

Opinnäytetyö (AMK)

Elektronikka

Elektronikkatuotanto

2014

Henri Särkkä

ELEKTRONIIKKATUOTTEEN VALMISTUSYMPÄRISTÖN SUUNNITTELUPROSESSIN KEHITTÄMINEN



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Elektroniikka | Elektroniikkatuotanto

2014 | 41 sivua

Yngvar Wikström, Mauri Aalto

Henri Särkkä

ELEKTRONIIKKATUOTTEEN VALMISTUSYMPÄRISTÖN SUUNNITTELUPROSESSIN KEHITTÄMINEN

Tämän työn tavoitteena oli valmistusympäristön suunnittelun optimointi kaapelitelevisioverkon vahvistimen päivitysprojektin myötä. Hukka-aikaa paikallistettiin arvovirtakarttamenetelmien avulla lean-ajatteluun perustuen. Konkreettisia ongelmia löytyi muun muassa testauskoodin luettavuudesta ja eri osastojen lean-implemентаatioista. Yhtiön eri osastojen välistä kollektiivista tietoa ja yhteistyötä tarkasteltiin poikkihallinnollisten ryhmien näkökulmasta.

Työn tuloksena uuden vahvistimen testausympäristö saatiin valmiiksi. Ronald Mascitellin hukkatyyppien mukaan jaoteltuja ehdotuksia jätettiin osasto- ja projektipäälliköiden harkintojen mukaan implementoitaviksi. Tuotteen valmistusympäristön holistista ymmärrystä edistettiin tuottamalla selkeä graafinen kuvaus valmistusympäristön suunnitteluprosessista. Testaussuunnittelijoille tehtiin lähdekoodin kommentointia koskeva ohjeistus. Ohjelmistojen lähdekoodin selkeyttämiseksi tehtiin ohjeistus kommentointia varten. Kommentoinnin lisäämisen tarkoitus oli nopeuttaa olemassa olevan koodin lukemista ja ymmärtämistä sekä uusien ominaisuuksien kehittämistä.

ASIASANAT:

tuotannonsuunnittelu, testaus, lean-ajattelu, arvoketjut, vahvistimet

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Electronics | Electronics Production

2014 | 41 pages

Yngvar Wikström M.Eng., Mauri Aalto M.Sc.

Henri Särkkä

DEVELOPMENT OF AN ELECTRONIC PRODUCT'S PRODUCTION PLANNING

The goal of this thesis was to improve the production planning process of a CATV amplifier. Waste time was located using value stream maps and lean methods. Problems were identified in test code commenting and the implementation of lean principles across departments.

The cooperation between the company's different functions was analyzed based on cross-functional team work. To enhance the holistic understanding of the product's production environment, a graphic map was produced. Improvement propositions were made based on Ronald Mascitelli's waste categories.

To improve the readability of test code, specifications were made for proper commenting and tagging of the LabVIEW code. New features of LabVIEW version 2013 played an important role, because a virtual continuous improvement board was produced using them.

KEYWORDS:

production planning, testing, lean, value chains, amplifiers

SISÄLTÖ

SANASTO

1 JOHDANTO	1
2 KTV-VERKKOVAHVISTIN AC3200	2
3 TYÖVÄLINEET	5
3.1 LabVIEW	5
3.2 TestStand	6
3.3 Lean-periaatteet	7
3.4 Arvovirtakartat	7
4 TUOTESUUNNITTELUN ELINKAARI	11
4.1 Tuotteen alkusuunnittelu ja arvioinnit	13
4.2 Tuotekehitys ja testaussuunnittelu	17
5 TUOTEPROJEKTI	22
5.1 Projektin tavoitteet	22
5.2 Projektin eteneminen	23
5.3 Havaitut ongelmat Ronald Mascitellin hukkatelijöiden mukaan	27
5.3.1 R&D:n ongelma 1: Sekava työympäristö	28
5.3.2 R&D:n ongelma 2: Resurssipullonkaulat ja tehtävien priorisointiongelmat	28
5.3.3 R&D:n ongelma 3: Arvaamattomat muutokset tuotevaatimukseen	30
5.3.4 ME:n ongelma 1: Sekava työympäristö	31
5.3.5 ME:n ongelma 2: Puutteellinen kommunikaatio yksiköiden välillä	32
5.3.6 ME:n ongelma 3: Puutteelliset tuotevaatimukset	32
5.3.7 ME:n ongelma 4: Ylisuoriutuminen ja -analysointi	33
5.3.8 ME:n ongelma 5: Liiallinen sähköpostittelu	34
6 KOODIN KOMMENTOINTI	35
6.1 Projektin esiselvitys	35
6.2 Kommentointispesifikaation rakentaminen	36
7 YHTEENVETO	40
LÄHTEET	41

LIITTEET

Liite 1. Valmistusympäristön suunnittelun VSM-kuvaus.

Liite 2. LabVIEW-kommentointikyselylomake.

SANASTO

CFT	Cross-Functional Team. Poikkihallinnollinen työryhmä, johon osallistuu henkilöstöä useilta eri osastoilta.
CSM	Current State Map. Prosessin nykytilaa mallintava kuvaus.
FSM	Future State Map. Prosessin tulevaa tilaa suunnitteleva kuvaus.
Hybridi	Hybrid Amplifier. Tässä yhteydessä SOT-115J-pakattu MESFET-vahvistinkomponentti.
Jigi	Moduulin tuotannossa ja testauksessa käytetty I/O-alusta. Myös testifikstuuri, kojekiinnitin.
Lean	Tuotantoperiaate, jossa keskitytään hukkatekemisen (<i>muda</i>), epätasaisuuden (<i>mura</i>) ja ylikuormituksen (<i>muri</i>) poistamiseen prosesseista.
ME	Manufacturing Engineering. Testaus- ja tuotantotekniikka.
Muda	Toiminta, joka ei tuo lisäarvoa tuotteelle. Tällaista ovat kuljetus, varastointi, liike, odottelu, ylituotanto, yliprosessointi ja viallisuudet.
Mura	Markkinoista johtumaton tuotannollinen epätasaisuus.
Muri	Tuotannon ja työntekijöiden liiallinen kuormitus.
NPI	New Product Introduction. Uuden tuotteen saattaminen tuotekehityksestä valmistukseen ja lopulta asiakkaalle.
OLS	Operations, Logistics, Sourcing. Tuotanto, logistiikka ja hankinta.
R&D	Research & Development. Tuotekehitys.
Sekvenssi	TestStand-kehitystyökalulla tehty ohjelmisto, jolla voidaan automatisoida esimerkiksi laitteen testausta.

Valmistusympäristö

Tuotteen valmistamiseen tarvittava laitteisto ja ohjelmisto.

VSM

Value Stream Map. Prosessin materiaalin ja informaation kulkua mallintava kuvaus, jonka tarkoituksena on eritellä hukka-aikaa.

1 JOHDANTO

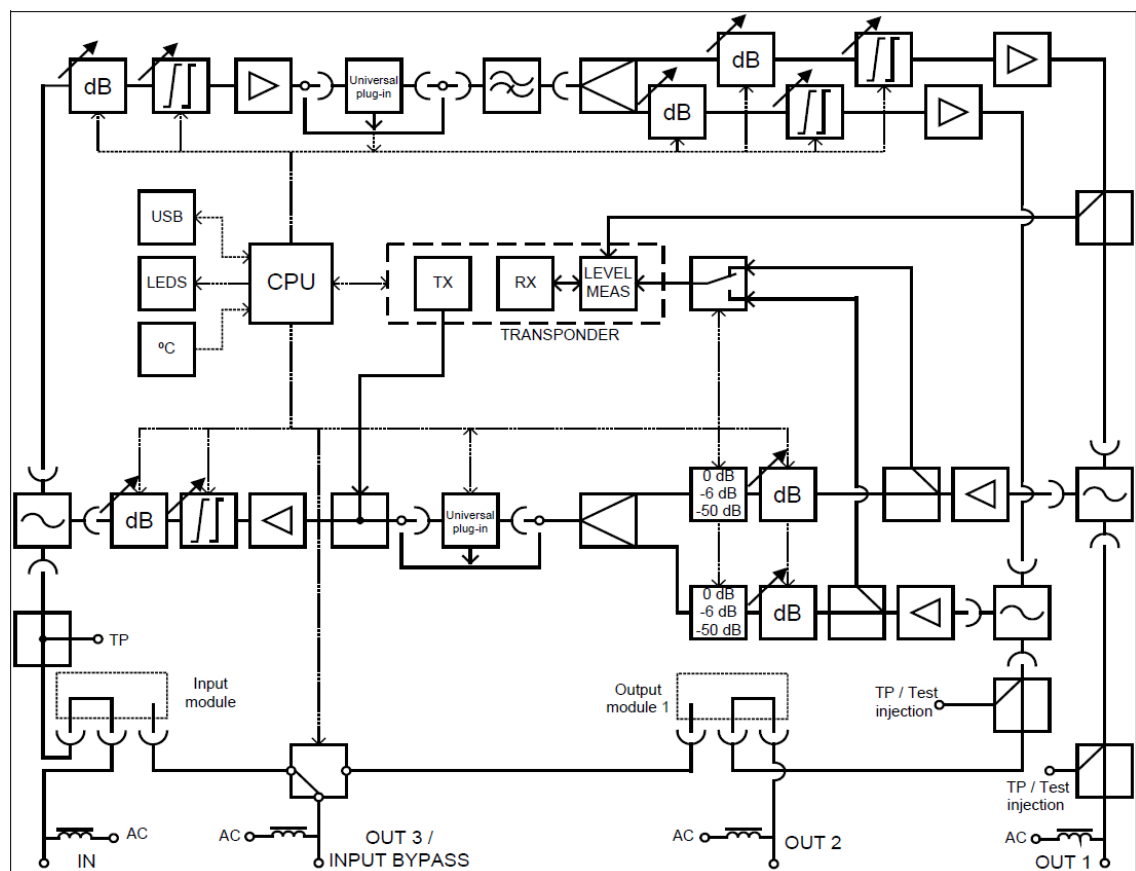
Työn tarkoituksena on katselmoida elektroniikkalaitteen valmistus- ja testaussuunnitteluprosessia sekä suunnitella siihen implementoitavia lean-periaatteiden mukaisia parannusehdotuksia. Vastaavia tutkimuksia ovat aikaisemmin tehneet muun muassa Ville Knuutila opinnäytetyössään ”Jatkuva parantaminen Lean-ajattelun työkaluna: case: Nordea Pankki Suomi Oyj, Osasto X” [1] sekä Obe Olabode Sunday opinnäytetyössään ”Value Stream Mapping of a mobile network element software - feature development cycle using Lean development concept” [2].

Tarkasteltava testaussuunnittelu koskee kaapelitelevisioverkon RF-vahvistinta sekä sen päivitysprojektia. Pääsääntöisenä arviointityökaluna toimivat arvovirtakartat (VSM), joiden avulla kyetään arvioimaan arvoa tuottavan ajan ja mudan suhde. Vertailukohteeksi päivitysprojektin rinnalle otetaan teoreettinen täysin uudenlainen tuote, jonka valmistusympäristön suunnittelun eri vaiheiden ajalliset kestot voivat poiketa huomattavasti pelkästä päivityksestä.

Työ keskittyy testaamiseen ja valmistettavuuteen, joten fokus on OLSin toiminnassa. Tästä syystä R&D:n ja muiden osastojen tekemisiä ei käsitellä samalla laajuudella. OLSiin kuuluu myös sourcing eli materiaalihankinta, jonka toimintaan ei ole tarkoitus syventyä.

2 KTV-VERKKOVAHVISTIN AC3200

Tässä työssä käsitellyn projektin laitteena oli kaapelitelevisioverkon RF-vahvistin AC3200. Laitteessa on kaksi aktiivista lähtöä, eli sekä ensimmäisessä että toisessa lähdössä (out 1 ja 2) on sama lähtötaso. Kolmas lähtö (out 3) voidaan ottaa käyttöön passiivista haaroitin- tai jaotinpistoyksikköä käyttäen joko out 2:n kanssa jaetuksi lähdeksi tai suoraan lähtöporttiin tulevaa signaalia jatkavaksi bypass-portiksi (kuva 1).



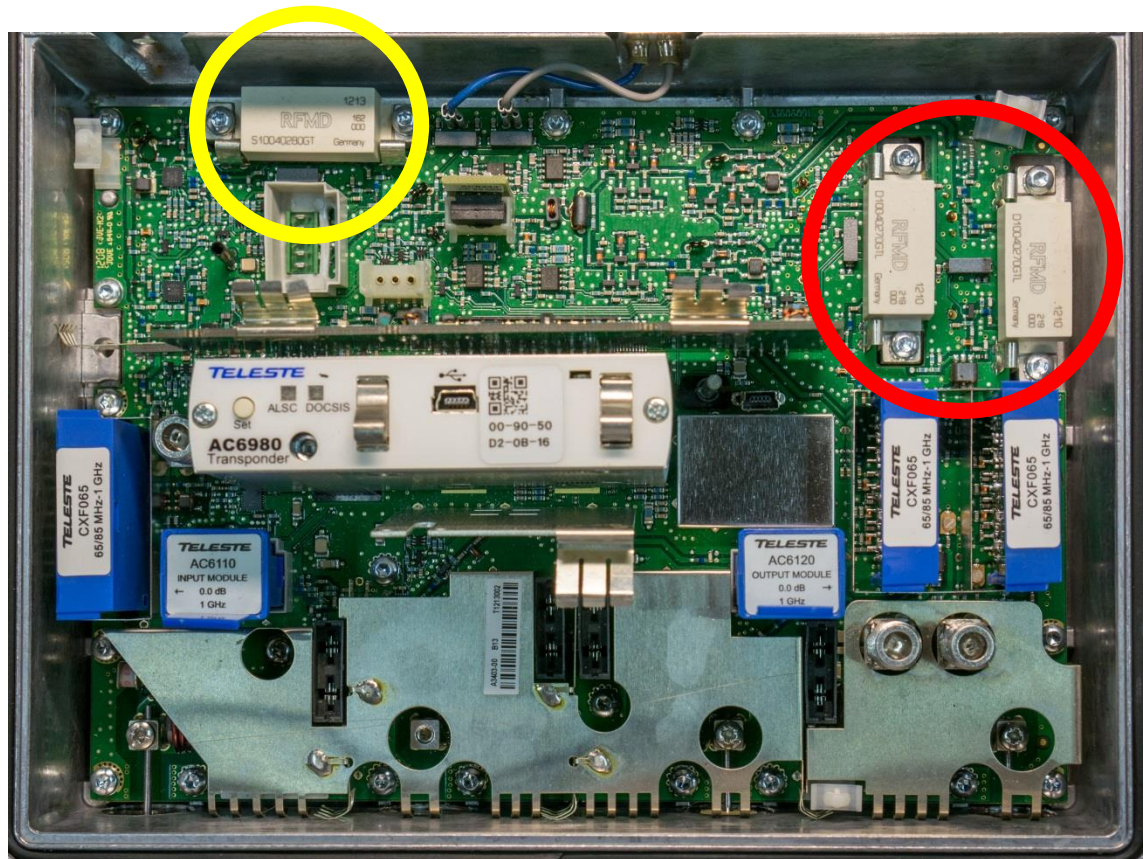
Kuva 1. AC3200:n lohkoakaavio. [3]

AC3200:n asetuksia voidaan ohjata etänä transponderimoduulin (kuva 2) avulla. Transponderissa olevalla set-painikkeella voidaan asettaa automatisoidusti säädöt vaimennuksille ja kaltevuuksille. Transponderi pystyy myös mittaamaan ja käsittelemään esiintyvää ingressiä eli ulkopuolisten signaalien aiheuttamaa häiriötä.



Kuva 2. Transponderi irrallaan sekä vahvistimessa kiinni.

Pääasiallisina vahvistavina komponentteina AC3200:n edellisessä versiossa käytettiin 3:a galliumarsenidi (GaAs) -hybridivahvistinta, joista 1 oli tuloasteessa ja 2 lähdöissä (kuva 3).



Kuva 3. Hybridivahvistinkomponenttien sijainnit AC3200:n emolevyllä.

Suunnitteilla olleessa laitteessa hybridit vaihtuivat. Lähtöhybridien tuli vaihtua GaAs-versioista galliumnitridi (GaN) -pohjaisiin komponentteihin ja tulohybridin tulisi vaihtua kokonaisuudessaan toisenlaiseksi SMD-pohjaiseksi piiriksi.

