

Kanban-metodi ja Jira-tehtävähallinta jatkuvassa ohjelmistokehitys- ja ylläpitoprojektissa

Teemu Sirviö



Tekijä(t) Teemu Sirviö	
Suuntautuminen Tietojärjestelmäosaamisen koulutusohjelma, YAMK	
Opinnäytetyön nimi Kanban-metodi ja Jira-tehtävähallinta jatkuvassa ohjelmistokehitys- ja ylläpitoprojektissa.	Sivumäärä + liitesivumäärä 46 + 19
<p>Tämän opinnäytetyön kehitystehtävänä oli toimeksiantajan 10 hengen ohjelmistokehityksen ryhmän prosessin parannus. Ryhmän tehtävien läpimenoajoissa on huomattavaa vaihtelua ja Kanban-metodin avulla tavoitteena on selvittää ryhmän nykyiset työtavat ja visualisoida ne kanban-taululle, minkä avulla on tarkoitus löytää prosessia hidastavia pullonkauloja ja parantaa ryhmän toimintaa.</p> <p>Kanban-metodi on esimerkiksi Scrumiin verrattuna helppo tapa lähestyä ketterän kehityksen ajatusmaailmaa, sillä se ei aluksi vaadi muutoksia ryhmän työtapoihin tai rooleihin. Siinä otetaan lähtökohdaksi ryhmän nykyiset työtavat ja niiden havainnollistaminen visuaalisesti taululla, joka on jaettu työvaiheita vastaaviin sarakkeisiin. Metodissa työn prosessia kehitetään kokeilevasti ja pienin muutoksin.</p> <p>Työ toteutettiin toimintatutkimuksena. Kanban-metodia sovellettiin ryhmän käytössä olevan Jira-tehtävähallinnan projektiin ja sen tarjoamiin keinoihin visualisoida työtä. Samoin Jiran tarjoamista erilaisista raporteista haettiin ryhmän työtehoa arvioivia kuvaajia ja mittaristoja.</p> <p>Ryhmän nykyiset työtavat selvitettiin pienoistyöpajoissa arvovirtakuvauksilla, jotka edelleen mallinnettiin tilakaavioiksi ja aina edelleen Jira-ohjelmiston asetuksiin. Ryhmän Jira-projektissa käytössä olevat tehtävätyypit ja niiden käyttämät työvuot päivitettiin selvityksen tulosten mukaisesti.</p> <p>Ryhmän käyttöön otettiin lyhyet päivittäiset tapaamiset, joiden avulla ryhmän sisäinen yhteistyö ja kommunikaatio koheni, ja tapaamisia on tarkoitus jatkaa myös tästä eteen päin. Kanban-metodi vaikuttaa hyödylliseltä ja helpolta ottaa käyttöön, mutta Jira-ohjelmiston kankeus ja hankala muokattavuus tuo rajoitteita sen soveltamiseen ryhmässä.</p>	
Asiasanat Kanban-metodi, Jira, tehtävähallinta, Daily, lean, ketterä.	

Author(s) Teemu Sirviö	
Degree program Degree Programme in Information Systems, master's level	
Title Kanban method and Jira Issue Tracker in an ongoing software development and maintenance project.	Number of pages and appendixes 46 + 19
<p>The development project for this thesis was improving the process of the software development team at the client organization. There is a lot of variation in lead times in the groups assignments and using Kanban-method the goal was to identify and visualize the development team's workflow to find bottlenecks and improve the team's work process.</p> <p>Kanban method, compared to for example Scrum, is an easy way to approach agile mindset in software development, because it requires no changes in the team's work methods or roles at first. As a starting point the method takes the current workflow and visualization of the workflow on a task board that is separated into columns describing each state in the workflow. Using the Kanban method, the work process is improved experimentally and with small changes.</p> <p>The project was implemented as action research. Kanban method was implemented using the Jira issue tracker and its capabilities to visualize the work. Diagrams and meters for evaluating team's work efficiency were searched from the different reports available in Jira.</p> <p>Development team's current workflow was traced using value-stream mappings that were created in small workshops. The mappings were remodeled as state diagrams and further as workflow settings in Jira software. Issue types and their workflows in team's Jira project were updated according to results.</p> <p>The development team started to have daily meetings for improving teamwork and communication, and the daily meetings are to be continued after the project ends. Kanban method seems useful and easy to use but it's implementation for the team's use was restricted by the use of Jira which in parts is cumbersome and rigid.</p>	
Keywords Kanban method, Jira, issue tracking, Daily, lean, agile.	

Sisällys

1	Johdanto	1
2	Tavoitteet	3
2.1	Odotetut lopputulokset.....	3
2.2	Tutkimuskysymykset.....	4
2.3	Rajaus.....	4
3	Toimintaympäristö.....	6
3.1	Nykyinen työnkulku.....	6
3.2	Jira ja Kanban ryhmän toiminnassa.....	8
4	Tietoperusta	10
4.1	Ketterä ohjelmistokehitys.....	10
4.2	Kanbanin periaatteet ja käytännöt	11
4.3	Metodin käyttöönotto	12
4.4	Keskeneräisen työn rajat	15
4.5	Mittaristot Kanban-metodissa	17
4.6	Lupa kokeilevaan kehittämiseen.....	19
4.7	Palautteen tärkeys	20
4.8	Tehtävävaraston ylläpito.....	21
4.9	Yhteenveto.....	22
5	Menetelmät	23
5.1	Työn visualisointi arvovirtakuvauksin.....	24
5.2	Jira-projektin kanban-taulun päivitys	25
5.3	Työvuon tehokkuuden mittaaminen Jiran avulla.....	25
5.4	Aikarajattu kehitys.....	25
5.5	Päivittäiset tapaamiset.....	26
5.6	Palautteen kerääminen sidosryhmiltä kehittämistyön tuloksista.....	27
6	Toteutus ja tulokset.....	29
6.1	Arvovirtakuvaukset	29
6.2	Työvuot ja tilakaaviot	30
6.2.1	Sub-taskin työvuot	33
6.2.2	On Hold -status.....	33
6.3	Jiran päivitys	33
6.4	Kanban-taulujen päivitykset.....	34
6.5	Mittaristot	36
6.6	Ketterien metodien käyttö ryhmässä	39
6.7	Ehdotukset kehittämistoimenpiteiksi.....	40
7	Johtopäätökset.....	42

7.1 Saavutetut tulokset	42
7.2 Vastaukset tutkimuskysymyksiin	43
7.3 Pohdinta.....	44
7.4 Omakohtainen reflektio työn tekemisestä.....	46
Lähteet.....	47
Liitteet	1
Liite 1. Arvovirtakuvaukset	1
Liite 2. Ehdotetut työvuot	11
Liite 3. Jira-tikettien tilojen uudelleenjärjestäminen.....	13
Liite 4. Työvuoaasetukset	16

1 Johdanto

Tämän ylemmän ammattikorkeakoulututkinnon kehitystehtävänä on Kanban-metodia käyttämällä parantaa toimeksiantajan ohjelmistokehitystä tekevän ryhmän työtä. Kymmenen hengen ryhmä kehittää ja ylläpitää lukuisia verkkopalveluita ja tuotantoja, sekä sisäiseen käyttöön, että ulkoisille asiakkaille ja yksityishenkilöille. Toimeksiantajan strategiaan on kirjattu vuosille 2020–2025 tavoitteeksi se, että tuotantojärjestelmät olisivat luotettavia ja nykyaikaisia. Ketterän kehityksen työtapojen ottaminen ryhmän käyttöön voidaan nähdä sopivan tähän strategiseen tavoitteeseen.

Vuonna 2013 tehdyn State of Agile Development -kyselyn vastaajista 88 % kertoi käyttävänsä ketteriä kehitysmetodeja ja 92 % kertoi vuosittaisista parannuksista kaikissa kyselyn kategorioista, joista erityisesti mainittiin kyky hallita vaihtuvia prioriteetteja, kasvanut tuottavuus, parantunut projektien näkyvyys, parantunut tiimin moraalit ja ohjelmistojen laatu (Greene & Stellman 2014, luku 2). Vuonna 2019 Analysis.Net Researchin toteuttamassa tutkimuksessa jo 95 % vastaajista, kertoi käyttävänsä ketteriä metodeja. Vastausten mukaan suurin syy ketterien menetelmien käyttöönotossa oli nopeutuva ohjelmistojen toimittaminen. Kyselyn tuloksissa ketterien menetelmien hyötyinä oli yleisimmin valittuna kyky hallita vaihtuvia prioriteetteja, projektien näkyvyys, liiketoiminnan ja IT:n kohdistaminen, toimitusnopeus ja tiimin moraalin koheneminen. (2020, 8–10). Vaihtuvien prioriteettien hallinta oli mainittu tärkeänä molempien kyselyiden tuloksissa ja myös ohjelmistokehityksen ryhmässä tehtävien tärkeysjärjestys muuttuu usein ja niiden hallinta on monimutkaista.

Ryhmän työnkuva on laaja ja töitä tehdään lukuisiin eri projekteihin sisäisten ja ulkoisten sidosryhmien kanssa. Työtehtävät ovat monipuolisia ja niiden hoitamiseen vaadittavat työvaiheet vaihtelevat paljon tehtävästä riippuen. Pääasiassa uudet tilaukset tulevat ryhmälle Jira-tehtävänhallintaohjelmiston kautta, mutta tämän lisäksi uusia tehtäviä syntyy myös erilaisten pikaviestisovellusten, sähköpostin, puheluiden ja käytäväkeskusteluiden kautta. Asiakastilausten lisäksi ryhmän työhön kuuluu myös ylläpitotehtävät, jotka syntyvät yleensä ryhmän omista tarpeista tai toisen, tuotantojärjestelmiä ylläpitävän ryhmän toiveesta. Ryhmän itse ylläpitämiä ja käyttämiä ohjelmistoja ja järjestelmiä on säännöllisesti päivitettävä ajantasaisiksi esimerkiksi tietoturvan vuoksi.

Asiakastilausten toimitusajaksi on aiemmin luvattu yksi kuukausi tilauksen saapumisesta, mutta tähän tavoitteeseen ei aina ole päästy, ja toimitusajan lupaus on sittemmin kasvatettu kolmeen kuukauteen. Pitkäksi venyneiden tilausten lisäksi myös sisäisen kehityksen tehtävät saattavat samalla tavalla jäädä työjonoon tai keskeneräisiksi pitkiksi ajoiksi (jopa useiden vuosien ajaksi), ja tilanne kaipaa parannusta. Töiden hallinta ja

seuranta on ollut haasteellista, ja pientenkin töiden läpimenoaikaa mitataan joskus päivien sijaan kuukausissa. Kanban-metodin avustamana on tavoitteena selvittää ryhmän työtavat ja parantaa työnohjausta hallitsemalla työtehtävien virtaa.

Opinnäytetyöraportin toisessa luvussa käydään tarkemmin läpi työn tavoitteet, tutkimuskysymykset sekä työn rajaus. Kolmannessa luvussa esitellään tarkemmin kehitystehtävän kohteena olevan ryhmän toimintaympäristöä. Neljännessä luvussa kerrotaan tarkemmin kehitystehtävässä sovellettavasta Kanban-metodista. Viidennessä luvussa kerrotaan työhön valituista toimintatutkimuksen menetelmästä, ja kuudennessa luvussa kuvataan työn eteneminen ja saadut tulokset. Seitsemännessä luvussa verrataan saavutettuja lopputuloksia odotettuihin, vastataan esitettyihin tutkimuskysymyksiin ja esitetään johtopäätökset ja jatkopohdinta.

2 Tavoitteet

Tässä luvussa kerrotaan tarkemmin opinnäytetyön odotetut lopputulokset sekä esitetyt tutkimuskysymykset, joihin työssä haetaan vastausta, sekä rajataan opinnäytetyö.

2.1 Odotetut lopputulokset

Tavoitteina kehitystyössä on työnkulun nykytilan ja ongelmakohtien selvittäminen Kanban-metodin avulla, sekä ketterän kehityksen päivittäisten tapaamisten käytännön kokeileminen. Ryhmän Jira-projektissa olevat työtehtävät eivät kaikki noudata samaa määriteltyä työvuota eikä tehtävien kuljettaminen kanban-aulun sarakkeiden välillä kuvasta oikein ryhmän työtapoja. Töiden seurantaan ei ryhmässä ole säännöllisesti käytetty Jiran tarjoamia erilaisia kuvaajia ja raportteja, eikä kanban-aulua ole käytetty metodin kuvaamalla tavalla rajoittamaan keskeneräisen työn määrää. Tilanteeseen haetaan parannusta kehitystehtävän myötä.

Tarkemmin kehittämistyöllä on neljä odotettua käytännön lopputulosta:

1. Ryhmän käytössä olevan Jira-projektin toiminnan tulee vastata ryhmässä toteutuvaa toimintaa.
2. Käytettävissä on Jiran kanban-aulu, joka kuvastaa ryhmän todellista työtilannetta ja toimii ajantasaisena tiedonlähteenä myös muille organisaation sisäisille sidosryhmille.
3. Ryhmän käytössä on Jirassa mittaristo, jolla voidaan objektiivisesti mitata ryhmän työtehoa tehtävien läpimenoaikojen muodossa.
4. Ryhmä käyttää kanban-aulun keskeneräisen työn rajoituksia hallitsemaan työvirtaa.

Jiran toiminnallisuudet on tarpeellista saada vastaamaan ryhmän työtapoja, jotta sitä voidaan käyttää tukemaan työntekoa, eikä sen käyttö saisi hankaloittaa tai viivästyttää töitä. Jiran käyttö on keskeisessä roolissa ryhmän työssä ja siitä on syytä ottaa irti kaikki sen tarjoama hyöty, mutta myös vähentää sen käytöstä aiheutuvia harmeja mahdollisimman paljon. Jiran kanban-aulua voitaisiin käyttää tehokkaammin, jos se kuvastaisi ryhmän työtapoja ja -tilannetta, jolloin myös muut käyttäjät voisivat hyödyntää näkymää ryhmältä tilaamiensa töiden seurannassa. Tehtävien kasaantumista eri työvaiheisiin voidaan yrittää hallita rajoittamalla niiden samanaikaisten tehtävien lukumäärää, ja tämä voidaan toteuttaa esimerkiksi kanban-aululle asetettävien rajoitusten muodossa.

2.2 Tutkimuskysymykset

Tutkimuskysymykset, joihin tällä opinnäytetyöllä halutaan löytää vastaukset ja joihin vastataan luvussa 5, ovat:

TK1 Minkälainen työvuo on nykyisin ohjelmistokehitysryhmän työtehtävillä?

TK2 Miten ryhmän laajojen työtehtävien prosessia voidaan mallintaa Jiran kanban-
taululle, niin että se toimisi informatiivisena lähteenä ryhmän työtilanteesta?

TK3 Mikä on nykyisten työtehtävien läpimenoaika ja mitkä asiat hidastavat läpimenoa?

TK4 Voidaanko Kanban-metodin, aikarajatun kehityksen ja päivittäisten tapaamisen avulla parantaa ryhmän työtehoa?

Ohjelmistokehityksen ryhmällä on oma kirjoittamaton työvuonsa jokaiselle työtehtävälle, ja jokainen niistä tulisi saada nähtäville kanban-taululle. Ryhmän Jira-projektissa on toistaiseksi ollut vain yksi työvuoa-asetus määrittämässä kaikkia ryhmän työtehtäviä, mikä hankaloittaa tehtävien visualisointia ja toisaalta ei välttämättä sovi kaikkiin käyttötapauksiin.

Työtehtävät saattavat tehtävätyypistä ja työntekijästä riippuen edetä työvaiheesta toiseen eri tavoin. Tarkoitus on löytää ryhmän työstä ne punaiset langat, joita voidaan käyttää yhteisten toimintatapojen määrittämiseen myös Jira-ohjelmistossa. Osassa tehtävistä työn eteneminen toisinaan pysähtyy, eikä pysähdyn syy ole aina ilmeistä. Tehtävillä voi olla riippuvuuksia toisten organisaation sisäisten tai ulkoisten sidosryhmien toimintaan ja syitä töiden hidastumisille on lukuisia. Kanban-metodin avulla toivotaan saatavan pullonkauloja paremmin näkyville, jotta niihin voitaisiin etsiä ratkaisua.

Tehtävien valmistumiseen kulunutta aikaa ei olla säännöllisesti ryhmän sisällä seurattu, vaikka tehtävänhallinnasta löytyy useita erilaisia raportointityökaluja ajankäytölle. Käytettävissä olevista kuvaajista toivotaan helppokäyttöistä seurantatyökalua, joka auttaisi ryhmää tunnistamaan ongelmakohtia ja toisaalta seuraamaan keskeneräisen työn rajoitteiden vaikutusta läpimenoaikoihin.

2.3 Rajaus

Tämän opinnäytetyön kehitystehtävä rajataan koskemaan toimeksiantajan organisaatiossa ohjelmistokehityksen ryhmän toimintaa, ryhmän käytössä olevaa tiettyä Jira-tehtävänhallinnan versiota ja ketteristä menetelmistä Kanban-metodia.

Vaikka osa ryhmän työstä tehdään yhteistyössä muiden organisaation yksiköiden tai ryhmien tai muiden kansainvälisten tahojen kanssa, tämä opinnäytetyö on rajattu koskemaan vain ohjelmistokehityksen ryhmän toimintaa. Kiinnostusta Kanban-metodin käyttöön on muissakin yksiköissä, ja olisi mahdollista laajentaa arvovirtakuvaukset ja työvuon visualisointi koskemaan myös ulkoisia sidosryhmiä. Tämä ei kuitenkaan ole osa kehitystehtävää, mutta sopiva kohde jatkokehitystä ajatellen.

Opinnäytetyössä tutkitaan ja kokeillaan Kanban-metodin käyttöä. Vaikka Kanban-metodi lasketaan osaksi erilaisia ketteriä menetelmiä, kehitystehtävän tavoitteena ei ole ottaa käyttöön jotain valmiista ketterän kehityksen metodologiaa kuten Scrum tai XP. Opinnäytetyön tavoitteena ei myöskään ole tarkoitus saada ryhmän työvuolle erityistä ketterän ohjelmistokehityksen leimaa, vaikka Greene ja Stellman ovatkin sitä mieltä, että tiimi voi olla ketterä myös ilman Scrumia tai Kanbania (2014, luku 2).

Ryhmän käytössä on yhdessä muun organisaation kanssa käytössä Jira-tehtävähallintaohjelmisto, jota ryhmässä on pääasiassa käytetty töiden seurantaan. Ryhmä on mukana myös projekteissa, joissa tehtävähallintaan käytetään muun muassa versionhallintasivusto GitHubin tarjoamaa tehtävähallintaa, mutta tässä opinnäytetyössä keskitytään Kanban-metodin käyttöön nimenomaan Jiran ympäristössä, joka on käytössä muissakin eri yksiköissä ja ryhmissä organisaation sisällä.

Jiran osalta rajaus tehdään myös erityisesti käytössä olevaan paikallisen asennuksen versioon, erotuksena pilvipalveluna tarjolla olevasta Jirasta. Pilvipalveluna tarjottava versio tulee olemaan tulevaisuudessa ainoa Jirasta tarjolla oleva versio, ja paikallisesti hallittavan version tuki päättyy vuonna 2024 (Atlassian 2021). Opinnäytetyötä tehdessä ei kuitenkaan ollut vielä tiedossa kuinka paljon vääjäämätön siirros pilveen tulee vaikuttamaan Jiran käyttöön, tai onko Jira organisaation käytössä vielä vuoden 2024 jälkeen.

3 Toimintaympäristö

Tässä luvussa kerron minkälaisessa toimintaympäristössä kehitystehtävä toteutetaan, minkälainen työn kulku nykyään on ja kuinka Jiran tehtävähallintaa on käytetty.

Ohjelmistokehityksen ryhmä toimii toimeksiantajan asiakaspalveluista vastaavassa yksikössä, mutta on vuoden 2022 alussa organisaation muutoksista johtuen siirtymässä toiseen yksikköön. Ryhmän vastuulla on lukuisten verkkopalveluiden, ohjelmistojen ja tuotteiden kehitys ja ylläpito, ja sen tehtävät ovat pääasiassa ohjelmointi- ja suunnittelutyötä. Osa ryhmän työstä tehdään yksityisille yrityksille tai viranomaisille, joihin kuuluu lukuisien eri toimialojen edustajia. Ryhmän työssä vaaditaan usein yhteistyötä muiden organisaation yksiköiden ja ryhmien kanssa, ja osa töistä tehdään kansainvälisiin projekteihin. Työtehtävien laajuus ja vaativuus vaihtelee pienestä asetustiedoston muutoksesta yli vuoden mittaisiin projekteihin. Ryhmän työhön kuuluu myös tietojärjestelmien ylläpito niiltä osin, kuin ne ovat ryhmän vastuulla.

Ohjelmistokehityksen ryhmän yksikössä on omat toimialakohtaiset ryhmät, joiden yhteyspäälliköt toimivat tiedonvälittäjinä asiakkaiden ja kehittäjien välillä. Asiakkaiden tekemät tilaukset tai ongelmat tulevat ryhmän tietoon pääasiassa yhteyspäälliköiden kautta. Tiedonkulku toiseen suuntaan, ohjelmistokehityksestä asiakkaalle, toteutetaan myös yleensä yhteyspäälliköiden välityksellä. Huolimatta tulevasta ryhmän yksikön muutoksesta, tähän toimintaan ei ole ollut toistaiseksi ilmoitettu olevan muutoksia.

Yksi ryhmän työntekijöistä kiertävässä vuorossa vastaa toimistoaikaan ongelmanratkaisun tehtävien vastaanottamisesta ja selvityksestä, sekä vastaa ryhmälle saapuviin tiedusteluihin ja on tämän lisäksi toimistoaikojen ulkopuolella puhelinvalmiudessa. Vuoron aikana on hankala keskittyä tekeillä oleviin tilauksiin, sillä satunnaisten ja säännöllisten ylläpitotehtävien ohella ongelmanratkaisu ja kyselyihin vastaaminen vievät suurimman osan työajasta. Kiertävä vuoro otettiin käyttöön alun perin juuri sen takia, että tällöin muut ryhmän jäsenet voivat paremmin keskittyä kehitteillä oleviin tilauksiin.

3.1 Nykyinen työnkulku

Töiden hallintaan käytetään pääasiassa Atlassianin kehittämää Jira-tehtävähallintaohjelmistoa. Sama ohjelmisto on myös muun organisaation käytössä ja siinä on ryhmän käyttöön erikseen nimetty alue, josta ohjelmistossa käytetään nimitystä projekti. Osa töistä tehdään eri ryhmien välisenä yhteistyönä, jolloin tehtävät on saatettu kirjata johonkin toiseen Jira-projektiin. Joissain tapauksissa tehtävähallinta on toteutettu kokonaan eri järjestelmään. Esimerkiksi silloin, jos tehtävien seurantaan tarvitaan

näkyvyys myös sisäverkon ulkopuolelta, tai jos ohjelmakoodia on tarpeen jakaa ulkopuolisten toimijoiden kanssa. Tällaisissa tapauksissa on käytetty esimerkiksi versionhallintapalvelu GitHubin tarjoamaa tehtävänseurantaa.

Tilaukset ja tiedustelut ja myös osa ongelmaraporteista tulevat ryhmälle yleensä yhteyspäälliköiden kirjaamina Jira-projektiin. Toisinaan – erityisesti ylläpitotehtävien tapauksessa – tehtävät kirjataan Jiraan ryhmän itsensä toimesta. Joskus tehtävien kuvaukset ovat selkeitä tai mahdolliset ongelmat on kuvattu yksiselitteisesti. Usein ne jättävät kuitenkin toivomisen varaa, ja kun tilauksesta ei käy ilmi asiakkaan tarve, tai sen perusteella ei ole mahdollista arvioida työn laajuutta, on asioiden tolan selvittämiseksi tarpeen pitää yhteyspäällikön kanssa palaveri.

Osa kehitystehtävistä voi vaatia erityisen tarkan teknisen määrittelyn ennen töiden aloitusta. Joidenkin toimialojen projekteihin liittyvistä kehitystehtävistä on tarkasti määrätty, kuinka muutokset tulee toteuttaa ja minkälaiset määrittelyt ennen työn suoritusta vaaditaan. Tästä huolimatta myös näissä tilauksissa voi olla epäselvyyttä, sillä niihin liittyy usein alakohtaista terminologiaa, jota ei säännöllisesti tarvita ohjelmointityössä. Jotta kehittäjät voisivat suunnitella asiakastarvetta parhaiten vastaavat tekniset ratkaisut ja arvioida toteutukseen vaaditun työmäärän, pitää asiakkaan tarpeet ymmärtää selkeämmin.

Organisaation tuottaman ja käyttämän datan määrä on suurta. Eri datalähteitä on käytössä useita satoja, ja niistä jokaiseen voi liittyä useita lukuisia eri parametreja. Kun käytettävissä on dataa lukuisista eri lähteistä, useilta eri mittausinstrumenteilta, ja jota voidaan visualisoida lukemattomilla eri tavoilla, on tarpeellista tehdä täsmällisiä tilauksia. Liian epämääräiset pyynnöt aiheuttavat ryhmässä joskus turhautumista, ja kehittäjien pitää pyytää yhteyspäälliköltä tarkennuksia, jotka olisi alun perin voitu kirjoittaa tilauksen tietoihin. Voi myös olla, ettei asiakkaalla ole alun perinkään tarkkaa käsitystä omista tarpeistaan, tai jos toimituksessa on kestänyt kovin pitkä aika ovat tarpeet saattaneet ehtiä muuttua. Nopeat toimitusajat ja iteratiivinen kehittäminen voisivat vähentää kehittäjien turhautumista puutteellisiin määrittelyihin ja parantaa asiakastytyväisyyttä. Jos uudet ominaisuudet saataisiin asiakkaan käyttöön useammassa, pienissä osissa, kehittäjät voisivat saada palautetta nopeammin, eikä kehitystyö ehtisi etenemään väärään suuntaan.

Saapuville työtehtäville nimetään tekijät yleensä ryhmän viikoittaisen palaverin aikana. Palaverissa esiin nostetut tehtävät ovat yleensä sellaisia, joista yhteyspäälliköt ovat maininneet ryhmäpäällikölle, jotta ne saataisiin nopeammin työn alle. Ilman nimeämistä kehittäjille ei ole yleensä selvää mitä Jira-projektin tiketeistä pitäisi seuraavaksi ottaa

käsittelyyn. Tämän lisäksi osa työtehtävistä syntyy kehittäjille tehdyistä henkilökohtaisista pyynnöistä, joita tulee esimerkiksi sähköpostitse tai pikaviestisovellus-Slackin välityksellä tai käytäväkeskusteluiden kautta. Näihin tehtäviin kulutettu aika jää usein piiloon, ja ne saattavat ohittaa valmiiksi kiireellisemmäksi priorisoidut työtehtävät.

