



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Aki Itävaara

KONEPAJAN VARASTONHALLINTA-
SOVELLUKSEN MÄÄRITTELY JA
KÄYTTÖÖNOTTO

Tekniikka
2016

TIIVISTELMÄ

Tekijä	Aki Itävaara
Opinnäytetyön nimi	Konepajan varastohallintasovelluksen määrittely ja käyttöönotto
Vuosi	2016
Kieli	suomi
Sivumäärä	39
Ohjaaja	Timo Kankaanpää

Työssä toteutettiin ohjelmistoprojektin määrittely ja käyttöönotto konepajaympäristössä. Ohjelmiston tarve tuli suoraan asiakkaalta ja sen läpivientiin pyrittiin käyttämään ketteriä menetelmiä parhaan lopputuloksen saavuttamiseksi.

Ongelman ytimenä oli materiaalin käytön tehostaminen tehtaalla. Siihen käytettiin aluksi perinteisiä menetelmiä, mutta viimeistään käyttäjähaastattelujen jälkeen selvisi erillisen ohjelmiston tarve. Kartoitusprosessin jälkeen valittiin ohjelmistotalo, joka kykeni käyttämään ketteriä menetelmiä, tässä tapauksessa scrumia. Siinä tulokset pyritään tuomaan nopealla syklillä asiakkaan arvioitavaksi.

Kommunikoinnin tärkeys, niin toimittaja-asiakassuhteessa kuin hallinto-tuotanto-suhteessa, osoittautui jopa käytettyä kehitysmenetelmää tärkeämmäksi. Sovelluksen käyttö tabletilla osoittautui yllättävän helpoksi sisäistää.

ABSTRACT

Author	Aki Itävaara
Title	Deployment and Specification of Material Control Software in Metal Industry
Year	2016
Language	Finnish
Pages	39
Name of Supervisor	Timo Kankaanpää

The specification and deployment project of a material control software in metal industry were key points in this thesis. The need for software features came from the customer. Agile development methods were used to ensure best results.

The main problem was the material usage and its rationalization in the factory. Initially, traditional methods were used. After user interviews the customer was sure, that new software is needed. Planning phase was thorough and we decided to use a small software house that was capable of using agile methods, in this case Scrum. In Scrum results are shown to the customer in a fast phase.

The importance of communication in all cases was more important than the development method. Using software with a tablet was surprisingly easy.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	6
2	TEKNOLOGIAT.....	7
2.1	BPNM.....	7
2.2	UML.....	7
2.3	SCRUM.....	8
2.3.1	Työskentelytapa.....	9
2.3.2	Srumin roolit.....	9
2.3.3	Scrumin tuotokset.....	11
2.3.4	Scrumin tapahtumat.....	12
2.4	Redmine.....	14
2.4.1	Tärkeimmät ominaisuudet.....	15
2.5	MES.....	16
3	TUOTANNONOHJAUS YRITYKSESSÄ.....	17
4	MES-MATERIAALINHALLINNAN MÄÄRITTELY.....	21
4.1	Haastattelut.....	22
4.2	Osta vai tee itse.....	25
4.3	Päätöksenteko.....	26
4.4	Projektisuunnitelma.....	26
5	MATERIAALIVIRTA OSANA TILAUS-TOIMITUSPROSESSIA.....	28
5.1	Entinen myyntiprosessi.....	28
5.2	Uusi myyntiprosessi.....	29
6	ALIHANKINTAPROSESSI.....	31
6.1	Roolit.....	31
6.2	Sprintin suoritus ja kommunikointi.....	31
7	PROJEKTIN LÄPIVIENTI YRITYKSESTÄ.....	32
7.1	Henkilöstön sitoutus.....	32
7.2	Pohjatiedot järjestelmää varten.....	32

7.3 Sovelluksen käyttöönotto.....	33
8 TESTAUS.....	35
9 YHTEENVETO, LOPPUPÄÄTELMÄ JA JATKOTOIMENPITEET	37
LÄHTEET.....	39

1 JOHDANTO

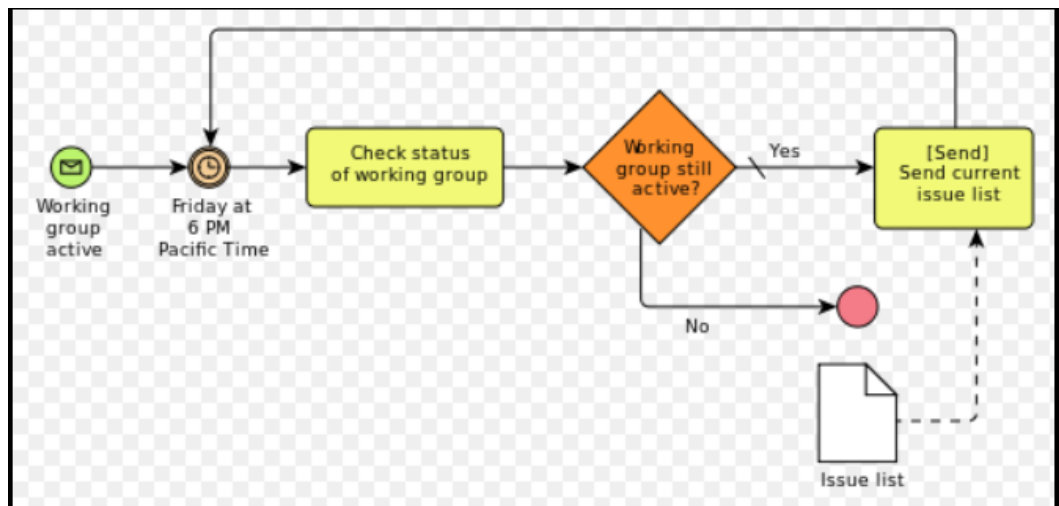
Opinnäytetyön aiheena oli ohjelmiston kehittämisprojektin läpivienti asiakkaan projektipäällikön näkökulmasta. Käytetyn ohjelmistokehityksen mukaan näkökulma oli Product Owner. Ohjelmistoprojekti käsitteli materiaalinhallintaohjelmistoa Pohjanmaalla toimivalle teollisuusyritykselle. Product owner -roolissa tehtiin määrittelyt ja siitä syntyneitä ohjelmistoprojektia vietin Scrum-mallin mukaisesti läpi ohjelmistotoimittajan kanssa. Product owner -roolin lisäksi työ käsittelee käytetyt teknologiat ja yrityksen sisäiset prosessit uuden ohjelmiston hankinnan yhteydessä.

Teoriaosuudessa käsitellään käytettyjä menetelmiä ja nykyisiä ohjelmia pääryhmätasolla. Lisäksi esitellään materiaalivirran tärkeyttä yrityksen arvontuoton kannalta. Tärkeimpänä tuloksena saatiin ensimmäinen versio ohjelmistosta ja se siirrettiin tuotantokäyttöön. Keskeisenä havaintona oli kommunikoinnin tärkeys kaikkien sidosryhmien kanssa. Lisäksi havaittiin, että tablettia oli helppo käyttää ohjelman kanssa.

2 TEKNOLOGIAT

2.1 BPNM

BPNM (Business Process Model and Notation) on vuonna 2005 julkaistu bisnesprosessin mallintamisen standardi. Se auttaa prosessin mallintamisessa olemalla yhteinen näkymä prosessin kehittäjän, teknisen toteuttajan ja loppukäyttäjien vastuhenkilön välillä. Tämä vähentää merkittävästi kommunikaation puutteesta syntyviä ongelmia. BPMN-mallit koostuvat yksinkertaisista diagrammeista (**Kuva 1.**), jotka on rakennettu rajoitetuista elementeistä. /1/



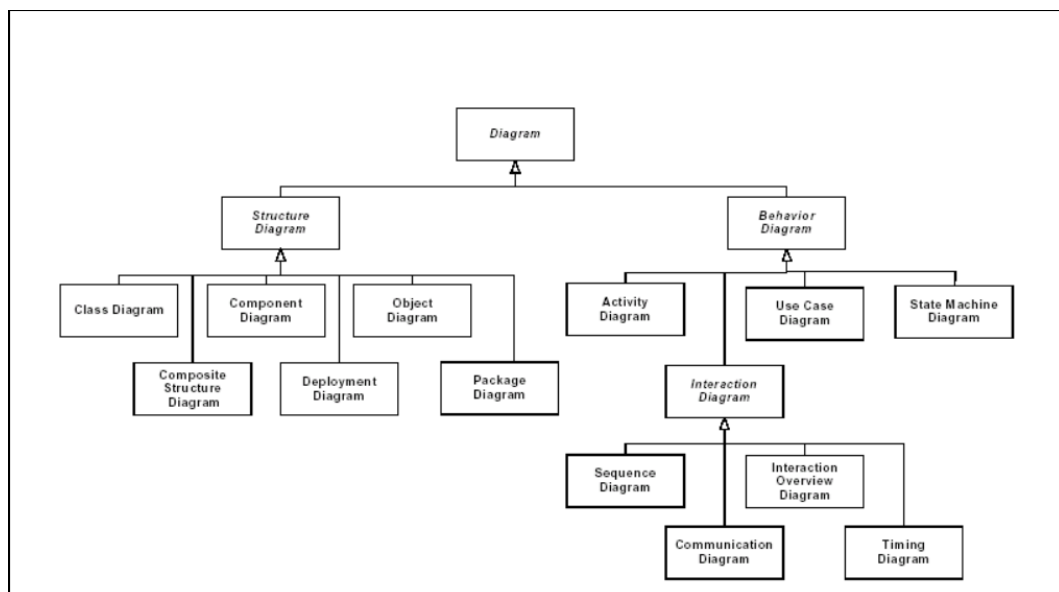
Kuva 1. Esimerkki BPNM-prosessikuvauksesta.

2.2 UML

UML (Unified Modeling Language) on vuonna 1997 standardoitu graafinen mallinnuskieli ja se sisältää 13 erilaista kaaviota. Niistä kuudella kuvataan rakennetta, kolmella käyttäytymistä ja neljällä vuorovaikutusta. Alun perin UML kehitettiin järjestelmä- ja ohjelmistokehitystä varten yhdistämään kolmea johtavaa oliomallinnustekniikkaa (**Kuva 2.**).

