

Pekka Koivusaari

PROJEKTIHALLINNAN TYÖKALUT OHJELMISTOKEHITYKSESSÄ

PROJEKTIHALLINNAN TYÖKALUT OHJELMISTOKEHITYKSESSÄ

Pekka Koivusaari
Opinnäytetyö
Syksy 2018
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, älykkäät järjestelmät

Tekijä: Pekka Koivusaari
Opinnäytetyön nimi: Projektinhallinnan työkalut ohjelmistokehityksessä
Työn ohjaaja: Eero Nousiainen
Työn valmistumislukukausi ja -vuosi: Syksy 2018
Sivumäärä: 33

Työ tehtiin Oulun ammattikorkeakoulu Oy:n tietotekniikan tutkinto-ohjelman tarjoamasta aiheesta tutkia modernien tuotekehitysprojektien työkalujen toimintoja, eroja ja käytettävyyttä. Tavoitteena oli tutustua neljään eri työkaluun ja saada konkreettisesti esille eri filosofioihin pohjautuvat työkalut sekä tehdä analyysiä työkalujen hyvistä ja huonoista puolista. Lisäksi tavoitteena oli tutustua syvällisemmin ketteriin menetelmiin kuuluviin Lean-filosofiaan, Scrum-viitekehukseen sekä Kanban-taulujen käyttöön.

Ohjelmistokehityksessä käytetään nykyisin yleisesti ketteriä menetelmiä projektien läpiviemisessä. Ohjelmistokehityksessä projektinhallintatyökalut perustuvat usein Agile Manifestoon tai Toyotan kehittämään Lean-ajatteluun. Scrum on ketterien menetelmien viitekehys, joka on kehittynyt Agile Manifeston pohjalta, ja Lean-ajattelu on pohjana Kanban-taulujen käyttöön projektinhallinnassa. Tutkittaviksi ohjelmiksi valittiin Axosoft, HacknPlan, Taiga.io sekä Taskworld. Ohjelmia pyrittiin käyttämään kuten ohjelmistokehitystyössä toimittaisiin projektien hallinnassa.

Lopputuloksena saatiin vertailu ohjelmien ominaisuuksien eroista sekä verrattiin Kanbanin perussääntöjen toteutumista valituissa ohjelmissa. Vertailussa mukana olleet työkalut olivat helppoja käyttää sekä tarjosivat kaikki riittävästi informaatiota projektinhallintaan. Kaikki tuotteet soveltuivat Kanbanin mukaiseen työskentelyyn, mutta kuitenkin Axosoft toteutti ehkä parhaiten kaikkia Kanbanin sääntöjä.

Asiasanat: Agile, Lean, Scrum, Kanban, ohjelmistokehitys, projektinhallinta

ABSTRACT

Oulu University of Applied Sciences
Degree program of Information Technology, Intelligent Systems

Author: Pekka Koivusaari
Title of thesis: Project Management Tools in Software Development
Supervisor: Eero Nousiainen
Term and year when the thesis was submitted: Autumn 2018
Number of pages: 33

The topic of this thesis was to compare features, differences and usability of product development project management tools. The topic was given by Oulu University of Applied Science. The goal of this thesis was to familiarise four different tools and introduce tools based on different philosophies. Another goal was to get deeper understanding of Scrum and Lean.

Project management in software development is nowadays often based on Agile Manifesto or originally Toyota's developed Lean thinking. Scrum framework has developed in the base of Agile Manifesto and Lean philosophy is behind the use of Kanban-boards. Selected softwares were Axosoft, HacknPlan, Taiga.io and Taskworld. Testing of software was performed with fictious software development projects.

A result of this thesis was comparison of the features of selected tools and consideration how tools follow Kanban rules. As a result of the comparison, all selected tools were easy to use, and they offer enough information for project management purposes. All of the selected softwares were suitable for working with Kanban principles but Axosoft was lightly better than the rest.

Keywords: Agile, Lean, scrum, Kanban, software development, project management

SISÄLLYS

1	JOHDANTO	6
2	KETTERÄN KEHITYKSEN FILOSOFIAT	7
2.1	Agile	7
2.2	Lean	9
2.2.1	Lean arvot ja periaatteet	10
2.2.2	Työn arvo Lean-filosofiassa	10
3	KETTERÄT PROJEKTINHALLINTAMENETELMÄT	12
3.1	Scrum	12
3.1.1	Scrumin periaatteet	12
3.1.2	Scrum-tiimi	14
3.1.3	Scrumin tapahtumat	14
3.1.4	Scrumin tuotokset	17
3.1	Kanban	18
3.2.1	Kanbanin periaatteet	18
3.2.2	Kanban-taulu	19
4	VIRTUAALISET PROJEKTINHALLINTATYÖKALUT	23
4.1	Vertailtavat työkalut	23
4.1.1	Axosoft	23
4.1.2	HacknPlan	24
4.1.3	Taiga.io	25
4.1.4	Taskworld	26
4.2	Ominaisuuksien vertailu	28
5	YHTEENVETO	31
	LÄHTEET	32

1 JOHDANTO

Ohjelmistokehitys on perinteisesti ollut ja on vielä nykyäänkin projektimuotoista. Aikaisemmin projektimallina ja ohjelmistotuotteen kehitystä ohjaavana hallinnointimenetelmänä on ollut vesiputousmalli. Vesiputousmallissa projektin vaiheet seuraavat toisiaan eikä aikaisempiin vaiheisiin enää palata myöhemmin. Vesiputousmalli on kankea eikä joustava projektin aikana tuleviin uusiin vaatimuksiin. (1, s. 15–18.)

Vesiputousmallin ovat nykyisin hyvin pitkälle korvanneet ketteriin menetelmiin perustuvat toteutusmenetelmät. Ketterien menetelmien voidaan katsoa syntyneen vastavetona vesiputousmallin kankeudelle. (1, s. 15–18.)

Tämän opinnäytetyön tavoitteena oli tutkia modernien tuotekehitysprojektien työkalujen toimintoja, eroja ja käytettävyyttä. Alkuperäisenä tavoitteena oli tutkia kolmea työkalua: Axosoft, HacknPlan ja Taskworld. Myöhemmin mukaan otettiin myös Taiga.io. Samalla tavoitteena oli tutustua Agile Manifestoon perustuvaan ohjelmistokehitykseen syvemmin. Tarkempaan käsittelyyn otettiin tuotteiden toimitukseen ja ylläpitoon tarkoitettu viitekehys Scrum sekä Lean-filosofiaan perustuva Kanban.

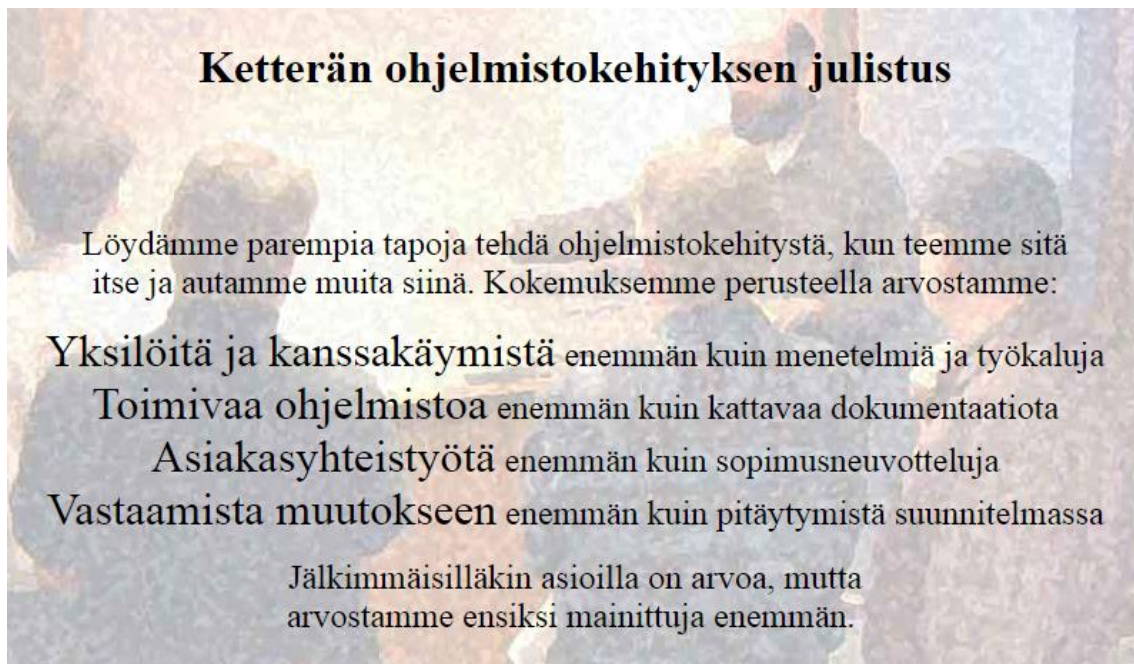
Työssä kerättiin ensin Agilesta, Scrumista, Leanista ja Kanbanista taustatietoja, joiden käyttöä voitiin sitten peilata ja verrata eri työkaluissa. Työkaluihin ja niiden ominaisuuksiin tutustuttiin luomalla joitakin projekteja sekä kuvitteellisesti seuraamalla niiden etenemistä ja valmistumista työkalujen avulla. Valittujen työkalujen ominaisuuksia tarkasteltiin Kanban-filosofian perussääntöihin verraten, ja tällä kriteerillä valittiin parhaiten Kanban-filosofian mukaiseen käyttöön soveltuva ohjelma.

