



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Henri Hämäläinen

# OEE-järjestelmän sulauttaminen tuotantoprosesseihin

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö ja automaatiotekniikka

Insinöörityö

28.1.2019

Tekijä Otsikko	Henri Hämäläinen OEE-järjestelmän sulauttaminen tuotantoprosesseihin
Sivumäärä Aika	35 sivua + 2 liitettä 28.1.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Ammatillinen pääaine	Automaatiotekniikka
Ohjaajat	Testausylläpidon tiiminjohtaja Teemu Loman Osaamisaluepäällikkö Jari Olli
<p>Insinööriyössä tutkittiin ja suunniteltiin tuotannon tehokkuutta mittaavan järjestelmän prototyyppi, joka OEE (Overall Equipment Effectiveness) -menetelmän mukaisesti muutti tuotantoasemilta saatavan datan tehokkuutta kuvaavaksi prosenttiarvoksi. Työ tehtiin Vaisalalle, jolle projektin tavoitteena oli saada tarkempaa tilastotietoa tuotantoasemien tehokkuudesta ja automaattisesti jalostaa tieto helposti visualisoitavaan muotoon.</p> <p>Projektissa selvitettiin tarvittavia tiedonlähteitä tehokkuuden mittaukseen sekä tapaa integroida suunniteltava järjestelmä osaksi Vaisalalan laajempaa tietoverkkoa. Suunnittelussa päädyttiin kolmiosaiseen ohjelmakokonaisuuteen. C#-kielellä kirjoitettu ohjelma keräsi tuotantoasemien luomaa dataa Vaisalalan tietokannasta ja tallensi sen jalostetussa muodossa OEE-tietoja varten rakennettuun SQL-tietokantaan. Tietokanta yhdistettiin Excel-ohjelmaan, joka toimi vaihtoehtoisena käyttöjärjestelmänä datan visualisointiin.</p> <p>Prototyyppijärjestelmässä ei projektin aikana päästy automaattisesti toimivaan malliin. Haasteet C#-ohjelman asemakohtaisten metodien suunnittelussa ja lähdetietokantojen yhdistämisessä hidastivat valmistumista. Insinööriyön lopputuloksena saatiin Vaisalalle soveltuvan ohjelman rakennemalli sekä selvitys järjestelmään liittyvistä haasteista.</p> <p>Ohjelmointiin perustuva automaattinen järjestelmä osoittautui parhaaksi menetelmäksi OEE-tietojen keräämiseen, sillä näin minimoitiin manuaaliprosessin riskit. OEE-arvot perustuivat tuotantoasemien ohjelmien automaattisesti lähettämään dataan, mikä paransi tiedon laatua ja käsittelyn nopeutta. Ohjelman rakenne sovitettiin mitattavaan tuotantoasemaan ja vaihteli huomattavasti asemien välillä. Insinööriyö toimii pohjarakenteena ja ohjeistuksena jatkotutkimuksissa sekä OEE järjestelmän rakentamisessa, koska se määrittelee perusteellisesti projektin vaatimukset ja haasteet.</p>	
Avainsanat	OEE, prototyyppi, käytettävyys, tuotantoprosessi

Author Title	Henri Hämäläinen Integration of OEE-system into production processes
Number of Pages Date	35 pages + 2 appendices 28 January 2019
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Automation Engineering
Instructors	Teemu Loman, Manager of Testing and Calibration Team, Vaisala Jari Olli, Head of Department, Clean Technology, Metropolia
<p>The thesis work consisted of the evaluation and development of a prototype that measures the effectiveness of manufacturing operations, employing the OEE (Overall Equipment Effectiveness) principle of utilizing the data of the manufacturing unit and converting it into a percentage that represents productivity. The work was conducted to benefit Vaisala with the aim of obtaining accurate statistics of the manufacturing units' effectiveness and grasp the data into easily viewable format.</p> <p>The project evaluated the required data needed to measure the manufacturing effectiveness and aimed to integrate the new system becoming part of Vaisala's broader information network. It was decided to divide the system into three parts. The system utilizing C# language collected data from Vaisala's information network and refined it into the SQL- database that was created to store the OEE information. The database was linked with Excel, which was used as an optional data visualization system.</p> <p>The prototype of the system failed to function as automatically working model. The challenges related to the planning of the C# methods and connectivity of the different databases were slowing down the project. The outcome of the engineering work was to create a structural model of the system and provide further details of the project's challenges.</p> <p>Programming-based automated system was proven successful in terms of gathering OEE-information due to its capability to limit the risks of manual processing. OEE value was based on the data sent automatically from the manufacturing unit which enhanced the quality and the speed of information. The structure of the program was adjusted to manufacturing units and greatly varied between the stations. The engineering work may act as a reference and instruction in the future research and product development process, as it defines profoundly the requirements and challenges of the project.</p>	
Keywords	OEE, prototype, availability, production process

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Tutkimusteoria	3
2.1	Automaation hyöty teollisuudessa	3
2.2	Lean-periaate	4
2.3	Tuottava kunnossapito ja kokonaistehokkuus	4
2.4	Mitä on OEE?	5
2.5	Tuotannon häviöihin vaikuttavat syyt	7
2.6	OEE:n osatekijät ja laskeminen	10
2.7	OEE-suunnitteluun vaikuttavia tekijöitä	13
2.8	Ohjelmasuunnittelu	14
2.9	Prototyypinohjelmointi	15
3	OEE-järjestelmän muodostaminen Vaisalalla	18
3.1	Tutkimustyö ja projektisuunnittelu	18
3.1.1	Haastattelut	18
3.1.2	Käyttäjät	20
3.2	OEE:n muuttujien määrittely ja projektin rajaus	20
3.3	Ohjelmarakenne	22
3.4	OEE-ohjelma	24
3.5	Tietokantamallin muodostaminen	25
3.5.1	Tietokannan vaatimukset	26
3.5.2	Tietokantarakenne	27
3.5.3	Tietokantaliitännät	28
3.6	Visualisointi	29
4	Yhteenveto	32
	Lähteet	34

## Liitteet

Liite 1. OEE-tietokannan SQL-koodi

Liite 2. Haastattelurakenne OEE:n hyödyllisyyden määrittämiseksi

## Lyhenteet

ERP	Enterprise Resource Planning. Tietojärjestelmä, joka yhdistää ja visualisoi yrityksen eri toimintoja, kuten varastonhallintaa, kirjanpitoa, laskutusta ja tuotantoa.
FPY	First Pass Yield. Ensimmäisen läpimenon periaate, jossa hyväksi lasketaan vain ne tuotteet, jotka pääsevät tuotannosta läpi ensimmäisellä yrittämällä.
KNL	Käytettävyys, Nopeus ja Laatu, englanniksi OEE, on kokonaistuotannon prosenttiluku, joka koostuu kolmesta kertoimesta. Kertoimet kuvaavat tuotannon tehokkuutta eri näkökulmista.
OEE	Overall Equipment Effectiveness. Suomeksi kokonaistuotanto.
Six Sigma	Tilastotieteessä käytettävä laadunvalvonnan työkalu, jossa tavoitteena on mitata prosessissa tapahtuvien virheiden tyyppiä ja määrää, jotta ne voidaan priorisoida ja eliminoida.
SQL	Structured Query Language on standardoitu kyselykieli, jolla relaatiotietokantaan voi tehdä hakuja.
SSMS	SQL Server Management Studio on Microsoftin julkaisema relaatiotietokantojen suunnittelu- ja muokkausohjelma.
TPM	Total Production Management, eli suomeksi tuottava kunnossapito, on tuotannon tehokkuuden parantamiseen keskittyvä ajattelutapa. Pää tarkoituksena vikojen ennakoiva poisto.
XML	Extensible Markup Language on merkintäkielistandardi, jota käytetään tiedon tallentamiseen.

## 1 Johdanto

Tehokkuuden parantaminen ja häviöiden karsiminen kaikille toimialoille: tällä periaatteella Toyota nousi 80-luvulla autonvalmistajien kärkilistoille. Lean-ajattelutapaa on siitä lähtien jalostettu ympäri maailmaa, ja se on laajasti todettu yhdeksi parhaimmista toimintatavoista. Insinööriyössä perehdytään lean-periaatteen mukaisesti tuotannon tehokkuutta mittaavan OEE-järjestelmän toteuttamiseen. Tuotannon kokonaistehokkuuden tarkan mittaamisen avulla näytetään ja eritellään tehokkuushäviöitä aiheuttavat syyt. Syiden tunnistaminen mahdollistaa niiden poistamisen ja ongelmien ennakoivan välttämisen.

### Tutkimuksen tavoite

Tavoitteena on suunnitella ja toteuttaa OEE-tietoa keräävä järjestelmä, joka sulautetaan osaksi Vaisalan tietokantoja ja prosesseja. Tietojärjestelmä kerää tarvittavaa dataa tuotantoasemilta ja tietokannoista, prosessoi datan jalostettuun muotoon, siirtää sen omaan tietokantaan muiden ohjelmien käyttöä varten sekä visualisoi datan selkeinä kaavioina ja taulukoina. Palvelimella toimivan ohjelman tulisi olla helposti laajennettavissa kymmenille erilaisille tuotteille ja prosesseille. Työn alussa tutkitaan, kuinka hyvin Vaisalan nykyiset tietokannat ja ohjelmat kattavat OEE-järjestelmän tarpeet. Tutkimuksen pohjalta raportoidaan nykytilanne ja ehdotetaan mahdollisista jatkotoimenpiteistä. Työn lopputuloksena on kahden tuotantoaseman kattava prototyypiohjelma, jolla asemien tuotannon tehokkuutta seurataan seuraavien kuukausien aikana.

### Vaisala

Insinööriyö on Vaisalan toimesta aloitettu projekti. Vaisala on maailman johtava ympäristön ja teollisuuden mittausratkaisuihin erikoistunut yritys, joka tarjoaa satoja erilaisia tuotteita sää- ja teollisuusolosuhteiden mittaamiseen pienistä sensoreista kokonaiisiin tutkajärjestelmiin. Työntekijöitä Vaisalalla on 1600 henkeä, ja asiakaskuntaa löytyy 150 maasta. Vaisala investoi miltei 12 % liikevaihdostaan tuotekehitykseen. Korkean laadun ansiosta Vaisalan teknologia on valittu myös Mars 2020 -mönkijään. Vaisala panostaa

jatkuvan kehityksen ilmapiiriin ja pyrkii parantamaan tuotteiden ja prosessien tehokkuutta koko henkilökunnan voimin. Insinööriyön OEE-projekti on osa suurempaa tehokkuutta mittaavaa kokonaisuutta, joka käsittää useita eri ohjelmia ja projekteja. [10.]

### Tutkimuksen metodologia

Tämän kappaleen tarkoitus on perehdyttää lukija työn tutkimusmenetelmiin sekä aineistonkeruuseen. Opinnäytetyön tarkoitus on luoda Vaisalalle OEE-järjestelmä, joka edistää eri tuotantolinjojen kokonaistehokkuutta. Tavoitteen saavuttamiseksi on tärkeä määritellä työn tutkimusmenetelmät, eli metodit, jotka definoivat opinnäytetyön systemaattisen suunnittelumallin esitellen sen erilaiset tutkimusvälineet sekä aineiston hankintamenetelmät, joiden perusteella tutkija pohdiskelee tutkimusongelmaa ja rakentaa suosituksia.

Opinnäytetyö hyödyntää olemassa olevaa teoreettista tietämystä, joten yhtenä tutkimusmenetelmänä käytetään teoreettista taustatutkimusta, jolla kartoitetaan jo olemassa olevaa kirjallisuutta tutkimusilmiöistä (luku 2.1). Teoreettinen tutkimustieto yhdistetään käytännössä kerättävään, empiiriseen, tietoon. Empiirisellä tutkimuksella tarkoitetaan kokemusperäistä tutkimusta, jossa tutkimusilmiötä havainnollistetaan käytännötilanteiden kautta. Tämä työ käyttää kvalitatiivista tietoa hyödyntäen asiantuntijoiden näkemyksiä. Tiedonkeruumetodina käytetään haastatteluja, jossa asiaa tutkitaan osallistujien kokemuksien ja tietotaidon avustuksella (luku 3.2). [15; 16.]

### Teoreettinen taustatutkimus

Ensimmäinen askel opinnäytetyön tiedonkeruussa on laaja perehtyminen jo olemassa olevaan tutkimuskirjallisuuteen. Kirjallisuuskatsaus sisältää pääosin tutustumisen OEE-menetelmän teoreettiseen taustatietoon (luku 2.1). Teoreettinen taustatutkimus käsittelee myös tarkemmin lean-ajattelun periaatteita sekä perehtyy tuotannon laatujohtamisen työkaluun, Six Sigmaan. Luku 2 tarkastelee laajemmin myös ohjelmointisuunnittelua koskevaa teoriaa. Tutkimusilmiön huomioon ottaen tietokantoihin ja ohjelmointikieliin liittyvän tiedon kartuttaminen on erittäin tärkeää. Erityisesti informaatio koskien SQL-tietokantaa ja C#-ohjelmointikieltä sekä perehtyminen Excelin VBA-makroiin on hyödyllistä



tutkimustyön tavoitteen saavuttamiseksi. Kyseisen tiedon käytännön soveltaminen demonstroidaan luvussa 3. Tiedonkeruumetodi teoreettiselle taustatutkimukselle on pääosin informaation keräämistä sähköisen tiedonhakuivujen sekä akateemisten oppikirjojen avustuksella.

Toinen tiedonhankintametodi on kvalitatiivinen tieto eli laadullinen menetelmä, jossa hyödynnetään strukturoimattomia haastatteluja. Vaisalan työntekijöiden asiantuntijatietoa hyödynnetään aineiston hankinnassa. Koska työntekijöillä on erityisosaamista sekä tietotaitoa tutkimusilmiöön liittyen, avoimet haastattelut todetaan parhaaksi tiedonkeruumenetelmäksi opinnäytetyön tavoitteen saavuttamista ajatellen.

## 2 Tutkimusteoria

### 2.1 Automaation hyöty teollisuudessa

Viime vuosikymmenien nopea teknologinen kehittyminen on tuonut suuria muutoksia teollisuuden toimintatapoihin. Teollisen ympäristön kasvava kansainvälinen kilpailu on motivoinut valmistajia etsimään ratkaisuja paremman laadun, alhaisempien kustannusten ja nopeamman tuotannon saavuttamiseen. On selvää, että laitteilla ja koneilla on suuri merkitys tuotannossa. Kuitenkin laitteiden koko tuotantokapasiteetin hyödyntäminen paremman tuotannon hyötysuhteen saavuttamiseksi on jatkuva haaste. Tämän vuoksi hyvät tuotannon hallintajärjestelmät ovat kovassa kysynnässä. [1, s. 8.]

