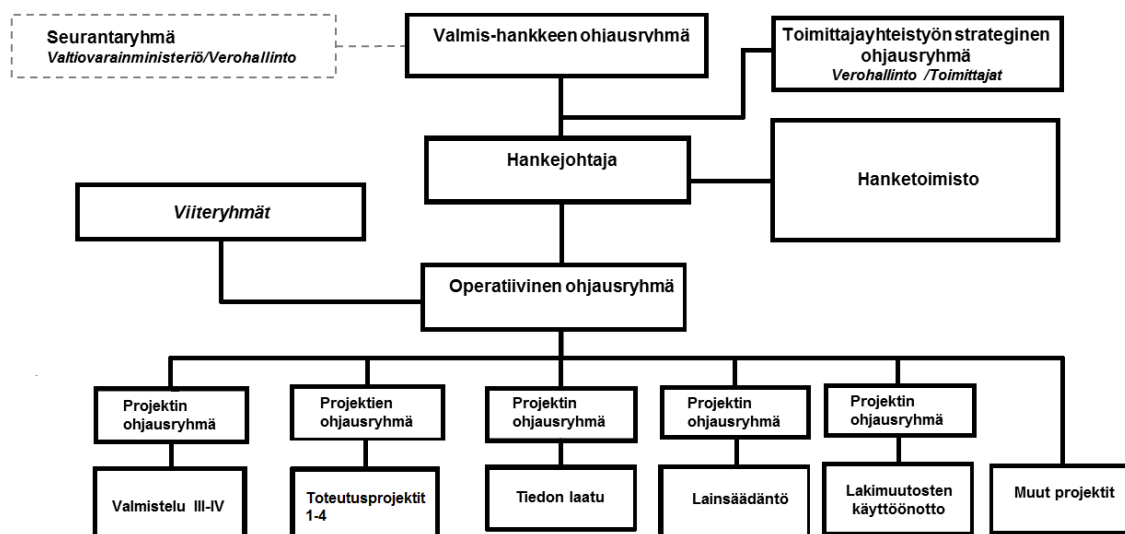
turhia paineita, joka taas vaikuttaa työskentelyyn. Tässä nousee esiin taas se, että kaikki eivät tunne kehitysprosessia ja sen rajoja läpikotaisin.

Projektin vaiheistus ja kokonaistyömäärän jakaminen eri vuosille toimii vaihtelevasti. Alkuperäisessä aikataulussa pysyminen on haastavaa projektin suuren koon ja pitkään kestävien vaiheiden vuoksi. Joskus taas ennalta määrätty aikataulut ovat aivan liian optimistisia. On enemmän kuin mahdollista, että joitakin toivottuja toimintoja joudutaan siirtämään seuraavaan rolloutiin. Yksittäisten tehtävien aikatauluista pystytään toisiinsa joustamaan, mutta isompien kokonaisuuksien valmistumisen myöhästyminen ei ole suotavaa. Suurempien kokonaisuuksien ja ohjelmiston osien valmistuminen vuodenvaihteeseen mennessä on erittäin tärkeää. Koko rolloutin myöhästymisen ja suurempien menetysten välttämiseksi määrittelyvaiheessa määritellään ohjelmistosta ne toiminnot, jotka voidaan tarvittaessa jättää projektin myöhäisempään vaiheeseen.

Uusien työntekijöiden kannalta vaiheistus on kuitenkin toimiva. Uusia työntekijöitä on helpompi kouluttaa ja palkata projektiin kesken kaiken, koska joka vuosi aloitetaan, ainakin jossain määrin, puhtaalta pöydältä.

3.2 Organisaation rakenne, tehtävät ja kommunikointi

Valmis-hankkeen organisaatio (kuva 11) on hyvin laaja, ja henkilöresursseja on valtavasti.



Kuva 12. Hankeorganisaatio. (Hankesuunnitelma)

Hankkeen ylin ohjaava elin on hankkeen ohjausryhmä, jonka puheenjohtajana toimii Verohallinnon pääjohtaja. Ohjausryhmä hyväksyy mm. hankkeen keskeiset tuotokset, räättälöinnit, vuosibudjetin muutokset sekä kysymykset, joilla on asiakasvaikutuksia. Hankkeen etenemisestä ohjausryhmälle raportoi kuukausittain hankejohtaja. Hankejohtajan vastuulla on koko hankkeen operatiivinen johtaminen ja tavoitteiden saavuttaminen ohjausryhmän asettamien tavoitteiden, aikataulun ja budjetin mukaisesti. Hankejohtaja toimii myös puheenjohtajana hankkeen operatiivisesta ohjauksesta vastaavalle operatiiviselle ohjausryhmälle. Sen tehtävänä on ohjata hankkeen projekteja ja käsitellä asiat, jotka liittyvät projektien tehtäviin, aikatauluihin, resursseihin ja riskeihin. Operatiivisella ohjausryhmällä on kokonaisnäkyvyys hankekokonaisuuteen ja sen kokouksissa hankkeen vastuuhenkilöt pystyvät yhteisesti ohjaamaan koko hanketta. (Hankekäsikirja 2016.)

