

Lappeenrannan-Lahden teknillinen yliopisto LUT
School of Engineering Science
Tietotekniikan koulutusohjelma

Kandidaatintyö

Lauri Hämäläinen

Laaturaportointityökalun suunnittelu ja toteutus teollisuuskonseptiin

Työn tarkastaja(t): Tutkijatohtori Ari Happonen

Työn ohjaaja(t): Tutkijatohtori Ari Happonen

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT

School of Engineering Science

Tietotekniikan koulutusohjelma

Lauri Hämäläinen

Laaturaportointityökalun suunnittelu ja toteutus teollisuuskonseptiin

Kandidaatintyö

2019

38 sivua, 6 kuvaa, 10 taulukkoa, 4 liitettä

Työn tarkastajat: Tutkijatohtori Ari Happonen

Hakusanat: kandidaatintyö, laatu, vaatimusmäärittely, LEAN, ohjelmistokehitys, arvo, hukka

Keywords: bachelors' thesis, quality, software requirements specification, LEAN, software development, value, waste

Tämän kandidaatintyön tarkoituksena on suunnitella ja toteuttaa laaturaportointityökalu ja kuvata toteutusta yleisen vaatimusmäärittelyn ja LEAN-ajattelun kannalta kohdeyrityksessä. Toteutetun laaturaportointityökalun avulla yritys saa kerättyä tärkeää dataa tuotteidensa laadusta ja pystyy informoimaan paremmin työntekijöitään tuotannon sen hetkisestä tilasta laadun näkökulmasta. Vaatimusmäärittelyn ja LEAN-ajattelun soveltumista tämän ohjelmistoprojektin toteutukseen tutkitaan vertaamalla toteutuksien eroavaisuuksia. Tuloksista perusteella perinteisen vaatimusmäärittelyn antaa hyvän rungon toteutukselle ja LEAN-ajattelun tehostavan etenkin työn testausvaihetta.

ABSTRACT

LUT University
School of Engineering Science
Degree Program in Computer Science

Lauri Hämäläinen

The design and implementation of a quality reporting tool for industrial concept

Bachelor's Thesis

2019

38 pages, 6 figures, 10 tables, 4 appendices

Examiners: D.Sc. (Tech.) Ari Happonen

Keywords: bachelors' thesis, quality, software requirements specification, LEAN, software development, value, waste

The purpose of this bachelor's thesis is to design and implement a quality reporting tool and describe the implementation in terms of general software requirement specification and LEAN thinking in the target company. The implemented quality reporting tool enables the company to gather important quality data of its products and inform its employees better about the current state of production from a quality point of view. The applicability of software requirement specification and LEAN thinking in this software project is researched by comparing differences between literature and practical implementations. According to the results, the general software requirement specification provides a good framework for implementation and LEAN thinking enhances especially the testing phase of the work.

ALKUSANAT

Kandidaatintyö on tehty tietotekniikan koulutusohjelmaan 2019. Aihe työlle tuli entiseltä työnantajalta, jota haluan kiittää mahdollisuudesta tehdä opinnäytetyö ja suorittaa samalla päätökseen kesken jääneet kandidaatin opinnot. Haluan myös kiittää työpaikkaohjaajaani kannustavasta asenteesta työn suorittamisessa ja työn ohjaajaa Ari Haposta, joka pitkäksi venyneestä aikataulusta huolimatta jaksoi valaa uskoa työn onnistumiseen.

Erityiskiitokset avovaimo Annille jaksamisesta ja kolmilapsisen perheen arjen pyörittämisestä, kun keskityin töiden ohella opinnäytetyön tekoon.

Lappeenrannassa, 1.11.2019,

Lauri Hämäläinen

SISÄLLYSLUETTELO

1	JOHDANTO	4
1.1	TAUSTA	4
1.2	TAVOITTEET JA RAJAUKSET	5
1.3	TYÖN RAKENNE	6
2	VAATIMUSMÄÄRITTELY SOVELLUSKEHITYKSESSÄ	8
2.1	VAATIMUSMÄÄRITTELY YLEISESTI	8
2.2	VAATIMUSMÄÄRITTELYN VAIHEET	10
2.2.1	<i>Vaatimusten kartoittaminen</i>	10
2.2.2	<i>Vaatimusten analysointi</i>	10
2.2.3	<i>Vaatimusten hyväksyminen</i>	11
3	LEAN-AJATTELU OHJELMISTOKEHITYKSESSÄ	12
3.1	LEAN YLEISESTI	12
3.1.1	<i>Hukat ohjelmistokehityksessä</i>	13
3.1.2	<i>Arvon tuottaminen ohjelmistokehityksessä</i>	14
3.1.3	<i>LEAN-ajattelun toteutus</i>	15
4	VAATIMUSMÄÄRITTELYN TOTEUTUS KOHDEYRITYKSESSÄ	17
4.1	VAATIMUSTEN KARTOITUS	17
4.2	VAATIMUSTEN ANALYSOINTI	19
4.3	VAATIMUSTEN HYVÄKSYNTÄ	20
5	OHJELMISTON TOTEUTUS KOHDEYRITYKSESSÄ	21
5.1	OHJELMISTON TOTEUTUKSEN VAIHEET	21
5.1.1	<i>Suunnittelu</i>	22
5.1.2	<i>Toteutus</i>	23
5.1.3	<i>Testaus</i>	26
5.1.4	<i>Käyttöönotto</i>	27
5.1.5	<i>Ylläpito</i>	27
5.1.6	<i>Jatkokehitys</i>	27
5.2	LEAN-AJATTELUN TOTEUTUMINEN	28
5.2.1	<i>Hukkien poisto</i>	28
5.2.2	<i>Arvon tuottaminen</i>	29
5.2.3	<i>LEAN-ajattelun toteutukselle tuottama arvo</i>	30
6	TULOKSET	31

7 JOHTOPÄÄTÖKSET JA YHTEENVETO.....	34
LÄHTEET.....	36
LIITTEET	

SYMBOLI- JA LYHENNELUETTELO

Excel pivot-tilukko	Taulukko, joka suodattaa tehokkaasti käyttäjän haluamia tietoja isosta tietokannasta
Excel-makro	Excelin toimintoja ja käyttäjän komentoja ketjuttava pieni ohjelma.
Excel-tietokanta	Microsoft Excel-tilukkolaskentaohjelman avulla luotu suurikokoinen tilukko, jota käytetään tiedon varastointiin.
Google Forms	Googlen maksuton verkkopalvelu, jossa voi luoda lomakkeita ja kyselyjä
ID	Tunniste (engl. Identifier)
LEAN	Tuotannon tehostukseen ja laadun parantamiseen keskittyvä ajattelutapa
.	
Microsoft Windows	Microsoft Windows -käyttöjärjestelmä
PDF	Portable Document Format -tiedostomuoto. Käytetään sähköiseen tulostukseen.
Pvm	Päivämäärä
UC-1	Käyttötapaus (engl. Usercase)
VBA	Visual Basic for Applications -ohjelmointikieli. Käytetään Microsoft Office -toimisto-ohjelmistoissa luomaan esimerkiksi makroja ja käyttöliittymiä.

1 JOHDANTO

Teollisessa ympäristössä toteutettu paperinen raportointi alkaa olla historiaa 2020-luvulle saavuttaessa. Yksi syy tälle kehitykselle on vahvasti päällä oleva digitalisaatio (Kortelainen & Happonen, 2017; Happonen & Minashkina, 2018; Kortelainen et al, 2019), joka muokkaa yhteiskuntaamme hyvin tehokkaasti kohti paperittomia prosesseja. Teollisuudessa laajasti käytössä olevan LEAN-filosofia tarvitsee myös tehostamista tämän päivän työkaluilla. (Promaint, 2016) (IBM, 2018). Teollisten prosessien digitalisaation eduiksi voidaan laskea materiaalikustannusten lasku, raportointiin käytetyn ajan lyheneminen ja raportoitavan tiedon tehokkaampi hyödyntäminen. Kandidaatintyö keskittyi juuri näiden hyötyjen saavuttamiseen graafisen laaturaportointityökalun avulla. Kandidaatintyö vertaa yleisiä sovelluskehityksen vaatimusmäärittelyn vaiheita käytännön sovelluskehitykseen päivittäistavaroita valmistavassa yrityksessä.

1.1 Tausta

Kandidaatintyö suoritetaan suomalaisessa teollisuusyrityksessä. Laatu on ratkaisevassa roolissa kohdeyrityksen toiminnassa ja arvoissa, niin ruohonjuuritasolla kuin myös yrityksen brändissä. Yrityksen tuotteet ovat hintatasoltaan ns. Premium-tuotteita. Kilpailevia tuotteita korkeampi hinta pitää sisällään lupauksen asiakkaalle laadullisesti paremmasta tuotteesta (Kihn 2015, 291). Tuotelaatua valvomaan yritys on ottanut käyttöön laadunhallintajärjestelmän. Osana laadunhallintajärjestelmän käytännötoteutusta yrityksen työntekijät arvioivat päivittäin edellispäivänä tuotettujen tuotteiden laatua. Laatua arvioidaan numeerisesti ennalta määriteltujen kriteerien mukaisesti. Laatukriteerit ovat esillä tuotanto-osastojen ja tuotearviontihuoneen seinillä olevissa tuotetauluissa. Tuoteteiden arvioitaviin ominaisuuksiin kuuluu muun muassa paino, pituus, leveys sekä aistipohjaiset ominaisuudet. Aistihavaintoihin pohjautuvat kriteerit estävät prosessin täysautomatisoinnin.

Laatuarvioinnin suorittaa aamuvuoroon tuleva tuotantotyöntekijä, joka kirjaa laatupisteet paperille samalla silmäillen tuotetauluja. Tämän jälkeen tuotantotyöntekijä laskee eri osalueiden pisteet yhteen ja kopioi yhteispisteet paperilta tuotearviontihuoneessa olevalla tietokoneella Excel-tietokantaan. Tietokantaan lisätään myös tieto tuotteiden laatuun liittyvistä poikkeamista ja mahdolliset lisäselitteet poikkeamien synnystä. Myöhemmin

aamulla aamupalaveria pitävä tuotantoesimies käy läpi laatupisteet ja poikkeamat tuotantotyöntekijöiden kanssa.

Laatuvastaava raportoi kerran kuussa laatupisteraportin eteenpäin yrityksen organisaatiossa. Raportti pitää sisällään yksinkertaisen tuotekohtaisen pistekeskiarvolaskelman ja kommentoinnin mahdollisista laatuun vaikuttavista poikkeamista ja niiden syistä. Kuukausiraporttien pohjalta organisaatio seuraa toiminnan laadukkuutta ja käyttää raportteja osana tuotanto-, budjetointi- ja investointipäätöksiä.

1.2 Tavoitteet ja rajaukset

Työn tavoitteena on toteuttaa ohjelma, jonka tärkein funktio on yhtenäistää ja helpottaa arviointiprosessia ja piilottaa arvioitsijoilta heille epäolennaisia, mutta organisaatiolle tärkeitä tietoja. Toinen tärkeä funktio on kerätä laaturvioinnissa saatu data tietokantaan, josta laatuvastaava ja tehtaanjohtaja voivat analysoida dataa, raportoida tuloksia eteenpäin organisaatiossa ja seurata laatuun liittyvien kehitysprojektien vaikutuksia. Kolmas tärkeä funktio työkalulla on luoda aamupalavereja pitävälle tuotantoesimiehelle graafinen esitys edellisen päivän, viikon tai kuukauden laaturaportoinneista. Työkalu korvaa tuotekatselmukseen käytetyn paperisen version.

Tavoitetila on laadunraportoinnin digitalisointi graafisen käyttöliittymän omaavalla laadunraportointityökalulla, jonka avulla laadunarvioija suoriutuu tuotekatselmuksesta 10 minuutissa ja joka dokumentoi halutun laatudatan rakenteellisesti selkeään tietokantaan, josta yksityiskohtaisempaa tietoa hakeva esimies löytää tiedon vaivattomasti. Työkalu helpottaa työnjohtajien työtä laaturaporttien analysoinnissa ja tulosten esittämisessä valmiilla kuvaajilla laatupisteiden kehityksestä haluttuna ajanjaksona ja laatu-poikkeamien määrän ja tyyppien. Selkeä ja informatiivinen esitys auttaa kaikkia laatupalavereihin osallistuvia tahoja (joka osastolta työntekijä ja mahdollisuuksien mukaan myös esimies) sisäistämään paremmin tämänhetkisen tilanteen, hahmottamaan ongelmakohtat, herättämään keskustelua ongelmista ja löytämään ongelmiin ratkaisuja.

Tavoitetilassa raportointityökalu on helposti muokattavissa ja laajennettavissa. Valmiin ohjelman muokkaaminen ja laajentaminen eivät vaadi ohjelmointitaitoja, koska ohjelma on

toteutettu dynaamisesti ja dokumentoitu kattavasti. Tämä mahdollistaa työkalun käyttöönoton varsin pienellä vaivalla myös yrityksen muissa vastaavissa yksiköissä.

Ohjelmistorajoituksena on yrityksen vaatimus toteuttaa ohjelma Microsoft Excel -taulukko-laskentaohjelmalla ja tarkemmin VBA (Visual Basic for Applications) -ohjelmointikielellä. Käyttöjärjestelmävaatimuksena oli toteuttaa ohjelma Microsoft Windows käyttöjärjestelmälle (versio 7 tai uudempi).

Tässä työssä toteutettavan raportointityökalun julkaisuversioon ei tulla sisällyttämään tukea yksityiskohtaisten tuotekuvien lisäykselle tietokantaan tai tukea mobiililaitteiden käytölle. Tietokannan rakenne ei mahdollista kuvien lisäystä ilman, että tietokannasta tulisi liian suuri toimiakseen. Myöskään pilvipalvelun käyttö kuvien ja niiden linkkien lisäämiseen ei tule työn toteutushetkellä kysymykseen suuren datakapasiteettivaatimuksen ja huomattavan vaikean toteutuksen vuoksi. Tuki mobiililaitteille ei myöskään ole toteutushetkellä mahdollista käyttöympäristön ohjelmallisten rajoitusten vuoksi.