#### Teollisuuden digitalisoituminen

Digitalisaatio on tullut kaikille teollisuuden tuotannon alueille. Kommunikaation paraneminen, datan keräämis- ja analysointimenetelmät, pilviprosessointi ja henkilökohtaiset älylaitteet avaavat mahdollisuuksia yritystoimintaan. Useat yritykset Vaisala mukaan lukien integroivat parhaillaan kaikkea tietoa henkilökunnasta tuotantolaitteisiin yhdeksi suureksi informaatioverkoksi, joka kattaa koko yrityksen. Tästä verkosta käytetään yleisesti termiä teollinen internet.

Teollisesta internetistä puhuttaessa keskitytään yleensä pääasiassa laitteisiin ja dataan, mutta työtä tekevät ihmiset ovat yhtä keskeinen osa tätä mullistusta. Itse asiassa juuri ihmisten työskentelytavan muuttuminen on yksi teollisen internetin suurimmista hyödyistä.

## 2.2 Lean-periaate

Lean-ajattelussa keskitytään resurssitehokkuuden sijasta virtaustehokkuuteen. Virtaustehokkuus tarkoittaa tuotettavan asian läpimenoajan minimoimista, eli yksittäiset tuotteet pyritään saamaan koko valmistusprosessin läpi mahdollisimman nopeasti ilman välivaiheita tai turhia vaiheita. Virtaustehokkuus varmistetaan poistamalla prosessista tarpeettomia vaiheita (hukkaa) ja tuotantoa tehdään vain asiakkaan tarvitsema määrä. Virtaustehokkuus tarkoittaa arvon siirron tiheyttä [7, s. 33], mikä ei viittaa pelkästään tuotevalmistuksen työrutiineihin, vaan myös abstrakteihin prosesseihin kuten ajatustyöhön. Tietojärjestelmissä virtaustehokkuus lisääntyy, kun oikea tieto kulkee nopeasti ilman tarpeettomia välikäsiä henkilöille, jotka sitä tarvitsevat. Tieto ei liiku pelkästään sitä kysytessä, vaan tieto hakeutuu automaattisesti oikeille henkilöille. Tästä hyvänä esimerkkinä toimii insinööriyön OEE-järjestelmä, joka tuotantoaseman tehokkuuden laskiessa normaalia alemmaksi tekee reaaliajassa ilmoituksen tehtaan ylläpidolle ja osoittaa hidastumisen todennäköisimmän syyn. Prosessi-insinööri saa tiedon suoraan tuotantoasemalta ennen kuin hän edes tiesi ongelmasta. Samoin ohjelmoitaessa metodien selkeä jakaminen, yhteisten kirjoitustapojen käyttäminen ja ohjeiden kirjoittaminen kasvattaa usean ohjelmoijan välistä virtaustehokkuutta.

## 2.3 Tuottava kunnossapito ja kokonaistehokkuus

Osana leanin kaltaisia kunnossapitostrategioita on tuottavan kunnossapidon ideologia, englanniksi Total Productive Maintenance (TPM), jonka tavoitteena on koko organisaation voimin parantaa tuotantolaitteiden tehokkuutta. Tuottavan kunnossapidon konseptin kehitti 1950-luvulla Seiichi Nakajima, joka toimi japanilaissa Nippon Denso -yrityksessä [4.]. Ideologiassa kunnossapitoon osallistuu jokainen henkilöstön osa ylimmästä johdosta operaattoreihin; näin kunnossapito huomioidaan prosessin jokaisessa vaiheessa jo tuotteiden suunnitteluvaiheesta lähtien.

The word 'Total' in TPM has these meanings: total effectiveness—pursuit of economic efficiency or profitability; total PM—maintenance prevention and activity to improve maintainability as well as preventive maintenance; and total participation—autonomous maintenance by operators and small group activities in every department at every level. —Nakajima [4.]

Nakajiman tavoite ei ollut kouluttaa henkilöstöä tekemään rutiininomaisia prosesseja, vaan ohjeistaa heitä ajattelemaan tuotantovaiheita osana kokonaisuutta ja tunnistamaan tuotannon tehokkuuteen vaikuttavat tekijät. Tuottavassa kunnossapidon ydinkohtana ei olekaan itse korjaustöiden tekeminen, vaan korjausta vaativien vikojen taustalla olevien syiden selvittäminen ja parannusten kautta vian ennaltaehkäisevä poistaminen. Ennakoiva ongelmien poistaminen vähentää tuotannon seisahduksia ja viallisia tuotteita, mikä huomattavasti nostaa tuotannon tehokkuutta ja laskee turhia kuluja. [4.]

Tuottavan kunnossapidon ytimessä on kokonaistehokkuuden mittari, englanniksi Overall Equipment Effectiveness (OEE). Se on laajasti ympäri maailmaa levinnyt tehtaissa käytettävä mittaustapa, joka otettiin käyttöön TPM-filosofian alkuaikoina. Kokonaistehokkuuden seurantaan tarvittiin selkeä mittari, jonka avulla tuotannon toimivuudesta saadaan tilastotieteellistä tietoa ja tuotannon muutoksia voidaan seurata pitkällä aikavälillä. [1, s. 8; 5.]

## 2.4 Mitä on OEE?

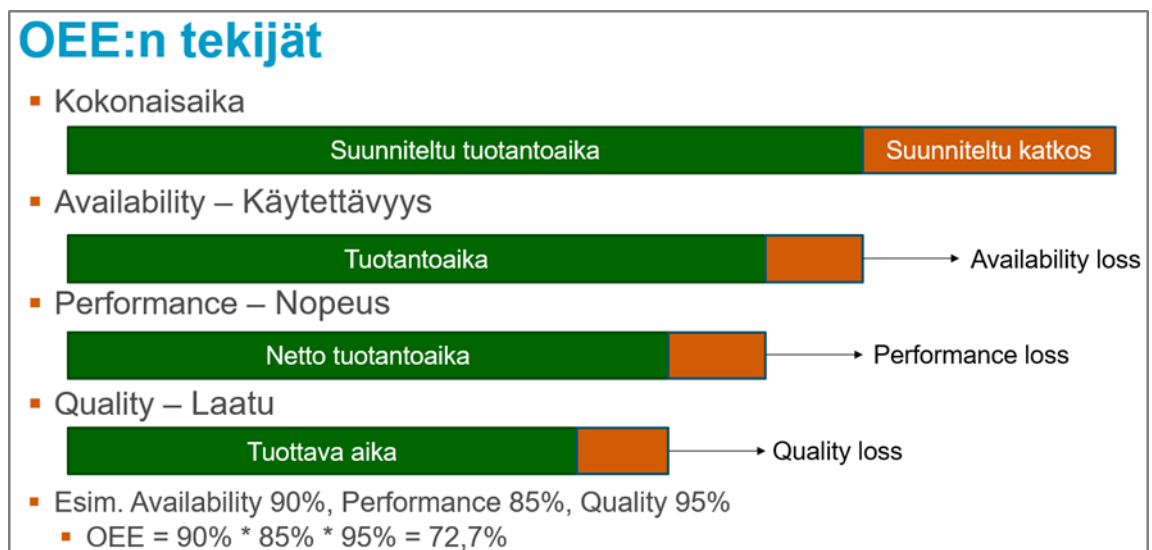
Overall Equipment Effectiveness (OEE) eli kokonaistehokkuus on kultaisen standardin saanut tuotannon tehokkuuden mittaustapa ja tunnusluku. Suomessa OEE:sta käytetään myös lyhennettä KNL (Käytettävyys, Nopeus, Laatu), mutta tässä työssä käytetään kansainvälistä termiä OEE. Kokonaistehokkuus on yleisesti arvioitu onnistuvan yritystoiminnan parhaaksi menettelytavaksi tehokkuuden mittaamisessa [6.]. Yksinkertaistettuna OEE tunnistaa prosentuaalisen osan tuotantoajasta, joka on oikeasti tuottavaa. 100 %:n OEE-pisteytys tarkoittaa, että tuotanto tekee vain hyviä tuotteita niin nopeasti kuin mahdollista ilman mitään pysähdysaikoja. Tällöin tuotanto toimisi siis teoreettisella maksiminopeudella.

OEE-luku muodostuu kolmen kertoimen yhteysvaikutuksesta; käytettävyys (Availability), nopeus (Performance) ja laatu (Quality). Kertoimien muodostumiseen palataan tarkemmin alaluvussa 3.3. Nakajiman mukaan (1982) OEE:n laskukaava on

$$OEE = A (Availability) * P (Performance) * Q (Quality), \quad (1)$$

jonka osatekijät A, P ja Q ilmoitetaan prosenttiyksikköinä, jotka yhteen kerrottaessa muodostavat OEE-arvon.

Kuitenkin OEE-laskenta lopulta pohjautuu aikamäärien vertaamiseen täysin tuottavan ajan ja suunnitellun tuotantoajan välillä. OEE-kaavan kertoimet leikkaavat häviöiden kautta aikaa pois täysin tuottavasta ajasta. Kuva 1 esittää, kuinka kolme osatekijää vaikuttavat tuotantoaikaan niihin liittyvien häviöiden kautta:



Kuva 1. Tuottavaan aikaan vaikuttavat tekijät. Lopullinen tuottava aika on osatekijöiden vaikutuksen yhteistulos.

Kuvan 1 ensimmäinen osa, suunniteltu tuotantoaika, syntyy kalenteriajasta, josta on poistettu aikavälit, jolloin tehdas on kiinni eivätkä tuotantolaitteet valmista mitään. Esimerkiksi viikonloput ja tehtaen sulkeminen yöksi eivät kuulu suunniteltuun tuotantoaikaan. OEE-arvo on tuottavan ajan ja suunnitellun tuotantoajan välinen erotus, joten suunnitellun tuotantoajan mahdollisimman tarkka tietäminen on tärkeää tehokkuutta las-

kettaessa. Kuvan toisessa vaiheessa oleva tuotantoaika koostuu suunnitellusta tuotantoajasta, josta on poistettu laitepysähdysten aiheuttama aikahäviö (Availability loss). Kaikki tuotantoajan aikana tapahtuvat pysähdykset kuuluvat häviöihin, myös suunnitellut ennakkohuollot. Jotkin yritykset saattavat laskea ennakkohuollot suunnitellun tuotantoajan ulkopuolelle, mutta tämä vääristää OEE-lukemaa, sillä ennakkohuollot tehdään usein työtuntien aikana samalla aikavälillä, jolloin tuotannon pitäisi olla päällä.

Nettotuotantoaika koostuu tuotantoajasta, josta on vähennetty hidastuneen tuotannon aiheuttamat aikahäviöt (Performance loss). Vaikka tuotantolaite olisi jatkuvasti käynnissä, sen nopeus saattaa olla huomattavasti alhaisempi teoreettisesta maksiminopeudesta; tästä käytetään termiä ideaalisykli aika (ideal cycle time). Hidastumisen seurauksena laite valmistaa pienemmän määrän tuotteita, mitä siinä ajassa on suunniteltu valmistettavaksi, ja tämä näkyy tuotantotavoitteeseen pääsyn viivästymisenä. Lopullinen tuottava aika koostuu netto tuotantoajasta, josta on vähennetty viallisten tuotteiden aiheuttamat aikahäviöt (Quality loss). Vaikka tuotantolaite toimisi ilman pysähdyksiä täydellä nopeudella, osa tuotteista voi silti olla viallisia tai vaatia uudelleenvalmistamista epäonnistuneen ajon vuoksi. Jokaisen viallisen tuotteen valmistukseen kuluva tuotantoaika on menetettyä aikaa. Lisäksi uudelleen ajettavat tuotteet saattavat hidastaa muuta tuotantoa ruuhkauttaen tiettyjä tuotantoprosessin vaiheita.

Merkittävä tekijä OEE-mittauksen suosion taustalla on sen yksinkertainen laskutapa ja laaja vaikutus tuotannon tehokkuuteen [1, s. 16]. Kuitenkin laskukaavan ottaminen käyttöön voi olla todellisuudessa melko monimutkaista. Erilaiset prosessit vaativat usein juuri niille tehtyjä ohjelmia ja kaavoja OEE-kertoimien tarvitsemien arvojen kasaamiseksi, ja tehtaas saattavat keskittyä enemmän tiettyyn tehokkuuden osatekijään, mutta pohjimmiltaan kaikki OEE-ohjelmat perustuvat samaan konseptiin. Tämän työn luvussa 4 keskitytään OEE:n soveltamiseen Vaisalan tarpeisiin ja siihen, kuinka laskentaohjelma saadaan mukautettua kymmeniin erilaisiin tuotantoprosesseihin.

## 2.5 Tuotannon häviöihin vaikuttavat syyt

Valmistusprosessia kuvitellessa usein ensimmäisenä mieleen tulee toimilaite, jota ohjaa operaattori. Prosessissa vastaan tulee usein ongelmia, kuten koneen yllättäviä pysäh-

dyksiä. Nämä ongelmat aiheuttavat eri asteisia häviöitä, jotka yhteen kasautuessaan alkavat vahvasti vaikuttamaan tuotantoon. Kuten Lean-periaatteessa huomattiin, häviöitä aiheuttavien ongelmien kasvaessa lineaarisesti tuotteiden läpimenoaika kasvaa eksponentiaalisesti [7, s. 42]. Tämän vuoksi TPM keskittyy häviöiden syiden etsimiseen ja poistamiseen. Nakajiman sanoja lainaten; laitteiston tehokkuuden maksimoiminen vaatii virheellisten tuotteiden, häiriöiden ja muiden negatiivisten tilanteiden täydellistä eliminointia [6.]. Kuva 2 esittää kuusi häviöiden pääkategoriaa ja sen, kuinka ne liittyvät OEE-mittarin kolmeen pääkertoimeen.

Overall Equipment Effectiveness	Recommended Six Big Losses	Traditional Six Big Losses
Availability Loss	Unplanned Stops	Equipment Failure
	Planned Stops	Setup and Adjustments
Performance Loss	Small Stops	Idling and Minor Stops
	Slow Cycles	Reduced Speed
Quality Loss	Production Rejects	Process Defects
	Startup Rejects	Reduced Yield
OEE	Fully Productive Time	Valuable Operating Time

Kuva 2. Kuusi yleisintä syytä tuotannon häviöihin. Syiden tunnistaminen ja eliminointi on tehokkaimpia tapoja nostaa OEE-lukua. [6.]

Paremmän ymmärryksen saavuttamiseksi kuusi yleisintä syytä käydään läpi yksitellen seuraavassa:

Käytettävyyteen vaikuttavat suunnitellut ja suunnittelemattomat pysähdykset:

- Laitevikaan/rikkoutumiseen (Equipment Failure) lasketaan kaikki merkittävät aikavälit, jolloin tuotantolaitteen olisi tarkoitus olla toiminnassa, mutta on häiriön seurauksena pysähdyksissä. Häiriö voi olla esimerkiksi laitteen

mekaanisen osan jumittuminen, ohjelman kaatuminen tai operaattorin poistuminen työpisteeltä. Myös tuotteen materiaalipula ylävirrasta, tai alavirran laitteen toimiminen pullonkaulana. Jos laitevika kestää alle viisi minuuttia, se määritellään usein pieneksi pysähdykseksi. [6.]