Jokaiselle projektille on myös oma ohjausryhmänsä. Projektit ovat hankkeen keskeisiä johtamistasoja, joista vastaavat projektipäälliköt. Projektipäälliköiden lisäksi olennaisessa asemassa ovat toiminnallinen vastaava, toiminnallinen asiantuntija ja käytännön asiantuntija. Toiminnallinen vastaava huolehtii koko projektin toiminnallisista vaatimuksista ja viestii ne toimittajalle, kun taas toiminnallinen asiantuntija vastaa vain oman alueensa (esim. testaus, koulutus) toiminnallisista vaatimuksista. Käytännön asiantuntijan tehtävänä on käytännön verotusmenettelyn ja virkailijatyön vaatimusten

tuonti valmisohjelmistomäärittelyihin. Käytännön asiantuntija raportoi töiden etenemisestä ja sisällöllisistä ratkaisuista toiminnalliselle asiantuntijalle, joka puolestaan raportoi toiminnalliselle vastaavalle ja projektipäällikölle. (Hankesuunnitelma 2013.)

Koko Valmis-hankkeen organisaatiomalli on lähestulkoon kuin laajennettu versio perinteisessä vesiputousmallissa. Merkittäviä samoja rooleja molemmissa ovat esimerkiksi projektipäälliköt, erilaiset ohjausryhmät ja koko kokonaisuutta hoitava ohjelmapäällikkö eli hankejohtaja. Toteutusprojekteissa vallitsee Scrumin ja vesiputousmallin tietynlainen sekoitus. Tiimejä ja tuotteen omistaja löytyy, mutta varsinaista Scrummaster ei ole. Tiimien johtajia voisi pitää Scrummastereina, mutta he osallistuvat yhtä paljon tiimin töihin kuin muutkin kehittäjät. Tämä hieman rikkoo Scrummasterin työnkuvaa. Tiimit eivät myöskään ole aina niin monitaitoisia kuin voisi kuvitella. Verotuksen ja sitä vastaavat GenTaxin osa-alueet ovat laajoja, että mikäli haluaa ymmärtää yhden osa-alueen hyvin, on siihen parempi syventyä ja erikoistua.

Jokainen verotuksen osa-alue on oma tehtävänsä ja jokaiselle osa-alueelle on oma tiiminsä. Kaikki kehittäjät pystyvät tekemään kaikkea järjestelmään liittyvää kehitystyötä, mutta pääideana on saada kehittäjille syventävä osaaminen oman tiiminsä vastuualueesta ja jakaa näin työtaakkaa ja tehostaa kehitystyötä. Eri osa-alueita ja samalla tiimejä ovat esimerkiksi maksut ja rahaliikenne, konversio, asiakastietojen hallinta ja kirjeet (erilaiset verotukseen liittyvät kirjeet, joita lähetetään asiakkaille). Tiimit koostuvat kehittäjistä, joita ovat lähinnä Fastin ja Nortalin työntekijät, ja Verohallinnon työntekijöistä. Kehittäjistä ja heidän työstä vastaavat tiimien johtajat (team leads), jotka ovat yleensä Fastin pitkäaikaisempia työntekijöitä ja tietävät näin GenTaxin toiminnan paremmin. Verohallinnon työntekijät toimivat lähinnä verotuksen asiantuntijoina ja GenTaxiin kehitettävien toiminnallisuuksien testaajina.

Tämä jaottelu aiheuttaa työmäärän jakautumisessa epätasaisuutta. Koska tiimit on jaoteltu eri verotuksen osa-alueiden mukaan ja joka vuosi keskitytään eri verolajeihin, aiheuttaa se vuosittain painotuksien vaihtelua tiimien välillä. Joidenkin tiimien työntekijät joutuvat tekemään valtavasti ylitöitä sillä välin, kun toiset tekevät hommia puoliteholla. Tämä on täysin Lean-ajattelua vastaan, ja henkilöresurssit pitäisi hallinnoida paremmin. Oikeat ihmiset ovat kyllä tekemässä oikeita asioita, mutta myös muitakin ihmisiä voisi ohjata avustamaan näissä asioissa.