Asiakastilausten lisäksi ryhmä tekee myös paljon sisäistä kehitystä ja siihen liittyviä ylläpitotehtäviä. Organisaatiossa on käytössään useita eri sisäisiä ja ulkoisia rajapintoja, ja näiden taustalla toimivat ohjelmistot vaativat usein säännöllistä päivitystä. Joskus muutoksia tehdään tilausten perusteella, mutta useimmiten päivitystarpeita syntyy sisäisen kehityksen tuloksena, jolloin ne lasketaan ylläpitotehtäviksi.

Eri ohjelmistojen säännöllisten päivitysten lisäksi käynnissä on useita projekteja, joilla on tarkoitus modernisoida ryhmän tarjoamien palveluiden taustajärjestelmiä. Ryhmän ohjelmistokehityksellä on pitkä historia, ja sen käytössä on vielä paljon eri ohjelmointikielillä toteutettuja komentosarjoja, joista iso osa on tarkoitus korvata esimerkiksi tarjolla olevilla rajapinnoilla. Tuotantojen määrä on kuitenkin suuri, eikä aivan kaikkea ole tarkoituksenmukaista uusia, ja osa on mahdollista lopettaa kokonaan. Erityisesti vanhojen komentosarjojen ylläpito voi olla välillä työlästä, varsinkin jos niiden alkuperäisiä tekijöitä ole enää saatavilla.

Organisaation toisen, tuotantojärjestelmistä vastaavan ryhmän tekemät päivitykset tietotekniseen infrastruktuuriin aiheuttavat ohjelmistokehityksen ryhmälle ylläpitotöitä. Eri palvelinkoneita joudutaan sammuttamaan ja käynnistämään toisinaan uudelleen, ja käynnissä olevia tuotantoja joudutaan siirtämään siksi aikaa toisille laitteille. Aina se ei suju ongelmitta, ja joskus muutoksista voi aiheutua ryhmälle töitä useiksi päiviksi.

3.2 Jira ja Kanban ryhmän toiminnassa

Ryhmän Jira-projektissa on käytössä erilaisia tehtävätyyppejä erottamaan toisistaan asiakastilaukset (Feature Request), korjattavat ongelmat (Problem Report ja Reclamation), ryhmälle esitetyt tiedustelupyynnöt (Request for Information) ja ylläpidolliset työt (Maintenance). Projektiin liitetystä tehtävistä käytetään ryhmässä yleisesti nimeä tiketti, ja niillä voi olla myös alitehtäviä (Sub-task). Pitkään kestäviä tehtäviä varten voidaan myös luoda Epic-tyyppinen tiketti, jonka alle tikettejä alitehtävineen voidaan kerätä. Tämän lisäksi projektissa on vielä kehitysideoita varten oma Development Idea -tehtävätyyppi, joihin on kirjattu ajatuksia mahdollisista tulevista kehityskohteista. Tehtävätyyppien määrää on vähennetty aiemmasta, ja vielä ennen vuotta 2019 esimerkiksi asiakastilaukset oli jaettu edelleen neljään eri tyyppiin työn laajuuden ja luonteen suhteen.

Vaikka ryhmän käytössä onkin Jiran kautta kanban-taulu, ei Kanban-metodi varsinaisesti ole ryhmälle tuttu. Kanban-metodi (josta erotuksena pienellä alkukirjaimella kirjoitettu kanban-taulu) on tapa visualisoida ryhmän työtä ja tehdä jatkuvan parantamisen kautta työvuosta toimivampi. Käytössä olevan kanban-taulun rakennetta on ajoittain muutettu, ja sopivia sarakkeiden nimiä ja työvaiheita on etsitty ja lainattu internetistä, mutta ryhmän työtavat ovat pääosin pysyneet samoina. Ryhmässä ei erityisesti seurata ketterän kehityksen periaatteita, eivätkä ketterän kehityksen käytännöt ohjaa ryhmän työtä tai työn suunnittelua. Nykyinen työvuoto ei noudata mitään nimettyä metodologiaa, vaan on muovautunut nykyiselleen ajan saatossa.

Ryhmän työtilanteesta kokonaisuuden hahmottaminen – mitkä tehtävät odottavat tekemistä, mitä on tekeillä ja mitä on saatu valmiiksi – on jo yhden Jira-projektin laajuudessa hankalaa, vaikka huomiotta jättäisi muihin projekteihin tai tehtävähallintaohjelmistoihin merkityt työt. Jira-projektissa on kanban-tilauja, joita on käytetty visualisoimaan asiakastilauksia, ratkaistavana olevia ongelmia, tai sitten taululla on voitu esittää johonkin tiettyyn isompaan kokonaisuuteen liittyviä tehtäviä Epic-tikettien avulla. Jirassa ei ole näkymää, josta saisi helposti kuvan koko ryhmän todellisesta työtilanteesta, ja ryhmällä on usein samanaikaisesti kesken niin paljon tehtäviä, ettei niitä yksinkertaisesti mahdu esittämään samanaikaisesti tietokoneen näytöllä.

Ongelmana työn visualisoinnissa on myös se, että erityyppiset työtehtävät kulkevat jokainen läpi omat työvaiheensa, mutta Jiran tehtävähallinnassa on toistaiseksi määriteltynä vain yksi työvuoto käytettäväksi kaikille projektissa näkyville tehtäville. Esimerkiksi ongelmien selvitykseen käytössä olevan Problem report -tyyppisen tehtävän tilana on käytännössä usein vain joko ratkaisematta tai ratkaistu, mutta projektin yhteinen työvuoto pakottaa niille moninkertaisen määrän tiloja, jotka määräävät kuinka niitä liikutetaan kanban-tilauilla. Tämä hankaloittaa tehtävien tietojen ajan tasalla pitämistä, eivätkä ryhmän Jira-projektin työtehtävien tiedot useinkaan vastaa niiden todellista tilaa.

4 Tietoperusta

Tässä luvussa kerrotaan, mitä Kanban-metodilla tarkoitetaan ja kuinka sitä voidaan hyödyntää ohjelmistokehityksen ryhmän toiminnassa ja mitä muita ketterän kehityksen tapoja kehitystehtävässä sovelletaan.

Kanban kirjoitetaan toisinaan isolla alkukirjaimella ja toisinaan pienellä alkukirjaimella. Hammarberg ja Sunden (2014, luku 2) käyttävät kirjassaan pientä alkukirjainta, ja toteavat, että käytännöt asiasta vaihtelevat paljon. Tässä opinnäytetyössä käytän isoa alkukirjainta, viitatessa Kanbaniin nimenomaan prosessin parantamisen metodina ja pientä alkukirjainta, kun tarkoitetaan visualisointiin käytettyä tehtävätaulua.

4.1 Ketterä ohjelmistokehitys

Greene ja Stellman määrittelevät ketterän kehityksen kokoelmana erilaisia metodeja ja metodologioita, jotka koostuvat erilaisista käytännöistä, mutta myös ajatusmallina, jonka tarkoituksena on avata kehitysprosessin kaikki osat koko tiimille. He esittävät Scrumin, XP:n, leanin ja Kanbanin neljänä eri ketterän kehityksen koulukuntana, joille kaikille yhteistä on pyrkimys muuttaa tiimin ajattelutapoja. (2014, luku 1.)

Shore ja Warden korostavat myös ketterän ajattelun tärkeyttä ja toteavat, ettei ketteryys ole rajattu mihinkään tiettyyn metodiin, vaan se on nimenomaan ajattelutapa (2021, luku 1). Greenen ja Stellmanin (2014, luku 2) mukaan kokonaisen ketterän kehityksen metodologian, kuten Scrum tai XP, omaksuminen kerralla on tehokkain tapa siirtyä työpaikalla ketterään kehitykseen, sillä metodologian määrittelemät roolit, vastuut ja käytännöt auttavat työnkulun hahmottamisessa kokonaisuutena. Myös Shore ja Warden pitävät parhaana jonkin valmiin metodologian käyttöönottoa (2021, luku 2). Kuitenkin, päinvastoin kuin Scrum tai XP, Kanban ei neuvo miten, kuka tai milloin kehitystyötä pitäisi tehdä, mutta sen sijaan se tarjoaa periaatteita ja käytäntöjä, joita soveltamalla on tarkoitus havaita ja paikata kehitystyön ongelmakohtia (Greene & Stellman, 2014, luku 9). Shore ja Warden mainitsevat Kanban-metodin helppona tapana tuoda ketteryyttä mukaan organisaation työhön, joskin pitäen sitä samaan aikaan hieman rajattuna, koska se keskittyy vain prosessin parantamiseen (2021, luku 5). Jatkuviin, ylläpitotehtäviä sisältäviin projekteihin Kanban-metodin nähdään sopivan Scrumia paremmin myös Hexactan blogikirjoituksessa, jossa Scrumin ehdottomuus kehitysjaksojen suhteen nähdään työtä hankaloittavana (Inzaurraga 7.8.2017). Samassa kirjoituksessa todetaan myös se, ettei Kanban-metodin valinnalla kuitenkaan estetä käyttämästä muita hyväksi todettuja ketteriä tapoja.

Vaikka Kanban-metodi tai lean eivät tarjoakaan ketteryyden tavoitteluun valmiita rooleja tai työohjeita, sopivat ne kuitenkin hyvin yhteen ketterään ajattelumalliin, jossa on tarkoitus avata prosessin parantaminen koko tiimille (Greene & Stellman 2014, luku 1). Kokonaisuuden hahmottamisen tärkeys korostuu Kanban-metodissa, jossa on tarkoitus visualisoida kaikki työn eri vaiheet, jotka vaaditaan sen valmiiksi saamiseksi (Sunden & Hammarberg 2014, luku 2.1). Sama kokonaisuuden hahmottaminen on myös yksi leanin periaatteista, jolla pyritään parantamaan koko systeemin tehokkuutta yksittäisten osien optimoinnin sijaan (Poppendieck & Poppendieck 2003, luku Introduction). On siis tavoiteltavaa, että myös ohjelmistokehityksen ryhmässä oma työ nähtäisiin osana laajempaa kokonaisuutta, johon kuuluu koko toimeksiantajan organisaatio.

Greene ja Stellman (2014, luku 2) muistuttavat, että vaikka ohjelmistokehityksen työtä on aiemmin verrattu tehdastyön kokoonpanovaiheeseen, jossa seurataan valmiita ohjeita, ei ole olemassa yhtä ja parasta tapaa tehdä ohjelmistokehitystä. Samanlaisen päätelmään on päätyneet myös Frederick Brooks esseessään No Silver Bullet (1986, s. 1), jossa todettiin aikanaan, ettei tekniikan tai johtamisen kehityksessä ole ollut löydettävissä mitään yksittäistä kaiken ratkaisevaa ominaisuutta. Tähän näkemykseen sopii Kanban-metodi, joka nimenomaan ei kerro kuinka ohjelmistokehitystä pitäisi tehdä (Hammarberg, Sunden. 2014, luku 2). Se ei ole projektinhallinnan työväline, kuten Scrum, tai ohjelmointityön opas, kuten XP, vaan se on ketterän kehityksen periaatteita hyödyntävä prosessinparannusmenetelmä (Greene & Stellman 2014, luku 9). Kanban-metodin rinnalla on mahdollista käyttää muita ketterän kehityksen käytäntöjä kuten esimerkiksi testivetoista kehitystä tai pariohjelmointia, ja myös tämän kehitystehtävän puitteissa on tarkoitus ottaa käyttöön Scrumista tuttu päivittäisten tapaamisten käytäntö.

4.2 Kanbanin periaatteet ja käytännöt

David J. Andersonin mukaan (18.3.2020) Kanbanin neljä periaatetta ovat:

- Aloita sillä mitä teet nyt.
- Sopikaa tavoittelevanne asteittaista, kehittyvää muutosta.
- Kunnioita nykyisiä prosesseja, rooleja, vastuualueita ja tehtävänimikkeitä.
- Rohkaise johtajuuden toimiin kaikilla organisaation tasoilla.

Kanbanin kuusi käytäntöä ovat:

1. Visualisoi (työ, työvuo ja liiketoiminnan riskit).
2. Rajoita keskeneräisen työn määrää.
3. Hallitse työvirtaa.
4. Tee prosesseista eksplisiittisiä.
5. Toteuta palautesilmukka.
6. Paranna yhteistyöllä ja kehity kokeellisesti (käyttäen malleja/tieteellistä metodia).

Kirjassaan Kanban David J. Anderson liittää alun perin metodiin kolme peruseriaatetta ja viisi ydintoimintoa (2011, luku 2.5). Myöhemmin listaa on päivitetty, ja nykyisin Anderson sisällyttää Kanban-metodiin neljä periaatetta ja kuusi käytäntöä (18.3.2020). Hammarberg ja Sunden ovat myös huomanneet käytetyn terminologian nopean kehittymisen, mutta pitävät kuitenkin tärkeimpinä visualisoinnin, työmäärän rajoittamisen ja työvirran hallinnan kolmea periaatetta – joka ovat nykyisin osa kuutta käytäntöä (2014, luku 2.1). Samassa he myös toteavat Kanban-metodin olevan vielä suhteellisen uusi ja kehittyvä metodi, mutta sen jatkuva kehittyminen on kuitenkin samaan aikaan linjassa leanin jatkuvan parantamisen periaatteen kanssa. Eri aikaan julkaistujen Kanban-metodia käsittelevien oppaiden sisältö voi siis mahdollisesti poiketa toisistaan, mutta sen ydintavoite, eli toiminnan parantaminen, on kuitenkin kaikissa sama.

4.3 Metodien käyttöönotto

Mitä sitten vaaditaan Kanbanin käyttöönottoon? Kanbanin ensimmäisen periaatteen ja ensimmäisen käytännön mukaisesti tulee aloittaa sillä mitä ryhmä tekee nyt ja visualisoida se. Hammarbergin ja Sundenin (2014, luku 2.1) mukaan siihen riittää, että tiimi visualisoi tekemänsä työn kanban-työkalulle nykyisessä muodossaan, ja alkaa hakemaan tiimille sopivaa lukemaa keskeneräisen työn rajoitteille. Myös Greene ja Stellman (2014, luku 9) muistuttavat, että kaiken lähtöpisteeksi tulee ottaa nykyinen tila. Yahoo!n Kanban-aiheisessa keskusteluryhmässä Using the Kanban Method (lakkautettu 15.12.2020) korostettiin lisäksi, ettei tiimin ole tarpeen soveltaa kaikkia eri käytäntöjä heti alussa, vaan käyttöönottoa kannattaa tehdä vaiheittain (2020). Asteittainen soveltaminen sopii hyvin Kanbanin ajatusmaailmaan, ja voi myös helpottaa myös sen käyttöönottoa verrattuna esimerkiksi Scrumin käyttöönottoon, jossa ryhmän toimintaan saatetaan vaatia paljonkin muutoksia.

Poppendieckit (2003, luku 5) ovat todenneet leanin ohjelmistokehityksen yhteydessä, että muualla erinomaisiinkin tuloksiin johtavat käytännöt eivät toiseen paikkaan kopioituna välttämättä tuota samaa tulosta, jos ei ole ymmärretty periaatteita, joiden pohjalta nämä

käytännöt ovat alun perin syntyneet. Myös Greene ja Stellman (2014, luku 9) varoittavat kopioimasta kanban-aulun rakennetta muualta, sillä se sotii Kanbanin ensimmäistä periaatetta vastaan. Samaa korostavat myös Hammarberg ja Sunden, joiden mukaan (2014, luku 3.2) kanban-aulun on tarkoitus havainnollistaa todellista ja toteutuvaa työvuota, eikä muuten suunniteltua tai työpaikan ohjeistuksissa saneltua. Heidän mukaansa (luku 3.2.2), kun työvuota alussa visualisoidaan, tulee myös välttää muutosyrityksiä havaittuun työvuohon, sillä vasta kun työvuoto on kokonaisuudessaan nähtävillä, voidaan tärkeimmät ongelmakohdat havaita. Soveltamalla Kanbanin periaatteita itse saavutetaan siis parempia tuloksia, kuin imitoimalla toisaalla luotuja malleja. Kohdeorganisaation ohjelmistokehityksen ryhmässä on aiemmin kokeiltu muuttaa Jiran kanban-aulu internetistä löydetyn ehdotuksen mukaiseksi. Tällöin aulun sarakkeissa kuvattiin kuitenkin työvaiheita, joita ryhmässä ei todellisuudessa tehty, eikä näitä työvaiheita käyty läpi vaikka se aululle lisättiin.

Nykytilan selvittämiseksi Poppendieckien mukaan lean-ohjelmistokehityksessä arvovirtakuvaus toimii aloituskohtana työvuon parantamiselle (2003, luku 1). Myös Greene ja Stellman ehdottavat kirjoittamattomien prosessien selvittämiseen arvovirtakuvausten tekemistä, joskaan ne eivät sellaisenaan vastaa kanban-aulun sarakkeita, vaan ovat tarkoitettu lisäämään ymmärrystämme systeemistä, jossa työtä tehdään (2014, luku 9). Arvovirtakuvauksia voidaan siis käyttää apuna, kun ollaan selvittämässä ryhmän työn kulkua.

Arvovirtakuvauksia voidaan ohjelmistokehitystyössä tehdä esimerkiksi saapuneesta tilauksesta. Jokainen tilaukseen liittyvä, arvoa tuottava työvaihe kirjataan ylös, ja siihen kulunut aika mitataan. Myös jokaisen työvaiheen välissä kulunut aika mitataan odotusaikana. Kuvauksesta saadaan näkyville koko prosessiin kulunut aika: arvontuottoon kulunut aika ja hukkaan mennyt aika. Kuvausten tekemiseen ei kuitenkaan tulisi käyttää liikaa aikaa, ja Poppendieckit lainaavat kirjassaan Kent Beckiä, jonka esimerkissä yhden kuvauksen tekoon oli käytetty noin puoli tuntia, ja varoitettiin kuluttamasta niihin enempää aikaa, jolloin niistä saattaa tulla tarpeettoman yksityiskohtaisia (2003, luku 1).

Poppendieckien mukaan arvovirtakuvausten avulla voidaan noudattaa leanin ohjelmistokehityksen ensimmäisen periaatetta hukan vähentämisestä (2003, luku 1). Greene ja Stellman viittaavat samaan leanin hukan vähentämisen periaatteeseen todetessaan kanban-aulun tuovan selvästi näkyville erilaiset tuotannon ongelmakohdat; esimerkiksi silloin, jos työt kerääntyvät aululla yhteen sarakkeeseen (2014, luku 9). Sundenin ja Hammarbergin mukaan jo pelkkä nykyisen työvuon tekeminen näkyväksi kanban-aulun avulla voi parantaa työtehokkuutta (2014, luku 2.1). Kanbanin ensimmäistä

käytäntöä – visualisointia – voidaan siis käyttää toteuttamaan leanin ensimmäistä periaatetta hukan vähentämisestä.

Hammarberg ja Sunden (2014, luku 3.2.2) myöntävät työvuon kuvaamisen voivan olla työlästä, mutta siihen käytetyn ajan olevan kuitenkin hyödyllistä, sillä se pakottaa tutkimaan tarkemmin vallalla olevia olettamuksia ja lisää ymmärrystä yhteistyöstä. Samoja olettamuksia tarkoittavat Greene ja Stellman (2014, luku 2) hajanaisella näkemyksellä, jonka mukaan projektin jäsenten lokeroituminen ja kommunikaation puute lisäävät ongelmia, kun projektia ei nähdä kokonaisuutena, vaan sitä tarkastellaan vain yksittäisten työtehtävien osalta. Myös McHenry ja Will IBM:ltä (2021) neuvovat käyttämään arvovirtakuvauksia etsiessä ohjelmistotuotannon pullonkauloja. Poppendieckit huomauttavat, että ohjelmistokehityksessä suurimmat hukka-ajat syntyvät nimenomaan varsinaisen kehitystyön ulkopuolella ja ehdottavatkin arvovirtakuvausten laajentamista kuvaamaan prosessia myös asiakkaan taholla, ja näin löytää lisää arvoa tuottavia toimia ja mahdollisia hukan kohteita (2003, luku 1).

Andersonin tavoitteena Kanbanin neljännelle käytännölle ”Tee prosesseista eksplisiittisiä” on se, että keskustelu ongelmien syistä ei olisi subjektiivista (2020). Jos on näkyvästi selvillä, kuinka asiat tulisi hoitaa, on helpompi löytää yhteinen ymmärrys myös ongelmien syistä sekä niiden korjaamiseksi vaadituista muutoksista. Greene ja Stellman näkevät kanban-taulun yhtenä tapana toteuttaa tätä eksplisiittisen prosessin käytäntöä (2014, luku 9). Esimerkiksi ohjelmistokehityksen ryhmän taululla tulisi olla yksiselitteisesti jokaisen nähtävillä kaikki eri tehtävätyypit, niiden työvaiheet ja keskeneräisen työn rajoitteet.

Scrumin tehtävätaulu on erilainen kuin kanban-taulu. Hammarberg ja Sunden (2014, luku 2) määrittelevät Kanbanin järjestelmäksi, jolla on pyrkimys parantaa toimintatapoja hallitsemalla ja optimoimalla visuaalisia signaaleja. Visuaalisena signaalina toimii tässä tapauksessa kanban-taulu, joka muistuttaa ulkoisesti Scrumin tehtävätaulua, mutta kattaa ryhmän työtavat huomattavasti sitä tarkemmin. Kun kohteet Scrumin tehtävätaululla kuvaavat kehitteillä olevaan käyttötapaukseen liittyviä yksittäisiä työsuorituksia, ovat kanban-taulun kohteet yksittäisiä tehtäviä laajempia työkohteita, ja kuvaavat esimerkiksi uutta kehitteillä olevaa ohjelmiston ominaisuutta (Greene & Stellman 2014, luvut 5 ja 9). Ohjelmistokehityksen ryhmälle saapuvasta tilauksesta voisi Jiraan syntyä useita eri työkohteita, joista jokainen kuvaa yhtä tarvittavaa ohjelmiston ominaisuutta. Nykyisin ryhmän käytössä oleva Feature Request -tyyppinen tiketti Jirassa voisi kuvata tällaista ominaisuutta, ja tarvittaessa yksittäiset työtehtävät voidaan eritellä sen alitiketeiksi. Näiden alitikettien työn laajuus voi olla jotenkin verrattavissa Scrumin tehtävätaulun kohteisiin.

Toinen eroavaisuus Scrumin ja Kanbanin taulujen välillä on sarakkeiden käytössä. Scrumissa tehtävillä voi olla jokin kolmesta statuksesta: tekemättä, tekeillä ja valmis (Greene & Stellman 2014, luku 5). Kanban-taulu kuvaa laajemmin koko työvuota ja sisältää sarakkeet jokaiselle prosessin vaiheelle (Greene & Stellman 2014, luku 9). Kanban-metodissa taulun sarakkeiden lukumäärä riippuu siitä, kuinka monta eri työvaihetta yksi kohde joutuu käymään läpi ennen kuin siinä kuvattu ominaisuus on asiakkaan käytettävissä.

Hammarberg ja Sunden (2014, luku 4.2.4) antavat vahvan suosituksen käyttää oikeaa fyysistä taulua ja muistilappuja elektronisen taulun sijaan, ja tarvittaessa kahdentaa visualisointi elektroniseen muotoon yksinkertaistettuna. Sähköisessä muodossa olevan taulun etuna on kuitenkin sen näkyvyys myös ryhmän työtilojen ulkopuolella eri sidosryhmiä varten. Hammarberg ja Sunden huomauttavat, että työn visualisoinnissa on tärkeää, ettei visualisointiin valittu media rajoita kuinka prosessia voidaan muokata (2014, luku 3.1.1). Tämä on kuitenkin aiheuttanut ongelmia ohjelmistokehityksen projektissa, jossa käytössä on Jira, ja jolla työvuon muokkaaminen on ollut ajoittain haastavaa. Jiran käytössä on kuitenkin etuna se, että se on helposti kaikkien työntekijöiden saatavilla myös etätöiden aikana. Jos visualisointia varten kuitenkin valitaan elektroninen työkalu, on Hammarbergin ja Sundenin mukaan varmistettava, että siinä on käytettävissä näkymä, josta kaikki tarpeellinen informaatio on saatavilla, ja asettaa tämä kaikkien nähtäville (2014, luku 3.1.1). Samassa he myös jatkavat, että on jatkuvasti syytä uudelleenarvioida ja tarvittaessa päivittää kanban-taulua, eikä siinä saisi kuitenkaan olla liikaa informaatiota, jotta siitä ei tule vaikeasti ymmärrettävä.

4.4 Keskeneräisen työn rajat

Kanbanin toisen periaatteen (Sopikaa tavoittelevanne asteittaista, kehittyvää muutosta) ja toisen ja kuudennen käytännön (Rajoita keskeneräisen työn määrää ja Paranna yhteistyöllä ja kehity kokeellisesti) mukaisesti, tulisi keskeneräisen työn määrälle asettaa ryhmän yhdessä sopimia rajoja, ja sitoutua yhdessä noudattamaan niitä ja tehdä niihin tarvittaessa muutoksia. Kanban-taululla keskeneräisen työn rajoite ilmoitetaan lukuarvona työvaihetta vastaavan sarakkeen kohdalla, ja tämä luku määrää kuinka monta tehtävää voi samassa työvaiheessa kerrallaan olla (Greene & Stellman 2014, luku 9). Hieman nimensä vastaisesti käytännön tarkoituksena ei ole estää työntekoa, vaan rajoitusten tarkoituksena on toimia keskustelun avaajana ongelmatilanteissa, kun rajoitteet tulevat vastaan (Hammarberg & Sunden 2014, luku 6).

Hammarberg ja Sunden (2014, luku 12) painottavat WIP-rajojen (Work in Progress eli keskeneräinen työ) tärkeyttä ja ovat sitä mieltä, että juuri rajoitusten myötä tulisi ryhmän

aloittaa keskustelut mahdollisista muutoksista työtapoihinsa, sillä kaikessa päteviä, oikeita arvoja ei ole olemassa. Kun kanban-aululla tikettiä ollaan siirtämässä sarakkeeseen, joka on jo täynnä, tulisi ryhmässä miettiä onko siinä kyse poikkeuksellisesta tilanteesta, vai onko ryhmän parempi muuttaa rajoja työvuon parantamiseksi.