- Huoltoon ja laitesäätöihin (Setup and Adjustments) luokitellaan kaikki aikavälit, jolloin laite on aikataulutettu tuotantoon, mutta laite ei ole toiminnassa muutosten tai huoltotöiden vuoksi. Yleisesti ajateltuna tähän kuuluvat kaikki suunnitellut pysähdykset, joita voivat olla esimerkiksi ennakkohuolto, laatutarkastus, puhdistus sekä alkukäynnistys. Yleisin laitesäätöihin liittyvä häiriö on operaattoreiden vuoronvaihto. [6.]

Nopeuteen vaikuttavat häiriöt:

- Tyhjäkäynti ja pienet pysähdykset (Idling and minor stops) luokittelee hetket, jolloin laite pysähtyy muutamaksi minuutiksi ja palautuu operaattorin toimesta. Yleisiä häiriösyitä voi olla väärät ohjelma-asetukset, antureiden virheellinen asento tai materiaalitukos. Tämän kategorian häiriöt kestävät usein alle viisi minuuttia, mutta ovat kroonisesti toistuvia, minkä seurauksena operaattorit eivät saata huomata niiden aiheuttamaa vaikutusta pitkällä aikavälillä. Yritykset eivät usein huomaa laskea pieniä pysähdyksiä. [6.]
- Hidastunut nopeus (Reduced speed) luokittelee aikavälin, jolloin laitteen toiminta on hitaampaa mitä sen ideaalinen sykli aika (teoreettinen nopein aika mikä kuluu yhden tuotteen valmistamisessa). Yleisiä syitä hidastumiselle ovat likaiset tai kuluneet laitteet, huono öljyäminen, operaattorin kokemattomuus ja materiaalin huonolaatuisuus. [6.]

Laatuun vaikuttavat häiriöt:

- Prosessihyltyt (Process defects) luokittelee viallisten tuotteiden vaikutuksen tuotantoon tasaisen tuotannon aikana. Jokaisen viallisen tuotteen valmistukseen kulunut aika on menetettyä tuotantoaikaa. Tähän kuuluu sekä hylätyt tuotteet että uudelleentyöstöön vietävät tuotteet, sillä OEE toimii FPY (First Pass Yield) -periaatteen mukaisesti. [6.]
- Vähentynyt saanto (Reduced Yield) sisältää kaikki vialliset tuotteet, jotka syntyvät laitteen käynnistyksen ja tasaisen ajon (steady-state) välillä. Vähentynyttä saantoa seurataan erityisesti laitteen muutostöiden jälkeen. Esimerkkejä syistä pienemmälle saannolle ovat huonosti toteutetut vuoronvaihdot, väärät asetukset uutta osaa tuotettaessa ja käynnistysaikaa vaativat laitteet. [6.]

Yrityksen OEE-järjestelmää suunnitellessa syyt kannattaa erotella mahdollisimman tarkasti, jolloin niistä voidaan luoda tilastotieteellistä dataa ja priorisoida eniten haittaa aiheuttavien häiriöiden korjaus ensimmäiseksi.

## 2.6 OEE:n osatekijät ja laskeminen

Organisaatiot ovat kehittäneet OEE-arvon laskemiseen useita erilaisia laskukaavoja, jotka parhaiten soveltuvat heidän tarpeisiin. Alkuperäisen laskentamallin kehitti Seiichi Nakajima, jonka kaava (1) esitelläänkin luvussa 3.1. Yksinkertaisin tapa laskea OEE on verrata täysin tuottavaa aikaa suunniteltuun tuotantoaikaan. Täysin tuottava aika on vain toinen tapa ilmaista pelkästään hyvien tuotteiden valmistamisen niin nopeasti kuin mahdollista ilman pysähdyksiä. Täten laskukaava voidaan esittää muodossa:

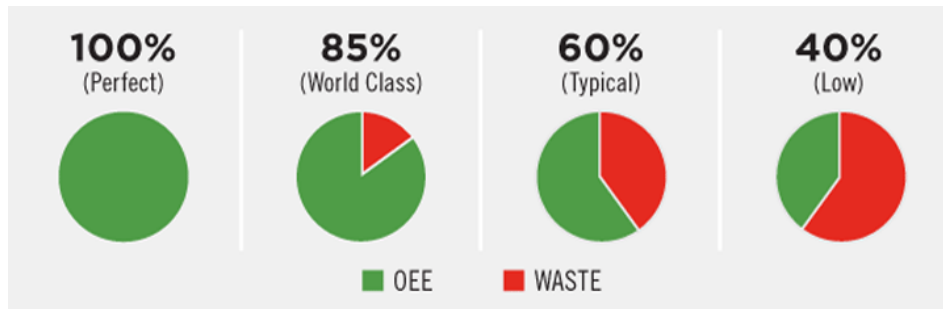
$$OEE = \frac{\text{Hyvät tuotteet} * \text{Ideaalisykli aika}}{\text{Suunniteltu tuotantoaika}} \quad (2)$$

- Hyvät tuotteet tarkoittavat valmistettuja tuotteita, joissa ei ole vikoja.
- Ideaalisykli aika viittaa teoreettisesti nopeimpaan aikaan valmistaa yksi tuote.
- Suunniteltu tuotantoaika on aikaväli, jolloin tuotantolaite on kalenterissa merkitty olevan toiminnassa.

Vaikka tämä kaava on täysin toimiva tapa laskea OEE, se ei anna tarkkaa kuvaa OEE-arvoon vaikuttavista taustatekijöistä. Nakajiman kaavan 1 kolmen osatekijän (käytettävyys, nopeus ja laatu) avulla tuotannon hidastavat syyt voidaan tunnistaa ja erotella toisistaan. Pelkkä OEE-arvon tietäminen ei ole tuotannon tehokkuuden suurin hyöty, vaan hyöty tulee häiriösyiden tuntemisesta ja eliminoimisesta. Tämä on ainut tapa nostaa tuotannon OEE-arvoa. Kaavan 2 yksinkertainen OEE kaava soveltuu kuitenkin hyvin kaavan 1 tuloksen tarkistamiseen, sillä jos kaavojen arvot ovat täysin paikkansapitävää, niiden lopputulos on sama.

OEE-arvo toimii samanlaisten tuotantoprosessien vertailuarvona. Tutkijat ja yritysten johto käyttävät OEE-lukuja hyväksi halutessaan antaa selkeitä numeroita teollisuuden tehokkuudesta. Näistä eri yritysten luvuista on kerätty teollisuuden keskiarvot, jotka näkyvät kuvassa 3.





Kuva 3. OEE-pisteytyksen rajapyykit kansainvälisesti. Tuotannon tehokkuuden keskiarvo on maailmanlaajuisesti 60 % tasolla. [8.]

Nakajiman suosittama hyvä OEE-arvo syntyisi, kun käytettävyys on 90 %, nopeus on 95 % ja laatu 99 %. Näitä lukuja käytettäessä OEE-arvoksi saataisiin 84 %, mikä menee maailmanluokan tasolle. Nakajima antoi laatu-kertoimelle erittäin korkean arvon, mikä ei ole sattumaa, sillä osatekijät eivät ole kustannusten näkökulmasta samanarvoisia. Laatu on kaikista tärkein kerroin, sillä menetetyt ajan lisäksi menetetään myös tuotteen valmistukseen kuluva materiaaleja. Heikolla laadulla on suuri merkitys yrityksen asiakkaille, sillä viallisia tuotteita voi päästä tuotantolinjasta hyväksytysti läpi myyntiin. Laadulla on kolmesta kertoimesta suurin vaikutus yrityksen imagoon.

#### Käytettävyys (Availability)

Käytettävyys ottaa huomioon kaikki tilanteet, jotka pysäyttävät tuotannon tarpeeksi pitkäksi ajaksi, että sen syyn löytämiseen kannattaa käyttää aikaa. Yleensä tähän kuuluvat pysähdykset, jotka kestävät muutaman minuutin tai pidempään. Käytettävyys lasketaan jakamalla käyntiaika suunnitellulla tuotantoajalla kaavan 3 mukaisesti.

$$\text{Käytettävyys} = \frac{\text{Käyntiaika}}{\text{Suunniteltu tuotantoaika}} \quad (3)$$

Käyntiaika on suunniteltu tuotantoaika, josta vähennetty pysähdysaika, joka sisältää kaikki eri syiden aiheuttaman tuotannon pysähdykset. Käyntiajan laskemiseen tarvitaan pysähdysten yhteenlaskettu aika, jonka selvittämisen jälkeen voidaan käyttää kaavaa 4.

$$\text{Käyntiaika} = \text{Suunniteltu tuotantoaika} - \text{Pysähdysaika} \quad (4)$$

Pysähdysajan määrittämisessä on vaihtelua yritysten välillä. Vaisalalla seisokkiaikoja, jolloin tuotantolaite ei ole toiminnassa koska tuotteita on jo valmistettu tarvittava määrä, ei lasketa mukaan pysähdysaikaan. Käytettävyyden tärkein tarkoitus Vaisalalla on ilmaista, onko tuotantoasema käytössä tai käytettävissä sillä hetkellä. Käytettävyyden rinnalle on lisätty toiminta-aika, johon perehdytään tarkemmin kappaleessa 4. [6; 1, s. 20; 9, s. 13.]

### Nopeus (Performance)

Osatekijöistä nopeus ottaa huomioon kaiken, mikä aiheuttaa tuotantoprosessin toimimisen alhaisemmalla nopeudella kuin teoreettinen maksiminopeus. Tuotannon nopeus riippuu valmistettavasta tuotteesta, ja operaattorikeskeisissä prosesseissa työntekijällä on selvä vaikutus nopeuteen. Nopeus on kuvan 1 mukaan netto tuotantoajan ja tuotantoajan välinen osamäärä. Nopeus lasketaan kaavan 5 mukaan.

$$\text{Nopeus} = \frac{\text{netto tuotantoaika}}{\text{tuotantoaika}}$$

$$\text{Netto tuotantoaika} = \text{ideaalisykli aika} * \text{kokonaiskappalemäärä}$$

$$\text{Nopeus} = \frac{(\text{ideaalisykli aika} * \text{kokonaiskappalemäärä})}{\text{tuotantoaika}} \quad (5)$$

- Netto tuotantoaika on nopein teoreettinen aika, joka kuluu halutun tuotemäärän valmistamiseen.
- Tuotantoaika on aikaväli, jolloin laite on toiminnassa ilman pysähdyksiä.
- Ideaalisykli aika on teoreettinen nopein aika, joka kuluu yhden tuotteen valmistamiseen.
- Kokonaiskappalemäärä tarkoittaa valmistettavien tuotteiden määrää, johon sisältyy sekä hyvät- että vialliset kappaleet.

Toinen tapa laskea nopeus on käyttämällä ideaalituotantonopeutta, joka on ideaalisykliajan käänteisarvo kaavan 6 mukaisesti.

$$\text{Nopeus} = \frac{\text{kokonaiskappalemäärä}}{\text{tuotantoaika} * \text{ideaalituotantonopeus}} \quad (6)$$

Nopeus ei koskaan saisi ylittää 100 %. Näin käydessä ideaalisykli aika on säädetty liian pitkäksi. OEE-arvon on tarkoitus kuvata tuotantolaitteen tehokkuutta sen teoreettiseen maksimitehokkuuteen, joten osatekijät eivät voi ylittää 100 %:a. [6; 1, s. 20; 9, s. 15.]

### Laatu (Quality)

Osatekijöistä laatu ottaa huomioon valmistetut tuotteet, jotka eivät täytä laatuvaatimuksia. Tähän lukeutuvat rikkinäiset sekä hylätyt tuotteet. OEE-laskennassa laatuun suhtaudutaan First Pass Yield -periaatteella, jossa hyväksi tuotteiksi lasketaan vain ensimmäisellä yrityksellä läpipäässeet tuotteet. Laatu muodostuu hyväksytyyn kappalemäärään ja kokonaiskappalemäärän suhteesta, joka on esitetty kaavassa 7.

$$Laatu = \frac{\text{hyväksytty kappalemäärä}}{\text{Kokonaiskappalemäärä}} \quad (7)$$

Kappalemääriä laskettaessa uusittavat kappaleet näkyvät laskuissa vain yhtenä tuotteena, vaikka sama kappale kulkisi tuotantolinjan läpi kuusi kertaa. Tuotteita laskeva ohjelma tunnistaa samalla sarjanumerolla olevat tuotteet, ja laskee niiden olemassaolon vain kerran. Kaavassa 8 uudelleenajon käyneet kappaleet ja poistoon laitettavat kappaleet ovat määriteltä omia muuttujina, mutta muodostavat yhdessä hylättyjen kappaleiden määrän:

$$\text{Hyväksytty kappalemäärä} = \text{kokonaiskappalemäärä} - \text{uusittavat kappaleet} - \text{reklamaatiokappaleet} \quad (8)$$

[6; 1, s. 21; 9, s. 16.]

### 2.7 OEE-suunnitteluun vaikuttavia tekijöitä

Suunniteltaessa OEE-järjestelmää on hyvä ottaa huomioon sen tärkein päämäärä: tuotantoa hidastavien tekijöiden tunnistaminen. OEE-menetelmää sovelletaan nykyään laajasti teollisuuden piirissä; osa yrityksistä on huomannut potentiaalinen imagon kohottami-

seen hyvää OEE-arvoa mainostamalla, mutta tämä kasvattaa riskiä arvojen keinotekoiseen kasvattamiseen. OEE:n alkuperäinen suunnittelija Nakajima viittaa tähän riskiin puheissaan:

“Overall Equipment Effectiveness (O.E.E.) percentages should only be used to compare equipment to itself over time, never to compare different equipment or equipment running different products.” –Nakajima [4.]

OEE:n käyttötarkoituksiin ja rajoituksiin on hyvä kiinnittää huomiota järjestelmää suunniteltaessa. Realististen lukujen saaminen on tärkeämpää, kuin korkeiden prosenttilukujen tuottaminen. Käytettävyyttä laskettaessa ennakkohuollot sekä kahvitauot kannattaa laskea häviöiden joukkoon, sillä näiden vaikutus voidaan mitätöidä jaksottamalla työvuorot oikein. Nopeuden ideaalisykliaikoja on päivitettävä säännöllisesti, mikäli tuotantoprosessiin tehdään parannuksia. Ideaalisykliajan on oltava mahdollisimman lähellä teoreettista maksiminopeutta. Työntekijäkeskeisten prosessien nopeutta määrittäessä mittanäytteitä kannattaa ottaa useita pitkältä aikaväliltä prosessin eri vaiheet eritellen; jokaisen vaiheen nopein suoritus valitaan ideaalisykliaikaan.