Raportointityökalun onnistumista mitataan tyytyväisyyskyselyillä ennen ja jälkeen työkalun julkaisun. Kyselyt toteutetaan Google Forms -ohjelmalla ja niistä julkaistaan omat versiot työntekijöille sekä esimiehille. Työntekijöille suunnatut kyselyt keskittyvät tuotekatselmuksen ja kirjausten toteutuksen numeeriseen arviointiin. Esimiehille suunnatut kyselyt keskittyvät puolestaan aamupalaverin laatuosion numeeriseen arviointiin ja laadudatan hyödyntämiseen työssä.

1.3 Työn rakenne

Työn tarkoituksena on vastata seuraaviin tutkimuskysymyksiin:

- Soveltuuko yleinen sovelluskehityksen vaatimusmäärittelyn malli toteutettavaksi tässä käyttötapauksessa?
- Kuinka LEAN-ajattelu soveltuu ohjelmistokehitykseen tässä käyttötapauksessa?

Työn alatutkimuskysymys on:

- Voidaanko paperinen raportointi korvata onnistuneesti digitaalisella raportointityökalulla?

Kandidaatintyö on jaettu kahdeksaan lukuun. Ensimmäisessä luvussa on johdanto ja tutkimusaiheen esittely. Toisessa luvussa käydään läpi vaatimusmäärittely sovelluskehityksessä teoriatasolla kuvaten yleisimmät vaatimusmäärittelyn vaiheet. Kolmannessa luvussa käsitellään ohjelmistokehitystä LEAN-ajattelun pohjalta hyödyntäen teorian esittelyssä kirjallisuutta ja julkaisuja. Neljännessä luvussa tarkastellaan vaatimusmäärittelyn toteutusta kohdeyrityksessä. Kuinka vaatimusmäärittely käytännössä toteutettiin. Viidennessä luvussa kerrotaan ohjelmiston toteutuksesta kohde yrityksessä ja kuinka LEAN-ajattelua hyödynnettiin ohjelmiston toteutuksessa. Kuudennessa luvussa käydään läpi ohjelmistoprojektin toteutuksen tulokset. Seitsemännessä luvussa käydään läpi johtopäätökset. Ovatko yleiset sovelluskehityksen vaatimusmäärittelyn vaiheet sovellettavissa samantyyppisiin teollisuusyrityksiin ja onko käyttäjäpalautteen kerääminen olennaista ohjelmistoprojektin onnistumisen kannalta. Kahdeksannessa luvussa yhteenveto, mitä projektista opittiin ja toimiko oma toteutus.

2 VAATIMUSMÄÄRITTELY SOVELLUSKEHITYKSESSÄ

Tässä luvussa käydään läpi vaatimusmäärittelyä sovelluskehityksessä yleisellä tasolla ja sitä mitä vaiheita vaatimusmäärittelyyn sovelluskehityksessä kuuluu. Luvun tarkoitus on antaa lukijalle perustieto vaatimusmäärittelyn rakenteesta. Luvussa ei kuvata vaatimusmäärittelyn jälkeisiä työvaiheita tai niiden hallintaan käytettäviä työkaluja tai menetelmiä.

2.1 Vaatimusmäärittely yleisesti

Vaatimusmäärittelyssä pohjimmiltaan määritellään mitä ohjelmistoa ollaan toteuttamassa, miksi ja miten ohjelmisto toteutetaan (Savolainen, 2019). Vaatimusmäärittelyllä pyritään luomaan yhteisymmärrys ohjelmistoprojektiin osallistuvien toteuttavien tahojen, asiakkaan (käyttäjryhmien ja eri sidosryhmien) välille projektin tavoitteista ja toteutuksesta. Yhteisymmärryksellä vältetään väärintulkinnat ja ristiriidat ohjelmiston toteutuksessa ja lopputuloksessa (Lintula, 2004, s.9). Vaatimusmäärittelydokumenttiin kirjataan ylös vaatimukset (Taulukko 1. Vaatimukset) ja rajoitukset (Taulukko 2. Rajoitukset), joita kukin käyttäjä- ja sidosryhmä projektille asettaa (Taulukko 3. Käyttäjryhmät) (Taulukko 4. Sidosryhmät) (JUHTA, 2018, s.13-14) (Savolainen, 2019).

Taulukko 1. Vaatimukset

ID	Pvm	Vaatimus	Vaatimuksen kuvaus	Prioriteetti

Taulukko 2. Rajoitukset

ID	Pvm	Lähde	Rajoite	Rajoitteen kuvaus	Huomautuksia

Taulukko 3. Sidosryhmät

ID	Pvm	Sidosryhmä	Sidosryhmän kuvaus ja rooli	Huomautuksia

Taulukko 4. Käyttäjärühmät

ID	Pvm	Käyttäjärühmä	Kuvaus ja rooli	Huomautuksia

Vaatimusmäärittelyn vaatimukset jakaantuvat toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Toiminnalliset vaatimukset ovat ohjelmiston ominaisuuksia, joita ohjelman tulee suorittaa (Lintula, 2004, s.9). Toiminnallisia vaatimuksia voidaan dokumentoida vaatimusmäärittelydokumenttiin luomalla yksityiskohtaisia käyttötapauskuvauksia (Taulukko 4. Käyttötapauskuvaus) (Savolainen, 2019)

Taulukko 5. Käyttötapauskuvaus

ID	UC-1
Nimi ja versio	
Suorittajat	
Esiehdot	
Kuvaus	
Poikkeukset	
Lopputulos	
Muut vaatimukset	

Ei-toiminnalliset vaatimukset kohdistuvat ohjelmiston ominaisuuksiin, jotka määrittelevät kuinka ohjelmisto toimii laadullisesta näkökulmasta. Ei-toiminnallisiin vaatimuksiin kuuluu esimerkiksi vaatimus tietystä suoritusnopeudesta tai ohjelmistokoodin dokumentaatio (Lintula 2004, s.9).

Rajoitteet pitävät sisällään ohjelmiston toteutukseen liittyviä teknisiä, taloudellisia tai ajallisia rajoitteita (JUHTA, 2018, s. 13). Vaatimuksista poiketen rajoitteita voi tulla myös ohjelmiston toteuttajan puolelta, kuten esimerkiksi rajoite käytettävissä olevien työntekijöiden määrälle.

2.2 Vaatimusmäärittelyn vaiheet

Vaatimusmäärittelyn vaiheet pitävät sisällään alustavan vaatimuskartoituksen, vaatimusten tarkentamisen ja vaatimusten hyväksymisen (Haikalta & Mikkonen, 2011, s. 66). Tärkeää vaatimusten määrittelyssä ja etenkin niiden keräämisessä on se, keneltä vaatimuksia tai niiden hyväksymiseen tarvittavaa tietoa kerätään ja miten tietoa kerätään tehokkaasti. Käyttäjä- ja sidosryhmien intressit ja osaamistasot voivat olla hyvin erilaisia, jolloin luotettavan tiedon kerääminen voi osoittautua vaikeaksi. (JUHTA, 2018, s. 18-20).

2.2.1 Vaatimusten kartoittaminen

Vaatimusten kartoitus lähtee alkuperäisen ongelman tai tarpeen analysoinnista. Kartoituksen tarkoituksena on kerätä asiakkaalta ja sidosryhmiltä mahdollisimman paljon tietoa mitä halutaan toteuttaa, miksi, ketkä osallistuu toteutukseen ja millä aikataululla. Nämä tiedot ovat olennaisia, jotta ohjelmistoprojekti saadaan määriteltyä tarkasti ja asiakas saa lopputuloksena mieleisen ja toimivan ohjelmiston (JUHTA, 2018, s. 10-12). Kartoitus voi tapahtua hyväksikäyttäen kyselylomakkeita, suullista kyselyä tai haastattelua tai ryhmätapaamista (Haikalta & Mikkonen, 2011, s. 66). Vaatimukset ja muut toteutuksen kannalta tärkeät tiedot kirjataan alustavaan vaatimusmäärittelydokumenttiin.

2.2.2 Vaatimusten analysointi

Vaatimusten hallinnan tarkoitus on kasata kartoituksen tiedot ja koostaa niistä paketti, joka tyydyttää asiakkaan tarpeet, mutta on myös toteutettavissa käytettävissä olevien resurssien puitteissa. Hallintaan kuuluu vaatimusten analysointi ja priorisointi (Holmala, 2019, s.10).

Vaatimusten analysoinnissa pohditaan vaatimusten oikeellisuutta ja tarkkuutta. Puutteelliset, moniselitteelliset ja virheelliset vaatimukset korjataan tai poistetaan (Lintula, 2004, s. 13). Analysointivaiheeseen osallistuu ohjelmiston kehittäjä ja asiakkaan edustaja, jolloin vaatimukset saadaan analysoitua keskitetysti.

Analysoinnin keskittämiseen liittyy vahvasti tarve priorisoida vaatimukset. Priorisoinnissa pyritään määrittämään vaatimusten tärkeysjärjestys, jolloin projektin koko, resurssit ja toteutusjärjestys pysyvät hallinnassa. Kaikkien käyttäjä- ja sidosryhmien mukaan ottaminen

priorisointivaiheeseen ei olisi missään nimessä järkevää. Jokaisen ryhmän subjektiivinen näkemys omien vaatimustensa tärkeydestä vaikuttaisi heikentävästi vaatimusten priorisoinnin tuloksiin ja lopputuloksena kaikki vaatimukset olisivat yhtä tärkeitä (Holmala, 2019, s.15-16).

2.2.3 Vaatimusten hyväksyminen

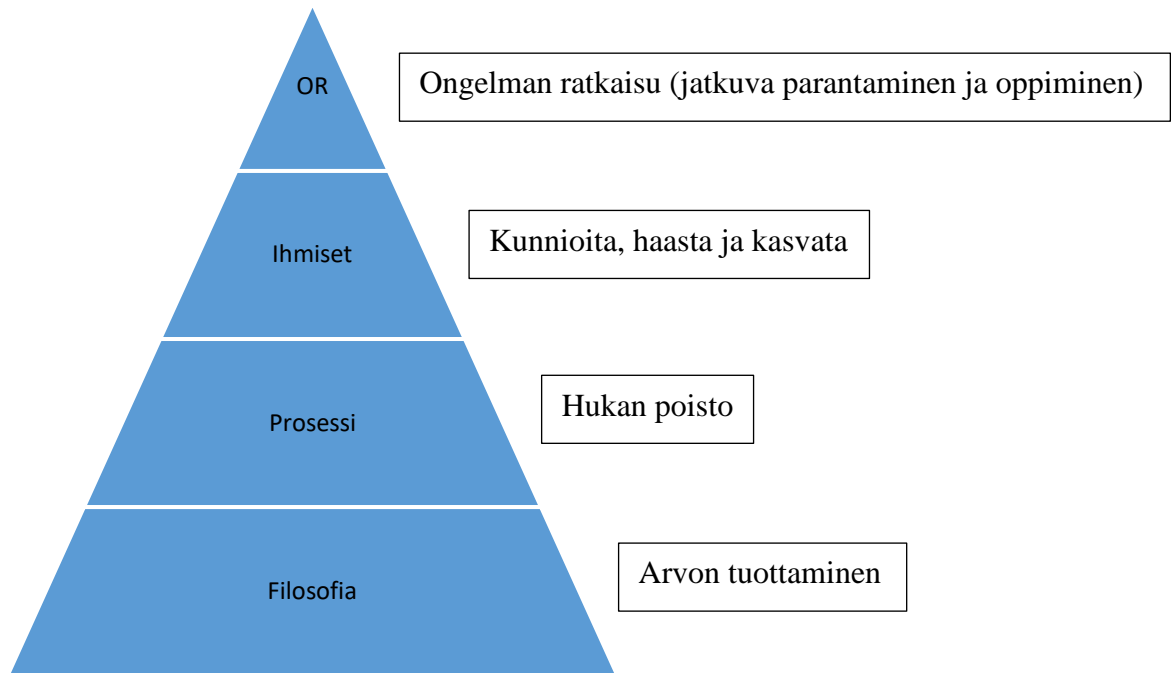
Vaatimusten ja vaatimusmäärittelyndokumentin kokoaa yhteen projektipäällikkö ja lopullisen hyväksymisen suorittaa ohjelman omistaja (Holmala, 2019, s. 16). Hyväksymistä edeltää kaikkein prototyypin, vaatimusdokumenttien tarkka läpikäyminen, mutta kattava taustatyö maksaa itsensä takaisin projektin edetessä vältettyjen virheiden korjauskustannusten kautta (Lintula, 2004, s. 14).

3 LEAN-AJATTELU OHJELMISTOKEHITYKSESSÄ

LEAN-ajattelu ohjelmistokehityksessä on suhteellisen uusi käsite. Ensimmäiset LEAN-ajattelua ohjelmistokehityksen näkökulmasta käsittelevät teokset ilmestyivät 2000-luvun alkupuolella (Lehtinen, 2011, s. 32-33). Tässä kappaleessa käydään läpi LEAN-ajattelu yleisesti, hukat ja arvon tuottaminen.