2 KETTERÄN KEHITYKSEN FILOSOFIAT

2.1 Agile

Ketterän kehityksen alkuna voidaan pitää vuonna 2001 Utahissa pidettyä 17 ohjelmistokehittäjän tapaamista, johon he olivat kokoontuneet keskustellakseen käyttämistään ohjelmistokehitysmenetelmistä. Kokouksen tuloksena menetelmiä päätettiin kutsua ketteriksi menetelmiksi, niiden käytön edistämiseksi perustettiin Agile Alliance -järjestö sekä julkaistiin ketterän ohjelmistokehityksen julistus. Ketterän ohjelmistokehityksen julistuksessa (Agile Manifesto) määritellään menetelmien pohjana olevat arvot sekä keskeiset periaatteet. (1, s. 15–18.)

Kuvassa 1 on esitetty Agile Manifeston arvot.



KUVA 1. Ketterän ohjelmistokehityksen julistus – arvot (2)

Yksilöitä ja kanssakäymistä painottavassa arvossa ketterien menetelmien yhteydessä on tarkoitus antaa kehittäjille täysi vastuu omasta työstään ja sen suunnittelusta. Jatkuva kanssakäyminen ja

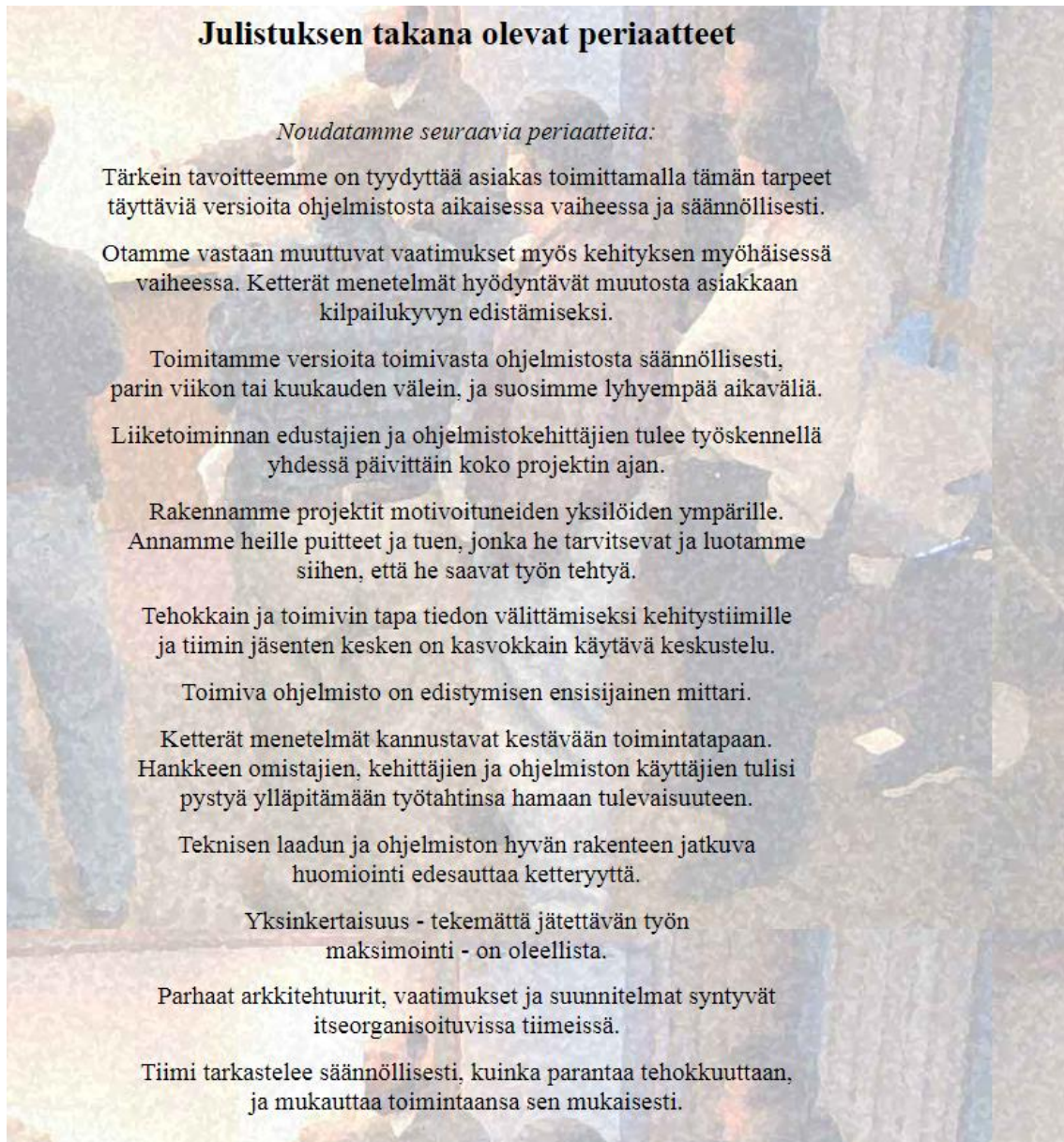
toimiva kommunikaatio tulee olla tavoitteena niin tiimin sisällä kuin sen ulkopuolisissa yhteyksissä. Tiimi päättää itsenäisesti työtehtävistään sekä työmenetelmistään ja niiden kehittämisestä. Itsenäisellä tiimillä on myös mahdollisuus vaikuttaa toteutuksen tekniseen sisältöön sekä ennen kaikkea työn aikataulutukseen, mitä on mahdollista saada tehdyksi aina tietyssä ajanjaksossa. Käytännössä työn tulisi pohjautua yksilöihin, jotka kommunikoivat tehokkaasti toistensa kanssa. Käytössä olevat työkalut ja menetelmät tukevat tiimin toimintaa. (1. s. 18.)

Toimivaa ohjelmistoa painottavassa arvossa asetetaan kehityksen mittariksi säännöllisin väliajoin toimitettu toimiva ohjelmisto. Ohjelmistokokonaisuus on paloiteltu pieniin osiin ja ohjelmistoon lisätään toimivia osia vähitellen kasvattaen kokonaisuutta. Jokainen lisättävä osa on tehty valmiiksi ja testaamalla todettu toimivaksi. Kattava dokumentaatio ei ole mikään tavoite tai itseisarvo vaan dokumentaation on tarkoitus tukea ohjelmointityön tekemistä. Joskus toimivan ohjelmiston asettaminen dokumentaation edelle on ymmärretty suunnittelun jättämiseksi vähäisemmälle huomiolle, mutta tämä on kuitenkin virheellinen tulkinta. ”Haluaisitko sinä esimerkiksi ajaa autolla, jota ei ole varsinaisesti suunniteltu, vaan se on vain koottu? Tarkoituksenmukaisen ohjelmiston luominen vaatii aina suunnittelutyötä.” (1, s. 18.)

Asiakasyhteistyötä painottavassa arvossa asiakas osallistuu aktiivisesti suunnittelutyöhön koko ohjelmistokehityksen ajan. Perinteisesti asiakkaan kanssa on aluksi neuvoteltu mahdollisimman yksityiskohtainen sopimus tuotteesta ja asiakas on nähnyt valmiin tuotteen vasta projektin lopussa. Asiakasyhteistyö arvon mukaan tuotantotiimillä on valmiita työn tuloksia esitettäväksi lyhyin väliajoin ja asiakas antaa palautetta sekä pysyy työn edistymisen tasolla koko ajan. Tuotantotiimillä ja asiakkaalla säilyy tavoite samana eikä projektin lopussa havaita, että jossain olisi tapahtunut tiedonkatkoksia. On tärkeää myös asiakkaan ymmärtää, että projekti vaatii panostuksia molemmilta osapuolilta. (3.)

Muutokseen vastaamista painottavassa arvossa korostetaan joustavuutta ja mukautumista tuotteen vaatimukseen kehitystyön aikana tuleviin muutoksiin. Ohjelmistoprojektin edetessä kaikkien osapuolten ymmärrys tuotteen ideaalisesta toiminnasta kasvaa. Myös muuttuvat ympäristötekijät saattavat aiheuttaa uusia vaatimuksia tuotteelle. Alkuperäiset suunnitelmat mukautuvat näihin muutoksiin ja niihin ollaan valmiita tekemään muutoksia tarvittaessa. Mitään ei myöskään lyödä lukkoon lopullisesti, vaan suunnitelmat tehdään aina vaan niin pitkälle, kuin työn eteneminen sillä hetkellä vaatii. (3.)

Kuvassa 2 on esitetty Agile Manifeston periaatteet.



KUVA 2. Ketterän ohjelmistokehityksen julistus – periaatteet (2)

2.2 Lean

Lean on asiakasorientoitunut tuotantofilosofia, joka pohjautuu japanilaisiin tuotantofilosofioihin ja johtamisjärjestelmiin, erityisesti Toyotan autotehtailleen kehittämän Toyota Production System -tuotantojärjestelmään. Lean-filosofian oleellisia käsitteitä ovat asiakkaan arvo, josta voidaan käyttää myös nimitystä lisäarvo sekä hukka tai turha toiminta. (4, s. 14–16.)

2.2.1 Lean arvot ja periaatteet

Lean tuli yleisesti tunnetuksi 1990-luvulla kirjassa ”The Machine That Changed the World”. Kirjassa esitellään Leanin viisi peruseriaatetta:

- arvon määrittämisen perustuminen asiakkaan näkemykseen
- arvoketjun tunnistaminen ja kaiken arvoa tuottamattoman toiminnan poistaminen
- arvoketjun perustaminen asiakkaan tarpeisiin perustuvaan imuohjaukseen
- työntekijöiden osallistuminen kehittämiseen
- toiminnan jatkuva kehittäminen. (5.)