## 2.8 Ohjelmasuunnittelu

Ohjelmasuunnittelu on projektin alkuvaiheessa tehtävä prosessi, jossa toteutettavan ohjelman rakenne määritellään ja ohjelman toiminnalle oleelliset osat eritellään. Ohjelmasuunnittelu koostuu esitutkimuksesta, vaatimusmäärittelystä ja suunnitteluvaiheesta. Prosessi antaa projektin jäsenille yhteisen kuvan tavoitteista ja helpottaa työtehtävien jakamista. Ohjelmaluonnos esitellään asiakkaille ennen koodaamisen aloittamista, jolloin säästytään tarpeettomalta työltä ja tavoitteet täsmäävät paremmin asiakkaan tarpeita. Ohjelmasuunnitteluun kuuluu ongelmanratkointia ja ohjelmaratkaisujen löytämistä. Tähän sisältyy sekä alhaisen tason komponentti- ja algoritmivalinnat että korkean tason arkkitehtuurisuunnittelu. [12, s.9; 13.]

Esitutkimusvaihe edeltää määrittelyvaihetta. Esitutkimus kartoittaa asiakkaan perustason vaatimukset järjestelmälle. Tutkimuksen päätavoite on vastata kysymykseen miksi järjestelmä tulisi tehdä tai jättää tekemättä. Vaatimusten selkeä määrittely on monien ohjelmistoprojektien suurin ongelma, sillä asiakkaan ja ohjelmoijan mielikuva halutuista

tavoitteista ei välttämättä ole sama, vaikka molemmat puhuvat aiheesta samoja termejä käyttäen. [12, s.10; 13.]

Esitutkimuksesta siirrytään määrittelyvaiheeseen. Ohjelman tarkoituksen pohjalta määritellään järjestelmän toiminnallisia vaatimuksia, jotka dokumentoidaan. Tähän dokumenttiin kuvataan järjestelmän rajoitukset ja toiminnalliset vaatimukset. Määrittelyssä on tärkeää huomioida kaikkien järjestelmää käyttävien henkilöiden näkökulma. Ohjelman hyödyllisyys riippuu usein ihmisten mielipiteisiin sen käytettävyydestä, joten psykologisten taitojen käyttö on tarpeellista. Näin ollen määrittely tehdään osittain haastatteluiden kautta, jossa selvitetään käyttäjien odotukset ja tarpeet ohjelman suhteen. Ohjelmasuunnittu vastaa luvussa 2 mainittuja tieteellisen tutkimuksen menetelmiä. Määrittely listaa esimerkiksi käytettävät ohjelmapohjat, ohjelman tekniset vaatimukset ja käyttötavat. Määrittely muuttaa asiakasvaatimukset täsmällisiksi ohjelmistovaatimuksiksi. [12, s.9; 13.]

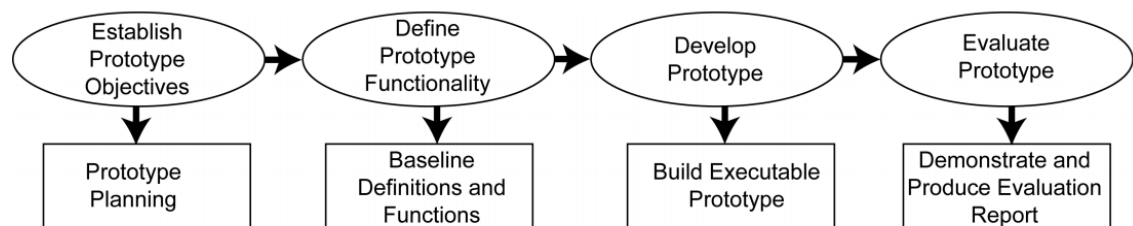
Suunnitteluvaiheessa aikaisemmat määriytykset muutetaan konkreettisiksi ohjelmarakenteiksi. Suunnitteluun sisältyy käyttöliittymän ulkonäkö ja rakenne, ohjelmakoodin osaluokat ja järjestelmäarkkitehtuuri. Suunnitteluvaihe mahdollistaa järjestelmän ensimmäisen version rakentamisen. Nykyään suosituimmat ohjelmasuunnittelumallit toimivat kiertävän periaatteen pohjalta, jossa projektin aikana ohjelmoijat ja käyttäjät ovat tiiviissä yhteistyössä, ja ohjelman suunnitteluun sekä määriytyksiin palataan uudestaan säännöllisin väliajoin. Tällä menetelmällä projektista saadaan jatkuvaa palautetta ja varmistetaan vaatimusten paikkansapitävyys. Yksi kiertävä ohjelmointimalli on prototyypiohjelmointi, jossa lopullisesta ohjelmasta tehdään ensin yksinkertaistettu prototyyppi, jonka avulla asiakas arvioi ohjelman hyödyllisyyttä. Tässä projektissa käytetään prototyypimallia, johon perehdytään tarkemmin seuraavassa kappaleessa. [12, s.5–18; 13.]

## 2.9 Prototyypiohjelmointi

Ohjelmasuunnitteluhistorian aikana on syntynyt useita eri käyttötarkoituksiin optimoituja kehitysmalleja. Prototyypiohjelmointi on yksi parhaista lyhyessä ajassa monimutkaisen järjestelmän mallintamiseen soveltuvista menetelmistä. Prototyyppi on yksinkertaistettu versio lopullisesta ohjelmasta, johon sisällytetään vain testattavan ohjelmaosion toimi-

vuuden kannalta kaikista oleelliset osat. Tarpeesta riippuen prototyyppi saattaa simuloida vain tiettyjä ominaisuuksia, kuten tietokantaa tai käyttöjärjestelmää, ja olla täysin erilainen lopullisesta ohjelmasta. Prototyypin avulla kehitetään ohjelmarakennetta ja varmistetaan, että alkuperäiset määritykset vastaavat haluttua todellisuutta. [13; 1; 11.]

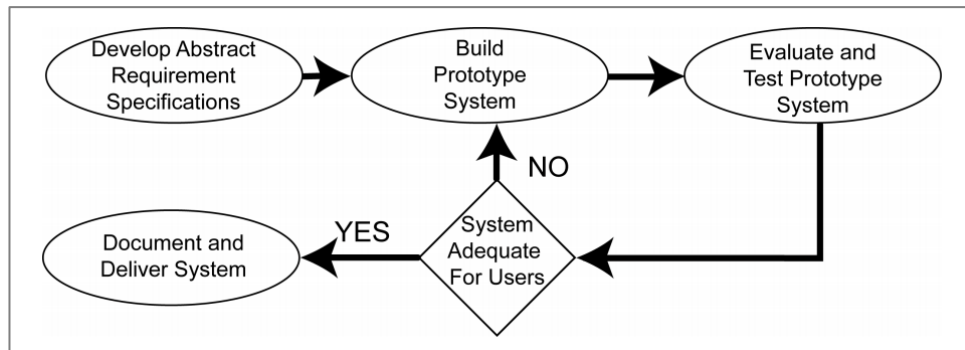
Prototyypimalli seuraa normaalia ohjelmasuunnittelukaavaa, mutta tarpeisiin rajatussa koossa. Prototyypin kehitys etenee esisuunnittelusta vaatimusmäärittelyyn ja suunnitteluvaiheeseen kuvan 4 mukaisesti.



Kuva 4. Prosessimalli prototyypin suunnitteluun [11, s. 2.]

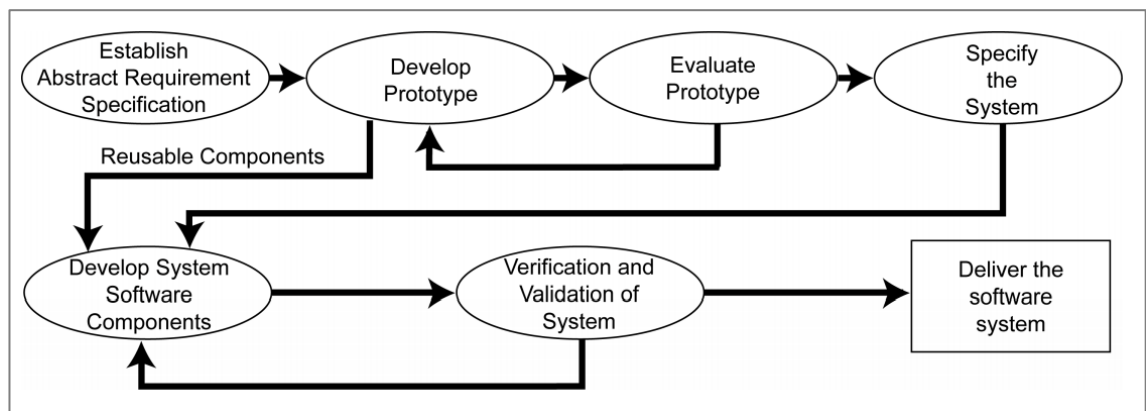
Prototyypimallintamisella on useita hyötyjä. Väärinkäsitykset kehittäjän ja asiakkaan välillä voidaan tunnistaa ohjelman toimintaa simuloimalla. Ohjelman käytön epäselvyydet ja puuttuvat funktiot tulevat esiin, mikä voidaan korjata seuraavassa versiossa. Yrityksen johto ymmärtää prototyypin tulosten avulla ohjelmasta saatavan potentiaalisen hyödyn ja pystyy luomaan riskinarvioinnin projektin kannattavuudesta. Yksinkertaistetun prototyypin luominen on kustannuksiltaan paljon halvempaa koko ohjelman luontiin verrattuna. Tutkimukset ovat todistaneet, että prototyypimallinnus leikkaa lopullisia kokonaiskustannuksia ja vähentää ongelmia vaatimusmäärittelyssä. [11, 13.]

Prototyypiohjelmoinnissa on kaksi eri vaihejakomallia: kehittyvä prototyypimalli (kuva 5) ja hylättävä prototyypimalli (kuva 6) Kehittyvän prototyypin tavoitteena on luoda rakenteeltaan hyvin vankka prototyyppi, jota jatkuvasti jalostetaan paremmaksi. Tämä prototyyppi toimii lopullisen ohjelman ytimenä, jonka ympärille lisätään loput ohjelman ominaisuudet. Kehittyvässä prototyypimallissa voidaan hyödyntää laajemmin aikaisemmassa projektissa tehtyä ohjelmapohjaa ja keskittyä ohjelmasta puuttuviin ominaisuuksiin. [13; 11; 1.]



Kuva 5. Kehittyvä prototyypimalli [11, s. 5.]

Hylättävässä prototyypimallissa kehitetään ohjelmamalli, joka projektin lopussa hylätään ja sen hyödylliset osat kierrätetään seuraavaan ohjelmaversioon alla olevan kuvan 6 mukaisesti. Hylättävä prototyyppi sisältää vain halutut osat lopullisesta ohjelmasta, ja prototyypin pääpaino on nopeudessa luoda toimiva malli. Nopeuteen keskittyessä projektin rakenne on usein melko epävirallinen ja kehittäjä keskittyy tekemiseen dokumentoinnin sijasta. Nopea valmistuminen mahdollistaa käyttäjärjestelmän ja tulosten aikaisen testaamisen. [1; 11; 13.]



Kuva 6. Hylättävä prototyypimalli [11, s. 6.]

Hylättävä prototyypimalli soveltuu tähän OEE-projektiin paremmin, sillä projektin tavoitteena on luoda toimiva ohjelma lyhyessä ajassa ja visualisoida käyttäjille järjestelmän käytettävyys.

### 3 OEE-järjestelmän muodostaminen Vaisalalla

Luku muodostaa ymmärryksen projektin etenemisestä ja ratkaisusta Vaisalalla. Tässä kuvatus projektin rakenne seuraa kappaleessa yksi mainittuja tutkimusmenetelmiä ja kappaleen kaksi hävitettävää prototyypimallia. Kirjoitus etenee esitutkimuksesta ja järjestelmän vaatimusmäärittelystä itse ohjelmarakenteen suunnitteluun. Lopulliseen ohjelmarakenteeseen päätyminen jälkeen itse järjestelmä eritellään kolmena osana C#-ohjelmaan, tietokantaan ja visualisointiohjelmaan. Ohjelma- ja tietokantarakenteet pohjautuvat kappaleessa kaksi kuvattuihin teorioihin ja mukautuvat Vaisalalan vaatimuksiin.

#### 3.1 Tutkimustyö ja projektisuunnittelu

Projekti alkoi Vaisalalan testausylläpitotiimin managerin pyynnöstä. Vaisalalla oli tarve saada tarkempaa tietoa tuotantoasemien käytettävyydestä ja OEE-lukemista. Testausylläpidon manageri tarjosi projektia OEE-järjestelmän prototyypin luomisesta, jonka tutkija otti vastaan. Projektin ensimmäisten kahden viikon aikana selvitettiin tutkimukseen tarvittavan tietopohjan laajuutta sekä tutustuttiin Vaisalalan ylemmän toimihenkilön käytäntöihin osana testausylläpitotiimiä. Luvussa 1 mainittuja tutkimusmenetelmiä hyödyntäen projektille kehitettiin toimintasunnitelma. Ulkoisten lähteiden ja Vaisalalan sisäisten tietojärjestelmien lisäksi kvalitatiivisena tutkimusmenetelmänä käytettiin eri asiantuntijoiden haastattelemista.

##### 3.1.1 Haastattelut

Haastattelujen tarkoitus oli selvittää, minkä tyyppisestä tiedosta Vaisalala saisi maksimaalisen hyödyn OEE-järjestelmän kehittämisessä. Työn reliabiliteetin optimoimisen kannalta oli tärkeää tutkia konseptia monesta eri näkökulmasta, joten haastattelun osallistujiksi haalittiin asiantuntijoita monilta toimialueilta. Projektille olennaisen tiedon keräämisen varmistamiseksi oli tärkeää määrittää sopivimmat toimihenkilöt ja asiantuntijat haastatteluihin. Haastatteluihin pyydettiin useita henkilöitä 'Development', 'Quality and Production', 'Engineering' sekä 'Test Engineering' -osastoilta. Haastatteluissa käytettiin liitteenä 2 olevaa pohjaa. Haastattelutilanteet koostuivat kahdesta kolmesta osasta, joista ensimmäisessä tutkija esitteli OEE-järjestelmän, toisessa esitettiin liitteen 2 kysymykset



ja kolmannessa keskityttiin tarkemmin teknisiin yksityiskohtiin. Haastattelut kestivät keskimäärin kaksi tuntia, joiden aikana tavoitteena oli sovittaa projekti eri osastojen käyttötarpeisiin ja perehtyä heidän työnkuvaansa. Oli tärkeää selvittää, mistä asiantuntijat keräävät tietoa ja miten OEE-järjestelmä voi mahdollisesti helpottaa heidän työskentelyä.