3.1 LEAN yleisesti

LEAN-ajattelu perustuu autovalmistaja Toyotan toisen maailmansodan jälkeen kehittämään filosofiaan tehostaa tuotantoa poistamalla tuotantoprosessista kaikki osa-alueet, jotka eivät tuota arvoa asiakkaalle ja lisäämällä asiakkaalle arvoa tuottavia asioita (LEAN Enterprise Institute, 2019). Ohjelmistokehityksessä LEAN otettiin käyttöön 90-luvulla tehostamaan ja parantamaan työskentelyn laatua (Rodrigues, s.50, 2013). Arvoa tuottamattomia ja toiminnan kannalta ei-välttämättömiä osa-alueita kutsutaan hukiksi. Tuotantoprosessin tehostamisen onnistumista voidaan mitata hukkiin käytettyjen taloudellisten resurssien ja aika- ja raaka-aineresurssien säästöillä. Puhtaasti taloudellisen hyödyn lisäksi LEAN-ajattelu pyrkii kunnioittamaan työtä tekeviä ihmisiä ja antamaan näille mahdollisuuden vaikuttaa omaan työhönsä (Rodrigues, s.51 2013). LEAN-ajattelun kantavana teemana on jatkuva parantaminen, jossa aiemmin opittua pyritään hyödyntämään. Jeffrey Likerin 4P (engl. Problem solving, People, Process, Philosophy) -malli (Kuva 1. 4P-malli) kuvaa LEAN-ajattelun periaatteita (Lehtinen, 2011, s. 14-16).



Kuva 1. 4P-malli

3.1.1 Hukat ohjelmistokehityksessä

Perinteiseen teollisuuteen suunniteltuun LEAN:iin kuuluu seitsemän hukkaa. Teollisuuden prosessien hukat eivät suoraan käy ohjelmistojen kehitykseen, koska ohjelmistot ja niiden kehitys ovat vaikeasti mitattavia niiden aineettomuuden takia, mutta ovat sovellettavissa (Tyynismaa, s. 38,2014). Matti Lehtinen pro gradututkielman (2011, s. 39-42) ja Deepti Karlan verkkojulkaisun (2017) mukaan Mary ja Tom Poppendeick ovat määritelleet LEAN software development- an agile toolkit -kirjassaan ohjelmistokehityksen vastineet teollisuustuotannon hukille seuraavan taulukon (Taulukko 6. LEAN-hukat) mukaisesti.

Taulukko 6. LEAN-hukat

LEAN hukat teollisuudessa	LEAN hukat ohjelmistokehityksessä
Ylituotanto	Ylimääräiset ominaisuudet
Odottelu ja viivästykset	Odottelu ja viivästykset
Tarpeeton kuljettaminen	Projektien välillä siirtyminen
Ylikäsittely	Ylikäsittely
Tarpeettomat varastot	Keskeneräiset ominaisuudet

Tarpeeton liike työskentelyssä	Tarpeeton projektien siirto
Laatuvirheet	Virheet

Ylituotanto on helposti ymmärrettävissä asiakkaan tarpeisiin ja vaatimuksiin nähden ylimääräisten ominaisuuksien toteuttamisella. Nämä ominaisuudet eivät tuota asiakkaalle arvoa, mutta kuluttavat kehitykseen osallistuvien tahojen aikaa. Aikaan sidottuja hukkia ovat myös odottelu ja viivästykset, jotka voivat johtua esimerkiksi projektin tehottomasta aikataulutuksesta. Ajan ja energian hukkana toimii myös projektien välillä siirtyminen. Tästä esimerkkinä on tilanne, jossa koodareilla on useampi projekti auki samaan aikaan. Toiseen projektiin siirtyminen vaatii ajatusmaailman uudelleen asemoinnin. Ylikäsittelyllä tarkoitetaan ohjelmiston tekemiseen käytettyä ylimääräistä työtä, kuten esimerkiksi saman asian uudelleen opettelu tai samojen ominaisuuksien uudelleen luominen sen sijaan, että olisi hyödynnetty samoja resursseja dynaamisesti (Lehtinen, 2011, s. 39-42).

Keskeneräiset ominaisuudet eivät tuota asiakkaalle arvoa, koska asiakas ei pääse hyödyntämään niitä. Huonoimmassa tapauksessa asiakas on maksanut niiden luomisesta, eikä ole saanut rahoilleen vastinetta toimivan ominaisuuden muodossa. Tarpeetonta projektien siirto tarkoittaa projektien välittämistä henkilöltä tai osastolta toiselle, jolloin siirron yhteydessä saattaa kadota projektin toteutuksen kannalta tärkeää tietoa tai osaamista. Tieto voi olla syvempää tuntemusta projektin perusrakenteesta, jonka hallinta edesauttaa perusrakenteen päälle luotavien uusien ominaisuuksien suunnittelua. Virheet ovat ohjelmistokehityksessä vastaan tulevia ohjelmointi- tai suunnitteluvirheitä. Virheiden määrä on helposti laskettavissa ja ymmärrettävissä suhteessa projektin onnistumiseen asiakkaan näkökulmasta. Laatuvirheiden korjaaminen projektin loppuvaiheessa ja etenkin julkaisun jälkeen on hankalaa ja kallista (Lehtinen, 2011, s. 39-42).

3.1.2 Arvon tuottaminen ohjelmistokehityksessä

LEAN-ajattelun toinen tärkeä pilari on arvon tuottaminen asiakkaalle. Arvoa asiakkaalle tuottaa käyttäjälähtöinen kehittäminen, kehityksen läpinäkyvyys, ketterä julkaiseminen, jatkuva parantaminen ja itseohjautuvuus (Luoto, 2019) (Tyynismaa s.44-45, 2014).

Käyttäjälähtöinen kehitys tarkoittaa asiakkaan tarpeen tiedostamista, konkreettisen ratkaisuvaihtoehdon luomista, testausta ja johtopäätösten tekemistä. Käyttäjälähtöinen kehitys on itseään ruokkiva iteroituva prosessi ja se on sovellettavissa kaikkiin ohjelmistoprojektin vaiheisiin. Prosessi tukee myös kehityksen läpinäkyvyyttä asiakkaalle päin. Kehityksen läpinäkyvyys antaa asiakkaalle selkeämmän kuvan projektin etenemisestä ja luo arvoa, kun asiakas kokee saavansa rahoilleen vastinetta helposti havainnoitavan kehityksen muodossa (Luoto, 2019).

Jatkuva parantaminen on pidemmän aikavälin toimintatapa, joka porautuu ennalta priorisoitujen ongelmien ratkaisuun hyödyntäen aiemmin opittua. Jatkuvan parantamisen periaatteet soveltavat aiemmin opittua yhteisinä toimintatapoina. Yhteiset toimintatavat vähentävät uudelleenopettelua ja luovat pohjan, joka toimii lähtötasona uusia ongelmia ratkottaessa (Luoto, 2019).

Itseohjautuvuus LEAN-ajattelutavassa tarkoittaa työntekijän muuntautumista suorittajasta toteuttajaksi. Toteuttajan rooli pitää sisällään tietoisuuden omasta työstä, valmiuden paikalliseen ongelmanratkaisuun ja sitä myöten prosessin kehitykseen, oman osaamisen kehittämisen ja jakamisen työympäristössä ja ymmärryksen toiminnasta yleisemmällä tasolla (Luoto, 2019).

Pilar Rodrigues kertoo väitöskirjassaan ohjelmistoprojektin asiakkaalle tuottaman arvon kertymisen jatkuvan tavallisen teollisuuden tuotetta kauemmin (Rodrigues, 2013, s. 48). Syynä tähän ovat muun muassa ohjelmiston korjaus, päivittäminen ja lisäomaisuuksien luonti ylläpitovaiheessa.

3.1.3 LEAN-ajattelun toteutus

Tuloksia LEAN-ajattelun hyödyntämisestä ohjelmistokehityksessä ei vielä ole laajasti tutkittu ajattelutavan suhteellisen tuoreen käyttöönoton vuoksi. Myös ajattelutavan moninaiset tulkinnat ja sekoittaminen ketterän ohjelmistokehittämisen työkaluihin ja niiden luomaan hyötyyn vaikeuttavat LEAN-ajattelun konkreettisten hyötyjen mittaamista (Rodrigues, s. 55 - 58 2013).

Tyynismaa (s.45-47, 2014) esittää opinnäytetyössään Oulun yliopiston julkaisemia lukuja LEAN:n ja ketterän kehittämisen yhdistämisen hyödyistä. Liiketoiminnan kannattavuuden kasvu 30 %, läpimenoajan nopeutuminen 30-50 % ja asiakkaan kokeman laadun parantuminen 50 % kertoo selkeästä parannuksesta. Tyynismaa nostaa kuitenkin esille erittäin tärkeän puolen LEAN-ajattelutavan onnistuneeseen hyödyntämiseen ohjelmistokehityksessä. Asiakkaalla on suuri rooli projektin etenemisessä koko sen eliniän ajan. Tämä vaatii asiakkaalta osaamista, sitoutumista ja jaksamista. Asiakkaalla on oltava teknistä tietämystä ja osaamista, jotta hän voi esittää ominaisuusvaatimuksia ja antaa palautetta ratkaisumalleista tai prototyypeistä. Projektin kesto ja prototyyppien julkaisutahti voivat myös olla haaste asiakkaan sitoutumiskyvyn ja jaksamisen kannalta. Suoranaista ratkaisua tähän ei ole, mutta kehityksen läpinäkyvyys, selkeä aikataulu ja asiakkaan koulutus LEAN-ajatteluun helpottaa asiakkaan sopeutumista.

4 VAATIMUSMÄÄRITTELYN TOTEUTUS KOHDEYRITYKSESSÄ

Ohjelmistoprojekti sai alkunsa kohdeyrityksen tarpeesta pyrkiä poistamaan paperiset raportointilomakkeet laadun arviointiprosessista ja saada käyttöön informatiivisempi ja visuaalisempi tapa esittää kerätty data. Vaatimusmäärittelyn kannalta nämä olivatkin ensimmäiset ja prioriteeteissa tärkeimmät vaatimukset. Vaatimusmäärittely toteutettiin yrityksessä maaliskuu-toukokuussa 2019. Tässä kappaleessa käydään läpi vaatimusmäärittelyn vaiheet vaatimusten kartoituksesta vaatimusten analysointiin ja hyväksyntään.

4.1 Vaatimusten kartoitus

Vaatimusten kartoitus tapahtui yhdessä kohdeyrityksen laatuvaikuttajan kanssa. Kartoitettavien vaatimusten aiheet perustuivat ohjelmistotuotanto-kurssin projektissa esiin nousseisiin vaatimuksiin (Savolainen, 2019). Kartoitusvaiheessa työntekijöillä ei ollut omaa erillistä edustajaa, koska työskentelin tuolloin itse yrityksessä työntekijänä. Katsoimme myös, että työntekijät pääsevät ilmaisemaan omat mielipiteensä ja ideansa projektille alustavan tyytyväisyyskyselyn kautta. Tyytyväisyyskyselyt suoritettiin käyttäen kvantitatiivista tutkimusmenetelmää Likertin-asteikon avulla. Likertin asteikko järjestää numeeriset vastausvaihtoehdot 4-5 portaiseen asteikkoon, jonka ääripäissä ovat vastakkaiset adjektiivit (Heikkilä, 2014, s. 15-16, 51). Kyselyt toteutettiin Google Forms-lomakkeiden avulla internetkyselynä, jotta kyselyyn vastaaminen olisi nopeaa, tulokset olisivat heti saatavilla ja haastattelijaa ei tarvittaisi (Heikkilä, 2014, s. 66-67). Saatekirje kyselyyn julkaistiin yhdessä kyselyn kanssa, jotta työntekijät saataisiin vastaamaan (Vehkalahti, 2008, s. 47-48) (Liite 4. Saatekirje).

Tyytyväisyyskysely esitettiin sekä työntekijöille ja esimiehille. Kysely jaettiin kahteen eri osaan johtuen sidosryhmien ohjelmistolle asettamista eri käyttötarpeista ja -tavoista. Työntekijöiden tyytyväisyyskysely painottui laaturaportoinnin käytännön osuuteen. Esimiesten tyytyväisyyskysely jakaantui kvantitatiivisiin kysymyksiin aamupalavereista ja kvalitatiivisiin kysymyksiin laatu- ja hyödyntämisestä työssä (Heikkilä, 2014, s. 15-16). Kyselystä saatua dataa käsiteltiin ryhmien mukaan kahtena erillisenä osana.

Työntekijöiden kysely piti sisällään seuraavat kysymykset:

- Kuinka selkeäksi arvioisit nykyisen tuotekatselmuksen? (1-5)
- Kuinka helposti löydät kaikki tuotekatselmukseen tarvittavat asiat? (1-5)
- Kuinka tyytyväinen olet nykyiseen paperiseen tuotekatselmukslomakkeeseen? (1-5)
- Kuinka helpoksi koet laatupisteiden syöttämisen tietojärjestelmään? (1-5)
- Kuinka paljon aikaa arvioisit käyttäväsi kokonaisuudessaan tuotekatselmuksen tekemiseen? (5min - 20min tai enemmän)
- Kehitysideoita tuotekatselmuksen sähköiseen versioon? (sanallinen)
- Kuinka tyytyväinen olet aamupalaverien laatuosioon? (1-5)
- Kuinka paljon arvioisit aamupalaverien laatuosion vaikuttavan laatutietouteen ja käyttäytymiseen työssäsi? (sanallinen)
- Kehitysideoita aamupalaverin laatuosioon tai aamupalaveriin yleensä? (sanallinen)

Esimiehille suunnatussa tyytyväisyyskyselyssä kysyttiin seuraavasti:

- Kuinka tyytyväinen olet aamupalavereihin? (1-5)
- Kuinka tyytyväinen olet laatuosioon aamupalaverissä? (1-5)
- Kuinka selkeäksi arvioisit nykyisen laatupisteiden esityksen? (1-5)
- Mitä laatuun liittyviä asioita haluaisit sisältyvän aamupalaveriesitykseen? (sanallinen)
- Hyödynnätkö nykyisessä työssäsi tuotekatselmuksista saatavaa laatudataa? (monivalintakysymys)
 - o Vaihtoehto: Ei
 - Miksi et hyödynnä? (sanallinen)
 - Mitä tietoa laatudatan pitäisi sisältää, jotta voisit hyödyntää sitä työssäsi? (sanallinen)
 - o Vaihtoehto: Kyllä
 - Miten hyödynnät laatudataa työssäsi? (sanallinen)
 - Mitä tietoa laatudatan pitäisi sisältää, jotta voisit hyödyntää sitä paremmin työssäsi? (sanallinen)

Kyselyillä pyrittiin rakentamaan kattava määrällisesti mitattavissa ja analysoitavissa oleva kokonaiskuva nykyisestä tilanteesta. Vastausvaihtoehdoissa numeroiden lisäksi oli Likert-asteikon mukaisesti arviointia selkeyttämässä termit ”Erittäin selkeä/helppo/tyytyväinen” tai

”Erittäin epäselvä/vaikea/tyytymätön”) (Heikkilä, 2014, s. 15-16, 51). Esimiesten kysymyksissä ei kysytty tuotekatselmuksen helppoudesta tai selkeydestä, koska esimiehet eivät itse tee tuotekatselmuksia tai käytä raportointityökalua, pois lukien laatuvaastaava.