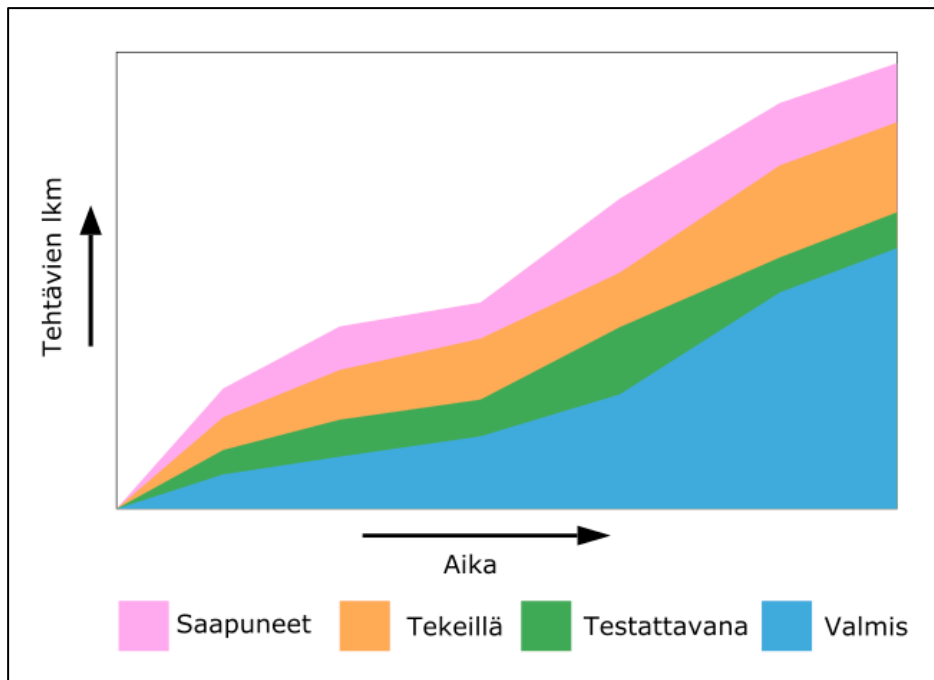
Hammarbergin ja Sundenin mukaan yksi yleisimpiä virheitä Kanbanin käyttöönotossa on WIP-rajojen pois jättäminen (2014, luku 12.5). He myös jatkavat, ettei ole olemassa mitään yleispätevää sääntöä, jonka perusteella rajoitteita voisi asettaa, vaan jokaisen ryhmän on löydettävä itselleen ja tilanteeseen sopivat arvot (2014, luku 6). Samaa mieltä tästä ovat myös Greene ja Stellman (2014, luku 9). Liian suuret raja-arvot voivat johtaa tilanteeseen, jossa kanban-aululle tulee niin paljon merkintöjä, ettei siitä enää voi nähdä kokonaiskuvaa, eikä taulun käyttö enää palvele tarkoitustaan kokonaiskuvan antamisesta ja läpimenoajat kasvavat kaikilla töillä (Hammarberg & Sunden 2014, luku 6.1.3). Greene ja Stellman muistuttavat kuinka on tärkeää varmistaa, että niin kehittäjät kuin ulkoisten sidosryhmien jäsenet ymmärtävät, että rajoitteiden tullessa vastaan tietyllä työvaiheella, on ryhmän fokuksen siirryttävä toiseen työvuon vaiheeseen (2014, luku 9). Tämä koskettaa myös ohjelmistokehityksen ryhmän ulkopuolisia, sillä muutokset ryhmän toiminnassa vaikuttavat myös toisten ryhmien ja yksiköiden ja mahdollisesti myös ulkoisten tahojen työhön. Tämän takia on tarpeellista keskustella muutoksista laajemmin, niin että muut ryhmät ymmärtävät muutoksen tuomat vaikutukset koko organisaation toimintaan.

Keskeneräisen työn rajoitteiden yhteydessä Greene ja Stellman viittaavat leanin ohjelmistokehityksen ajattelutyökaluun vaihtoehtojen miettimisestä, jonka mukaan kanban-aululla olevat työt ovat vaihtoehtoja toisilleen, ja jonkin sarakkeen ollessa täynnä, on mahdollista tehdä jotain muita töitä (2014, luku 9). Poppendieckit kuvaavat leanin ohjelmistokehityksen vaihtoehtojen ajattelua ennemminkin päätöksenteon ominaisuutena niin, että kehitystyössä ei ole syytä sitoutua valintoihin tai tarkkoihin suunnitelmiin ennenaikaisesti, vaan pitää eri vaihtoehtoja mahdollisena ja tehdä vastuulliset päätökset mahdollisimman myöhään (2003, luku 3). Asian voisi nähdä myös niin, että jos esimerkiksi jokin ohjelmiston ominaisuus on saatu valmiiksi, mutta tuotantoon viennin vaihe jää rajoitusten vuoksi odottamaan, on kuitenkin jo sitouduttu saattamaan tämä ominaisuus valmiiksi. Tällöin tärkeintä lienee selvittää mistä täyteen menneet rajoitukset johtuvat, ja onko prosessissa jotain korjattavaa sen sijaan, että asian annettaisiin olla ja siirryttäisiin vapaana olevien sarakkeiden täyttämiseen. Ryhmän työssä on usein myös niin, että vain yksi henkilö suorittaa läpi kaikki työvaiheet, jolloin sarakkeesta toiseen siirtäminen ei toimi signaalina kellekään muulle, joka vastaanottaisi sen uutena tehtävänä itselleen.

Aiemmin, kun ryhmässä ei ole vielä ymmärretty tai tiedetty Kanbanin periaatteita, on päädytty sekavaan ja tukkoiseen kanban-tauluun, kun asetettuja raja-arvoja ei ole noudatettu. Opinnäytetyön yhteydessä on tarkoitus sitoutua yhdessä ryhmänä noudattamaan rajoituksia, ja tarvittaessa yhdessä muuttaa niitä. Ryhmän Jira-projektissa ovat kanban-taululle asetetut keskeneräisen työn rajoitteet lähes koko ajan ylittyneet, mutta asiaan ei ole kiinnitetty juuri huomiota, sillä ryhmässä ei ole yhdessä sovittu niiden noudattamisesta. Asettamalla rajoitteet keskeneräiselle työlle, ja erityisesti sitoutumalla noudattamaan ja muuttamaan niitä tarvittaessa, ryhmässä olisi mahdollista vähentää päällekkäisten työtehtävien määrää ja työn kuormittavuutta.

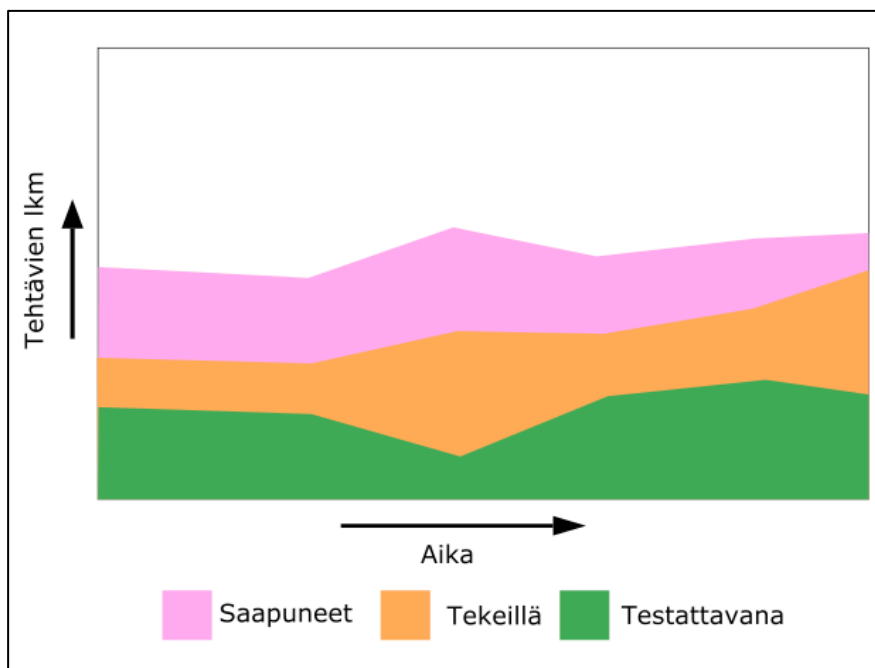
4.5 Mittaristot Kanban-metodissa

Työn virralla tarkoitetaan taajuutta, jolla työkohteet liikkuvat systeemin läpi, ja virran huippu voidaan saavuttaa rajoittamalla epätasapainoa ja ylikuormitusta ja tekemällä työt valmiiksi saakka ennen uusien aloittamista (Greene & Stellman 2014, luku 9). Greene ja Stellman korostavat virranhallinnassa mittaamisen tärkeyttä, ja ehdottavat kumulatiivista työvirran kuvaajaa (Cumulative Flow Diagram, CFD) keskeneräisen työn rajojen säätämiseen. (2014, luku 9). Myös Hammarberg ja Sunden pitävät kumulatiivista virtakuvaajaa suosittuna ja informatiivisena (2014, luku 11). Kun kanban-taululla on nähtävillä, kuinka monta tehtävää kussakin sarakkeessa tietyinä hetkenä on, voi kuvaajan avulla tarkastella tilannetta pidemmältä aikaväliltä. Esimerkissä (Kuva 1) tehtävien kokonaismäärä kasvaa ajan suhteen, ja väritettyjen alueiden paksuus kuvaa kuinka paljon työtehtäviä kussakin kanban-taulun sarakkeessa on. Jos jonkin sarakkeen leveys kasvaisi suhteessa muihin, merkitsisi se mahdollista ongelmatilannetta työvaiheen kohdalla.



Kuva 1. Esimerkki kumulatiivisesta työvirran kuvaajasta

Keskeneräisen työn aluekuvaajalla (WIP area chart, eli Work in Progress area chart) voidaan mitata kehitteillä olevien ominaisuuksien lukumääriä ja niiden tekemiseen tai niistä johtuvaan odottamiseen kulunutta aikaa. Kuvaajaa seuraamalla voidaan havaita järjestelmän mahdollisia ongelmakohtia ja toisaalta se myös näyttää stabiililta, kun järjestelmä, jota kuvaajalla mitataan, on tasapainossa. (Greene & Stellman 2014, luku 8.)



Kuva 2. Keskeneräisen työn aluekuvaaja

Toisin kuin kumulatiivisessa työvirran kuvaajassa, ei keskeneräisen työn aluekuvaajassa (Kuva 2) ole mukana valmiiksi saatuja tehtäviä. Tämä tekee kuvaajasta hieman yksinkertaisemmän, ja voi helpottaa ongelmallisten työvaiheiden havaitsemista. Jos kuvaaja alkaisi kumuloitumaan voitaisiin todeta uusien työtehtävien määrän kasvavan nopeammin, kuin mitä entisiä ehdittäisiin saamaan valmiiksi.

Myös leanissa mittausten tekeminen on osa kokonaisuuden näkemisen periaatetta, mutta Poppendieckien mukaan on erityisesti tärkeää mitata kokonaisuutta yksittäisten suoritusten sijaan, sillä ohjelmistokehityksessä suurin osa virheiden juurisyyistä on yksittäisten henkilöiden sijaan käytetyissä menetelmissä ja toimenpiteissä, ja mittaamalla yksilöiden suoritusta voidaan helposti piilottaa todelliset ongelmat. (Poppendieck & Poppendieck 2004, luku 7.)

4.6 Lupa kokeilevaan kehittämiseen

Neljäs periaate – Rohkaise johtajuuden toimiin kaikilla organisaation tasoilla – tarkoittaa Andersonin mukaan oikeutta tarpeen vaatiessa tehdä perusteltuja muutoksia, ja työntekijöillä tulee kaikilla organisaation tasoilla olla mahdollista muuttaa työtapoja ilman rangaistuksen pelkoa, vaikka jotain menisikin pieleen (2020). Myös Greene ja Stelman toistavat kuinka tärkeää ketterälle kehitykselle on se, että on myös lupa epäonnistua, olettaen, että virheistä myös otetaan opiksi (2014, luku 10). Samoin Poppendieckien mukaan on leanin periaatteiden mukaista olettaa, että työntekijät tietävät kuinka työ kannattaa tehdä, ja että johtajien tulisi keskittyä muuttamaan asioita, jotka ovat työn tekemisen haittana. (2004, luku 5).

Hammarberg ja Sunden muistuttavat myös kanban-työkalun tekemisessä tiimin yhteistyön tärkeydestä, sillä ulkopuolelta tullut suunnitelma saatetaan helposti torjua, kun taas tiimin on helpompi hyväksyä itse tekemänsä työvuo (2014, luku 2.1). Toimeksiantajan ohjelmistokehityksen ryhmässä on uskallettu kokeilla uusia tekniikoita ja työtapoja, ja pääasiassa työtavoista on päätetty ryhmän kesken.

Muutosten tekemiseen on Anderson (2020) varannut myös Kanbanin käytännön numero kuusi – Paranna yhteistyöllä ja kehity kokeellisesti (käyttäen malleja/tieteellistä metodologiaa). Samoin kuin neljäs periaate antaa tämä käytäntö tilaa epäonnistumisille ja kokeiluille, ja niiden kautta uuden oppimiselle. Anderson tarkentaa, että kun kaikilla osallisilla on sama tietämys käytettävistä teorioista ja malleista, ovat muutoksista odotetut tulokset perusteltavissa ja tehdyistä muutoksista voi olla ryhmässä yhteinen ymmärrys.

Hammarberg ja Sunden (2014, luku 2.1) näkevät tämän enemmänkin leanin periaatteena kuin varsinaisena Kanbanin käytäntönä, mutta ovat kuitenkin samaa mieltä sen sisällöstä.

Myös Greene ja Stellman ovat sitä mieltä, että jotta Kanban-metodista olisi hyötyä, tulee tiimin käyttää aikaa oman ohjelmistokehityksen systeeminsä ymmärtämiseksi, ja muutoksia tulisi tehdä niin, että niiden tulokset voidaan mitata (2014, luku 9). Jos tätä ajatusta haluaisi verrata musiikkiin, ei kaikkien yhtyeen jäsenten ole välttämätöntä tuntea musiikin teoriaa, jotta musiikkia voisi soittaa, mutta ryhmän jäsenten kommunikointi ja yhteistyö voi helpottua, kun käytetyt termit ja ymmärrys ovat yhteisiä ja samalla tasolla. Ryhmän jäsenten koulutus ja osaaminen on monipuolista, mutta ketterän kehitykseen liittyvät teoriat eivät ole olleet osa ryhmän päivittäistä työtä. Kanban-metodi ei tuo mukanaan erityisen paljon uutta terminologiaa, mutta on hyödyllistä, jos sen periaatteista on ryhmässä yhteinen ymmärrys.

4.7 Palautteen tärkeys

Anderson kertoo viidennestä, palautesilmukan käytännöstä suoremmin, että Kanban-metodissa päivittäiset tapaamiset ja erilaiset katselmoinnit ovat keinoja palautesilmukan toteuttamiseksi, ja niillä pyritään pääasiassa vertaamaan odotettua tulosta toteutuneeseen ja päivittämään tarvittaessa prosesseja ja käytäntöjä (2020). Kanban-metodi ei itsessään määrää aikarajattua kehittämistä, mutta Hammarbergin ja Sundenin mukaan siitä on erityisen paljon hyötyä esimerkiksi priorisoinnin kannalta, ja ryhmä voi valita itselleen sopivan aikarajauksen (2014, luku 12.2). Palautesilmukan voisi nähdä myös viikoittaisena reflektointina ryhmäpalaverin yhteydessä, jossa voitaisiin käydä läpi viikon aikana valmiiksi saadut tehtävät ja niiden kanssa olleet ongelmat ja miten niitä voitaisiin jatkossa estää.

Greene ja Stellman tarkoittavat palautesilmukalla kehitteillä olevan ominaisuuden tarkastelua kehitysjakson tai työvaiheen päätteeksi, ja kanban-aululla työtehtävän kulkeminen työvaiheesta toiseen toteuttaa palautesilmukan. Esimerkiksi kun kehitteillä oleva ominaisuus siirtyä hyväksymistestaukseen, se voi siirtyä edelleen tuotantoon vietäväksi tai se voi vaatia lisää työtä ja palautua takaisin työjonoon. Työtehtävien palautesilmukan pituutta voi säätää muuttamalla sarakekohtaisia keskeneräisen työn rajoitteita. Kun palautesilmukka on lyhyt, ja keskeneräisen työn rajoite on pieni, voi työtehtävien siirtyminen työjonossa ruuhkautua. Jos taas palautesilmukka on liian pitkä, ja rajoite on suuri, voi työtehtävien kerääntyminen kuormittaa jotain työvaihetta liikaa. (Greene & Stellman 2014, luku 9.)

Jos esimerkiksi tuotantoon viennin työvaiheeseen kertyy useita kymmeniä tehtäviä odottamaan, voi tämä tehdä päivityksistä suuria ja kokonaisuuden hallinta voi olla vaikeaa. Jos työvaiheen raja on liian pieni, toimii se hidasteena edelliselle työvaiheelle.

Hammarberg ja Sunden myös varoittavat, että liian pitkä palautesilmukka voi heikentää laatua ja hankaloittaa virheiden korjaamista, ja että korjaustyö on usein helpompaa, jos sitä päästään tekemään, kun virheen sisältävä ominaisuus on äskettäin viety tuotantoon ja se on yhä kehittäjän tuoreessa muistissa (2014, luku 5.2.5). Kovin monimutkaisissa ohjelmistoissa voi myös käydä niin, että koodissa voi olla jokin virhe jopa vuosia, ennen kuin se aiheuttaa havaittuja ongelmia. Tällaisissa tapauksissa virheidenkorjaus voi olla haastavaa ja aikaa vievää. Päivittäisten tapaamisten avulla työtehtävissä ilmenneisiin ongelma-kohtiin voidaan puuttua nopeasti. Työtehtävien etenemisen ei tarvitse pysähtyä odottamaan ratkaisuja useiden päivien ajaksi, kun niitä käsitellään säännöllisesti päivittäin.

4.8 Tehtävävaraston ylläpito

Scrumissa tehtävien varasto (englanniksi backlog) tarkoittaa nimettyä listaa uusista ominaisuuksista, joista yhdelle rajatulle kehitysjaksolle valitaan sillä hetkellä tarpeellisimmaksi koetut sprinttikohtaiseen backlogiin (Greene & Stellman 2014, luku 4). Kanban-metodin käytännöt eivät määrää käyttämään erillistä backlog-listaa, mutta se on mahdollista. Hammarberg (2013) esittää blogikirjoituksessaan myös vaihtoehdoisen tavan kuvata backlogissa esitettyjä tehtäviä mindmapin avulla. Tämä ei välttämättä sovi kaikille, mutta osoittaa kuitenkin sen, että Kanban-metodi on joustava toteutustavoissaan. Jirassa on valittavissa käyttöön erillinen backlog-näkymä, jossa listataan projektiin kirjatut, vielä aloittamattomat työtehtävät, ja josta työtehtäviä siirretään edelleen näkyviin kanban-aululle, kun työ on tarkoitus aloittaa.

Poppendieckien mukaan useita työvuosia pitkän työjonon ylläpitäminen ja niille tehtävät aika-arviot ovat ajanhukkaa, ja työjonon tulisi sisältää vain sen verran työtä, joka on realistisesti organisaation tehtävissä lähitulevaisuudessa. He rinnastavat pitkän työtehtävien listan suureen varastoon, jonka sisältö vanhenee erityisesti ohjelmistokehityksessä nopeasti ja käy tarpeettomaksi (2006, luvut 1 ja 4). Myös Agile Alliancen (2021) ohjeissa neuvotaan tarkistamaan säännöllisesti, että backlogissa pidetään vain oikeasti merkityksellisiä kohteita. Ohjelmistokehityksen ryhmällä on Jira-projektissaan kirjattuna useita satoja tehtäviä, joista monet on kirjattu järjestelmään lukuisia vuosia sitten. Tämä on haastava tilanne, ja työlistan läpikäynti ja siivoaminen voi olla aikaa vievää.

Backlogin ylläpidon yhteydessä viitataan usein myös tuoteomistajiin ja tuotekohtaiseen backlogiin (Van Rooden 2016; Agile Alliance 2021). Kehitystehtävän kohteena olevalla ryhmällä tehtävien varasto sisältää kuitenkin työkohteita useilta eri sidosryhmiltä, ja ne sisältävät asiakastuotteiden kehityksen lisäksi sisäisiä kehitystehtäviä ja ylläpidollisia

tehtäviä. Tehtäviä on kirjattu järjestelmään, minkä jälkeen niitä ei ole koskaan priorisoitu työlistalle tai sitten ne ovat siirretty työlistalta pois, minkä jälkeen ne ovat unohdettu backlogiin. Priorisointi on ollut haasteellista, eikä laajaa tehtäväkenttää kuvaavassa Jira-projektissa ole ollut mahdollista pitää tuotekohtaisia backlogeja.

4.9 Yhteenveto

Ei ole yhtä ainoa tapaa tai käytäntöä, joka toimisi aina yhtä hyvin ja samalla tavoin jokaisessa organisaatiossa. Kanban-metodi käyttää hyväkseen leanin ajatusmalleja, ja se on yleinen ja tehokas tapa tuoda lean-ajattelua mukaan organisaatioon (Greene & Stellman 2014, luku 9). Sen periaatteet ja käytännöt ovat yksinkertaisia, mutta sen käyttöön vaaditaan kuitenkin halua kehittyä ja panostusta esiin nousseiden ongelmien selvitykseen.

Metodin tuomista yksittäisistäkin käytännöistä voi olla hyötyä, mutta on hyvä ymmärtää periaatteet, joiden pohjalta käytännöt on luotu, jotta niitä voi soveltaa omassa organisaatiossaan tehokkaasti. Yksi tärkeimmistä Kanban-metodin käytännöistä on keskeneräisen työn rajoitteet, joita noudattamalla on tarkoitus saavuttaa tasapaino saapuvien työtehtävien ja niiden aiheuttaman työkuorman kesken. Rajoitteet ovat tapauskohtaisia ja sopivat arvot niille löytää vain kokeilemalla. Jotta muutokset voitaisiin tehdä perustellusti, on työvirrasta ja sen toimivuudesta tarpeellista tehdä objektiivista ja säännöllistä mittausta. Epäonnistumiset ovat kuitenkin samalla mahdollisuus oppia ja saada kokemusta.

Esimerkiksi päivittäisten tapaamisten voidaan nähdä toteuttavan Kanban-metodin käytäntöä palautesilmukasta. Käytännön voi nähdä toteutuvan myös kehitysjaksojen päätteeksi pidettävissä katselmoineissa. Saapuvien tehtävien varastoa (backlogia) on syytä seurata ja ylläpitää säännöllisesti, jotta työtehtävien määrä ja tärkeysjärjestys olisi hallittua.

5 Menetelmät

Tämän kappaleen alaluvuissa on tarkemmin eritelty eri toimet, joilla opinnäytetyön kehitystehtävän tavoitteisiin pyritään.

Opinnäytetyö toteutetaan toimintatutkimuksena ja sen tarkoituksena on auttaa paikallistamaan ongelmakohtia ohjelmistokehityksen ryhmän työvuossa ja helpottaa niiden ratkaisussa. Ojasalon, Moilasan ja Ritalahden mukaan toimintatutkimus on käytännönläheinen ja siinä on tärkeää kehityksen kohteena olevan organisaation kehitystyöhön osallistuminen (2015. s. 37). Tehtävänä ryhmän toimintamallien selvitys ja niiden kehittäminen olisi mahdotonta ilman ryhmän aktiivista osallistumista siihen.

Toimintatutkimus on toteutustapana käytännönläheinen, ja kun tavoitteena on yhdessä ryhmän kanssa tehdä muutoksia ryhmän toimintatavoissa, sopii toimintatutkimus siihen hyvin. Tämän lisäksi toimintatutkimuksessa on tarkoituksenmukaista, että kehittäjä on osallisena ryhmän toiminnassa (Ojasalo, Moilanen & Ritalahti. 2015. s. 61), ja tämän kehitystehtävän aikana opinnäytetyön tekijä on mukana kohteena olevan ryhmän toiminnassa.

Ryhmällä ei ole ollut ohjelmistokehityksen työssään käytössään mitään erityistä nimettyä kehitysmetodia, eikä Scrumia rooleineen ja käytäntöineen ole koettu sopivaksi ryhmän ylläpitotyötäkin sisältävään toimintaan. Kanban-metodin avulla voidaan saada tarkempi kuva ryhmän nykyisestä toiminnasta ja toiminnoista ilman, että ryhmän tulisi tehdä mitään toisin kuin aiemmin. Pidemmällä tähtäimellä metodin avulla on kuitenkin tarkoitus parantaa ryhmän toimintaa, mutta vähän kerrallaan ja niin, että muutosehdotukset tulevat ryhmän itsensä sisältä, jolloin niihin voi olla helpompi sitoutua. Toimintatutkimus myös etenee toistuvissa sykleissä, jotka voidaan eritellä suunnittelun, toiminnan ja arvioinnin jaksoihin (Ojasalo, Moilanen & Ritalahti. 2015. s. 61). Kanban-metodi on jo itsessään jatkuvan parantamisen metodina verrattavissa vastaavanlaiseen suunnittelun, tekemisen ja arvioinnin sykliin ja sopii siksi hyvin toimintatutkimuksena kehitettäväksi kohteeksi.

Ojasalo, Moilanen ja Ritalahti nimeävät toimintatutkimusta tukeviksi menetelmiksi mm. erilaisten prosessikarttojen piirtämisen ja prosessien selvityksen, ja niissä piilevien ongelmien etsimisen (2015. s. 44). Tähän kuvaukseen sopii arvovirtakuvausten tekeminen ryhmän työvaiheista ja niiden visualisointi edelleen tarkemmin ryhmän työtä kuvaavalle kanban-taululle, jonka avulla on tarkoitus myös saada paremmin havaittua mahdolliset prosessin pullonkaulat. Ryhmän työvuon arvovirtakuvaukset eri tehtävätyypeistä mallinnetaan kehittäjien kanssa ja ryhmän kesken niistä koostetaan yhteinen työvuon kuvaus. Arvovirtakuvauksista saadaan mahdollisesti aikaan monta eri prosessikuvausta,

ja niistä tulee ryhmän kesken päättää sopiva yhteinen muoto. Myös lean-ajattelussa korostetaan, että paras henkilö kertomaan miten työt kannattaa tehdä, on työntekijä itse (Poppendieck & Poppendieck 2003, luku 5). On siis parempi antaa yhteisen työvuonokuvauksen syntyä ryhmän sisältä.

Myös erilaiset ideointipalaverit ryhmätöinä ja aivoriihenä on sopiva toimintatutkimusta tukeva menetelmä (Ojasalo, Moilanen & Ritalahti. 2015. s. 44). Vaikka opinnäytetyössä on tärkeää selvittää ryhmän nykyiset toimintatavat, on Kanban-metodin tarkoituksena myös jatkuva kehittäminen, ja näin ollen myös siis selvittää minkälaiset toimintatavat prosessissa toimisivat paremmin. Ongelmatilanteita tai pullonkauloja prosessista löydettyä on tarkoitus aloittaa keskustelu ryhmässä vaivojen parantamiseksi. Päivittäisissä tapaamisissa, viikkopalavereissa ja kehitysjaksojen päätteeksi on mahdollista tuoda esille erilaisia hidasteita, ja tällöin voidaan ideoida erilaisia tapoja, joilla parantaa ryhmän sisäistä ja sitä ympäröivää toimintaa.