Leanin tavoitteena yksikertaisesti on maksimoida asiakkaan saama lisäarvo minimoimalla samalla kaikki turha toiminta. Asiakkaalle pyritään tuottamaan laatua ja nopeutta sekä toimimaan asiakasorientoituneesti. Kaikki toiminta, mikä ei lisää arvoa, jätetään tekemättä ja työskennellään vain asioiden parissa, jotka on ehdottomasti saatava tehtyä kyseisellä hetkellä. (6.)

Vastoin yleistä harhakäsitystä Lean ei ole parhaimmillaan vain tuotantoon keskittyvissä yrityksissä, vaan se on toimintamalliksi soveltuva filosofia kaikkeen liiketoimintaan ja kaikenlaisiin prosesseihin. Lean ei ole mikään taktinen tai kustannuksia vähentävä ohjelma tuotantolaitoksille, vaan se on tapa ajatteluun ja toimintaan koko organisaation sisällä. Monet yritykset ja yhteisöt teollisuuden ja palveluiden aloilla käyttävät Leanin periaatteita tapana ajatella ja toimia. Kaikki eivät käytä toimintamallistaan nimitystä Lean, vaan yritykset ovat luoneet omia järjestelmiään, kuten Leanin pohjana ollut Toyota Production System tai terveydenhoitoalan Danaher Business System. Yritykset haluavat korostaa tällä myös, että Lean ei ole niille mikään tiuhassa tahdissa vaihtuva lyhyen tähtäimen kustannussäästö ohjelma vaan koko yrityksen pitkän ajan toimintatapa. Yrityksen siirtyminen Lean-filosofian mukaiseen toimintatapaan vaatii täydellisen muutoksen liiketoiminnoissa. Muutos edellyttää pitkäjänteisyyttä ja periksiantamattomuutta. (7.)

2.2.2 Työn arvo Lean-filosofiassa

Lean filosofian ymmärtämiseksi täytyy sisäistää käsitteet arvo (lisäarvo) sekä hukka tai turha toiminta. Lisäarvoa on kaikki se, millä asiakas kokee olevan arvoa, mitä asiakas pitää tärkeänä tai mistä asiakas on valmis maksamaan. Yksinkertaisimmillaan asiakas kertoo mitä hän tarvitsee, ja toimittaja tekee tuotteen tai palvelun sekä toimittaa sen perille. Hukan tai turhan toiminnan

tunnistaminen on oleellista Leanissa. Hukaksi voidaan ymmärtää kaikki paitsi pienin mahdollinen määrä uhrattuja resursseja, jotka ovat ehdottomasti tarpeellisia lisäarvon tuottamiseksi asiakkaalle. (4, s. 18–24.)

Työhön liittyvät aktiviteetit voidaan jakaa kolmeen kategoriaan.

- Arvoa lisäävät aktiviteetit, jotka joko muuttavat tuotetta valmistusprosessissa tai vähentävät muuten epävarmuutta prosessissa. Asiakkaan tulle olla halukas ja valmis hyväksymään toiminta sekä maksamaan siitä. Lisäksi tämä aktiviteetti täytyy tehdä oikein ensimmäisellä kerralla.
- Välttämättömät ei lisäarvoa tuovat aktiviteetit, jotka eivät kuulu ylläoleviin mutta joita ei voida jättää pois koska ne perustuvat lakiin, sopimukseen, käytettävään teknologiaan tai ovat muuten vastaavasti perusteltuja.
- Ei lisäarvoa tuovat aktiviteetit, jotka kuluttavat resursseja eivätkä tuota lisäarvoa. Nämä puhdasta hukkaa olevat toiminnot ovat esimerkiksi turha raportointi, seisonta-aika ja virheet, jotka vaativat korjausta. (4, s. 18–24.)

3 KETTERÄT PROJEKTIHALLINTAMENETELMÄT

3.1 Scrum

3.1.1 Scrumin periaatteet

Scrum on nykyään yleisin ketterään kehitykseen perustuvaan ohjelmistokehityksen hallintamenetelmä. Viime vuosina Scrum on kehittynyt projektinhallintamethodiikasta uudenaikaiseksi ymmärrykseksi, kuinka hallita vajaatoimisia tiimejä, osastoja, suurempia organisaatioita tai jopa kokonaisia yrityksiä. Tyypillisesti yritykset käyttävät Scrumia aluksi yhden tiimin tai projektin sisällä projektin hallinta menetelmänä. Jotkut myös pysyvät tällä tasolla, kun taas monet kehittävät toimintaansa laajemmalle organisaatiossaan Scrumin periaatteiden mukaan. (8, s. 11–13.)

Scrum koostuu itseasiassa vain muutamista periaatteista ja vähälukuisista säännöistä. On huomattava, että näitä periaatteita ja sääntöjä on noudatettava tiukasti. Agile Manifeston periaatteiden mukaisesti Scrum kunnioittaa yksilöä ja tiimeille annetaan vapaus toteuttaa tehtävänsä siten että jokaisen tiimin jäsenen lahjat ja kyvyt tulisivat parhaalla mahdollisella tavalla käyttöön. (8, s. 11–13.)

Käsitykset että Scrum olisi ohjelmistojen kehittämismenetelmä tai että se antaisi tiimin jäsenille rajoittamattoman vapauden toimia ovat yleisiä väärinymmärryksiä. Agile manifeston periaatteet vaikuttavat siten että Scrum on mielekäs yhdistelmä sääntöjä, kuria, henkilökohtaista vastuullisuutta, yhteistä ajatustyötä, toisten auttamista ja oman osaamisen käyttöä tiimin hyväksi. (8, s. 11–13.)

Scrumin arvot ovat

- sitoutuminen
- rohkeus
- keskittyminen
- avoimuus
- kunnioitus. (8, s. 11–13.)

Arvojen mukaan toimiessaan Scrum-tiimi vahvistaa Scrumin kolmea tukijalkaa: läpinäkyvyyttä, tarkastelua ja sopeuttamista. (8, s. 11 – 13).

Seuraavassa on esitetty joitain tavanomaiseen ohjelmistokehitykseen liittyviä asioita ja vertailtu niiden eroja toimittaessa täsmällisesti Scrumin mukaan.

1. Tuotekehitys vai projektin hallinta

Scrum ei johda ennalta määriteltyyn lopputuotteeseen vaan on pikemminkin jatkuva virta lopputuotteen osia, joka päättyy lopulta kaikki vaatimukset kattavaan lopputuotteeseen. Scrumin mukaan toimivassa projektissa tuotekehittäjät lupaavat tuotteen, jossa arvioidaan tulevaisuuden tarpeita jatkuvilla lisäyksillä nykyisiä muutoksia. Scrumin oletuksen mukaan jokaisen sprintin jälkeen tuotteena olisi toiminnallinen, jatkuvasti kasvava osa kokonaisuohjelmistoa. Projektin etenemisen mittaaminen on mahdollista toimitettujen osien perusteella. Tämän takia ketterät menetelmät poikkeavat entisistä toimintavoista. Lyhyissä sykleissä toimitetaan koko ajan valmiita osia tuotteesta, mikä ei suinkaan ole toimintamalli perinteisissä projektinhallinta menetelmissä. (8, s. 11–13.)

2. Viitekehys hallinnalle vai kehittämismenetelmä

Scrum ei anna mitään sääntöjä työskentelyyn, mutta se määrittelee tarkasti roolit ja niihin kuuluvat vastuut. Scrumissa määritetään myös tarkat rajat tuotekehityksen kululle. Periaate toistuvista aikaikkunoista tuottaa paitsi tuloksia luovaa painetta niin myös turvallisuutta mikä on tarpeen itseohjautuvalle tiimille. (8, s. 11–13.)

3. Tuoteomistaja vai projektipäällikkö

Termiä projektipäällikkö ei esiinny Scrumin mukaan työskentelevässä organisaatiossa. Tuoteomistaja (Product Owner) on henkilö, joka antaa idean tuotteesta tiimille ja rohkaisee tiimin jäseniä tuottelaisuuteen. Tuoteomistaja on vastuussa projektin taloudesta, mutta päinvastoin kuin normaalisti projektimanagerilla hänellä on vahva ymmärrys asiakkaan toimialasta ja vaatimuksista. (8, s. 11–13.)

3.1.2 Scrum-tiimi

Scrum-tiimiin kuuluvat tuoteomistaja, kehitystiimi ja scrummaster. Scrum-tiimi on itseohjautuva ja monitaitoinen. Itseohjautuva tiimi pystyy itse päättämään, miten työ saadaan tehdyksi ilman ulkopuolista ohjausta. Monitaitoisuus taas varmistaa, että tiimillä on riittävä osaaminen ilman ulkoista apua. (9, s. 6.)

Tuoteomistajalla on vastuu kehitystiimin työn tuloksena saatavan tuotteen arvon maksimoinnista. Tuoteomistaja hallitsee tuotteen kehitysjonon ja vastaa siitä, että kehitysjohto on riittävän avoin, läpinäkyvä ja ymmärrettävä. Scrum-tiimin on oltava koko ajan selvillä mitä seuraavaksi tehdään ja tämä on tuoteomistajan vastuulla. (9, s. 7.)