Aloituspalaveri järjestettiin maaliskuun alussa 2019. Aloituspalaverissa kävimme läpi vaatimusmäärittelyn rakennetta ja kirjasimme vaatimusmäärittelydokumenttiin alustavia vaatimuksia (Liite 1. Vaatimukset), rajoitteita (Liite 2. Rajoitteet) ja käyttäjä- ja sidosryhmiä (Liite 3. Käyttäjä ja sidosryhmät). Vaatimukset olivat pääsääntöisesti toiminnallisia vaatimuksia eli mitä kaikkea ohjelmalla tulisi pystyä tekemään. Alustavat toiminnalliset vaatimukset pitivät muun muassa sisällään vaatimuksen tiedon syöttämisestä graafisen käyttöliittymän avulla ja tallentamisesta Excel-tietokantaan, vaatimuksen tallennetun tiedon yksinkertaisesta esityspohjasta ja mahdollisuuden tulostaa raportteja tiedostoiksi. Ei-toiminnallisia alustavia vaatimuksia olivat paperittomuus, helppokäyttöisyys ja tietoturvan huomioiminen.

Toisessa vaatimusmäärittelypalaverissa maaliskuun lopussa 2019 kirjasimme lisää vaatimuksia ja tarkensimme edellisellä kerralla kirjattuja vaatimuksia ja rajoitteita. Palaverissa oli käytössä myös käyttöliittymäprototyyppi helpottamassa ominaisuuksien ja rajoitteiden hahmottamista ja vaatimusten luontia. Kolmanteen palaveriin huhtikuussa osallistui myös yksikön päällikkö, joka toi esiin omat vaatimuksensa, ehdotuksensa ja mielipiteensä ohjelmiston käytöstä ja mahdollisesta laajennettavuudesta yksikön ja organisaation sisällä.

4.2 Vaatimusten analysointi

Vaatimusten analysointipalaveri toteutettiin yhdessä laatuvaastaavan ja yksikön päällikön kanssa. Analysointivaiheen palavereissa kävimme läpi työntekijöiden ja esimiesten ensimmäisessä tyytyväisyyskyselyissä nousseet palautteet ja ideat. Laatuvaastaavan kanssa keskityimme tarkentamaan olemassa olevia vaatimuksia (Liite 1. Vaatimukset) ja selvittämään mahdollisia rajoitteita (Liite 2. Rajoitteet). Analysointivaiheessa syntyi myös uusia vaatimuksia, joista osa oli kokonaan uusia toiminnallisia ominaisuuksia ja osa aiemman ei-toiminnallisen vaatimuksen korvaavia vaatimuksia.

Osana analysointivaiheen priorisointiosuutta toteutimme vaatimusten priorisoinnin neliportaisen asteikon mukaan. Asteikon portaat olivat P(pakollinen), 1(Tärkeä), 2(ei välttämätön), 3(hyvä lisä). P-prioriteetin saaneet vaatimukset olivat pääsääntöisesti toiminnallisia vaatimuksia, jotka asiakkaan oli helppo nostaa esille. 1-prioriteetin saaneet vaatimukset olivat ei-toiminnallisia, mutta käytön kannalta tärkeitä ominaisuuksia kuvaavia vaatimuksia. 2- ja 3-prioriteetin vaatimukset olivat ominaisuuksia, jotka eivät olleet pakollisia, ja jotka toteutettaisiin ajan salliessa. Hylkäsimme vaatimuksia (Liite 5. Hylätyt vaatimukset), jotka eivät olleet toteutettavissa joko teknisistä tai ajallisista rajoitteista johtuen.

4.3 Vaatimusten hyväksyntä

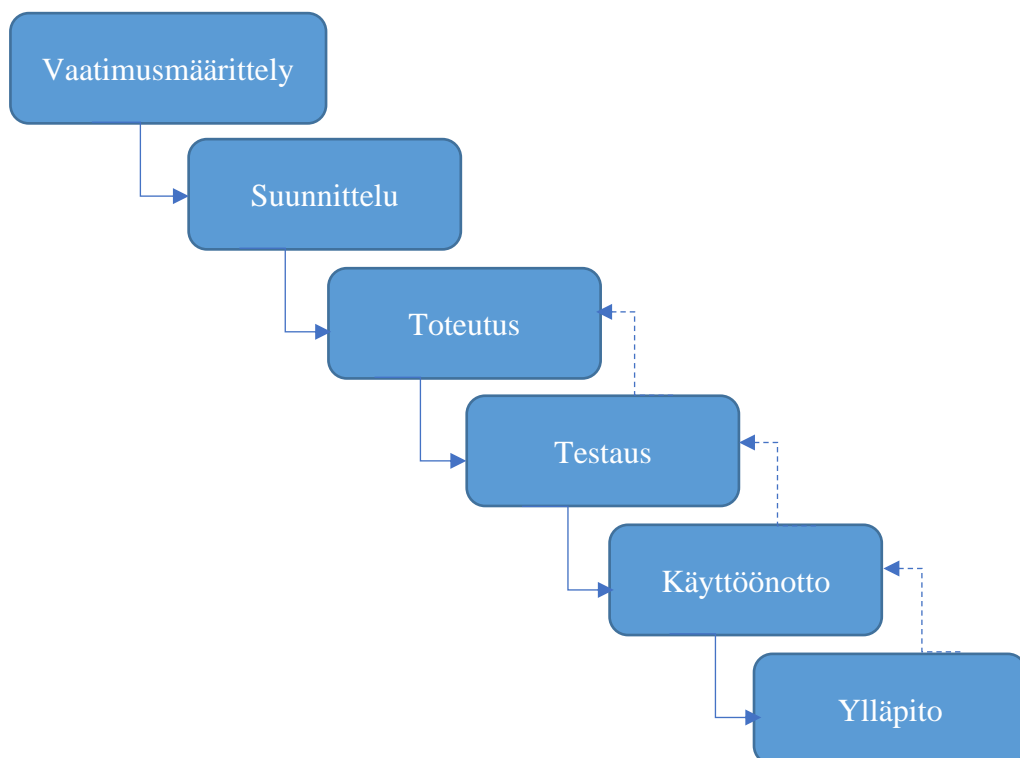
Vaatimusten hyväksyntä tapahtui samassa yhteydessä analysoinnin kanssa. Yksikön päällikkö hyväksyi ohjelman vaatimukset ja toteutussuunnitelman ja oli optimisten ohjelman käyttöarvon suhteen tulevaisuudessa. Projektin pienen koon ja vähäisten käyttäjä- ja sidosryhmien vuoksi vaatimusten hyväksyntä ei vaatinut omaa palaveriaan tai laaja-alaista valmistelua. Ohjelma on suunniteltu toimimaan itsenäisesti, eikä se integroidu jo olemassa olevaan järjestelmään, joten eri järjestelmärajapintojen tutkiminen ja dokumentointi ei ollut tarpeellista.

5 OHJELMISTON TOTEUTUS KOHDEYRITYKSESSÄ

Ohjelmistoa lähdettiin toteuttamaan vaatimusmäärittelyn pohjalta toukokuussa 2019. Tärkeimmät kappaleessa esille tulevat asiat toteutusvaiheista ovat vaatimusmäärittelyn vaikutus kaikkiin vaiheisiin ja perinteisen ohjelmistokehityksen vaihekuvausmallin systemaattisen toteutuksen positiivinen vaikutus ohjelmiston valmistumiseen. Toisaalta LEAN-ajattelun soveltamisen osalta on hyvä mainita, että sen anti tälle työlle tuli esiin erityisesti testaus- ja käyttöönottovaiheissa, jossa interaktio asiakkaan kanssa oli tiuhaa.

5.1 Ohjelmiston toteutuksen vaiheet

Ohjelmiston toteutuksessa käytettiin Winston W. Roycen vuonna 1970 esittelemää mallia mukaillen ohjelmistokehityksen vaiheita perinteisesti kuvaavaa vesiputousmallia (Kuva 1. Vesiputousmalli). Vaatimusmäärittelyn jälkeen ensimmäisenä työvaiheena oli ohjelmiston rakenteen suunnittelu, toisena koodaus, kolmantena testaus, neljäntenä käyttöönotto ja viidentenä ylläpito (Haikala & Mikkonen, 2011, s.36-38). Poiketen perinteisestä vesiputousmallista neljä viimeistä vaihetta olivat LEAN-ajattelun mukaisesti iteratiivisia.

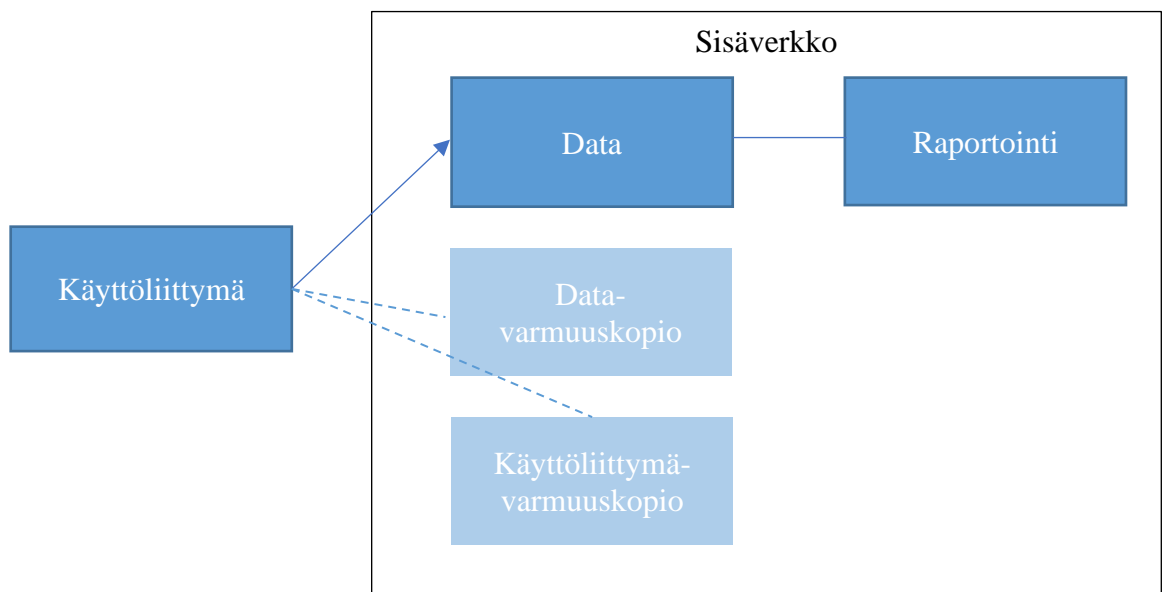


Kuva 2. Vesiputousmalli

5.1.1 Suunnittelu

Ohjelmiston ja samalla koko suunnitteluvaiheen raamit määräytyivät vaatimusmäärittelyn pohjalta. Kohdeyrityksen vaatimusmäärittelyvaiheessa tekemät rajoitteet velvoittivat ohjelmiston toteutettavaksi Microsoft Excel-käyttöympäristöön ja ohjelmointikieleksi VBA:n (Visual Basic for Applications). Yrityksen käytössä olevan laadunpisteytysjärjestelmän integrointi ohjelmistoon oli pakollista ja ehdottoman tärkeää, koska pistelaskukaavojen muutos olisi tehnyt vanhojen ja uusien laatupisteiden vertailusta mahdotonta.

Suunnittelussa oli ehdottomasti otettava huomioon tietoturva, eli oli luotava järjestelmä, jossa käyttäjä ei pääse käsittelemään tietokannassa olevaa dataa. Käyttäjä ei myöskään saa vahingossa tai tahallisesti päästä muuttamaan mitään ohjelmiston tiedostoista. Tietoturvaominaisuudet toteutettiin eristämällä käyttäjälle näkyvä käyttöliittymä ja kaikki käyttäjän kannalta olennainen tieto yhteen tiedostoon. Ohjelma varmistaa aina käynnistyksen yhteydessä varmuuskopioiden olemassaolon (kuva 2. Tiedostorakenne). Tiedostorakenne pitää sisällään kolme tiedostoa. Käyttöliittymätiedosto on asennettu kannettavalle tietokoneelle. Data- ja raportointitiedostot on eristetty käyttäjältä piiloon sisäverkkolevylle.



Kuva 3. Tiedostorakenne

Käyttöliittymätiedosto pitää sisällään graafisen käyttöliittymän laadun syöttämiseen, tiedon olemassa olevista tuotelinjoista ja tuotteista, makroja, joilla tiedot saadaan lähetettyä datatiedostoon ja tiedon varmuuskopiointitiedostojen sijainnista. Datatiedostossa on kaikki käyttäjien syöttämä data ja datan analysointi- ja esitysokalut yrityksen esimiehille. Raportointitiedostossa on makroja, jotka hakevat datatiedostosta laadun dataa ja muodostavat siitä laatuvaikuttavalle olennaisia tunnuslukuja ja graafeja ja mahdollistaa niiden tulostuksen PDF (Portable Document Format) -tiedostoon.