Ojasalo, Moilanen ja Ritalahti mainitsevat myös toimijoiden välisen keskustelun ja sen dokumentoinnin yhtenä toimintatutkimuksen menetelmistä (2015. s. 62). Myös opinnäytetyön aikana ryhmän kesken pidettävissä palavereissa tullaan keskustelemaan Kanban-metodin käytöstä ja tehdään yhdessä päätöksiä, jotka myös dokumentoidaan osana opinnäytetyötä.

5.1 Työn visualisointi arvovirtakuvauksin

Kanban-metodin ensimmäisen periaatteen ja ensimmäisen käytännön mukaisesti on prosessin parantamiseksi aloitettava selvityksellä ryhmän nykyisistä työtavoista ja sen visualisoinnista. Ryhmän Jira-projektin kanban-taulu ei nykyisellään kuvasta ryhmän toteutuvaa työvuota, ja sen selvittämiseksi tehdään erilaisia arvovirtakuvauksia, joiden avulla taulun sarakkeet ja Jira-töiden käyttämät statukset saadaan näkyville.

Vaikka ryhmässä ei seurata mitään tiettyä tarkkaan määriteltyä ohjelmistokehityksen metodia, kuten Scrumia, on jokaisella työntekijällä kuitenkin olemassa jokin tietty tapa tehdä asioita. Näiden kirjoittamattomien työtapojen selvittämistä varten jokaisen ryhmäläisen kanssa pidetään tunnin mittainen pienoistyöpaja, jossa käydään läpi muutama viimeisin valmiiksi saatu työtehtävä. Työnkulusta kirjoitetaan ylös työvaiheet, työvaiheisiin käytetty aika ja niiden välissä kulunut aika. Selvitys tehdään käyttäen apuna Jiraan tallennettuja tilamuutosten päivämääriä ja työaikakirjauksia.

Kuvausten perusteella voidaan nähdä, minkälainen työtehtävien yleinen työnkulku on tehtävän kirjaamisesta sen valmiiksi saamiseen. Kuvauksista voidaan myös nähdä yleisiä läpimenoaikoja ryhmän työtehtäville, sekä mahdollisia pullonkauloja, joihin ryhmän

kannattaa kiinnittää huomiota. Arvovirtakuvausten perusteella tehdään ehdotus työkuukua vastaavista tilakaaviosta eri tehtävätyypeille. Tilakaaviot esitetään ryhmälle ja niihin tehdään tarvittaessa muutoksia, kunnes ne vastaavat ryhmän yhteistä näkemystä.

5.2 Jira-projektin kanban-aulun päivitys

Jokaiselle tehtävätyypille luodut tilakaaviot vastaavat Jiran työvuosasetuksissa käytettyjä tila-asetuksia. Nykyisten työvuosasetusten tiloja tulee vastata jokin uusi tila, ja Jiran muutostyötä varten tulee tehdä kaaviot, joissa nykyiset tilat osoitetaan vastaavaan uuteen tilaan. Nämä kaaviot tarkistetaan ryhmän kesken, ja niihin tehdään tarvittaessa korjauksia.

Kun uudet tilakaaviot ja nykyisten ja tulevien tila-asetusten vastineet on mallinnettu, voidaan päivittää Jiran työvuosasetukset jokaiselle työtehtävälle. Muutoksen voi suorittaa ainoastaan Jiran ylläpidosta vastaava ryhmä, jolle tulee tehdä tilaus muutoksesta. Kun uudet työvuot on saatu käyttöön, päivitetään kanban-aulun sarakkeet vastaamaan päätettyä työvuota. Tämä voidaan suorittaa ryhmän kesken, eikä vaadi ylläpidolta toimia.

Ryhmä päättää yhdessä aluksi sopivat työmäärärajoitteet kanban-aulun sarakkeisiin, ja näitä on tarkoitus päivittää kokeilujakson aikana, mittaamalla ensin työvuon tehokkuutta. On kuitenkin otettava huomioon erilaiset poikkeustapaukset, joita ryhmän operatiivisessa toiminnassa tulee vastaan, ja tilanteen niin vaatiessa on rajoitteista mahdollisesti joustettava, jotta esimerkiksi kiireelliset ongelmatilanteet voidaan ratkaista nopeasti.

5.3 Työvuon tehokkuuden mittaus Jiran avulla

Kun kaikki ryhmän jäsenet sitoutuvat noudattamaan sovittua tapaa tehdä työt, voidaan työvuon toimivuudesta ja tehokkuudesta tehdä Jirassa myös mittauksia. Jira-projektissa on käytettävissä erilaisia raportointimalleja. Opinnäytetyön aikana näistä etsitään parhaiten Kanban-metodin kanssa käyttöön soveltuvat kuvaajat tai tilastot, joiden avulla työvuon tehokkuutta tullaan seuraamaan aikana.

Sopivaa mitattua tietoa saadaan, kun kehitystyö jaetaan lyhyisiin aikarajattuihin jaksoihin, jolloin edellisen jakson tuloksia voidaan verrata seuraavaan. Mittareista saataneen parhaiten hyötyä pitkällä aikavälillä, mutta myös kokeilun aikana voidaan arvioida niiden soveltuvuutta ryhmän toiminnan analysointiin.

5.4 Aikarajattu kehitys

Vaikka Kanban-metodi ei erityisesti määrää tekemään ohjelmistokehitystä aikarajatuissa jaksoissa, kuten esimerkiksi Scrumin sprintit, on hyödyllistä pitää säännöllisesti retrospektiivi, jossa voidaan analysoida ja mitata ryhmän toimintaa kokonaisuutena.

Jokaisen kehitysjakson alussa valitaan Jiran tehtävähallinnasta jakson aikana valmiiksi saatavat asiakastilaukset ja ylläpitotehtävät. Valintapalaveriin osallistuvat asiakastilausten ja ylläpitotehtävien priorisoinnista päättävät henkilöt sekä (kokeilujakson aikana) ohjelmistokehityksen ryhmän jäsenet.

Valittujen työkohteiden laajuus pitää olla sellainen, että ne ovat mahdollista saada valmiiksi jakson aikana. Jos jokin tehtävä on liian suuritöinen, se jaetaan pienempiin osiin, joista arvokkaimmat ja tärkeimmät valitaan kehitysjakson aikana toteutettavaksi. Valintaa varten tilaus- ja ylläpitotehtävien kuvaukset tulee olla niin selkeitä, että niiden perusteella voidaan tehdä mahdollisimman tarkka arvio vaaditusta työmäärästä. Asiakastilauksista vastaavien ryhmien päälliköt priorisoivat tilaukset, ja niistä valitaan niin monta, kuin kahden viikon jakson aikana on ryhmän kesken mahdollista toteuttaa.

Jaksolle valittavien tikettien työmäärissä on otettava huomioon jakson aikana operatiivisen toiminnan vaatimukset, ja sen myötä saapuvat uudet Problem report ja Request for information -tyyppiset tiketit, jotka vaativat usein nopeasti toimenpiteitä. Niiden määrää ja laajuutta ei voida etukäteen tarkkaan tietää, mutta niihin tulee varautua jättämällä niitä varten vapaata työaika. Tämän lisäksi on otettava huomioon työt, joita ryhmän jäsenet tekevät muihin projekteihin, jotka eivät ole näkyvillä ryhmän Jira-projektin kautta.

Mikäli kehitysjakson alussa kanban-työkalulle on jäänyt keskeneräisiä tai aloittamattomia työtehtäviä, ne palautetaan takaisin priorisoitavaksi. On mahdollista, että jotain arvokkaampaa on keksitty tilalle, tai asiakkaan tarpeet ovat saattaneet muuttua, eikä keskeneräiseksi jääneillä tiketeillä olisi enää tarvetta. Jos taas kehitysjaksolle on otettu liian vähän työtehtäviä, niitä voidaan lisätä tarvittaessa priorisoinnin mukaan. Sopiva työmäärä ryhmälle kehitysjakson ajaksi löytyy vain kokemuksen myötä.

Jokaisen kehitysjakson jälkeen ryhmä käy läpi jakson aikana huomatuista onnistumisista ja mahdollisista ongelmista. Toimivat keskeneräisen työn rajoitteet kanban-työkalulle löytyvät vain kokeilemalla, ja jokaisen kehitysjakson eli iteraation jälkeen on tarkoitus pitää katselmointi prosessin toimivuudesta, ja tehdä pieniä muutoksia, joilla pyritään parantamaan prosessia seuraavaa iteraatiota varten. Jakson aikana Jiraan syntyy lisää tietoa eri tehtävien statuksista ja niiden ajallisesta kestosta. Tätä статистиikkaa ryhmä voi käyttää objektiivisena datana, jonka perusteella voidaan tehdä muutoksia kanban-työkalun työmäärärajoitteille.

5.5 Päivittäiset tapaamiset

Kehitysjakson aikana pidetään Scrumista lainattuja päivittäisiä, alle 15 minuutin mittaisia tapaamisia, joissa jokainen ryhmän jäsen kertoo lyhyesti mitä edellisen tapaamisen jälkeen teki, mitä sinä päivänä on tekemässä, ja onko työtä estämässä jotain hidasteita,

jotka vaativat ratkaisua. 10 hengen ryhmässä voi jokaisella puheenvuoro kestää korkeintaan hieman yli minuutin, ja muu keskustelu, ja ongelmien tarkempi selvitys tehdään tämän tapaamisen ulkopuolella.

Tapaamiset pidetään päivittäin aina samaan aikaan ja niihin osallistuvat kaikki ryhmän jäsenet. Tapaamisessa tarkastellaan kanban-aulun näkymästä koko ryhmän tilanne, ja siinä esitettyjä tehtäviä siirretään tarvittaessa sarakkeesta toiselle. Myös ryhmän päivystäjän vuorossa oleva henkilö osallistuu tapaamiseen ja, kuten muutkin, kertoo lyhyesti edellisen tapaamisen jälkeen tehdyt ja sille päivälle suunnitellut työt ja mahdolliset ongelmat.

Päivittäisen tapaamisen ei ole tarkoitus korvata täysin ryhmän pitämää viikoittaista palaveria, jossa on käsitelty tiedotusasioita, lomien suunnittelua ja muita tärkeitä tulevia tai menneitä tapahtumia. Viikkopalaverissa ei olisi kuitenkaan tarpeen käydä samalla tapaa tilannekatsausta kuin päivittäisissä tapaamisissa, ellei viikkopalaveria sitten yhdistetä pidettäväksi heti päivittäisen tapaamisen jälkeen.

5.6 Palautteen kerääminen sidosryhmiltä kehittämistyön tuloksista

Koska Jira on päivittäisessä käytössä koko organisaatiossa, tulevat siihen tehdyt muutokset välittömästi näkyville kaikille ryhmän jäsenille sekä ryhmän Jira-projektia seuraaville sidosryhmille. Kanban-metodin iteratiivinen ja kokeiluun kannustava luonne sopii hyvin nopeiden, pienten muutosten tekemiseen.

Ennen kuin muutoksia tehdään ryhmän käyttämään Jira-projektiin, ehdotetut muutokset esitetään ryhmälle ja tarvittaessa tehdään vielä korjauksia suunnitelmaan. Muutosten jälkeen palautetta on mahdollista antaa ryhmälle mm. pikaviestisovellus Slackin ja sähköpostin kautta. Muiden sidosryhmien, erityisesti yhteyspäälliköiden mielipiteitä uudistuksista toki kaivataan, mutta toisaalta erityistä palautetilaisuutta ei ole tarpeen järjestää, sillä Kanban-metodin periaatteiden mukaisesti ryhmän toimintatapoja ei ole alussa tarkoitus muuttaa, ne on vain tarkoitus tehdä näkyväksi.

Pikaviestisovellus Slack on huomattu nopeaksi ja näkyväksi yhteydenpidon välineeksi, ja tehdyistä uudistuksista kerrotaan Slackin keskusteluryhmän kautta. Saman keskusteluryhmän välityksellä pyydetään mahdollisia parannusehdotuksia. Ryhmän sisäisesti kanban-aulun visualisoinnin paikkansapitävyyttä ja toimintaa tarkastellaan sen omissa viikkopalavereissa ja oman Slack-ryhmän kautta. Mahdolliset ongelmatilanteet korjausehdotuksineen raportoidaan Jira-sovelluksen ylläpidosta vastaavalle ryhmälle, jos tilanne vaatii muutoksia työvuon asetukseen. Muutokset visualisointiin taululla ja

keskeneräisen työn rajat ovat tarvittaessa ryhmän itse muutettavissa esimerkiksi juuri viikkopalavereiden yhteydessä.

6 Toteutus ja tulokset

Tässä luvussa kerrotaan, kuinka ryhmän toteutuva työvuo selvitettiin, ja miten se mallinnettiin Jira-tehtävähallintaan. Luvussa kerrotaan myös, kuinka ryhmässä otettiin käyttöön Daily-tapaamiset.

6.1 Arvovirtakuvaukset

Arvovirtakuvausten tekemiseen varasin jokaisen ryhmän jäsenen kanssa tunnin mittaisen pienoistyöpajan. Nämä työpajat pidettiin kesällä 2021 kahden viikon mittaisen jakson aikana. Kesälomakauden aikana tulosten läpikäynti koko ryhmän kanssa samanaikaisesti ei onnistunut, mutta kerättyä aineistoa katsottiin ryhmäpalaverissa uudelleen lomakauden jälkeen. Tunnin mittaisen työpajan aikana saimme jokaisen ryhmän jäsenen kanssa tehtyä noin kaksi arvovirtakuvausta, joten keskimäärin yhden tekemiseen kului noin puoli tuntia. 10 hengen ryhmän töistä saimme tehtyä 20 kuvausta.

Ryhmän käytössä olevista Jiran tehtävätyypeistä tuli mukaan suurin osa. Aineistoon ei tullut mukaan tehtävätyyppejä Reclamation, Development Idea ja Sub-task. Reclamation-tehtävätyyppi oli poistettu tarpeettomana Jira-projektista jo ennen kuvausten tekemistä, eikä siksi ollut mukana tuloksissa. Development Idea -tyyppisiä tehtäviä ei ollut tekeillä ainoatakaan. Tämän tyyppisiä tehtäviä ei ryhmän toimesta käydä aktiivisesti läpi, eikä niitä siksi päädy työtehtäviksi saakka. Sub-task-tyyppisiä tehtäviä ei erikseen käsitelty, vaikka osa läpikäydyistä tehtävistä niitä olisikin sisältänyt. Tämän takia niitä ei ollut mukana tuloksissa, vaikka myöhemmin, kun Jiran työvuo oli jo muokattu, osa ryhmän jäsenistä kertoi käyttäneensä niille aiemmin laajempaa työvuo (tästä enemmän luvussa 6.2.1).

Aineistoa saatiin sinänsä tarpeeksi, mutta tarkasteltavat työtehtävät olisi voitu valikoida monipuolisemmin. Nyt valittiin kuvausten lähteeksi viimeksi valmiiksi saatuja työtehtäviä, jotta niiden työnkulku olisi tarkemmin kehittäjien muistissa. Tämän lisäksi olisi ollut hyvä koittaa valikoida mukaan enemmän erityyppisiä työtehtäviä. Mukana tuloksissa oli vain yksi Request for Information -tyyppinen tiketti, kun Problem Report, Feature Request ja Maintenance -tyyppisiä oli kuusi, seitsemän ja kuusi kahdestakymmenestä.

Arvovirtakuvaukset sopivat kuitenkin oivasti työvuo selvittämiseen, ja ne toivat erinomaisen hienosti näkyville läpimenoaikojen suuren vaihtelun. Pisimmillään tarkasteltujen tehtävien läpimeno Jiraan kirjaamisesta niiden sulkemiseen oli kestänyt yli 585 päivää, kun nopeimmassa tapauksessa oli aikaa kulunut yhteensä alle kaksi tuntia (Liite 1). Etsin kuvauksissa esiin tulleista työvaiheista yhteneväisyydet, joiden perusteella tein ehdotukset jokaisen tehtävätyypin työvuoksi tilakaavion muodossa (Liite 2). Kaikkia

tehtävätyyppiä ei ollut tuloksissa edustettuna, ja tein niistä ehdotuksen oman harkintani mukaan.

Feature Request, Maintenance ja Problem Report -tehtävätyyppien työvuoto oli keskenään lähes samanlainen. Eri tilat olivat niissä samat, ja eroja tilakaavioissa oli ainoastaan muutamien siirtymien välillä. Esimerkiksi kun Problem Report alun perin luodaan, se menee ensin ryhmän vuorossa olevan päivystäjän arvioitavaksi. Jos päivystäjä katsoo, että ongelma on akuutti, pitää se ottaa heti tehtäväksi. Jos se taas liittyy johonkin toiseen, ei-kiireelliseen projektiin, siihen liitetään linkitys projektiin, ja sen priorisointi arvioidaan myöhemmin projektiryhmän toimesta.

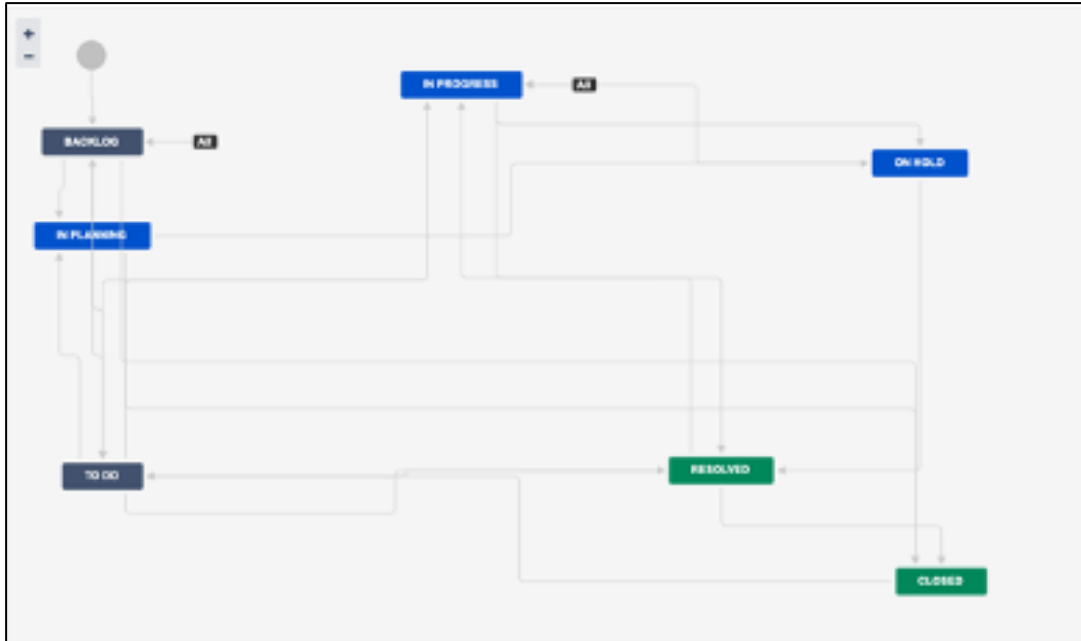
Toinen poikkeava siirtymä oli ylläpidollisissa Maintenance-tehtävätyypeissä, joita ryhmässä on käytetty ennalta suunniteltuja tuotantoon vientejä varten. Niiden tapauksessa varsinainen testaus on jo tehty aiemmin, ja ne toimivat vain eräänlaisena muistilappuna. Samoin nämä tehtävätyyppit kuvaavat tällöin jo itsessään tuotantoon vientiä, jolloin vastaavan niminen tila voidaan ohittaa. Tein lisäksi ehdotelman omasta tuotantoon vienti -tehtävätyyppistä ja sille mahdollisesta työvuosta omana tilakaavionaan. Sellaista ei kuitenkaan nähty ryhmässä tarpeellisena lisätä Jiraan.

Tein ryhmän harkittavaksi myös Development Idea tehtävätyyppille oman tilakaavionsa. Ehdotin, että jos tehtävätyyppiä halutaan jatkossa pitää mukana projektissa, tulisi niiden sisältöä arvioida säännöllisesti ja joko hylätä ehdotukset suljettuna tai muuntaa ne tilaukseksi tai ylläpitotehtäväksi. Hammarberg ja Sunden varoittavat tuomasta parannusehdotuksia tai muita optimointeja mukaan vielä työn visualisoinnin yhteydessä (2014, Luku 2.2). Tämä oli hyvä neuvo, ja oli haastavaa kuvata työnkulkua sellaisenaan arvovirtakuvauksia apunaan käyttäenkin.

6.2 Työvuot ja tilakaaviot

Arvovirtakuvaukset ovat tarkoitettu auttamaan ymmärtämään ryhmän toimintaa ja kokonaisuutta, eikä niitä ole sellaisenaan tarkoitus käyttää kanban-aulun työvuon mallintamiseen. Työvuotoa mallinnettaessa on tärkeää, että se kuvataan sellaisena kuin se oikeasti tapahtuu, mutta jos jokin työvaihe tehdään vain satunnaisesti, sen ei pitäisi olla kanban-aululla. Jos taas jokin työvaihe on nimettynä, on tärkeää, että kaikki ryhmän jäsenet noudattavat työvuotoa ja työkohteet käyvät myös tämän työvaiheen läpi. (Greene ja Stellman 2014, luku 9)

Aiempaan Jiran työvuon kuvaukseen verrattuna (Kuva 3) muutoksia tuli Feature Request, Maintenance ja Problem Report -tehtävätyyppien työvuohon (liite 2).



Kuva 3. Työvuon asetukset Jirassa ennen muutoksia

Työtehtävän statuksen nimi Backlog (suomeksi varasto) muutettiin muotoon Incoming (suomeksi saapuva). Uudella nimellä voidaan paremmin kuvastaa tehtävän tilaa kuin aiemmalla, jolla tarkoitetaan yleensä tekemättömien tehtävien listaa, eikä tehtävän varsinaista statusta. Sekaannusta on myös voinut aiheuttaa se, että backlog-sana on ollut aiemmin käytössä kuvaamaan niin tehtävän tilaa kuin tehtävien listaa Jirassa. Jatkossa backlog-nimellä tarkoitetaan erityisesti Jirassa olevaa tekemättömien tehtävien listaa, erotuksena tehtävän statusta kuvaavasta nimestä.

Erillinen, aiemmin ennakkoon tehtävää suunnittelua varten tarkoitettu In Planning -vaihe poistui käyttämättömänä. Ajatuksena on ollut aiemmin, että työlistalle priorisoituja tehtäviä suunniteltaisiin tarkemmin ennen kehitystyön aloitusta, ja että suunnittelijana voisi toimia joku muu kuin varsinainen kehittäjä. Ryhmälle saapuvat tehtävät on kuitenkin yleensä priorisoitu ja niille on nimetty joku tekijä, joka on aloittanut kehitystyön, kun siihen löytyy aikaa. Erillinen, kehityksestä irrallaan oleva suunnitteluvaihe ei tullut esille yhdessäkään arvovirtakuvauksessa.

Arvovirtakuvauksissa mainittiin useita kertoja koodikatselmointi ja testaus, ja tämän johdosta lisäsin uutena erillisen vaiheen tarkistusta ja testausta varten. Jirassa oli valmiina käytettäväksi asetukset nimeltä Under Review (suomeksi arvioitavana), joka on tarpeeksi väljä määritelmä käsittämään erilaiset testaukset ja katselmoinnit, joita ryhmän eri projekteissa joudutaan tekemään.

Resolved (suomeksi ratkaistu) -tilan sijaan otettiin käyttöön In Deployment (suomeksi tuotantoon viennissä). Erityisesti verkkosivuja ylläpitävässä projektissa on toimittu niin, että kehitetyt ominaisuudet odottavat tuotantoon vientiä, ja tähän oli käytetty Resolved-tilaa. Aiemmin ajatuksena on ollut, että asiakastilaukset merkittiin kehitystyön jälkeen Resolved-tilaan, minkä jälkeen yhteyspäälliköiden tehtävänä olisi ollut sulkea ne lopullisesti asiakashyväksynnän jälkeen. Näin ei kuitenkaan arvovirtakuvausten perusteella ole toimittu, ja edellä mainittu Under Review -tila on siihen parempi.

Request for Information -tehtäviä oli arvovirtakuvauksissa mukana vain yksi kappale, mutta aiemman työkokemuksen perusteella tein ehdotuksen sen tilakaavioksi. Aiempaan työvuohon (Kuva 3) verrattuna se sisälsi vain kolme tilaa:

- Incoming (kysymys on saapunut)
- In Progress (kysymykseen ollaan vastaamassa)
- Closed (vastaus on selvillä)

Epic-tehtävätyyppejä on käytetty ryhmässä kuvaamaan projekteja Jirassa jo olevan projektin sisällä. Niitä ei ollut mukana arvovirtakuvauksissa, ja ehdotin niille sopiviksi tiloiksi Open ja Closed. Ensin ehdotetun tilalle muutin kuitenkin nimet In Progress ja Closed, jottei Jiraan tarvitsisi luoda uusia tila-asetuksia. In Progress tilana kuvastaa kuitenkin hyvin käynnissä olevaa projektia.

Sub-task tehtäviä ei ollut mukana arvovirtakuvauksissa ja niihin ehdottamani työvuoto oli hyvin yksinkertainen: To Do – In Progress – Closed, eli tekemättä – tehtävänä – tehty. Tämä havaittiin jälkepäin ehkä hieman yksinkertaiseksi (luku 5.2.1).

Tein tilakaaviot Atlassianin wikiohjelmiston PlantUML Macro -nimisellä liitännäisellä suoraan sisäiselle wikisivulle, jonka jaoin ryhmän tarkasteltavaksi. Tehdyt kaaviot käytiin läpi ryhmän kanssa erillisessä palaverissa. Palaverissa todettiin muun muassa, että Development Idea on tarpeeton, sillä niitä ei aktiivisesti käydä läpi, ja että se voidaan poistaa järjestelmästä. Kaikki Development Idea -tyyppiset tehtävät pyydetäisiin tulevan muutoksen yhteydessä ylläpidon toimesta muuntamaan Feature Request -tyyppisiksi ja niiden tilaksi tulisi Closed.

Erillistä tuotantoon vientiä kuvaavaa tehtävätyyppiä (liite 2) ei ryhmäläisten mielestä ollut järkevää luoda, ja päätettiin pitää vielä toistaiseksi Maintenance -tyyppiset tehtävät käytössä. Tikettien ehdotetut tilakaaviot vaikuttivat ryhmän mielestä sopivilta muuten, mutta tilojen muutokset toivottiin vapaammaksi siltä varalta, että niitä siirretään esimerkiksi väärälle statukselle. Jiran työvuotoasetuksissa tämä voidaan määrittää niin, että johonkin tilaan siirtymä sallitaan kaikista mahdollisista muista tiloista.