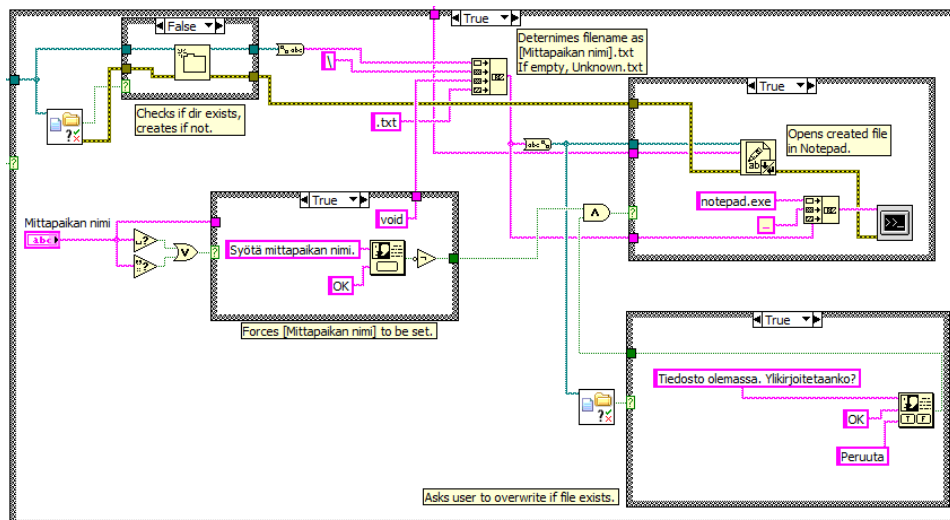
3 TYÖVÄLINEET

3.1 LabVIEW

LabVIEW on National Instrumentsin kehittämä graafinen ohjelmointityökalu. Sen yleisimpiä käyttökohteita ovat datan kerääminen & prosessointi, laitteiston hallinnointi ja automaatio. Testauksessa ohjelman rooli on siis suuri.

LabVIEW'n koodi muistuttaa toiminnaltaan suurinta osaa muita ohjelmointikieliä. Eteneminen on tavallisesti lineaarista ja käyttää ohjelmoinnin peruskäsitteitä kuten eri tyyppisiä muuttujia, logiikkaportteja sekä if-valitsimien ja while-silmukoiden kaltaisia rakenteita.

Yksittäistä ohjelmistonosaa kutsutaan nimityksellä VI, Virtual Instrument. Yleensä yksittäinen VI sisältää usean ali-VI:n. Ali-VI:t ovat tyypillisesti monikäyttöisiä funktioita ja niitä käytetään useissa erilaisissa kokonaisuuksissa. Visuaalisesti niitä kuvastavat eri kuvakkeilla kuvitetut laatikot (kuva 4).



Kuva 4. Esimerkki LabVIEW-koodista.

Telesten ME-osasto on jaettu toiminnallisuudeltaan kahteen osaan, ohjelmisto- ja laitteistopuoliin. Ohjelmistopuolen työt pyörivät juuri LabVIEW'n sekä TestStandin ympärillä, joten kaikki näiden työtapojen tehostaminen ja leanin mukainen optimointi edesauttaa tuottavuutta. Tähän aihepiiriin keskitytään tarkemmin kappaleessa 6.

3.2 TestStand

LabVIEW-koodia hyödynnetään useimmiten samaan ekosysteemiin kuuluvassa TestStand-ohjelmistossa. TestStandilla luodaan toimintasarjoja eli sekvenssejä, joissa halutut VI:t ja muut toiminnot suoritetaan asetetussa järjestyksessä (kuva 5). Mittausten suoritusjärjestys määritellään testausvaatimuksessa.

Step	Description	Settings
1: Voltage_Test	Call Voltage_Test in <Current File>	Post Action
Rv_Control_DUT_2	Call ControlDut_Return in <Current File>	Result Recording: Disabled, Disable Tracing
Control DUT_4	Call ControlDut_Fwd in <Current File>	Result Recording: Disabled, Disable Tracing
2: Return path response 4 -> 1		Post Action, Disable Tracing
2a: Return path response 4 -> 1 NoRefCable		Post Action, Disable Tracing
Ret Zero Offset		
3: CalibRetGainOffset	Call CalibRetGain in <Current File>	Result Recording: Disabled, Disable Tracing
Rv_Control_DUT_6	Call ControlDut_Return in <Current File>	Result Recording: Disabled, Disable Tracing
6: Return path response 4 -> 1 Slope		Post Action, Disable Tracing
Rv_Control_DUT_4	Call ControlDut_Return in <Current File>	Result Recording: Disabled, Disable Tracing
4: Return path response 3 -> 1		Post Action, Disable Tracing
4a: Return path response 3 -> 1 NoRefCable		Post Action, Disable Tracing
Ret Zero Offset		
5: CalibRetGainOffset_2	Call CalibRetGain in <Current File>	Result Recording: Disabled, Disable Tracing
Rv_Control_DUT_7	Call ControlDut_Return in <Current File>	Result Recording: Disabled, Disable Tracing
7: Input TP transmission 1 -> 5		Post Action, Disable Tracing
1=>6		
SetMeas	Action, TDriverCall.vi	Result Recording: Disabled
8: Port 1 reflection		Post Action, Disable Tracing
9: Port 3 reflection		Post Action, Disable Tracing
10: Port 4 reflection		Post Action, Disable Tracing
Rv_Control_DUT_7B	Call ControlDut_Return in <Current File>	Result Recording: Disabled, Disable Tracing

Kuva 5. Sekvenssi avattuna TestStandissa.

Ensin esimerkiksi voidaan ajaa koodi, jossa mittalaite alustetaan. Tämän jälkeen mittalaitteelle lähetetään halutut käskyt, ja lopulta mittaustulokset haetaan testattavalta laitteelta ja tallennetaan. Tavallisessa testausympäristössä koodi suorittaa valtaosan mittauksista ja datan prosessoinnista automatisoidusti, jolloin operaattorin tehtäväksi jää fyysisesti manipuloida laitetta ja käsitellä mahdollisia virhetiloja.

TestStandia käytetään yleisesti LabVIEW'n kanssa. Sekvenssiin pystyy lisäämään LabVIEW'illa tehtyjä VI-ohjelmia, joiden tulevia ja lähteviä arvoja voidaan asettaa, lukea ja hyödyntää. Laitteen sähköinen testaus voidaan siis suorittaa kokonaisuudessaan yhden pääsekvenssin käynnistämällä.

3.3 Lean-periaatteet

Lean-ideologia sai alkunsa Toyota-autoyhtiössä kehitetystä Toyota Production Systemistä (TPS) eli Toyotan tavasta järjestää tuotanto-organisaatio ja sen toimintatavat. TPS:n eli myös leanin keskeisenä ideana on eliminoida kolmen tyyppistä tuotannollista haittaa: hukka-aikaa ja -tekemistä eli mudaa, tuotannon epätasaisuutta eli muraa ja liiallista kuormitusta eli muria.

TPS on luonteeltaan sosiotekninen, eli siinä on tuotantoteknisten käsitteiden lisäksi myös kulttuurillista ja sosiologista rajapintaa. [4] Leanin edustamia ideoita ei voi implementoida maailmanlaajuisesti samanlaisena pakettina, vaan lean-strategiat tulee soveltaa jokaisen yrityksen omaan työkuulttuuriin sopivasti. Leania ja muita vastaavia toimintamalleja kammoksutaan monissa työympäristöissä turhaan, sillä ne nähdään juurikin valmiina työntekosäännösteinä joita on noudatettava täsmällisesti. Leanista on vuosien saatossa kasvanut valtava kattokäsite jonka alle voidaan asettaa lukuisia muita nykyaikaisia toimintaraameja. Oleellista on olla vieroksumatta uusia ideoita ja joustavasti vakiinnuttaa niistä toimivimmat käyttöön.

3.4 Arvovirtakartat

Arvovirtakartat eli value stream mapit ovat prosessikuvauksia, joiden avulla pyritään kartoittamaan ja erittelemään prosessista se aikamäärä jolla tuotteelle saadaan lisäarvoa, sekä aika, joka kuluu mudaan. Tavallisesti VSM:n mallintamat prosessit ovat yksittäisiä tuotteita. [5]

Tyypilliseen VSM-pakettiin kuuluu kaksi eri karttaa: Current State Map (CSM) kuvastaa mahdollisimman realistisesti kuvattavan prosessin nykytilaa, ja tästä iteroitu Future State Map (FSM) kuvastaa samasta prosessista tilaa, jota kohti halutaan kehittyä. Tavoitteena on minimoida mudaa, jolloin mahdollisimman suuri osa tekemisestä ja prosessiajasta tuottaa tuotteelle lisäarvoa.

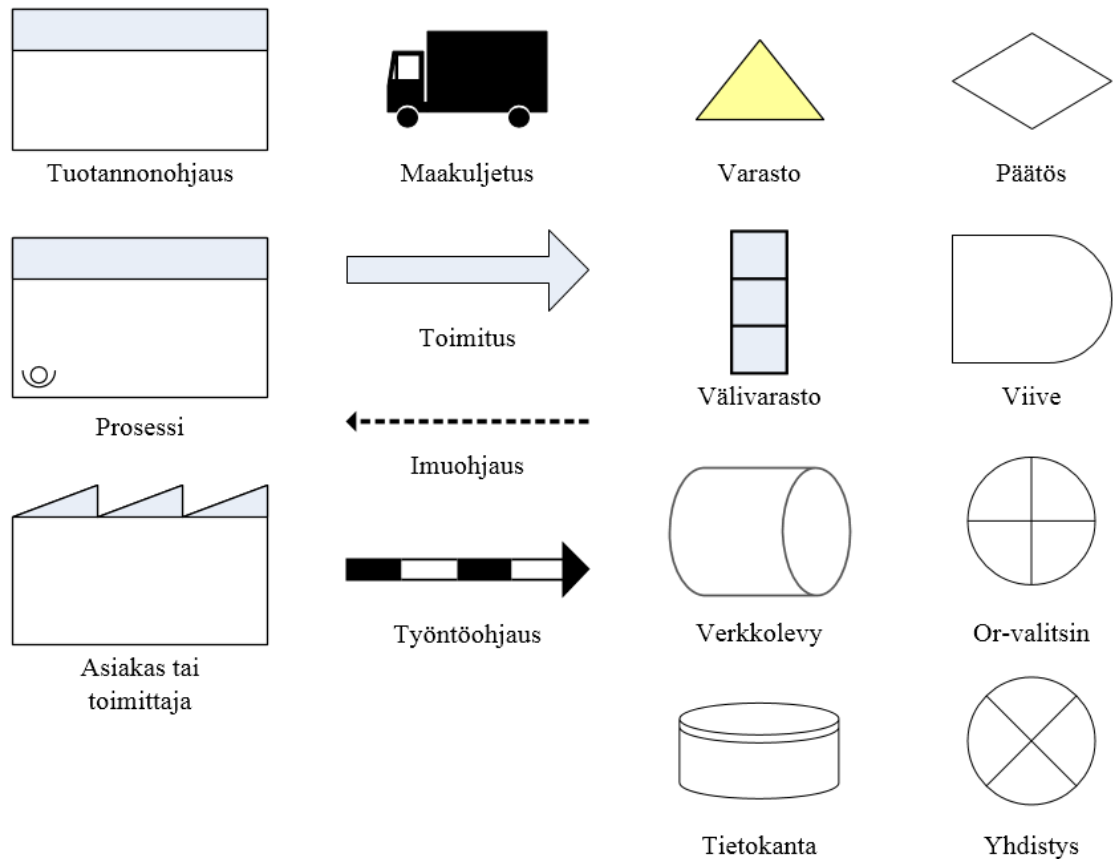
Lean Enterprise Institute on määrittänyt VSM:n tekemiseen ja toimintaan neljä eri askelta:

1. Valitaan tuote tai tuoteperhe ja määritellään sen prosessin ominaisuuksia.
2. Luodaan CSM, jonka pohjalta tutkitaan mahdollisia parannuskohteita.

3. Luodaan FSM, johon sisällytetään mahdollisimman paljon edellisen kohdan parannusehdotuksia.
4. Luodaan toimintasuunnitelma FSM:n implementointiin. [6]

Prosessia on tarkoitus toistaa inkrementaalisesti jatkuvan parantamisen periaatteiden eli kaizenin mukaan. Telestellä tätä ei tapahtunut, sillä jatkuvan parantamisen tavat olivat sopivuudeltaan poikkeavia. Siinä missä VSM:t yleensä antavat suurempina kehitysprojekteina ulostulonaan radikaalimpia muutoksia informaatio- ja materiaalivirtaan, pyrittiin yrityksessä tuomaan jatkuvana virtana pienempiä parannuksia eri työvaiheisiin, joista pitkillä aikaväleillä syntyy aikasäästöjä. Tällaisia parannuksia ovat esimerkiksi testausohjelmistojen sekvenssien turhien vaiheiden poistaminen tai uudelleenjärjestys.

VSM:t käyttävät Lean Enterprise Institutun standardoimia ikoneita, joilla kuvataan myös joitain tuotannollisia käsitteitä, kuten kaizenia tai tuotannossa käytettävää kanban-korttiohjausjärjestelmää. Kuvakkeet ovat osa myös Microsoftin Visio-diagrammiohjelman vakiopaletteja. Tässä työssä käytettiin sekä VSM- että perinteisiä prosessikulkukoneita (kuva 6).



Kuva 6. Liitteen 1 arvovirtakartassa käytettyjen ikonien selityksiä.

Vaikka testaus- ja valmistusympäristöjen suunnittelua voidaan standardisoida, tarvitaan VSM:t tehdä spesifimmin. Eri tuotteiden eri suunnitteluvaiheisiin kuulu poikkeavat määrät aikaa. Esimerkiksi voidaan ottaa moduulitestaussekvenssin työstö. Telestellä sekvenssin teko onnistui yksinkertaisimpien päivitysprojektien kohdalla kahdessa työpäivässä, mutta täysin uudenlaisten laitteiden projekteissa sekvenssin teko saattoi viedä muiden töiden lomassa useamman kuukauden. Koska yksittäisiä vaiheita ei voi verrata toisiinsa, ei voi myöskään projekteja tai niiden VSM:iä. Tehty CSM koski päivitysprojektia, jossa tuotteen uuden version valmistusympäristö vastasi hyvin lähelle vanhaa versiota. Tästä syystä testausympäristösuunnittelun eri vaiheiden kestot olivat lyhyitä.

Tuotantoa yleensä ajatellaan fyysisen esineen jalostamisena. Valmistusympäristösuunnittelu kuitenkin tuottaa pääosin sähköistä materiaalia. Koska suurin osa mudasta on enemmän prosessien järjestelyihin ja toteutukseen liittyvää, voidaan monia lean-työkaluja soveltaa myös immateriaaliprosesseihin. Esimerkiksi 5S-toimintamalli keskittyy fyysiseen ympäristöön viidellä metodilla:

1. sorteeraa: järjestetään tarvittavat ja tarpeettomat esineet
2. systematsoi: määritetään ja rajataan kullekin esineelle paikka
3. siivoa: siivotaan työpisteet säännöllisesti
4. standardisoi: edeltävistä kolmesta tavasta tehdään työpaikan yhteisiä käytäntöjä
5. seuraa: sovittujen standardien noudattamisesta pidetään huolta. [7]

Tällöin pääosin sähköisen prosessin tarkastelussa joudutaan käyttämään erilaisia työkaluja. Tässä VSM:n etuna on, ettei se ota suoranaisesti kantaa mudan tyypeihin vaan katselmoi suoraan arvoa tuottavaa aikaa verrattuna prosessin kokonaisläpimenoaikaan. Tämä saavutetaan prosessin eri työvaiheiden organisoinnilla, keskittymällä asianomaisten osastojen ongelmattomaan kommunikaatioon ja samanaikaiseen työskentelyyn eli concurrent engineeringiin. Eri mudan tyypeihin paneudutaan erikseen kappaleessa 5.3.

FSM:n tekemisen ydinajatuksena on karsia kaikki CSM:ssä näkyvät viiveet minimiin. Tehottomassa tuotantoympäristössä joudutaan tekemään radikaalimpia uudistuksia, jotta liikaa mudaa sisältävät askeleet saataisiin järjestettyä tehokkaammin.