Käyttötarkoituksia ovat muun muassa:

- ohjelmistosuunnittelu
- ajettavan koodin luominen
- järjestelmien arkkitehtuurit
- prosessitekniikka
- internet-portaalien rakenteet
- työvirtojen kuvaus
- liiketoimintaprosessien mallintaminen.



Kuva 2. UML:n kaaviohierarkia. /2/

2.3 SCRUM

Scrum on ketterän ohjelmistokehityksen projektinhallinnan viitekehys. Scrum on tarkoitettu alun perin ohjelmistoprojekteihin, mutta se sopii myös yleiseen projektinhallintaan. Ideana on, että yksi ainoa monitaitoinen ryhmä suorittaa koko kehi-

tysprosessin vahvasti lomittuneella vaiheistuksella. Tämän kaltaisen ryhmän toimintaa voidaan verrata rugby-joukkueeseen, jossa ryhmä muodostaa tiiviin yksikön ja pyrkii etenemään ja toimimaan tiiviissä yhteistyössä. Termi Scrum onkin rugby aloitusryhmitys. Eri tilanteisiin sopeutuvuus, nopeus ja itseohjautuvuus ovat tunnusomaisia niin Scrum-tiimille kuin rugby-joukkueellekin.

2.3.1 Työskentelytapa

Scrumissa työskennellään iteratiivisesti eli toistuvasti näin optimoiden ennustettavuutta ja kontrolloiden riskejä. Tuotteeseen lisätään ominaisuuksia vähän kerrallaan, kehittyen näin kohti valmiimpaa ja täydellisempää. Kehitysjaksoja on useita, niitä kutsutaan sprinteiksi, jotka ovat 1-4 viikon mittaisia. Sprintin aikana pyritään tuottamaan ”valmis” tuote, joka on käyttökelpoinen ja mahdollisesti julkaisukelpoinen. Sprintin sisältö suunnitellaan erillisessä suunnittelupalaverissa ennen aloitusta. Sprinttiin valitaan sellaisia tuotteen kehitysjonon kohtia, jotka ovat sillä hetkellä tärkeimpiä projektin kannalta. Sprintin päätteeksi pidetään katselmus, jossa esitellään konkreettiset saavutukset tuoteomistajalle (Product owner) sekä mahdollisesti muille sidosryhmien edustajille. Tällä tavalla saadaan palautetta ja lisätään ymmärrystä projektin etenemisestä. Ennen seuraavan sprintin aloitusta pidetään retrospektiivi, eli katsaus menneeseen, sprinttiin prosessin näkökulmasta. Tällöin tarkastellaan mikä meni hyvin ja mitä voitaisiin parantaa seuraavassa sprintissä.

2.3.2 Scrumin roolit

Kehityksessä käytetään yhtä tai useampaa scrumtiimiä, jotka koostuvat tuoteomistajasta, scrummasterista ja kehitystiimistä (**Kuva 3**). Tiimi päättää sprintin tavoitteet ja tehtävät, sekä vastaa yhdessä siitä, että näihin tavoitteisiin päästään. Scrumtimeilla on valta päättää omista työmenetelmistään, ne ovat siis itseohjautuvia. Tämän mahdollistaa jäsenten monitaitoisuus.

Product owner, eli tuoteomistaja, vastaa kehitystiimin työn sekä tuotteen arvon täysimääräisestä hyödyntämisestä. Hän määrittelee tuotteen vaatimukset sekä järjestää tuotteen kehitysjonon yhdessä muun tiimin kanssa. Tuoteomistaja on vain yksi henkilö, ei esimerkiksi komitea, mutta tuoteomistaja voi hyödyntää sellaisia

tai edustaa komitean toiveita. Tämä tapahtuu tuotteen kehitysjonon kautta, mutta tuoteomistajan pitää olla itse vakuuttunut muutoksen tarpeellisuudesta. Tuoteomistaja on tuotteen liiketoiminnan asiantuntija ja hän edustaa asiakkaan lisäksi käyttäjiä. Hänen tehtävänä on maksimoida tuotekehitykseen sijoitetun pääoman tuotto ja varmistua siitä, että scrumtiimi toteuttaa jokaisessa sprintissä olennaisimpia asioita. Product owner antaa siunauksen sprittikatselmuksessa edellisen sprintin aikaansaannokselle eli tuoteversiolle ja osallistuu sprintin suunnittelupalaveriin. Näin hän varmistaa, että kehitystiimi ymmärtää tuotteen kehitysjonon tehtävät tarvittavalla tarkkuudella. Jos vaatimuksen ymmärtämisessä on vaikeuksia, on product ownerin tehtävä auttaa kehitystiimiä ymmärtämään niitä, ja tarvittaessa tarkentaa niiden toteutustapaa sprintin aikana.

Development team, eli kehitystiimi, koostuu ammattilaisista, jotka muodostavat tuotteen kehitysjonon tehtävistä tuoteversion jokaisen sprintin aikana. Kehitystiimin jäsenet ovat ainoita, jotka osallistuvat tuoteversion kehitykseen. Kehitystiimillä on monia taitoja, joihin sisältyy kaikki tarvittava osaaminen tuoteversion kehittämiseksi. Jäsenillä voi olla erityisosaamista tai erilaisia painopisteitä työssä, mutta vastuu on kuitenkin yhdessä koko kehitystiimillä.

Scrummaster vastaa siitä, että kaikki käyttävät scrumia ja myös ymmärtävät sitä. Tämä tapahtuu varmistamalla, että kaikki scrumtiimit pitäytyvät Scrumin teoriassa, säännöissä ja käytännöissä. Scrummaster on palveleva johtaja, joka perinteisestä esimiehestä poiketen, ei omaa jäseniin suoraa määräysvaltaa, vaan vaikutus tapahtuu Scrum-prosessin kautta. Scrummasterin tehtävä on pääasiassa ryhmän työskentelyn helpottamista, poistamalla esteitä ja ryhmän valmentamista itseohjautuvuuteen. Hän vastaa siitä, että tiimi työskentelee päivittäin tuottavasti noudattaen Scrumin pelisääntöjä. Samalla hän tarkkailee, kuinka työ etenee ja jos sprintin tavoitteet näyttävät epärealistiselta, hän kommunikoi kehitystiimin ja tuoteomistajan kanssa tilanteen korjaamiseksi, esimerkiksi sisältöä kaventamalla. Scrummaster varmistaa myös, että tiimillä on työrauha, eikä ulkopuolelta tule uusia sprinttiin kuulumattomia vaatimuksia.



Kuva 3. Scrumin roolit.

2.3.3 Scrumin tuotokset

Tuotoksilla kuvataan lisäarvoa tai työtä tavoilla, jotka pyrkivät lisäämään läpinäkyvyyttä ja tilaisuuksia sopeuttamiselle ja tarkastelulle. Scrumissa tuotokset keskittyvät maksimoimaan keskeisten tietojen läpinäkyvyyden. Näin scrumitiimi pystyy todennäköisemmin toimittamaan tuoteversion, joka on osaltaan valmis.

- **Tuotteen kehitysjo** (Product Backlog) on lista, johon on järjestetty kaikki mitä tuotteessa saatetaan tarvita. Se on ainoa lähde vaatimuksille ja muutoksille. Tuoteomistaja, eli product owner, vastaa tuotteen kehitysjonosta. Vastuuseen kuuluu niin sen sisältö kuin saatavuus ja järjestäminen.

Kehitysjonon muuttuu koko ajan, kehittyen tuotteen ja sen ympäristön mukana.

- **Julkaisusuunnitelma** on kehitysjonon, johon on valittu tuotteen seuraavan julkaisun kohdat (product backlog item) sekä arvion sprinttien määrästä, joita vaaditaan julkaisun toteuttamiseen.
- **Sprintin tehtävälista** (Sprint Backlog) on suunnittelupalaverissa valitut sprintissä toteuttavat kohdat ja siinä on myös suunnitelma niiden toteuttamiseksi. Kehitystiimi antaa tehtävälistan muodossa arvion siitä, mitä toiminnollisuuksia seuraavassa kehitysversiossa on sisällytetty ja mitä tehtäviä niiden toteuttaminen vaatii. Tehtävälistassa pyritään määrittämään ja tekemään näkyväksi kaikki työ, mitä kehitystiimi kokee tarpeelliseksi, että sprintin tavoitteeseen päästään ja saadaan tuoteversio valmiiksi. Listan sisältöä ei edes tuoteomistaja pysty vaihtamaan. Toisaalta, kehitystiimi voi halutessaan milloin tahansa muuttaa tai poistaa tehtävälialta tehtäviä. Tällä tavalla tiimi varmistaa, että tavoitteet ja asetetut kehitysjonon kohdat toteutuvat sprintissä.
- **Edistymiskäyrä** (Burndown Chart) kuvastaa jäljellä olevaa työmäärää ajan funktiona, joko tuotteen tai sprintin osalta.
- **Tuoteversio** (Increment) koostuu kaikista kehitysjonon kohdista. Tähän lasketaan mukaan kaikki aiemmissa sprinteissä tehdyt ja meneillään olevassa sprintissä valmistuneet kohdat. Tuoteversion tulee olla valmis sprintin lopussa. Se täyttää scrumitiimin valmiin tuotteen määritelmän ja on käyttökelpoinen versio, vaikkei tuoteomistaja sitä julkaisisikaan.

2.3.4 Scrumin tapahtumat

Scrumissa on tärkeää, että toiminta on säännöllistä. Niinpä siinä käytetään ennalta sovittuja tapahtumia, joiden lisäksi tulisi olla mahdollisimman vähän muita kuin

Scrum-palavereita. Tapahtumat ovat tiukasti aikarajattuja, niillä on siis maksimipituus. Tällä varmistetaan se, että myös suunnittelulle on varattu riittävästi aikaa, kuitenkin niin, ettei suunnitteluprosessissa synny hukkaa.