6.2.1 Sub-taskin työvuoto

Kun Sub-task-tehtävien työvuoto oli lyhennetty, huomautti yksi ryhmän jäsen, että heidän projektissaan oli aiemmin käytetty myös tämän tyyppisille työtehtäville laajempaa, Feature Request -tehtävää vastaavaa työvuotoa. Tämä huolimatta siitä, että sub-taskin uutta työvuotoa oli esitelty useita kertoja ennen varsinaista muutosta ryhmäpalavereissa sekä Slackin keskusteluissa. Työvuotoasetukset pidettiin kuitenkin toistaiseksi uudessa muodossaan, ja sitä voidaan ryhmän päätöksestä muuttaa myöhemmin.

6.2.2 On Hold -status

Hammarberg ja Sunden suosittelivat, että syystä tai toisesta estyneitä tehtäviä varten ei lisättäisi omaa On Hold -saraketta kanban-työkalulle. Perusteluinaan he sanovat ongelmatilanteiden ratkeavan nopeammin, kun ne ovat näkyvämmiin esillä ja vaikuttavat keskeneräisen työn rajoitteisiin. (2014 luku 4.2.5)

Tämä suositus on vielä ryhmän harkinnassa, mutta saraketta on käytetty nykyisin paljon. Osa tehtävistä on ns. parkissa, ja mahdollisesti kehitys niiden osalta on jo valmis, mutta ne syystä tai toisesta odottavat tuotantoon vientiä. Tämän suhteen on vielä ryhmän toiminnassa kehittymisen varaa, mutta toisaalta sarakkeen läsnäolo sinällään kuvaa ryhmän nykyisiä toimintatapoja oikein. Toisinaan tehtävät joutuvat odottamaan toisten ryhmien tai yksiköiden toimia, ja yhteistyö niiden välillä on joskus hidasta. Joskus taas tehtävät odottavat vastauksia asiakkaille esitettyihin kysymyksiin.

6.3 Jiran päivitys

Varsinaista muutosta varten tein muutospyyntöni Jiran ylläpidosta vastaavalle ryhmälle. Muutospyyntöön liitettiin mukaan kuvankaappaukset uusista työvuotoasetuksista (liite 3), sekä nykyisten tehtävien tilojen vastaavuudet muutoksen jälkeen (liite 4).

Uusia työvuotoasetuksia varten Jiraan oli lisätty käyttöni testiprojekti, jossa oli mahdollista tietyissä rajoissa tehdä omia kokeiluja ja suunnitella tulevia muutoksia. Jira on kuitenkin käytöltään hyvin rajattu, eikä ilman pääkäyttäjän oikeuksia ole mahdollista mm. tehdä täysin uutta työvuotoa eri tehtävätyypeille edes testiprojektin sisällä. Ilman pääkäyttäjioikeuksia ei ollut myöskään mahdollista tallentaa asetuksia XML-muodossa niin että ne olisi voinut helpommin siirtää toiseen projektiin. Tästä johtuen uudet asetukset tehtiin testiprojektissa yhdelle tehtävätyypille. Uusista asetuksista otettiin ruudunkaappauskuva ja tämän jälkeen asetukset muutettiin toiselle tehtävätyypille sopivaksi.

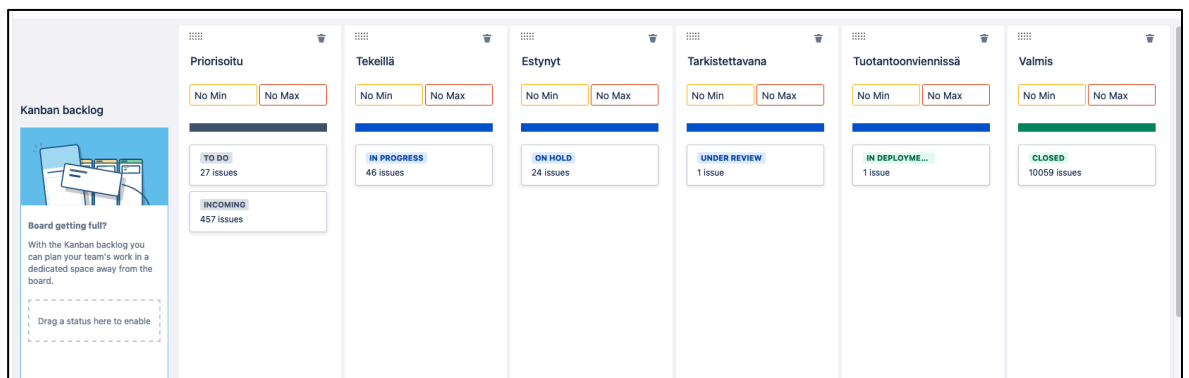
Kun esimerkiksi Epic-tehtävätyypissä tila-asetusten lukumäärä putosi seitsemästä kahteen, tein etukäteen suunnitelman siitä, mitkä olemassa olevat tilat on muutettava toiselle tulevalle tilalle (liite 4). Esitin suunnitelmat ryhmälle Slackin kautta, eikä niitä vastustettu.

Jiran ylläpidosta vastaava ryhmä päivitti muutoksen Jiraan maanantai-iltana, ja kun tiedotin tehdystä muutoksesta seuraavana aamuna Slackin kautta, sain muutaman ilmoituksen ongelmista. Osa Jiraan tallennetuista suodattimista oli rikkoontunut, sillä niissä käytettiin Backlog-statusta tehtävien etsimiseen, ja suodattimeen piti vaihtaa tila-asetukseksi Incoming.

6.4 Kanban-taulujen päivitykset

Jo ennen muutoksia työvuohon otin kehitettäväkseni kaksi kanban-taulua. Toisessa tehtävät esitettiin eriteltynä työntekijäkohtaisille vaakaradoille, joista käytetään myös termiä uimarata, ja toisessa vaakaratoja ei ollut käytössä. Ensin mainittua alettiin käyttää myöhemmin Daily-tapaamisten yhteydessä, sillä siitä näki nopeasti jokaisen viimeaikaiset työtehtävät.

Toisen taulun suunnittelin esittämään koko ryhmän työtilannetta ja tekeillä olevien työkohteiden lukumäärää. Ongelmana oli se, kuinka eri työvuota käyttävät työkohteet saatiin näkymään taululla samaan aikaan. Backlogin esittämisestä vasemmassa sivupalkissa jouduttiin luopumaan, jotta näkyville saataisiin Incoming-tilassa olevat ongelmatiketit, mutta ei incoming-tilassa olevia Feature Request tai Maintenance -tikettejä (Kuva 4). Tämä siksi, koska ongelmatiketit koettiin ryhmässä niin tärkeiksi, että niitä ei ole syytä piilottaa erilliselle backlogin listalle, vaan pitää ne aina näkyvillä.

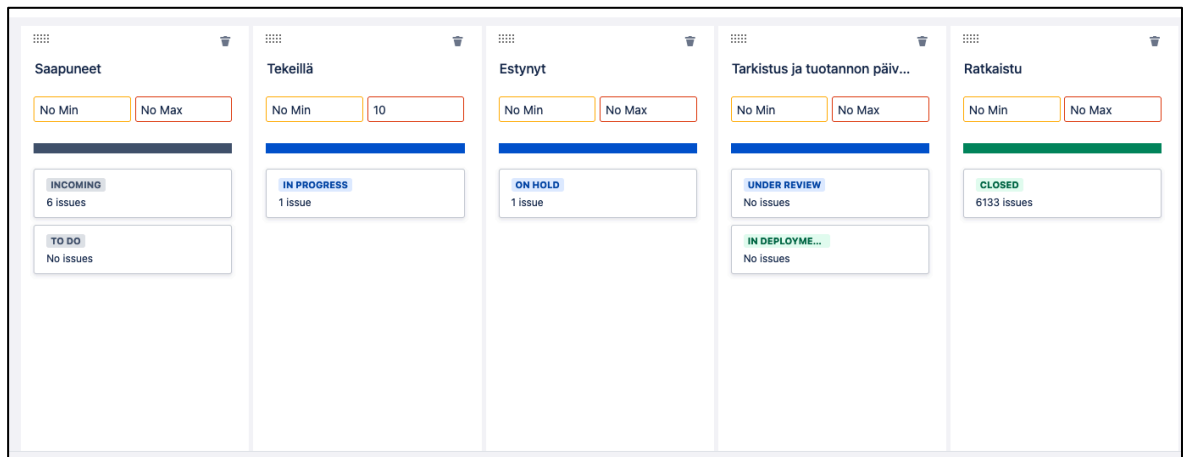


Kuva 4. Kanban-taulun asetukset

Dailyjen yhteydessä käytettävien taulun sarakkeiden ja tilojen vastaavuudet ovat muuten samat, mutta kaikki Incoming-tilan tehtävät pidetään backlogin puolella. Tässä on

perusteluna se, että Dailyssa tarkastellaan meneillään olevia tehtäviä, ja myös Problem Report -tyyppiset näkyvät siinä, jos ne ovat tekeillä tai priorisoituna.

Kolmas kanban-taulu, joka on erityisesti päivystäjän vuorossa olevien käytössä, vaati hieman korjauksia tehtyjen Jira-muutosten jälkeen. Taululla näkyvillä on kaikki Problem Report ja Request for Information tehtävätyypit, eli käytännössä kaikki ne, joista päivystäjä vastaa päivittäisessä työssään.



Kuva 5. Ongelma ja Kysymykset -kanban-taulu

Tutkimuskysymyksessä TK2 mietittiin, miten ryhmän laajojen työtehtävien prosessia voidaan mallintaa Jiran kanban-tilalle, niin että se toimisi informatiivisena lähteenä ryhmän työtilanteesta. Jotta erilaista työvuota käyttäviä tehtäviä voitaisiin esittää samalla kanban-tilalla, tulee niiden kuitenkin käyttää samoja tila-asetuksia. Esimerkissä (Kuva 5 Kuva 8) on sarakkeet kanban-tilalle, jolla esitetään Problem Report ja Request for Information tehtävätyyppejä. Kun ensimmäisellä on käytössään kaikki kuvassa näkyvät tilat, on RFI-tehtävillä käytössään vain kolme näistä. Tästä huolimatta molempien tehtävien kaikki tilat on mahdollista esittää samaan aikaan. Ainoa eroavaisuus tulee Jiran toiminnassa, kun taululla halutaan siirtää tehtäviä sarakkeesta toiseen. RFI-tehtävillä on käytössään vain kolme saraketta, kun tässä tapauksessa Problem Report -tehtävillä on käytössään viisi.

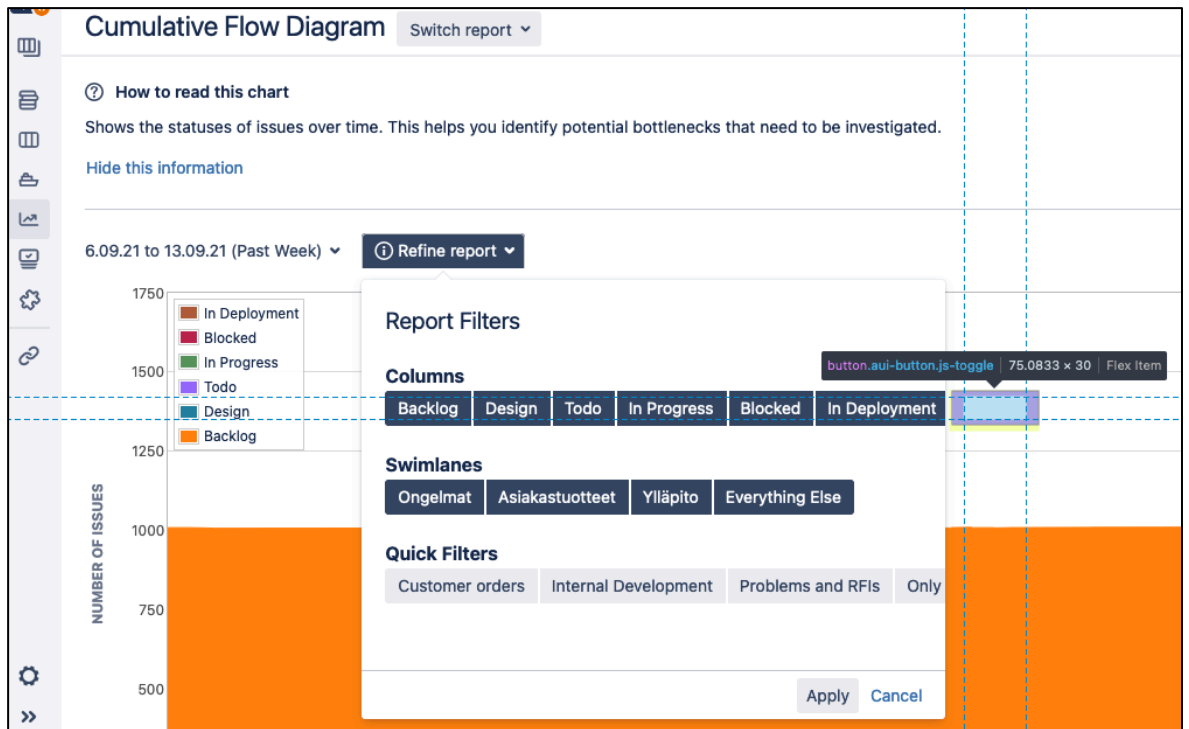
Jira ei kuitenkaan salli kanban-tilalla tehdä siirtymiä saman sarakkeen sisällä. Esimerkissä (Kuva 5) Saapuneet-sarakkeeseen on laitettu tehtäviä, jotka voivat olla Incoming tai To Do -tilassa. Jos tehtävä haluttaisiin päivittää Incoming-tilasta To Do -tilaan pitää se avata esikatseluun ja tehdä muutos Jiran valikoiden kautta. Tämä rajoittaa hieman Jiran käytettävyyttä, mutta kuvattu tilanne on sinänsä poikkeus, joka aiheutuu, kun pitkää työvuota esitetään vähennetyllä sarakkeiden lukumäärällä. Tehdyn visualisoinnin

on kuitenkin tarkoitus elää ja kehittyä, ja sitä on tarkoituksenmukaista ryhmän toimesta päivittää, kun siitä löytyy jotain parannettavaa.

6.5 Mittaristot

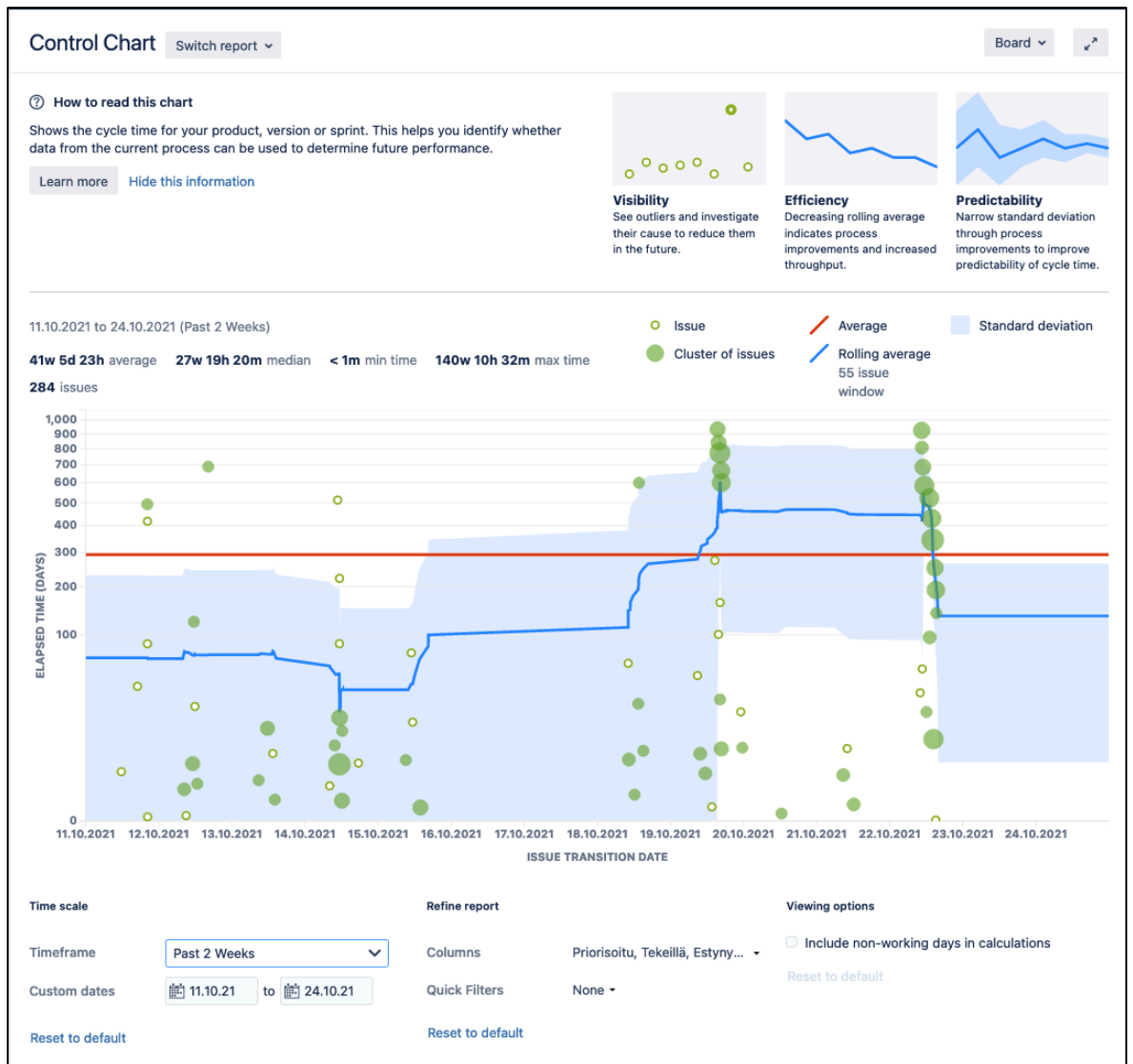
Arvovirtakuvausten perusteella läpimenoajat (tehtävän tallennuksesta sen sulkemiseen valmiina) vaihtelivat paljon. Tarkastelluista tehtävistä jotkin olivat kestäneet yli vuoden ja osa vain muutaman tunnin (liite 1). Suurimmat hidasteet läpimenoaikoihin on havaittavissa ennen varsinaisen kehittämistyön alkua. Pisimmillään tehtävä odotti työn aloittamista 504 päivää, minkä jälkeen tehtävä oli kuitenkin saatu valmiiksi muutaman viikon aikana ja varsinaista työaikaa oli kulunut vain 12 tuntia. Kerätyn aineiston perusteella suurin hidaste on ennen varsinaisen kehitystyön aloitusta. Kun työhön sitten kerran ryhdytään, saadaan ne myös tehtyä kohtuullisessa ajassa.

Läpimenoaikojen mittaaminen Jiran avulla oli myös yksi kehitystehtävän toivotuista lopputuloksista. Tämä osoittautui kuitenkin haastavammaksi kuin olisi toivonut. Jiran kumulatiivisen työn kuvaajan käyttöliittymä ei ole erityisen toimiva. Kaikkia asetuksia ei voida käyttää oikein Jirassa olevan ongelman takia. Asetukset eivät ole käytettävissä, sillä virhe Jiran käyttöliittymässä piilottaa osan valittavista napeista pois näkyviltä (Kuva 6). Asiasta on merkintä Atlassianin (2020) omassa tehtävähallinnassa, mutta siihen ei ole tullut päivityksiä lähes vuoteen ja sen statuksena on ”pitkän aikavälin varasto”. Ongelmaan ei vaikuttaisi olevan odotettavissa korjausta pitkään aikaan, jos ollenkaan.



Kuva 6. Jirassa osa valittavista napeista ei ole näkyvillä

Sen sijaan Control Chart (suomeksi ohjauskaavio) oli toimivampi, ja siinä hallintaan tarvittavat valinnat esitettiin suoraan kuvaajan alapuolella rikkinäisen modaalivalikon sijaan (Kuva 7). Jiran ohjauskaavio esittää työtehtäviä, jotka ovat olleet näkyvillä tietyllä kanban-taululla taulun asetusten mukaan. Kuvaajassa esitetään menneen kahden viikon ajanjaksolla tekeillä olleet ja valmiiksi saadut työtehtävät (Kuva 7). Niiden keskimääräinen läpimenoaika on ollut noin 300 päivää, mutta myös huomattavasti tätä nopeammin on joitain asioita saatu valmiiksi. Lukemia saattaa kohottaa myös esimerkiksi tehtävät, jotka on aikoinaan kirjattu Jiraan, mutta niille ei ole tehty mitään lähes vuoteen, minkä jälkeen ne ovat siirretty vanhentuneina tilaan suljettu. Suuri muutos kuvaajan keskihajonnassa (standard deviation) saatiin aikaan nimenomaan sulkemalla saman päivän aikana useita vanhentuneita ylläpitotehtäviä.



Kuva 7. Ohjauskaavio yleisestä kanban-taulusta

Jos kuvaajasta rajaa pois ylläpitoon kuuluvat, Maintenance-tyyppiset tehtävät (Kuva 8), voi keskiarvon huomata putoavan jo alle 50 päivään, ja voisi siis sanoa asiakastilausten valmistuneen keskimäärin viidessä viikossa. Tämä alittaa selvästi luvatus kolmen kuukauden toimitusajan.



Kuva 8. Feature Request -tehtävien edistyminen kahden viikon aikana.

Huomattavaa myöskin on, että vaikka keskiarvo on ollut useita viikkoja, on mediaani vain kolme päivää ja 13 tuntia. Keskiarvoa nostavia tapauksia on mahdollista katsoa tarkemminkin, ja tarkasteltavaksi valitsin satunnaisen kuvaajalla esitetyn Feature Request -tyyppisen työtehtävän, jonka läpimenoaika oli yli 300 päivää.

Tarkasteltu työtehtävä (Kuva 9) oli kirjattu Jiraan, missä se oli ollut tehtävävarastossa (Backlog) yli 117 päivän ajan ennen kuin se oli merkitty aloitetuksi (In Progress). Kun työtehtävä oli merkitty aloitetuksi, kesti 194 päivää ennen kuin se oli merkitty valmiiksi (Resolved). Tehtävää ei ollut tämän jälkeen lopullisesti suljettu, vaan se oli odottanut tilassa 174 päivää, kunnes Jirassa oli työvuon muutosten yhteydessä vaihdettu Resolved-tilan nimeksi In Deployment. Vasta tämän jälkeen tehtävä oli suljettu kokonaan, kun ryhmän työntekijä oli käynyt vanhoja merkintöjä läpi ja sulki pois suuren määrän vanhoja jo aiemmin valmiiksi saatuja tehtäviä.

Activity

All Comments Work Log History Activity Transitions Notification History

Transition	Time In Source Status	Execution Times
made transition - 08.10.2020 9:42 BACKLOG → IN PROGRESS	117d 22h 23m	1
made transition - 20.04.2021 11:14 IN PROGRESS → RESOLVED	194d 1h 31m	1
made transition - 11.10.2021 20:18 RESOLVED → IN DEPLOYMENT	174d 9h 3m	1
made transition - 12.10.2021 11:10 IN DEPLOYMENT → CLOSED	14h 52m	1

Kuva 9. Tilan muutokset läpimenoajan keskiarvoa nostaneella tehtävällä

On mahdotonta jälkikäteen päätellä, paljonko aikaa tehtävän varsinaiseen suorittamiseen oikeasti kului. Tehtävän tilojen siirtymät vaikuttavat kuitenkin kuvaajien keskiarvoon, ja 174 päivän odotus valmiissa tilassa ei ole varsin pientä. Pitkä odotusaika ennen varsinaisen työn aloitusta kuvaa varsin hyvin toteutuvaa työvuota.

Jotta Jiran kuvaajia voisi paremmin käyttää todellisen työtehon mittaamiseen, pitää sinne kirjattujen tehtävien tietoja päivittää huolellisemmin ja säännöllisesti. Ryhmässä on ehdotettu viikoittaista tehtävähallinnan katselmointia, jonka yhteydessä suljettaisiin pois vanhentuneita tehtäviä. Tämän lisäksi on kuitenkin tarpeen vaatia jokaiselta käyttäjältä omien tehtäviensä ylläpitoa, mihin voi saada apua jo rajoittamalla samanaikaisten työtehtävien lukumäärää. Myös tehtävien laajuus olisi syytä pitää mahdollisimman samanlaisena ja tarvittaessa pilkkoa suuret työt pienempiin osiin, jotta niitä voidaan vertailla keskenään.

6.6 Ketterien metodien käyttö ryhmässä

Ryhmässä otettiin kokeiluun päivittäiset lyhyet tapaamiset ja käytäntöä päätettiin jatkaa. Nämä päivittäiset tapaamiset koettiin ryhmässä hyödyllisiksi, sillä ne nopeuttivat avunsaantia, jos jonkin työtehtävän kanssa oli kysymyksiä tai muita ongelmia. Ongelmallisinta oli päivittäisen tapaamisen aikataulutus. Liukuvan työajan puitteissa osa ryhmäläisistä aloittaa päivänsä useita tunteja aiemmin tai myöhemmin kuin muut ja jotta tulevan päivän toimiin voisi parhaiten saada apua, on tapaaminen hyödyllisin työpäivän alussa.

Ensimmäisenä kokeiltiin Slackin kautta pidettävää tekstimuotoista Dailya, jossa viestiketjuun jokainen kirjoittaisi omat tekemisensä ja kertoisi mahdollisista haasteista. Tämä palaverin muoto todettiin läsnä olleiden kesken yksimielisesti huonoksi ja tapaaminen päätettiin jatkossa pitää videopuheluna Slack-sovelluksen välityksellä.

Läheskään aina eivät kaikki ryhmäläiset pääse osallistumaan päivittäiseen tapaamiseen johtuen sitten muista päällekkäisistä palavereista ja tekemisistä. Ryhmässä päätettiin, että Dailyn käynnistää Slack-puheluna se, kuka ensimmäisenä ehtii. On kuitenkin vaarana, että tehtävä, joka on kaikkien, mutta samaan aikaan ei kenenkään vastuulla, jää tekemättömäksi. Olisi selkeämpää, että päivittäisen tapaamisen aloittaisi aina joku nimetty henkilö. Päivittäisen tapaamisen käytäntö kokee varmaan vielä muutoksia, jos tai kun etätyön määrä ryhmässä muuttuu riippuen koronapandemian kehittymisestä.

Keinotekoinen aikarajaus ryhmän työlle aloituspalaverin ja retrospektiivin kanssa todettiin ryhmäpalaverissa hankalaksi järjestää, koska ryhmän työ sisältää paljon jatkuvaluonteista ylläpitotyötä. Kanban-metodi ei suoraan tällaista vaadi, ja sen 5. käytäntö palautesilmukan luomisesta voidaan periaatteessa toteuttaa ryhmän viikkopalaverin yhteydessä, jos sen ajasta varataan osa ryhmän toiminnan tarkasteluun ja kehitykseen.