Olellisena osana suunnittelua oli tuote- ja tuotantolinjakohtaisten ominaisuuksien ja asetusten määrittely. Ohjelmiston oli kyettävä ottamaan vastaan parikymmentä erilaista tuotetta, joiden mitattavat laatuominaisuudet poikkesivat toisistaan huomattavasti. Lisäksi ohjelmisto täytyi olla laajennettavissa ilman tarvetta koskea koodiin. Käytännössä tämä tarkoitti dynaamisten tuotantolinja- ja tuotelistojen sekä neljänkymmenen muokattavissa olevan ominaisuusvaihtoehdon luomista ja niiden dynaamista esitystä käyttöliittymässä. Vaatimusmäärittelyssä tärkeänä toiminnallisena ominaisuutena esiin noussut mahdollisuus tuotepoikkeamien kirjaukseen täytyi myös toteuttaa osittain esitetyin vaihtoehdoin ja osittain käyttäjän vapaana kuvailuna.

5.1.2 Toteutus

Ohjelmiston toteutuksen voidaan katsoa alkaneeksi jo vaatimusmäärittelyvaiheessa käyttöliittymäprototyypin tekemisellä. Prototyypin teko auttoi huomattavasti haluttujen ominaisuuksien ja toimintojen toteuttamista ja kokonaisuuden hahmottamista. Ohjelmiston koodaus alkoi suunnitelman mukaisesti kesäkuun alussa 2019, mutta alkuperäisistä suunnitelmista poiketen toteutusta ei tehty yrityksen tiloissa yhteistyössä laatuvaikuttavan kanssa johtuen muutoksista työsopimuksissa vaan etätyöskentelyä kotoa käsin. Etätyöskentelyn mahdollisti hyvä keskusteluyhteys laatuvaikuttavan kanssa sähköposteilla.

Käytännön toteutus jakautui viiteen vaiheeseen:

- käyttöliittymän toteutus
- asetusten toteutus
- tiedostorakenteen toteutus

- datatiedoston toteutus
- raportointitiedoston toteutus

Henkilökohtaisen preferenssin takia käyttöliittymä toteutettiin ensimmäisenä rinnan asetusten kanssa. Intuitiivisen käyttäjäkokemuksen saavuttamiseksi käyttöliittymästä haluttiin mahdollisimman yksinkertainen ja dynaaminen. Dynaamisen käyttöliittymästä tekee tarpeen mukaan esiin tulevat valintavaihtoehdot, joita voi säätää ohjelman asetuksista. Käyttäjälle haluttiin näyttää mahdollisimman vähän vaihtoehtoja, jolloin ohjelman käyttö olisi nopeampaa. Käyttöliittymäpohjaksi valikoitui VBA:n käyttäjälomake (userform), joka mahdollisti yksinkertaisen nappuloiden, pudotusvalikoiden ja tekstikenttien luomisen. VBA-ohjelmointikieli ei ollut ennestään tuttu, joten tutustuin Tutorials Pointin verkkosivustoon, josta löytyy muun muassa perustietoa VBA-funktioista ja tietotyypeistä (Tutorials point,2019). Käyttäjälomakkeen standardi Windows-ympäristöstä tuttu ulkonäkö koettiin eduksi. Käyttöliittymä jakautuu kahdeksaan segmenttiin (Kuva 3. Käyttöliittymä), joista ensimmäiseen käyttäjä lisää nimensä ja valitsee halutun päivän, tuotantolinjan sekä tuotteen. Kuudessa seuraavassa segmentissä on tuotekohtaisten laatuominaisuuksien pisteytys- tai poikkeamavaihtoehtoja. Viimeisessä segmentissä näkyy kokonaispisteet ja

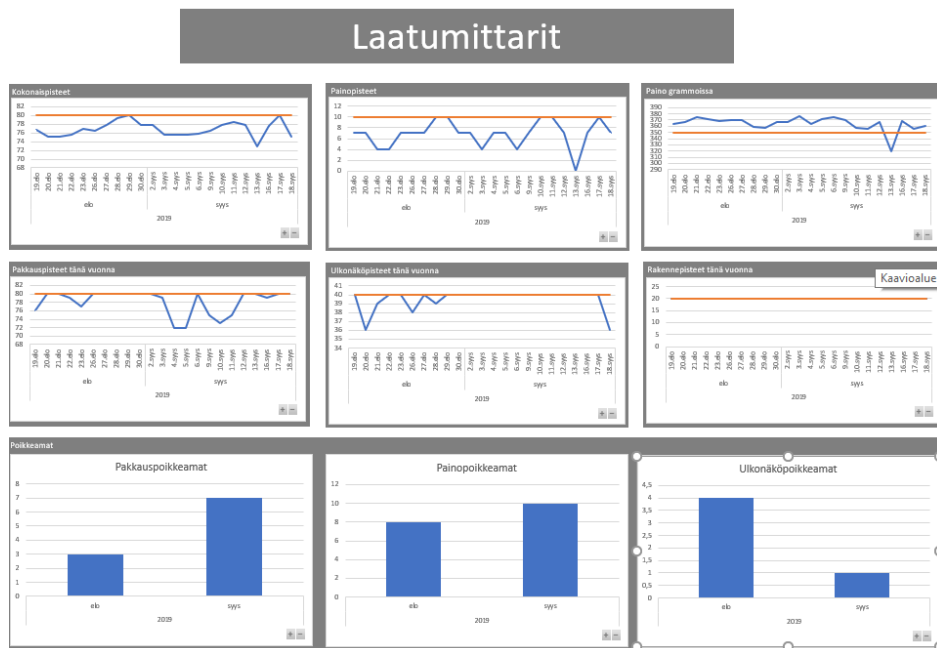
Kuva 4. Käyttöliittymä

vaihtoehdot käyttäjälomakkeen tyhjentämiseksi ja tallentamiseksi. Osa grafiikasta ja ominaisuuksista on poistettu kuvasta yritysalaisuuksien kunnioittamisen vuoksi.

Käyttöliittymän toimintaan olennaisesti liittyvät tuote- ja linjakohtaiset asetukset toteutettiin luomalla Excel-taulukot, joista toiseen tuli tieto linjoista ja mitä tuotteita linjoilla on ja toiseen tuotteen ja mitä laatuominaisuuksia ja maksimilaatupisteitä tuotteilla on. Käyttöliittymässä olevat pudotusvalikot etsivät Excelin VLOOKUP-funktioiden avulla halutut vaihtoehdot linjoille, tuotteille, tuotteiden ominaisuuksien pisteille ja poikkeamille.

Tiedostorakenteen toteutusvaihe keskittyi datan siirron toteuttamiseen kolmen tiedoston välillä Excel-makroja ja VBA-funktioita hyödyntäen. Käyttöliittymän tallennusnappulan painaminen käynnistää funktion, joka kerää aikaleiman, eri laatusegmenttien pisteet, tuotekohtaiset maksimipisteet ja mahdolliset poikkeamat ja lähettää ne datatiedostossa olevaan tietokantaan. Raportointitiedosto hakee datatiedoston tietokannasta laatuun liittyvät numeeriset arvot ja poikkeamat ja luo niistä keskiarvotaulukoita.

Tietokannan kanssa samassa datatiedostossa on viikko-, kuukausi- vuosikohtaiset ja vapaasti ajan suhteen muokattavissa graafiset esityspohjat (Kuva 4. Esityspohja). Esityspohjissa hyödynnetään Excelin pivot-taulukoihin kuuluvia dynaamisia työkaluja, kuten aikajana, kuvaaja ja suodatin. Suodattimien avulla esityspohjia käyttävä esimies voi helposti vaihtaa



Kuva 5. Esityspohja

laatupiste- ja poikkeamakertymäkuvaajia tuotekohtaisesti. Kuvaajissa pyrittiin mahdollisimman yksinkertaiseen ja informatiiviseen esitykseen (Heikkilä, 2014, s. 148). Raportointitiedosto ja sen sisältämät toiminnallisuudet olivat yksi ei-pakollisista, mutta tärkeistä toiminnallisista ominaisuuksista, joita haluttiin toteutettavan. Raportointitiedoston saamat tiedot datatiedostosta esitetään linja- ja tuotekohtaisesti kokonaispistekeskiaarvotaulukkona halutulta ajanjaksolta. Tuotekohtainen kokonaispistekeskiaarvotaulukko pitää sisällään myös tiedon mahdollista poikkeamista. Taulukon siistimiseksi ja epäolennaisen tiedon karsimiseksi työkirja pitää sisällään automaattisen makron, joka piilottaa kaikki taulukon tyhjät poikkeamarivit, koska Excelin pivot-tilit eivät osaa piilottaa tyhjiä tietosoluja automaattisesti. Raportointitiedosto pitää sisällään Excel-makron, joka tulostaa käyttäjän valitseman tai ennalta määritellyn alueen PDF-tiedostoksi, jonka VBA-funktiot automaattisesti nimeävät päivän ja raporttityypin mukaisesti.

5.1.3 Testaus

Testausvaihe sijoittui heinä-elokuun vaihteeseen 2019. Testaukseen osallistui laatuvaikuttajan lisäksi yksi työntekijä, joka syötti järjestelmään tietoja testaten ohjelmiston perustoimintoja ja antoi yleisesti palautetta käytettävyydestä tavallisen käyttäjän näkökulmasta. Päävastuun testauksesta ja korjaus-/parannusehdotuksista otti laatuvaikuttaja, joka oman työnsä ohessa testasi ohjelmiston määrittelyssä vaadittuja ominaisuuksia. Testauksen aikana pidettiin yksi palaveri, jossa julkaistiin ohjelmiston ensimmäinen kaikki vaaditut ominaisuudet sisältävä testiversio. Loput testiversiot julkaistiin sähköpostilla, jolloin eri testiversioiden julkaisusykli saatiin lyhyemmäksi. Testausvaiheen lopussa julkaistiin yksinkertaiset käyttöohjeet työntekijöille ja laajemmat käyttö- ja asennusohjeet laatuvaikuttajalle. Myös käyttö- ja asennusohjeet tarkastettiin ja niiden sisältämät asiavirheet korjattiin.

Normaalien koodausvirheiden lisäksi testausvaiheessa korjattiin yksi kriittinen suunnitteluvirhe, jossa Excel antoi virheilmoituksen päällekkäisten taulukoiden vuoksi. Käytännössä tämä tarkoitti, että taulukot olivat fyysisesti liian lähellä toisiaan ja tietokannan kasvaessa ne eivät pystyneet kasvattamaan itseään. Ratkaisuna oli jokaisen taulukon sijoittaminen omalle välilehdelleen, jolloin ne voivat kasvaa lähes rajatta.

Testausvaiheessa tuli uusia vaatimuksia ohjelmiston ominaisuuksien suhteen. Uusien vaatimusten pohjalta toteutettiin ohjelmiston käytettävyyttä parantavat automaattisesti päivittyvät taulukot, kuvaajat ja pdf-raportin otsikkotiedot, käyttäjän kannalta epäolennaisten välilehtien piilotus ja työkirjojen suojaus salasanoilla.

5.1.4 Käyttöönotto

Lopullinen versio ohjelmistosta julkaistiin elokuun puolivälissä 2019. Julkaisupalaverissa käytiin läpi viimeiset korjaukset, käytiin läpi vaatimusmäärittelyn toiminnalliset vaatimukset ja todettiin ohjelman toimivan tavoitteiden mukaisesti. Ohjelmiston varsinaista käyttöönottoa hidasti tuotannolliset syyt, jotka aiheuttivat muutoksia ohjelmiston tietokantoihin ja asetuksiin. Tästä syystä ohjelmisto otettiin käyttöön tuotannossa pari viikkoa julkaisun jälkeen.

5.1.5 Ylläpito

Ylläpitovaiheessa ohjelmiston perusversioon tuli enää pieniä graafisia muutoksia käyttöliittymään, taulukoihin ja kuvaajiin sekä pieniä korjauksia raportointitiedoston tulostusalueisiin. Normaalisti ylläpitovaiheesta poiketen ohjelmistosta eriytettiin toinen versio, joka joutui uudelleen testausvaiheeseen. Toinen versio luotiin, koska yrityksen sisällä haluttiin selvittää mahdollisuus ohjelmiston hyödyntämiseen organisaatiossa.

Ylläpitovaiheessa julkaistiin toinen tyytyväisyyskysely, jonka tarkoitus oli kerätä palautetta ohjelmiston toimivuudesta sekä työntekijöiltä, että esimiehiltä. Kysely vastasi rakenteeltaan suurilta osin ensimmäistä kyselyä, mutta siitä puuttui kysymykset ideoista liittyen ohjelmiston toteutukseen ja tavoista laadutatan hyödyntämiseen työssä.

5.1.6 Jatkokehitys

Yksi ohjelmiston alkuperäisistä vaatimuksista oli tehdä ohjelmistosta helposti laajennettava ja muokattava. Syynä vaatimuksen olemassa oloon oli tahto toteuttaa LEAN-ajattelun jatkuvan parantamisen mallia mahdollistamalla saman ohjelmiston käyttö yrityksen muissa toimipisteissä. Jotta ohjelmisto palvelisi tarkoitustaan koko organisaatiossa sitä tulisi

jatkokehittää. Keskusteluissa jatkokehityskohteiksi nousi graafisen käyttöliittymän luominen asetusten hallintaan, monikäyttäjätuki pilvipalvelun kautta, usean tietokannan datan kokoava raportointi, parempi tiedostojen suojaus ja laaja rasiustestaus suurella tietokannalla.

5.2 LEAN-ajattelun toteutuminen

Kohdeyrityksessä on vuosien ajan ollut käytössä LEAN-ajatteluun pohjautuva tuotannon kehitysohjelma, joten LEAN-ajattelua hyödyntävän sovelluskehitysprojektin toteuttaminen ei tuottanut ongelmia. Tässä kappaleessa käydään läpi hukkien poisto ja arvon tuottaminen sovelluskehityksen kannalta kohdeyrityksessä.