- **Sprintti (Sprint)** on koko Scrumin ydinasia. Se on korkeintaan kuukauden mittainen, jonka sisällä tuotetaan valmis tuote, joka täyttää määritelmän olemalla samalla käyttökelpoinen ja julkaisukelpoinen versio tuotteesta. Sprintin pituus on sama koko tuotekehitysprosessin ajan. Uusi alkaa välittömästi edellisen jälkeen.
- **Sprintin suunnittelupalaveri** (Sprint Planning Meeting) on palaveri, jossa suunnitellaan työ, joka sprintin aikana suoritetaan. Suunnitelmaa on tekevässä koko scrumtiimi. Kuukauden mittaisella sprintillä suunnittelupalaveri voi olla enintään kahdeksan tunnin mittainen. Vastaavasti esimerkiksi kahden viikon sprintille varataan enintään neljä tuntia.
- **Päiväpalaveri** (Daily Scrum) on lyhyt, enintään 15 minuutin mittainen palaveri, jossa kehitystiimi suunnittelee seuraavan 24 tunnin työnsä. Käytännössä palaverissa katsotaan, mitä edellisen palaverin jälkeen on saatu tehtyä ja mitä voidaan tehdä ennen seuraavaa palaveria. Palaverikäytäntö halutaan pitää mahdollisimman yksinkertaisena, sen takia päiväpalaveri on aina samassa paikassa. Palaverissa jokainen kehitystiimin jäsen kertoo
 - mitä on tehnyt edellisen päiväpalaverin jälkeen
 - mitä aikoo tehdä ennen seuraavaa päiväpalaveria
 - mitä esteitä työn etenemiselle mahdollisesti on.

Päiväpalaveri on kehitystiimin oma tapaaminen, jolla optimoidaan kehitystyön arvo ja nostetaan sprintin tavoitteiden onnistumisen todennäköisyyttä. Jos aika uhkaa venyä yli 15 minuutin mittaiseksi, on mielekästä sopia erillinen palaveri, vaikka se alkaisikin heti päiväpalaverin jälkeen.

- **Tuotteen kehitysjonon työstö** (Product Backlog Grooming) on järjestyksen lisäämistä yksityiskohdille, työmääräarvioille ja kehitysjonon kohdille. Tämä on toistuva prosessi, jossa tuoteomistaja ja kehitystiimi lisäävät yksityiskohtia tuotteen kehitysjonoon yhdessä. Kehitysjonon kohtia katselmoidaan ja arvioidaan kehitysjonon työstön aikana. Kuitenkin tuotteen omistajalla on oikeus muulloinkin päivittää kehitysjonoa. Kapasiteetin käyttö on enintään 10 prosenttia koko kehitystiimin kapasiteetista. Työmääräarvio tulee aina kehitystiimiltä. Jos tuotteen omistaja haluaa vaikuttaa työmääräarvioon, hän voi auttaa kehitystiimiä ymmärtämään paremmin vaatimuksia.
- **Sprinttikatselmus** (Sprint Review) pidetään sprintin lopussa ja on muodoltaan epämuodollinen. Siinä tarkastellaan kehitetty tuoteversio ja muutetaan tarvittaessa kehitysjonoa. Katselmuksessa scrumitiimi ja sidosryhmät selvittävät sprintin aikana kehitetyt asiat. Tähän perustuen, lisättyinä mahdollisilla sprintin aikana tehtyihin kehitysjonon muutoksiin, muodostetaan käsitys siitä, mitä kehitetään seuraavaksi. Kehitystiimi esittelee tuotedemoa tavoitteenaan saada palautetta. Samalla edistetään keskustelua ja luodaan pohjaa seuraavalle sprintin suunnittelupalaverille. Ajallisesti katselmukseen varataan aikaa sprintin keston mukaan, neljä tuntia kuukauden sprintille ja lyhyemmille sprinteille sitten suhteessa vähemmän aikaa.

Sprintin retrospektiivi (Sprint Retospective) on scrumitiimin tilausuus tarkastella työskentelyään ja mahdollisuus tehdä suunnitelma parannuksille, jotka toteutetaan seuraavassa sprintissä. Retrospektiivi pidetään katselmuksen jälkeen ennen seuraavaa suunnittelupalaveria. Ajallisesti se on hieman lyhyempi kuin sprinttikatselmus, kuukauden sprintille varataan kolme tuntia. /3-5/

2.4 Redmine

Redmine on joustava web-pohjainen projektinhallintatyökalu. Se on kehitetty Ruby on Rails kehykseen ja on alustariippumaton, kuten myös tietokantariippumaton. Redmine on avoimen lähdekoodin ohjelmisto ja se on julkaistu GNU General Public License v2 alla.

2.4.1 Tärkeimmät ominaisuudet

Usean projektin tuki mahdollistaa sen, että kaikkia projekteja voidaan hallita yhden Redmine-asennuksen kautta. Näissä projekteissa voi jokaisella käyttäjällä olla eri rooli eri projekteissa. Projektin voi määritellä julkiseksi, jossa kaikki näkevät sen tai sitten yksityiseksi, jossa projekti näkyy vain projektin jäsenille. Redmine sisältää eri moduuleita, kuten wiki ja asia seuranta (issue tracking ja tiedon säilytys). Nämä moduulit voidaan aktivoida projektikohtaisesta. Projekti voi myös jakaantua eri alaprojekteihin. Käyttöoikeuksien alla löytyvät roolit, joita voi olla useita. Käyttöoikeuksien hallinta on helppoa käyttöliittymällä.

Seurantasysteemi on joustava ja perustuu asia (issue) perusteiseen systeemiin. Tähän voidaan olemassa olevien statuksien lisäksi määritellä omia statuksia ja asiatyyppejä. Työnkulun siirtymät voidaan asettaa joka asiatyypille ja joka roolille web-pohjaisen käyttöliittymän avulla. Asian voi luoda myös sähköpostitse.

Ohjelmassa luodaan automaattisesti Gantt-kaavio ja kalenteri, joka perustuu asian aloitus- ja tavoitepäivään. Uutiset voidaan lähettää helposti. Helppoa on myös dokumenttien ja tiedostojen jako. Syötteen ja sähköposti ilmoitukset saadaan Atom-tyyppisenä. Ohjelmassa on projektikohtainen wiki ja foorumi, joka noudattaa Wikipediasta tuttua syntaksia ja muokkausta.

Aikaseuranta tapahtuu aikakirjauksilla, jotka voidaan kirjata joko projekti- tai tikititasolle. Mukana on myös yksinkertainen raportointinäköymä, jolla voidaan muodostaa raportteja joko käyttäjittäin, asioittain, kategorioittain tai aktiivisuuden mukaan. Ohjelmassa on myös muokattavat kentät asioille, aikamääritteille, projekteille ja käyttäjille. Käytettävänä on useita kenttämuotoja ja muokattavat kentät näkyvät kuten normaalit kentät.

Tuki ja integraatio on kattava, ohjelmassa on SMC-integraatio (SVN, CVS, Git, Mercurial, Bazaar ja Darcs). Lisäksi usean LDAP-(Lightweight Directory Access Protocol) autentikoinnin tuki. Tilien luonti onnistuu heti, kun käyttäjä löytyy hakemistosta. Käyttäjän omatoimisen rekisteröinnin tuki, joko automaattisesti (ilman vahvistusta), manuaalisesti (pääkäyttäjän toimesta) tai automaattisesti generoidun

URL:n kautta. Usean kielen tuki, saatavana kieliä Bosniasta Vietnamiin, ja myös Suomesta. Usean tietokannan tuki. Toimii MySQL-, PostgreSQL- ja SQLite- tietokannoilla. /6/

2.5 MES

Manufacturing execution systems (MES) ovat tietokoneella käytettäviä järjestelmiä valmistuksessa. Niillä seurataan ja dokumentoidaan raakamateriaalin jalostamista lopputuotteeksi. MES tarjoaa tietoa tuotannon johdolle siitä, kuinka nykyisistä tuotanto-olosuhteista saadaan paras mahdollinen lopputulos. MES toimii reaali-ajassa ja näin mahdollistaa usean tuotantoprosessin osa-alueen yhtäaikaisen hallinnan, kuten syötteen, henkilöstö, koneet ja tukipalvelut.

MES voi toimia usean eri toiminnon alueella. Tällaisia ovat tuotteen hallinnan elinkaaren näkökulmasta resurssien aikataulutus, tilausten vastaanotto ja lähetys, tuotantoanalyysi ja käyttökatkosten hallinta yleisen tehokkuuden kannalta, laatu tai materiaalin seuranta. MES seuraa niitä asioita, mihin se on määritelty tallentamalla dataa, prosesseja ja tuotoksia. Seuranta voi olla erityisen tärkeää säädellyillä aloilla, kuten ruuan, alkoholin tai lääkkeiden valmistus. Näillä aloilla dokumentointi ja todisteet prosesseista, tapahtumista ja teoista ovat pakollisia.

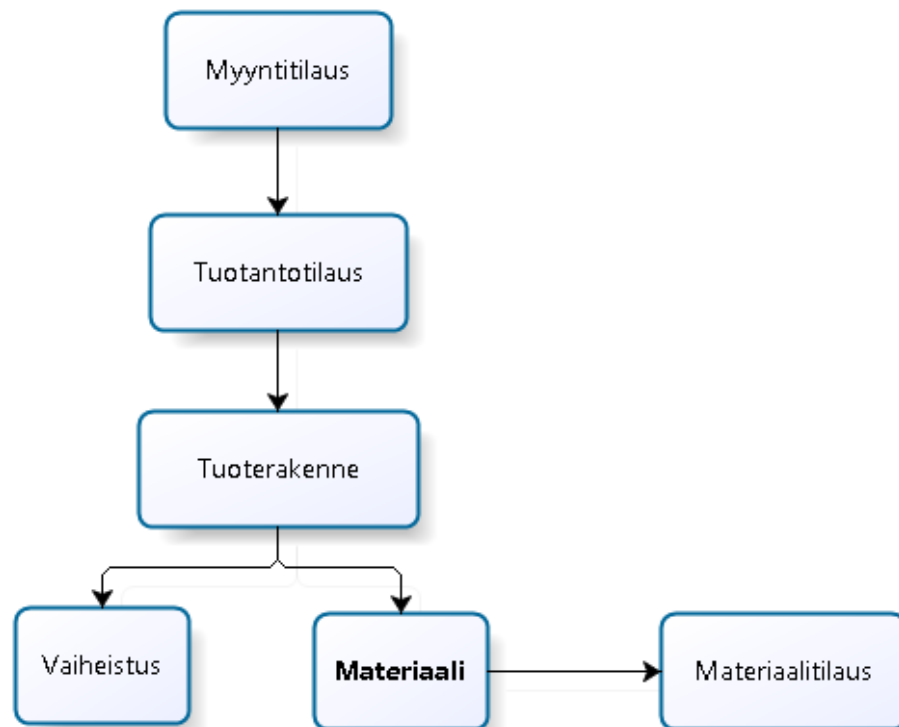
Rajanveto MES:n ja muiden järjestelmien, kuten ERP (Enterprise Resource Planning) ja SCADA (Supervisory Control and Data Acquisition) välillä on hankalaa. Monesti MES mielletäänkin vain välivaiheeksi. Voidaan todeta, että ERP sisältää MES-järjestelmän. /7/

3 TUOTANNONOHJAUS YRITYKSESSÄ

Yrityksen perustamisvaiheessa voidaan vielä tulla toimeen ilman tietokoneella olevaa MES/ERP-ohjelmistoa, mutta tuotantoa tulee voida ohjata heti alusta asti. Toiteuttamistapa voi lähteä yksinkertaisesta ruutupaperi- tai Excel kirjanpidosta. Tässä vaiheessa monesti keskitytään yrityksen kannattavuuden saamiseen plussan puolelle sekä kasvuun. Kun nämä asiat on saavutettu, otetaan seuraava askel eteenpäin.