Kehitystiimi on itsenäinen ryhmä ammattilaisia, jotka pystyvät muuttamaan jokaiselle sprintille asetetun tavoitteen valmiiksi, käyttökelpoiseksi ja potentiaalisesti julkaisukelpoiseksi inkrementiksi. Inkrementti on summa kaikista tuotteen kehitysjonon kohdista, jotka ovat valmistuneet sprintin tai aiempien sprinttien aikana. Julkaisuja saattaa olla useampiakin, mutta vähintään yksi julkaisukelpoinen on kuitenkin jokaisessa sprintissä tuotettava. Kehitystiimi on täysin itsenäinen eikä edes scrummaster puutu tiimin työntekoon. Kehitystiimin on myös oltava riittävän monitaitoinen saamaan inkrementin valmiiksi ilman ulkopuolista apua. Kehitystiimin sisällä kenelläkään ei ole mitään titteleitä eikä tiimissä ole alitiimejä esimerkiksi testaukseen tai arkkitehtuuriin. Tiimin optimaalinen koko on vähintään kolmesta henkilöstä enintään yhdeksään henkilöön, jotka kantavat vastuun kehitystyöstä yhdessä. (9, s. 7.)

Scrummaster toimii tiimissä palvelevana ja avustavana johtajana. Scrummasterin tehtävänä on toimia tuoteomistajan, kehitystiimin ja muun organisaation kanssa yhteistyössä mahdollistaen samalla Scrum-tiimin mahdollisimman häiriöttömän työskentelyn. Scrummasterin vastuulla ovat yleensä lähes kaikki työn etenemiseen ja Scrumin tapahtumiin liittyvät asiat. (9, s. 8.)

3.1.3 Scrumin tapahtumat

Scrumissa tapahtumat ja niiden aikataulutus on ennalta sovittu, jotta voidaan luoda säännöllisyyttä ja voidaan minimoida kaikkien muiden kuin Scrum-palaverien tarve. Scrumin tapahtumille on asetettu aikaraja ja niillä jokaisella on maksimipituutensa. Kaikissa Scrumin tapahtumissa on

mahdollista tarkastella ja sopeuttaa tuotekehitykseen liittyviä asioita. Tapahtumia ei pidä jättää pois koska silloin menetetään läpinäkyvyyttä sekä tilaisuus kehitystyön tarkasteluun ja sopeuttamiseen. (9, s. 9.)

Sprintti on Scrumin ydin, enintään kuukauden mittainen tai lyhyempi ajanjakso, jonka aikana tuotetaan valmis määrittelyn täyttävä, käyttökelpoinen sekä mahdollisesti julkaisukelpoinen inkrementti. Sprintit ovat aina samanmittaisia koko tuotekehityksen ajan ja seuraava alkaa heti edellisen päätyttyä. Sprinttiin kuuluu neljä muodollista tapahtumaa joissa prosessin ja tuotosten tarkasteluun sekä sopeuttamiseen:

- sprintin suunnittelupalaveri
- päivittäispalaverit
- sprintin katselmointi
- sprintin retrospektiivi. (9, s. 9.)

Suunnittelupalaverissa määritellään sprintin tavoite yhteistyössä koko Scrum-tiimin kanssa. Sprintin suunnittelu rajataan enintään kahdeksan tunnin mittaiseksi tapahtumaksi ja siinä vastataan seuraaviin kysymyksiin:

- Mitä on mahdollista toimittaa alkavan sprintin inkrementissä?
- Miten inkrementin toimittamiseen liittyvä työ voitaisiin toteuttaa? (9, s. 11.)

Suunnittelupalaverissa kehitystiimi antaa arvion mitä toiminnallisuuksia ehditään toteuttaa sprintin aikana. Yhdessä tuoteomistajan kanssa keskustellaan sprintin tavoitteesta sekä valittavista tuotteen kehitysjonon kohdista. Seuraavaa sprinttiä peilataan edelliseen inkrementtiin, kehitystiimin kapasiteettiin ja aiempaan suorituskyykyyn. Kehitystiimi tekee päätöksen sopivasta työmäärästä koska vain he voivat arvioida mitä seuraavassa sprintissä pystytään toteuttamaan. (9, s. 10–12.)

Suunnittelupalaverissa kehitystiimi suunnittelee myös, miten määritelty työ toteutetaan valmiiksi inkrementiksi. Sprintin kehitysjono sisältää sprinttiin valitut kehitysjonon kohdat sekä suunnitelman niiden toteuttamiseksi. Kehitystiimi toimii itseohjautuvasti sprintin suunnittelupalaverissa sekä sprintin aikana mutta tuoteomistaja voi myös osallistua toteutuksen suunnitteluun tarkentaakseen kehitysjonon sisältöä ja helpottaakseen kompromissien tekemistä. Kehitystiimi saa kutsua suunnittelupalaveriin myös henkilöitä, jotka voivat antaa teknistä tai liiketoiminnallista neuvontaa. Suunnittelupalaverin lopputuloksena kehitystiimi osaa selittää tuoteohjaajalle ja scrummasterille

miten kehitystiimi työskentelee tavoitteen saavuttamiseksi sekä luoda ennustamansa inkrementin. (9, s. 10–12.)

Päivittäispalaveri on korkeintaan 15 minuutin pituiseksi aikarajattu säännöllinen tapahtuma. On suositeltavaa pitää päivittäispalaveri joka päivä samaan aikaan ja samassa paikassa. Palaverissa suunnitellaan seuraavan 24 tunnin aikaiset työtehtävät sekä tarkastellaan edellisen vuorokauden aikana tehtyä työtä. Samalla ennustetaan sprintissä jäljellä olevaa työmäärää. Päivittäispalaverissa voidaan asioita käsitellä eri tavoin ja varsin käyttökelpoinen tapa on käydä läpi seuraavat kolme kysymystä:

- Mitä tein eilen auttaakseni kehitystiimiä saavuttamaan sprintin tavoitteen?
- Mitä aion tehdä tänään auttaakseni kehitystiimiä saavuttamaan sprintin tavoitteen?
- Havaitsenko mitään estettä, joka estää minua tai kehitystiimiä saavuttamasta sprintin tavoitteen? (9, s. 12.)

Päivittäispalaveri on kehitystiimin sisäinen tapahtuma. Scrummasterin tehtävänä on luoda olosuhteet palaverille, valmentaa Scrum-tiimiä pitämään palaveri enintään 15 minuutin mittaisena sekä estää mahdollisten paikalla olevien ylimääräisten henkilöiden häiritsevä osallistuminen kokoukseen. (9, s. 12.)

Sprintin katselmointi pidetään sprintin lopussa ja siihen osallistuu Scrum-tiimin lisäksi tuoteomistajan kutsumana sidosryhmien edustajia. Katselmoinnin kesto pidetään korkeintaan neljässä tunnissa kuukauden mittaisessa sprintissä. Katselmoinnissa käydään läpi seuraavat kohdat:

- Tuoteomistaja selittää mikä osa työstä on ”valmista” ja mikä ei ole ”valmista”.
- Kehitystiimi keskustelee, mikä toteutuksessa meni hyvin, mitä ongelmia se kohtasi ja kuinka ongelmat ratkaistiin.
- Kehitystiimi esittelee ”valmiin työn ja vastaa inkrementtiin liittyviin kysymyksiin.
- Tuoteomistaja kertoo tuotteen kehitysjonon tilanteen ja arvioi todennäköistä valmistumisajankohtaa perustuen tähänastiseen edistymiseen.
- Koko ryhmä pohtii, mitä voidaan ja kannattaa tehdä seuraavaksi, jotta sprintin katselmointi antaa hyvän pohjan seuraaville sprintin suunnittelupalaverille.
- Tarkistetaan, kuinka markkinatilanne tai tuotteen mahdolliset käyttötavat ovat vaikuttaneet siihen mikä olisi arvokkainta toteuttaa seuraavaksi, ja

- Tarkistetaan tuotteen julkaisun aikataulu, budjetti, markkinatilanne ja potentiaaliset toiminnallisuudet. (9, s. 13–14.)

Katselmoinnissa tuloksena saadaan tuotteelle päivitetty kehitysjojo, jossa on todennäköisimmät kehitysjojon kohdat seuraavalle sprintille. (9, s. 14).

Sprintin retrospektiivi järjestetään sprintin katselmoinnin ja seuraavan sprintin suunnittelupalaverin välissä. Palaverin kesto tulee olla korkeintaan kolme tuntia. Sprintin retrospektiivin tarkoituksena on

- tarkastella, kuinka edellinen sprintti sujui liittyen ihmisiin, yhteistyöhön, prosessiin ja työkaluihin
- tunnistaa asiat, jotka sujuivat hyvin sekä määritellä tärkeimmät parannukset
- luoda suunnitelma Scrum-tiimin työskentelytapojen parantamiseksi. (9, s. 14.)

Retrospektiivin tuloksena Scrum-tiimi kykenee tunnistamaan prosessin parannukset, joita tiimi aikoo toteuttaa seuraavan sprintin aikana. (9, s. 14).

3.1.4 Scrumin tuotokset

Scrum on lähinnä viitekehys, jonka avulla tuotehallinta ja työmenetelmien suorituskyky saadaan läpinäkyväksi. Tämän vuoksi Scrumin tuotoksetkin on suunniteltu läpinäkyvyyden maksimoimiseksi. Scrumin tuotoksia ovat

- tuotteen kehitysjojo
- sprintin kehitysjojo
- inkrementti. (9, s. 14.)