Immateriaaliprosesseissa käsitteen alaa on laajennettava. Yksi helposti havaittava ongelmia tuottava pullonkaula on usein tarvittavien kriittisten tietojen keskittyminen tiettyihin työntekijöihin niin sanotusti laittaen kaikki munat samaan koriin. Tästä seuraa, että nämä työntekijät voivat olla ylityöllistettyjä tai heidän ollessa poissa syntyy tietotyhjiöitä ja tämän mukana mudaa. Poikkihallinnollisten ryhmien eli Cross-Functional Teamien (CFT) käyttäminen keventää paineita yksilöistä, sillä työskentelyn aikana tieto jakautuu tasaisesti sekä yksiköiden sisällä että niiden välillä. Vaikka ymmärrettävästi asiantuntijatasen tietous eli tiedon syvyys ei tule olemaan sama työntekijöiden välillä, olisi kuitenkin suotavaa, että usea ihminen tietää asioita laaja-alaisesti joskin yleistasolla. Jatkuvassa tiimityöskentelyssä nämä tiedot syventyvät ajan mittaan ja syntyy dominoefekti, jossa oppilaista tulee opettajia ja tieto leviää entisestään. Koska asiakkaan näkökulmasta henkilökunnan koulutukseen kuluva aika on mudaa, on arvovirran suhteen tehokasta suorittaa työläheistä koulutusta normaalin tiimityöskentelyn ohessa.

4 TUOTESUUNNITTELUN ELINKAARI

Tuotteen elinikää suunnittelun alusta tuotannon loppuun kuvaava Product Offering -prosessi eli tuotteenluontiprosessi on hyvä jakaa eri vaiheisiin ja etappeihin. Tällöin prosessista saadaan yhtenäinen ja universaalisti sovitettava kuvaus tapahtumien etenemisestä. Telestellä tämä prosessi oli jaettu kuuteen eri vaiheeseen, jossa jokaisella oli omat kriteerinsä joiden tuli täytyä ennen seuraavaan vaiheeseen siirtymistä. Tätä toimintatapaa kutsutaan vaihe-porttimalliksi eli stage-gate modeliksi. [8] Vaiheita merkattiin numeroidusti juoksevilla tunnuksilla E0–E6. Koko projektin käynnistävän E0:n kriteerin täyttää päätös tuotteen tarpeellisuudesta ja sen kehittämisen käynnistämisestä.

E1:een sisältyy projektinjohdon järjestäytymistä, alkupalavereja, uuden tuotteen tuotantoon ja testaukseen liittyviä arviointeja, tuotantosuunnitelman tekemistä sekä laitteen spesifikaatioiden mitoittamista. Projektisuunnitelman tulee olla tehtynä ennen seuraavaan vaiheeseen siirtymistä.

E2:ssa testausympäristön suunnittelun kannalta oleellisin etappi on moduuli- ja tuotetestausvaatimusten valmistuminen sekä testausympäristön suunnittelu. Mikäli tarvitaan uutta testauslaitteistoa, alkaa vertailu- ja hankintaprosessi tässä kohtaa. Laitteen ensimmäinen kytkentäkaavio valmistuu viimeistään toisessa vaiheessa, joskaan valmistuminen jo ensimmäisessä vaiheessa ei ole epätavallista. Kaavion olemassaolo helpottaa muita toisen vaiheen tapahtumia, sillä sen myötä komponenttilista ja täten luettelo laitteen valmistukseen tarvittavista osista eli Bill of Materials (BOM) on suurpiirteisesti olemassa. Oleellisimpien komponenttien, kuten esimerkiksi optisten osien tai virtalähteiden, pohjalta tehdään BOM-riskianalyysi, jossa näiden osien saatavuutta eri toimittajilta arvioidaan. Riskialttiutta nostavat muun muassa aikaisemmat huonot kokemukset toimittajan kanssa sekä muut alueelliset tekijät, kuten esimerkiksi tuotannon luotettavuus poliittisesti tai geologisesti epävakaaalla alueella. Samalla otetaan huomioon single source -komponentit eli osat joilla on olemassa vain yksi toimittaja. BOM:n pohjalta luodaan myös alustava osien kokonaishinta laitteelle, joka otetaan huomioon laitteen kustannusanalyysiä tehtäessä. Muita huomioitavia parametreja ovat esimerkiksi piirilevyn kerrosten määrä ja käytetyt juotosprosessit.

E3 keskittyy dokumentaation tekemiseen. Laitteen lopullinen nimi ja tarraspesifikaatio eli tuotetarroihin tulevat tiedot päätetään, tuotepakkaus suunnitellaan ja alustava ohjekirja työstetään. Kun tuotedokumentit ovat valmiina, projektinjohto hyväksyy ne. Valmistuneen testausvaatimuksen pohjalta valmistetaan myös prototyyppituotteet, joiden avulla testaussuunnittelua voidaan tehdä progressiivisesti paremmin lopullista tuotetta ajatellen. Kolmanteen vaiheeseen sisältyy myös paljon ME:n työtä. Toisessa vaiheessa valmistuneen moduulitestausvaatimuksen perusteella voidaan tehdä ohjelmistossa ajettava testaussekvenssi, kun taas uusimman piirilevypiirroksen mukaan rakennetaan testausjigi. Käytännössä nämä työstetään ajallisesti rinnakkain, jolloin niiden valmistuttua ohjelmistoa voidaan testata jigin kanssa ja suorittaa mittausspekvenssiin tarvittavia sovituksia, jotka ovat tarpeen jigin signaaliin aiheuttamien muutosten takia. Samaan aikaan ME alkaa tekemään testaus- ja tarravaatimusten mukaisesti tuotteen virityssivuja, lopputestaussivuja, työohjeita, ajureita ja muita merkintöjä kuten tarroja. Koska nämä kaikki pohjautuvat R&D:n tarjoamaan tuotekehitysmateriaaliin, tulisi vaatimusdokumentaation olla mahdollisimman huolitellusti tehtynä alusta alkaen.

E4 koostuu laitteen valmistelusta tuotantoon. Laitteen ulkoisesti ja sisäisesti ajettavat ohjelmistot ovat valmiit ja dokumentoidut. Testausympäristön tulee olla valmis ja laitteen täytyy käyttäytyä vaatimusten mukaisesti testilaitteistolla ja -ohjelmistolla. Testausprosessi täytyy olla dokumentoitu: lean-periaatteiden mukaisesti prosessien tulee olla mahdollisimman pitkälle standardisoituja. Kokonaisuudessaan ennen seuraavaa vaihetta tuotteen julkaisualustan täytyy olla valmis. Siihen liittyy myös tuotannon ulkopuolista aineistoa, kuten koulutus- ja markkinointimateriaalin olemassaolo ja esittelykappaleen saatavuus.

E5:ssä ajetaan pilottierä, jonka valmistuksesta tullutta palautetta tarkastellaan ja ilmenneitä ongelmia paikataan. Koesarjoja valmistetaan, kunnes tuotantolaadun katsotaan olevan riittävä. Mikäli laatu katsotaan puutteelliseksi, joudutaan mahdollisesti tekemään muutoksia kolmannessa vaiheessa olevaan testaussuunnitelmaan. Tuotteen BOM käydään läpi viimeisimmän laiteversion pohjalta. Tarkastelu on etenkin tarpeen, mikäli tuotanto tapahtuu ulkoistetusti. Jos tuotannolliset parametrit ovat kunnossa, voidaan laitteen myynti ja lopulta massatuotanto aloittaa.

E6 käsittää massatuotantopäätökseen liittyviä asioita. Uuden tuotteen valmistuksesta järjestetään koulutusta huollolle ja tuotannolle. Useimmiten koulutus tapahtuu valmiin materiaalin mukaan. Vain täysin uusista ja täten tuotannolle vieraista laitteista on

tarpeen valmistaa laitekohtaista erikoismateriaalia, jolloin myös koulutusperiodit pidentyvät ja syventyvät.

Onnistuneen Product Offering -prosessin sulkee loppukatselmointi, jonka tarkoitus on analysoida projektin onnistumista ja tuoda esille koko organisaatiota koskevia parannusehdotuksia. Vaihtoehtoisesti manufacturability studyssa eli valmistettavuuskatselmoinnissa voidaan todeta, ettei tuotteen valmistuskyky ole kunnossa. Tällöin aloitetaan uusi tuotekohtainen kehitysprojekti. Todennäköisin tuotannollinen ongelma on testausympäristön epästabiilius, jonka oikaisu on ME:n vastuulla. Mikäli tuote itsessään on syynä, otetaan ongelmat puheeksi R&D:n kanssa.

4.1 Tuotteen alkusuunnittelu ja arvioinnit

Uusi tuote saa alkunsa R&D-osastolla. Tuoteprojektin alussa järjestetään kick off -palaveri, johon OLSin puolelta osallistuu NPI-päällikkö. Palaverissa annetaan projektista pohjatiedot ja suoritetaan alustavaa työnjakoa. Mikäli aikaisempia vastaavia tuotteita on ollut, selvitetään niiden yhteydessä ilmenneitä tuotannollisia ongelmia, jotta ne voitaisiin huomioida tuotesuunnitteluvaiheessa. Tyypillisiä mekaanisia ongelmia ovat esimerkiksi ruuveihin liittyvät seikat. Mekaniikkasuunnittelija määrittelee, millaisia ruuveja kiristetään milläkin momentilla käyttötarkoitukseen perustuen. Ideaalitulanteessa ruuveja olisi mahdollisimman vähän erilaisia ja momentit olisivat yhdenmukaisia. Tällöin ruuvien ja vääntimien hankinta, säilöntä ja käsittely olisi mahdollisimman selkeää. Yhdenmukaistus näkyy myös hinnoissa. Mitä vähemmän erilaisia komponentteja on, sitä suuremmissa erissä niitä tilataan, jolloin kappalehinta yleensä laskee. Useammalle osakategorialle tarvitaan omat nimikekoodit ja hyllytila sekä ihmiset ylläpitämään niitä. Vääntimiä on myös säännöllisesti huollettava ja kalibroitava. Mitä samanlaisempia vääntimet ovat tyypiltään ja momentiltaan, sitä helpompia ja nopeampia niiden huoltotoimenpiteet ovat.

Virheanalyysiä tehtäessä voidaan myös käyttää ohjelmallisia työkaluja. Korjauspisteillä voidaan kerätä tietokoneohjelman avulla läpi menevien tuotteiden virhetyypit ja korjausajat. Tietokantaan tallennettua dataa voidaan myöhemmin aggregoida analyysiä varten. Telestellä käytössä oli talon sisällä kehitetty BADS (Board Assembly Defects Software) -ohjelmisto. Korjauspaikoilla käytettävällä ohjelmalla voitiin tallentaa korjattujen osien sijainti levyllä ja vikatyypit. Tietoa voitiin myöhemmin katsella

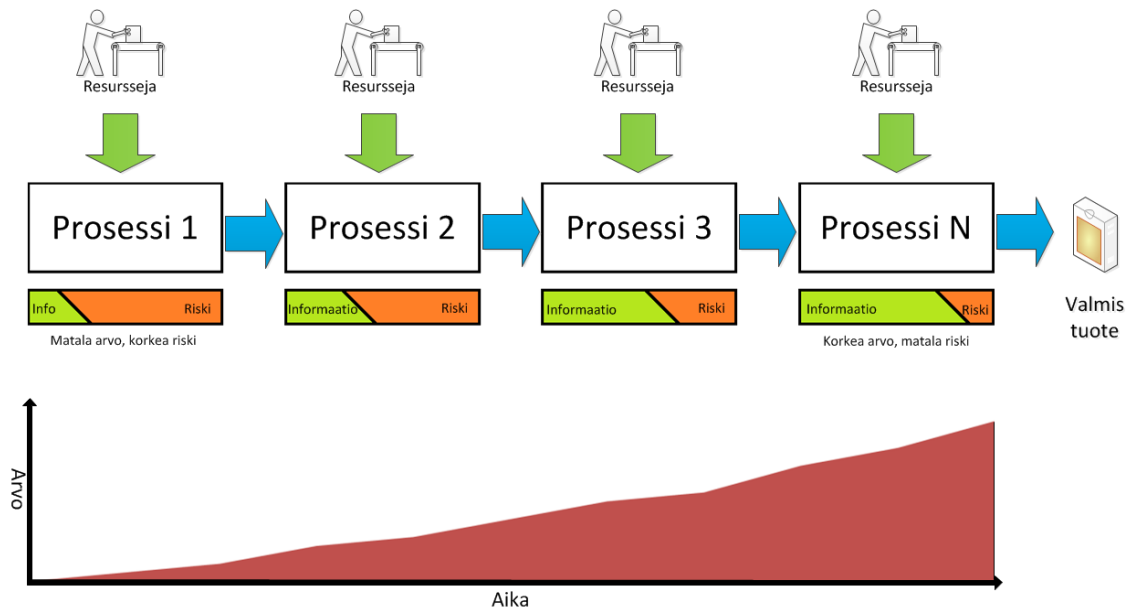
graafisesti ohjelmalla, joka piirsi moduulin ladontakuvan päälle korjattujen osien sijainnit.

Syy uuteen tuoterevisioon voi myös olla vanhan version ailahtelevuus spesifikaatioihin verrattuna. AC3200:n GaAs-versiossa vahvistus jäi odotetun alarajan lähelle. GaN-versiossa vahvistus paranee, jolloin edelliset tuotanto-ongelmat eliminoiduvat. Vahvistinkomponenttien vaihtaminen kuitenkin johtaa myös muihin suunnittelumuutoksiin, sillä liian suurella vahvistuksella myös säröt ja muut signaalin epätoivottavuudet voimistuvat.

Tuotteen toteutussuunnitelmaa laaditaan osa-alueittain: NPI katselmoi valmistettavuutta ja logistiikkaa, ME testattavuutta ja huolto-osasto huollettavuutta.

Valmistettavuusarviointi on kokonaisuudessaan visio siitä, miten tuotetta haluttaisiin valmistaa. Tämän lisäksi siihen kuuluvat myös logistiikan ja pakkaamisen arvioinnit. Luonteeltaan se muistuttaa palapelin kasaamista. Koska tuotanto koostuu eri vaiheista, täytyy tuotteelle valita oikeat tapahtumat oikeisiin vaiheisiin. Arvioinnissa syvennyttään myös teknisempiin yksityiskohtiin, kuten tuotteen mahdollisten kytkimien ohjaus ja moduulien ohjelmistojen asennus.

Testattavuusarviointi keskittyy yleensä testauksen maksimoimiseen tuotteen valmistusputken alkuvaiheessa. Alhaisen jalostusarvon tuotteessa ilmenneet viat aiheuttavat vähemmän hävikkiä kuin vasta myöhään havaittavat viat, sillä tuotantoriski vähenee saadun informaation kasvaessa. Tämän takia ladottua piirilevyä pyritään testaamaan mahdollisimman kattavasti heti sen saapuessa, sillä jokaisessa vaiheessa tuotteeseen upotetut resurssit kasvattavat jalostusarvoa (kuvio 1).



Kuvio 1. Jalostusarvon kehitys.

Testien kattavuudessa ilmenee tiedonsiirtotekniikoiden välillä eroja. Optiset tuotteet vaativat testeihin optisen lähettimen ja vastaanottimen, sekä esimerkiksi laserin jäähdätyksen asennuksen. Optisten laitteiden moduulit voidaan kuitenkin hyvin rajoitetusti testata ennen kokoonpanoa. RF-tuotteissa testit tehdään 75 Ω :n sisääntuloimpedanssia vastaan, kun taas optisissa laitteissa vastassa on fotodiodi, jonka jälkeinen sisääntuloimpedanssi vaihtelee taajuuden funktiona hyvin suuresti kilo-ohmialueella. Jigien suunnittelulla sovitusta saadaan parannettua, mutta epäsovitusta jää silti huomattavasti jäljelle. Tästä syystä moduulitestausvaiheen vasteet saadaan RF-tuotteilla huomattavasti tarkemmin kohdalleen kuin optisilla laitteilla, joilla testaus jää lähinnä perustoimivuuden tarkistamiseksi. Levytason testeissä myöskään maadoitusta ei ole toteutettu riittävän luotettavasti.

Testattavuusarvioinnissa tarkastellaan myös testauslaitteiston riittämistä niin laadullisesti kuin määrällisestikin. Tavallisesti pelkkien päivitysprojektien laitteet voidaan testata samassa laitteistoympäristössä kuin aikaisemmatkin versiot. Täysin uudentyyppisissä tuotteissa tarvitsee todennäköisesti tehdä jonkinlaisia laitehankintoja, jotta tarvittavat testit saadaan suoritettua. Tällöin saattaa myös olla tarpeen kalustaa uusia testausolosuhteita, joka taas vaatii referenssimittalaiteräkkien kasaamista ja konfiguroimista sekä fyysisen tilan järjestämistä.