5.2.1 Hukkien poisto

LEAN-ajattelun seitsemästä hukasta ohjelmiston kehityksessä esiintyi viittä. Ainoat poissaolevat hukat olivat tarpeeton projektin siirto ja keskeneräiset ominaisuudet. Tarpeeton projektin siirto -hukan puuttuminen selittyy ohjelmiston pienellä koolla ja sillä, että toteuttajia oli yksi eikä projektia tarvinnut siirtää toiselle ihmiselle. Keskeneräiset ominaisuudet- hukan poissaolo selittyy myös osittain projektin pienellä koolla, mutta myös selkeillä ja hyvin tehdyillä ja rajatuilla vaatimusmäärittelyn vaatimuksilla.

Ohjelmiston kehityksessä esiintyi laatuvirheitä, ylikäsittelyä, ylituotantoa, projektien välillä siirtymistä, odottelua ja viivästyksiä. Laatuvirheiden minimoiseksi testausvaihe kesti kuukauden ja sen aikana julkaistiin uusi korjattu versio kerran viikossa. Asiakkaan kannalta laatuvirheiden määrää tärkeämpää oli nopea vastaus virheraporttiin ja korjausehdotuksen tai -version julkaisu. Ylituotantoa ohjelmiston kehityksessä tapahtui paljon. Suurimmat ylituotannon paikat olivat lopullisesta versiosta karsitut toiminnalliset vaatimukset, kuten oppiva poikkeamatietokanta, ja dynaamisten Excel-kuvaajien turhan pikkutarkka säätäminen.

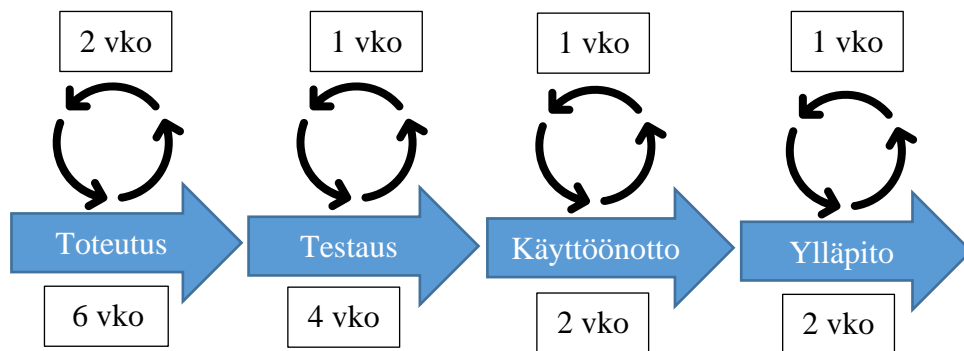
Aikasidonnaisista hukista ylikäsittelyä ohjelmiston kehityksessä tapahtui samojen asioiden uudelleenopettelu muodossa, koska aikataulullisten haasteiden vuoksi ohjelmisto

toteutettiin alkuperäisestä suunnitelmassa poiketen lyhyissä pätkissä toisen työn rinnalla. Samoista syistä tapahtui hukista seuraavaa eli odottelua ja viivästyksiä. Molempiin edellisistä hukista ratkaisuna oli avoin keskustelu kohdeyrityksen laatuvaastaavan kanssa ja yhteisten aikataulujen muokkaaminen palvelemaan molempien tarpeita. Laatuvaastavalla oli projektin aikana kesäloma, joka lykkäsi toteutusvaiheen julkaisuja kolmella viikolla.

5.2.2 Arvon tuottaminen

LEAN-ajattelun mukaisesti koko ohjelmistoprojektin tavoite oli luoda tuote, joka tuottaa asiakkaalle arvoa. Konkreettisen arvon tuotto tässä tapauksessa tarkoitti ajallisen ja laadullisen hukien poistoa laaturaportointiprosessissa ja rahallista säästöä, joka syntyy, kun paperinen raportointi korvataan digitaalisella raportoinnilla. Arvon tuottamisen voidaan katsoa alkaneeksi jo vaatimusmäärittelyvaiheessa, jossa ohjelmisto ja sen kehitys pyrittiin raamittamaan asiakkaan ja käyttäjän tarpeiden mukaiseksi.

Tiivis ja toimiva yhteistyö yrityksen laatuvaastaavan kanssa mahdollisti arvoa tuottavan käyttäjälähtöisen ja läpinäkyvän sovelluskehityksen. Keskustelua sovelluskehityksen vaiheista ja tilasta käytiin laatuvaastaavan kanssa palavereissa ja sähköposteilla. Käytännössä testausvaiheessa ilmenneiden ongelmien ratkaisussa uusi ohjelmistoversio tai korjaus oli valmis saman tai seuraavan päivän aikana. Jos ongelma oli laajempi, niin uusi korjattu versio oli saatavilla seuraavan viikon alussa. Ketterään kehitykseen kiinteästi liittyvä tiivis uusien ohjelmistoversioiden julkaisusykli loi asiakkaalle arvoa ja selkeän kuvan ohjelmistoprojektin etenemisestä. Uusien ohjelmistoversioiden julkaisusykli vaihteli läpi projektin (kuva 5. Uuden version julkaisusykli). Toteutusvaiheessa uusi versio julkaistiin kahden viikon välein, testaus- ja käyttöönotto vaiheessa julkaisusykli oli minimissään yhden kerran viikossa. Ylläpitovaiheen poikkeuksellisen tiiviin julkaisusyklin selittää muutostyöt, joita ohjelmistoon täytyi tehdä, jotta ohjelmistoa voitiin testata muualla organisaatiossa. Jokainen iteratiivinen sykli piti sisällään toteutus-, testaus- ja käyttöönottovaiheet.



Kuva 6. Uuden version julkaisusykli

LEAN-ajatteluun olennaisesti kuuluvaa jatkuvaa parantamista ohjelmistoprojektin toteutuksessa tapahtui henkilökohtaisten ja yhteisten työskentelytapojen muokkautumisen muodossa. Käytännön toteutuksessa jatkuvaa parantamista tapahtui jokaisen ohjelmistoversion yhteydessä, sekä ohjelmiston laadun, että koodauksen laadun ja tehokkuuden suhteen. Ohjelmiston laadussa parannukset näkyivät dynaamisempina ja käyttäjäystävällisempinä käyttöliittymänä. Laadun ja tehokkuuden parannus koodissa tarkoitti yksinkertaisempia ja kattavampia toistorakenteita ja selkeämpää dokumentointia.

5.2.3 LEAN-ajattelun toteutukselle tuottama arvo

Ohjelmistoprojektin toteutuksen kannalta iteraatiopohjaisen ketterän kehitysmallin valinta osoittautui hyväksi ratkaisuksi. Täydellisyyttä kohti pyrkiminen iteratiivisen ohjelmistokehityksen avulla tapahtuu askel tai iteraatio kerrallaan hukkia poistaen (Lehtinen, 2019, s. 44-45). Hukista erityisesti virheiden poistaminen osoittautui luontevimmaksi toteuttaa iteraatioiden välissä. Asiakkaan uusien vaatimusten toteutus oli myös helpompaa, kun edellisistä iteraatiosta opittua pystyttiin käyttämään uudelleen, kuten esimerkiksi vapaasti muokattavissa olevien pivot-kaavioiden aikajanan toteutus datatiedostoon. LEAN-ajattelun mukainen ohjelmistokehityksen läpinäkyvyys antoi toteutukselle lisäarvoa samaan tapaan kuin suuremmissakin projekteissa (Rodrigues s. 117-118). Läpinäkyvyys auttoi sidosryhmien edustajat pysymään samassa tahdissa ohjelmiston kehityksessä.

6 TULOKSET

Ohjelman onnistumista mitattiin pääasiallisesti alustavalla ja kokoavalla tyytyväisyyskyselyllä. Tyytyväisyyskyselyiden keskiarvoiset tulokset on esitetty alla olevissa taulukoissa (Taulukko 7. Tyytyväisyyskysely työntekijöille) (Taulukko 8. Tyytyväisyyskysely esimiehille) ja kertymäjakaumakuvaajissa (Taulukko 9. Tyytyväisyysjakauma työntekijät) (Taulukko 10. Tyytyväisyysjakauma esimiehet). Taulukoiden ”vanha” sarakkeen arvot viittaavat alustavaan kyselyyn ja ”uusi” sarakkeen arvot kokoavaan kyselyyn, jonka toteutuksen aikaan ohjelmistoa oli ehditty käyttämään kolme viikkoa. Kokonaisuudessa työntekijöiden tyytyväisyys laaturaportointiprosessin tuotekatselmus- ja aamupalaveriosioon parani pistekeskisarvosta 3,84 pistekeskisarvoon 4,22 ja esimiehillä 3,5:stä 4,5:een. Tuotekatselmuksen suoritus aika lyheni 15,5 minuutista 12,2 minuuttiin.

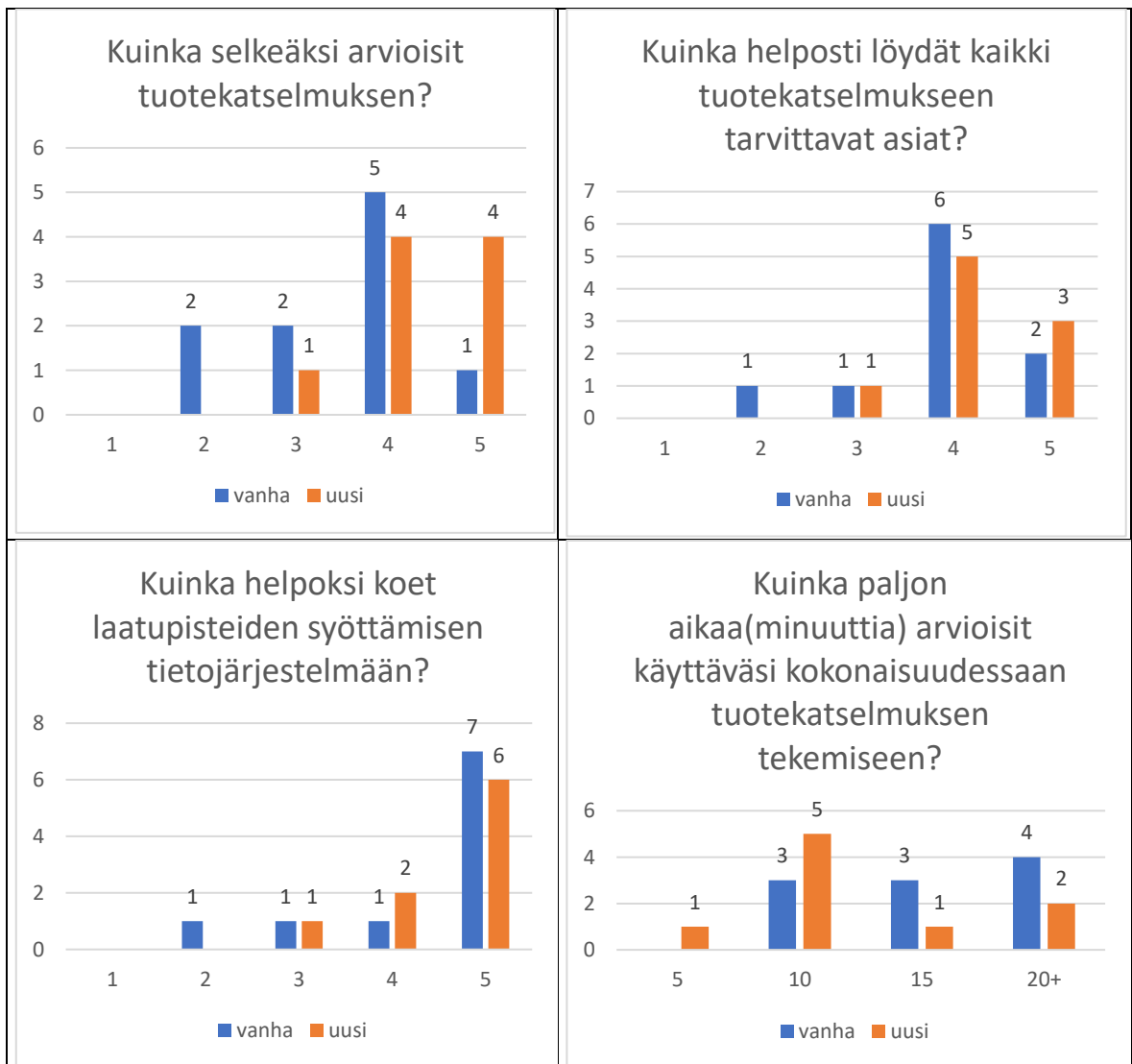
Taulukko 7. Tyytyväisyyskysely työntekijöille