Tuotteen kehitysjojon ensimmäisessä versiossa esitetään vain sillä hetkellä parhaiten tunnetut vaatimukset. Tuotteen kehitysjojo jatkaa kehittymistään koko tuotteen elinkaaren ajan eikä se ole ikinä valmis. Tuotteen kehitysjojonsta löytyvät kaikki toiminnot, ominaisuudet, parannukset sekä korjaukset, jotka toteutetaan tulevissa inkrementteissä. Kehitysjojon kohta sisältää kuvauksen, järjestyksen, arvon työmäärästä ja arvon. Tuotteen kehitysjojo jalostuu, kun siihen lisätään yksityiskohtia ja työmääräarvioita sekä kohtien keskinäistä järjestystä. Tuotteen kehityksen edetessä seuraavaan sprinttiin valitaan tuotekehitysjojon kohdat, jotka ovat jalostuneet niin

pieniksi, että ne voidaan tehdä valmiiksi sprintin loppuun mennessä. Tuoteomistaja vastaa tuotteen kehitysjonon sisällöstä, saatavuudesta ja järjestämisestä. Kehitystiimin vastuulla on antaa työmääräarviot. (9, s. 15.)

Sprintin kehitysjonoon kuuluu sprinttiin valitut tuotteen kehitysjonon kohdat sekä suunnitelmat inkrementin toimittamisesta. Sprintin kehitysjono on ennuste seuraavan inkrementin sisältämästä toiminnallisuudesta sekä siihen vaadittavasta työmäärästä. Sprintin kehitysjono muuttuu päivittäispalavereissa sprintin aikana ja ainoastaan kehitystiimi voi tehdä muutoksia tänä aikana. Sprintin kehitysjonon on oltava riittävän yksityiskohtainen työmäärän ja työn edistymisen seuraamiseksi. (9, s. 16.)

Inkrementissä on mukana kaikki kyseisen sprintin sekä myös aikaisempien sprinttien aikana valmistuneet kehitysjonon kohdat. Inkrementissä olevien kohtien tulee ”valmiita” siten kuin Scrum tiimi on määritellyt itselleen mitä ”valmis” tarkoittaa. Tiimillä tulee olla yhteinen näkemys mitä tarkoittaa, kun joku kohta on tehty valmiiksi. Samaten jos samaa inkrementtiä valmistele useampia Scrum-tiimejä, täytyy kaikilla olla sama ymmärrys valmiin määritelmästä. (9, s. 17.)

3.1 Kanban

3.2.1 Kanbanin periaatteet

Kanban menetelmä esitetään lyhyesti ja selkeästi kolmen perussäännön kautta:

1. Näkyvöitä työnkulku
 - Pilko työt sopivankokoisiin tehtäviin.
 - Kirjaa kukin tehtävä paperilapulle ja kiinnitä laput Kanban (tai Scrum) -tauluun.
 - Kuvaa taulun sarakkeilla missä työvaiheessa kukin tehtävä on.
2. Määritä WIP taulun jokaiselle sarakkeelle
 - WIP (Work in Progress) tarkoittaa suurinta määrää tehtäviä, joita kyseisessä sarakkeessa saa kerrallaan olla, jottei sarakkeeseen kasaantuva työ hidasta muita työvaiheita.

3. Mittaa tehtävien läpimenoajat

- Läpimenoaika (Lead Time) tarkoittaa keskimääräistä aikaa yhden tehtävän valmistumiseen. Kun olet mitannut läpimenoajan, optimoi pikkuhiljaa prosessia ja kokeile erilaisia WIP-arvoja läpimenoajan lyhentämiseksi ja ennustettavuuden parantamiseksi. (10.)

David Anderson esitteli Kanban metodin vuonna 2003. Hän esitteli kolme sääntöä yllä olevien kolmen yleisesti käytetyn perussäännön lisäksi.

4. Pidä käytännöt selkeinä

- Yksiselitteisesti ymmärrettävä ja näkyvä tehtävän liikkuminen läpi Kanban-taulun auttaa tiimejä keskustelemaan entistä puolueettomammin ja totuudenmukaisemmin prosessissa olevista esteistä ja ongelmista.

5. Toteuta palautteesta saatu kokemus

- Tiimi arvioi omien mittareidensa mukaan mikä systeemissä toimii ja mikä taas vaatii muutoksia. Tähän palautteeseen perustuvat muutokset parantavat suorituskykyä tulevaisuudessa.

6. Paranna yhteistyötä

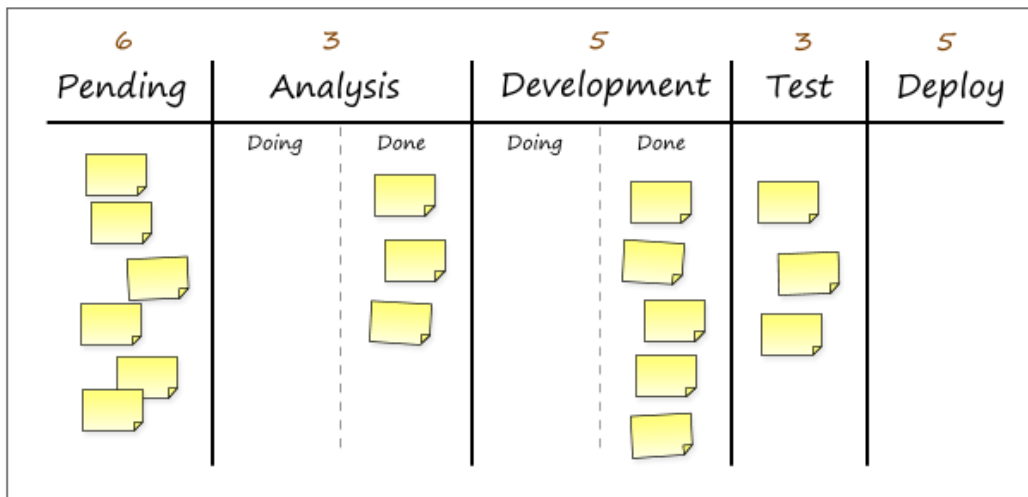
- Kanbanin avulla tiimillä on yhteinen käsitys projektista, sen osa-alueiden edistymisestä sekä projektin tämän hetkisestä tilasta. Tämän ymmärryksen pohjalta tiimin on helpompi saavuttaa yhteisymmärrys senhetkisistä haasteista sekä kehittää keinoja niistä selviytymiseen. (11.)

3.2.2 Kanban-taulu

Yksinkertaisimmillaan Kanban-taulu on seinälle sijoitettu fyysinen taulu, johon on sijoitettu kortteja numerolla varustettuihin sarakkeisiin. Kortit edustavat kehitysprosessin läpi kulkevia työtehtäviä ja yläreunassa oleva numero kuvaa keskeneräisten töiden rajoitettua lukumäärää. Tämä rajoitus

työtehtäviin (WIP, work-in-progress) on Kanban-taulun merkittävin ero muihin visuaalisiin työn edistymisen kuvausjärjestelmiin. Rajoitus estää ylituotannon missään kohdassa ja paljastaa helposti tuotannon pullonkaulojen syntyminen. (12.)

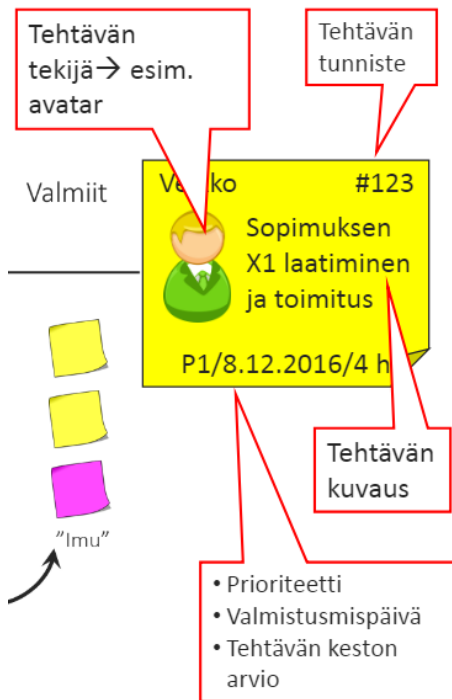
Kuvassa 3 on esitetty esimerkki Kanban-taulusta. Kuvan mukaisessa tilanteessa työt eivät etene missään ennen kuin testaus on saanut yhden kortin siirrettyä seuraavaan vaiheeseen. Tässä tilanteessa muiden tulisi miettiä keinoja, joilla testausta voidaan auttaa. Kuvassa yhdessä sarakkeessa olevat työt on jaettu tekeillä oleviin ja tehtyihin töihin. Sarakkeen yllä oleva rajoitus koskee näitä molempia. Kun yksi testauksen alla olevista korteista siirtyy käyttöönottoon, niin kaikkiin muihin vaiheisiin voidaan "imeä" uusi kortti edeltävästä sarakkeesta. Tässä imuohjauksessa ketjun seuraava vaihe on tavallaan aina asiakas, jonka tarve määrää edellistä vaihetta. (12.)



KUVA 3. Kanban-taulu (12)

Ohjelmistotuotannossa ja muussakin toiminnassa on tilanteita, jolloin jokin kriittinen tehtävä on saatava valmiiksi mahdollisimman pian. Silloin on sovittu, että se voi ohittaa tärkeysjärjestyksessä kaikki muut tehtävät. Tällainen tilanne saattaa olla esimerkiksi tuotantoon asti päässyt ohjelmointivirhe. Näitä tilanteita varten Kanban-tauluun voidaan perustaa uimaradaksi tai tulipalokaistaksi kutsuttu ohituskaista. Nämä ohituskaistalla olevat työt menevät kaikkien muiden töiden edelle. Myös ohituskaistalle päästettävien töiden määrää voidaan ja tulisikin rajoittaa yhteisellä sopimuksella. (13, s. 39.)