6.7 Ehdotukset kehittämistoimenpiteiksi

Päivittäisiä tapaamisia kannattaa jatkaa, sillä ne on koettu ryhmässä hyödyllisiksi, eivätkä ne vie liikaa työaikaa pois. Ryhmän kannattaa viikoittain tarkastella omaa toimintaansa esimerkiksi Jiran ohjauskaavion kautta, jotta toimintaa voidaan tarkastella objektiivisesti. Kuvaajien tulosten perusteella on syytä harkita muutoksia ryhmän toiminnassa ja ne olisi syytä tehdä käyttämällä Kanban-metodin keskeneräisen työn rajoituksia.

Tehtävävaraston ylläpito tulisi ottaa osaksi ryhmän päivittäistä tai viikoittaista työtä. Jiran tarjoamien mittaristojen hyödyllisyys riippuu paljon siitä, miten hyvin tehtävähallintaa pidetään ajan tasalla. Kerran vuodessa tehtävä vanhentuneiden tehtävien siivous vaikuttaa paljon kuvaajien (kuva 4) lukemiin ja saattaa hämärtää tuloksia. Jos niitä taas käydään läpi säännöllisesti tasaantuvat kuvaajat ajan kanssa ja niistä voidaan paremmin hahmottaa normaalin päivätyön vaatimat ajat. Tämä ei kuitenkaan tarkoita, että pitäisi erityisesti varoa kirjottamasta Jiraan muistiin mahdollisesti hyödyllisiä kehitystehtäviä, vaan sitä, että niitä on käytävä säännöllisesti läpi.

Ryhmän kannattaa jatkossa tutkia tarkemmin On Hold -sarakkeessa olevia tehtäviä ja selvittää yhdessä viivästysten syitä. On mahdollista löytää kehityskohteita toisten ryhmien ja yksiköiden välisestä yhteistyöstä ja yhteyspäälliköiden ja asiakkaiden kanssa käydystä

kommunikaatiosta. Ongelmia on syytä ratkoa niin, että yhteisiä työtapoja muuttamalla viivästyksille saataisiin pysyviä ratkaisuja.

Ryhmässä innostutaan hyvin käytännön tehtävistä, mutta olisi myös hyvä olla tietoinen eri käytäntöjen taustalla olevasta teoriasta, jotta ymmärrettäisiin, miksi käytännöt on luotu. Kanban-metodia ja muita ketterän kehityksen teorioita ja käytäntöjä olisi hyvä tuoda paremmin ryhmän tietoisuuteen esimerkiksi erilaisten kurssien tai koulutusten avulla. Kun useammalla on sama käsitys käytetyistä metodeista ja niiden taustalla olevista teorioista, voi olla helpompi saada aikaan muutoksia organisaatiossa.

7 Johtopäätökset

Tässä luvussa tarkastellaan, miten kehitystehtävällä saavutettiin luvussa 2 mainitut odotetut lopputulokset ja vastataan luvussa esitettyihin tutkimuskysymyksiin. Omana alalukunaan on pohdintaa siitä, miten kehitystyötä kannattaa jatkaa ja lopuksi omakohtainen reflektointi työn tekemisestä.

7.1 Saavutetut tulokset

Odotettu lopputulos 1. Ryhmän käytössä olevan Jira-projektin toiminnan tulee vastata ryhmässä toteutuvaa toimintaa.

Työn aikana tehtyjen muutosten avulla saatiin ryhmän Jira-projekti vastaamaan suurin piirtein ryhmässä toteutuvaa työvuota. Ryhmän työt ovat erittäin monipuolisia ja Jirassa yksittäinen projekti ei taivu helposti kuvaamaan lukuisia pienempiä projekteja. Tilanne on kuitenkin muutosten jälkeen parempi, ja käytettävät tilat ja työvuot ovat nyt samat mitä tehtyjen kuvausten perusteella pääteltiin.

Odotettu lopputulos 2. Käytettävissä on Jiran kanban-taulu, joka kuvastaa ryhmän todellista työtilannetta ja toimii ajantasaisena tiedonlähteenä myös muille sisäisille sidosryhmille.

Ennen kehitystehtävää eivät kanban-työkalut olleet koko ryhmällä päivittäisessä käytössä, mutta Daily-tapaamisten aikana erityisesti kehittäjäkohtaisilla uimaradoilla varustettu taulu on tullut säännölliseen käyttöön. Se kuvaako taulu ryhmän todellista työtilannetta riippuu siitä, kuinka hyvin ryhmän jäsenet pitävät huolen tehtävien tilojen päivityksistä. Taulun sarakkeet vastaavat kuitenkin aiempaa paremmin ryhmän toimintatapoja.

Odotettu lopputulos 3. Ryhmän käytössä on Jirassa mittaristo, jolla voidaan objektiivisesti mitata ryhmän työtehoa tehtävien läpimenoaikojen muodossa.

Jiran tarjoamat mittaristot toimivat hieman hankalasti, mutta ohjauskaavio oli käytettävissä kohtuullisen helposti, ja sen avulla on mahdollista mitata objektiivisesti ryhmän työtehoa. Kuvaajien lukemissa on kuitenkin yhä paljon heittelyä, koska ryhmän projektiin on kirjattuna erittäin suuri määrä tehtäviä, joista paljon on ajan kanssa käynyt tarpeettomaksi. Tämä mittaristo on ryhmän vapaasti käytettävissä, mutta se ei ole vielä tarkoittanut, että sitä säännöllisesti käytettäisiin ohjaamaan ryhmän toimintaa. Jotta siitä olisi enemmän apua, se kannattaa ottaa mukaan viikoittaiseen ryhmäpalaveriin ja yhdistää sen käyttö keskeneräisen työn rajoitteiden käyttöön.

Odotettu lopputulos 4. Ryhmä käyttää kanban-taulun keskeneräisen työn rajoituksia hallitsemaan työvirtaa.

Keskeneräisen työn rajoitteita ei kehitystehtävän aikana otettu varsinaisesti käyttöön. Rajoitukset taululla ovat olleet olemassa, mutta niiden ylittyessä ei asiaan ole kiinnitetty huomiota. Tämä on sikäli harmi, sillä kirjallisuudessa rajoitusten tarpeellisuutta korostettiin.

7.2 Vastaukset tutkimuskysymyksiin

TK1 Minkälainen työvuo on nykyisin ohjelmistokehitysryhmän työtehtävillä?

Vastaus ensimmäiseen tutkimuskysymykseen saatiin näiden arvovirtakuvausten ja niiden perusteella tehtyjen tilakaavioiden muodossa (Liite 4). Ongelmanratkaisu- ja ylläpitotehtävät ja asiakastilaukset saivat kaikki käyttöönsä saman työvuoasetuksen Jirassa. Käytännössä näiden välillä on kuitenkin eroavaisuuksia. Esimerkiksi kriittiset ongelmanratkaisutehtävät otetaan tarvittaessa välittömästi työn alle ilman asiakastilausten tapauksessa läpikäytävää priorisointia. Ylläpitotehtävissä on toisinaan erityisiä koonteja muiden tehtävien tuotantoon vienneistä, jolloin työvuo on suoraviivaisempi.

Sub-task, Epic ja Request for Information -tyyppisillä tehtävillä havaittiin edellisiin verrattuna yksinkertaisempi työvuo, ja Development Idea -tehtäville ei havaittu nykyisellään käyttötarvetta lainkaan ja nämä poistettiin ryhmän käytöstä.

TK2 Miten ryhmän laajojen työtehtävien prosessia voidaan mallintaa Jiran kanban- taululle, niin että se toimisi informatiivisena lähteenä ryhmän työtilanteesta?

On mahdollista tuottaa Jiraan lukuisia erilaisia kanban- tauluja, joista jokainen palvelee tiettyä käyttötarvetta. Ryhmän päivittäisiä tapaamisia varten tehtävät ryhmitellään taululla tekijöiden mukaan, jotta tapaamisen aikana on helpompi käydä järjestyksessä jokaisen töitä. Päivystäjän vuorossa oleva henkilö taas tarvitsee kanban- taulun, jossa esillä ovat ainoastaan ongelmanratkaisutehtävät ja ryhmälle esitetyt kysymykset.

Ryhmän työtilanteesta kokonaisuutena on myös oma kanban- taulu. Siinä näkyvillä on kaikki tehtävät pois lukien priorisoimattomat asiakastilaukset ja ylläpitotehtävät. Taulu kuvaa työtilannetta sinänsä oikein, mutta vielä puuttuvien WIP- rajoitteiden takia taululla on tarpeettoman paljon tehtäviä. On kätevää, että eri tarpeita vastaavat taulut saa Jiran avulla esille suhteellisen helposti. Vastaavien taulujen tekeminen paperisten tehtävälappujen muodossa ja niiden ylläpitäminen voisi olla hankalaa.

TK3 Mikä on nykyisten työtehtävien läpimenoaika ja mitkä asiat hidastavat läpimenoa?

Arvovirtakuvausten perusteella läpimenoajat (tehtävän tallennuksesta sen sulkemiseen valmiina) vaihtelevat todella paljon. Osalla tehtävistä läpimeno kestää yli vuoden, toisilla tehtävillä läpimenoajat lasketaan tunneissa. On ymmärrettävää, että esimerkiksi äkillisestä ongelmatilanteesta aiheutuva työtehtävä on käyty läpi erittäin nopeasti.

Arvovirtakuvausten perusteella suurin hidaste on prosessin alkupuolella, kun Jira-projektiin lisätyt tehtävät odottavat vuoroaan joissain tapauksissa useita kuukausia. Tilanteeseen voi odottaa parannusta, jos tehtävävaraston ylläpitoon kiinnitetään ryhmässä enemmän huomiota, ja sieltä saadaan siivottua tarpeettomia asioita pois ja priorisoitua tärkeämpiä tehtäviä korkeammalle.

Jiran ohjauskaavion avulla saa tarkemman kuvan tehtävien läpimenoajoista, mutta tässäkin vielä toistaiseksi tuloksia voi vääristää Jiraan unohdetut, tarpeettomat tehtävät. Asiakastilausten mediaaniksi saatiin kahden viikon tarkastelujaksolla kolme päivää ja 13 tuntia, kun niiden keskiarvo oli yli viisi viikkoa.

TK4 Voidaanko Kanban-metodin, aikarajatun kehityksen ja päivittäisten tapaamisen avulla parantaa ryhmän työtehoa?

Jiran tarjoamien kuvaajien perusteella ei ole vielä havaittu varsinaisesti muutosta ainakaan tehtävien läpimenoajoissa. Kysymystä olisi ollut ehkä syytä rajata vielä tarkemmin. Jos muutosta työtehosta olisikin havaittavissa, on vaikea sanoa varmuudella, onko parannuksen syy erityisesti tutkimuskysymyksessä mainituissa seikoissa. Syytä ajankäytön seurantaan kuitenkin on, ja apuna siihen kannattaa käyttää Jiran tarjoamia kuvaajia ja Kanban-metodin keskeneräisen työn rajoitteita.

7.3 Pohdinta

Työtehtävien kulkua olisi tarpeen selvittää vielä tarkemmin erityisesti työvuon alkupäästä. Nyt selvitettävänä oli ryhmän sisäinen prosessi, mutta tehtävien priorisoinnin suhteen jäi vielä kysymyksiä auki. Jos prosesseista halutaan Kanban-metodin mukaisesti eksplisiittisiä, tulee vielä tutkia, miten, kuka ja milloin työtehtävien priorisointia tehdään ja minkälaisin perustein. Keskustellessani yhteyspäälliköiden kanssa olemme huomanneet, ettei heilläkään ole aina tietoa koska tai miten työtehtävät valikoidaan ryhmän työlialle. Tarvitaan selkeä kuvaus siitä, miten ja mitkä tehtävät valikoidaan työhön seuraavaksi ja kuka valinnan tekee ja koska.

Asiakastilausten toimitusajaksi on aiemmin luvattu yksi kuukausi tilauksen saapumisesta, mutta tähän tavoitteeseen ei aina ole päästy, ja toimitusajan lupaus on sittemmin kasvatettu kolmeen kuukauteen. Onko mielekästä luvata erikseen kuukauden tai kolmen kuukauden toimitusaikaa ylipäättänsä? Ryhmälle tulevien tehtävien laajuus vaihtelee paljon, ja vaikka itse työn tekemiseen kuluisi vain tunteja tai päiviä, voi työn aloittaminen venyä. Voidaanko kolmen kuukauden toimitusajan sanoa olevan erityisen pitkä tai sopivan lyhyt. Jatkotutkimusta voisi tehdä siitä, kuinka pitkiä toimitusajat ovat yleensä tämän tyyppisessä ohjelmistokehityksessä. Ehkä benchmark-tyyppinen tutkimus voisi sopia tällaisen asian selvittämiseen. Organisaatio tekee paljon kansainvälistä yhteistyötä eri tahojen kanssa ja vertailukohtia voisi etsiä myös ulkomailta.

Päivittäisten tapaamisten osallistujamäärä vaihtelee paljon päivittäin. Osalla ryhmän jäsenistä on muita velvoitteita, eivätkä sen tähden pääse osallistumaan ryhmän omaan tapaamiseen, mutta osallistuminen on voinut jäädä väliin myös muista syistä. Päivittäinen tapaaminen on nähty ryhmässä kuitenkin hyödyllisenä, ja ehkä vastaavaa voisi viedä käyttöön myös suurempiin, pitkäkestoisiin projekteihin, joihin ryhmän jäsenet osallistuvat.

Kehitystyötä erityisinä aikarajattuina jaksoina 5. luvun kuvauksen mukaisesti ei kehitystehtävän aikana tehty. Päivittäiset tapaamiset ja niiden lisäksi pidettävät viikkopalaverit sinänsä jaksottavat työtä, mutta siihen olisi syytä liittää mukaan töiden priorisointi tulevalle viikolle, jotta tehtäviin kulunutta aikaa voitaisiin helpommin seurata. Kokemus töihin kuluneen ajan tarkemmasta seurannasta voisi myös helpottaa ryhmässä tehtyjen työaika-arvioiden antamista jatkossa.

Työn aikana huomasit useaan otteeseen kuinka hankalaa Jiran asetusten muokkaaminen on. Useita asetuksia on rajattu pääkäyttäjäoikeuksien taakse, eikä Jiraan lisätyn projektin omistajalla ole mahdollisuutta tehdä projektikohtaisia muutoksia useisiin asetuksiin. Syy ohjelmiston kankeudelle on varmaan sen iässä. Jira julkaistiin alun perin jo lähes kaksikymmentä vuotta sitten, mutta toisaalta sillä on ollut jo lähes kaksikymmentä vuotta aikaa kehittyä paremmaksi. Jiran käyttötavat ovat muuttumassa, eikä sille ole enää vuoden 2024 tarjolla nykyisen kaltaista paikallisesti hallittua asennusta, vaan sen tilalle tulee pilvipalvelu. Onko tulevaisuudessa mahdollista, että ryhmässä siirrytään käyttämään kokonaan muuta tehtävähallintaa, ja mikä se voisi olla? Ryhmä on joissain projekteissa käyttänyt GitHubin tehtävähallintaa, mutta toisaalta se, että ryhmän käytössä on ollut kaksi eri tehtävähallintaa, on ollut hankalaa. On varmasti haastava löytää ratkaisu, joka toimisi hyvin kaikille, mutta jokin muutos tulevaisuudessa vääjäämättä siintää.

7.4 Omakohtainen reflektio työn tekemisestä

Kehitystehtävässä tutkittiin ohjelmistokehityksen töiden läpimenoaikoja, ja tulosten perusteella joissain tehtävissä kesti hyvinkin pitkään. Myös kehitystehtävän tekemisessä kesti pitkään ja alkuun pääseminen oli haastavaa. Ensimmäinen ideointipalaveri aiheesta käytiin jo kesällä 2019. Palaverissa hieman ehkä hauskasti pohdittiin voiko Jiran kautta käytettävä kanban-taulu olla ”hopealuoti”, jolla saataisiin helpotusta tehtävien kaaokseen. Hopealuodiksi Kanbanista ei ollut, eikä sitä sellaiseksi ole tarkoitettukaan. Se toimii kuitenkin hyvänä aloituspisteenä ryhmän toiminnan kehittämiseksi, ja olen toiveikas, että sen avulla saadaan tuotua näkyville erilaiset ryhmän työhön vaikuttavat pullonkaulat ja solmukohdat.

Haastavinta kehitystehtävässä oikeastaan oli sen aikatauluttaminen. Vaikka suunnitelmia työn eri vaiheiden ajoittamiseksi oli tehnyt, ne eivät useinkaan pitäneet ja muut kiireet töissä tai vapaa-ajalla ajoivat sen yli. Aikataululle ei toimeksiantajan puolesta ollut erityisiä vaatimuksia, minkä voi nähdä hyvänä ja huonona asiana. Koska Kanban-metodi on jatkuvan parantamisen prosessi, eikä sille ole varsinaisesti loppua, olisi kehitystehtävää pitänyt rajata tarkemmin myös ajallisesti. Opinnäytetyössä olisi varmasti ollut hyödyksi säännöllinen etenemisen seuranta oppilaitoksen ja toimeksiantajan kanssa, mutta sain tehdä kehitystehtävää melko omatoimisesti. Opinto-oikeuden rajallinen kesto määräsi kuitenkin lopulta raamit myös kehitystehtävän kestolle.

Lähteet

Agile Alliance. Backlog Refinement. Luettavissa:

<https://www.agilealliance.org/glossary/backlog-grooming>. Luettu: 23.10.2021.

Analysis.Net Research. 2020. 14th Annual State of Agile Report. Digital.ai. Luettavissa:

<https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494>.

Luettu: 31.10.2020.

Anderson, D. 2011. Kanban. dpunkt. E-Kirja. Luettu: 17.10.2020.

Anderson, D. 18.3.2020. The Principles and General Practices of the Kanban Method. David J Anderson School of Management. Blogi. Luettavissa: <https://dja.com/the-principles-and-general-practices-of-the-kanban-method/>. Luettu 18.10.2020.

Atlassian. 2021. Atlassian Server end of life (sale/support) information. Luettavissa:

<https://www.atlassian.com/migration/assess/journey-to-cloud>. Luettu 11.10.2021.

Atlassian. 2020. Refine Filters Modal dialog hides buttons on Cumulative Flow report.

Atlassianin julkinen tehtävähallinta. Luettavissa:

<https://jira.atlassian.com/browse/JSWCLLOUD-20586>. Luettu: 9.13.2021.

Greene, J & Stellman, A. 2014. Learning Agile. O'Reilly Media, Inc. E-Kirja. Luettu 10.9.2020.

Inzaurraga, J. 7.8.2017. Can we truly be Agile in maintenance projects? Hexacta. Blogi.

Luettavissa: <https://www.hexacta.com/can-we-truly-be-agile-in-maintenance-projects/>.

Luettu 6.11.2021.

Hammarberg, M & Sunden, J. 2014. Kanban in Action. Manning Publications. E-Kirja.

Luettu: 18.10.2020.

Hammarberg M. 2013. Some tools for improved focus, improve teamwork and faster delivery. CodeBetter.Com. Luettavissa:

<http://codebetter.com/marcushammarberg/2013/08/13/some-tools-for-improved-focus-improve-teamwork-and-faster-delivery/>. Luettu 23.10.2021.

McHenry, B., Will S. Identify bottlenecks with value stream mapping. Luettavissa:

https://www.ibm.com/garage/method/practices/discover/practice_value_stream_mapping.

Luettu 5.6.2021.

Ojasalo, K., Moilanen T. & Ritalahti, J. Sanoma Pro Oy. Helsinki. 2015. Kehittämistyön me-netelmät. Uudenlaista osaamista liiketoimintaan.

Poppendieck, M. & Poppendieck, T. 2003. Lean Software Development: An Agile Toolkit. Addison-Wesley Professional. E-Kirja. Luettu: 10.10.2020.

Poppendieck, M & Poppendieck, T. 2006. Implementing Lean Software Development: From Concept to Cash. Addison-Wesley Professional. E-Kirja. Luettu: 20.6.2021.

Shore, J., Warden, S. 2021. The Art of Agile Development, 2nd Edition. O'Reilly Media, Inc. E-kirja. Luettu 1.11.2021.

Using the Kanban Method. Luettavissa:

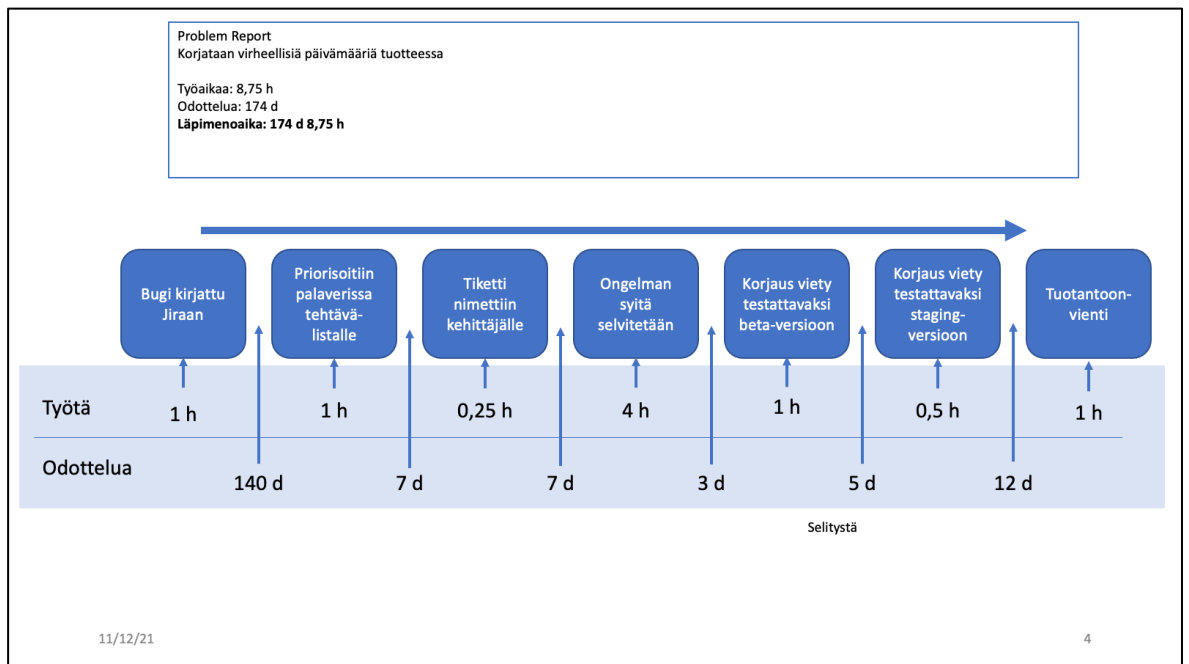
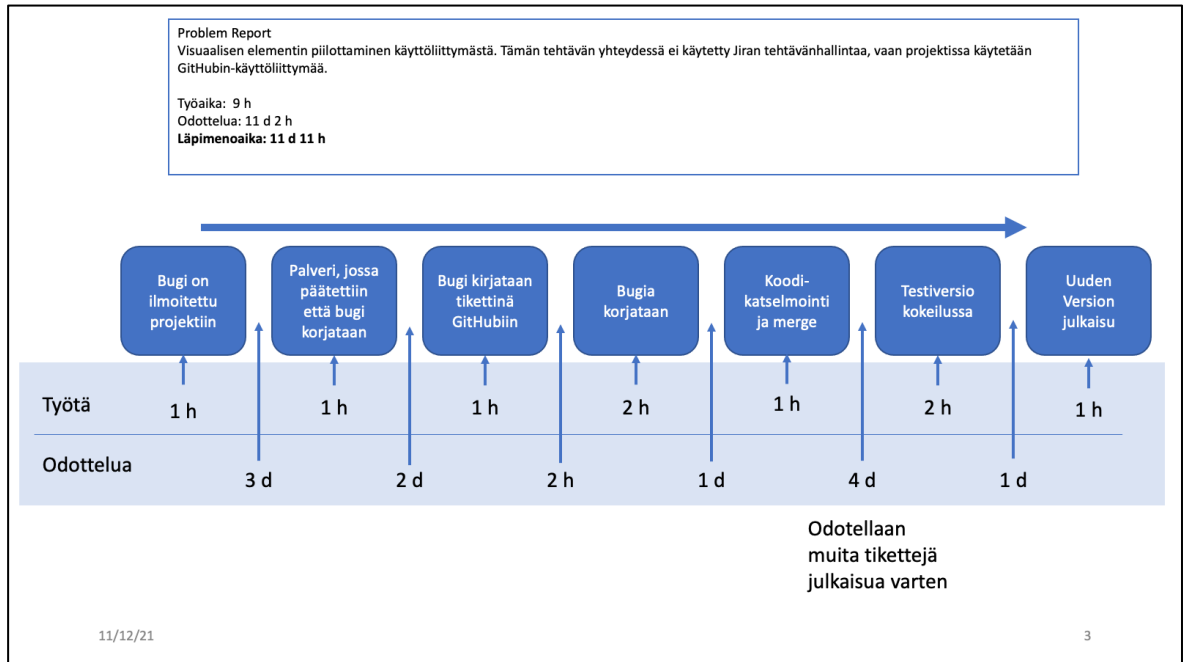
<https://web.archive.org/web/20201214151832/https://groups.yahoo.com/neo/groups/kanbandev/info>. Luettu 18.10.2020.

Van Rooden, S. 2016. Product Backlog Refinement explained (1/3). Scrum.org.

Luettavissa: <https://www.scrum.org/resources/blog/product-backlog-refinement-explained-13>. Luettu: 23.10.2021.

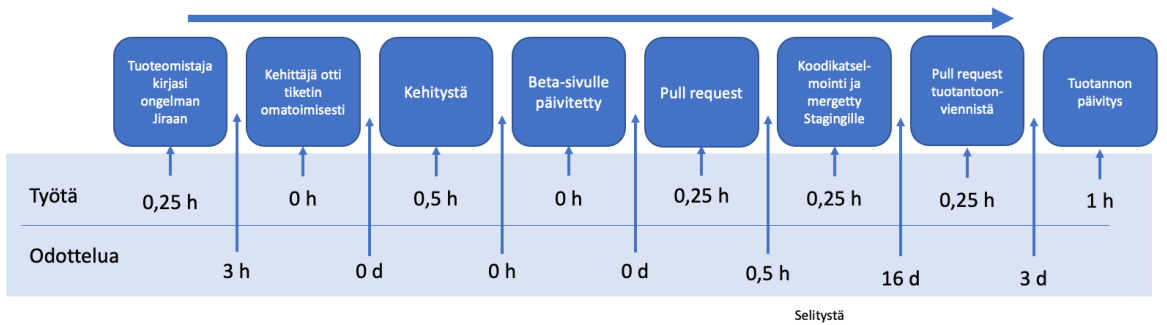
Liitteet

Liite 1. Arvovirtakuvaukset



Problem Report
Verkkopalvelussa lokalisoinnin korjaus.