Strukturoitujen kysymyksien sijasta käytettiin valmiiksi määriteltyjä teemoja, jotka mahdollistivat ilmiön monipuolisen käsittelyn. Asiantuntijat pohtivat OEE-järjestelmän hyödyllisyyttä ja sen konkreettista soveltamista Vaisalán työympäristöön. Täten jokainen avoin haastattelutilanne kehittyi erilaiseen suuntaan riippuen haastateltavan erityisosaamisesta. Tutkijan tehtävänä oli laatia ja valikoida mahdollisimman oleellinen tieto projektin tavoitteen saavuttamiseksi. Haastatteluprosessin aikana tutkijan otti muistiinpanoja tiedon tallentamiseksi myöhempää analyysia varten. Opinnäytetyön haastattelut edesauttoivat tutkimuksen mahdollisten haasteiden perinpohjaista selvitystä. Yleisesti haastattelut tiedonkeruumetodina ovat keskimääräistä kalliimpi vaihtoehto. Koska haastattelut tapahtuvat työajan ohella, se ei tule kuitenkaan viemään merkittävästi aikaa projektista. Haastatteluista saatua tietoa tuki viikoittaiset palaverit työnjohtajan ja tutkijan kesken, joissa tavoitteena oli keskustella projektin etenemisestä ja mahdollisista haasteista. Tämän lisäksi tiedonsaannissa hyödynnettiin eri koulutustilaisuuksia, jotka järjestettiin Vaisalán tuotantoasemilla. Koulutustilaisuudet hajautettiin kahden kuukauden aikajaksolle.

Kvalitatiivisen tutkimustiedon perusteella voitiin havaita, että ohjelman, joka kattaa jokaisen tuotantoaseman tarpeet on mahdotonta toteuttaa opinnäytetyöhön annetussa ajassa. Tämän seurauksena tutkija ehdotti projektin uudelleenmäärittämistä viikoittaisessa kehityspalaverissa. Projektin uudeksi tavoitteeksi päätettiin luoda prototyyppi OEE-järjestelmästä. Prototyyppi on yleisesti teollisuuden tuotekehitystoiminnassa käytetty termi ensimmäisestä testiversiosta. Koska prototyypin luominen tuotekehitysprosessin alussa on välttämätöntä, työn rajaaminen OEE-prototyypin kehittämiseen takaa Vaisalalle merkittävän hyödyn ja pohjatyön OEE-järjestelmän kehittämisprosessin elämänkaarella. Jotta työn tavoite pysyy realistisena, prototyyppi päätettiin luoda pelkästään yhteen tuotantosoluun, joka koostuu kolmesta eri asemasta.

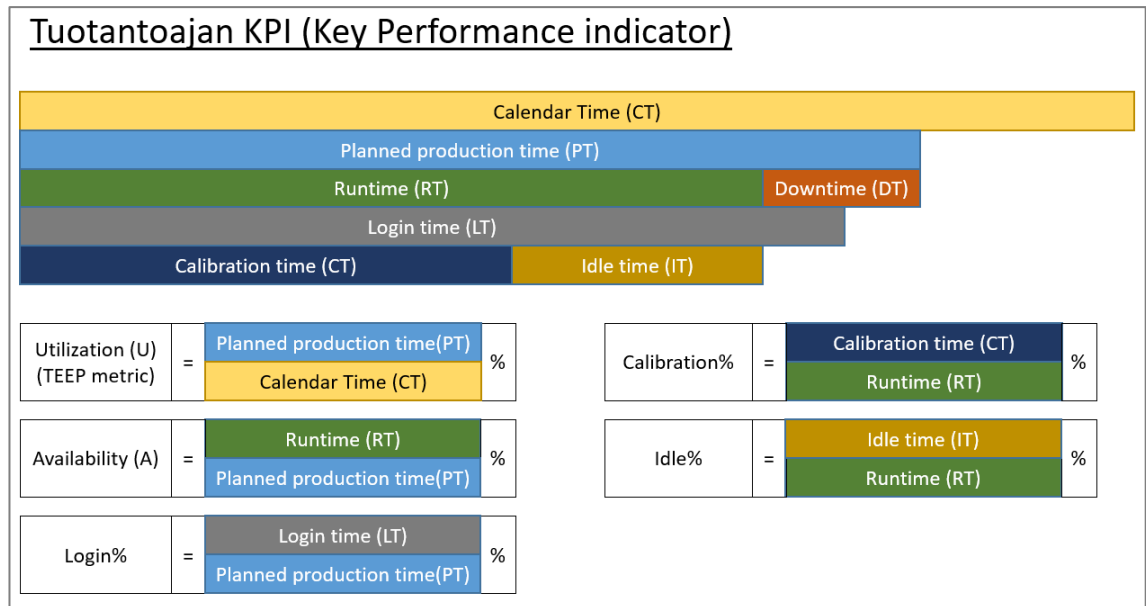
### 3.1.2 Käyttäjät

OEE-järjestelmän käyttäjäkunta jakautuu laajasti yrityksen eri osastoille. OEE-järjestelmän tekniikka vastaa testausylläpito- sekä engineering-osastojen osaamista, joten järjestelmän ylläpidosta ja huollosta vastaavat henkilöt valitaan näistä osastoista. Ylläpito toimii läheisessä yhteistyössä yrityksen ohjelmistosuunnittelijoiden ja tuotannon insinöörin kanssa. Uusia tuotantosoluja järjestelmään lisättäessä tuotantoinsinöörit antavat tarvittavan tiedon ylläpidolle tiedostojen päivitystä varten.

Asemien OEE-tietoja hyödynnetään kaikilla yrityksen tasoilla. Asemilla toimivat operaattorit saavat visualisointiohjelman kautta tietoa päivän tuotemääristä sekä tuotannon nopeudesta. Järjestelmän kautta operaattorit käsittävät tuotannon nykytilan ja voivat osallistua tehokkuuden parantamiseen. Yrityksen keskitasolla OEE-tietoja käyttäjiin kuuluvat managerit, plannerit sekä tuotantoinsinöörit. Tuotantotiimien managerit seuraavat asemien tehokkuutta sekä pitkällä aikavälillä että reaaliajassa. Plannerit arvioivat OEE-tietojen avulla tilausten valmistumisnopeutta ja suunnittelevat tämän pohjalta realistiset tilausmäärät. Tuotantoinsinööreille tiedoista on huomattava hyöty tuotantoasemien pulonkaulojen tunnistamisessa. Ylempään johtoon siirryttäessä OEE-tietoja käytetään osastojen tuotantomäärien ja tehokkuuden arvioimiseen. Tietojen pohjalta päätetään mahdollisista investoinneista lisäkapasiteettia tarvitsevien asemien tehokkuuden nostamiseksi.

### 3.2 OEE:n muuttujien määrittely ja projektin rajaus

Vaisalalle OEE-järjestelmän tärkeimpiä tarkoituksia on visualisoida tuotantolaitteiden käytettävyys. Käytössä on olemassa jo työkaluja, jotka visualisoivat asemien saantoa sekä nopeutta, mutta käytettävyyden analysoinnissa on todettu puutteita. Vaisalän huomio käytettävyydessä liittyy tuotantoasemien reaaliaikaisen saatavuuden hahmottamiseen. Virtaustehokkuuteen erikoistuneessa tuotannossa viikoittaiset tuotantomäärät riippuvat asiakkailta tulleiden tilausten määrästä. Tuotanto alkaa välittömästi uuden tilauksen tullessa ja pysähtyy, mikäli tilauksia ei ole. Tämän nopean reagoinnin kannalta on tärkeää tietää, onko tuotantoasema juuri tällä hetkellä käynnissä vai vapaana uuden tilauksen aloittamiseen. Käytettävyysarvo yksin ei kuvaa tarpeeksi hyvin aseman tämänhetkistä saatavuutta, joten osatekijöitä laajennetaan kuvan 7 mukaisesti.



Kuva 7. Vaisalan tarpeeseen suunniteltuja osatekijöitä asemien tarkemman käytettävyyden mittaamiseen.

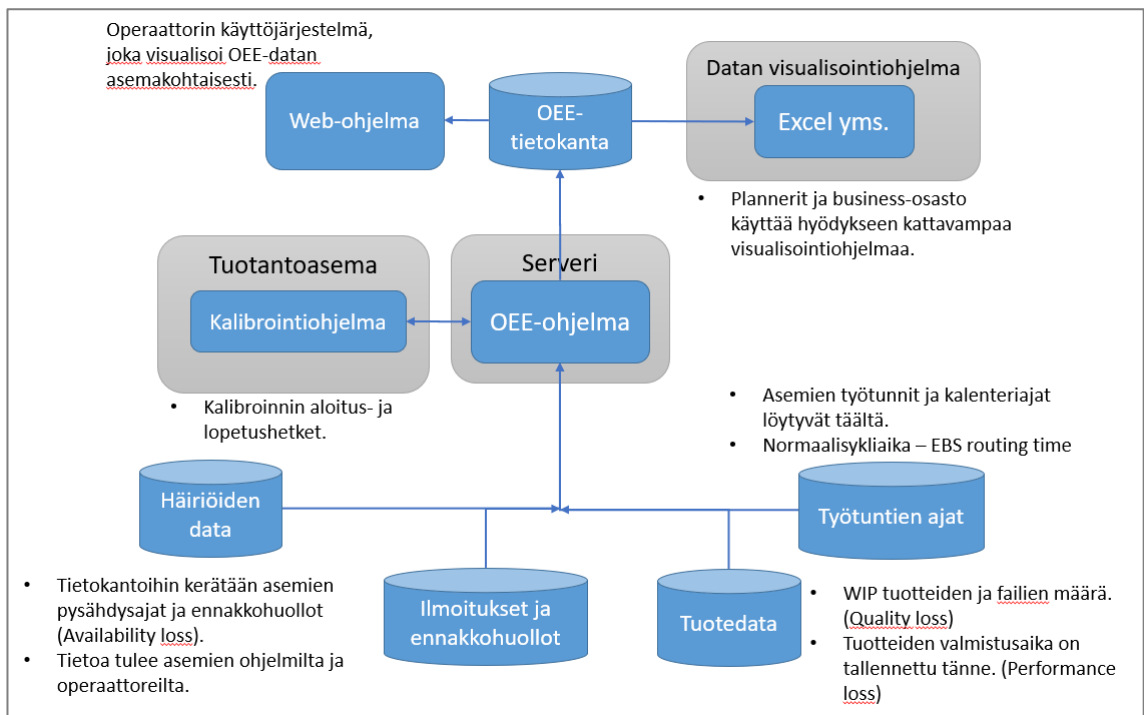
Login-prosentti liittyy operaattorin sisäänkirjautumiseen työasemalle. Login-arvon avulla voidaan seurata, kuinka ison osan ajasta työasema on operaattorin käytössä. Login-arvon voi liittää leimaisinjärjestelmään, jossa operaattori asemalle tullessaan kirjautuu siihen sisään leimaisinlaitteella.

Tuotteiden kalibrointiohjelmat ovat yksi yleisimmistä järjestelmistä Vaisalan tuotantoasemilla. Kalibrointiajan (Calibration time) ja seisokkiajan (Idle time) tarkoitus on seurata kalibrointiohjelman käytettävyyttä. Tuotantoasemilla työ koostuu vaiheista, joissa operaattori ja ohjelmisto tekevät eri tehtäviä. Ohjelman kalibroidessa sensoreita operaattori valmistelee seuraavaa tuotetta kalibrointiin ja kasaa jo kalibroituja tuotteita valmiiksi. Yleensä operaattorin tuotantovaiheet ovat ohjelmaa pidempiä, jolloin ohjelma odottaa seisokitilassa. Tavallisessa OEE-järjestelmässä nämä seisokit merkattaisiin käytettävyyshäviöiksi, mikä vääristäisi käytettävyyssastetta ja sotkisi seisokit oikeiden aseman pysähdystilanteiden sekaan. Kalibrointiaikaa ja tuotantoaikaa vertaamalla voidaan tunnistaa eri tuotantoasemien pullonkauloja. Jos asemalla on pitkiä seisokkiaikoja tuotantoaikaan verrattuna, voidaan päätellä, että asemalle tarvitaan lisää operaattoreita, kun taas seisokkiajan ollessa olematon aseman pullonkaulana toimii todennäköisesti ohjelma.

### 3.3 Ohjelmarakenne

Projektin esitutkimuksen ja haastattelujen avulla ohjelmalle määritettiin tarvittavia muuttujia. Näiden muuttujien arvojen löytäminen oli ohjelman rakennuksen ensimmäinen vaihe. Vaisalan kattavan tietoverkoston ansiosta arvot pystyttiin hankkimaan suoraan tuotantoaseman ohjelmasta tai useista tuotantotietoa säilöivistä tietokannoista. Tietokantojen sisältöä tutkittaessa ilmeni, että ne sisältävät valmiiksi lähes kaiken OEE-järjestelmän tarvitseman pohjatiedon. Tämän perusteella päätettiin, että OEE-järjestelmä kerää tarvitsemansa tiedon muista tietokannoista, eikä kommunikoi suoraan tuotantoasemien kanssa. Tiedon kerääminen tuotantoasemasta olisi vaatinut huomattavia muutoksia aseman ohjelmapohjaan, mikä hidastaisi projektin etenemistä huomattavasti.

Vaisalan tiedonsiirron digitalisoitumisen seurauksena on syntynyt kymmeniä ohjelmia kattava tietoverkko, jossa ohjelmien jalostama tieto kerätään eri käyttötarkoituksia varten luotuihin tietokantoihin. OEE-kaavojen laskemiseen tarvitsemia muuttujia etsittäessä huomattiin, että tarpeellinen tieto löytyy pääsääntöisesti neljästä eri tietokannasta, joita kuva 8 havainnollistaa.



Kuva 8. Yleiskuva OEE-järjestelmän eri osista.

Luvussa 2.6 puhuttiin kolmesta OEE-arvoon vaikuttavasta kertoimesta (käytettävyys, nopeus ja laatu). Kertoimet koostuvat osatekijöistä, kuten tuotantoajasta, jotka jalostettiin tietokantojen datan pohjalta. Käytettävyiden muodostamiseen vaikuttavat tekijät ovat tuotantoaika sekä pysähdysaika. Pysähdysaika (availability loss) muodostettiin keräämällä tuotantoasemaan vaikuttavat häiriötilanteet ja ennakkohuollot, joiden tieto on kerätty kuvan 8 mukaisesti kahteen tietokantaan. Tuotantoajan pohjana käytettiin asemalla toimivien työntekijöiden työtunteja, jotka löytyivät henkilöstöhallinnon tietokannasta. Työtunteja täsmennettiin vertaamalla niitä kalibrointiohjelman ilmoittamiin toiminta-aikoihin. Näin työtunneissa huomioitiin asemalla tehty ylityöt. Nopeuden mittaamiseen tarvittiin tuotantoajan lisäksi tuotteiden valmistusaika ja normaalisykلياika. Valmistusaika saatiin kalibrointiohjelman täydentämästä tietokannasta, jossa on jokaisen tuotteen valmistuksen aloitus- ja lopetusajankohta. Normaalisykلياika kertoi tuotteen keskimääräisen valmistusnopeuden, joka selvitettiin Vaisalalla aikaisemmin järjestetyssä tutkimuksessa. Viimeinen kerroin on laatu, jonka tekijät rakentuivat hyväksytyjen ja hylättyjen tuotteiden lukumääristä. Tietokantojen data siirrettiin SQL-hakujen avulla OEE-ohjelmaan.