Henkilöstön vaihtuvuutta esiintyy jonkin verran. Hankkeeseen otetaan jatkuvasti uusia työntekijöitä ja samalla vanhoja siirtyy muihin tehtäviin. Uusille työntekijöille järjestetään peruskoulutus, joka sisältää perusteet hankkeen tavoitteisiin, organisaatioon, työskentelytapoihin ja työkaluihin. Valmisohjelmiston kehitystyölle ja osa-alueille on myös omat koulutuksensa. Uudelle työntekijälle nimitetään yleensä myös toiminnallinen vastaava, joka on vastuussa käytännön työnopastuksesta. Kehittäjät aloittavat työskentelyn ja perehdytyksen yleensä kirjeitimissä kehittämässä verovelvollisille lähetettäviä tulosteita. Tulosteet koetaan olevan suoraviivaisin ja kaikenkattavin osa-alue. Myöhemmin kehittäjiä siirretään myös muihin tehtäviin, jotta saadaan tarpeeksi osaamista jokaiselle valmisohjelmiston osa-alueelle.

Vesiputousmallin rooleja voi tiimeistä poimia muutamia. Tiimien johtajat ovat tavallaan projektipäälliköitä ja kehittäjiä samaan aikaan. Lisäksi on tietenkin varsinainen projektipäällikkö, joka vastaa kaikista tiimeistä. Lisäksi aiemmin mainitut, lähinnä asiantuntijoina toimivat Veron työntekijät muodostavat ikään kuin tietynlaisen ohjausryhmän jokaiselle tiimille. Näitä ohjausryhmiä johtaa SME-tittelillä toimiva henkilö, jota voidaan pitää palvelupäällikkönä.

Aiemmin käsiteltyä Lean-filosofiaa voi myös huomata henkilöstön hallinnassa ja kommunikoinnissa. Samoin kuin Leanissa myös Valmis-hankkeen toteutusprojekteissa yritetään saada oikeat ihmiset oikeisiin paikkoihin. Jokaisen osa-alueen jakaminen eri tiimeille myös tukee Lean-ajattelumallia. Koko kehitystyö nojautuu näiden tiimien toimivuuteen ja keskinäiseen kommunikointiin. Tämä muodostaa ikään kuin Lean-talon vasemmanpuoleisen tukipilarin. Aiemmin mainittiin myös, että Lean-talosta oikea tukipilari ja katto ovat tukevasti paikoillaan. Talolta puuttuu kuitenkin tukeva perustus. Prosessit ovat hyvin standardoituja, ja johtaminen on yleisesti ottaen toimivaa. Talon perustuksia haittaa kuitenkin asiakkaan eli Verohallinnon kysynnän ymmärtäminen. Se on ajoittain haastavaa, ja tämän johdosta lisäarvon tuottaminen vaikeutuu.

Koska Valmis-hankkeessa työskentelee valtavasti ihmisiä ja eri yhtiöiden edustajia, aiheuttaa kommunikointi välillä ongelmia. Valmisohjelmisto sisältää omat työkalunsa asioiden ja tehtävien hallitsemiseen ja eteenpäin viemiseen ja Verohallinnolla on omat käytäntönsä. Yleensä jokaiseen muutokseen ja tehtävään vaaditaan selkeät ohjeistuk-

set ja mielipiteet molemmilta osapuolilta. Yksinkertaisiakaan muutoksia ei saa tehdä, ellei jokainen asiassa jollain tavalla sidoksissa oleva henkilö ole siitä tietoinen tai ilmaissut asiasta mielipidettään. Tämä on ymmärrettävää ja toisaalta erittäin hyvä asia, mutta suuri byrokratian määrä saattaa aiheuttaa ”rikkinäinen puhelin”-efektin, joka taas puolestaan johtaa väärinymmärryksiin. Väärinymmärryksien seurauksena voi olla, että jokin asia toteutetaan joko osittain tai kokonaan ei-toivotulla tavalla. Tämä taas puolestaan johtaa kehitystyön lisääntymiseen, mikäli asiaa ei korjata ajoissa. Perimmäisenä ongelmana tähän kaikkeen on ehkä se, että kehittäjät eivät ole täysin perillä verotuksen prosesseista eivätkä kaikki testaajat tai asiantuntijat täysin mukana kehitystyössä. Sattunaisia lisäongelmia aiheuttaa myös kielimuuri suomen ja englannin kielen välillä.