Huollettavuuden kannalta paras optimointikohde on virhediagnosointi. Diagnostiikka on yleensä korjausaikaa eniten kasvattava vaihe. Komponenttien määrä kasvaa laitteiden saadessa uusia ominaisuuksia, ja vastaavasti tekniikan kehittyessä levykoko pienenee. Tiheässä komponenttiasettelussa tulisi olla selkeä lohkosuunnittelu, jolloin diagnostiikka helpottuu ja konkreettisen ongelmakohdan löytäminen piirilevyiltä nopeutuu. Mikäli huoltoajan takaraja lähestyy virheen löytymättä, joudutaan laite vaihtamaan uuteen, jolloin diagnosointiaika on todennäköisesti ollut kokonaisuudessaan mudaa. Huoltoajat saattavat olla huoltosopimusten määrittelemänä hyvinkin lyhyitä, asiakkaille kriittisten laitteiden kanssa vain muutama päivä. Sekä diagnostiikkaa että materiaalihävikin minimoimista helpottaa modulaarisuus. Moduulit jakavat laitteen selkeisiin pienempiin alueisiin, joiden testaus on suurta kokonaisuutta yksinkertaisempaa. Moduulit ovat myös vaihdettavissa yksitellen, jolloin muuten toimivia osia laitteesta ei tarvitse turhaan vaihtaa.

Komponenttien tiivis asettelu kuitenkin aiheuttaa haasteita levyn suunnittelulle ja käsittelylle. Mikäli komponentteja joudutaan asettamaan mekaaniseen rasitukseen altistuville paikoille, voivat komponentin liitokset murtua huomamaattakin. Yleensä yksittäisten komponenttien paikan vaihto ei kuitenkaan käy yksinkertaisesti. Pienikin muutos tarvitsee uuden piirilevyrevision ja tilauksen PWB-valmistajalta. Kriittisiä ongelmia varten on olemassa Engineering Change Request (ECR) -menettely, jonka kautta yhdellä osastolla havaittuun virhetilaan pyydetään korjausta siihen erikoistuneelta osastolta. Ei-kriittisiä ongelmia korjataan lähinnä suurempien muutosten, kuten kokonaisvaltaisten laitepäivitysten, yhteydessä.

Huoltotoimenpiteet käyvät helpommin, mikäli levy pidetään yksipuoleisena. Emolevy on usein valussa kiinni ruuveilla, jolloin levyn alla olevia komponentteja mitatessa tai vaihdettaessa emolevy on irrotettava. Nykyinen trendi siirtyä läpiladottavista komponenteista SMD-osiin auttaa tilannetta, sillä läpiladottavat komponentit joudutaan juottamaan levyn alapuolelta.

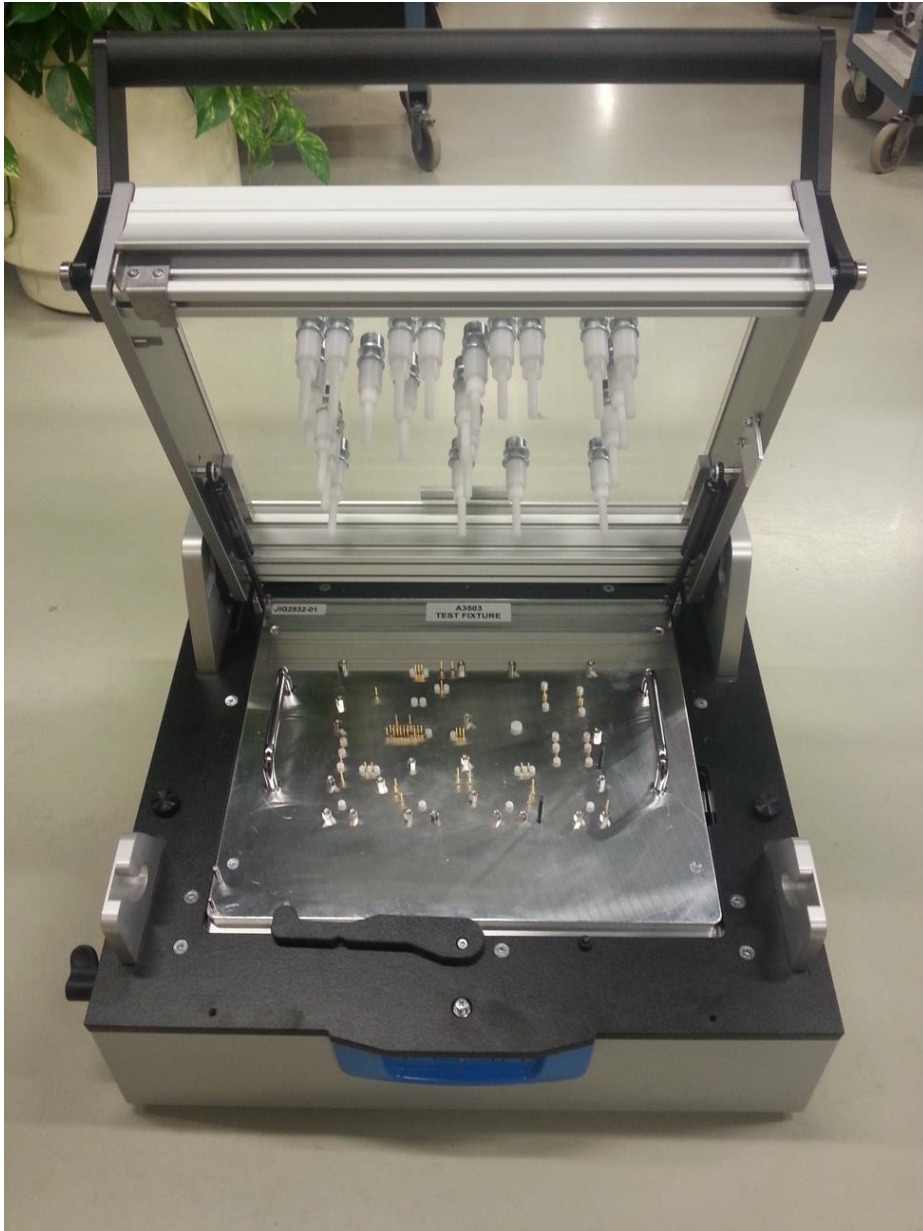
On otettava huomioon kiinnitetäänkö levyyn suojapeltejä tai muita komponentteja peittäviä osia. Yleisesti vioittuvia komponentteja tulisi sijoittaa levyllä näkyviin, ellei suojapellin tarpeellisuudelle ole syytä. Tällöin niiden vaihto käy mahdollisimman yksinkertaisesti, etenkin jos suojapelti on kiinnitetty juottamalla. Tuotannon korjaussolussa AC3200:sta vaihdettiin useimmiten SOT-115J-pakattuja galliumarsenidi-MESFET-vahvistinmoduuleita, jotka oli sijoitettu selville ja helposti käsiteltäville paikoille.

Solutuotannon ja varastointitapojen hallinnointi vaikuttaa huollon toimintatapoihin. AC3200:n eri osat eli valu, emolevy ja muut moduulit tuodaan Kiinasta. Suomessa tuote kootaan, viritetään, testataan ja pakataan. Koska tästä saatava base-tuote eli ilman asiakaskonfiguraatiota oleva tuote on ainoa varastoitava tuotetaso, varastossa ei ole valmiiksi testattuja ja konfiguroituja emolevymoduuleita. Mikäli huolto-osasto sellaisen tarvitsisi, joudutaan tilaus tekemään kokonaisesta tuotteesta. Koska usein vain jonkin moduulin vaihto on tarpeen, syntyy laitteen loppuosasta hukkaa, sillä yli jääviä osia ei voida palauttaa varastoon.

Kokonaisuudessaan alkuarvioinnit pohjautuvat menneisiin kokemuksiin. Aikaisemmista samankaltaisista tuotteista otetaan selville kaikki tuotannolliset ongelmat, joiden pohjalta arviointeja voidaan lähteä tekemään kehittävältä kannalta. Testaukseen data saadaan käymällä läpi vanhojen versioiden testausympäristöjä ja miettimällä uusien ominaisuuksien vaikutusta testaussekvensseihin ja laitteistokokoonpanoon. Valmistettavuuden kannalta tulee käydä läpi tuotantosoluista tullutta tuotannollisten ongelmien synnyttämää palautetta ja korjauspaikoilta kerättyä vikaantumisdataa. Huolto-osasto pitää kirjaa asiakkailta palautuneista laitteista ja niistä korjatuista vioista. Mikäli aikaisempia vastaavia tuotteita ei ole, joudutaan arviointeja tekemään paljon yleistävämmillä kriteereillä.

4.2 Tuotekehitys ja testaussuunnittelu

Alun tuotekehityksen valmistuttua R&D tuottaa ensimmäisen piirilevyversion. Tämän pohjalta tehdään prototyyppitilaus, jotta tuotetta päästäisiin testaamaan käytännössä. OLSin rooli tilauksessa on prototyypin tarkastus ja tilauksen hyväksyntä. R&D tekee prototyyppilaitteelle sopivan moduulin testausvaatimuksen. Moduulitestausvaatimuksella tarkoitetaan niitä parametreja, joita laitteen emolevystä tulee testata. Näitä ovat jännitteiden ja RF-ominaisuuksien mittaukset määrätyissä paikoissa. Koska emolevyt tulevat valmiina moduuleina Kiinan toimipisteestä Suomeen lopputuotteen kasausta ja testausta varten, joudutaan moduulille sopivat laitteet ja ohjelmistot valmistamaan Suomessa. Kun moduulitestausvaatimus on valmistunut ja toimitettu sähköisesti ME:lle, voidaan alkaa rakentamaan sille sopivaa testausympäristöä. Tähän kuuluvat laitteistopuolelta testausjigi (kuva 7) ja ohjelmistosta moduulitestaussekvenssi.



Kuva 7. AC3200:n emolevymoduulin testausjigi.

Jigin valmistus jakautuu kolmeen eri osaan. Ensimmäisenä suunnitellaan jigin piirilevy, josta tehdään tilaus. Levyn saapumista odotellessa valmistetaan työpajalla jigin piikkipeti eli aluslevy jolle on asetettu metalliset kontaktit määriteltyihin pisteisiin (kuva 8). Itse runko ostetaan ulkoiselta yhtiöltä. Levyn tulisi saapua samoihin aikoihin kuin kehikko valmistuu. Tällöin jigin kalustaminen eli tarvittavien johtojen ja liitinten kiinnittäminen levyllä ja kehikkoon voi alkaa saman tien.



Kuva 8. Piikkipeti.

Testaussekvenssin koodaaminen tapahtuu rinnakkain jigien valmistuksen kanssa. Useimmiten uusi moduuli pohjautuu vanhaan, jolloin myöskään sekvenssi ei muutu oleellisesti. Tällöin itse sekvenssin ensimmäisen version valmistaminen ei kestä kauaa, vain joitakin päiviä. Mikäli tuote on täysin uudenlainen esimerkiksi prosessoriltaan, saattaa sekvenssin valmistusaika pitkittyä useisiin kuukausiin. Ohjelmiston valmistuttua sitä testataan yhdessä jigien kanssa. Kuten kaikki muukin välissä oleva kaapelointi, myös jigi vääristää kulkevaa signaalia, joten sen vaikutus tulee sovittaa pois testausohjelmiston parametreissa. Yhteistestauksen jälkeen suoritetaan moduulitestauksen evaluointi. Tässä vaiheessa jigien saatetaan vielä tehdä muutoksia ilmenneiden ongelmien korjaamiseksi.

Moduulitestauspuolen valmistuessa R&D:ltä tulee myös valmiin tuotteen testausvaatimukset. Näiden pohjalta ME alkaa työstämään tuotteen valmistukseen ja testaukseen tarvittavia ohjelmistoja ja dokumentteja. Kokonaisuutena nämä jakautuvat viiteen osaan, joita tavallisesti tekevät kutakin eri työntekijät. Yksi osa on tuotteen virityssivut, joilla tarkoitetaan testiohjelmiston eri parametreja, esimerkiksi

vahvistuksen, kaltevuuden ja vasteiden kohde- ja raja-arvoja. Raja-arvoina toimivat väljimmillään tuotteen spesifikaatioiden ylä- ja alaraja, mutta usein arvot ovat näitä tiukemmat sillä asiakkaille ilmoitettuja rajoja ei saa ylittää. Toinen osa on tuotteen lopputestaussivut. Niissä määritellään valmistuneen tuotteen testattavat ominaisuudet, esimerkiksi erilaiset särö- ja kohinamittaukset. Vahvistimiin voi asettaa myös transponderin, jonka avulla suoritetaan transponderilinjan toimivuustestaus.

Kolmantena on tuotteen työohjeiden teko. Työohjeissa käydään vaihe vaiheelta laitteiden tarvittava kokoaminen, ohjelmointi sekä pakkaaminen. Jokaiselle vaiheelle tehdään oma ohjeensa: kokoonpano, viritys, mittaukset, ohjelmointi ja pakkaus. Ne tehdään yhtenä työurakkana Microsoft Visio -dokumenteiksi. Tuotannossa ne esitetään sähköisesti ohjelmistoon upotettuna. Sähköisyys mahdollistaa niin turhien paperien vähentämisen kuin reaaliaikaisen päivittämisen.

Neljäs kokonaisuus on ajurit. Ajuri on ohjelma, jolla testattavaa laitetta voidaan ohjata. Se avaa oikeat tietoliikenneportit ja suorittaa tarvittavat toiminnot laitteen ymmärtämällä komennoilla. Ajurit tehdään LabVIEW'illa, joka toimii saumattomasti testaussekvenssin TestStand-ohjelman kanssa. Koska laitteiden firmwaressa käytetyt komennot vaihtelevat tuoteperheiden sisällä harvoin, saattaa ajurin päivitys olla helpoimmillaan vain uusien komentojen lisäämistä yhteen arvotaulukkoon.

ME:n valmistautuessa prototuotteeseen viimeinen osio ovat tuotetarrat sekä laitekohtainen vaiheistus. Tuotetarroja on useita, mutta tässä kohdassa niistä täytyy suunnitella vain laitteeseen liimattava kansitarra. Sen sisällöt kirjoitetaan kaikille tuotteille yhteiseen tiedostoon, josta ne voidaan tunnisteiden avulla kopioida valmiiseen tarrapohjaan ohjelmallisesti. Muut tarrat, kuten sarjanumerot, tehdään ja tulostetaan vasta laitetta kasatessa. Laitekohtainen vaiheistus tarkoittaa sähköistä asetustiedostoa, jossa luetellaan laitteen käymät vaiheet ja niissä käytettävät testaussekvenssit. Vaiheisiin viitataan numerotunnuksin: 6_1 voi tarkoittaa viritystä ja 6_2 erinäisiä mittauksia. Tiedostossa myös kerrotaan muita testausohjelmien käyttämiä tietoja, kuten EEPROMin ominaisuuksia, kiteen taajuutta ynnä muuta vastaavaa.

Näiden myötä valmistusympäristö on käytännössä toteutettavissa. R&D tekee prototyypistä tilauksen, jonka ME tarkastaa ja hyväksyy. Prototyypilaitteita tuotetaan aina jonkin muutoksen tullessa, jotta testattavana olisi aina tuorein versio. Syitä muutoksiin voivat olla erilaiset testauksessa ilmenneet ongelmat, kuten vääristä komponenteista tai vedoista aiheutuneet ongelmat. Mikäli piirilevylle joudutaan

tekemään muutoksia, vaikuttaa se helposti moneen muuhun asiaan. Moduulitestausjigissä testipisteet ovat määriteltynä tiettyihin kohtiin, joten uusi piirilevy tarkoittaisi myös uuden jigin valmistamista. Myös testausohjelmistoja saatetaan joutua muuttamaan, mikäli prototyypimuutokset johtavat uusien parametrien mittaamiseen. ME on aina osallisena prototyyppien kokoonpanossa ja testauksessa. Koska R&D tekee testitulosten perusteella tuotekehitystyötä ja ME parantaa testausta, on yhteistyön oltava tiivistä.