Kuvassa 4 on esitetty esimerkki Kanban-taulun kortista. Kortin väri voi vaihdella riippuen tehtävän luonteesta: normaali tehtävä, reklamaatio, uusi idea, kiireellinen tehtävä yms. Korttiin olisi merkittävä ainakin tehtävän nimi ja muu identifikaatio, tehtävän toivottu valmistumispäivä, lyhyt kuvaus tehtävästä, tehtävään nimetyt henkilöt, arvio tehtävän kestosta sekä tehtävän prioriteetti. (13, s. 10.)



KUVA 4. Esimerkki Kanban-taulun kortista (13, s. 10)

Kanban on alun perin lähtöisin tuotantolinjalta ja tuotantotyölle on ominaista, että tuotteet ovat jatkuvasti samoja tai samankaltaisia sekä tuotantoprosessi on yhtenäinen. Tuotantolinjalla läpimenoaika voidaan käyttää suoraan mittarina prosessin kehittymisestä. Luovassa kuten ohjelmistojen kehityksessä työ tai tuotettava asia ei ole koskaan samanlainen. Tämä kuuluu luovan työn luonteeseen myös jopa testauksessa ja virheen korjauksessa. Luovassa työssä ja tuotantolinjalla arvon käsite on erilainen. Tuotantotyössä vaihtelu ja muutokset aiheuttavat hukkaa, kun taas luovassa työssä arvoa saavutetaan kyvillä ratkaista ongelmia. (14.)

Yksi Kanbanin peruseräiteitä on jatkuva prosessin kehittäminen, ja se pitäisi olla ohjenuorana myös luovassa työssä käytetyssä Kanbanissa. Kanbanin tarkoitus on optimoida tuotteen läpimenoa ja arvon tuottamista eikä resursseja. Tämän takia WIP rajojen yhteenlaskettu lukumäärä ei saa ylittää tiimin henkilömäärää. Luovassa työssä tehtävät ovat usein vaihtelevat mittaisia, joten eri tasoisia tehtäviä ei pitäisi laittaa samalle Kanban-taululle vaan tulisi tehdä useita eritasoisia

Kanban-tauluja. Kanbanin käyttö ei saa siloittaa organisaatiota, vaan koko tiimin tulee osallistua projektin kaikkiin vaiheisiin ketterien menetelmien perusajatuksen mukaisesti. Arvon muodostumisen ymmärtäminen sekä pyrkiminen jatkuvaan oppimiseen ovat perusedellytyksiä Kanbanin menestyksekkäälle käytölle luovassa työssä. (14.)

4 VIRTUAALISET PROJEKTIHALLINTATYÖKALUT

Tuotekehityksessä tarvitaan aina varsinaisten kehitystyökalujen lisäksi projektihallintatyökaluja. Ketterän kehityksen mukaisessa toiminnassa projektihallinta on minimoitu ja koko tiimi on siitä vastuussa. Projektihallintatyökaluja käytetään ensisijaisesti kehitystyötehtävien seurantaan ja hallintaan. Yksinkertaisimmillaan tämä voidaan tehdä seinälle sijoitetulla, fyysisellä tehtävätaululla. Fyysinen taulu on helppo ja halpa hankkia, helppo käyttää sekä yhteiseen tilaan sijoitettuna herättää keskustelua. Kuitenkin etäkäyttö, historian seuranta ja integraatio yrityksen muihin järjestelmiin ei onnistu.

Kehityshistorian seuranta on usein tärkeää ohjelmistokehityksessä. Lisäksi nykyisin tiimin jäsenet eivät välttämättä työskentele samoissa tiloissa. Ketterän kehityksen periaatteiden mukaan ohjelmistokehityksessä on noussut merkittävään asemaan ihmisten välisen yhteistyön ja jatkuvan oppimisen korostaminen. Tuotekehitysprojektien hallintaan käytetäänkin erilaisia virtuaalisia työkaluja, joista neljä erilaista valittiin vertailtavaksi tässä työssä.

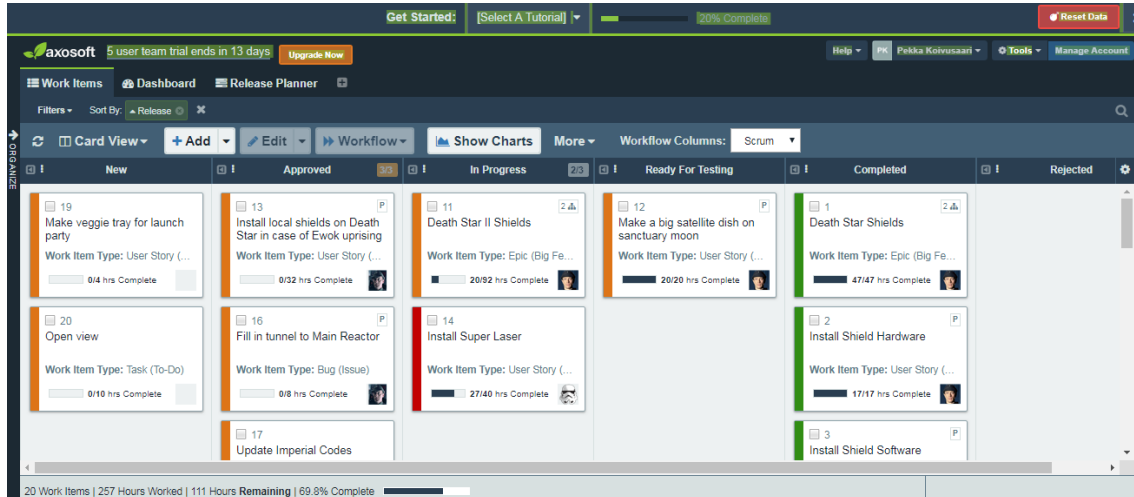
4.1 Vertailtavat työkalut

4.1.1 Axosoft

Axosoft on samannimisen, Axosoft-yhtiön kehittämä projektihallintaohjelmisto, jonka ensimmäinen versio, Axosoft V1.0 (silloiselta nimeltään OnTime), julkaistiin vuonna 2002. (15).

Axosoft kertoo sivuillaan tehtävien korttinäkömän olevan täysin vuorovaikutteinen Kanban-taulu, jolla tehtävien eteneminen ja status voidaan esittää visuaalisesti. Taulussa on selvästi esitettyinä WIP-rajat kullekin vaiheelle eikä tehtävää pysty siirtämään eteenpäin, mikäli vaiheessa on jo WIP-rajaa täynnä. Tehtävien siirtäminen vaiheesta toiseen onnistuu helposti paitsi kortin statusta muuttamalla myös hiirellä siirtämällä. Axosoft mahdollistaa sisäänrakennettujen tai räätälöityjen työvirtojen rakentamisen projektin sisälle. Näin voidaan toteuttaa esimerkiksi nopeaa reagointia tehtäviin vaativia uimaratoja. Axosoft mainostaa sivuillaan tuotteensa pystyvän tukemaan Agilen mukaisen ketterän ohjelmistokehityksen mukaisia prosesseja. Axosoftissa onkin paljon sekä Scrumin että Kanban filosofian mukaisia ominaisuuksia. (15, linkki Feature Tour.)

Kuvassa 5 on esitetty esimerkki Axosoft- ohjelmassa käytössä olevasta taulusta. Taulu oli esimerkinomainen projekti, jota kahden viikon kokeilu jaksolla oleva käyttäjä voi hallinnoida.



KUVA 5. Esimerkki Axosoft-taulusta (15)

4.1.2 HacknPlan

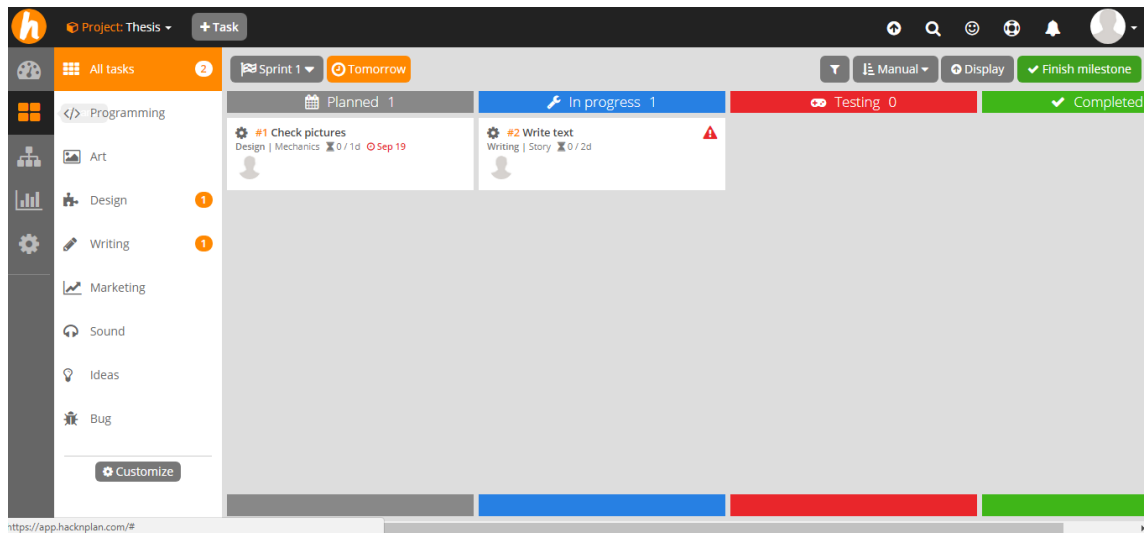
Espanjalainen pelinkehittäjä ja Scrummaster Chris Estevez ei löytänyt tarpeisiinsa sopivaa työkalua ja hän alkoi kehittää omaa projektinhallinta järjestelmää. HacknPlanin avoin beetaversio kulkauttiin 2015 ja HacknPlan 1.0 vuonna 2017. HacknPlanin ansaintamalli perustuu ilmaisversioon josta kuukausi/vuosimaksua vastaan on saatavilla kehittyneempi versio. (16, linkki Blog.)