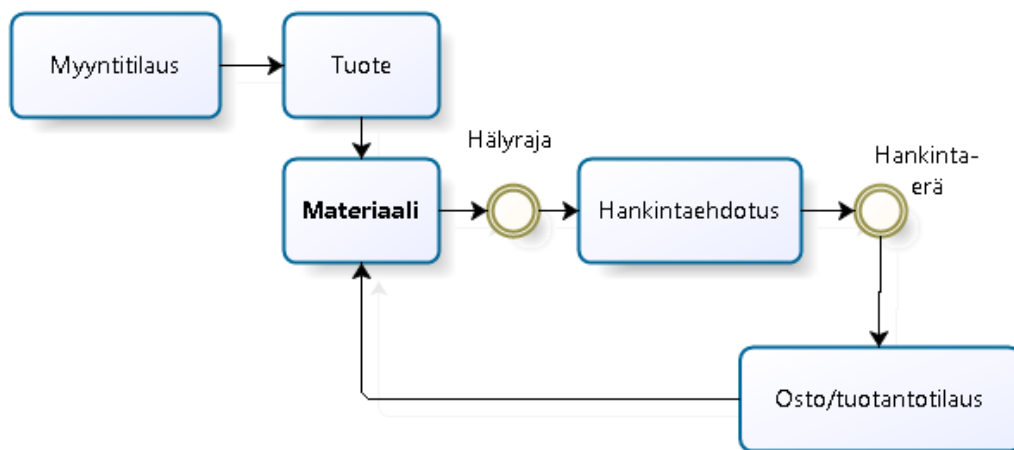
Materiaalinhallinta voidaan nähdä koko tuotannon tärkeimpänä osana. Väite on rohkea, mutta oikea materiaali väärään aikaan on lähes yhtä vakava asia kuin väärä materiaali oikeaan aikaan. Materiaalinhallinnalla pyritään siihen, että tarvittavalla hetkellä on saatavana sekä oikeaa materiaalia, että riittävä määrä materiaalia. Määrän määrittelyyn apuna voidaan käyttää erilaisia tuotannonohjaustapoja ja/tai myynnin ennusteita.

Tuotetta voidaan ohjata eri tavalla ja tämä ohjaus vaikuttaa materiaalin tarpeeseen. Tässä tapauksessa tuotteen ohjaamisella tarkoitetaan asioita, joita tapahtuu kun asiakas tekee tilauksen. Kun kyseessä on tilausohjaus (**Kuva 4.**), muodostuu tuotteesta suoraan tuotantotilaus. Tuotantotilaus taas syntyy tuoterakenteesta ja tuoterakenteella on tuotteen resepti eli osaluettelo ja vaiheistus. Rakenne voi olla monimutkainen, sisältäen alarakenteita tai hyvinkin yksinkertainen. Tätä mallia käytetään yleensä pienissä sarjoissa tai yksittäisissä kappaleissa. Tällöin ei ole voitu tietää menekkiä etukäteen, eikä materiaalia ole välttämättä varastossa. Toimitusaikakin on lähtökohtaisesti pidempi.



Kuva 4. Materiaali tilausohjautuvassa tuotannossa.

Varasto-ohjauksessa tuotetta tai sen komponentteja on jo valmiiksi varastossa (**Kuva 5.**). Tällöin asiakkaan tilaus muuttaa varastosaldoa. Kun saldo menee asetettujen rajojen ulkopuolelle, muodostuu materiaalille joko osto tai valmistustarve. Hankittava erä on riittävän suuri, ei pelkästään täyttääkseen tarpeen vaan mennäkseen juuri sopivasti yli, jottei aivan heti tarvitse tehdä uutta tilausta. Tämän sopivan määrän löytäminen on tässä tavassa haastavinta.



Kuva 5. Materiaali varasto-ohjautuvassa tuotannossa

Osto-ohjaus on tilausohjauksen kaltainen tapa. Se on käytössä pääasiassa tuotteissa, jotka ostetaan suoraan toiselta toimittajalta eli on ns. läpimenotuote.

Kaksoislaatikko-ohjauksessa tai toisin sanoen visuaali-ohjauksessa, ei ole saldoseurantaa MES:n näkökulmasta. Tässä menetelmässä käytetään kahta laatikkoa. Kun toinen on tyhjä, otetaan toinen käyttöön ja aloitetaan tyhjän täyttötoimenpiteet.

Osto-osille tai puolivalmisteille voidaan asettaa hälyraja ja minimi tuotanto/ostomäärä. Kun asetettu hälytysraja eli kriittinen saldo alittuu MES:n näkökulmasta, aloitetaan tuotanto- tai ostoprosessi määrällä, joka on asetettu pienimmäksi määräksi, mitä aloitetaan tekemään tai ostetaan. Näiden lisäksi voidaan asettaa vielä erä koko eli minkälaisissa erissä voidaan tuottaa tai ostaa. Tehtaalla voi olla toimittajaan tai asiakkaaseen päin käytössä kotiinkutsu, missä asiakas tai toimittaja varastoi tuotetta ja saa sen käyttöönsä nopeasti ilman valmistusprosessia. Tästä tyyppillisesti veloitetaan lisämaksua.

Yleensä varastoa pyritään pitämään mahdollisimman pienenä erilaisilla menetelmillä. Toisaalta varsinkin Vaasan seudulla toimivat isot toimijat vaativat lyhyitä toimitusaikoja. Tämä johtaa monesti siihen, että hätätilanteessa tuotetta ei olekaan olemassa, vaan joudutaan tekemään ylitöitä tai siirtämään muita töitä tuotteen valmistamiseksi. Tämä taas on pois kannattavuudesta. Tästä johtuen jotkut yritykset

pyrkivät varastoimaan asiakkaiden tuotteita pitääkseen toimitusvarmuuden ja ennakoitavuuden korkealla tasolla.

4 MES-MATERIAALINHALLINNAN MÄÄRITTELY

Yrityksessä käytettävän ERP-järjestelmän käyttöaste oli käyttöönottovuodesta 2008 laskenut vähän kerrallaan. Tähän liittyen, aloitettiin palaverit tuotannon työntekijöiden kanssa. Palaverit sisälsivät koulutusta kirjaamiskäytännöistä sekä vapaa-
muotoista keskustelua. Keskusteluissa kävi ilmi, että yksi merkittävä haaste tuotannossa on materiaalin löytäminen.

Ongelma koski koko ketjua. Tilauksen saapuessa, myynnin henkilö kysyy materiaalin saatavuutta työnjohdolta. Työnjohto ei voinut olla varma, onko ERP-järjestelmässä oleva materiaalisaldo oikea, vaan joutui tarkistamaan asian monesti manuaalisesti. Tämä manuaalinen tarkistus siirtyi usein koneenkäyttäjälle. Vastauksesta riippuen, työnjohto joko varmisti toimitusajan tai tilasi materiaalia.

Ongelma oli samankaltainen myös tarjouspyyntövaiheessa, sillä nykypäivänä toimitusaika on avainasemassa kaupan onnistumiselle. Tilanne saattoi myös olla niin, että erikoisemman materiaalin olemassaoloa ei tiedetty, vaan sitä tilattiin varmuuden vuoksi.

Työstöohjelman suunnittelija voi luoda ohjelman väärälle materiaalille, jos saldot eivät ole tiedossa. Tämän väärän materiaalin joutui koneen koneenkäyttäjä sitten vaihtamaan oikeaan. Tässä meni työaika hukkaan ja virheiden mahdollisuus kasvoi.

Ensimmäisessä vaiheessa päädyttiin aiemmin testattuun menetelmään, jossa hyllyt järjesteltiin uudestaan ja hyllyt myös siivottiin. Menetelmän ydin oli kuitenkin yhteinen sopiminen materiaalien paikoista. Tämä menetelmä osoittautui toimivaksi jonkin aikaa, mutta jo muutamassa kuukaudessa oli tilanne lähes yhtä huono kuin ennen muutosta.

Tilannetta uudestaan pohdittaessa, otettiin esille aiemmin käytössä ollut Excel-pohja, johon haluttiin lisäominaisuuksia. Lisäominaisuudet sisälsivät muun muassa nimikkeen ja varaston valinta vetovalikosta. Näiden lisäksi kaivattiin saldon ylläpitoa sekä myös jonkinlaista mobiilikäyttöä. Viimeistään tässä vaiheessa oli selvää,

että pelkkä Excel ei tästä tehtävästä enää selviä, vaan tarvitaan uusi ohjelma (**Kuva 6.**).