OEE-ohjelma toimii koko järjestelmän ytimenä, jossa itse laskutoimitukset tapahtuvat. Ulkopuolisista tietokannoista hankittu tieto jalostettiin luvussa 2.6 määritetyiksi kertoimien osatekijöiksi, ja tämän perusteella jokaiselle asemalle mitattiin OEE-kertoimet. Ohjelma toimi palvelimella ja kommunikoi SQL-hakujen kautta tietokantojen ja mahdollisesti tuotantoaseman kalibrointiohjelman kanssa. Tässä projektissa ohjelman rakenne pidettiin yksinkertaistettuna prototyypimallina, joka seurasi vain kahden tuotantoaseman tehokkuutta, mutta lopullisen ohjelman oli tarkoitus olla rakenteeltaan modulaarinen ja helposti laajennettava. OEE-ohjelman toimintaan perehdytään tarkemmin seuraavassa luvussa.

OEE-ohjelma mittausten lopputuloksena saadut tehokkuusarvot siirrettiin SQL-hauilla toiselle serverille OEE-tietokantaan. Erillisen tietokannan tarkoituksena oli mahdollistaa usean ohjelman samanaikainen pääsy OEE-tietoihin. Tietokanta mahdollisti laajan datamäärän tallentamisen helposti käytettävään muotoon, jota visualisointiohjelmat pystyivät rajaamaan käyttäjän tarpeen mukaan. Tietokanta mahdollisti myös datan reaaliaikaisen visualisoinnin ja päivityksen eri ympäristöissä. Tietokannan suunnitteluun palataan tarkemmin luvussa 3.5.

Tietokannan käyttö tallennuspohjana mahdollisti datan visualisoinnin useilla eri menetelmillä. OEE-tietokannan ansiosta mikä tahansa Vaisalán lähiverkossa oleva ohjelma pystyi käyttämään dataa hyväksi. OEE-dataa suunniteltiin visualisoitavaksi operaattoreiden käyttämässä web-ohjelmassa, mutta tämän projektin aikana visualisoinnissa keskityttiin Excel-pohjaisen ratkaisun tekemiseen. Excel yhdistettiin VBA-makroilla tietokantaan, mikä mahdollisti käyttäjän määrittelemän datan automaattisen visualisoinnin taulukkomuotoon. Datan visualisoinnista puhutaan lisää luvussa 3.6.

### 3.4 OEE-ohjelma

Palvelimella toimiva OEE-ohjelma suunniteltiin järjestelmän ytimeksi, joka analysoi ja tallentaa tehokkuusseurannassa olevien tuotantoasemien dataa. Suunnitteluvaiheessa pohdinnan alla oli idea rakentaa pieni ohjelma jokaiselle tuotantoasemalle, ja jokainen ohjelma raportoisi aseman tehokkuustiedot yhteen tietokantaan. Idea hylättiin, koska ohjelman asentaminen jokaiselle asemalle ja ohjelmien ylläpito olisi tarpeettoman työlästä yhden ohjelman ylläpidon sijasta. Lopullinen ohjelmamalli rakennettiin toimimaan serverillä ja kommunikoimaan lähiverkon kautta tietokantojen ja tuotantoasemien kanssa. Tämä mahdollisti selkeän ohjelmarakenteen, mikä helpotti ohjelman ylläpitoa sekä jatkokkehitystä.

OEE-ohjelma kirjoitettiin C#-ohjelmointikielellä käyttäen Microsoft Visual Studio 2017 -kehitysympäristöä. C#-kieli valittiin ohjelman kirjoituskieleksi, koska se on laajasti käytössä Vaisalalla ja soveltuu hyvin tietokantojen kanssa kommunikointiin. Tärkeänä tavoitteena projektissa oli luoda koodipohja, joka on tuttu muille ohjelmoijille sekä helposti laajennettavissa. Ohjelmarakenne seurasi näin ollen Vaisalán yleisiä ohjelmointikäytäntöjä.

Ohjelmointirakenne, luokat ja metodit

C#-ohjelmoinnin rakenne pohjautuu olio-ohjelmointiin. Ohjelman toiminnot jaettiin omiin luokkiin ja metodeihin. Luokkien selkeä nimeäminen käyttötarkoituksen mukaan ja jokaisen komennon jakaminen omaan metodiin teki koodista helposti luettavaa. Hyvän ohjelmoinnin käytäntöjä sovellettiin kirjoittamalla metodeille ohjeistus ja välttämällä avonais-

ten muuttujien käyttämistä. Tuotantoasemien erojen takia jokaista asemaa kohti rakennettiin oma ohjelmasekvenssi, joka määritteli kyseisen aseman tehokkuuden mittaamiseen käytettävät metodit. Kuva 9 toimii esimerkkinä OEE-laskentaan käytettävästä metodista. CalculateQuality-metodi on selkeästi nimetty ja laskee luvun 2.6.3 kaavan mukaisesti laatukertoimen.

```
46 public void CalculateQuality(double totalCount, double rejectedCount)
47 {
48     double goodCount = totalCount - rejectedCount;
49     quality = goodCount / totalCount;
50 }
51
```

Kuva 9. Metodi laatukertoimen laskentaa varten.

Ohjelmaa suunniteltaessa pyrittiin helppoon laajennettavuuteen, jossa muut ohjelmoijat lisäävät uusia tuotantoasemia järjestelmän piiriin mahdollisimman pienellä työmäärällä. Tätä tavoitetta varten mahdollisimman moni asemakohtainen muuttuja kirjoitettiin XML (Extensible Markup Language) -tiedostoihin, joiden sisältämää tietoa C#-ohjelma lukee ja käyttää. XML-tiedostoissa määritettiin jokaisen seurattavan tuotantoaseman seuraamiseen tarvittavat muuttujat, kuten aseman nimi ja tuotetunnukset. XML-tiedostoon tallennettiin tietokantayhteyksien avaimet ja ohjelman tarvitsemat SQL-haut. Hyöty XML:n käytössä on, ettei muuttujia ”kovakoodata” suoraan ohjelmaan, vaan muuttujat tallennetaan selkeisiin ja helposti muokattaviin tiedostoihin. Näin C#-ohjelma pidetään puhtaana tarpeettomista koodipätkistä ja uusia asemia lisättäessä vain XML-tiedostoja tarvitsee muokata. C#-ohjelma ei pysty muokkaamaan XML-tiedostoja, mikä varmistaa muuttujien pysyvyyden ja vähentää mahdollisia ohjelmavirheitä.

### 3.5 Tietokantamallin muodostaminen

Luku vastaa seuraaviin kysymyksiin: mikä tietokannan käyttötarkoitus oli projektissa? Miten tietokantarakenne toteutettiin? Tietokannalla oli tärkeä rooli järjestelmän muiden osien suunnittelussa. Tietokanta vastaanotti C#-ohjelman jalostamaa dataa ja säilöi sen visualisointiohjelmien tarvitsemaan muotoon. Näin ollen tietokantarakenteen täytyi soveltua molempien ohjelmien käyttöön. Tietokannan suunnitteluprosessin alussa tutkittiin,

mitä dataa tallennetaan ja kuinka data rakentuu ja liittyy ulkoisiin lähteisiin. Tieto muokattiin täsmäämään käyttäjien tarpeita.

### 3.5.1 Tietokannan vaatimukset

Vaatimusmäärittely aloitettiin käyttäjien tarpeiden selvittämisellä. OEE-data haluttiin saataville lähiverkon kautta usealle ohjelmalle eri puolilla Vaisalaa. Suurinta osaa käyttäjistä kiinnostivat tuotantoasemien tehokkuusarvot päivittäin ja kuukausittain. Lisäksi ehdotettiin aikajanan muodostamista aseman käytöstä, mikä näyttää aseman toiminnan tietyllä aikavälillä työtunnit ja pysähdykset huomioon ottaen. Tietokannan kapasiteetti suunniteltiin kattamaan asemien tehokkuusdata kymmenien vuosien ajalta, jolloin tietokanta laajenisi sisältämään satoja tuhansia tietueita.

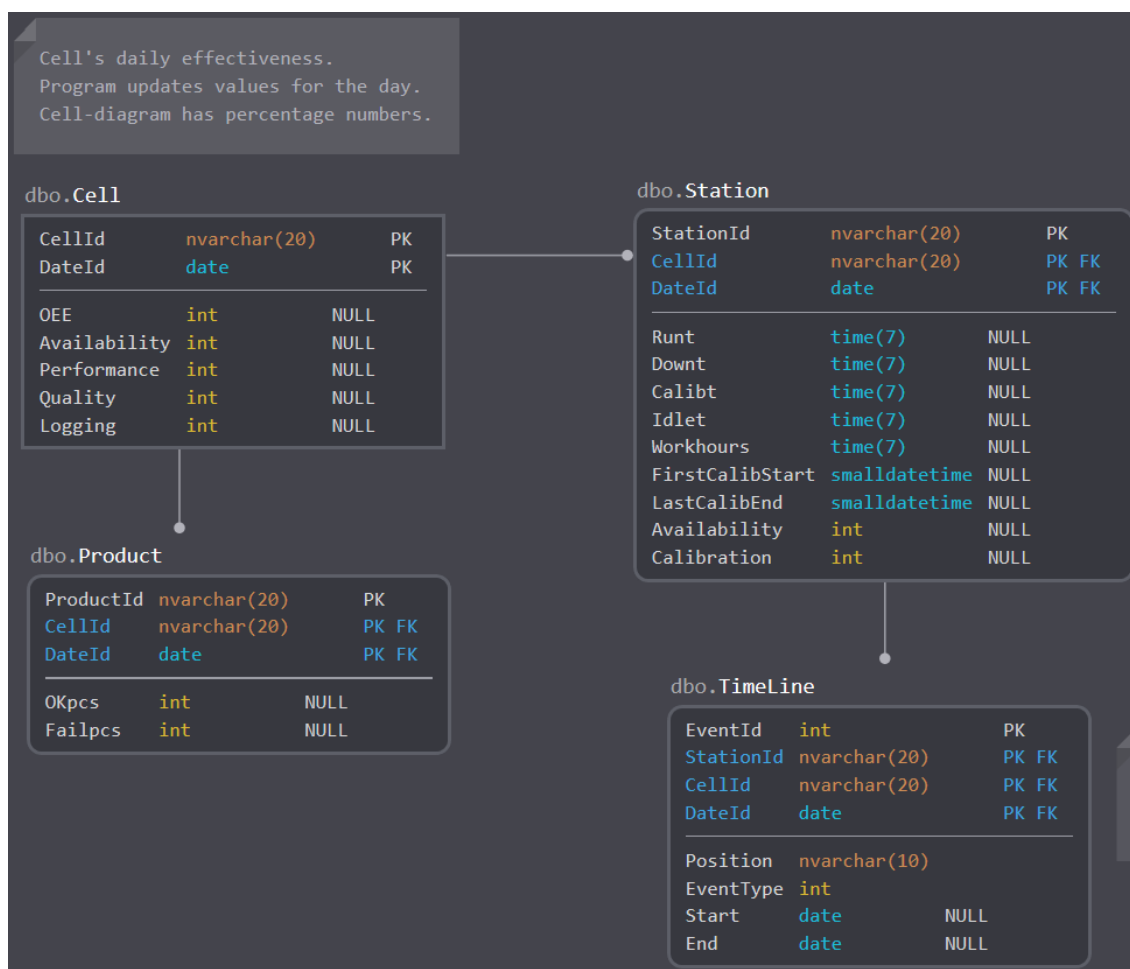
Tietokannan rakentamista varten tärkeää oli määrittää sen keskeisimmät tekijät, eli entiteetit, joiden tehokkuutta mittaamalla syntyy selkeä kuva tuotannon tehokkuudesta. Tällaisiksi entiteeteiksi osoittautuivat tuotantosolu, siihen liittyvät tuotantoasemat sekä valmistettavat tuotteet. Tuotantosolu oli yhdestä tai useammasta tuotantoasemasta koostuva kokonaisuus, jossa käydään läpi tuotteenvalmistuksen kaikki vaiheet alkukasaamisesta ja ohjelmoinnista loppupakkaamiseen. Entiteettien tunnistuksen jälkeen määritettiin niihin liittyvät ominaisuudet. Tuotantosolun alaisuuteen kirjattiin siihen liittyvät OEE-arvot, tuotantoasemaan liitettiin sen käytettävyyden mittaamiseen tarvittavat aikatekijät ja tuotteisiin yhdistettiin hyväksytyjen ja hylättyjen tuotteiden määrät. Seuraavassa luvussa oleva kuva 10 visualisoi tietokannan entiteetit ja niiden ominaisuudet.

Tietokannassa pyrittiin yksinkertaisuuteen, jossa kantaan on tallennettu vain OEE-mittauksen oleelliset tiedot. Tietokannan turhan paisumisen estämiseksi tiedot tallennettiin yhden päivän paketeiksi, jossa OEE-ohjelma työpäivän edetessä päivittää lukemia saman päivän aikaisempien tietojen päälle. Esimerkiksi tietokannassa saattoi olla aseman käytettävyyssprosentti työpäivän ensimmäisen kahden tunnin ajalta, jonka päälle ohjelma lisää viime päivityksen välillä tapahtuneen käytettävyyssarvon muutoksen.



### 3.5.2 Tietokantarakenne

Tietokanta rakennettiin SQL Server Management Studio -ohjelmalla, joka on yleisessä käytössä tietokantojen ylläpidossa ja muokkaamisessa. Ensimmäinen malli OEE-tietokannasta tehtiin Vaisalan testipalvelimelle, jossa tietokantaa pystyttiin vapaasti muokkaamaan häiritsemättä muiden ohjelmien toimintaa. Tietokannan samanaikainen rakentaminen C#-ohjelman ja visualisoinnin yhteydessä varmisti järjestelmän osien yhteensopivuuden keskenään. Prosessin aikana tietokantaan tehtiin useita muutoksia, joista viimeisin malli esitellään seuraavassa luvussa.



Kuva 10. OEE-järjestelmän tietokantarakenne.