3.3 Palaveri- ja tiedotuskäytännöt sekä dokumentointi

Pääasiallisena tiedotusvälineenä hankkeessa toimii sähköposti. Käytettävä sähköpostiohjelma on Microsoft Outlook. Suuren henkilömäärän takia käytössä on erilaisia sähköpostiryhmiä, jotka helpottavat viestin välittämistä oikeille henkilöille. Mikäli esimerkiksi halutaan kaikille kehittäjille lähettää postia, käytetään developers-ryhmää, jolloin viesti menee kaikille kehittäjille. Palaverit sovitaan ja niille tarvittavat tilat varataan Outlookin kautta. Tärkeimmät palaverit ja raporttien toimituspäivät löytyvät hankkeen operatiivisesta viikkorytmistä (kuva 13). (Hankekäsikirja 2016.)

	Maanantai	Tiistai	Keskiviikko	Torstai	Perjantai
Viikko 1	Toiminnallinen viiteryhmä Valmis-hankkeen ohjausryhmä		Kehittäjien tilannekatsaus Hankekokous 1 krt/kk	Muutospyynnöt Etenemäraportit	Operatiivisen ohjausryhmän materiaalit
Viikko 2	Hankkeen operatiivinen ohjausryhmä		Toteutusprojektin ohjausryhmä	Muutospyynnöt	
Viikko 3	Toiminnallinen viiteryhmä	Hanketiedote	Kehittäjien tilannekatsaus	Muutospyynnöt Etenemäraportit	Operatiivisen ohjausryhmän materiaalit
Viikko 4	Hankkeen operatiivinen ohjausryhmä	Hankkeen ohjausryhmän materiaalit	Toteutusprojektin ohjausryhmä	Muutospyynnöt	

Kuva 13. Valmis-hankkeen operatiivinen viikkorytmi. (Hankekäsikirja 2016.)

Toteutusprojektien näkökulmasta isompia ja säännöllisiä kokouksia ovat kerran kuukaudessa koko hankkeelle pidettävä hankekokous ja kahden viikon välein järjestettävä kehittäjien tilannepalaveri (Developer's status meeting). Hankekokouksessa käydään yleisesti läpi, mitä viimeisen kuukauden aikana on tehty ja mitä seuraavan kuukauden aikana on tarkoitus hoitaa. Kehittäjien tilannekatsauksessa keskitytään ainoastaan kehittäjien työhön ja käydään asioita paljon teknisemmin läpi kuin hankekokouksessa. Hankkeen operatiivinen ohjausryhmä ja toteutusprojektien ohjausryhmät kokoontuvat joka toinen viikko. Koko Valmis-hankkeen ohjausryhmä kokoontuu vain kerran kuukaudessa. Näiden kokouksien lisäksi joka kuukausi julkaistaan hanketiedote, joka on näkyvissä kaikille Valmis-hankkeessa työskenteleville ja kaikille Verohallinnon työntekijöille.

Tiimien edistymistä seurataan kahden viikon välein tehtävällä raportilla, jonka tiimin johtaja toimittaa projektipäällikölle. Raportista selviää, mitä viimeisen kahden viikon aikana on työstyetty ja saatu valmiiksi sekä mitä on tarkoitus tehdä seuraavan kahden viikon aikana. Projektipäällikkö toimitetaan hankkeen ja operatiivisen ohjausryhmän

materiaalit kokouksia varten. Projektipäällikölle kuuluu toimittaa myös viikoittain muutospyyntö, mikäli muutoksille on tarvetta.

Edistymisen seuranta ja palaverikäytännöt ovat hyvinkin läheisiä Scrum-mallin kanssa. Aikaisemmin puhuttiin, kuinka toteutusprojekteja voidaan verrata Scrumin sprintteihin. Yhtäläisyyksiä ilmenee myös verrattaessa palaveri- ja tiedotuskäytäntöjä. Voidaan ajatella, että yksi kuukausi Valmis-hankkeessa vastaisi sprintin yhtä vuorokautta. Scrumissa tilanpalaveri pidetään joka päivä ja Valmis-hankkeessa se pidetään kerran kuukaudessa hankekokouksen muodossa. Myös kehittäjien tilannekatsaus voidaan katsoa ikään kuin Scrumin päiväpalaverina, eli jokainen kahden kuukauden aikaväli vastaisi sprintin vuorokautta.