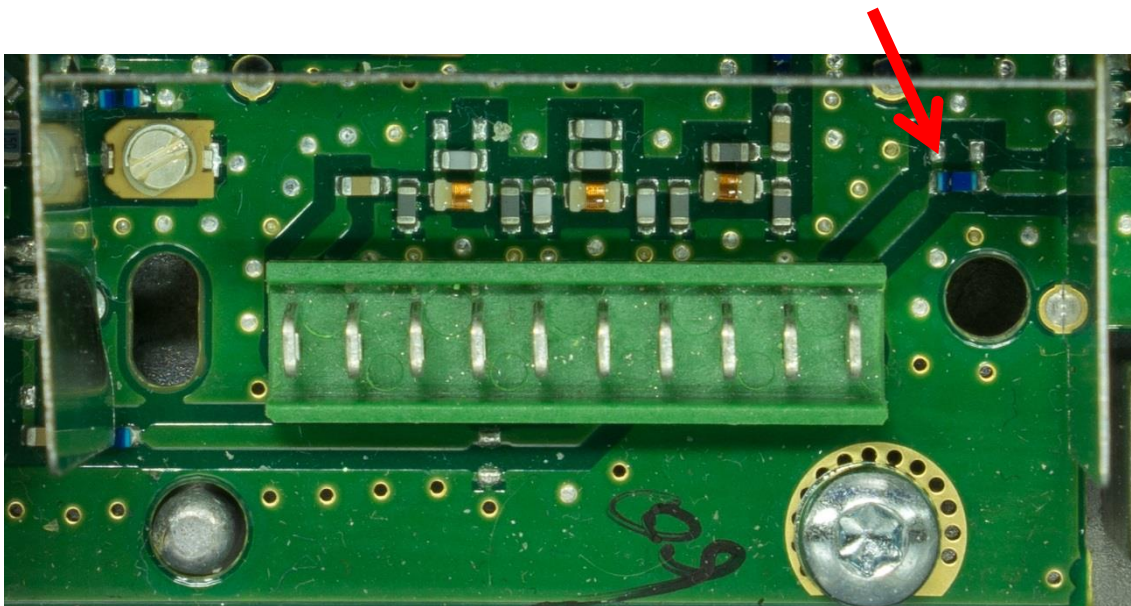
Hyväksytyyn prototyyppisarjan laaturaportin jälkeen voidaan tehdä pilottisarjatilaus, josta valmistetaan koesarjoja. Siinä missä prototyyppituotesarjan koko on joitain kappaleita, ovat koesarjat kymmeniä ellei jopa satoja tuotteesta riippuen. Vasta onnistuneiden koesarjojen jälkeen voidaan todeta tuotannollisuuden olevan kohdallaan. Tällöin tuotantosolulle tarkoitettua ohjelmistosta voidaan tehdä valmis asennuspaketti asennettavaksi ja ME voi ilmoittaa massatuotannon mahdolliseksi. NPI-prosessi loppuu ja massavalmistus alkaa.

5 TUOTEPROJEKTI

5.1 Projektin tavoitteet

AC3200:n päivitysprojektin tarpeet kumpusivat tuotannollisuusongelmien poistamisesta. Vahvistimessa käytetään 2:a GaAs-MESFET-vahvistinkomponentteja, joiden toleranssien vuoksi laitteen vahvistus jäi alle spesifikaation salliman rajan. Aikaisemmin tätä tuotannollisuusongelmaa oli hoidettu viemällä kokoonpanovaiheessa havaittu virheellinen laite korjauspaikalle, jossa vahvistinkomponentteja vaihdettiin sopivan yksilön löytämiseksi. Tämän operaation vuoksi laitteiden kokonaisläpimenoaikaan tuli huomattavasti hukka-aikaa. Ongelmaa aloitettiin purkamaan eri tyyppin vahvistinkomponentteja käyttämällä. Vahvistusongelmaa aloitettiin ratkaisemaan uudenlaisen tulovahvistinpiirin avulla. Uuteen laitteeseen haluttiin myös suunnitella GaAs-komponenttien tilalle GaN-komponentit. GaN-hybrideille oli ominaista huonommat vahvistusominaisuudet, mutta paremmat säröominaisuudet. Muutos oli myös markkinoinnin kannalta katsottu edistäväksi.

Projektin sekondäärisiä tavoitteita oli tehdä huolto-osaston havaitsemien virhetilanteiden mukaisia muutoksia. Huollon antaman virheanalyysin mukaan yleisin vikaantuva osa kaikissa eri vahvistimissa oli virtalähde, joten virtalähteen vaihtamisen helpottaminen olisi huomioitava. Osasijoitteluun liittyvä ongelma oli, että diplekserin jalustojen sisäpuolelle oli aikaisemmin sijoitettu komponentteja. Mikäli diplekseripistoyksikköä yritettiin asentaa laitteeseen virheellisesti, saattoi moduulin ohjuri osua ja irrottaa jalustan alueelle sijoitetun SMD-komponentin (kuva 9).



Kuva 9. Mekaanisesti riskialttiiseen paikkaan sijoitettu komponentti.

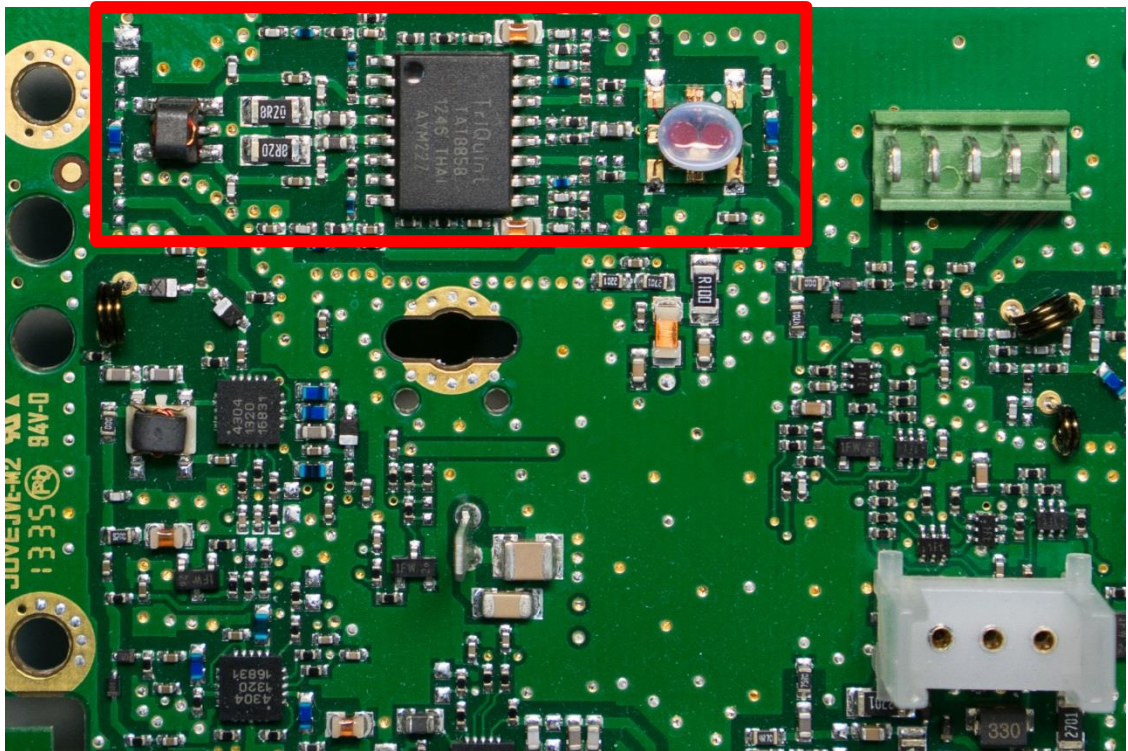
Laitteen päivitystä uuteen DOCSIS 3.1 -standardiin ei tehty. Yhtiön DOCSIS 3.1 -etenemissuunnitelma eli roadmap valmistui vasta projektin alun jälkeen, ja uusi 3.1-yhteensopiva vahvistin tuli olemaan kokonaisuudessaan eri malli.

5.2 Projektin eteneminen

Projektin aikana oli tarkoitus mitata eri vaiheiden kestoja arvovirtakarttaa varten. Projektin oli esitietojen perusteella tarkoitus olla yksinkertainen päivitysprojekti, mutta useat ennustamattomat vastoinkäymiset viivästyttivät projektia.

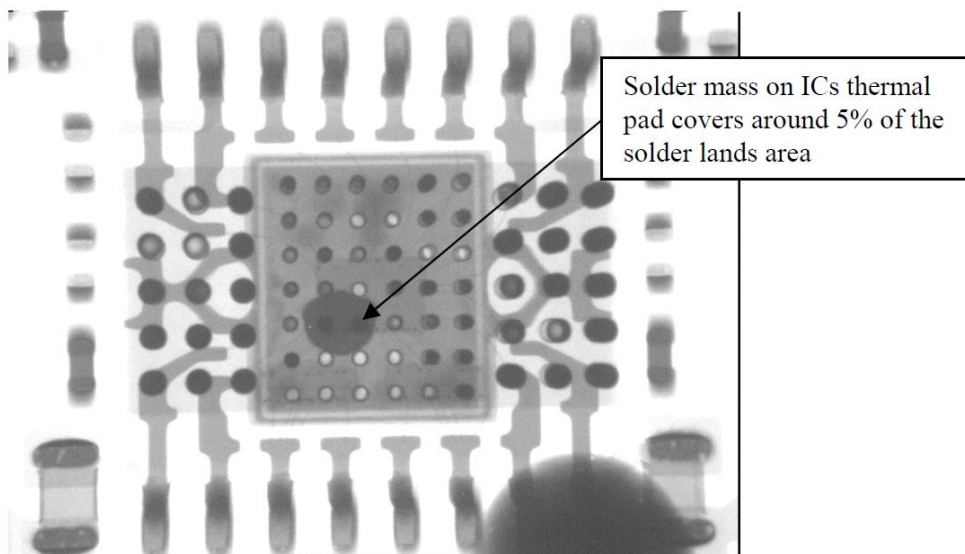
Ensimmäisessä revisiossa laitteeseen vaihdettiin lähtöihin GaN-hybridit sekä tuloon vaihdettiin vanhan hybridin tilalle uusi tulovahvistinpiiri. Signaaliominaisuudet olivat hyvät, mutta piirin pienen koon ja korkean tehon vuoksi jouduttiin miettimään uudenlaisia jäähdytysratkaisuja. Selvityksen tuloksena päätettiin vaihtaa tulopiiriä, sillä tehokas jäähdytys osoittautui lopulta mahdottomaksi.

Toisessa revisiossa tilalle vaihdettiin uusi tulopiiri (kuva 10), jossa oli edellistä piiriä huonommat säröominaisuudet mutta jota oli mahdollista jäähdyttää olemassa olevin ratkaisuin.



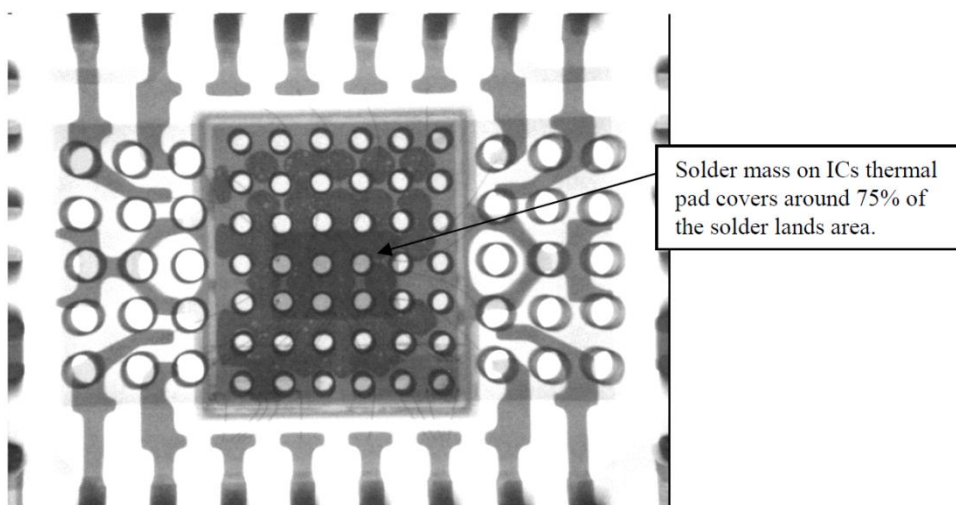
Kuva 10. Toisen protorevision tulovahvistinpiiri.

Ensimmäisenä vastaan tullut ongelma oli komponentin jäähdytys-elementin huono juottuminen, jolloin sekä lämmönsiirto että piirin maadoitus kärsivät. Röntgentutkimuksen mukaan juote levittyi vain noin 5 %:lle elementin pinta-alasta (kuva 11).



Kuva 11. Ensimmäinen röntgenkuva juotosongelmasta.

Ongelma oli havaittu myös valmistajalla. Ratkaisuna piirin profiilia oli madallettu, jolloin juote levittyi noin 75 %:lle pinta-alasta (kuva 12). Tämä katsottiin riittäväksi kontaktiksi.



Kuva 12. Toinen röntgenkuva juotosongelmasta.

Täysillä tehoilla ajettuna vahvistin toimi moitteetta, mutta tulovahvistinpiirin ja hybridien välistä välivaimennusta kasvattaessa syntyi ongelmia tulopiirin ollessa riittämätön.

Kolmannessa revisiossa käytettiin samaa tulopiiriä, mutta hybridikomponentit vaihdettiin takaisin GaAs-versioihin. Käytännössä alkutilanteeseen verrattuna laitteesta vaihtui tulovahvistinpiiri, piirilevyn osasijoittelu sekä käytetyt ferriittikomponentit. GaN-hybridien kannalta oleellisin ongelma oli, ettei ollut olemassa tulopiiriä jossa yhdistyivät sekä hyvä vahvistus että hyvät säröominaisuudet.

Projektin viivästyksen keskeisin syy oli muiden projektien viemä aika. Tuotteen pääsuunnittelijan aika sitoutui korkeammalle priorisoiuihin tehtäviin, jolloin kolmannen revision prototyypit jäivät seisomaan työjonoon. Myös R&D:n eri funktioiden välisiä pullonkauloja oli havaittavissa. Testausympäristön sekvenssien ohjailu vaati tietoja tuotekehityksen ohjelmisto-osastolta, jotka taas tarvitsivat toimivat prototyypit. Ohjelmistopuolella ei ollut työjonoa, eli vastaanotettu laite olisi mennyt viiveettä käsittelyyn ja siitä eteenpäin tuotantoputkea. Laitteistopuoli ei kuitenkaan saanut työnnettyä protolaitetta ohjelmistokehitykseen omien viiveidensä vuoksi.

Ensimmäinen prototyypisarja koostui kolmesta identtisestä laitteesta. Protosarjan tarkoitus oli tarkastaa tuotannollinen valmius eli valmistusympäristön toimivuus. Testausympäristö oli tällöin vielä lähes identtinen verrattuna laitteen edelliseen

versioon. Laitteen komponenttien ja sisäisten liittimien paikat muuttuivat, mikä aiheutti muutoksia kokoonpano-ohjeisiin. Testisekvenssiin tuli pilottierää varten muutoksia, mutta protosarja suoritettiin vanhan version sekvenssillä. Testeissä jotkin yksittäiset vaiheet epäonnistuivat esimerkiksi rajojen ylittävien vasteiden ja vahvistuksen vuoksi. Tällä ei kuitenkaan ollut suurta merkitystä protosarjan yhteydessä, sillä laitteen sähköisten ominaisuuksien hiominen oli tarkoitus jatkaa pilottisarjaan asti ja sen aikana. Lopullisen testausympäristön ohjelmistojen suunnittelu ja toteutus kävi yksinkertaisesti, sillä mittaukset olivat samat kuin vanhassa versiossa. Tämän takia uutta laitetta testatessa voitiin käyttää vanhoja sekvenssejä, joihin tehtiin pelkästään pieniä muutoksia eri vasteiden raja-arvoihin.

Projekti aikataulun epäsäännöllisyyden vuoksi aikamittauksia saatiin luotettavasti suoritettua vain jigin valmistuksesta. Taulukossa 1 on nähtävissä sekä liitteessä 1 mainitut työvaiheiden arviot että niiden toteutuneet kestot.

Taulukko 1. Työvaiheiden toteutuneita kestoja verrattuna ennakoarvioon.

Vaihe	Arvioitu kesto	Toteutunut kesto
Työjono R&D:ltä ME:lle	5 päivää	23 päivää
Jigin PWB:n kytkentäkuva	2 päivää	2 päivää
Jigin PWB:n suunnittelu	3 päivää	3 päivää
Jigin rungon toimitus	5 päivää	11 päivää
Piikkipedin valmistus	5 päivää	12 päivää
Vaiheet yhteensä	20 päivää	51 päivää

Tulokset eivät ole suoraan verrannollisia, sillä projekti tulkittiin kiireettömäksi, eli vaiheita ei suoritettu korkealle priorisoituna. Samaan aikaan työstettiin muitakin projekteja sekä töiden arkirutiineja. Jigin rungon toimitus kesti alkuoletusta kauemmin, mutta myös samaan aikapaikkaan sijoittuva piikkipedin valmistus venyi suurin piirtein yhtäläisesti.