OEE-tietokanta muodostettiin neljästä taulukosta kuvan 10 mukaisesti. Päätaulukkona toimii Solu(Cell) -taulukko, joka kertoo tuotantolinjan OEE-arvot. Solu-taulukon pääavai-

met CellId ja DateId toimivat kaikkien muiden taulukoiden vierasavaimina. Tuote (Product) -taulukko sisältää hyvien ja hylättyjen tuotteiden kappalemäärät. Asema (Station) -taulukko sisältää OEE-arvojen laskemiseen käytettävät aikayksiköt, kuten tuotantoajan, häiriöajan ja kalibrointiajan. Aikajana (Timeline) -taulukkoa käytetään aikajanan luomiseen visualisointiohjelmissa. Aikajana on tuotantoaseman päivän aikana tehtyjä toimintoja kuvaava linja, joka alkaa työpäivän alusta ja päättyy työpäivän loppuun. Aikajanan visualisoi häiriötilojen aikavälit punaisella, seisokkiajat keltaisella ja tuotantoajat vihreällä.

Tietokannan muuttujien datatyypit valittiin vastaamaan lähdedatan formaattia, jolloin vältyttiin formaattimuutoksilta C#-ohjelman toiminnoissa. Esimerkiksi kuvan 10 Station-taulukossa olevat aikamuuttujat täsmäitiin lähdetietojen aikayksikköön. Datatyyppejä testattaessa varmistettiin tyyppien soveltuvuus Excelin ja web-ohjelmien käyttöön. Cell-taulukossa sijaitsevat OEE-prosenttiarvot tallennettiin kokonaisluvuiksi helpomman käsittelyn vuoksi. Tietokannan muodostamista varten tarvittava SQL-koodi löytyy raportin lopusta liitteestä 1.

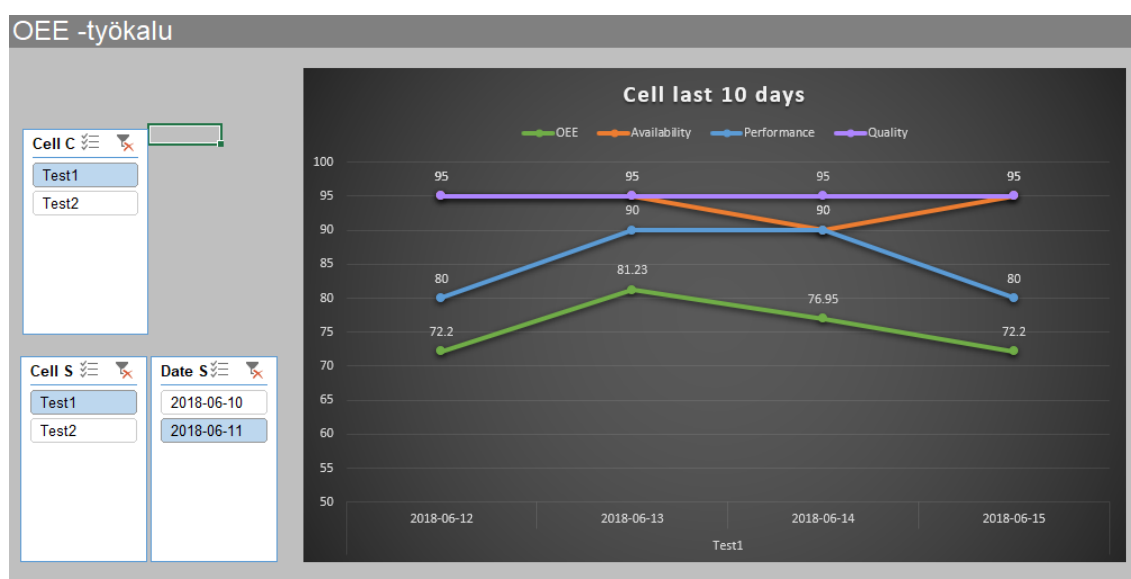
### 3.5.3 Tietokantaliitännät

OEE-tietokannan sijaitseminen Vaisalan lähiverkon serverillä mahdollisti minkä tahansa lähiverkossa olevan ohjelman yhdistämisen tietokantaan. Tietokanta ja C#-ohjelma muodostivat yhteyden helposti Microsoft Visual Studion serverituen ansiosta. Visual Studio pystyy yhdistymään tietokantaan suoraan ja luo automaattisesti ohjelman tarvitseman yhteysavaimen. Datan visualisointia varten luotu Excel tukee myös suoraan tietokantaan yhdistämistä. Excelin connections-listaan luotiin kaksi yhteyttä, jotka mahdollistivat datan automaattisen päivytyksen aina Excel-tiedoston aukaisun yhteydessä ja 30 minuutin aikajakson välein. Serverille tehtyjen valmiiden hakujen ansiosta Exceliin ladattiin vain käyttäjälle hyödyllinen data, mikä vähensi turhaa verkkoliikennettä ja kevensi Excelin tehovaatimuksia. C#-ohjelma kommunikoi usean tietokannan kanssa, joiden rakenne ja serverisijainti vaihtelu huomattavasti. Tuotantoasemat tallensivat datan tuotteesta riippuen eri tietokantoihin, mikä vaikeutti uusien asemien lisäämistä OEE-järjestelmään. Tämän haasteen testaamiseksi projektissa keskityttiin kahteen eri tuotantosoluun, joiden

tietokantarakenne ja toiminta eroavat suuresti toisistaan. C#-ohjelman tietokantaliitännöiden määrittäminen XML-tiedostoissa osoittautui hyväksi menetelmäksi tämän ongelman välttämiseen.

### 3.6 Visualisointi

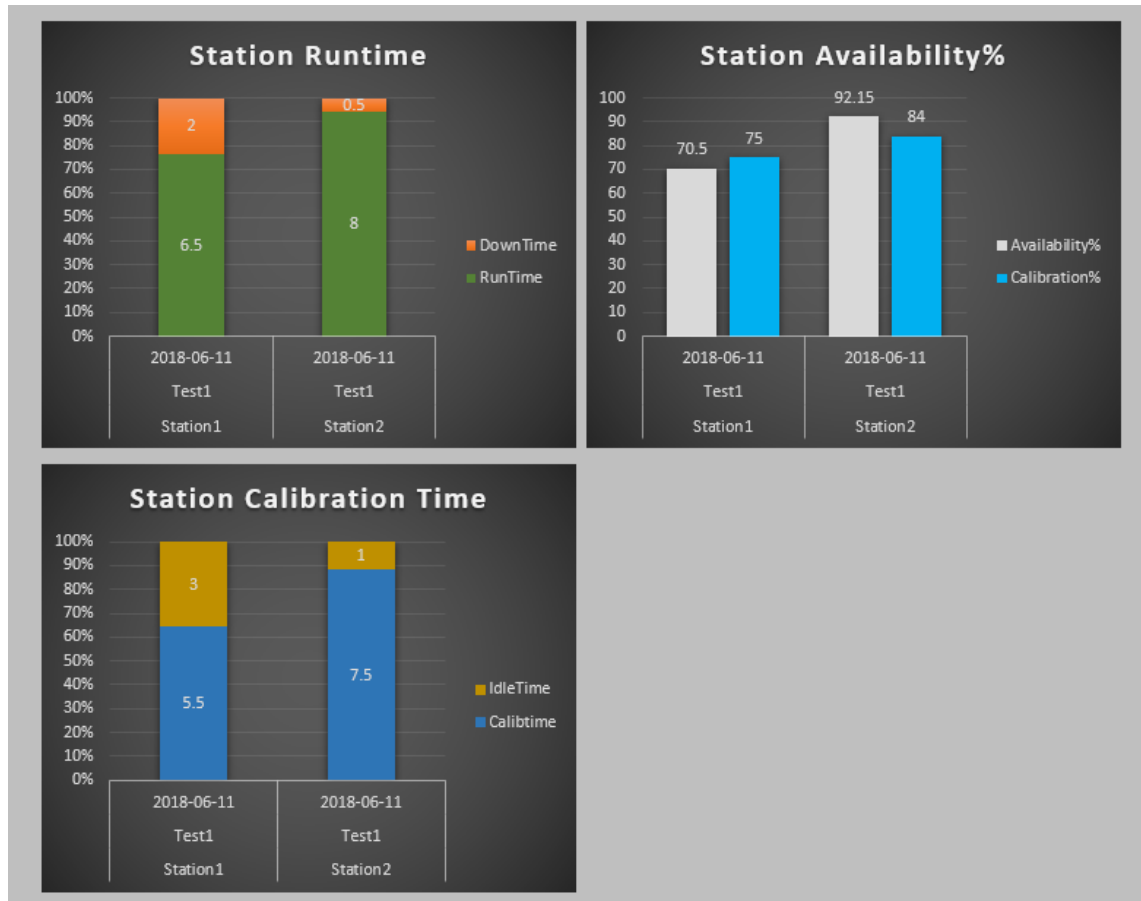
Avoin tietokanta mahdollisti kattavat vaihtoehdot OEE-tiedon visualisoimiseen. Jokainen dataa tarvitseva ohjelma pystyi liittymään tietokantaan ja lukemaan sinne puolen tunnin välein päivitettyä dataa tuotantoasemien tilanteesta. Vaisalan eri osastot käyttävät dataa eri tarkoituksiin, joten myös visualisointiohjelmien tulee olla käyttäjän tarpeisiin sovitettuja. Operaattoreiden käyttöön OEE-data voitiin visualisoida Planeetta- tai Andon-ohjelmien kautta, jolloin OEE-data näkyi operaattoreiden normaalisti käyttämien ohjelmien yhteydessä. Opinnäytetyön aikana visualisointia varten luotiin Excel-ohjelma, joka hakee tietokannasta uusimman tiedon automaattisesti Excel-tiedoston käynnistyksen yhteydessä. Excelissä valitaan tuotantoasema ja aikaväli, jolta OEE-data näytetään. Excelin pääsivu esitellään kuvissa 11 ja 12.



Kuva 11. Excel-ohjelman pääsivun ensimmäinen osa, josta voi valita tuotantoaseman sekä aikavälin OEE-datan esittelyyn.

Kuvassa 11 näkyvän Excel-ohjelman pääsivulla voitiin valita tuotantosolun OEE-arvojen visualisointi joko viimeisen 10 päivän, 12 viikon tai 6 kuukauden ajalta. Sivun vasemman

laidan valikoista valittiin näytettävä tuotantosolu, tuotantosolun asema sekä päivämäärä, josta eteenpäin data visualisoidaan. Valikot mahdollistivat datan helpon rajaamisen käyttäjän tarpeisiin. Rajaaminen on erittäin tarpeellista, kun järjestelmä seuraa kymmeniä tuotantosoluja usean vuoden ajan. OEE-tiedot visualisoitiin valintojen mukaan taulukoksi, jolla voi helposti seurata OEE-arvojen muuttumista halutulla aikavälillä.



Kuva 12. Excel-ohjelman pääsivun toinen osa, joka visualisoi asemien OEE-arvot halutulta päivältä.

Kuvassa 12 näkyy Excel-ohjelman pääsivun toinen osuus, jossa visualisoitiin tarkemmin valitun tuotantosolun asemien käytettävyyteen liittyviä arvoja. Luvussa 3.2 määritettiin käytettävyyden arvioimista parantavat osatekijät. Station runtime ja Station calibration time ilmaisevat aseman käytön tunteina. Runtime vertaa aseman suunnitellun tuotantoajan jakautumista tuotantoaikaan ja hävikkiaikaan. Calibration time taas vertaa aseman

kalibrointiohjelman suunnitellun tuotantoajan jakautumista kalibrointi-aikaan ja seisokki-aikaan. Tuntimäärien avulla voidaan laskea asemien käytettävyy- ja kalibrointi-pro-sentti, jotka näytetään Station Availability% -taulukossa.

12					
13	<b>Row Labels</b>	<b>OEE</b>	<b>Availability</b>	<b>Performance</b>	<b>Quality</b>
14	<b>Test1</b>	<b>460.8</b>	<b>560</b>	<b>520</b>	<b>570</b>
15	2018-06-10	81.23	95	90	95
16	2018-06-11	76.95	90	90	95
17	2018-06-12	72.2	95	80	95
18	2018-06-13	81.23	95	90	95
19	2018-06-14	76.95	90	90	95
20	2018-06-15	72.2	95	80	95
21	<b>Grand Total</b>	<b>460.8</b>	<b>560</b>	<b>520</b>	<b>570</b>
22					

Kuva 13. Excelin tilasto testiaseman OEE-arvoista viikon ajalta.

Excel-ohjelma tarvitsi laajan määrän dataa etusivun taulukoiden muodostamista varten. Tämä data kerättiin kahdelle sivulehdelle tietokannasta SQL-hakujen avulla. SQL-hauilla kerätty data riippui käyttäjän tarpeista, ja datan sisältö määritettiin ennen haun tietokan-nalle lähettämistä tarpeettoman tiedonsiirron vähentämiseksi. Yllä olevassa kuvassa 13 on tietokannasta haettu raakadata kuvan 11 taulukon muodostamista varten. SQL-haun sisältöä säädeltiin tietokannan puolella tehtyjen valmiiden hakupohjien avulla tai Excelin puolella VBA-ohjelmointikielellä tehtyjen makrojen mukaan. VBA (Visual Basic for Appli-cations) on Excelin tukema makrokieli, jolla Excel-tiedostoon ohjelmoitiin lisäominaisuuksia, kuten tarkkoja tietokantahakuja. Makrojen toiminnan aktivointi yhdistettiin painikkeisiin helpon käytettävyyden lisäämiseksi.

Projektin osalta visualisointi suoritettiin ainoastaan Excel-ohjelmalla. OEE-tietokannan rakenteessa ja sijainnissa huomioitiin muiden ohjelmien tarpeet, mutta OEE-tietojen yh-distäminen Vaisalan verkko-ohjelmiin tarvitsi ohjelmien ylläpitäjien osallistumista proses-siin, minkä tekeminen arvioitiin kannattavaksi tehdä omana projektina.

## 4 Yhteenveto

### Rajoitukset

Prototyypimallin vahvuutena on nopeampi ohjelman valmistuminen ja mallidatan käyttöön saaminen, mutta prototyypin yksinkertaistetun rakenteen seurauksena se rajoittaa laajasti ohjelman soveltamista eri tuotantoasemiin. Monet Vaisalan tuotantoasemat ovat ohjelmaltaan uniikkeja ja kommunikoivat tietokantaan dataa eri muodoissa; ohjelmavaiheet sisältävät eri muuttujia ja lähettävät ne eri tietokantoihin. Prototyypin ohjelma soveltuu projektissa valittujen kahden aseman tietojen käsittelyyn. Tietorakenteen yhtenäisyyden puutteen seurauksena uudet asemat tarvitsevat muutoksia ohjelman metodeihin niiden dataa käsittelemään.