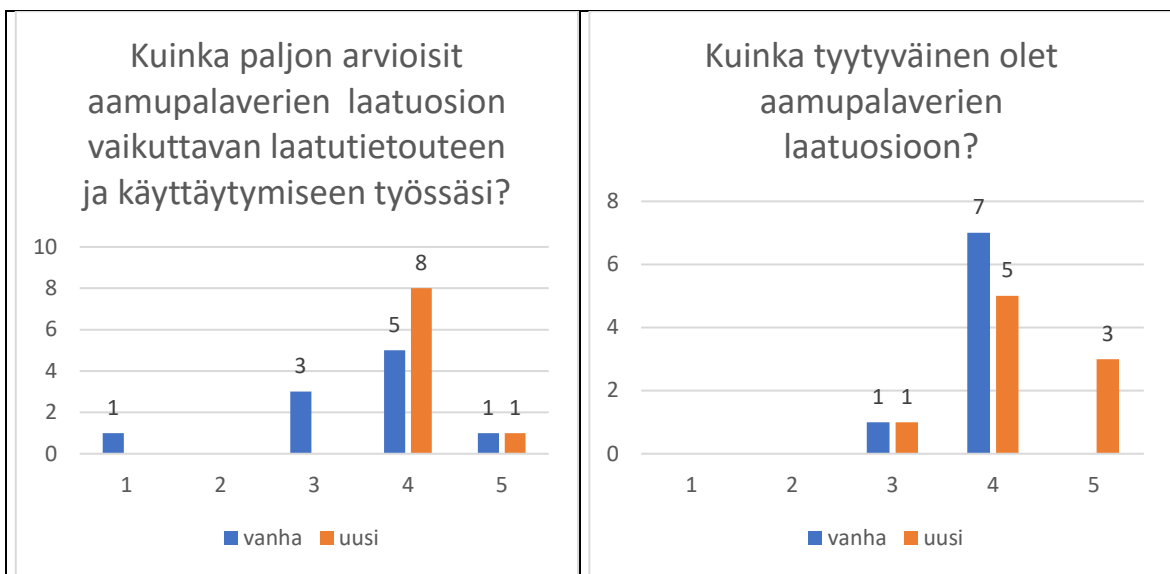
Kysymykset työntekijöille	vanha N=10	uusi N=9
Kuinka selkeäksi arvioisit tuotekatselmuksen? (1-5)	3,5	4,3
Kuinka helposti löydät kaikki tuotekatselmukseen tarvittavat asiat? (1-5)	3,9	4,2
Kuinka helpoksi koet laatupisteiden syöttämisen tietojärjestelmään? (1-5)	4,4	4,3
Kuinka paljon aikaa(minuuttia) arvioisit käyttäväsi kokonaisuudessaan tuotekatselmuksen tekemiseen? (5-20+)	15,5	12,2
Kuinka paljon arvioisit aamupalaverien laatuosion vaikuttavan laatu-tietoutteen ja käyttäytymiseen työssäsi? (1-5)	3,5	4,1
Kuinka tyytyväinen olet aamupalaverien laatuosioon? (1-5)	3,9	4,2

Taulukko 8. Tyytyväisyyskysely esimiehille

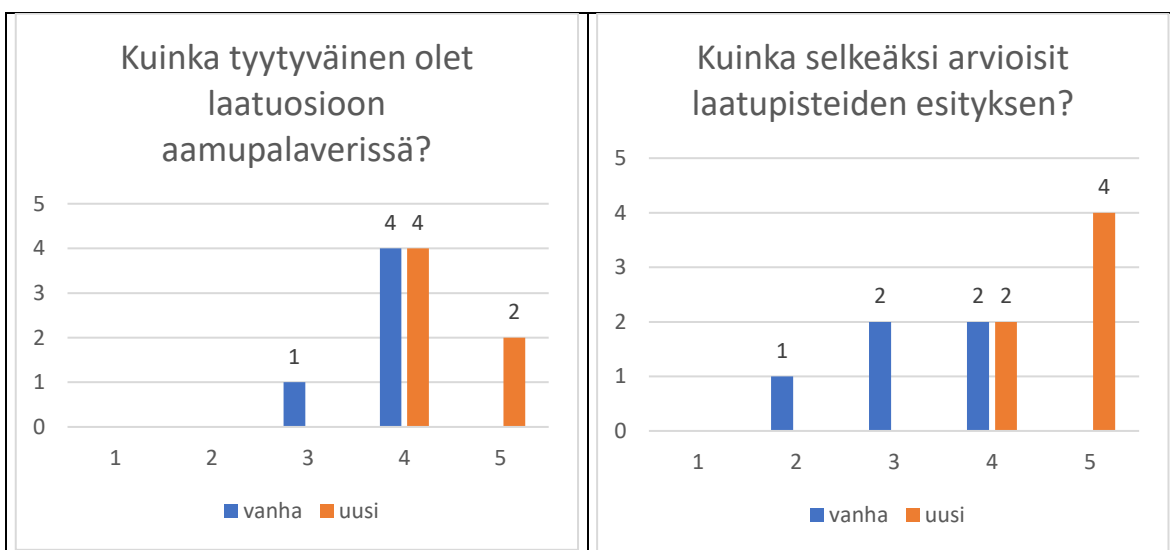
Kysymykset esimiehille	vanha N=5	uusi N=6
Kuinka tyytyväinen olet laatuosioon aamupalaverissä? (1-5)	3,8	4,3
Kuinka selkeäksi arvioisit laatupisteiden esityksen? (1-5)	3,2	4,7

Taulukko 9. Tyytyväisyysjakauma työntekijät





Taulukko 10. Tyytyväisyysjakauma esimiehet



7 JOHTOPÄÄTÖKSET JA YHTEENVETO

Tyytyväisyyskyselyiden pohjalta voidaan todeta ohjelmiston saavuttaneen miltei kaikki alkuperäiset tavoitteensa tuotekatselmukseen käytetyn ajan optimaalista minimointia lukuun ottamatta. Työntekijät olivat tyytyväisempiä laadun arviointiprosessiin kauttaaltaan, mutta eritoten tyytyväisyyttä lisäsi tietojen paperille kirjauksen ja laatupisteiden pisteytyksen laskemisen poistuminen. Aamupalaverin uusi ilme laatupisteitä valittuna ajankohtana esittävine kuvaajineen näytti parantavan merkittävästi tietoisuutta laadun kehittymisestä. Esimiesten kohdalla tyytyväisyys aamupalaveriin ja eritoten laatudatan esitykseen parani huomattavasti. Alun perin tavoitteena olleen tuotekatselmuksen suoritusajan lyheneminen viidestätoista minuutista kymmeneen minuuttiin ei aivan toteutunut, mutta asiaan saattoi vaikuttaa alustavan kyselyn liian lyhyt aikajakauma. Kyselyssä maksimi käytetty aika oli 20 minuuttia. Palautteen mukaan katselmuksiin oli käytetty jopa 30 minuuttia. Tulevaisuudessa tuotekatselmukseen käytetty aika todennäköisesti lyhenee, kun työntekijät tottuvat uuteen käyttöliittymään.

Kirjausten järjestelmään syöttämisen selkeyden tyytyväisyyspisteiden samana pysyminen johtui mahdollisesti siitä, että näennäisesti työmäärä pysyi samana. Alkutilanteessa järjestelmään syötettiin käsin lasketut kokonaispisteet kullekin laatusegmentille. Uudessa järjestelmässä syötettäviä tietoja oli lukumääräisesti enemmän, mutta laskenta tapahtui järjestelmässä ja numeroiden kirjaamisen sijaan työntekijät valitsivat klikkaamalla sopivat luvut.

Vaativuusmäärittelyn toteuttaminen kohdeyrityksessä onnistui todella hyvin ja vaatimusmäärittelyn luomat raamit tukivat erinomaisesti ohjelmiston toteutusta eikä ohjelmistolle tarvinnut tehdä laajoja muutostöitä projektin loppuvaiheessa (Juvonen 2018, s. 59). Vaatimusten ja rajoitteiden kirjaus ja analysointi antoivat kaikille sidosryhmille selkeämmän kuvan projektin luonteesta ja tavoitteista. Palaverit ja kyselyt toimivat tehokkaina vaatimusten kerääjinä kaikilta sidosryhmiltä.

LEAN-ajattelun soveltaminen ohjelmistokehitykseen kohdeyrityksessä onnistui myös hyvin kiitos ammattitaitoisen ja projektille omistautuneen laatuvaastaavan, joka jaksoi olla mukana jokaisessa ohjelmiston versioiteraatiossa. Uusien ohjelmistoversioiden tiuha julkaisusykli

sitoi laatuviastaavan olennaiseksi osaksi ohjelmiston toteutusta, joka selkeästi lisäsi halua ajaa projekti onnistuneesti maaliin. Nopealla julkaisusyklillä saavutettiin jatkuvan parantamisen hyödyt ja saatiin karsittua LEAN-ajattelun mukaisia hukkia, etenkin laatuviirheitä. Muista hukista ylituotantoa ja viivästyksiä ja odottamisia saatiin karsittua hyvällä keskusteluyhteydellä. Kaikkien hukkien, kuten esimerkiksi ylikäsittelyn poistaminen osoittautui ympäristötekijöiden vuoksi lähes mahdottomaksi. Suurimman hyödyn LEAN-ajattelusta saisi varmasti seuraavia ohjelmistoprojektia toteuttaessa, jolloin jatkuvan parantamisen avulla luotu pohja tiedoille ja taidoille pääsisi parhaiten esiin.

LEAN-ajattelun avulla asiakkaalle tuotetun arvon mittaamisen hankaluuden vuoksi voidaan tässä tapauksessa ohjelmistoprojektin katsoa tuottaneen asiakkaalle arvoa, koska tyytyväisyyskyselyiden mukaan yleinen tyytyväisyys laadunraportointiprosessiin ja laadudatan esityksiin kasvoi työntekijöillä keskimäärin 9,8 % ja esimiehillä 28,6 %. Toinen mittari, jolla arvontuottamista ja tyytyväisyyttä voidaan mitata, oli ohjelmiston käytön mahdollinen leviäminen yrityksen muihin pisteisiin. Palautteena tullut kommentti asiakkaan odotusten ylittämisestä oli myös hyvä merkki onnistumisesta.

Kokonaisuudessa projekti antoi paljon onnistumisen tunteita ja auttoi rakentamaan kokonaan uudenlaisen käytännönläheisen kuvan ohjelmistoprojektin toteuttamisesta ideasta oikeasti toimivaksi käytössä olevaksi ohjelmaksi. Ohjelmistoprojektin vaatimusmäärittelyn toteuttaminen huolellisesti auttoi todella paljon käytännön toteutuksessa ja lähensi näkemyksiä halutusta lopputuloksesta asiakkaan edustajan kanssa. Jatkuva kommunikaatioyhteys sähköpostien välityksellä asiakkaaseen oli myös ensiarvoisen tärkeää. Sen avulla asiakkaan subjektiiviset näkemykset ohjelmiston kehityksestä saatiin rajattua teknisessä kontekstissa mahdollisiin ja mahdottomiin vaihtoehtoihin. Näin asiakkaan ei tarvinnut tehdä taustatutkimusta mahdollisista ohjelmallisista rajoituksista.

LÄHTEET

1. Antti Huuskonen, Talentree, 2018, Miten laatu järjestelmän rakentaminen etenee, <https://talentree.fi/blogi/miten-laatu-jarjestelman-rakentaminen-etenee/>, Vierailtu 15.9.2019
2. Deepti Karla, Agile Coach Diaries, 2017, LEAN software development – 7 wastes of software development, <https://agilecoachdiaries.wordpress.com/2017/01/19/lean-software-development-7-wastes-of-software-development/>. Vierailtu 17.9.2019
3. Ella Holmala, Kandidaatintyö, 2019 Asiakasvaatimuksien täyttymisen haasteet tietojärjestelmien käyttäjäkeskeisessä suunnittelussa, <https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/27278/Holmala.pdf>. Vierailtu 15.9.2019
4. Haikala Ilkka & Mikkonen Tommi, 2011, Ohjelmistotuotannon menetelmät 12.painos. Talentum.
5. Happonen, A., Minashkina, D., 2018, Operations automatization and digitalization – a research and innovation collaboration in physical warehousing, improvement ideas in ASRS and 3PL logistics context, LUT Reports series report 86, ISBN 978-952-335-293-3, ISSN 2243-3376, p. 66
6. Heli Lintula, Pro Gradu -tutkielma, 2004, Vaatimusten validointi ja verifiointi, <http://cs.uef.fi/uku/tutkimus/Teho/helingradu.pdf>, Vierailtu 15.9.2019
7. IBM, Kauppalehden verkkojulkaisu, 2018, Perinteinen teollisuus digitalisoituu, <https://studio.kauppalehti.fi/studiovieras/perinteinen-teollisuus-digitalisoituu-mita-edellakavijat-tekevät-eri-tavalla>. Vierailtu 30.10.2019
8. JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta, 2018, JHS 173 ICT-palvelujen kehittäminen: Vaatimusmäärittely, <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS173/JHS173.pdf>. Vierailtu 15.9.2019
9. Karoliina Luoto, Codento Oy, 2019, Lean ohjelmistokehityksessä, <https://www.slideshare.net/totoroki/lean-ohjelmistokehityksess>. Vierailtu 16.9.2019
10. Kimmo Vehkalahti, 2014, Kyselytutkimuksen mittarit ja menetelmät, Tammi

11. Kortelainen, H., Happonen, A., 2017, Digitalisaatio muokkaa tiedon hallintaan pohjautuvien palveluiden markkinoita, In VTT Blog, DOI: 10.13140/RG.2.2.10075.52006
12. Kortelainen, H., Happonen, A., Hanski, J., 2019, From asset provider to knowledge company - transformation in the digital era, In Lecture Notes in Mechanical Engineering, ISSN: 2195-4356, ISBN 978-3-319-95710-4, p. 333–341, DOI: https://doi.org/10.1007/978-3-319-95711-1_33
13. LEAN Enterprise Institute, 2019, Brief history of LEAN, <https://www.lean.org/WhatsLean/History.cfm>. Vierailtu 16.9.2019
14. Lili-Anne Kihn, 2015, Laatu kolmesta tarkastelukulmasta: tekninen, kaupallinen ja palveluun liittyvä laatu, https://tampub.uta.fi/bitstream/handle/10024/100640/laatu_kolmesta_tarkastelunak_okulmasta.pdf. Vierailtu 28.9.2019
15. Matti Lehtinen, Tietotekniikan Pro gradu-tutkielma, Jyväskylän Yliopisto, Tietotekniikan laitos, 2011, <https://jyx.jyu.fi/bitstream/handle/123456789/39954/URN%3aNBN%3afi%3ajyu-201210102653.pdf?sequence=1&isAllowed=y>. Vierailtu 30.10.2019
16. Mika Tyynismaa, Seinäjoen ammattikorkeakoulu, Opinnäytetyö, 2014, Lean- ja Kanban-menetelmät ohjelmistotuotannossa, https://www.theseus.fi/bitstream/handle/10024/78004/Mika_Tyynismaa.pdf?sequence=1&isAllowed=y. Vierailtu 16.9.2019
17. Minna Partti, HSY, 2014, Toimistopaperin kulutuksen vähentäminen Kuntaliitossa. https://www.hsy.fi/fi/yhteisollejayritykselle/jatehuolto/jatteidenvahentaminen/Documents/Kuntaliitto_raportti.pdf. Vierailtu 5.4.2019
18. Paula Savolainen, Opintojakso CT60A4002 Ohjelmistotuotanto, Lappeenranta-Lahden teknillinen yliopisto, Tietotekniikan koulutusohjelma, lukuvuosi 2018-2019. <https://moodle.lut.fi/course/view.php?id=6704>, Vierailtu 15.9.2019
19. Pilar Rodrigues, University of Oulu, Väitöskirja, 2013, Combining Lean thinking and Agile software development, <http://jultika.oulu.fi/files/isbn9789526203324.pdf>. Vierailtu 20.9.2019
20. Promaint, 2016, Digitalisaatio etenee vauhdilla suomalaisiin tehtaisiin ja logistiikkakeskuksiin, <https://promaintlehti.fi/Tuotantotehokkuuden->