Kehittäjien tilannekatsaukset ovat olennaisia myös Lean-ajattelun kannalta, sillä niissä pyritään ehkäisemään turhaa työntekoa. Palavereissa kerrotaan esimerkiksi mahdollisista tulevista komponentti- tai käytäntömuutoksista. Näin niihin osataan varautua ajoissa ja turhalta kehitystyöltä vältytään. Palavereissa on myös kaikilla mahdollisuus jakaa havaintoja, vinkkejä tai mielipiteitä kehitystyöstä.

Kaikkea hankkeessa tuotettua dokumentaatiota hallitaan Valmis-hankkeelle perustetussa Microsoft Sharepoint -pohjaisessa työtilassa. Jokaisella osaprojektilla, vastuualueella sekä hanketoimistolla on omat työtilansa hankkeen työtilan sisällä. Yhteisellä työtilalla pyritään takaamaan dokumentaation saatavuus kaikille hankkeen henkilöille ajantasaisena. Lisäksi työtilat sallivat dokumenttien versioinnin sekä turvallisen säilytyksen.

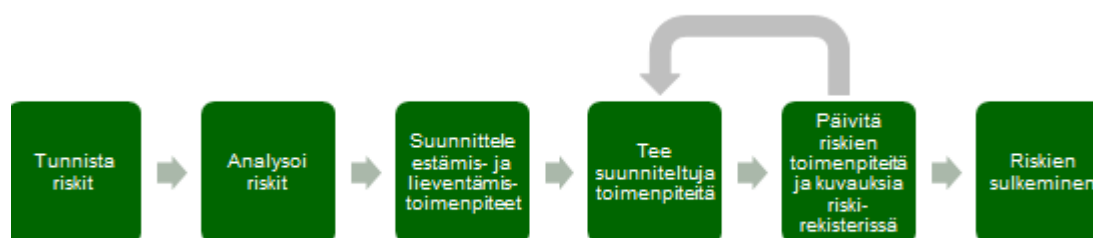
RUP-mallin suositukset on huomioitu vahvasti, sillä paperidokumentteja on erittäin vähän nähtävissä, jos ollenkaan. Toteutusprojekteissa dokumentointiin käytetään Sharepointin lisäksi Fastin omaa Fast Central Repository -projektinhallintaympäristöä. FCR sisältää projektia varten tehdyn alustan, joka sisältää jokaiselle rolloutille oman projektisuunnitelman. Projektisuunnitelma sisältää jokaiselle ohjelmassa työstettävälle osaluueelle oman kansion. Näihin kansioihin on tarkoitus merkitä kaikki kehitettävät toiminnallisuudet. Jokaiselle toiminnallisuudelle tai osalle on tarkoitus merkitä kehityksen aloituspäivämäärä ja arvioitu päättymispäivämäärä sekä dokumentoida sen toimintaa ja kehityksen edistymistä. Tehtävien edistymistä havainnollistetaan ja seurataan FCR:ssä olevan Gantt-kaavion avulla.

Dokumentointikäytännöt ovat hyvinkin monimuotoiset, ja niissä on hyödynnetty niin vesiputousmallia kuin Scrumiakin. Nämä kaksi näkemystä ovat kuitenkin selkeästi jakautuneet Verohallinnon ja Fastin välillä. Verohallinto luottaa enemmän vesiputousmallin kattavaan dokumentointikäytäntöön ja toivoo, että kaikesta, mistä vain dokumentaatio on mahdollista tehdä, se myös tehdään. Ketteristä menetelmistä on napattu lähinnä dokumentoinnin avoimuus. Sharepoint ja sen sisältävät dokumentit ovat nähtävissä kaikille Valmis-hankkeessa työskenteleville. Fast puolestaan luottaa ketterään työskentelytapaan. Tehdään työtä sen dokumentoinnin sijaan. Tästä syystä GenTaxin ja sen työkalujen dokumentaatio on rajallista. Tämä aiheuttaa toisinaan ongelmia kehittäjille. Eniten vaikeuksia tulee uusille kehittäjille, sillä he joutuvat turvautumaan suurimmaksi osaksi vain muiden kehittäjien neuvoihin.