Tietokannat ovat jäykkiä rakenteita, ja käyttötarkoitukseen suunnittelun ja käyttöönoton jälkeen niihin on vaikea tehdä muutoksia. Projektin lähdetietokannat eivät sisällä kaikkea tietoa asemien pysähtymisistä ja häiriötilanteista, mikä rajoittaa OEE-ohjelman tarkkuutta; aseman tuotanto-ohjelman kaatumisen pysäyttää tiedonsiirron kantaan ja pysyy huomaamattomana OEE-järjestelmässä. Operaattoreiden työtunnit sisältävä tietokanta on erotettu muista kannoista, mikä estää OEE-järjestelmän pääsyn suunniteltujen työaikataulujen tietoihin. Tämä rajoittaa asemien aikataulutiedot arvioiksi prototyypimallissa.

OEE-tietojen visualisointi määräytyy prototyypin tietokannan rakenteen mukaan. Tietokantarakenne on suunniteltu selkeäksi ja yhteensopivaksi useimpien visualisointityökalujen kanssa, mutta rajoittuu samalla suunnittelussa määritettyihin tietoihin. Monimutkaisempia kaavioita vaadittaessa tietokanta täytyy suunnitella uudelleen. Vaisalan ohjelmaympäristö on jatkuvan muutoksen ja kehityksen alla. Uudet visualisointiohjelmat vaativat OEE-järjestelmän tietokantarakenteen jatkuvaa sopeuttamista.

### Jatkotutkimus

Projektin lopputulos antaa suuntaa OEE-järjestelmän jatkokehitykselle. Intuitiivinen tapa toteuttaa lopullinen järjestelmä olisi sulauttaa asemien testausohjelmaan tuotantotietoja keräävä aliohjelma, joka kommunikoii OEE-ohjelman kanssa ja tarkentaa dataa. Vaisa-

lalla on useita tuotannon tehokkuutta mittaavia järjestelmiä. OEE-järjestelmän integroiminen osaksi muita järjestelmiä mahdollistaa kaiken oleellisen tiedon visualisoinnin saman kanavan kautta. Jatkotutkimus voi keskittyä OEE-järjestelmän implementointiin osaksi testiohjelmaa ja visualisointiohjelmaa. Tämän tutkimuksen pääpaino on hyödyllisen käytettävyydestiedon hankkiminen malliasemilta ja laajempien OEE-projektien esimäärittely.

#### Yhteenveto

Projektin tavoitteena oli suunnitella ja toteuttaa prototyypimalli, joka parhaalla tavalla mittaa ja visualisoi tuotannon tehokkuutta. Prototyypissä päädyttiin OEE-menetelmää käyttävään ohjelmajärjestelmään. Työ aloitettiin laajalla tutkimustyöllä OEE-menetelmän toimintaan, tietojärjestelmien suunnittelun teoriaan sekä Vaisalan prosesseihin, joita kappale 3 selventää. Teorian pohjalta muodostettiin alustava ohjelmarakenne, jota tiimi-palaverien kautta jalostettiin lopulliseen muotoon.

Rakennemallin pohjalta järjestelmä jaettiin kolmeen eri osaan. Ensimmäisessä osassa C#-ohjelma keräsi tuotantoasemien datan Vaisalan tietokannasta SQL-hauilla ja jalosti datan OEE-arvoja kuvaavaan muotoon. Jalostettu data siirrettiin OEE-tietoja varten rakennettuun tietokantaan, joka muotoiltiin yhteensopivaksi useiden eri verkkosovellusten ja visualisointiohjelmien kanssa. OEE-tietojen visualisointiin käytettiin projektissa Excel-ohjelmaa, joka VBA-makroilla automaattisesti haki datan tietokannasta ja muokkasi sen taulukoiksi. Excelin avulla käyttäjät pystyivät seuraamaan määritettyjen tuotteiden ja asemien tehokkuutta aikalinjana.

Insinööriyössä ei päästy tavoitteisiin automaattisesti toimivan ohjelman osalta. C#-ohjelman rakennuksen aikana ilmenneet haasteet tietokantojen yhteyksien välillä ja ohjelmamethodien suunnittelussa hidastivat etenemistä. Prototyypistä saatiin valmiiksi rakennekaava, tietokanta sekä visualisointi. Lisäksi projektin aikana tehty esitutkimus ja ohjelman rakentaminen toi hyvin esille tärkeitä seikkoja lopullisen ohjelman toimintakuntoon saattamiseksi. Insinööri työ toimii ohjeistuksena OEE-järjestelmän luomisessa ja jatkokehityksen suunnittelussa.

## Lähteet

- 1 Tang, Tiantian. 2017. Developing an OEE prototype. Lahti University of Applied Sciences. Verkkoaineisto. <[http://www.theseus.fi/bitstream/handle/10024/131182/Tang\\_Tiantian.pdf?sequence=2&isAllowed=y](http://www.theseus.fi/bitstream/handle/10024/131182/Tang_Tiantian.pdf?sequence=2&isAllowed=y)>. Luettu 20.11.2018.
- 2 Pohjalainen, Henri. 2015. Virtual Operations Center for Smart Maintenance. Aalto University. Diplomityö. Verkkoaineisto. <[https://aaltodoc.aalto.fi/bitstream/handle/123456789/16836/master\\_Pohjalainen\\_Henri\\_2015.pdf?sequence=1&isAllowed=y](https://aaltodoc.aalto.fi/bitstream/handle/123456789/16836/master_Pohjalainen_Henri_2015.pdf?sequence=1&isAllowed=y)>. Luettu 15.9.2018.
- 3 Annunziata, Marco and Peter C. Evans. 2013. The Industrial Internet@Work. Verkkoaineisto. <[https://www.ge.com/sites/default/files/GE\\_IndustrialInternetatWork\\_WhitePaper\\_20131028.pdf](https://www.ge.com/sites/default/files/GE_IndustrialInternetatWork_WhitePaper_20131028.pdf)>. Luettu 9.8.2018.
- 4 Williamson, Bob. 2015. Uptime: Remembering the father of TPM. Verkkoaineisto. <<https://www.efficientplantmag.com/2015/06/remembering-the-father-of-tpm/>>. Luettu 1.9.2018.
- 5 Vorne Industries Inc. 2018. What is Overall Equipment Effectiveness. Verkkolähde. <<https://www.oeo.com/>>. Luettu 4.9.2018.
- 6 Vorne Industries Inc. 2018. Six Big Losses. Verkkoaineisto. <<https://www.oeo.com/oeo-six-big-losses.html>>. Luettu 10.10.2018.
- 7 Modic, Niklas ja Åhlström, Pär. 2013. Tätä on Lean. Ratkaisu tehokkuusparadoksiin. Rheologica Publishing.
- 8 Vorne Industries Inc. 2016. LeanProduction. OEE (Overall Equipment Effectiveness). Verkkoaineisto. <<https://www.leanproduction.com/oeo.html>>. Luettu 10.10.2018.
- 9 Niskanen, Harri. 2011. Kokonaistehokkuus PK-yrityksissä. Savonia-ammattikorkeakoulu. Verkkoaineisto. <[https://www.theseus.fi/bitstream/handle/10024/38574/Niskanen\\_Harri.pdf?sequence=1&isAllowed=y](https://www.theseus.fi/bitstream/handle/10024/38574/Niskanen_Harri.pdf?sequence=1&isAllowed=y)>. Luettu 16.9.2018.
- 10 Vaisala tiedote 27.9.2018. Verkkoaineisto. Vaisala Oyj. <<https://www.vaisala.com/fi/lehdistotiedotteet/2018-09/vaisalan-uuden-tuotekehityskeskukseen-rakennustyot-alkavat-yhdysvalloissa>>. Luettu 28.9.2018.



- 11 Helm, James. 2017. Methods of Software Prototyping. University of Houston-Clear Lake. Verkkoaineisto. <[http://sce.uhcl.edu/helm/REQ\\_ENG\\_WEB/My-Files/mod4/Software\\_Prototyping.pdf](http://sce.uhcl.edu/helm/REQ_ENG_WEB/My-Files/mod4/Software_Prototyping.pdf)>. Luettu 15.9.2018.
- 12 Mustonen, Veli-Matti. 2015. Prototyypit ja vaatimusmäärittely sovelluskehityksessä. Hämeen Ammattikorkeakoulu. Opinnäytetyö. Verkkoaineisto. <[https://www.theseus.fi/bitstream/handle/10024/87766/Mustonen\\_Veli-Matti.pdf?sequence=1](https://www.theseus.fi/bitstream/handle/10024/87766/Mustonen_Veli-Matti.pdf?sequence=1)>. Luettu 20.10.2018.
- 13 Software Prototyping. 2018. Verkkoaineisto. Wikipedia. <[https://en.wikipedia.org/wiki/Software\\_prototyping](https://en.wikipedia.org/wiki/Software_prototyping)>. Luettu 20.10.2018
- 14 S.P. Overmyer, 1991. Revolutionary vs Evolutionary Rapid Prototyping: Balancing Software Productivity and HCI Design Concerns. George Mason University. Verkkoaineisto. <[http://www.academia.edu/328275/Revolutionary\\_Vs.\\_Evolutionary\\_Rapid\\_Prototyping\\_Balancing\\_Software\\_Productivity\\_and\\_Hci\\_Design\\_Concerns](http://www.academia.edu/328275/Revolutionary_Vs._Evolutionary_Rapid_Prototyping_Balancing_Software_Productivity_and_Hci_Design_Concerns)> Luettu 21.11.2018.
- 15 Bryman, A. 2006 Integrating Quantitative and Qualitative Research: How is it done? Qualitative Research Vol: 6.
- 16 Töttö, P. (2004) Syvällistä ja Pinnallista: Teoria, Empiria ja Kausaalisuus sosiaalitutkimuksessa Tampere: Vastapaino
- 17 Davis, Alan M. 1995 Software Prototyping. University of Colorado at Colorado Springs. Insinöörityö. Verkkoaineisto. <[https://books.google.fi/books?hl=fi&lr=&id=DPVwyHS86IkC&oi=fnd&pg=PA39&dq=software+prototyping+&ots=V6LojBk0K\\_&sig=a\\_wwhrtl4DMT\\_UKxble-QwbP6bdE&redir\\_esc=y#v=onepage&q&f=false](https://books.google.fi/books?hl=fi&lr=&id=DPVwyHS86IkC&oi=fnd&pg=PA39&dq=software+prototyping+&ots=V6LojBk0K_&sig=a_wwhrtl4DMT_UKxble-QwbP6bdE&redir_esc=y#v=onepage&q&f=false)>. Luettu 20.10.2018.

**OEE-tietokannan SQL-koodi**

```

DROP TABLE [dbo].[Events];
GO

DROP TABLE [dbo].[Stations];
GO

DROP TABLE [dbo].[Products];
GO

DROP TABLE [dbo].[Cells];
GO

--***** [dbo].[Cells]

CREATE TABLE [dbo].[Cells]
(
  [CellId]          NVARCHAR(30) NOT NULL ,
  [DateId]          DATE NOT NULL ,
  [OEE]              DECIMAL(5,2) NULL ,
  [Availability]    DECIMAL(5,2) NULL ,
  [Performance]    DECIMAL(5,2) NULL ,
  [Quality]         DECIMAL(5,2) NULL ,
  [Logging]         DECIMAL(5,2) NULL ,

  CONSTRAINT [PK_Cells] PRIMARY KEY CLUSTERED ([CellId] ASC, [DateId] ASC)
);
GO

--***** [dbo].[Stations]

CREATE TABLE [dbo].[Stations]
(
  [StationId]       NVARCHAR(30) NOT NULL ,
  [CellId]          NVARCHAR(30) NOT NULL ,
  [DateId]          DATE NOT NULL ,
  [RunTime]         DECIMAL(5,3) NULL ,
  [DownTime]       DECIMAL(5,3) NULL ,
  [CalibTime]      DECIMAL(5,3) NULL ,
  [IdleTime]       DECIMAL(5,3) NULL ,
  [PlannedProdTime] DECIMAL(5,3) NULL ,
  [FirstCalibStart] DATETIME NULL ,
  [LastCalibEnd]   DATETIME NULL ,
  [StationAvailab] DECIMAL(5,2) NULL ,
  [StationCalib]   DECIMAL(5,2) NULL ,

  CONSTRAINT [PK_Stations] PRIMARY KEY CLUSTERED ([StationId] ASC, [CellId]
ASC, [DateId] ASC),
  CONSTRAINT [FK_Stations_Cells] FOREIGN KEY ([CellId], [DateId])
REFERENCES [dbo].[Cells] ([CellId], [DateId])
);
GO

--***** [dbo].[Products]

CREATE TABLE [dbo].[Products]
(
  [ProductId] NVARCHAR(30) NOT NULL ,
  [CellId]    NVARCHAR(30) NOT NULL ,

```

```
[DateId]    DATE NOT NULL ,
[OKpcs]     INT NULL ,
[Failpcs]   INT NULL ,

CONSTRAINT [PK_Products] PRIMARY KEY CLUSTERED ([ProductId] ASC, [CellId]
ASC, [DateId] ASC),
CONSTRAINT [FK_Products_Cells] FOREIGN KEY ([CellId], [DateId])
REFERENCES [dbo].[Cells]([CellId], [DateId])
);
GO

--***** [dbo].[Events]

CREATE TABLE [dbo].[Events]
(
    [EventId]    INT NOT NULL ,
    [StationId] NVARCHAR(30) NOT NULL ,
    [CellId]     NVARCHAR(30) NOT NULL ,
    [DateId]     DATE NOT NULL ,
    [GraphLine] NVARCHAR(15) NOT NULL ,
    [EventType]  INT NOT NULL ,
    [StartTime]  DATETIME NULL ,
    [EndTime]    DATETIME NULL ,

    CONSTRAINT [PK_Events] PRIMARY KEY CLUSTERED ([EventId] ASC, [StationId] ASC,
[CellId] ASC, [DateId] ASC),
    CONSTRAINT [FK_Events_Stations] FOREIGN KEY ([StationId], [CellId], [DateId])
REFERENCES [dbo].[Stations]([StationId], [CellId], [DateId])
);
GO
```

## Haastattelurakenne OEE:n hyödyllisyyden määrittämiseksi.

Haastattelun kesto: Noin 40 minuuttia.

Haastattelun kysymykset:

- Mitä mieltä olette projektissa käytettävästä tuotannon mittausmenetelmästä?
- Onko teillä käyttöä OEE-järjestelmän tuloksille?
- Kuinka projekti voisi hyödyttää teidän osastoa?
- Mikä visualisointitapa toisi OEE-datan parhaiten teidän käyttöön?
- Olisiko järkevää yhdistää OEE-data teillä jo käytössä olevaan järjestelmään?
- Miltä tuotantoasemilta on tärkeintä kerätä OEE-data?

Asiantuntijoiden tietämykseen muokatut kysymykset:

- Mitkä ovat ohjelman/tietokannan rajapinnat?
- Kuinka kyseinen tuotantoasema toimii?
- Mitä ohjelmia asemalla toimii tällä hetkellä?
- Onko kannattavaa integroida OEE-järjestelmä aseman ohjelmaan vai tehdä erillinen ohjelma?
- Ketkä tietävät aiheesta lisää?