



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Jarno Mertanen

Automatisoitu valaisinluettelo MagiCAD-sovellukselle

Metropolia Ammattikorkeakoulu

Insinööri (YAMK)

Talotekniikan tutkinto-ohjelma

Insinöörityö

14.5.2020

Tekijä Otsikko	Jarno Mertanen Automatisoitu valaisinluettelo MagiCAD-sovellukselle
Sivumäärä Aika	62 sivua + 4 liitettä 14.5.2020
Tutkinto	Insinööri (YAMK)
Tutkinto-ohjelma	talotekniikka
Ammatillinen pääaine	talotekniikka
Ohjaajat	projektipäällikkö Roni Leppänen suunnittelupäällikkö Mika Lamminen opinto-ohjaaja Jarmo Tapio
<p>Tämä opinnäytetyö keskittyi automatisoidun VBA-pohjaisen Excel-taulukkolaskentasovelluksessa käytettävän valaisinluettelopohjan kehittämiseen ja käyttöönottoon talotekniikan sähkö-, automaatio- ja telesuunnittelua tarjoavassa yrityksessä.</p> <p>Edellisen taulukon suurimpana heikkoutena pidettiin sen täysin manuaalista käytettävyyttä. Projektikohtaisten tietojen syöttö sekä mahdolliset muotoilut ja asetelut tehtiin taulukkoon käsin. Ylläpito oli myöhemmissä työvaiheissa työlästä ja virheiden mahdollisuus suureni valaisintietokannan kasvaessa.</p> <p>Uuden työkalun päämääränä oli virtaviivaistaa muutosaineiston laadintaa sekä rajata virhemahdollisuuksia ja vaikuttaa toimintatapoihin laadunhallinnallisesti. Tuotteella pyrittiin kehittämään edeltäjänsä toiminnallisuutta ja käyttöä automatisoimalla joko täysin tai osittain kaikki projektien työvaiheissa tarvittavat toimenpiteet ja prosessit. Uuden luettelon toiminnallisuus rakennettiin olemassa olevan pohjan tavoin toimiston käytössä olevaan MagiCAD-sovelluksen ja tietokantapohjaisesti tuotettujen raporttien ympärille.</p> <p>Kattavan testauksen jälkeen ja myönteisen käyttäjäpalautteen ansiosta toteutettu sovellus on tällä hetkellä käytössä kahdessa suuremmassa suunnitteluprojektissa. Tämän opinnäytetyön ansiosta automatisoitua prosessia halutaan soveltaa myös muihin luetteloihin, kuten massaluettelon automatisointiin.</p>	
Avainsanat	Excel, VBA, Visual Basic, AutoCAD, MagiCAD, Automaatio

Author Title	Jarno Mertanen Automated Light Catalogue for MagiCAD
Number of Pages Date	62 pages + 4 appendices 14 May 2020
Degree	Master of Engineering
Degree Programme	Building Services Engineering
Professional Major	Building Services Engineering
Instructors	Roni Leppänen, Project Manager Mika Lamminen, Design Manager Jarmo Tapio, Principal Lecturer
<p>The goal of the final year was to create a fully automated VBA-based lamp catalogue on MS Excel for a company that provides electrical, automation and telecommunication designs for the construction industry. The catalogue was meant to replace a catalogue that had to be used manually which creates a risk for progressive human error as the quantity of the lamps during a project increased.</p> <p>The new tool was meant to improve the general quality control by streamlining the revision process as well as to reduce the risk of errors. The product aimed at enhancing the functionality and usage of the previous catalogue by partial or complete automation of project related tasks. The company's current MagiCAD design software and its database driven report methods served as a base for new features.</p> <p>After throughout testing and positive feedback from colleagues, the new catalogue is now used in two construction projects. The company also wishes to develop the automated processes further in other areas such as automated bills of quantities.</p>	
Keywords	Excel, VBA, Visual Basic, AutoCAD, MagiCAD, Automation