[kehittaminen/Digitalisaatio-etenee-vauhdilla-suomalaisiin-tehtaisiin-ja-logistiikkakeskuksiin/\(offset\)/3](#). Vierailtu 1.11.2019

21. Rami Juvonen, 2018, Ohjelmistoprojektin sudenkuopat ja kuinka ne vältetään
22. Tarja Heikkilä, 2014, Tilastollinen tutkimus 9. painos, Edita Publishing Oy
23. Toni Huhtakangas, Diplomityö, 2019, Tiedonhallinnan kehittäminen ja tiedon analysointi tietojärjestelmän vaatimusmäärittelyn suunnittelussa, https://lutpub.lut.fi/bitstream/handle/10024/159316/Diplomityö_Huhtakangas_Toni.pdf. Vierailtu 13.9.2019
24. Tutorials point, 2019, VBA tutorial, <https://www.tutorialspoint.com/vba/index.htm>. Vierailtu 3.6.2019

LIITE 1. Vaatimukset

P pakollinen

1 Tärkeä. Toteutetaan ensimmäisenä.

2 Ei välttämätön. Toteutetaan käytettävissä olevan ajan puitteissa.

3 Hyvä lisä

ID	Pvm	Vaatus	Vaatimuksen kuvaus	Prioriteetti
V1	10.3.2019	Helppokäyttöisyys	Ohjelman pitää olla mahdollisimman helppokäyttöinen	1
V2	11.3.2019	Ohjelmisto	Microsoft Excel 365	P
V3	11.3.2019	Käyttöliittymä	Excel VBA Userform - käyttäjäformaatti	P
V4	11.3.2019	Laajennettavuus	Mahdollisuus lisätä uusia tuotteita ja tuotantolinjoja	P
V5	13.3.2019	Tietoturva	Työntekijöillä ei pääsyä varsinaiseen laatudataan	P
V6	13.3.2019	Tiedostoarkkitehtuuri	Useampi Excel-tiedosto.	1
V7	13.3.2019	Tuotevalikoima	Mahdollisuus tallentaa useita laatuarvioita kerralla	3
V8	13.3.2019	Laatupoikkeamien tilastointi	Laatupoikkeamien kommenttien automaattinen tilastointi ja esitys	P
V9	13.3.2019	Ehdotetut poikkeamat	Yleisimmät poikkeamat ehdotukset tai käyttäjän itselisäämä.	1
V10	13.3.2019	Oppiva poikkeamatietokanta	Jos samaa poikkeamaa ehdotetaan useasti, järjestelmä lisää sen ehdotuksiin.	3
V11	13.3.2019	Raportin tallennus	Laatudatan tallennus Excel-tietokantaan	P

LIITE 1. Vaatimukset (jatkuu)

V12	13.3.2019	PDF-raportti	Päivä-/Viikkokohtainen -pdf-raportti (sähköpostilla)	1
V13	13.3.2019	Paperittomuus	Muutos paperisesta sähköiseen järjestelmään	P
V14	13.3.2019	Arviointi	Nykyisten arviointialgoritmien ja laatupainotusten käyttö	P
V15	13.3.2019	Päivämäärän vaihto	Päivämäärän muuttaminen	P
V16	13.3.2019	Laatupisteytys	Laatupistevähennysten ja poikkeamien korrelaatio	P
V17	13.3.2019	Nimi	Kirjaajan koko nimen pakollinen tallennus. Helpottaa jäljitettävyyttä.	P
V18	13.3.2019	Kirjausten korjaus	Laatudatan kirjausten korjaus ja muokkaus jälkikäteen.	P
V19	13.3.2019	Arvioididatan syöttö	Graafinen käyttöliittymä datan syöttämiseksi	P
V20	13.3.2019	Käyttöohjeet	Käyttäjille graafinen käyttöohje	P
V21	13.3.2019	Komentointi	Koodin kattava kommentointi helpottamaan koodin muokkaamista	1
V22	13.3.2019	Dokumentointi	Dokumentointi	1
V23	13.3.2019	Tyytyväisyyskyselyt	Ennen ja jälkeen projektin tehtävät tyytyväisyyskyselyt	P
V24	13.3.2019	Laatuominaisuudet	Nykyisten kirjattavien laatuomaisuuksien käyttö	P
V25	13.3.2019	Kirjausdata	Uusi looginen tiedosto kirjausdatalle	1

LIITE 1. Vaatimukset (jatkoa)

V26	13.3.2019	Näkyvyys	Kirjaajan ei tarvitse nähdä edellispäivien tietoja	2
V27	31.3.2019	Pikakuvake	Pikakuvake tuotekatselmuskoneen työpöydälle	P
V28	18.4.2019	Toistuvat poikkeamat	Toistuvien poikkeamien esilletuonti	P
V29	18.4.2019	Kommenttien pakollisuus	Jos laatupisteet jäävät alle standardin ohjelma vaatii sanallisen kommentin	P
V30	18.4.2019	Tuotteiden ryhmittely	Tuoteryhmien luominen, muokkaaminen ja painottaminen	1
V31	18.4.2019	Yksittäisen tuotteen poiminta	Yksittäisen tuotteen datan poiminta tietyllä ajanjaksolla	P
V32	18.4.2019	Esityspohjan rakenne	Ajanjakso, tuotteet, poikkeamat, graafit	P

LIITE 2. Rajoitteet

ID	Pvm	Lähde	Rajoite	Rajoitteen kuvaus	Huomautuksia
R1	10.3.2019	L.H	Käyttöliittymä	Kannettava tietokone/keskustyksikkö + näyttö	Yrityksessä käytössä oleville Android-tableteille/-puhelimille ei ole VBA tukea.
R2	31.3.2019	L.H.	Käyttöjärjestelmä	Microsoft Windows 10	Yrityksessä käytössä olevien tietokoneiden käyttöjärjestelmä
R3	11.3.2019	L.H	Ohjelmisto	Microsoft Excel 365	Yrityksessä käytössä oleva toimisto-ohjelmisto
R4	11.3.2019	L.H	Ohjelmointikieli	VBA (Visual Basic for Applications)	
R5	11.3.2019	L.H	Käyttöpaikka	Tuotekatselmushuone	
R6	11.3.2019	L.H	Ohjaus	Laatuvastaava/työnohjaaja käytössä 13.3.2019 – 1.7.2019	

LIITE 2. Rajoitteet (jatkoa)

R7	11.3.2019	L.H	Suunnittelu-aika	10.3.2019 2.6.2019	–	
R8	31.3.2019	L.H	Tyytyväisyyskysely-aika	19.3.2019 16.4.2019	–	Tyytyväisyyskysely
R9	18.4.2019	L.H	Tyytyväisyyskysely-aika	16.9.2019 30.9.2019	–	Loppukysely
R10	11.3.2019	L.H	Toteutusaika	Työnohessa 3.6.2019 1.8.2019	–	.

LIITE 3. Käyttäjä- ja sidosryhmät

ID	Pvm	Käyttäjärühmä	Kuvaus ja rooli	Huomautuksia
K1	10.3.2019	Työntekijät	Käyttää raportointityökalua tuotteiden laadun määrittämiseen ja datan tallentamiseen.	
K2	10.3.2019	Työnjohtajat	Käyttää esitystyökalua laaturaporttien esittämiseen, analysointiin ja organisaatiossa eteenpäin raportointiin.	
K3	13.3.2019	Laatuvastaava	Analysoi laatu dataa ja raportoi tuloksia edelleen organisaatioon	
K4	31.3.2019	Organisaatio	Käyttää kerättyä dataa laaduntarkkailuun	

ID	Pvm	Sidosryhmä	Sidosryhmän kuvaus ja rooli	Huomautuksia
S1	10.3.2019	Työntekijät	Osallistuu tyytyväisyys- ja ehdotuskyselyyn ennen ja jälkeen projektin.	
S2	10.3.2019	Työnjohtajat	Osallistuu tyytyväisyys- ja ehdotuskyselyyn ennen ja jälkeen projektin.	
S3	10.3.2019	Laatuvastaava	Toimii projektin ohjaajana yrityksen puolelta.	
S4	10.3.2019	IT-osasto	Toimittaa projektin toteutukseen käytetyn tietokoneen ja käyttäjätunnukset	

LIITE 3. Käyttäjä- ja sidosryhmät (jatkoa)

S5	31.3.2019	HR-osasto	Tietosuoja- ja salassapitolomakkeet	Allekirjoitettava ennen projektin toteutuksen aloitusta
S6	10.3.2019	Yksikön päällikkö	Osallistuu projektin aloitukseen ja määrittelee alkuperäisen tarpeen. Vertaa lopullista tulosta tavoitteisiin ja tarkastaa dokumentaation organisaation tietosuojan ja tietojen salassapidon osalta.	Ei osallistunut projektin seurantaan sen aikana

LIITE 3. Hylätyt vaatimukset

Taulukko 11. Hylätyt vaatimukset

ID	Hylkäys-pvm	Vaatus	Syy
H1	13.3.2019	Arvioitavien tuotteiden kuvien tallennus tietokantaan	Ei järkevästi toteuttavissa
H2	18.4.2019	Ohjelmiston toiminta mobiililaitteilla ja tietokantojen pilvitallennus	Ei toteutettavissa annetussa aikataulussa
H3	18.4.2019	Ohjelmiston laajentaminen tiettyyn poikkeavaan työpisteeseen	Hajauttaa/laajentaa ohjelmaa liikaa
H4	18.4.2019	Excel-taulukoiden täydellinen suojaus	Ei mahdollista ilman, että toiminnallisuus kärsii. Ratkaistaan muulla tavalla.
H5	16.8.2019	Oppiva poikkeamatietokanta	Ei mahdollista toteuttaa yhdessä raportointitiedoston keskiarvotaulukoiden kanssa

LIITE 3. Käyttötapauskuvaus

ID	UC-1
Nimi ja versio	Kirjausten syöttäminen
Suorittajat	Työntekijä
Esiehdot	Tietokone ja ohjelma käynnissä
Kuvaus	Työntekijä syöttää tiedot ohjelmaan
Poikkeukset	
Lopputulos	Tiedot on syötetty ohjelmaan
Muut vaatimukset	

ID	UC-2
Nimi ja versio	Tietojen tallennus
Suorittajat	Työntekijä
Esiehdot	Ohjelmaan on syötetty tarpeeksi tietoja
Kuvaus	Työntekijä painaa tallenna-nappulaa
Poikkeukset	
Lopputulos	Tiedot tallennetaan ulkoiseen tietokantaan
Muut vaatimukset	

ID	UC-3
Nimi ja versio	Laatudatan esitys
Suorittajat	Esimies
Esiehdot	Tietokanta auki
Kuvaus	Esimies aukaisee viikkovälilehden ja käyttää välilehdellä olevia laatumittarikuvaajia aamupalaveriesityksessään
Poikkeukset	
Lopputulos	onnistunut esitys
Muut vaatimukset	

LIITE 4. Saatekirje

Tyytyväisyyskysely tuotekatselmuksen tekijöille

Esitiedot

Olen suorittamassa tietotekniikan kandidaatin tutkintoa ja seuraava vaihe on tehdä opinnäytetyö. Opinnäytetyönä minulla on suunnitella ja toteuttaa raportointityökalu korvaamaan tuotekatselmuksessa käytössä oleva lomake ja luoda graafinen esityspohja tuotantoesimiehille laatukirjauksista saadun datan esittämiseen.

Työn tarkoitus on helpottaa kirjausten tekoa poistamalla paperiset lomakkeet ja luomalla tuotekatselmuksen tekijälle käyttöliittymä, jonka käyttö on nopeaa ja yksinkertaista. Työ toteutetaan kesä-heinäkuussa 2019.

Tyytyväisyyskysely

Osana opinnäytetyön alustusta ja mahdollisten muutosten mittaamista teen kaksi kyselyä. Ensimmäinen kysely ajoittuu maaliskuuhun ja toinen heinäkuun alkuun.

Tyytyväisyyskysely on lyhyt monivalintakysymyksiä sisältävä mielipidekysely. Kysely sisältää kaksi vapaaehtoista kysymystä kehitysideoista. Kyselyn monivalintatuloksia käytetään lähtötasona kandidaatintyön suunnittelulle ja kehitysideoita toteutetaan mahdollisuuksien mukaan.

Vastaaminen

Vastaamiseen on vapaaehtoista ja siihen menee pari minuuttia. Vastaukset tallennetaan nimettömänä ja vastauksia käsitellään luottamuksellisesti. Tuloksia esitellään ryhmätasolla, niin ettei yksittäisiä vastauksia pysty tunnistamaan.

Vastata voit, jos teet tai olet joskus tehnyt tuotekatselmuksia.

Vastaamaan pääset kannettavan tietokoneen työpöydällä olevan ”Tyytyväisyyskysely”-linkin kautta. Täytä monivalintakohdat ja paina ”SUBMIT” lähettääksesi kyselyn.

Vastausaikaa on **16.4.2019** asti.

Tulokset

Tulokset kyselyistä julkaistaan osana opinnäytetyötä.

KIITOS OSALLISTUMISESTA!

Ystävällisin terveisin,

Lauri Hämäläinen