Työaika: 2,25 h
Odottelua: 19 d 3,5 h
Läpimenoaika: 19 d 5,75 h

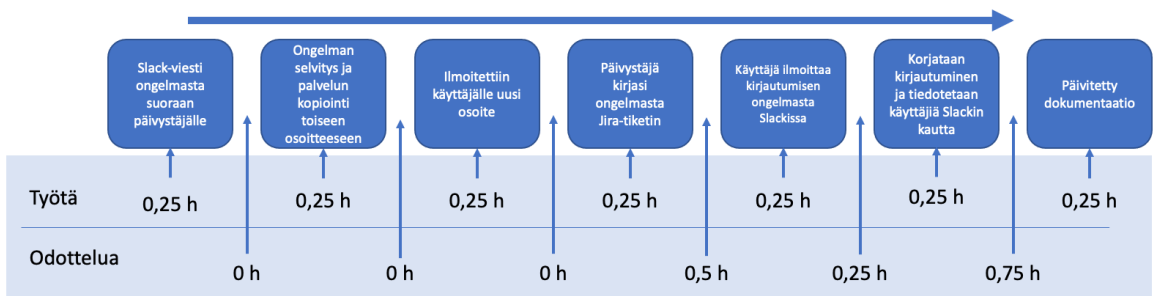


11/12/21

5

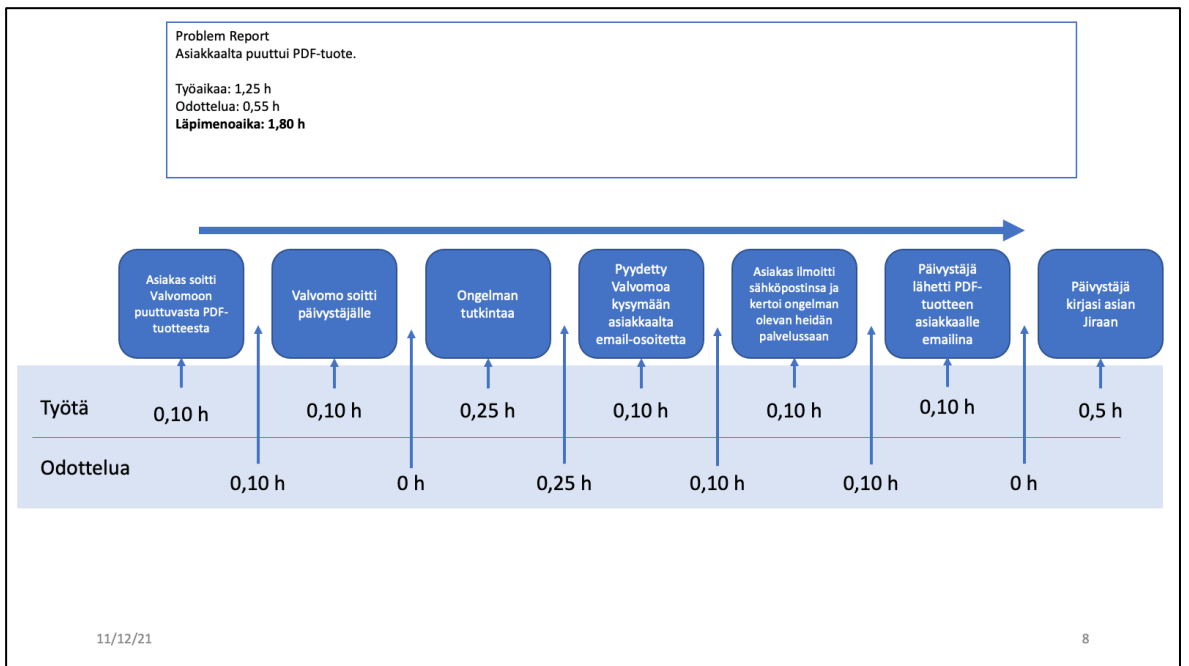
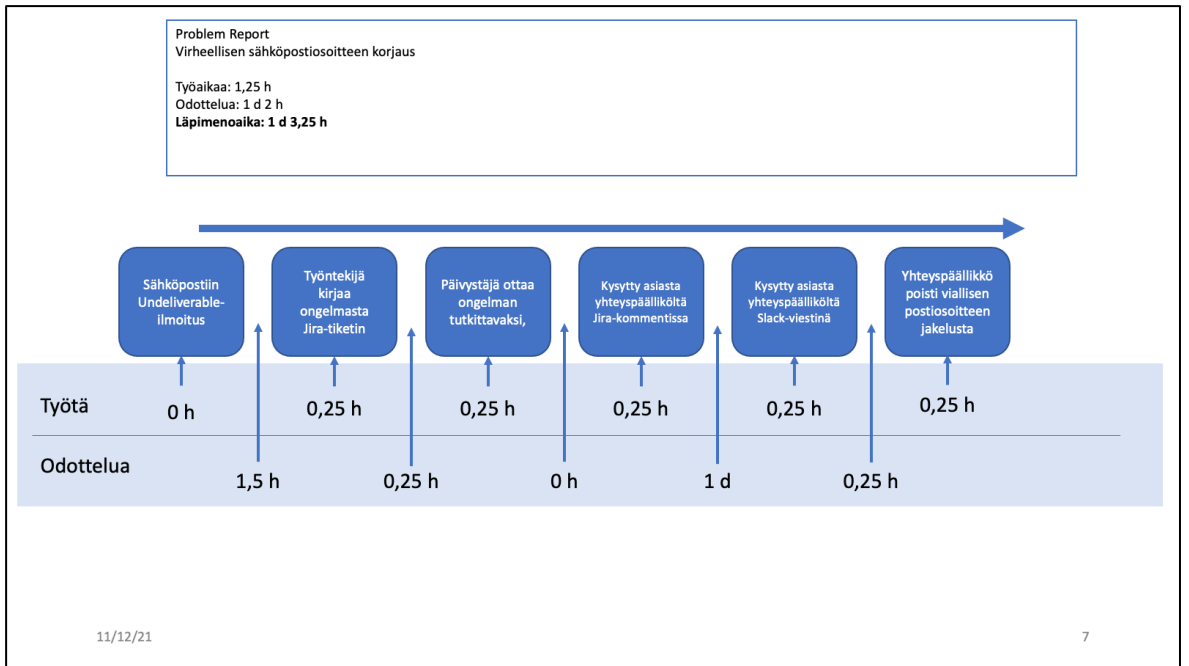
Problem Report
Verkkopalvelu oli kadonnut käytöstä ja palautettiin takaisin.

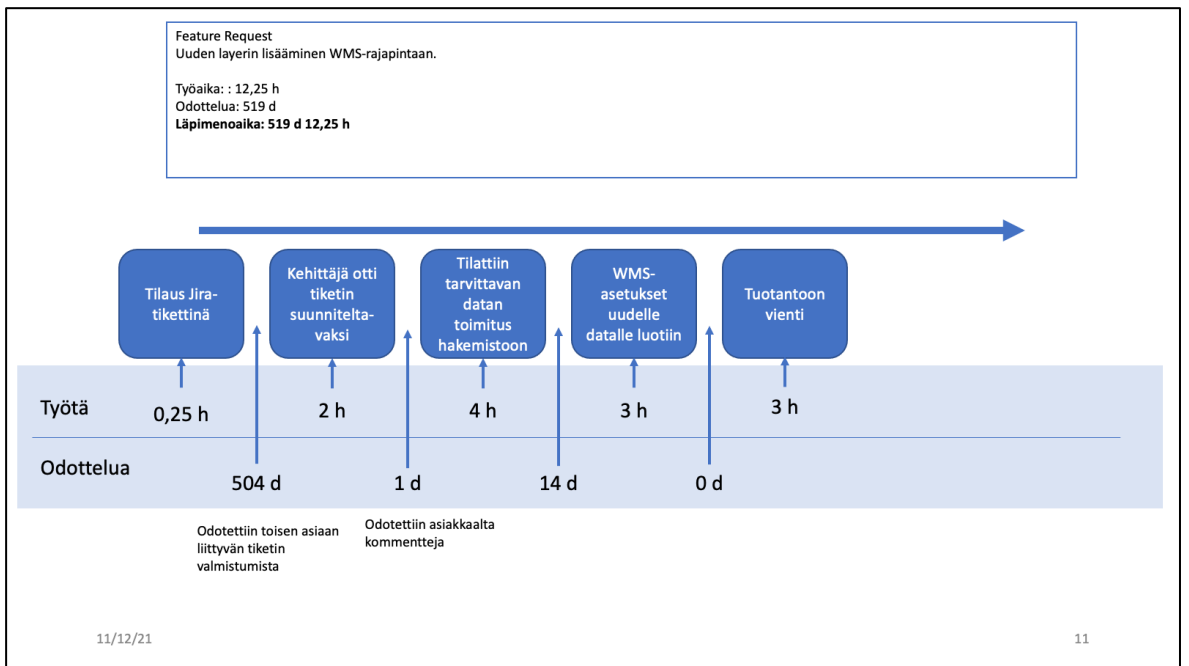
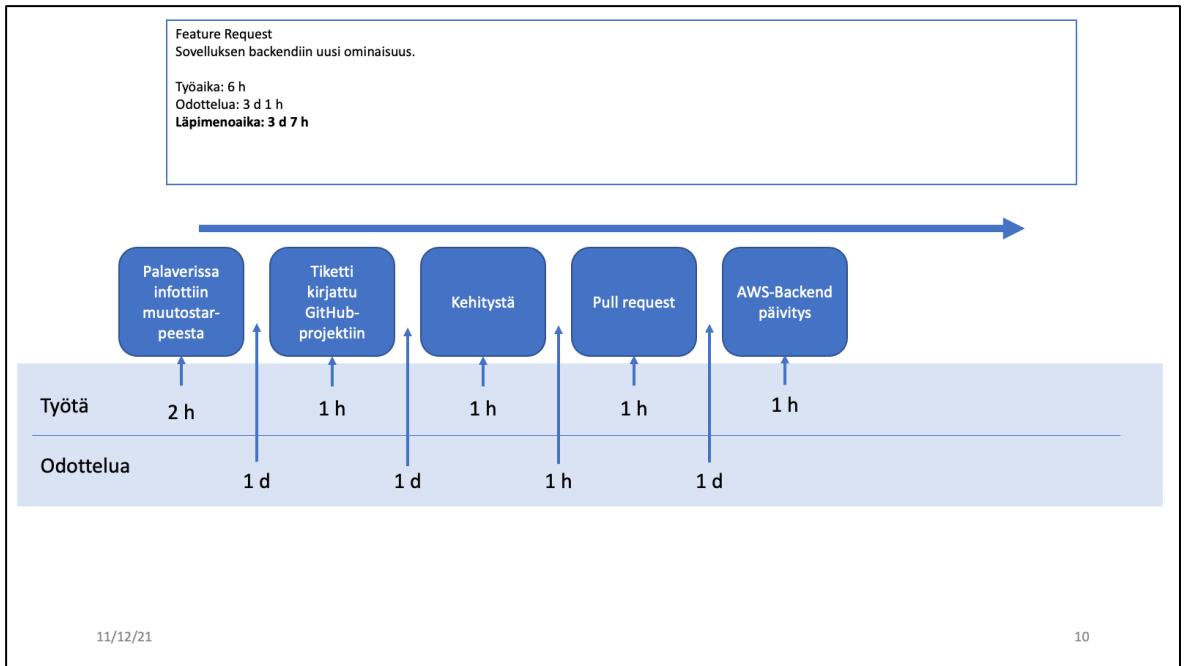
Työaika: 1,75 h
Odottelua: 1,5 h
Läpimenoaika: 3,25 h

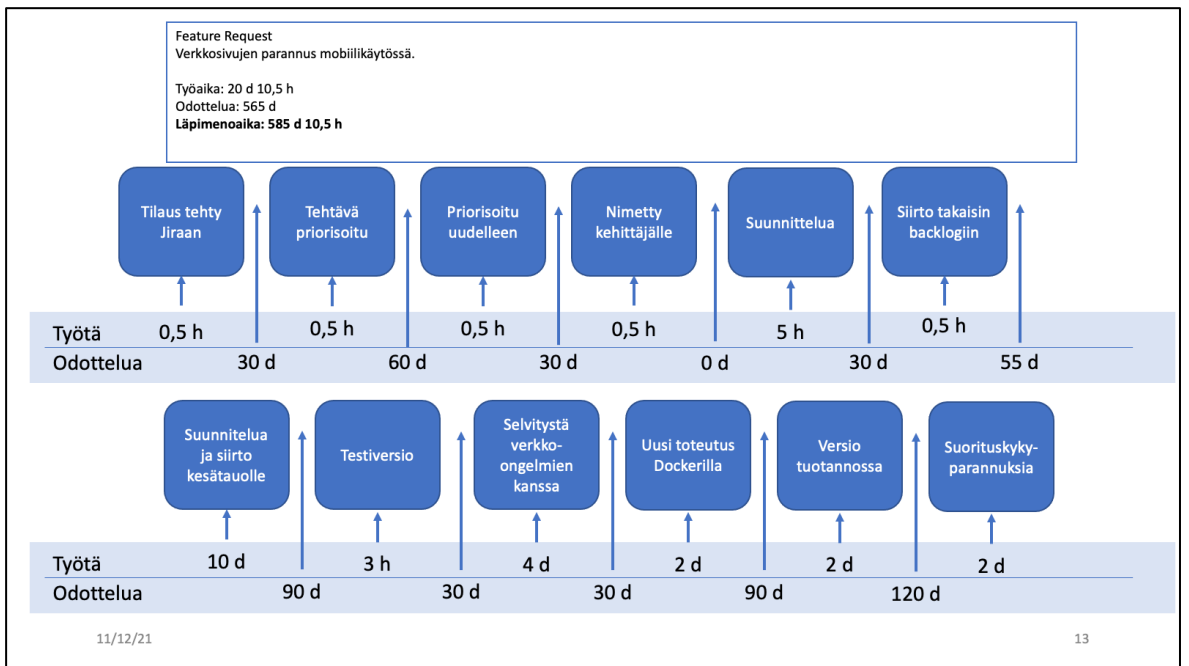
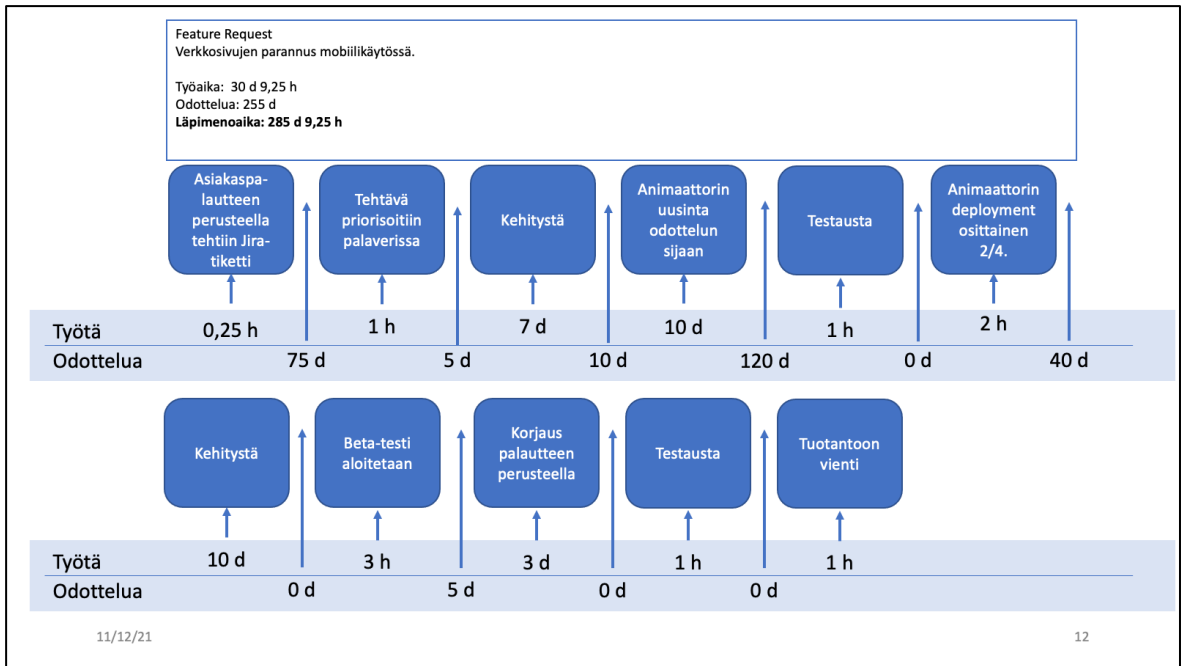


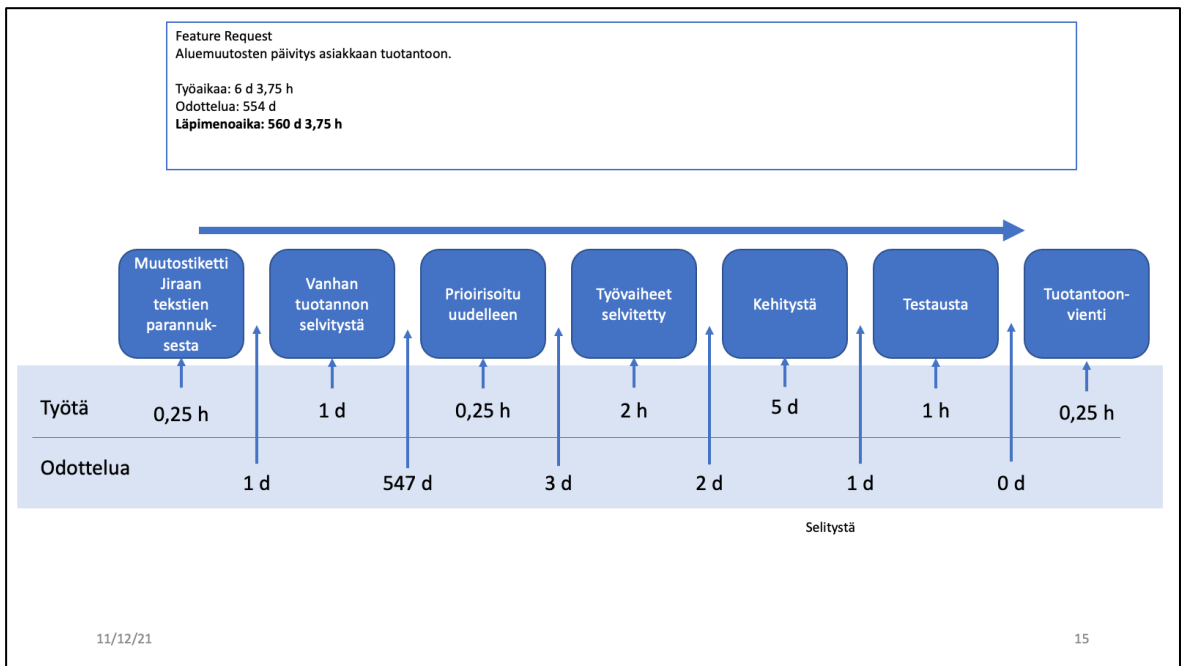
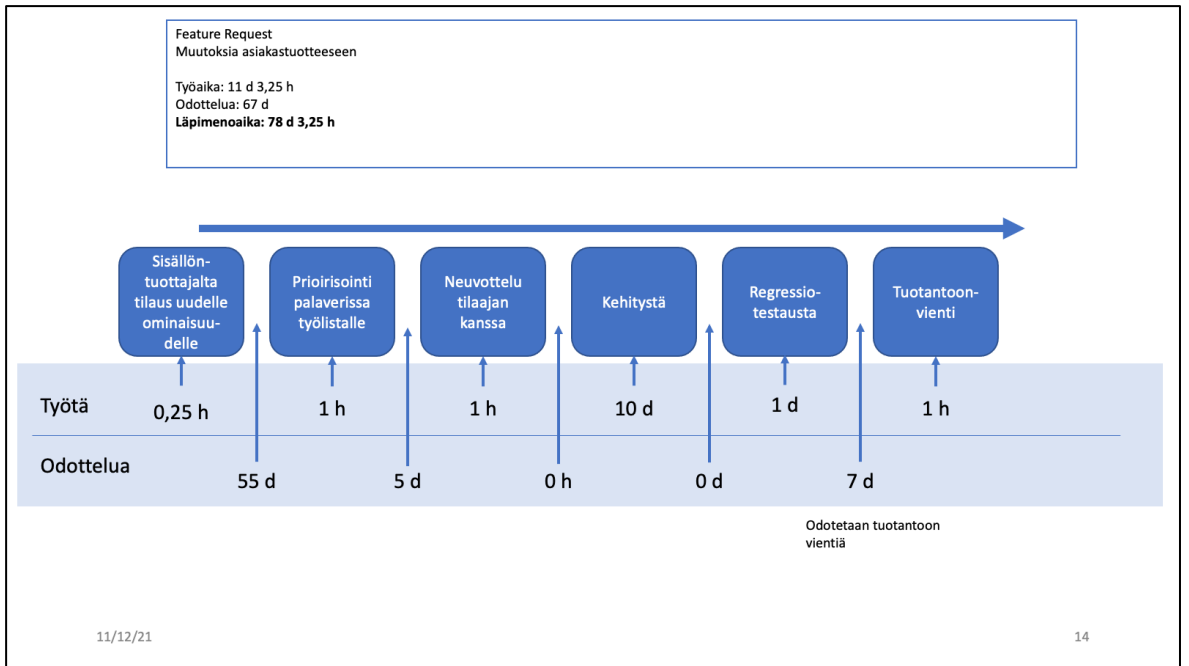
11/12/21

6



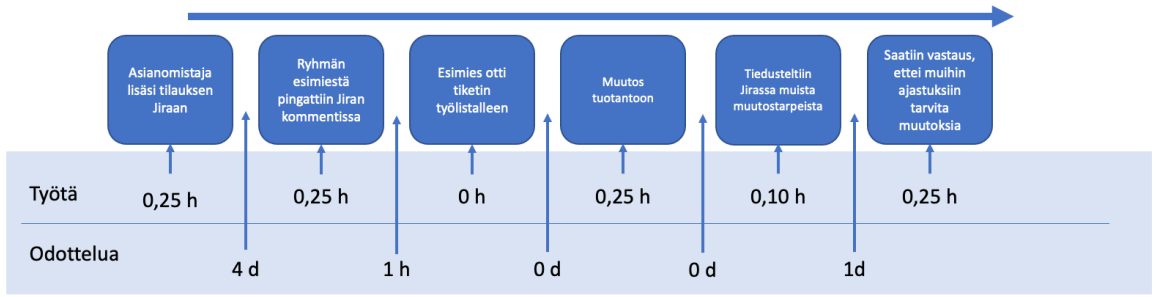






Feature Request
Tuotannon ajastuksen muutos.