Sisällys

Lyhenteet

1	Johdanto	4
2	Suunnittelumenetelmät ja kehitystaustat	8
2.1	MagiCAD-suunnitteluohjelma	8
2.1.1	Valaisintietokanta	8
2.1.2	Raportointityökalu	13
2.2	Nykyinen luettelo	16
2.3	VBA-ohjelmointikieli	18
2.3.1	Valintaperusteet	18
2.3.2	Heikkoudet	20
2.3.3	Visual Basic Editor	25
2.3.4	Lähteitä ohjelmointiin	30
3	Tuotekehitys ja testausprosessi	31
3.1	Toimintatapamuutokset	31
3.2	Luettelon rakennemuutokset	33
3.3	Ulkoasu ja visualisointi	37
3.4	Automatisoinnin toteutus	40
3.4.1	Kehittämismenetelmän valinta	40
3.4.2	Sovelluksen tarvekartoitus	41
3.4.3	Tuotteen kehitysvaiheet	43
3.5	Ohjelmointiosuus	53
3.5.1	Muuttujien määrittelyt	53
3.5.2	Päämoduulit	55
3.5.3	Lisämoduulit	56
3.6	Tuotekehityksen tulokset	56
4	Yhteenveto	59
	Lähteet	61

Liitteet

Liite 1. Esimerkkikoodeja päämoduulien toteutuksista

Liite 2. Esimerkkikoodeja lisämoduulien toteutuksista

Liite 3. Valaisinluettelon rakenne ja ulkoasu

Liite 4. Valaisinluettelon käyttöohjeet

Lyhenteet

API	Application Programming Interface. Valmis kehitystyökalukokoelma.
AutoCAD	Autodeskin julkaisema CAD-pohjainen suunnitteluohjelma.
Attribuutti	Symbolikohtainen ominaisuus tai tieto.
Demo	Esitysversio tietokonepohjaisesti sovelluksesta tai ohjelmasta.
Iterointi	Samana työvaiheen toistaminen tuloksien saavuttamiseksi.
Excel	Microsoftin julkaisema Office-tuoteperheen taulukkolaskentaohjelma.
Finlex	Oikeusministeriön omistama lainsäädännön tietopankki.
MagiCAD	Talotekniikan suunnittelijoille suunnattu AutoCAD-lisäosa.
Makro	Isäntäohjelman sisäisiin toimintoihin perustuva ohjelmointikieli.
Revisio	Dokumentoitu muutos työpiirustusaineistoon.
Symboli	Suunnitteluohjelmaan tallennettu valmis vektorikuvio.
Teams	Microsoftin julkaisema yhteydenpito- ja hankehallintasovellus.
VB	Visual Basic. Microsoftin kehittämä ja ylläpitämä ohjelmointikieli.
VBA	Visual Basic for Application. Office-tuoteperheen käyttämä ohjelmointikieli.
.dwg	AutoCAD-sovelluksen käyttämä tiedostoformaatti.
.NET	Microsoftin kehittämä ohjelmistopaketti ja -komponenttikirjasto.
.xlsm	Excel-sovelluksen tukema tiedostotyyppi, joka sisältää makroja.

1 Johdanto

Työn taustoja

Tämän työn päämääränä on luoda uusi VBA-ohjelmointikieleen perustuva automatisoitu Excel-pohjainen valaisinluettelo. Opinnäytetyö laaditaan osana talotekniikan insinööritutkintoa (YAMK) ja kehitetty sovellus tuotetaan Rejlers Finland Oy:lle. Työn toimipisteenä on entinen Lausamo Oy:n toimistotila. Yritys sulautui Rejlers-konsernin rakennustoimialaan vuoden 2015 alussa ja tarjoaa talotekniikan sähkö-, automaatio- ja telesuunnittelua uudisrakennuskohteisiin sekä kiinteistöjen korjausrakentamiseen. Automatisoitu valaisinluettelo tuotteena tulee olemaan osana sähkötekniistä suunnitteluyksikköä, mutta kehitystyöllä on myös muita tavoitteita.