Nykytilan kuvaamista varten NPI-prosessista oli tehtävä selkeä graafinen kartta (liite 1). Aikaisemmin selkeäksi ongelmakohtaksi oli mainittu, ettei R&D-osastolla ollut kovin selvää kuvaa ME-osaston toimista NPI-prosessin aikana. Aikaisemmat kuvaukset NPI-kaaresta löytyivät monimutkaisena Excel-tiedostona sekä vanhentuneena lohkodiagrammina, joten tieto ei ollut saatavilla ilman syvempää perehtymistä.

Uudessa kartassa selvennettiin myös eri immateriaalituotteiden varastointitapoja, imu- ja veto-ohjausta sekä yksittäisten prosessien tuloksiin pohjautuvia takaisinsyöttösilmukoita.

Tahtotilan graafinen kuvaus jätettiin tekemättä. NPI-prosessin monimutkaisuuden vuoksi radikaaleja muutoksia ei voitu tehdä. Sen sijaan työvaiheista ja -tavoista etsittiin parannuskohteita, joista kerrotaan tarkemmin luvussa 5.3. Tulevaisuuden painopiste oli CFT-työskentely ja concurrent engineering. Testaussuunnittelua oli jo kokeilumaisesti aikaistettu yhtäaikaiseksi tuotesuunnittelun kanssa, jolloin testaussuunnittelija on mukana tuotekehitysprojektissa alusta alkaen. Yleensä testausvaatimus muuttui jatkuvasti laitekehityksen myötä ja tällöin testaussuunnittelussa tehtiin jatkuvasti osittain turhaa työtä. Jatkovaa yhteistä kehitystyötä kuitenkin oli tapahduttava käytännön vakiinnuttamiseksi ja yhteistyön positiiviset vaikutukset olivat negatiivisia merkittävämmät.

5.3 Havaitut ongelmat Ronald Mascitellin hukcatekijöiden mukaan

Tuoteprojektin aikana Telesten työtavoissa tuli esille niin projektiin liittyviä kuin liittymättömiäkin ongelmakohtia. Kirjassa *The Lean Product Development Guidebook*, Ronald Mascitelli listaa kymmenen pahinta NPI-prosessin hukcatekijää:

1. sekava työympäristö
2. resurssipullonkaulat
3. tehtävien priorisointiongelmat
4. puutteellinen kommunikaatio yksiköiden välillä
5. puutteelliset tuotevaatimukset
6. arvaamattomat muutokset tuotevaatimuksiin
7. puutteellinen valmistettavuusarviointi
8. ylisuoriutuminen ja -analysointi
9. liialliset kokoukset
10. liiallinen sähköpostittelu. [9]

Telestellä lean-strategiaa implementoitiin lähinnä OLSissa, jolloin työtavat eivät olleet suoraan yhteensopivia R&D:n kanssa. Lean-periaatteet kuitenkin ovat hyvin yleispäteviä työympäristöön tai toimialaan riippumatta, joten niiden avulla voitiin

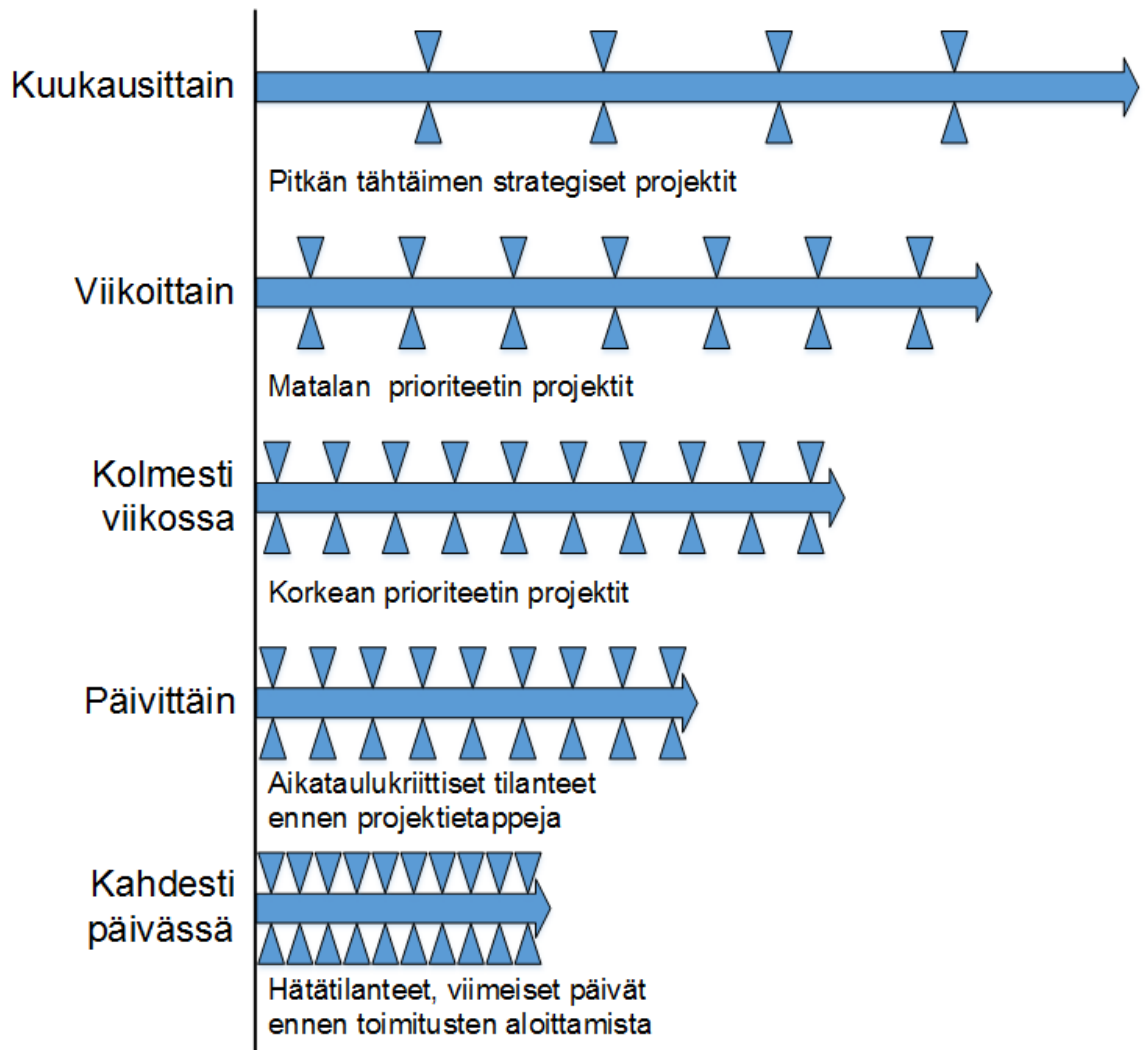
tarkastella niin OLSin kuin R&D:nkin toimintatapoja. Esitettyjä kohtia järjesteltiin sen mukaan, missä niistä koituva muda aiheutui.

5.3.1 R&D:n ongelma 1: Sekava työympäristö

R&D:ssä oli yleisesti ottaen epäsiistit työpisteet. Yhteisten mittalaitteiden sijaintia ei seurattu, eikä tiedetty voiko poissaolevan työntekijän pisteeltä ottaa tarvittavaa laitetta. Asian ratkaisemiseksi voisi käyttää 5S-metodeja järjestyksen tuomiseen ja pitää kirjaa yhteisen laitteiston sijainneista ja käyttöprioriteeteista. OLSissa laitteistoa oli huomattavasti enemmän ja niiden käyttöpaikkoja seurattiin aktiivisesti.

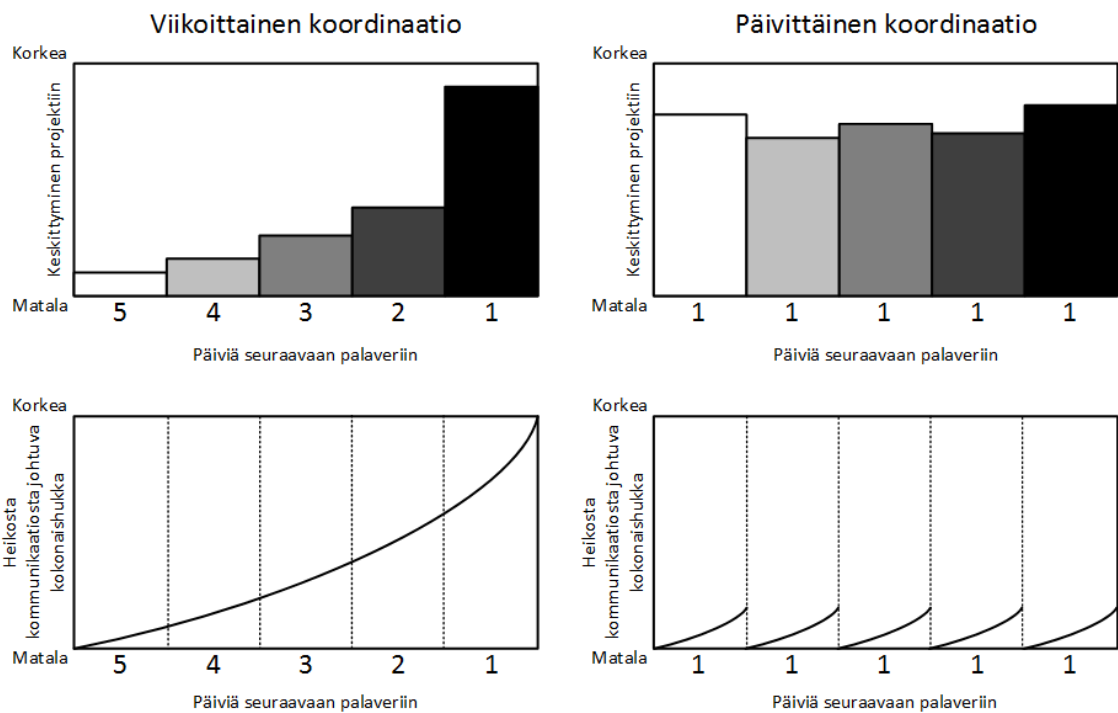
5.3.2 R&D:n ongelma 2: Resurssipullonkaulat ja tehtävien priorisointiongelmat

Nämä kaksi ongelmaa niputettiin yhteen koska niille katsottiin olevan yhtenevät syyt. Työkuorma oli jakautunut epätasaisesti yksittäisten suunnittelijoiden välillä. R&D:n eri funktioiden työjonot etenivät eri tahdissa. Joissain yhteyksissä syytettiin vanhanaikaisen aikataulutuksen toteuttamista: teknologia oli kehittynyt ja sen myötä tuotesuunnittelu oli monimutkaistunut, mutta aikatauluja rakennettiin samaan kaavaan kuin vuosia aiemminkin. Muran ja murin poistaminen tilanteista on yksinomaan yksittäisten projektijohtajien hallinnassa. Mascitelli ehdottaa toisessa kirjassaan, *The Lean Design Guidebook*issa, lyhyiden säännöllisten palavereiden eli stand-up meetingien käyttöä. [10] OLSin tuotannon puolella nämä olivat jo vakiintuneet käyttöön. Projektin ja sen yksittäisten vaiheiden prioriteeteista riippuen palavereita voidaan pitää eri tahdituksella (kuvio 2).



Kuvio 2. Esimerkki projektipalaverien jaksotuksesta. [10]

Kuten luvussa 4.2 mainittiin, lean-strategiat on rakennettava jokaiseen yrityskulttuuriin sopivasti. Palaverien tiheydestä ei siksi pystykään tekemään universaalia toimintakehystä. Pahimmillaan väärä implementointi voi tuoda lisää mudaa liiallisen kokoustamisen muodossa kuitenkin ratkaisematta ongelmaa. Strategisten palaverien määrää ja kestoja olisi kuitenkin tarkasteltava tietotyhjiöiden estämiseksi ja toiminnan yleisen sulavuuden edistämiseksi. Lyhyillä ja päivittäin pidettävillä palavereilla on kaksi etua. Ensinnäkin, ihmisillä on yleisesti ottaen taipumus jättää työt lähelle deadlinea. [11] Mascitellin mukaan tasaiset stand-up-palaverit tasoittavat työkuormaa tasaisemmin ajan mittaan. Toisennakin, päivittäiset palaverit estävät tietotyhjiöistä johtuvaa mudaa (kuvio 3).



Kuvio 3. Esimerkki projektipalaverien jaksotuksesta. [10]

Mascitelli kertoo haluavansa lyhyissä stand-up-palavereissa vastaukset kolmeen kysymykseen:

1. Mitä on saatu aikaan sitten viime palaverin?
2. Mitä tullaan saamaan aikaan seuraavaan palaveriin mennessä?
3. Mitä apua kollegoilta tarvitaan?

Tällaista rakennetta noudattavia stand-up-palavereita hyödynnetään myös agile-kehitykseen kuuluvan scrum-viitekehyksen päivittäisissä palavereissa. [12] Liiallinen yksityiskohtiin pureutuminen ei ole tuottavaa, vaan se tulisi hoitaa henkilökohtaisissa keskusteluissa kohdan 3 mukaisesti. Yhteiset palaverit tulee pitää yleistasolla.

5.3.3 R&D:n ongelma 3: Arvaamattomat muutokset tuotevaatimuksiin

Projektin kannalta suurin mudan tuottaja oli tuotevaatimuksen muutokset. Asiaan on kuitenkin hyvin vaikea puuttua, sillä epäonnistuneiden prototyyppien suunnittelu on osa R&D:a ja ongelmat ovat tapauskohtaisia. Kyse ei siis ole niinkään virheestä vaan ominaisuudesta, joka tulee hyväksyä prosessissa. Asiaa voi kuitenkin lievittää pitkällä

aikavälillä. Samoja virheitä ei tarvitse toistaa, mikäli samat suunnittelijat työskentelevät samojen tuoteperheiden parissa. Telestellä yleisimpien tuoteperheiden suunnittelussa oli 15–20 keskenään kutakuinkin yhdenvertaista suunnittelijaa, kun taas erikoistuneempien laitteiden suunnittelutietämys rajoittui vain 1–2:een suunnittelijaan. Tieto ongelmatilanteista liikkui suusanallisesti. Problemaattista tilanteen suhteen on mahdollinen arvaamaton työntekijöiden vaihtuvuus, jolloin vanhaa tietoa ei ole enää saatavilla. Työdokumentaatiota R&D:ssä ei tehty, vaikka erinäisiä loppuraportteja pilottisarjojen suhteen oli harkittu. R&D:n näkemyksen mukaan dokumentointiin menevä aika oli yhtä suuri tai suurempi kuin se aikahukka mitä sillä yritettäisiin ratkaista. Lean-periaatteiden pohjalta dokumentointi olisi kuitenkin suositeltavaa. Jonkin tuoteperheen dokumentaatioissa oleva maininta menneestä ongelmasta ja sen ratkaisusta saattaa olla tulevaisuudessa relevanttia tietoa ja säästää turhaa työtä. Kirjatun tiedon mahdollista tarpeettomuutta ei tulisi käyttää tekemättömyyden syynä. Uudelle metodille voitaisiin määrittää kokeilujakso, jolloin se olisi osa normaaleja työtapoja. Kohdattujen ongelmien vuoksi tapaa ei kannata tyrmätä, vaan tulisi katselmoida esimerkiksi dokumentaation syvällisyyden tarpeellisuutta. Jo pintapuolinen ja nopeasti tehty dokumentaatio voi olla avuksi.

5.3.4 ME:n ongelma 1: Sekava työympäristö

Tuotantolattialla 5S-käytännöt olivat hyvin implementoituja ja noudatettuja. ME-suunnittelussa työpisteet olivat joiltain osin sotkuisia. Monia kaappeja ei pidetty järjestyksessä vaan ne toimivat epäjärjestelmällisenä varastotilana. Työalue itsessään oli jatkuvan muutoksen alla. Pinta-ala oli osan ajasta hyvin pieni verrattuna työntekijämäärään, mutta tilannetta korjattiin myöhemmin laajentamalla osaston pinta-alaa. Kehotuksia työpisteiden siivouksesta jaettiin lähinnä arvovaltaisten vieraiden saapuessa. Asian korjaus vaatii pysyvämpää asennemuutosta 5S:n neljännen ja viidennen askeleen eli standardisoinnin ja seurannan mukaisesti. Asian korjaamiseen ehdotettiin jokaisen henkilökohtaiselle pisteelle viikoittaista siivousaikaa. Yhteisten säilytystilojen, kuten kaappien, järjestely voitaisiin allokoida osastolla kulloinkin työskentelevälle harjoittelijalle. Koska kaappien käyttöaste on vähäisempi, ei niiden järjestelyä tarvitse tapahtua viikoittain vaan esimerkiksi kuukausittain. Kaappien sisällöt tulisi järjestää hyllyittäin ja kaapin sisällöstä ilmoittaa esimerkiksi oveen liitettävällä listalla. Jos järjestyksestä pidetään jatkuvasti huolta, pitäisi kokonaissiivousajan vähetä kerta kerrasta.