HYLLY A			
PUOMI	MATERIAALI		kg max
6	ML05N15125250	1.5*1250*2000 ZN DX 51D+Z275MBC	
5	ML051J15125200BOBI	1.5*1250*2000 ZN DX 51D+Z100MBU	
4	ML051J15125163BOBI	1.0*1250*1638 ZN DX 51D+Z100MBU	
3	ML051J15125163BOBI	1.0*1250*1638 ZN DX 51D+Z100MBU	
2	ML655J15125200	1.5*1250*2500 ZN DX52D+Z275-M-C, RR44	
1	ML655J15125200	1.5*1250*2500 ZN DX52D+Z275-M-C, RR44	
0	ML35N15125250BOBI	1.0*1250*2000 RST KYVA SSCR 1.4301 2B	
HYLLY B			
PUOMI	MATERIAALI		kg max
5	ML851J10125158	Bob1 1*1250*1580 galvannealed ZF100RBO	
4	ML851J10125158	Bob1 1*1250*1580 galvannealed ZF100RBO	
3	ML851J10125158	Bob1 1*1250*1580 galvannealed ZF100RBO	
2	ML851J10125158	Bob1 1*1250*1580 galvannealed ZF100RBO	
1	ML851J10125158	Bob1 1*1250*1580 galvannealed ZF100RBO	
0	ML851J10100200	Bob1 1*1000*2000 galvannealed ZF100RBO	
HYLLY C			
PUOMI	MATERIAALI		kg max
6	ML055N12125250	1.25*1250*2500 ZN DX51D+Z275MBC	
5	ML055N12125250	1.25*1250*2500 ZN DX51D+Z275MBC	
4	ML055N15125250	1.5*1250*2500 ZN DX51D+Z275MBC	
3	ML055N15125250	1.5*1250*2500 ZN DX51D+Z275MBC	
2	ML055N20125250	2*1250*2500 ZN DX51D+Z275MBC	
1	ML055N20125250	2*1250*2500 ZN DX51D+Z275MBC	
0	Ravisteltavat komponentit		
HYLLY D			
PUOMI	MATERIAALI		kg max
6	ML055N07125250	0.7*1250*2500 ZN DX51D+Z275MBC	
5	ML055N07125250	0.7*1250*2500 ZN DX51D+Z275MBC	
4	ML055N07125250	0.7*1250*2500 ZN DX51D+Z275MBC	
3	ML055N10125250	1.0*1250*2500 ZN DX51D+Z275MBC	
2	ML055N10125250	1.0*1250*2500 ZN DX51D+Z275MBC	
1	ML055N10125250	1.0*1250*2500 ZN DX51D+Z275MBC	
0	Ravisteltavat komponentit		

Kuva 6. Aiemmin käytössä ollut excel.

Koska ohjelmistoprojektit monesti epäonnistuvat prosessin sisäänajovaiheessa, päätettiin ottaa henkilöstön mukaan heti alkuvaiheessa. Tämän päätöksen kautta saatiin arvokasta tietoa ohjelman hankintaa tai mahdollista tekoa ajatellen. Lisäksi henkilöstö sitoutuu projektiin parhaalla mahdollisella tavalla, koska se oli itse saaneet siihen vaikuttaa.

4.1 Haastattelut

Haastatteluissa kysyttiin seuraavia asioita:

- Mitä haasteita nykyisessä levyhyllyn käytössä on?
- Miten nämä haasteet vaikuttavat työhön, tuottavuuteen ja toimitusaikaan?
- Mitä ohjelmassa tulisi olla?
- Miten näet tilanteen vuoden päästä siitä, kun ohjelma on otettu käyttöön?

- Miten pidetään huoli siitä, että kaikki käyttävät ohjelmaa?
- Mikä olisi hyvä sijoituspaikka mobiililaitelle?
- Miten haluat osallistua projektiin?

Vastaajiksi valittiin kaksi levyhyllyä pääasiassa käyttävää koneenkäyttäjää, ohjelmoijan, varastomiehen ja harvemmin levyhyllyä käyttävän henkilön. Haastateltavista valittiin pääkäyttäjä, joka sitoutettiin projektiin.

Kysymyksiin haasteista ja vaikutuksista työhön vastattiin todella hyvin. Tässä vaiheessa mainitusta siivouksesta oli kulunut vuosi ja todettiin, että siitä oli ollut jonkin verran hyötyä. Todettiin kuitenkin, että uudelleenjärjestely oli toteutettu huonosti ja esimerkiksi alumiini oli kaukana. Nykyinen järjestelmä perustui haastateltavien mukaan siihen, että kokeneet käyttäjät muistavat, missä materiaali sijaitsee.

Haastateltavat kuvailivat ongelmatilanteita, joissa esimerkiksi ERP-järjestelmän mukaan materiaalia ei ole ja sitä on siitä syystä tilattu lisää. Toinen ongelma on materiaalitilauksen ylimitoitus projektille. Mainittiin myös, että nykyinen levyhyllyn nimeämisjärjestys on ihan pielessä, tämä pitäisi tehdä uudestaan. Suunnittelija joutui välillä menemään itse tarkistamaan materiaalin saldon. Kun ohjelma suunniteltiin tarkoitettua isommalle koolle, muodostui hukka isommaksi.

Mainituista ongelmista johtuen, oltiin tilanteessa, jossa materiaali sijoitettiin sinne, minne se mahtui, jos mahtui mihinkään. Monesti materiaali saattoi myös olla sijoitettuna lattiatasolle tilanpuutteen takia. Tästä syystä aiemmin sovitusta järjestyksestä ei enää voitu pitää kiinni. Todettiin myös, että kaikki eivät palauta ottamaansa lavaa, vaan se jää jonkun toisen huolehdittavaksi.

Tässä yhteydessä saatiin myös monia käytännön ehdotuksia koskien uudelleensijoitusta, turvallisuutta ja järjestystä, sekä yleisesti prosessia. Näistä suurin osa toteutettiin erillisinä toimina.

Ohjelman sisältöön liittyvässä kysymyksessä tärkein toive oli helppokäyttöisyys, eli ohjelma tulisi olla helppokäyttöinen, jotta sen pystyisi myös kokemattomampi

käyttäjää omaksumaan nopeasti. Ohjelmassa tulisi myös olla hakutoiminto materiaalille ja erilaisia näkymiä sitä tukemaan. Materiaali ja levyhylly tulisi myös varata laitteen käyttäjälle (**Kuva 7.**). Materiaalin palautuksen yhteydessä tulisi myös olla mahdollista syöttää levymäärä, montako palautettiin (**Kuva 8.**). Inventointitoiminnon pitäisi senkin olla helppokäyttöinen.

ML25N20100200

A0 - 605 kg

Finnpower

Nouda varastosta valitulle työpisteelle

Kuva 7. Materiaalin työnalle otto koneelle. Ohjelmaan on syötetty materiaali ja valittu hyllypaikka sekä kone.

Finnpower	ML25N20100200	605.0	Admin - 20.11.2016 12:26:27
-----------	---------------	-------	-----------------------------

Määrä

605.00 kg

tai 112.03703703702 mm

tai 56.01851851851 kpl

Alkuperäinen varastopaikka

A0

Palauta varastoon

Kuva 8. Materiaalin palautus käytön jälkeen koneelta takaisin varastoon. Määrän voi syöttää eri suureina.

Tilanne vuoden päästä käyttöönotosta nähtiin varsin valoisana, kunhan kaikki ovat sitoutuneet asiaan. Työ helpottuu, kun kaikki osapuolet tietävät materiaalin sijainnin. Virheitä syntyy vähemmän ja inventointiaika voitaisiin sijoittaa tuottavaan työhön.

Kaikki saadaan käyttämään ohjelmaa, kunhan pidetään riittävä koulutus kaikille. Tämä työkalu myös mielletään tärkeäksi työkaluksi ja tähän halutaan panostaa. Tämän kaltaista ohjelmaa on odotettu 15 vuotta.

Hyvänä sijoituspaikkana pidettiin trukkia ja varastomiehen koppia. Käytännön testeissä tämä hieman muuttui.

4.2 Osta vai tee itse

Kun haastattelut saatiin päätökseen, muodostettiin niiden tuloksista ominaisuuslista, mitä ohjelman tulisi sisältää. Tätä täsmennettiin vielä tuotantopäällikön, työjohtajien ja tietohallintopäällikön kanssa.

Ominaisuuslista:

1. Helppokäyttöinen
 1. Sisäänluku viivakoodilla
 2. Ei mitään turhia ominaisuuksia
 3. Helppo sisäistää
2. Toimii myös tabletilla
3. Työnalle otto eli materiaalin ja hyllyn varaus
 1. Tätä varten piti olla myös koneet
4. Materiaalin palautus
5. Inventaari helpottuu
 1. Voidaan tehdä puomeittain
6. Erilaiset näkymät ja haut
7. Hinta-laatusuhde

Saatus listaa verrattiin jo olemassa oleviin tuotteisiin. Nopeasti kävi selväksi, että markkinoilla olevat tuotteet sisältävät liikaa ominaisuuksia, kuten liitäntä automaattihyllyyn tai tilauksien hallinta. Lähes kaikki toivotut ominaisuudet kyllä löytyivät, mutta ominaisuuksien paljous oli käyttäjien tärkeintä toivetta, eli helppokäyttöisyyttä vastaan. Pyydettiin kuitenkin muutaman tarjouspyynnön, hinnat olivat niin korkeita, että niihin ei saatu hankerahaa.

Oltiin siis tilanteessa, jossa kehitystarve oli ilmeinen, mutta valmiin ratkaisun hankinta olisi liian kallis, eikä lopputuloskaan vastaisi käyttäjän tahtotilaa täysin. Heräsi ajatus ohjelmiston itse teettämisestä valmiin ohjelman sijaan. Kontaktien kautta saatiin tieto hiljattain aloittaneesta yrityksestä, jolla olisi tarpeellinen osaaminen ja uuden yrityksen into palvella asiakasta. Tähän yritykseen otettiin yhteyttä ja he tulivat paikan päälle määrittelykäynnille, jotta saisivat mahdollisimman tarkan tarjouksen tehtyä.

4.3 Päätöksenteko

Määrittelyn tuloksena syntyi tarjous, joka sisälsi ohjelmaan halutut toiminnallisuudet. Hyvän etukäteisvalmistelun ansiosta toimittaja pystyi tekemään sitovan tarjouksen, josta kävi ilmi kaikki kustannukset. Tämän tarjouksen perusteella oli helppo tehdä investointilaskelmia. Vaihtoehtolaskelmina oli olemassa olevan ohjelmiston hankinta ja muokkaus tai nykyisellä tavalla jatkaminen. Oman ohjelman teko oli selvästi paras vaihtoehto, joten tähän päädyttiin.