Työaika: 1,10 h
Odottelua: 5 d 1h
Läpimenoaika: 5 d 2,10 h

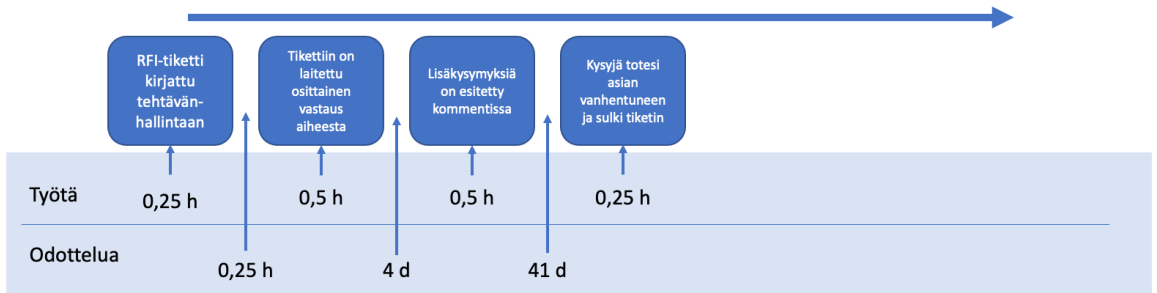


11/12/21

16

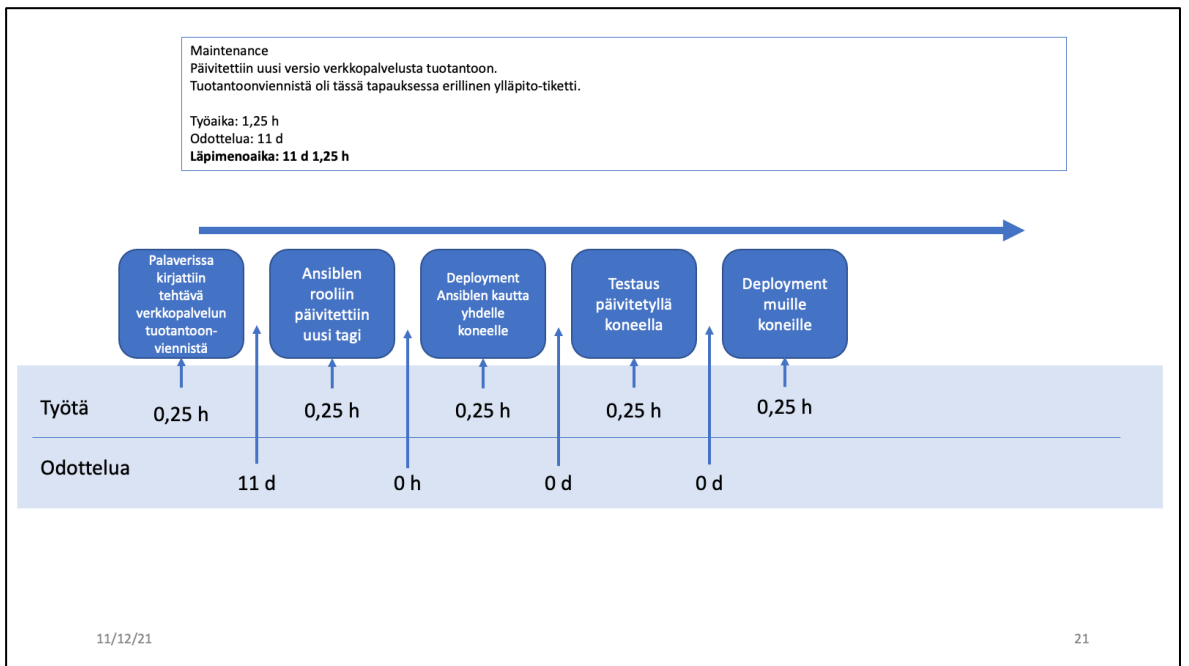
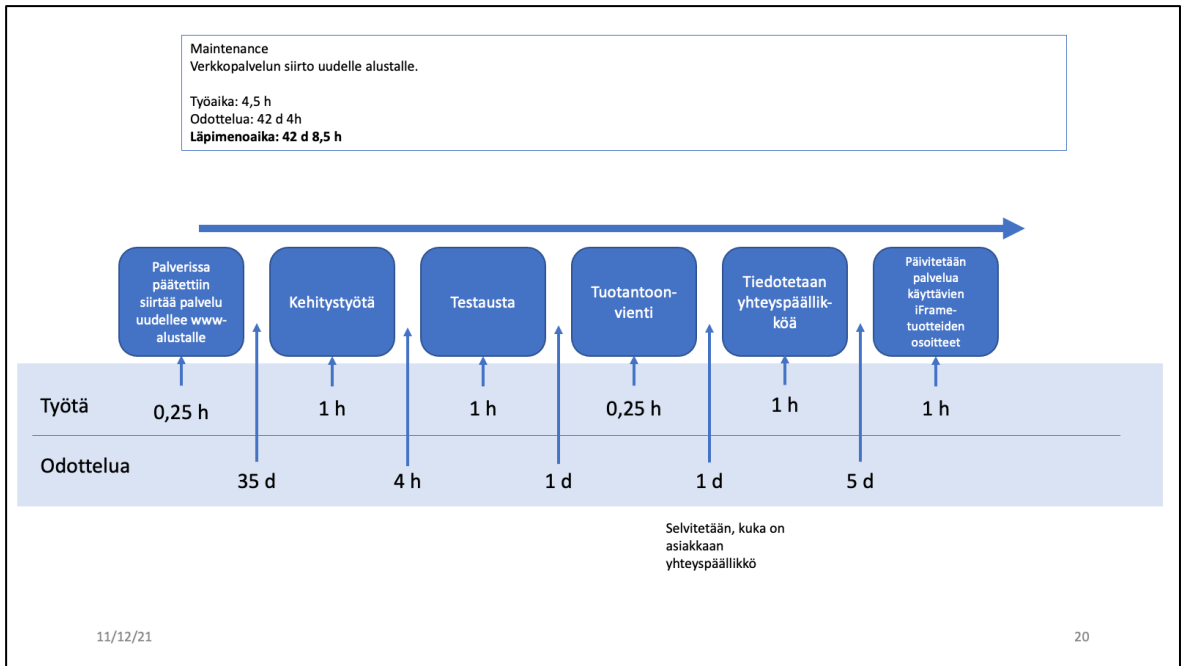
Request for Information

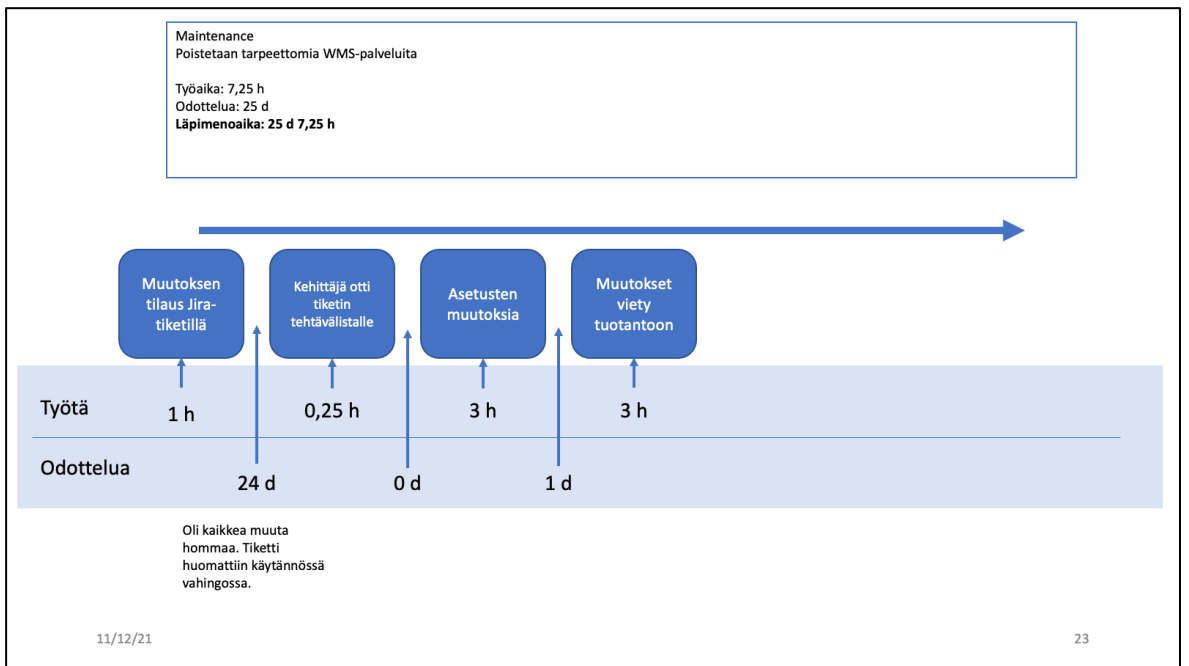
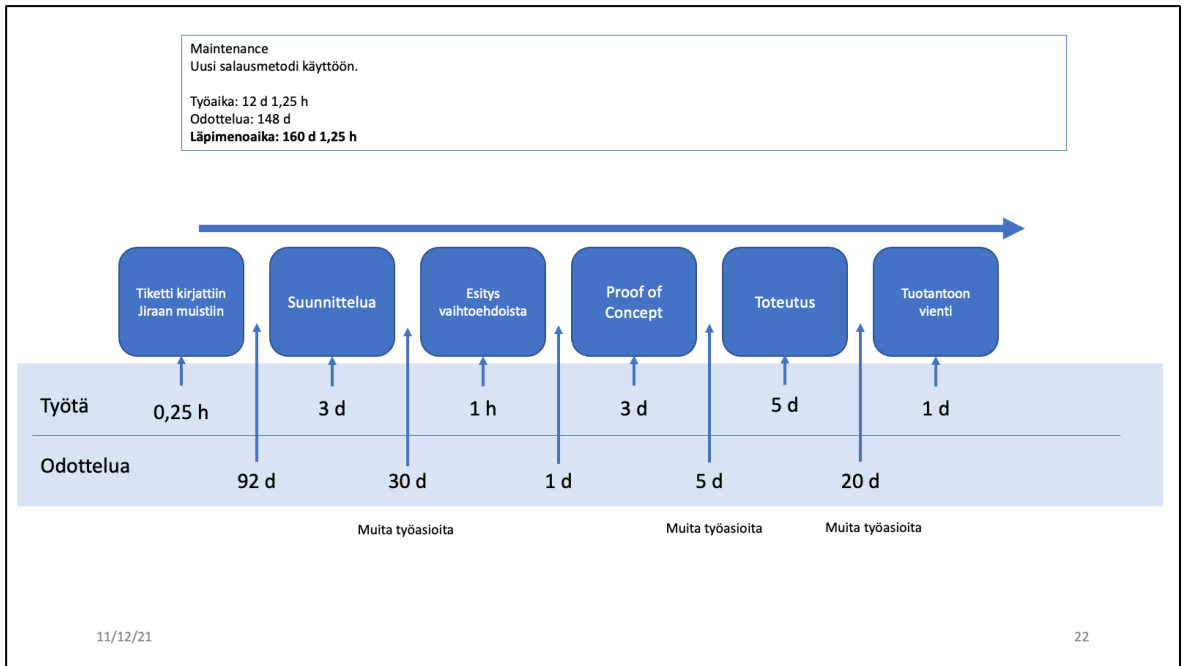
Työtä: 1,5 h
Odottelua: 45 d 0,25 h
Läpimenoaika: 45 d 1,75 h



11/12/21

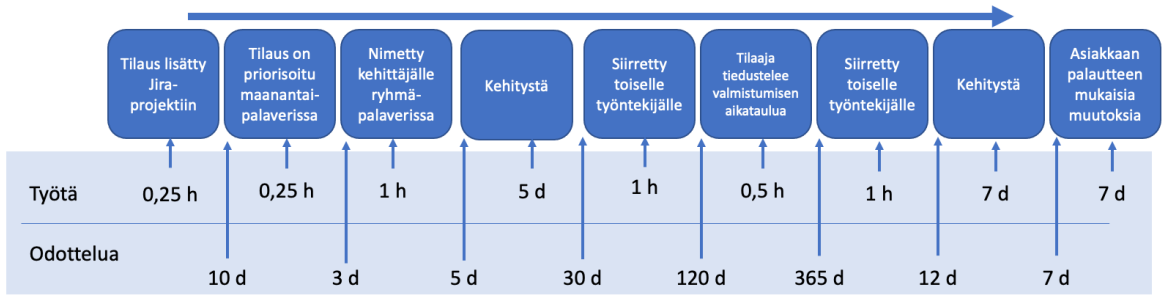
18





Maintenance
Vanhan verkkopalvelun ominaisuuksien porttaaminen uudemmalle alustalle.

Työaika: 19 d 2,75 h
Odottelua: 552 d
Läpimenoaika: 571 d 2,75 h

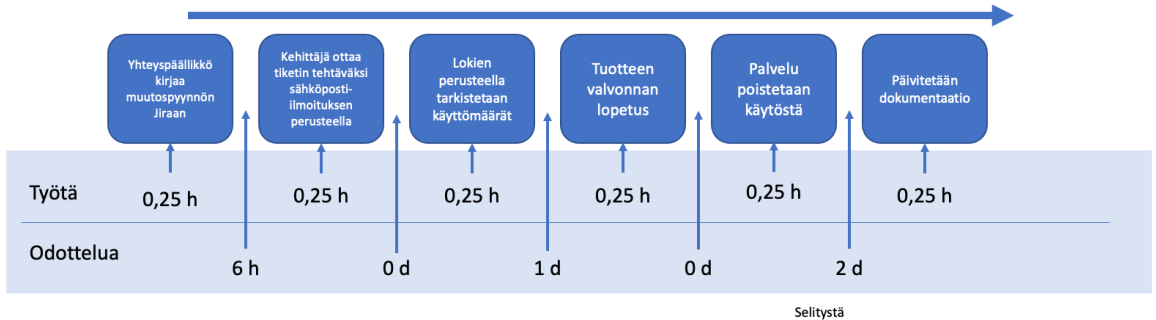


11/12/21

24

Maintenance
Vanhan verkkopalvelun sulkeminen.

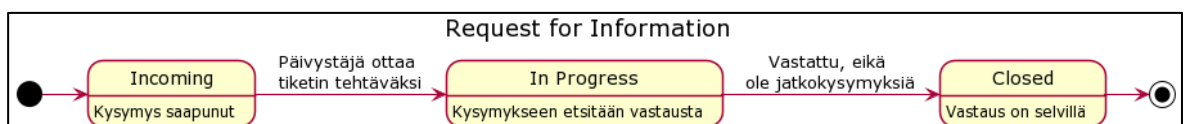
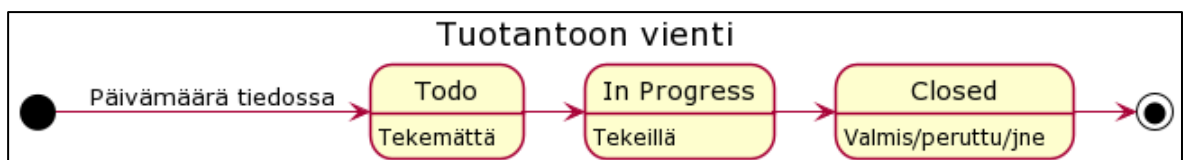
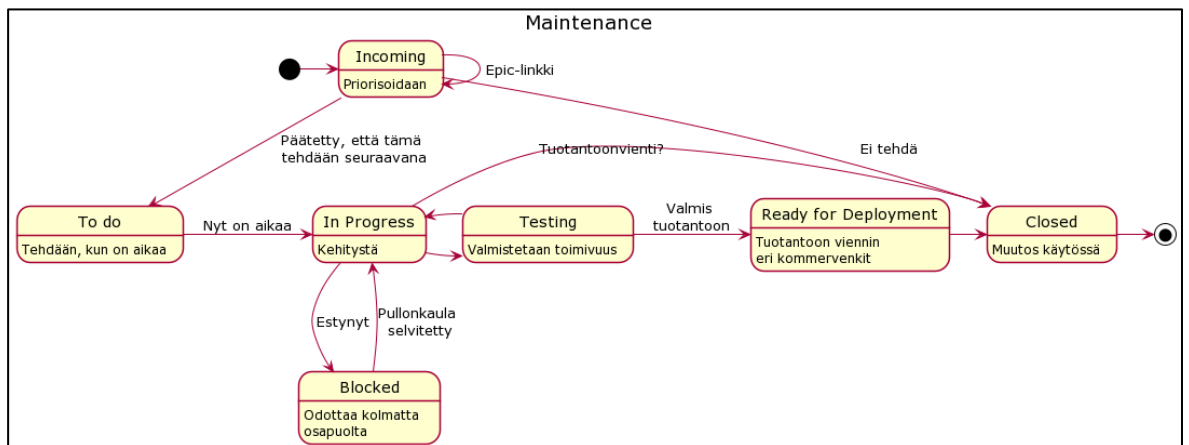
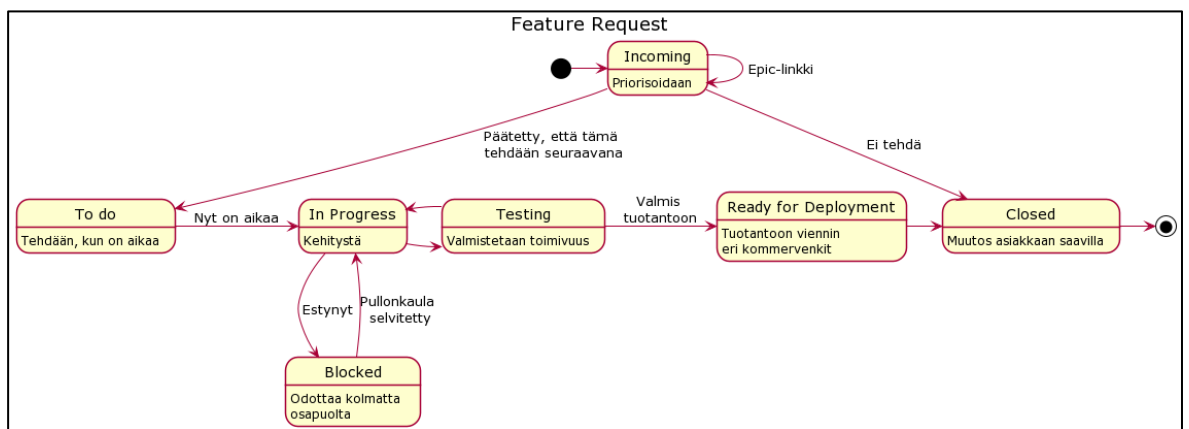
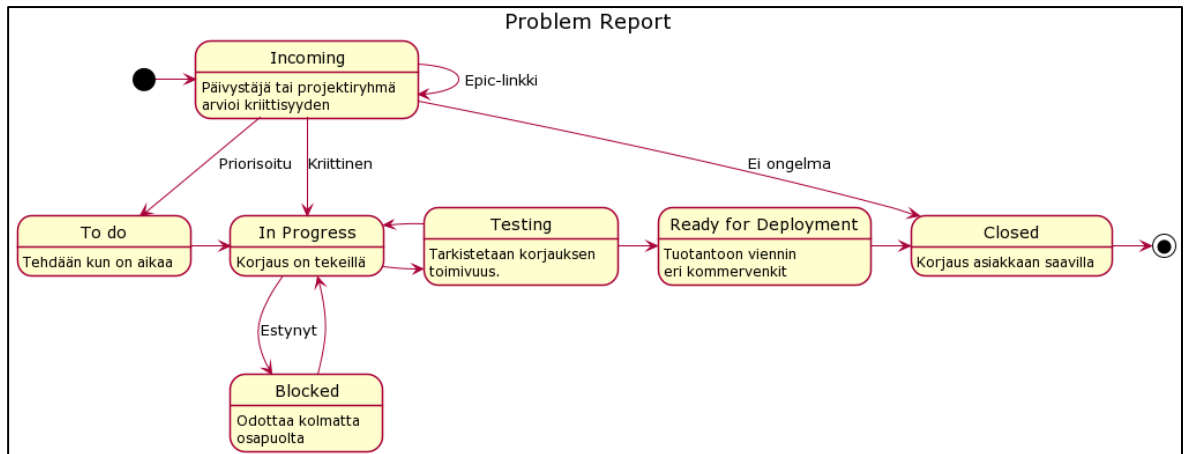
Työaika: 1,5 h
Odottelua: 3 d 6 h
Läpimenoaika: 3 d 7, 5 h

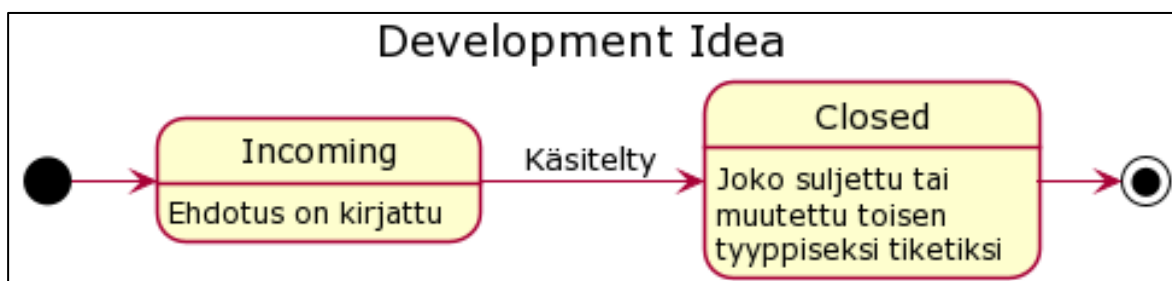
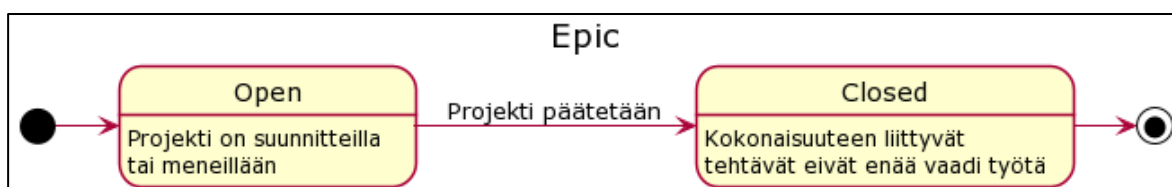
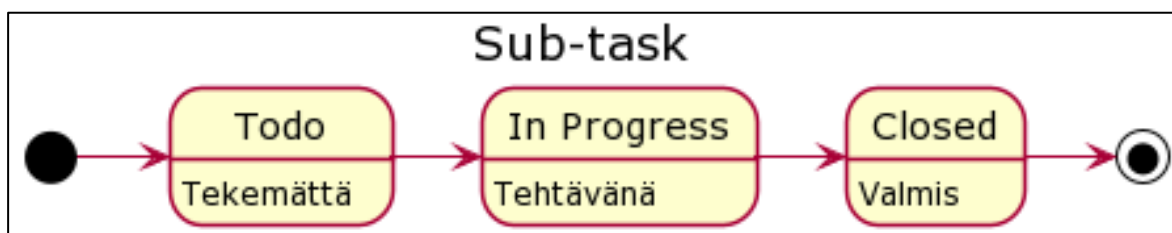


11/12/21

25

Liite 2. Ehdotetut työvuot

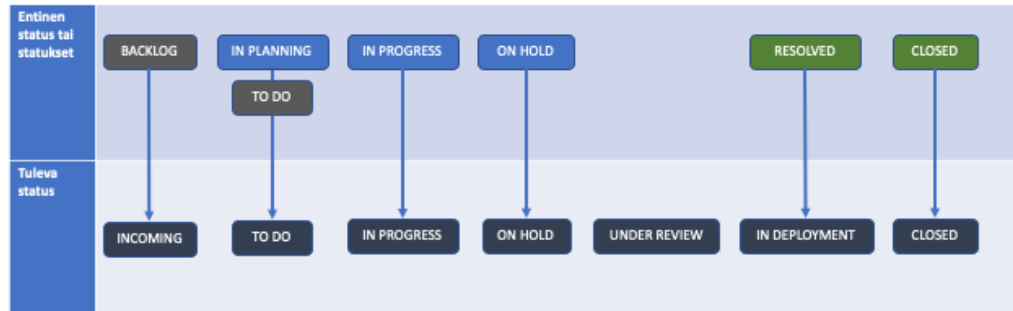




Liite 3. Jira-tikettien tilojen uudelleenjärjestäminen

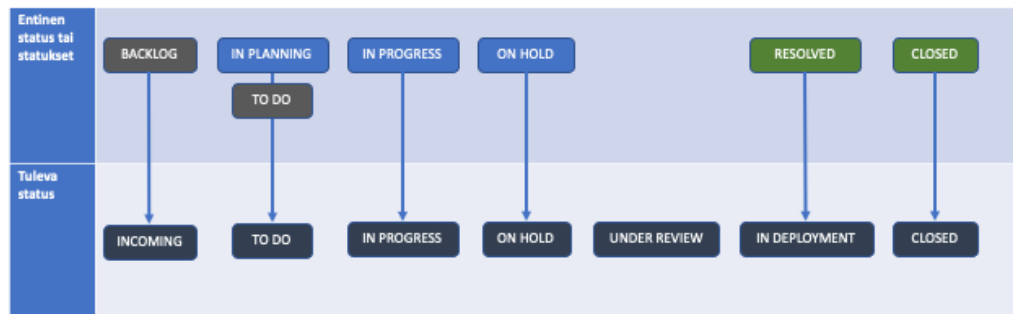
Maintenance

Nykyisten Jira-statusten muutos tilauksille.



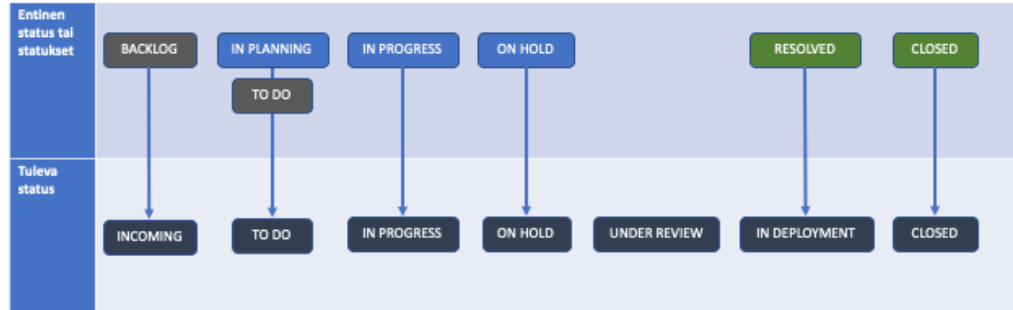
Feature Request

Nykyisten Jira-statusten muutos tilauksille.



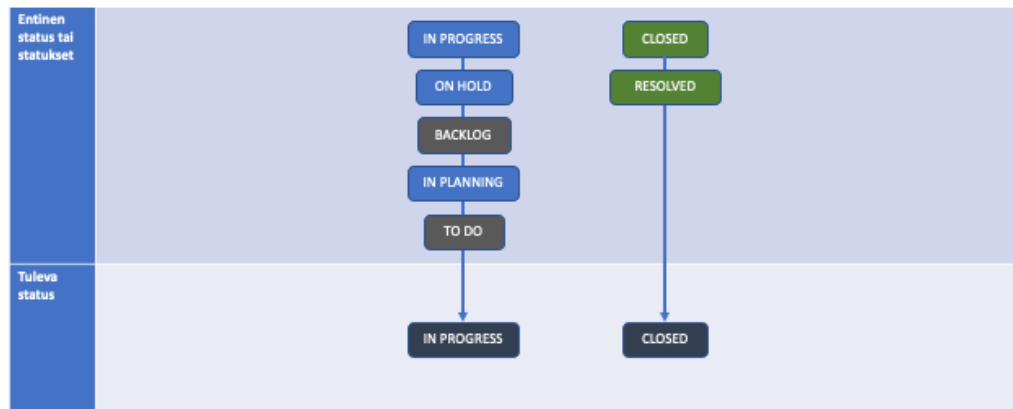
Problem Report

Nykyisten Jira-statususten muutos ongelma-tiketeille.



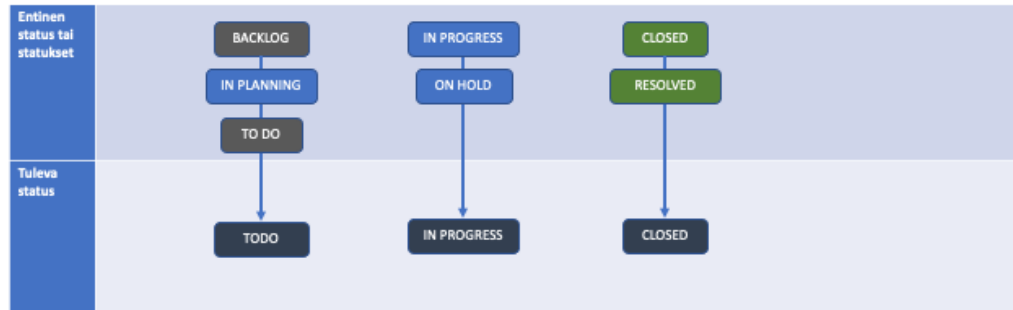
Epic

Nykyisten Jira-statususten muutos Epic-tiketeille.



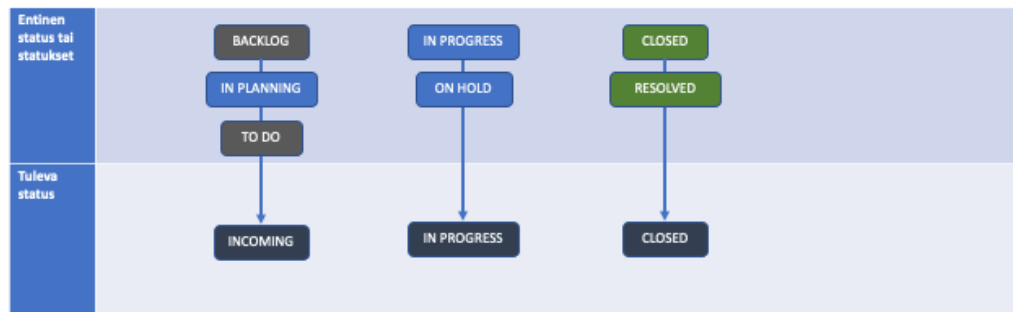
Sub-task

Nykyisten Jira-statusien muutos tikettien tehtäville.



Request for information

Nykyisten Jira-statusien muutos kysymyksille.



Liite 4. Työvuasetukset