5.3.5 ME:n ongelma 2: Puutteellinen kommunikaatio yksiköiden välillä

R&D:n aikatauluongelmat säteilivät myös ME:n puolelle. Joissain tapauksissa prototyyppi oli yhtäkkisesti saapunut testaussuunnittelijalle, jolloin hänelle ei jäänyt aikaa sopeuttaa projektia omaan työjonoonsa. Telestellä oli testausympäristön suunnittelussa hyödynnetty ketterää kehittämistä (eng. agile development). Yksi ketteryyden keskeisistä teemoista on nopea mukautuminen muutokseen ennalta-asetetun suunnitelman noudattamisen sijaan. [13] Tämän ajatuksen liian fanaattinen implementointi kuitenkin aiheuttaa muria, jolloin tilanne ei ole holistisesti katselmoituna optimaalinen. Sen sijaan kullakin projektiin osallisena tulisi olla selkeä käsitys projektin nykytilanteesta. Usean työntekijän valituksena oli, ettei kokousten päätöksenteko ollut tarpeeksi jämäptä, joka vaikutti sekä palaverien laatuun että pituuteen. Osana Mascitellin kokouskäytäntöjä oli aikarajojen asettaminen. Ennen kokousta kaikilla tulee olla tarkka tieto kokouksen kestosta ja päätöksiä tulee tehdä aikarajan mukaan. Käytännössä asiat tulevat etenemään joustavammin, mutta säntillisempi tempo asettaa oman paineensa kokouksen etenemiselle.

5.3.6 ME:n ongelma 3: Puutteelliset tuotevaatimukset

Koska ajurit tehtiin ME-suunnittelussa ja laitteen sisäinen ohjelmisto R&D:ssä, vaati sujuva kehitystyö toimivaa dokumentointia ja kommunikaatiota. ME:n tekemisiin ajureihin laitettiin sisään komentoja ASCII-merkkijonoina, jotka ajurissa käännettiin tavuiksi eli heksakoodiksi. Eri numeeriset arvot kuitenkin käyttivät eri tarkkuuksia, eli yksi asetettava arvo saattoi olla yhden ja toinen kahden desimaaliluvun tarkkuudella. Koska desimaaliarvot olivat liukulukuina, ne täytyi muuttaa kokonaisluvuiksi ennen heksakoodiksi kääntämistä. Tähän tarvittiin dekadimuotoinen kerroin, jotka tallennettiin joko ajurikoodin sisäiseen taulukkoon tai ulkoiseen ini-asetustiedostoon. Esimerkkinä, ini-tiedostossa olevan merkinnän `COMMAND_A = "0x0A;0.1*[INT8]"` mukaan lähetettävä merkkijonokomento "COMMAND_A" tarkoitti komentoa "0x0A;0.1*[INT8]", joka oli formaattia "A;B*C", jossa A oli heksa-arvona se parametri jota halutaan muuttaa, B oli kerroin ja C oli tietotyyppi, tässä tapauksessa 8-bittinen eli tavun pituinen kokonaisluku. Asetettava arvo siis kirjoitettiin parametriin 0x0A. Nämä parametrit olivat listattuna laitteen ohjelmistoon, eli 0x0A saattoi tarkoittaa esimerkiksi jotain vahvistusparametreista. Asetettava luku kerrottiin kymmenesosalla, eli esimerkiksi

10:stä tulee 1. C viittasi sisääntulevaan tietotyyppiin, jotta ajuriohjelmisto osasi käsitellä lukua oikein. Halutut tarkkuudet eli tarvittavat kertoimet (B) eivät olleet aina selviä, ellei niitä oltu dokumentoitu riittävän hyvin. Jos kertoimia jouduttiin tiedustelemaan ylävirtaan R&D:ltä, tuli ajurin valmistumisessa turhaa mudaa odotusajan muodossa. Tuotteesta ja R&D:n ohjelmiston dokumentoijasta riippuen tätä ongelmaa esiintyi aika ajoin. Koska dokumentoinnin laatu tuntui olevan henkilöstä riippuvaa niin tämän ongelman ratkaisuun riittäisi R&D:n yhteisten dokumentointivaatimusten laatiminen.

Toinen tämän kategorian alle asettava ongelma liittyi osaston omiin standardeihin sen toimittaman ohjelmiston suhteen. Ohjelmistot olivat yleisesti ottaen toimivia, mutta sen dokumentoinnissa oli puutteita. Koodin kommentoimisen puutteesta koitui mudaa. Ongelmasta ja sen ratkaisusta kerrotaan tarkemmin luvussa 6.

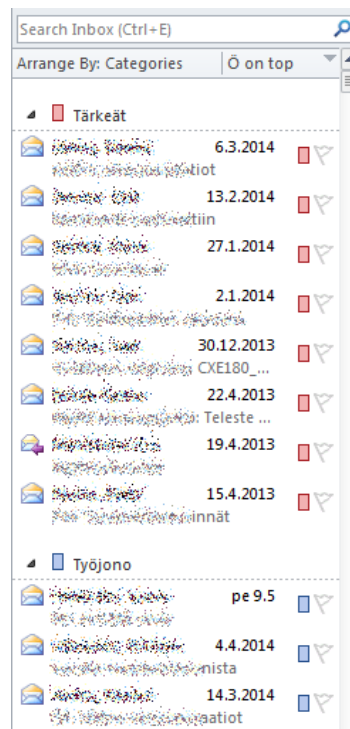
5.3.7 ME:n ongelma 4: Ylisuoriutuminen ja -analysointi

Joissain tapauksissa testausvaatimukset olivat liian perusteelliset. Sekvenssissä saatettiin testata sellaista parametria jonka toimimisen edellyttämä testaus on testattu jo sekvenssin aikaisemmassa vaiheessa. Toisin sanoen, jos myöhemmin testattava parametri olisi virheellinen, sekvenssi ilmoittaisi virheestä jo aikaisemmassa vaiheessa. Esimerkiksi AC3200:n testaussekvenssissä paluusuunnan särömittaukset olivat aikaisemmin toimivuusmittauksia, sillä olemassa olleilla analogisilla testauslaitteilla tarkempaan mittaukseen vaadittavaa kuormaa ei saatu aikaiseksi. Mittauksen kannalta kriittisten transistorien toimivuus kuitenkin ilmenee jo aikaisemmin tehtävissä vastemittauksissa. Jotkin yksittäisistä testeistä olivat selvästi tuotekehitystä varten, mutta niitä silti haluttiin tuoda myös tuotantoon testauksen perusteellisuuden nimissä.

ME oli useasti aikaisemmin kritisoinut R&D:ä käytännössä turhien testien vaatimisesta. R&D:llä oli kuitenkin painavampi sana asian päätöksen suhteen. Ongelma saattoi olla syvemmällä yrityskulttuurissa, joten sen korjaaminen oli vaikeaa. CFT:t ja testaussuunnittelun mukaanotto aikaisemmin NPI-kaaressa oli hyvä alkuaskel. Testaussuunnittelijan läsnäolo ja vaikuttaminen tuotekehittäjien ympäristössä alensi kynnystä kysymyksille ja laski osastojen välisiä ennakkoluuloja osaamisen ja toimintatapojen suhteen. Osastojen tiiviimpää yhteistyötä tulisi jatkaa ja integroida pidemmälle.

5.3.8 ME:n ongelma 5: Liiallinen sähköpostittelu

Sähköpostien määrä oli sopivalla tasolla mutta frekvenssi hyvin epätasainen. Pahimmillaan yksittäisen päivän aikana saattoi tulla kuusi lyhyttä yhteisiin töihin liittyvää sähköpostia, joka edusti yli puolta koko viikon aikana testaussuunnittelijoille lähetetyistä yhteisistä sähköposteista. Liiallinen informaatiotulva voi johtaa välinpitämättömyyteen sähköposteista, jolloin relevanttia tietoa voi jäädä huomiotta. Ongelma ei ole yksiselitteinen, eikä tarpeellisten sähköpostien määrä ole aina ennustettavissa. Määrää ei myöskään pidä karsia pelkästään paljouden vuoksi, koska nopea tiedonkulku on oleellista tehokkaan työnkulun vuoksi. Sen sijaan vastuuta voi vierittää lukijalle. Telestellä käytetyssä Microsoft Outlook -sähköpostiohjelmistossa on mahdollista määrittää käyttäjän mukauttamia värikoodillisia kategorioita (kuva 13).



Kuva 13. Microsoft Outlookin mukautettuja sähköpostikategorioita.

Merkitsemällä sähköposteja relevantteihin kategorioihin pitää tiedon ja sähköpostilla välitetyt työnannot järjestyksessä ja helposti löydettävissä. Yhteisiä sähköposteja tulee kirjoittaa niin, että niistä selviää mahdollisimman helposti oleellinen sisältö. Tällöin kukin postilistalla oleva voi tehdä henkilökohtaisen päätöksen tiedon tärkeydestä ja asettaa postin oikeaan kategoriaan mikäli tarve.

6 KOODIN KOMMENTOINTI

Yksi selkeistä esiin tulleista parannuskohteista oli koodin kommentointi. Etenkin nuorempien testaussuunnittelijoiden kohdalla oli yleistä, että jonkin koodin funktionaalisuutta tutkiessa vastaan tuli epäselviä kohtia joiden toiminnan kulku ei ollut yksinkertainen. Tällöin yleisin ratkaisu oli ensin käyttää omaa aikaa koodin tutkimiseen mutta lopulta päätyä kysymään asiasta joltain joka on joko kehittänyt tai käyttänyt kyseistä koodia aikaisemmin. Tällöin hukka-aikaa oli kertynyt ensin työntekijältä A ja sen jälkeen samanaikaisesti työntekijältä A sekä työntekijältä B. Pahimmillaan epäselvyyksien selvittämiseen saattoi mennä puolikin tuntia, jolloin hukka-aika nousi jo lähelle kokonaista työtuntia. Yksinkertainen ratkaisu radikaalisti lyhentämään tai jopa kokonaan eliminoimaan tätä hukkaa oli koodin kommentointi. Jo pienellä ytimekkäällä kuvauksella koodipalan tai VI:n toiminnasta saatiin riittävän tarkka käsitys, jotta VI:n sisältävän sekvenssin työstäminen pystyi jatkumaan sulavasti.

6.1 Projektin esiselvitys

Kommentointikäyttäytymisestä järjestettiin anonyymi työntekijäkysely (liite 2), joka jaettiin kaikille LabVIEW-ohjelmoinnin kanssa säännöllisesti työskenteleville, yhteensä 11 henkilölle. Tulokset näkyvät taulukossa 2.

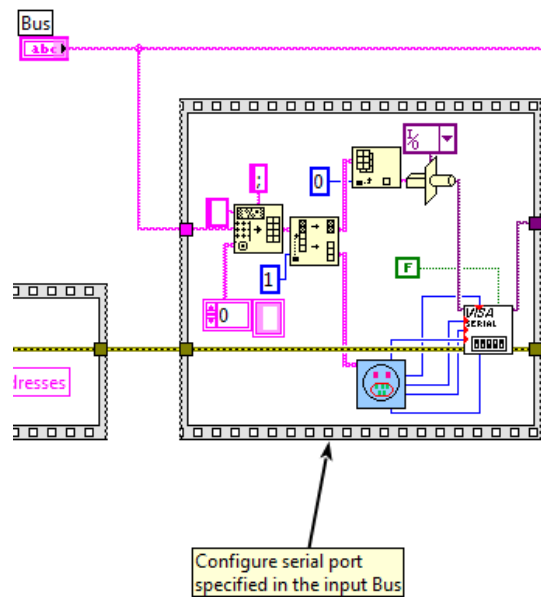
Taulukko 2. Kommentointikyselyn tuloksia.

Kysymys	Kyllä	Joskus	En/Ei	Muu/EOS	Yhteensä
1.	5 (45 %)	6 (55 %)	0 (0 %)	-	11 (100 %)
2.	10 (91 %)	-	0 (0 %)	1 (9 %)	11 (100%)
3.	0 (0 %)	-	10 (91 %)	1 (9 %)	11 (100 %)
4.	5 (45 %)	-	6 (55 %)	-	11 (100 %)
5.	3 (27 %)	-	5 (46 %)	3 (27 %)	11 (100 %)

Vastausten perusteella oltiin lähes yksimielisiä siitä, että kommentointi on tarpeellista, mutta sitä ei tehdä tarpeeksi. Esille tuotiin vahvasti yhteisten toimintatapojen puute.

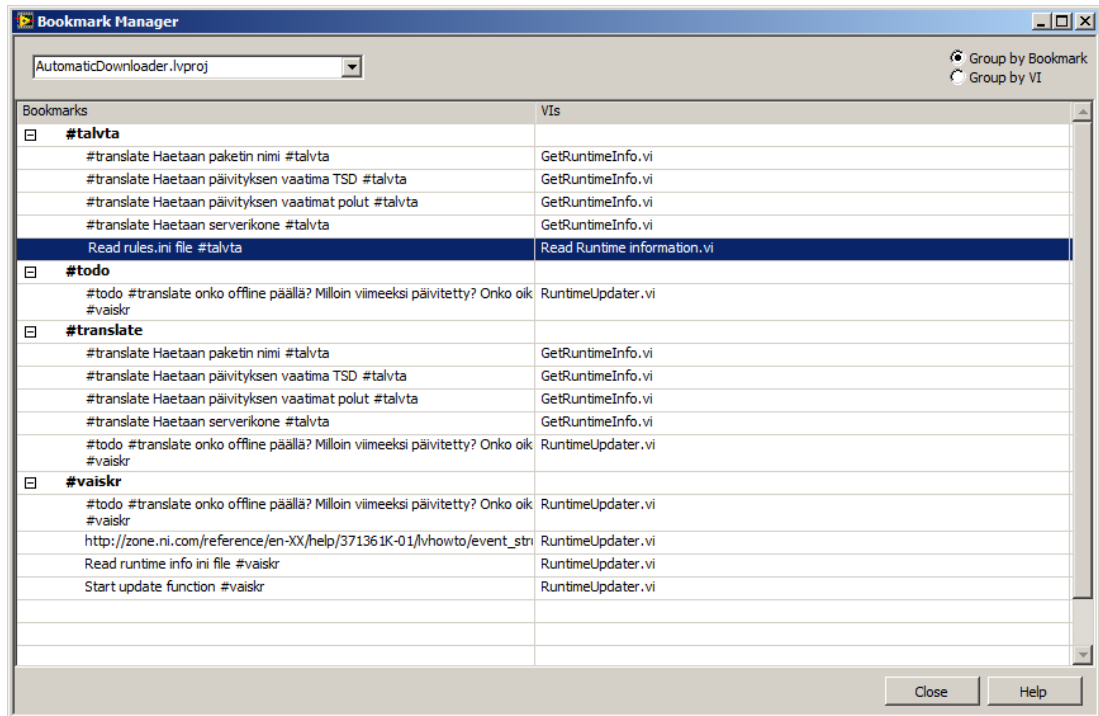
6.2 Kommentointispesifikaation rakentaminen

LabVIEW 2013 toi edelliseen versioon nähden uusina ominaisuuksina muun muassa kommentointia helpottavia työkaluja. [14] Kommenttilaikoita voidaan visuaalisesti linkittää nuolella niihin liittyviin aliohjelmiin tai struktuureihin, jolloin asemointia muutettaessa kommentit eivät eksy itse ali-VI-palikoista (kuva 14).



Kuva 14. Linkattu kommentti.

Kommentteja pystyy myös korvamerkitsemään eli tagaamaan avainsanoilla, jolloin ne ilmestyvät järjesteltyinä omassa kirjastossaan (kuva 15). Tätä kirjastoa voi pitää eräänlaisena virtuaalisena JAPA-tauluna, josta työntekijät voivat rutiinomaisesti käydä tekemässä koodeihin tagattuja muutoksia. Järjestelyn toimivuus kuitenkin vaatii yhtenäisiä käytäntöjä ja avainsanalistoja, joka kävi selville myös kyselyn tuloksissa, joissa yhteisesti sovitut kommentointiperiaatteet tuotiin esille seitsemässä palautteessa.



Kuva 15. LabVIEW 2013:n tagikirjasto.