Näiden kahden tavan yhdistäminen dokumentaatiossa on erittäin toimiva kokonaisuus, mikäli se saadaan resursoitua järkevästi. Dokumentoinnin on hyvä olla laajaa ja avointa, koska se tukee työntekijöiden tietoisuutta projektin sisällöstä ja tehdyistä päätöksistä. Tästä kaikesta huolimatta liiallinen ajankäyttö pelkkään dokumentointiin voi haitata kehitystyötä. Hyvin suunniteltu on puoliksi tehty, mutta jostain se toinenkin puoli on saatava.

3.4 Riskien hallinta

Valmis-hankkeen riskienhallintaprosessin tarkoituksena on kartoittaa hanketta tai projekteja vaanivat sudenkuopat, joihin työn edetessä voidaan langeta sekä suorittaa toimenpiteitä riskin toteutumisen ehkäisemiseksi tai toteutuneiden riskien mahdollisten vaikutusten pienentämiseksi. Valmis-hankkeeseen on luotu oma riskienhallintaprosessi, jota noudatetaan projekteissa (kuva 12). (Hankekäsikirja 2016.)



Kuva 14. Valmis-hankkeen riskienhallintaprosessin vaiheet. (Hankekäsikirja 2016.)

Ensimmäisessä vaiheessa tunnistetaan riskit. Riskien tunnistaminen suoritetaan ennen projektin alkamista. Tämän jälkeen riskit analysoidaan. Analysointiin kuuluu riskien vakavuuden ja todennäköisyyden arvioinnin lisäksi riskien tason arviointi. Tasoja on kolme: hanke, projekti tai vastuualue. Hanketason riskejä ovat lähinnä hankkeen aikataulua tai tavoitteita vaarantavat riskit. Riskien analysointia seuraa riskien estämis- ja lieventämistoimenpiteiden suunnittelu, jonka jälkeen niitä aletaan suorittaa. Toimenpiteitä ja riskien kuvauksia päivitetään tarvittaessa riskirekisterissä. Kun riskit on hoidettu tai niiden uhka on ohi, voidaan riskit sulkea. (Hankekäsikirja 2016.)

Kukin hankkeen projekti käsittelee ja kirjaa omat riskinsä. Riskit ja näitä koskevat toimenpiteet kirjataan hankkeen riskiluetteloon. Projektien riskit kirjataan projektien riskirekisteriin. Riskirekisteriä voivat päivittää kaikki projektiryhmän jäsenet. Riskejä hallitaan pääasiassa projektipäälliköiden toimesta, projektien ohjausryhmissä ja hankkeen operatiivisen ohjausryhmän kokouksissa. Riskienhallintasuunnitelman ja riskeihin liittyvien toimenpiteiden toteutumisen ohjaamisesta, seurannasta ja päivittämisestä hankkeen aikana vastaa hankkeen laatupäällikkö. (Hankesuunnitelma 2013.)

Jo pelkkä riskienhallinnan löytyminen poistaa yhtäläisyyksiä perinteisiin projektinhallintamenetelmiin; näistä se puuttuu lähes kokonaan. Ongelmiin on kuitenkin varauduttu satunnaisesti. Toteutusprojekteissa on olennaista pitää tuotanto sujuvana kehityksen ja testauksen ongelmista riippumatta. Tämä tukee erityisesti Scrumin näkemystä kehitystyöstä.

Kehitystyö on turvattu useilla eri varmuuskopioilla, joita otetaan päivittäin järjestelmän tietokannasta ja tilasta. Mikäli kehittäjä tekee virheen, joka vaikuttaa kaikkien työskentelyyn olennaisesti, voidaan järjestelmän tai tietokannan tila palauttaa aikaisempaan vaiheeseen. Lisäksi tuotantoympäristö pidetään erillään kehitysympäristöstä. Tämä pitää huolen siitä, että kehitysympäristössä tehtävät asiat eivät vaikuta tuotannon tilaan. Vaikka koko kehitysympäristö kaatuisi, tuotantoa voidaan jatkaa häiriöttä.