Valaisinluettelo on yksi merkittävimmistä hankinta-asiakirjoista, joka on osana talotekniikan sähkösuunnittelua. Suurin ongelma tämänhetkessä luettelomallissa on käsin tehtävän työn määrä sekä siitä johtuva virheherkkyys. Toimiston käytössä oleva MagiCAD-suunnitteluohjelmisto mahdollistaa raporttien luomisen projektiin tallennetuista valaisinsymboleista. Tarpeellinen informaatio siirretään manuaalisesti nykyiseen luetteloon ja toimenpide suoritetaan jokaisen projektin alkuvaiheessa. Lisäksi joissain tapauksissa sekä projektikohtaisia tietoja että taulukon muotoilua on muokattava. Vaikka suunnitteluohjelmasta voidaan ottaa uusi raportti päivitettyillä tiedoilla, on projektin aikana itse valaisinluettelo ylläpidettävä manuaalisesti. Etenkin suuremmissa asiakasprojekteissa, joissa valaisintietokannat ovat laajoja, ei tämä ole ainoastaan raskas, mutta myös virheherkkä toimintatapa.

Toimipisteen käytössä ei tällä hetkellä ole ulkoisia automatisoituja suunnitteluprosesseja, mutta suunnitteluohjelmiston ominaisuudet ja työkalut ovat työntekijöiden tietoisuudessa ja kattavasti hyödynnettynä. Myöskään standardiin pohjautuvaa laadunhallintajärjestelmää ei ole toistaiseksi integroitu toimiston suunnitteluprosessia.

Näin ollen valaisinluettelon sisällöstä sekä ulkoasusta päättävät ensisijaisesti projektista vastuussa olevat henkilöt, minkä vuoksi tasalaatuisen ja tuotteistetun suunnittelun tuottama valaisinluettelo on vaikeasti löydettävissä. Mikäli henkilökohtaisia muokkauksia on

tehty, on useissa tapauksissa pohjana käytetty edellisen projektin asiakirjaa. Tämä tuottaa usein tarpeettomia ongelmatilanteita etenkin henkilöresurssien muutoksissa projektin työvaiheiden aikana.

Työn rajaus

Uusi automatisointiin perustuva valaisinluettelo tulee korvaamaan tämänhetkisen luettelopohjan, jota käytetään manuaalisesti, ja se tulee palvelemaan kaikissa projektin työvaiheissa luovutusaineistoon saakka. Kehitettävä tuote kattaa vain valaisintietokannan ylläpitoon tarvittavan asiakirjan, vaikka kehitystyön esilletuomisen jälkeen on ehdotettu ja keskusteltu myös muiden dokumenttien, kuten piirustus- ja muutosluettelon kehittämistä automatisoiduilla ratkaisuilla. Rajaus on esitelty toimiston esimiehille sekä työstä vastaavalle projektipäällikölle, joiden toimesta tuotteen toiminta-alueeksi on valittu valaisimet. Ohjelmointivirheitä pyritään poistamaan jo kehitysvaiheessa, jotta lopputuotteesta tulee mahdollisimman luotettava. Vastuuhenkilöiden hyväksymänä integroidaan myös käyttäjäpalautteen myötä saatuja kehitysideoita taulukkoon.

Työ keskittyy ensisijaisesti uuden luettelon automatisoimiseen. Tämän lisäksi luettelon yleisulkoasua muokataan loogisemmaksi sekä oletustietoja tullaan päivittämään nykyaisilla lähtöarvoilla. Nykyinen pohja sisältää muun muassa valonlähde- ja liitäntälaitetietoja, jotka ovat käytössä nykyprojekteissa vain poikkeustilanteissa. Valaistusstandardin mukaisen valaistuksen suunnittelu ei ole osana tätä toimeksiantoa, mutta työssä tullaan käsittelemään suunnitteluohjelman valaisintietokannan sekä uuden valaisuluettelon välistä yhteistoimintaa.

Kattavampi sovelluskehittäminen sekä ohjelmointikielten käsittely on poissuljettu työn aihepiiristä. Myöskään tietoturvallisuuteen liittyvät riskit eivät ole osana projektia, mutta VBA-ohjelmointikielen heikkoudet sekä haavoittuvuus käsitellään pintapuoleisesti. Opin näytetyön pääpainona on käytännöllinen tuotekehitys, eikä se hankinta-aineistoon pohjautuen keskity tutkimuksellisen tiedon käsittelyyn. Tästä huolimatta hankinnan kannalta tärkeimmät valaistiedot käsitellään työssä, sillä ne ovat keskeisessä osassa tuotekehitystä.