HacknPlan on SaaS-projektinhallintasovellus pelinkehitysstudioille. HacknPlanin omien sivujen mukaan se saanut vaikutteita muista työkaluista, kuten Jirasta tai Trellosta, mutta työnkulun uudelleensuunnittelu ja joidenkin ominaisuuksien lisääminen on tehnyt siitä intuitiivisemmän ja tuottavamman työkalun pelikehitykseen. HacknPlanin suurin ero muihin ratkaisuihin on pelisuunnittelu dokumentaation integrointi osaksi muuta projektin tehtävienhallintaa. (16, linkki Read our FAQ.)

HacknPlan ei omien sivujensa mukaan perustu mihinkään tiettyyn filosofiaan. HacknPlan on kuitenkin voimakkaasti inspiroitunut Agilen periaatteista ja käyttää Kanban-tauluja pääasiallisena ratkaisunaan työn edistymisen seurantaan. HacknPlanin ideologian mukaan työ edistymistä

kuvaavalla taululla tulee olla vain rajoitettu määrä tehtäviä pitämään projekti selkeänä. Projekti jaetaan etappeihin tai iteraatiokierroksiin, jotka sisältävät rajatun määrän työtä kunakin aikajaksona. Idealisesti jokaisen etapin tuloksena tulisi olla pelattava versio tekeillä olevasta pelistä. (16, linkki [Read our FAQ.](#))

Kuvassa 6 on esitetty esimerkki HacknPlanin taulusta. HacknPlanin ominaisuuksista saa hyvän kuvan, sillä ilmaisversion käyttäjällekin on saatavilla sängen kattavat ominaisuudet.



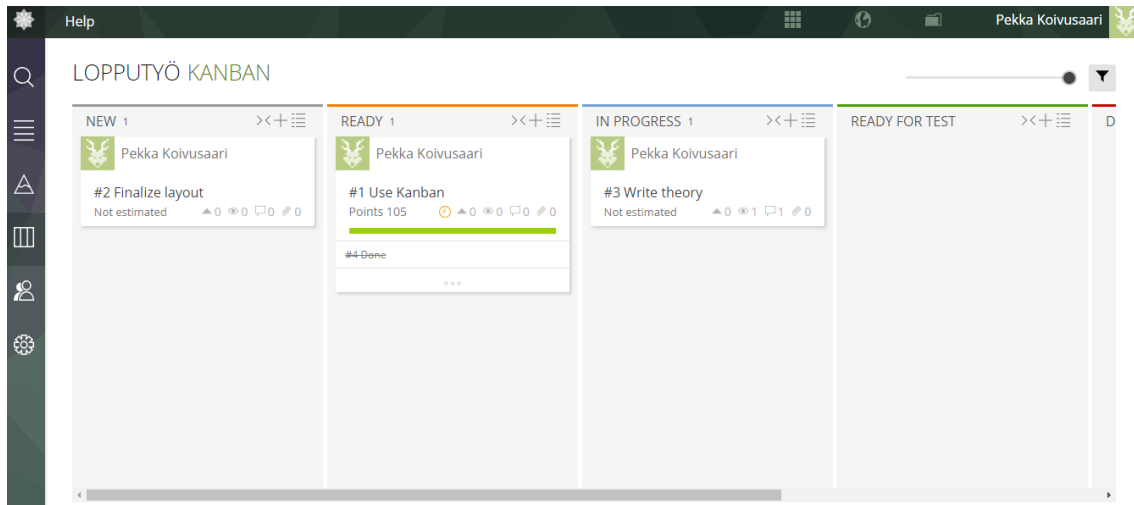
KUVA 6. Esimerkki HacknPlan-taulusta (16)

4.1.3 Taiga.io

Taiga.io on Enrique Posnerin ja Pablo Ruiz-Murquizin alun perin omien projektien hallintaan perustama avoimen lähdekoodin yhteisössä toimiva ohjelmisto. Taiga lupaa projektia perustettaessa mahdollisuuden käyttää sekä Scrum-viitekehyksen mukaista prosessia että Kanban-taulua. Tässä vaiheessa ei valinnalla ole suurta merkitystä koska myöhemmin valintaa voidaan muuttaa ottamalla käyttöön molemmat tai vaihtamalla valintaa. (17, linkki [Support.](#))

Taiga.io:ssa käyttäjä voi olla ilmaisessa versiossa omistajana vain yhdessä yksityisessä projektissa ja samassa projektissa voi olla korkeintaan kolme jäsentä. Julkisia projekteja voi sen sijaan olla yhden henkilön omistuksessa ääretön määrä, samoin kuin yksi henkilö voi olla mukana lukuisissa projekteissa. (17, linkki [Support.](#))

Kuvassa 7 on esitetty esimerkki Taiga.io-taulusta. Taiga.io tarjosi myös mahdollisuuden luoda oma projekti varsin laajoin ominaisuuksin.



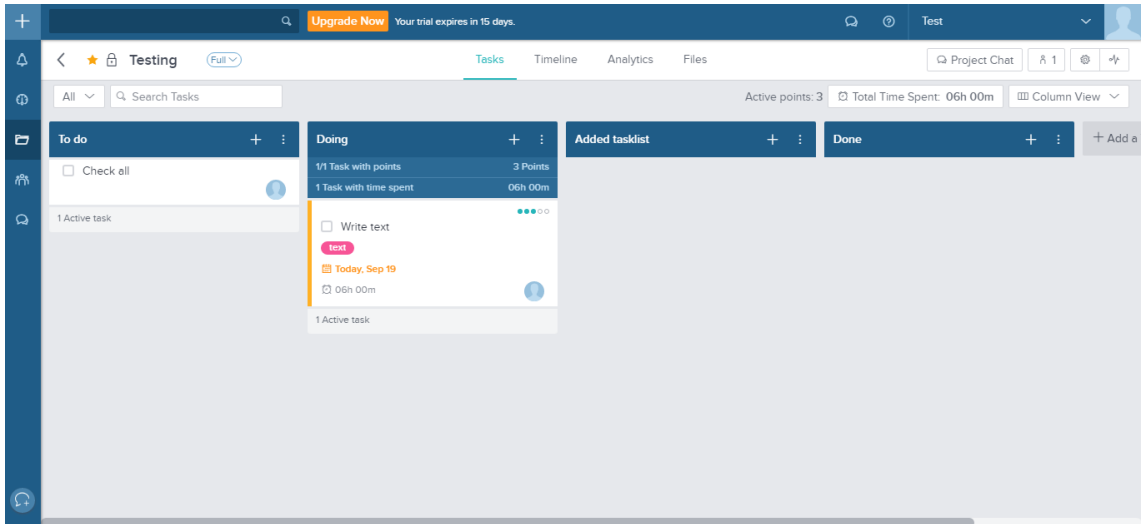
KUVA 7. Esimerkki Taiga.io-taulusta (17)

4.1.4 Taskworld

Taskworld on syntynyt vuonna 2006 sarja yrittäjä Fred Muoawadin tarpeesta löytää käyttöönsä riittävän toimiva tehtävienhallintaohjelmisto. Ongelma päätettiin ratkaista itse ja sen parissa alkoi työskentelemään Synergia One yhtiön IT tiimi. Myöhemmin ohjelman nimeksi muotoutui Taskworld ja yhtiön nimeksi Taskworld Inc. (18, linkki About Us.)

Taskworld tarjoaa sivuillaan varsin kattavan käyttöoppaan. Sen perusteella Taskworldin käyttö noudattaa Kanban filosofiaa visuaalisine Kanban tauluineen. Varsinainen käyttö on hyvin ohjeistettu tiimin työtilan luomisesta lähtien. Tehtäviä voidaan esimerkiksi päivittää samanaikaisesti useammassa projektissa. (18, linkki User Guide.)

Kuvassa 8 on esitetty esimerkki Taskworldin taulusta. Taskworldissa oli mahdollisuus luoda kokeilu jaksolla useampia projekteja, mikä ominaisuus puuttui muista tutkituista ohjelmista.



KUVA 8. Esimerkki Taskworld-taulusta (18)

4.2 Ominaisuuksien vertailu

Taulukossa 1 on vertailtu valittujen tuotteiden ominaisuuksia. Taulukkoon valittiin vertailtaviksi ominaisuuksiksi joitain teoriaosuudessa Scrumiin ja Kanbaniin pohjautuvia ominaisuuksia. Lisäksi arvioitiin joitain yleiseen käyttöön liittyviä asioita, jotka tulisivat esiin, mikäli oltaisiin valitsemassa työkaluja todellisessa tilanteessa.