Kehitys- ja tuotantoympäristön lisäksi löytyy testausympäristö ja tietynlainen väliympäristö (Staging) tuotannon ja testauksen välillä. Staging-ympäristö on kopio tuotantoympäristöstä, ja se on ikään kuin kehittäjien ikkuna tuotantoympäristöön. Kehittäjät voivat käyttää sitä helposti tuotannon tuessa eivätkä siellä tehdyt toiminnot vaikuta varsinaiseen tuotantoympäristöön. Turhaa tekemistä ilmenee kuitenkin siinä vaiheessa, kun jotain täytyy viedä kehityksestä tuotantoon. Konfiguroinnit on ensin vietävä kehityksestä testausympäristöön, sieltä staging-ympäristöön ja vasta sieltä tuotantoon. Kehittäjät voivat viedä vapaasti muutoksia kehityksestä testaukseen, mutta ainoastaan tiimien johtajilla ja projektipäälliköllä on oikeus hyväksyä konfiguraatioiden viemistä testauksesta eteenpäin. Jos tätä riskienhallintaa ajattelee Leanin kannalta, saattaa turhaa tekemistä olla jonkin verran. Kehityksen ja tuotannon välinen matka on niin pitkä, että pahimmassa tapauksessa se saattaa haitata kehitystyötä tai kasvattaa sen määrää. Ketterien menetelmien kannalta prosessi ei oikeastaan tämän paremmaksi muutu. Ongelmat havaitaan jo varhaisessa vaiheessa ja niihin ehditään reagoimaan ajoissa. Mikäli tätä järjestelmää ei olisi, saattaisi jotkin ongelmat päätyä epähuomiossa tuotantoon asti.

Tämän ympäristökokonaisuuden lisäksi yksi olennaisista asioista toteutusprojektien riskienhallinnan ja laadunvarmistuksen kannalta on viimeinen kuukausi ennen käyttöönottoa. Tästä kuukaudesta käytetään termiä ”development freeze”. Tämä tarkoittaa sitä, että uusia ohjelmiston osia ei enää kehitetä, ainoastaan korjataan testauksessa ilmenneitä virheitä ja ongelmia. Tämän kuukauden tarkoituksena on minimoida virheet ja mahdollistaa koko vuoden aikana kehitetyn kokonaisuuden sujuva tuotantoon vieminen. Koska kaikkia virheitä ei tietenkään ole mahdollista eliminoida, käyttöönoton jälkeen ensimmäinen kuukausi tai pari on pyhitetty tuotannon tuelle. Resursseja käytetään enemmän loppukäyttäjien avustamiseen ja tuotannossa ilmenneiden virheiden korjaukseen, kuin täysin uusien toiminnallisuuksien kehittämiseen. Aivan kuten Lean-

talon toinen tukipilari, ohjataan oikeat ihmiset tekemään oikeita asioita oikeaan aikaan.
Priorisoidaan kehittäjät hetkellisesti keskittymään tuotantoon kehitystyön sijaan.

4 Johtopäätökset

Hyvä projektinhallinta on lähestulkoon elintärkeää projektin onnistumisen kannalta. On tärkeää suunnitella projekti hyvin ja luoda vahva pohja, jonka avulla ohjata projektin kulkua ja pitää se hallinnassa.

Valmis-hankkeen suuren henkilöstömäärän ja skaalan vuoksi projektinhallinta on erittäin tärkeässä roolissa. Näin suuren projektikokonaisuuden pitäminen kasassa on haastavaa.

Valmis-hankkeen projektimalli on hyvin monimuotoinen, ja se kehittyy koko ajan. Perinteisiä menetelmiä ja ketteriä menetelmiä käytetään sekaisin lähes jokaisella osalla. Hankkeen kahdella pääosapuolella on eri näkemykset tehokkaasta projektinhallinnasta ja suosivat siksi eri projektinhallintamenetelmiä. Tämä pakottaa työskentelymenetelmien jatkuvaan kehitykseen ja muovaukseen molempia miellyttäväksi.

Ennen toteutusprojektien ja kehitystyön aloittamista Valmis-hankkeessa vallitsi eräänlainen vesiputousmallin, Scrumin ja RUP:n sekasorto. Valmis-hanke on kuitenkin ensimmäisen toteutusprojektin myötä muuttunut. Yhä enemmän ja enemmän ollaan kallistumassa ohjelmistoprojekteille sopivampien ketterien menetelmien puolelle, perinteistä vesiputousmallia täysin unohtamatta. Tästä suuresta, useiden projektien kokonaisuudesta löytyy vielä paljon kehitettävää projektimallinsa ja projektinhallintansa puolesta, mutta alkuperäisestä on parannettu ja suunta on oikea.

Lähteet

Forsberg, K., Mooz, H. & Cotterman, H. 2003. Projektinhallinta:malli kaupalliseen ja tekniseen menestykseen. Edita

Haikala, I. & Mikkonen, T. 2013. Ohjelmistotuotannon käytännöt. Kariston Kirjapaino Oy.

Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. Talentum.