4.4 Projektisuunnitelma

Yrityksen sisäinen projektisuunnitelma sisälsi pääkohdat projektin läpivientiin. Sen tavoite oli selkeä: saada materiaalinhallintaohjelmisto tuotantokäyttöön. Sille määriteltiin aikataulu, joka sisälsi alku- ja loppuajan. Vastuunjako merkittiin selvästi kunkin pääkohdan alle. Ennen vastuunjakoa määriteltiin osapuolien roolit, vastuut, resurssit sekä projektin toteutusorganisaation. Annetut resurssit olivat määritelty selkeästi ja pyrittiin siihen, ettei kenelläkään tule kohtuutonta kuormaa (**Kuva 9**). On normaalia, että projekti tehdään osana muita tehtäviä, niin myös tässä tapauksessa.

Asia	Vastuu	Tilanne	VK 1	VK 2	VK 3	VK 4	VK 5	VK 6	VK 7	VK 8	VK 9	VK 10	VK 11	VK 12	VK 13
Esikartoitus ja määrittely		Tehy													
Projekti suunnitella		Tehy													
Käyttäjien haastattelut		Tehy													
Avausneuvotteluiden valinta		Tehy													
Ohjelmointityön seuranta		Sprint meetingit 1 vko välein													
Ohjeen tekö sisäisesti		Lähes valmis													
Rakelankinnat		Tehy													
Koulutussuunnitelma															
Käyttökoulutus															
Hyötymerkinnot		Tehy													
Hyötyjen siivous															
Materiaalien oimäminen		Ei tarvita													
Materiaalien laskenta		Odotetaan koulutusta													
Lähtötietojen syöttö		Odottaa laskentaa													
Käyttöönotto		Odottaa lähtötietoja													
Sirto tuotantoon		Odottaa käyttöönottoa													
Testaus		Suoritetu testannut myös													
Langaton verkko		Tehy													

Kuva 9. Excel ohjelmalla tehty projekti aikataulu.

5 MATERIAALIVIRTA OSANA TILAUS-TOIMITUSPROSESSIA

Uuden ohjelmiston tavoitteena on materiaalivirran kehittäminen. Vaikka kyseessä on yksittäiseltä tuntuva asia, on se prosessin kannalta tärkeä. Tilaus.toimitusprosessi on yrityksen tärkein liiketointaprosessi ja, koska se tuottaa ulkoisille asiakkaille arvoa, on se sitä kautta ydinprosessi.

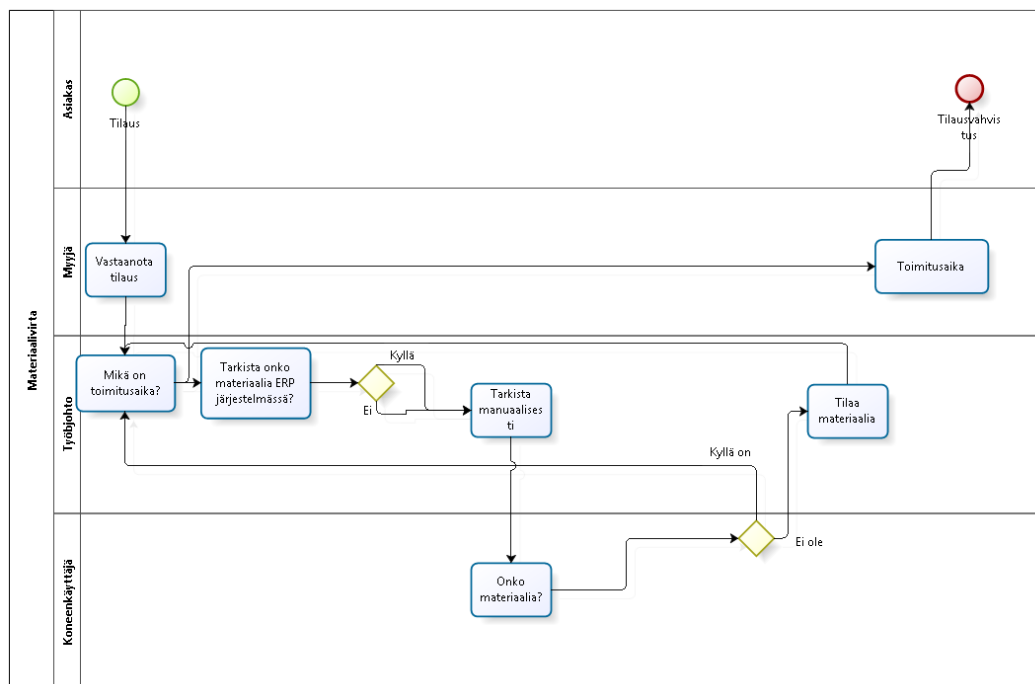
Tässä prosessissa on sekä sisäisiä että ulkoisia asiakkaita ja siinä prosessille tyypillisesti mennään myös organisaation rajojen yli. Sitä pystyy myös arvioimaan sekä sisäisen että ulkoisen asiakkaan näkökulmasta ja saadut hyödyt pystytään selkeästi todentamaan.

Koska uusi prosessi vaatii ohjelmiston käyttöä, tarvitaan tähän ohjelmiston käyttöön liittyvä tukiprosessi.

5.1 Entinen myyntiprosessi

Koska kyseessä on pääasiassa ohutlevystä tuotteita jalostava yritys, on myyntiprosessi vahvasti sidoksissa materiaalinhallintaan. Esiselvityksessä esille tullut prosessi piirrettiin, ja siitä kävi vahvasti ilmi, että varsinkin erikoisemman materiaalin ollessa kyseessä, jouduttiin saatavuus tarkastamaan manuaalisesti käytössä olevan ERP-järjestelmän puutteellisten tietojen takia. Tämä manuaalinen tarkistus oli pois tuottavasta työstä ja pahimmillaan asialla jouduttiin vaivaamaan useita henkilöitä.

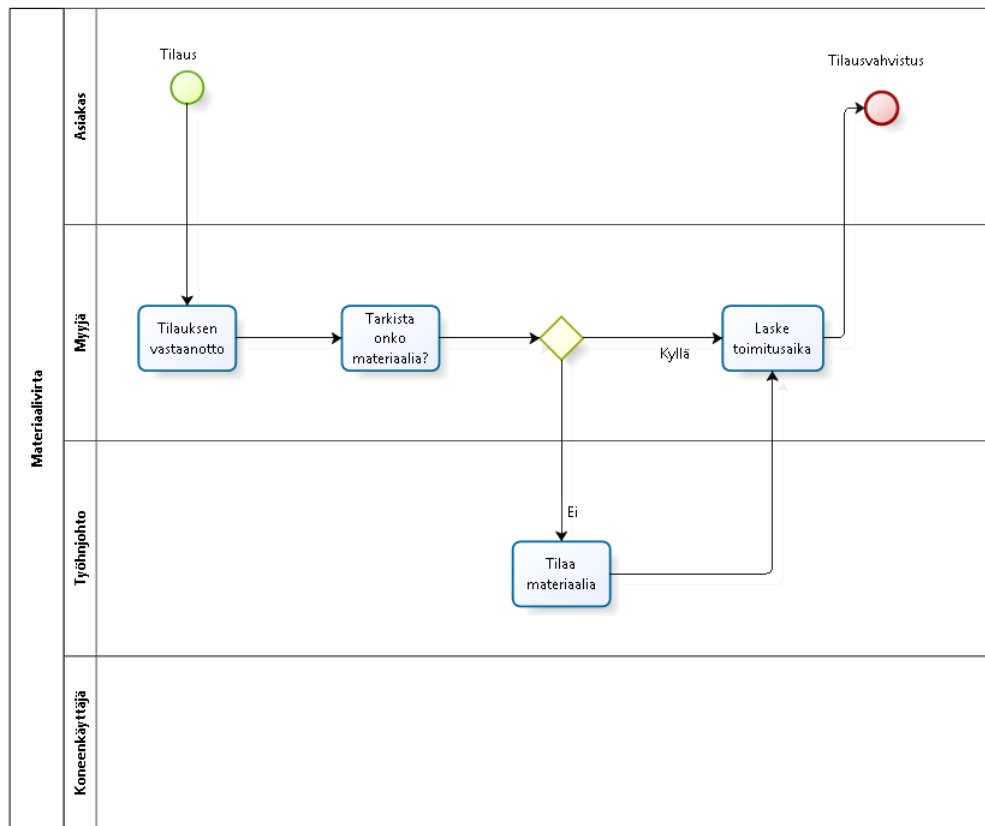
Useimmin käytettävät materiaalit ovat luonteeltaan sellaisia, että niitä on aina varastossa, joten niiden kohdalla tärkeämpää oli löytää ne. Tässä nojattiin aiemmin tehtyyn toimenpiteeseen, eli saman materiaalityypin sijoittamiseen samaan paikkaan ja käyttäjien muistiin. Useimmiten tämä toimikin, mutta varsinkin loma-aikoina, kun vakituinen henkilökunta oli poissa, materiaali etsittiin samaan tapaan kuin harvinaisemman kohdalla (**Kuva 10.**).



Kuva 10. Entinen myyntiprosessi.

5.2 Uusi myyntiprosessi

Uutta myyntiprosessia lähdettiin rakentamaan siitä ajatuksesta, että myyjä saa tarvittavan tiedon materiaalista omalta päätteeltään, eikä hänen tarvitse lähteä asiaa itse tarkistamaan tai pyytämään tietoa muualta. Tällä tavalla kevennettiin myös työnjohdon kuormaa ja koneenkäyttäjä jäi tästä prosessista kokonaan pois (**Kuva 11**). Myyjä tarkistaa varastosaldon omalta koneeltaan ja jos materiaalia on saatavilla, pystyy hän määrittämään toimitusajan itse. Työnjohtoa tarvitaan silloin, kun materiaalia ei ole saatavilla ja sitä pitää tilata. Tällöin työnjohto tilaa oikean erän materiaalia ja informoi materiaalin luvatusasta toimitusajasta myyntiin, joka vahvistaa tilauksen.