TAULUKKO 1. Valittujen ohjelmien vertailua

	Axosoft	HacknPlan	Taiga.io	Taskworld
Taulu				
Scrum	X	X	X	-
Kanban	X	X	X	X
Projektin vaiheet muokattavissa	X	-	-	X
Liitetiedostot tehtävään	X	X	X	X
WIP-rajat Kanban-tauluun	X	-	-	-
Mobiilisovellus				
Apple	X	-	X	X
Android	X	-	X	X
Ympäristö				
Pilvipalvelu	X	X	X	X
Asennus omalle palvelimelle	X	-	X	X
Sähköposti muistutus	X	X	-	X
Kalenteri	X	X	-	X
Chat	-	-	X	X
API	X	-	X	X
Integraatio muihin ohjelmiin	Trello GitHub Bitbucket Slack	Google GitHub Bitbucket Slack	Trello Jira Github Asana	Box Dropbox Google Drive
Avoin lähdekoodi	-	-	X	-

Kuten taulukosta 1 havaitaan, monet oleelliset ominaisuudet ovat mukana kaikissa tarkastelluissa ohjelmissa. Taulukkoon luetteloimattomista ominaisuuksista esimerkiksi työtaulun vetäminen hiirellä työvaiheesta toiseen onnistuu kaikissa ohjelmissa. Tilanne myös päivittyi samanaikaisesti mobiilisovelluksessa ja päinvastoin. Projektin luominen ja tehtävätaulun, käyttäjän tai liitetiedoston lisääminen oli kaikissa tarkastelluissa ohjelmissa suhteellisen selkeää ja helposti opittavissa.

Kaikki ohjelmat kertoivat omilla sivuillaan soveltuvansa Kanbanin mukaiseen työskentelyyn, mutta tässä oli kyllä joko venytetty Kanbanin perussääntöjä tai sitten ominaisuudet eivät ole ainakaan selkeästi havaittavissa. Kanbanin ensimmäinen perussääntö työnkulun visualisoinnista tulee kyllä kaikissa toteutettua siinä mielessä, että työvaiheet ovat samassa taulussa näkyvissä. Työn visualisointiin kuuluu kylläkin myös mahdollisuus muokata työvaiheet sopivan kokoisiksi paloiksi oman työnkulun mukaan. Tämä ominaisuus löytyy vain Axosoftista ja Taskworldista. HacknPlanissa ja Taiga.io:ssa työvaiheet ovat kiinteästi määrättyjä eikä niiden nimeä, lukumäärää tai järjestystä pysty muuttamaan.

Visualisointiin voidaan ajatella kuuluvan myös työkortti ja sen sisältö. Fyysisen taulun hyvänä puolena voidaan pitää tärkeiden tehtävien korostamista esimerkiksi punaisella kortilla. Yhdestäkään tarkastellusta ohjelmasta ei löytynyt niin selkeästi erottuvaa priorisoinnin esille tuovaa keinoa. Kaikissa toki tehtävät voidaan priorisoida eri tärkeysjärjestykseen ja se tulee myös korttiin näkyville.

Kanbanin toisen perussäännön mukaan on määriteltävä suurin sallittu määrä yhdenaikaisia tehtäviä kussakin vaiheessa. Yllättäen tämä ominaisuus löytyy vain Axosoftista, missä ohjelma ilmoittaa, että vaiheeseen ei voi siirtää uutta tehtävää. Axosoftissa pystyy muuttamaan WIP-rajaa, kuten Kanbanin periaatteiden mukaan kuuluukin. Axosoft myös estää uuden tehtävän siirtämisen seuraavaan vaiheeseen, mikäli WIP-rajaa on jo saavutettu.

Kanbanin kolmantena perussääntönä on työnkulun ja läpimenoajan mittaaminen. Parhaat ratkaisut tähän tarjoaa Axosoft, josta löytyy muun muassa burndown-esitys, joka graafisesti kuvaa jäljellä olevan työmäärää suhteessa aikaan. Muissakin työkaluissa työn edistymistä voidaan seurata, mutta prosessin optimoimiseen Axosoft tarjoaa eniten valmiita välineitä.

Kanbanin kolme viimeistä perussääntöä käytäntöjen pitämisestä selkeinä, palautteen toteuttamisesta ja yhteistyön parantamisesta toteutuvat pitkälti enemmänkin tiimin oman

käyttämisen mukaan kuin työkalun toimintojen mukaan. Kaikki tarkastellut työkalut tarjoavat mahdollisuuden toteuttaa nämä kolme Kanbanin perussääntöä.

5 YHTEENVETO

Opinnäytetyön tavoitteena oli tutkia modernien tuotekehitysprojektien työkalujen toimintoja, eroja ja käytettävyyttä. Aiheena se oli kiinnostava ja varsinkin teoriaosuuteen tutustuminen toi esiin uusia näkökulmia työn tehokkuuteen. Työn teoriaosassa käsiteltiin syvemmin Agile-kehitystä, Lean-filosofiaa ja näihin liittyviä Scrum-viitekehystä sekä Kanban-ajattelua. Työn tarkoituksena oli tutkia neljää työkalua ja saada esille niiden hyviä ja huonoja puolia.

Teoriaosuuteen löytyi tietoa eri lähteistä varsin hyvin, mutta työkaluista löytyi informaatiota pääasiassa niiden omilta sivuiltaan ja siihen on tietenkin aina suhtauduttava tietyllä varauksella. Ohjelmat olivat käytettävissä yleensä kahden viikon ilmaisella jaksolla ja se on suhteellisen lyhyt aika syvällisempään analyysiin ja käyttäjäkokemukseen. Teoriaosuudessa käsiteltiin tuotekehitysprojektin eteneminen sekä Scrumin ja Kanbanin mukaisesti. Valittuja ohjelmistoja tarkasteltiin vain Kanbanin perussääntöihin peilaten, sillä ilman mitään olemassa olevaa konkreettista työnkulkua on vaikea luoda käsitystä ohjelman taipumisesta Scrum viitekehukseen.

Aihe oli opintojen kautta jollain lailla tuttu, mutta minulla ei ole varsinaista työelämästä tullutta kokemusta ohjelmistokehitystyöstä, joten varsinaiset käytännön kokemukset puuttuvat tästä opinnäytetyöstä. Löysin kuitenkin uusia asioita ja pohdinnan aiheita työnkulkuun ja sen seurantaan liittyvistä säännöistä. Varsinkin Kanbanin mukainen toiminta vaikuttaa yrityksen tai muun yhteisön toimintamallina erittäin tehokkaalta ja koko tiimin kannalta toimivalta sekä yhteistyötä korostavalta. Toisaalta Kanbanin käyttö vaatii monesti vakiintuneiden toimintatapojen muuttamista yhteisen tavoitteen saavuttamiseksi.

LÄHTEET

1. Juvonen, Rami 2018. Ohjelmistoprojektien sudenkuopat ja miten ne vältetään. Helsinki: BoD.
2. Ketterän ohjelmistokehityksen julistus. 2018. Agile Alliance. Saatavissa: <http://agilemanifesto.org/iso/fi/manifesto.html>. Hakupäivä 13.6.2018.
3. Lehtonen, Teijo – Tuomivaara, Seppo 2015. Sulautettujen järjestelmien ketterä käsikirja. Turun yliopisto. Saatavissa: <http://trc.utu.fi/embedded/kasikirja/>. Hakupäivä 14.6.2018.
4. Oppenheim, Bohdan 2011. Lean for Systems Engineering with Lean Enablers for Systems Engineering. New Jersey: John Wiley & Sons.
5. Lean. 2018. Wikipedia. Saatavissa: <https://fi.wikipedia.org/wiki/Lean> . Hakupäivä 18.6.2018
6. Digitalisoinnin opas, Ketterät menetelmät, agile, LEAN ja scrum. 2018. Itewiki. Saatavissa: <https://www.itewiki.fi/opas/ketterat-menetelmat-agile-lean-ja-scrum/>. Hakupäivä 18.6.2018.
7. What is Lean?. 2018. Lean Enterprise Institute. Saatavissa: www.lean.org/WhatsLean/. Hakupäivä 18.6.2018.
8. Opelt, Andreas – Gloger, Boris – Pfarl, Wolfgang – Mittermayr, Ralf 2013. Agile Contracts, Creating and Managing Successful Project with Scrum. New Jersey: Wiley & Sons.
9. Sutherland, Jeff - Shwaber, Ken 2017. Scrum-opas, Scrumin määritelmä ja pelisäännöt. Saatavissa: <https://scrumwell.files.wordpress.com/2018/03/2017-scrum-guide-fi-v1-02.pdf>. Hakupäivä 16.8.2018.
10. Lekman, Lare. 2009. Mikä ihmeen Kanban?. Saatavissa: <https://lekman.fi/2009/09/26/mika-ihmeen-kanban/> . Hakupäivä 15.6.2018.
11. Orzen, Michael - Paider, Thomas 2017. The Lean IT Field Guide, A Roadmap for Your Transformation. Boca Raton, Fl: CRC Press.
12. Peterson, David 2015. What is Kanban? Kanban blog. Saatavissa: <http://kanbanblog.com/explained/index.html>. Hakupäivä 17.8.2018.
13. Moisio, Jussi 2017. Kanban-taulut asiantuntijatyön tilanneohjauksessa. Saatavissa: <https://docplayer.fi/62106535-Kanban-taulut-asiantuntijatyon-tilanneohjauksessa.html>. Hakupäivä 20.8.2018.

14. Hämäläinen, Henri 2016. Tehokas Kanban vai kasa lappuja seinällä - webinaari.
Saatavissa: <https://contribyte.fi/2016/12/15/tehokas-kanban-vai-kasa-lappuja-seinalla/>.
Hakupäivä 11.9.2018.
15. Axosoft. Saatavissa: <https://www.axosoft.com/>. Hakupäivä 14.9.2018.
16. HacknPlan. Saatavissa: <https://hacknplan.com/>. Hakupäivä 14.9.2018.
17. Taiga.io. Saatavissa: <https://taiga.io/>. Hakupäivä 18.9.2018.
18. Taskworld. Saatavissa: <https://taskworld.com/>. Hakupäivä 19.9.2018.