Hughes, B. & Cotterell, M. 2005. Software Project Management. McGraw-Hill Education.

Kasurinen, J. 2015. Ohjelmistotestauksen perusteet kalvosarja. [Luettu 18.3.2016]
<http://docplayer.fi/7192802-Ct60a4150-ohjelmistotestauksen-perusteet-jussi-kasurinen-etu-suku-lut-fi-kevat-2015.html>

Kruchten, P. 2004. The Rational Unified Process: An introduction. Pearson Education.

Kärenaho, J. 2016. Suuret ohjelmistoprojektit ja agile. [Luettu 5.4.2016].
<http://w3.fi/suuret-ohjelmistoprojektit-ja-agile/>

Lean Enterprise Institute. 2016. What is Lean? [Luettu 18.3.2016]
<http://www.lean.org/WhatsLean/>

Lester, A. 2007. Project management: planning and control. Butterworth-Heinemann Ltd.

Litke, H. & Kunow, I. 2004. Projektinhallinta. Rastor.

Malmi, P. & Åkerlund, K. 2013. CPM Creative Project Management. PLUS Akatemia Oy.

Mikkelin ammattikorkeakoulu. 2013. Projektiosaaminen. [Luettu 13.4.2016]
cna.mikkeliyamk.fi/Public/KosonenH/projekti_aikuis_2013/Peruskalvosarjat/PROJEKTIO SAAMINEN_MAMK.pptx

Mikko Jokela. 2011. Lean tuotekehitys. [Luettu 5.4.2016]
<http://inside-the-plm.blogspot.fi/2011/11/lean-tuotekehitys.html>

Murch, Richard. 2002. IT-projektinhallinta. IT Press. Edita Prima Oy

Projekti-instituutti. 2016. Liiketoiminta ja projektit. [Luettu 16.3.2016].
http://www.projekti-instituutti.fi/ratkaisut/liiketoiminta_ja_projektit

Projekti-instituutti. 2016. Projektijohtamisen sanastoa. [Luettu 16.3.2016].
http://www.projekti-instituutti.fi/ratkaisut/liiketoiminta_ja_projektit

Pöyry. 2016. Riskienhallinnan järjestelmä. [Luettu 12.4.2016]
<http://www.poyry.com/fi/sijoittajat/hallinnointi/riskienhallinta-ja-kontrolli/riskienhallinnan-jarjestelma>

Ruuska, K. 2007. Pidä projekti hallinnassa - suunnittelu, menetelmät, vuorovaikutus. Talentum.

Schwaber, K & Sutherland, J. 2013. The Scrum Guide. [Luettu 17.3.2016]
<https://scrumwell.files.wordpress.com/2014/03/scrum-guide-2013-fi-v1-1.pdf>

Scrum Alliance. 2016. Learn About Scrum. [Luettu 17.3.2016]
<https://www.scrumalliance.org/why-scrum>

Sininen Meteoriiitti. 2013. Ketteryys haltuun: Scrum pähkinänkuoressa. [Luettu 17.3.2016]
<http://www.meteoriiitti.com/2013/06/06/ketteryys-haltuun-scrum-pahkinankuoressa/>

Suntuubi. 2016. Projektin rooleja. [Luettu 10.4.2016]

<http://hybridimenetelma.suntuubi.com/?cat=15>

Vahtolampi, K. 2010. Esipuhetta ketteristä menetelmistä. [Luettu 17.3.2016]

<https://karivahtolampi.wordpress.com/2010/12/11/esipuhetta-ketterista-menetelmista/>

Taina, J. 2009. Ohjelmistoprosessit ja ohjelmistojen laatu – luentokalvot. [Luettu 16.3.2016].

http://www.cs.helsinki.fi/u/taina/opol/k-2009/pdf/luku-6_2.pdf

Valmis-hankkeen hankesuunnitelma. 2013.

Dokumentti Valmis-hankkeen intranetissä

Valmis-hankkeen hankekäsikirja. 2016.

Dokumentti Valmis-hankkeen intranetissä

Valtiovarainministeriö. 2015. Valmis-hanke ja verotusmenettelyjen uudistaminen. [Luettu 18.3.2016]

<http://vm.fi/documents/10623/1844434/Valmis-hanke+ja+verotusmenettelyjen+uudistaminen/1b224647-bbec-4c09-9491-a6c960c22bec>

Wikipedia. 2016. Vesiputousmalli. [Luettu 16.3.2016]

<https://fi.wikipedia.org/wiki/Vesiputousmalli>