Kommentoimattomuutta kysyttäessä yleisin syy oli kiire. Yleistä oli myös jättää kommentoimatta itsestään selväksi katsottu toiminnollisuus. Esille tuli myös, ettei kommentoimattomia koodia selvitettäessä tai editoitaessa kommentteja myöskään lisätä uusiin tai olemassa oleviin osuuksiin. Yksimielistä oli kuitenkin että kommentointi on tarpeellista, vaikka joukossa olikin yksi ”joskus”-vastaus. Parannusehdotuksissa otettiin esille tarve rajanvedolle siinä mitä on aiheellista kommentoida ja mitä ei. Yhdessä palautteessa katsottiin mahdolliset katselmointipalaverit tarpeellisiksi. Toisessa palautteessa ongelmalliseksi miellettiin myös kielimuuri, sillä dokumentoinnin pääkieli on englanti.

Kyselyn ja todetun tarpeen perusteella kommentointiohjeistossa oli siis otettava kantaa

1. kommentoinnin laajuuteen ja vähimmäisvaatimuksiin
2. vanhojen koodien kommentointiin
3. tagimääritelmiin.

Teknisesti tagit määritellään etumerkillä #. Esimerkiksi jonkin ohjelmistovirheen sisältävään koodiin voidaan lisätä kommenttina kuvaus bugista ja lisätä loppuun ”#bug”. Tällöin merkintä näkyy tagikirjastossa kyseisen osion alla. Toiminto on nykyään tuttu myös useista sosiaalisista medioista, kuten Twitteristä.

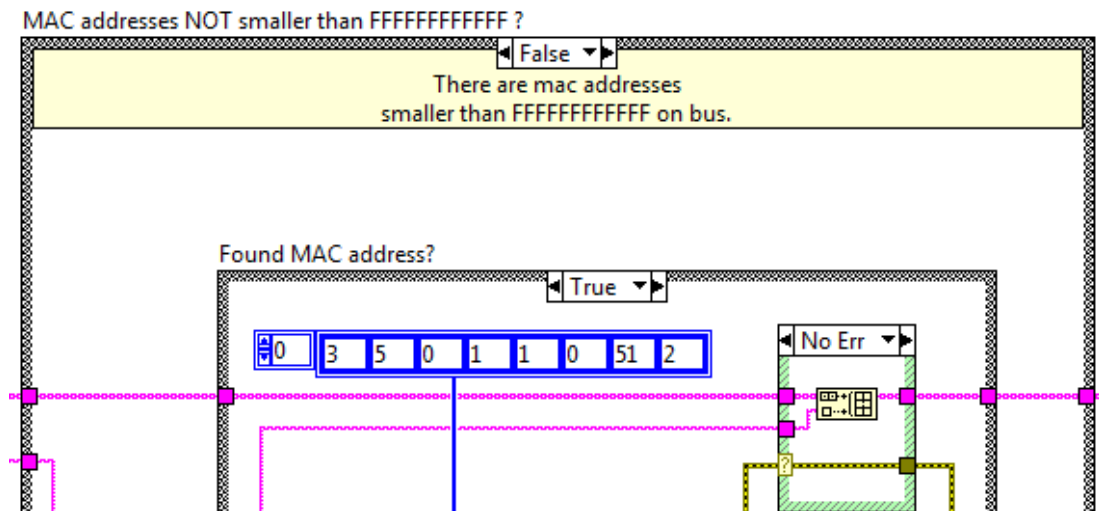
Tagivalikoima haluttiin pitää vähälukuisena ja helposti muistettavana. Yhteisten tagien lukumääräksi arvioitiin 4–5 kappaletta. Näiden lisäksi jokaiselle tehtiin oma henkilökohtainen tagi, jota käytetään muistiinpanomaisesti. Nimitagia käytetään myös allekirjoituksena omille kommenteille.

- `#bug`: tarkoitettu selville bugeille tai muulle rikkinäiselle toiminnollisuudelle, joilla ei ole välitöntä korjaustarvetta.
- `#todo`: ilmaisee puuttuvista ominaisuuksista tai lisätoiminnallisuuden tarpeesta.
- `#needcomment`: epäselvä osio vaatii kommentointia. Tavallisesti tätä tagia ei tule käyttää, sillä koodi olisi kommentoitava itse tällaisen puutteen huomattaessa. Käytettävä vain tuotannollisuuden kannalta perustellussa kiireessä tai oman osaamisalueen loppuessa.
- `#cleanup`: koodin palikkadiagrammin yleisilme vaatii siivoista selkeyttämiseksi.
- `#translate`: kommentointi on tehty, mutta vaatii käännöksen englanniksi. Tagin sisällytyksen syy oli aikaisemmin mainitun kielimuurin kiertäminen.
- nimitagi: jokaisen henkilökohtainen tagi, joka muodostuu sukunimen neljästä ja etunimen kahdesta ensimmäisestä kirjaimesta. Liitetään jokaisen oman kommentin loppuun eräänlaisena allekirjoituksena.

Tagatut kommentit olisivat muotoa `[tag] [kommentti] [nimi]`. Työntekijä Ville Virtasen lisäämä esimerkkikommentti olisi siis muotoa

`#bug #translate Sisääntulo ei hyväksy boolean-arvoja. #virtvi`

Koodin toiminnollisuutta kommentoitaessa ei käytetä prefiksitagia vaan ainoastaan nimitagia. Oleellista ei ole jokaisen pienen osion funktio, vaan kokonaisen VI:n tai monimutkaisemman ali-VI:n tasolla. Myös sekavat tai muuten monimutkaiset struktuurit tulee kommentoida erikseen. Sisällöllisesti kommentista tulee käydä selville sisälle tuleva data, prosessointitapa sekä lähtevä data, mikäli se ei käy selville prosessoinnin kuvauksesta. Aiemmin mainittu kommenttien linkittäminen helpottaa tätä. Aikaisemmissa LabVIEW'n versioissa VI:n lohkodegrammia muuttaessa kommentti saattoi eksyä siihen liittyvästä objektista tai kommenttilaatikko oli vaikeasti asemoitavissa. LabVIEW 2013:n myötä kommenttilaatikon ja objektin välille syntyy viiva jolloin nämä eivät enää eksy toisistaan yhtä herkästi. Valmiille struktuureille voitiin uudessa versiossa asettamaan kuvaus, joka säilyi näkyvillä aina struktuurin yläosassa (kuva 16).



Kuva 16. Struktuurin kuvaus.

Useamman objektin kokonaisuuden kommentoiminen on kuitenkin hankalaa. Mahdollisiksi ratkaisuksi nostettiin joko flat frame sekä flat sequence. Flat frame on käytännössä pelkkä graafinen raami jonka saa piirrettyä mielivaltaisesti lohkiagrammiin, jolloin sen sisältämän kokonaisuuden kommentoinnin voisi linkata raamiin. Flat sequence taas on toiminnallinen strukturi, jossa sen sisältämä koodi ajetaan aina määritellyssä järjestyksessä. Framen ongelma on ettei sitä liikutellessa sen sisältämä koodi liiku mukana. Sequence taas on toiminnallista, eli sitä käyttäessä tässä tarkoituksessa tulee aina lisättyä turhaa koodia. Koodin optimoinnin vuoksi ratkaisuksi valittiin flat frame.

Koska dokumentointi ja standardointi ovat oleellisia osia lean-periaatetta, tuli koodin kommentoinnin olla myös regressiivistä. Tämän vuoksi oli mietittävä miten vanhaa koodia kommentoitaisiin jälkikäteen. Yksittäisiä VI-tiedostoja oli olemassa tuhansia, joten niiden systemaattista kommentointia ei katsottu mielekkääksi. Järkevimpänä ratkaisuna #needcomment-tagia luotiin tätä varten. Pääsääntöisesti kommentit tuli lisätä aina kommentoimatonta koodia löydettyessä.

7 YHTEENVETO

Projektin jatkuvien viivästysten vuoksi arvovirtakartan tuomien ajoitusten arviointi koitui hankalaksi, jonka seurauksesta kartasta tehtiin aiottua yleispätevämpi katsaus nykyiseen NPI-prosessiin. Toisaalta viivästysten aiheuttajat antoivat lisätietoa siitä, miten yrityksessä tavallisesti priorisoidaan keskenään ajasta kilpailevia projekteja. Yrityksen eri elimien välistä yhteistyötä edistettiin tuomalla paremmin selville muiden osastojen rooleja eri vaiheissa projektia.

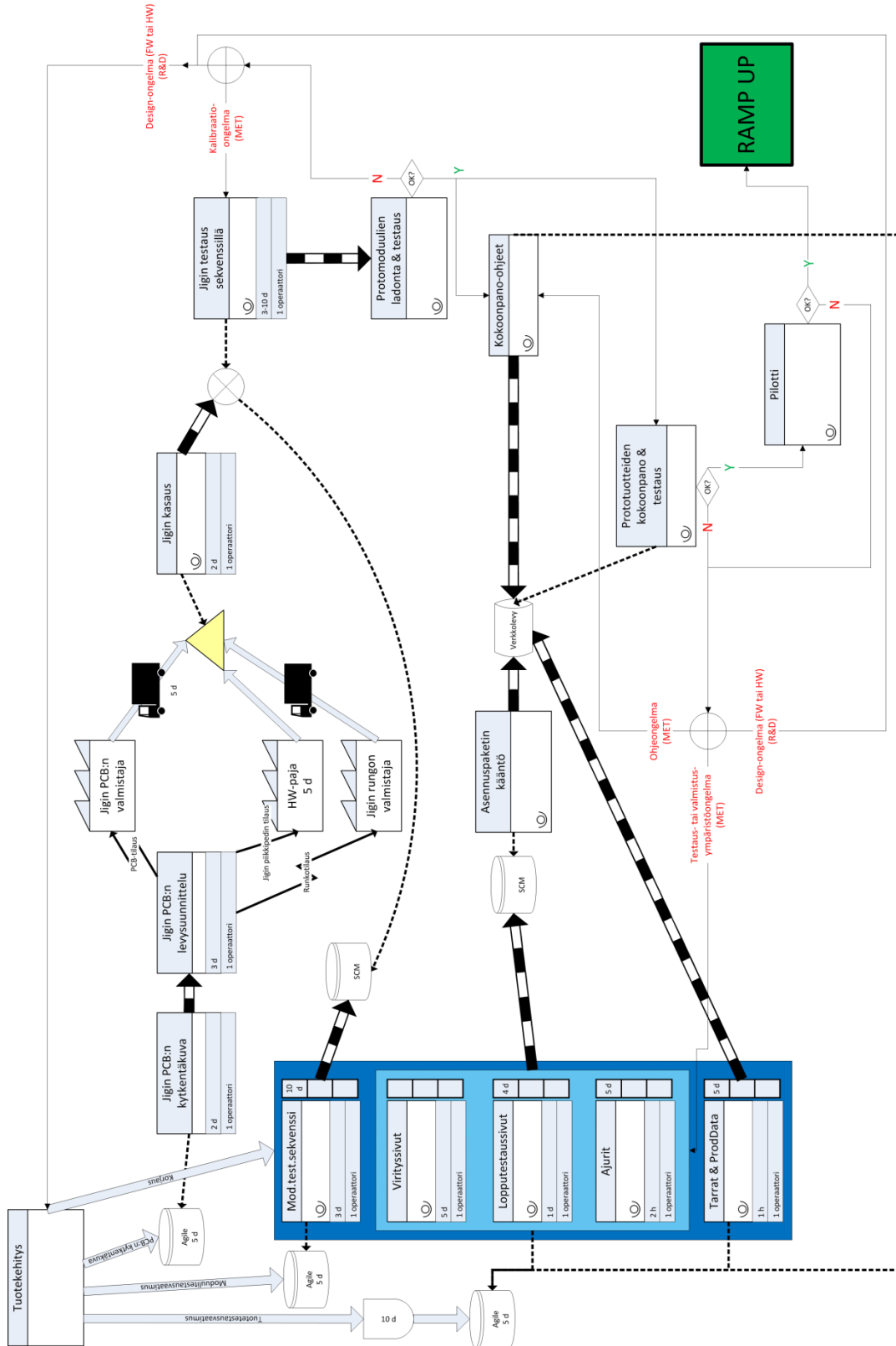
Sekä tuotekehitykseen että testaussuunnitteluun tehtiin lean-periaatteiden mukaisia parannusehdotuksia mudatyypeittäin. Muutosten implementointi jäi esimiesten ja projektipäälliköiden harkittavaksi.

Testausohjelmiston suunnittelun nopeuttaminen kommentoinnin lisäämisellä oli avuksi etenkin uusille työntekijöille. Vähemmän kokeneiden työntekijöiden ongelmat yleensä vaativat aikaa vievää apua vanhemmilta työntekijöiltä, joten ajan säästäminen tästä oli testaussuunnittelun optimoinnin kannalta oleellista. Kommentoinnista tehtiin yhtenäinen ohjeistus jonka tarkoitus on edesauttaa kommentoinnin määrää ja laatua sekä edistää lean-käytäntöjen mukaista dokumentointia.

LÄHTEET

- [1] Knuutila, V., *Jatkuva parantaminen Lean-ajattelun työkaluna : case: Nordea Pankki Suomi Oyj, Osasto X* [www-dokumentti].
Saataavilla: publications.theseus.fi/handle/10024/33955 (luettu 23.5.2014).
- [2] Obe, O., *Value Stream Mapping of a mobile network element software - feature development cycle using Lean development concept* [www-dokumentti].
Saataavilla: publications.theseus.fi/handle/10024/26754 (luettu 23.5.2014).
- [3] Teleste Oyj, *AC3200 User Manual*. 2011.
- [4] Feingold, J., *Lean Roots – A Quick History Lesson* [www-dokumentti].
Saataavilla: www.qualitydigest.com/may08/articles/04_article.shtml (luettu 5.5.2014).
- [5] Rother, M. ja Shook, J., *Learning to See: Value Stream Mapping to Add Value and Eliminate Muda*. Lean Enterprise Institute Inc., 1999, 102 s. 978-096-678-430-5.
- [6] The Lean Enterprise, *Four Steps to Value Stream Mapping* [www-dokumentti].
Saataavilla: www.nwlean.net/toolsCD/VSM/4 steps to VSM.pdf (luettu 14.1.2013).
- [7] Chapman, C.D., *Clean House with Lean 5S* [www-dokumentti].
Saataavilla: www.iquality.com/articles/lean.pdf (luettu 29.4.2014).
- [8] Stage-Gate International, *The Stage-Gate Process* [www-dokumentti].
Saataavilla: www.stage-gate.com/resources_stage-gate_full.php (luettu 21.10.2013).
- [9] Mascitelli, R., *The Lean Product Development Guidebook*. Technology Perspectives, 2006, 310 s. 978-096-626-973-4.
- [10] Mascitelli, R., *The Lean Design Guidebook*. Technology Perspectives, 2004, 320 s. 978-096-626-972-7.
- [11] Gafni, R. ja Geri, N., *Time Management: Procrastination Tendency in Individual and Collaborative Tasks*. Interdisciplinary Journal of Information, Knowledge & Management vol. 5, 2010. 1555-1229.
- [12] Scrum Alliance, *Agile Atlas: Scrum* [www-dokumentti].
Saataavilla: www.agileatlas.org/atlas/scrum (luettu 13.5.2014).
- [13] Agile Alliance, *The Agile Manifesto* [www-dokumentti].
Saataavilla: www.agilealliance.org/the-alliance/the-agile-manifesto/ (luettu 12.5.2014).
- [14] National Instruments, *LabVIEW 2013 Features and Changes* [www-dokumentti].
Saataavilla: zone.ni.com/reference/en-XX/help/371361K-01/lvupgrade/labview_features/ (luettu 6.3.2014).

Valmistusympäristön suunnittelun VSM-kuvaus



LabVIEW-kommentointikyselylomake



LabVIEW-
kommentointikysely

1 (1)

March 7, 2014

1 Kommentoitko koodiasi?

- Kyllä Joskus En

Jos et, miksi?

2 Onko kommentointi mielestäsi tarpeellista?

- Kyllä Ei

3 Kommentoidaanko koodia Telestellä nykyään mielestäsi tarpeeksi?

- Kyllä Ei

4 Oletko tutustunut LabVIEW 2013:n uusiin kommentointityökaluihin, kuten tagit ja kommenttien linkkaus blokkeihin?

- Kyllä En

5 Vaikuttaako LabVIEW:n kommentointityökalujen kehittyminen omaan suhtautumiseesi kommentointia kohtaan?

- Kyllä Ei Ei tarpeen

6 Millä tavoin voisit parantaa omia tai yhteisiä kommentointitapoja?