Kuva 11. Uusi myyntiprosessi

6 ALIHANKINTAPROSESSI

Ohjelman tehnyt yritys käytti scrumia, jonka ansiosta tilaaja-organisaatio pysyi hyvin perillä projektin etenemisestä. Tilauksen jälkeen määriteltiin aloituspalaverissa roolit, sprintin pituuden ja sovittiin kommunikoinnista. Samalla kertaa työstettiin tehtävälista eli product backlog, joka toimi koko tekemisen tukirankana projektin ajan. Kyseessä oleva projekti oli ehkä hieman liian pieni scrumille, joten sprintin perusajatus toimivasta versiosta ei tullut alkuvaiheessa kysymykseen, koska ensin piti rakentaa niin sanottu backend eli käyttäjälle näkymätön osa valmiiksi. Lopuvaiheessa pystyttiin joka sprintin lopuksi toimittamaan toimiva versio.

6.1 Roolit

Product ownerina oli tehtaan tietohallintopäällikkö. Scummasterina toimi ohjelman tehneen yrityksen toimitusjohtaja. Kehitystiimissä oli toimitusjohtajan lisäksi yrityksen toinen omistaja ja myöhemmin siihen lisättiin vielä ohjelmistosuunnittelija.

Product owner edusti tehdasta, jonka sisällä oli komitea, jonka kokoonpano oli tietohallintopäällikön lisäksi tuotantopäällikkö, työnjohtaja sekä avainkäyttäjäksi valittu koneenkäyttöjä.

6.2 Sprintin suoritus ja kommunikointi

Sprint määriteltiin kahden viikon mittaiseksi. Sprintin päätyttyä pidimme sprinttikatselmuksen, jossa tutustuttiin tehtyihin ominaisuuksiin ja sovimme seuraavan sprintin sisällöstä. Projektin myöhemmässä vaiheessa, kun backend oli rakennettu, oli sprintin päätyttyä esittelyssä ohjelmaversioversio. Alkuvaiheessa työstettiin taustajärjestelmiä, joten konkreettista näytettävää ei vielä ollut. Se ei kuitenkaan haitannut, sillä Product owner omaa sen verran ohjelmointitaustaa, että pystyi näkemään kehityksen ja tehdyn työmäärän valmiin koodin perusteella.

7 PROJEKTIN LÄPIVIEN TI YRITYKSESTÄ

Huolimatta projektin hallintavasta, on projektin onnistumisen kannalta tärkeintä kuitenkin käyttöönotto. Tutkimusten mukaan vain noin kolmasosa IT-projekteista onnistuu täydellisesti. Suurempi osa projekteista asettuu epämääräiselle alueelle, jossa saatiin jotain tuloksia, mutta niihin ei olla täysin tyytyväisiä. Valitettavasti neljäsosa projekteista lasketaan täysin epäonnistuneiden joukkoon. Näistä asioista löytyy käytännön kokemusta myös asiakasyhtiössä, joka on perustettu jo 1960-luvulla. Yli 50 vuoden ajanjaksoon mahtuu onnistumisia ja hieman pieleen menneitä projekteja.

7.1 Henkilöstön sitoutus

Yksi asia, jolla saa projektin varmimmin pieleen, on epäonnistua henkilöstön sitouttamisessa siihen. Tällä tavalla toimiessa vain ilmoitetaan uudesta ohjelmasta, pahimmillaan aivan loppuhetkillä. Tällöin henkilössä syntyy voimakas vastareaktio ja tunne siitä, että asioihin ei ole päässyt vaikuttamaan lainkaan.

Tämän työn aiheena oleva projekti sai alkunsa juuri henkilöstön tarpeesta. Otimme henkilöstön mielipiteitä ylös aiemmin kuvatussa haastatteluvaiheessa ja valitsimme henkilöstön edustajan komiteaan. Lisäksi pidimme infotilaisuuksia myös koko henkilöstölle. Tällä pyrittiin luomaan aito keskusteluyhteys henkilöstöön päin, sillä vain ohjelman omaksi ottanut henkilökunta käyttää ohjelmaa.

7.2 Pohjatiedot järjestelmää varten

Järjestelmässä oli muutamia pohjatietoja, jotka syötettiin järjestelmään ennen käyttöönottoa. Ne olivat nimikkeet, varastopaikat, koneet ja käyttäjät. Järjestelmään ei luotu graafista lisäympäristöä, vaan tiedot syötettiin CSV tyyppisesti järjestelmään. Tähän tapaan päädyttiin siitä syystä, että alkuvaiheen käyttöönoton jälkeen muutoksia tulee harvoin ja hallintaympäristöön panostaminen olisi vienyt resursseja pois tärkeimmistä ominaisuuksista.

Nimiketiedot saatiin suoraan ERP-järjestelmästä. Niistä saatiin kohtuullisen vähällä vaivalla CSV tyyppinen tiedosto (**Kuva 12.**), jonka siirto ohjelmistoon oli vaivatonta. Haasteet olivat Excelissä, jossa piti muuttaa sen käyttämä tuhaterotin toiseksi, eli pilkusta pisteeksi.

```

id;code;name;technicalName;sheetThickness;sheetWeight
1;MK055N07022;Luukku Raina 0.7x227;DX51D+Z275MAC;1;0
2;MK055N07036;Luukku Raina 0.7x360;DX51D+Z275MAC;1;0
3;ML000;Jätepala;Jätepala;0;0
4;ML051J15125163;Bobi 1.0x1250x1638 DX51D+Z100MBU;Business;1;16.36
5;ML051J15125200;Bobi 1.5x1250x2000 DX51D+Z100MBU;;2;25
6;ML055N05100200;Kuumasinkitty levy 0.5x1000x2000;DX51D+Z275MBC ROHS;1;8
7;ML055N06125250;Kuumasinkitty levy 0.6x1250x2500;DX51D+Z275MBC ROHS;1;15
8;ML055N07100200;Kuumasinkitty levy 0.7x1000x2000;DX51D+Z275MBC ROHS;1;11.2
9;ML055N07100250;Kuumasinkitty levy 0.7x1000x2500;DX51D+Z275MBC ROHS;1;14
10;ML055N07125200;Kuumasinkitty levy 0.7x1250x2000;DX51D+Z275MBC ROHS;1;14
11;ML055N07125250;Kuumasinkitty levy 0.7x1250x2500;DX51D+Z275MBC ROHS;1;17.5
12;ML055N07125250;Kuumasinkitty levy 0.7x1250x2500;DX51D+Z275MBC ROHS;1;17.5
13;ML055N07125300;Kuumasinkitty levy 0.7x1250x3000;DX51D+Z275MBC ROHS;1;21
14;ML055N07150300;Kuumasinkitty levy 0.7x1500x3000;DX51D+Z275MBC ROHS;1;25.2
15;ML055N07100200;Kuumasinkitty levy 0.7x1000x2000;DX51D+Z275MBC ROHS;1;11.2

```

Kuva 12. Materiaalien tuonti CSV-tiedostolla

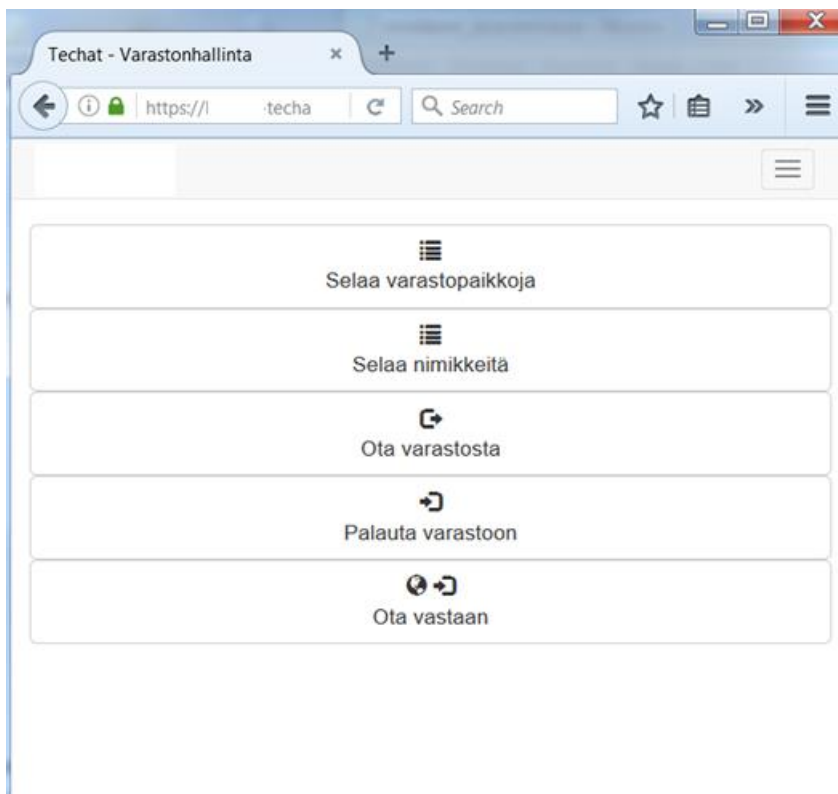
Varastopaikoissa oli selvästi eniten tehtävää ennen käyttöönottoa. Yrityksessä oli hiljattain tehty layout eli tuotantokoneiden järjestysmuutos, jonka seurauksena hyllymerkinnät eivät olleet ajan tasalla. Hyllykirjaimen lisäksi otettiin käyttöön uusi taso merkinnöissä eli puomi. Aiemman pelkän hyllyn sijaan merkittiin nyt hylly ja puomi. Tällä tavalla saatiin tarkkuuteen merkittävä parannus.

Käyttäjiksi lisättiin koko henkilöstö. Käyttäjien lisäämisessä suurin haaste oli salasanan muodostus, sillä se generoitiin tietoturvalisella menetelmällä.

7.3 Sovelluksen käyttöönotto

Yrityksen sisällä oli paljon eri tehtäviä, mitkä kuuluivat projektiin. Lähdimme liikkeelle esikartoituksesta ja määrittelystä, minkä jälkeen siirryttiin hyväksyntävaiheen jälkeen projektisuunnitelman tekoon. Monista projekteista viisastuneina, panostimme henkilöstön sitouttamiseen. Tästä syystä valitsimme henkilöstöstä avainhenkilöt, jotka olivat vahvasti mukana projektin edetessä, niin testausvaiheessa kuin koulutuksessakin. Tällä tavalla onnistuimme sitouttamisessa keskimääräistä paremmin.

Valmista ohjelmaa käytettiin pääasiassa tabletilla (**Kuva 13.**), joten hankimme sopivia laitteita ja samalla laajensimme tehtaassa olevaa langatonta verkkoa käyttötarkoitukseen sopivaksi. Itse varastohyllyille piti tehdä useita asioita, kuten merkintää ja siivousta. Projektin loppupuolella oli testauksen jälkeen materiaalin laskeinta ja lähtötietojen syöttö. Lopuksi siirryttiin tuotantokäyttöön ja tämä projekti päättyi. Uusia ominaisuuksia lisätään vaiheittain.



Kuva 13. Alustariippumaton ohjelma skaalaa aloitusnäkyvän laitteen ja sen asennon mukaan.

8 TESTAUS

Yhtenä tärkeänä osana ohjelmistoprojekteissa on testaus. Testausta voidaan suorittaa useilla tasoilla, joista ensimmäinen, eli yksikkötestaus, testaa pieniä ohjelmiston komponentteja tai moduuleja. Testaamalla jokaisen yksikön varmistutaan siitä, että suunnittelu on toteutettu oikein. Integraatiotestaus taas paljastaa viat rajapinnoissa ja vuorovaikutuksessa. Järjestelmätestauksessa testataan jo kokonaan integroitua järjestelmää ja siinä varmistutaan, että se vastaa vaatimuksia.

Keskeneräistäkin koodia voidaan testata dynaamisesti ajamalla koodia läpiä testitapauksilla. Tämä vaihe toteutetaan ohjelman ensimmäisillä ajokerroilla. Ennen ohjelmiston viimeistä versiota, suoritetaan usein Alpha-testaus ja Beta-testaus. Ensimmäisessä vaiheessa tutkitaan tehtyjen ominaisuuksien toimintaa käytännön tilanteissa. Tässä pyritään siihen, että ominaisuus toimii oikein. Lisäksi viimeistään tässä vaiheessa saatetaan sisäinen koodi dokumentoiduksi. Tämä osuus testauksesta on tyypillisesti ohjelmoitsijoiden tehtävä. Beta-testauksessa pyritään löytämään ohjelmointivirheitä eli bugeja. Bugeja saattaa syntyä, kun alpha-vaiheessa tehdään muutoksia. Beta-testaus on tyypillisesti erittäin aikaa vievä vaihe.

Vaikka valmistelut olisivat joka vaiheessa huolelliset ja jopa täydelliset, virheitä löytyy aina. Niiden löytäminen on ensiarvoisen tärkeää, sillä virheitä täynnä oleva ohjelmisto saa aikaan huonon käyttökokemuksen. Tässä kyseisessä työssä haluttiin saada käyttäjille mahdollisimman virheetön ohjelma, jotta käyttöönotto sujuu parhaalla mahdollisella tavalla. Projektissa ei missään tapauksessa haluttu tilannetta, jossa loppukäyttäjä löytää kriittisen virheen, sillä jokainen ohjelman ominaisuuksista on tärkeä ja siinä oleva virhe kaataisi koko toimintaketjun.

Niin tärkeää kuin testaus onkin, ei se yksinään takaa laadukasta lopputuotetta. Tämä tarkoittaa sitä, että projektin ensimmäisestä vaiheesta alkaen on syytä mennä laatu edellä. Tämä siksi, että huolellinenkaan testaus ei löydä kaikkia virheitä, ja virheiden korjaaminen aiheuttaa usein uuden virheen muualla. Testauksen jälkeen tehdään tarvittavat muutokset. Jos muutos vaaditaan, ylätasolle eli toiminnallisuuteen, ollaan oltu huolimattomia määrittelyvaiheessa.

Koska tavoitteena on saada aikaiseksi mahdollisimman virheetön ohjelma, pitäisi testauksessa löytää mahdollisimman paljon virheitä. Käytännössä ohjelma pyritään saamaan virhetilaan. Hyvä testaaja omaa hajottavan asenteen, vaikka löydetty virhe aiheuttaakin lisätöitä. Tästä syystä testaajien on järkevä olla eri henkilö kuin ohjelmiston toteuttaja. Tällöin ei jätetä virheitä kirjaamatta, koska se aiheuttaisi itselle lisätöitä.

Ennen asiakkaan laitteilla ja olosuhteissa toteutettavaa hyväksyttämistestausta on toteutettu järjestelmätestaus ohjelmoitsijoiden toimesta. Hyväksyttämistauksessa on tavoitteena todentaa järjestelmän valmius ja kyky toteuttaa asetetut vaatimukset. Käytännön toteutus suoritettiin siten, että testiympäristö tuotiin tabletilla käytettäväksi, ensin avainkäyttäjälle ja myöhemmin varastohenkilölle.

Tabletin valinnassa tärkeintä oli hinta/laatusuhde. Laitteessa piti olla hyvä kamera, koska sitä tarvittiin viivakoodin lukua varten. Käyttöjärjestelmän puolesta iOS ja Microsoft eivät tulleet kysymykseen, sillä niillä varustetut laitteet olivat liian hintavia. Kestävät tabletti olisi teoriassa ollut paras vaihtoehto, sillä käyttökohde on tehdashalli ja olosuhteet sen mukaiset, mutta niiden hinta ja tekniset ominaisuudet eivät olleet tasolla, joka olisi hyväksyttävä. Käytännössä tavallinen tabletti normaallilla suojakotelolla oli paras vaihtoehto. Laitteen tarkka malli oli Lenovo Tab2 A10 ja siihen suojakotelo. Sen hankintahinta oli 200 €:n luokkaa. Tätä hintaa voi verrata pudottamista kestävään malliin, jonka hinta on 1500 €+ luokassa.

9 YHTEENVETO, LOPPUPÄÄTELMÄ JA JATKOTOIMENPITEET

Ohjelmistoprojekti sai alkunsa tehtaan henkilöstön muutosehdotuksesta. Muutosta yritettiin ensin toteuttaa noudattamalla sovittua järjestystä tuotteelle, mutta jonkin ajan kuluttua kävi selväksi oman ohjelman tarpeellisuus. Hankinnassa noudatettiin huolellista valmistelua, jossa haastateltiin käyttäjät, kartoitettiin olemassa olevat ohjelmat ja valittiin ohjelmistotoimittaja. Toimittaja sai hyvän määrittelyn ja pystyi antamaan tarkan hinta-arvion ohjelmasta, jota käytetään pääasiassa tablettilla. Ohjelmistokehityksessä toimittaja käytti scrum-menetelmää, jonka kantava ajatus on lyhyet jakson, jonka jälkeen asiakkaalle toimitetaan osittain toimiva versio lopputuotteesta. Ohjelmoinnin jälkeen suoritettiin koulutus ja käyttöönotto.

Käytetystä ohjelmistokehitysmenetelmästä huolimatta ensimmäinen toimiva versio oli esittelyssä noin projektin puolivälissä. Tämä selittyy osittain sillä, että ensin piti tehdä runkoa valmiiksi, jotta sen päälle voi rakentaa mitään asiakkaalle näkyvää. Toisaalta asiakas olisi product ownerin kautta voinut vaatia esim. kuvia tulevasta käyttöliittymästä. Niiden puuttuessa jouduttiin vain kuvittelemaan miltä lopputulos näyttää ja luottamaan siihen, että toimittaja tekee sen mitä luvattiin. Kahden viikon välein pidettiin scrumiin kuuluvat palaverit ja niissä esiteltiin valmista koodia, mutta niistä oli vaikea rakentaa kokonaiskuvaa edistyksestä. Todettakoon kuitenkin, että toimittaja suoriutui erinomaisesti palvelen jopa käyttöönottopäivänä, joka oli vappu.

Ohjelmiston koodausvaiheen ollessa valmis pystyimme suorittamaan henkilöstön koulutuksen. Sen toteutimme siten, että sitoutetut pääkäyttäjät kouluttuvat työtovereitaan. Tällä tavoin toteutettuna koulutus sujui hyvin. Käyttöönottopäivä oli vapuna, jolloin varastosaldot syötettiin suoraan järjestelmään. Tässä oli työ mennä hukkaan, mutta tilanteesta selvittiin vähin vaurioin.

Ohjelma on tätä kirjoittaessa ollut käytössä noin puoli vuotta ja siihen on tehty muutamia muutoksia, joihin on tullut ehdotus suoraan käyttäjiltä. Lisäksi inventointitoimintoja on testattu jo kahdesti ja näistä on saatu uusia ideoita. Käyttöaste on korkea,

mutta sitä voidaan kuitenkin vielä parantaa koulutuksin. Seuraavia merkittäviä kehityshankkeita on hälyrajojen lisääminen ohjelmistoon ja mahdollinen integraatio ERP-järjestelmän kanssa.

LÄHTEET

/1/ Business Process Modeling Notation foorumi ja sen yleisimmät kysymykset Viitattu 19.11.2016 <http://bpmnforum.com/bpmn-faq/>

/2/ Unified Modeling Language. Virallinen organisaatiosivusto Viitattu 19.11.2016 <http://www.uml.org/>

/3-5/ Lean Software Engineering Viitattu 19.11.2016 <http://leansoftwareengineering.com/ksse/scrum-ban/>

Kniberg, H. 2009 Kanban vs Scrum – How to make the most of both Viitattu 19.11.2016 <https://www.crisp.se/file-uploads/Kanban-vs-Scrum.pdf>

Infoq – Enterprise Software Development Community Viitattu 19.11.2016 <https://www.infoq.com/presentations/kanban-for-software>

/6/ Redmine organisaation virallinen sivusto Viitattu 19.11.2016 <http://www.redmine.org/>

/7/ The Benefits of MES: A Report from the Field, 05/97 Viitattu 19.11.2016 <http://www.cpdee.ufmg.br/%7Eseixas/PaginaII/Download>

/DownloadFiles/pap1.pdf