Työn laatija on ohjelmointikielen perusosaaja. Täten opinnäytetyöraportin pääpainona on VBA-ohjelmointikielen sisäistäminen sekä koodisisällön optimointi. Ohjelmointiratkaisut tuodaan esille poikkeuksellisen laajasti ja yksityiskohtaisesti, jotta lopputulos kykenee palvelemaan parhaiten samankaltaisia kehityshankkeita ja vastaavia osaamistasoja. Käyttöönottovaiheen jälkeisen palautteen myötä pyritään tuotteen jatkokehitysmahdollisuuksia kartoittamaan laajempaan kokonaisuuteen. Mukaan lukien luettelopohjalle laaditut käyttöohjeet, käsittelee työ myös sovelluksen integroimista suunnittelumenetelmiin. Varsinainen testausprosessi sekä yleinen ylläpito on rajattu aihealueen ulkopuolelle tuotekehityksen laajuuden vuoksi.

Työn tavoitteet

Uuden luettelon keskeisimpänä aiheena on paranneltu ylläpito sekä tuottavuuden että tasalaatuisuuden edistys. Asiakasprojektin suunnitteluprosessi on useamman työvaiheen muodostama kokonaisuus luonnoksista luovutuspiirustuksiin. Näiden vaiheiden välillä muutoksien laatiminen on pääasiassa suurin työtaakka suunnittelijoille ja tällöin taulukko altistuu myös huolimattomuusvirheille. Vaikka ylläpidon aikana itse taulukkoon tehtävät muutokset eivät välttämättä ole huomattavia, ovat ne systemaattisia ja useimmiten niitä on laadittava viikoittain. Tämä antaa selvän perustan ja tarpeen automatisoinnille. Täten luettelopohjan toiminnallisuutta lähdettiin kehittämään automatisoimalla muutoksien laatiminen ja myöhemmin lisäämällä muita tarpeellisia toimintoja, kuten käyttöönotto.

Edellä mainittu ongelma on erityisen merkityksellinen laajoissa asiakasprojekteissa, mutta uuden luettelon tehtävänä on palvella myös muita hankkeita niiden kokoluokasta riippumatta. Uuden automatisoidun valaisinluettelon toimintaperiaate tulee pohjautumaan nykyisiin suunnittelumenetelmiin ja sen käyttöönotto on oltava yhtä vaivatonta kuin edeltäjänsä. Uuden toiminnallisuuden myötä rajoitetaan käyttäjien manuaalisia muokausmahdollisuuksia huomattavasti ja taulukkosovelluksen on siten kyettävä suorittamaan tarpeelliset toiminnot yhdellä painonapilla taulukon solujen ollessa lukittuna. Suunnitteluprosessit sekä toimintamallit kehittyvät ja muuttuvat jatkuvasti. Lisäksi henkilöstö sekä heidän työtehtävänsä vaihtelevat toistuvasti. Siten on tärkeää, että uuden luettelopohjan ohjeet ovat käytännönläheisiä ja niiden käyttöönotto on mutkatonta.

Automatisoidun muutoksen laatimisen lisäksi integroidaan taulukkoon myös muita suunnittelua edesauttavia ominaisuuksia. Taulukko muun muassa ilmoittaa sen tekemät muutokset ponnahdusikkunassa päivityksen jälkeen sekä ylläpitää erillistä katalogia välilehdessä, josta päivittyneet tiedot voidaan sujuvasti kopioida muutosluetteloon. Kaikki tarvittavat hakemistot haetaan koodin avulla ja siten navigointi oikeaan hakemistoon on joka tilanteessa suoraviivaista. Rajoitetun muokattavuuden vuoksi täydennetään painonappia käyttäjävalinnoilla, joilla voidaan merkittävästi osin vaikuttaa taulukon toiminnallisuuteen. Tämän lisäksi kaikki tarvittavien tietojen syöttäminen taulukkoon tapahtuu helpokäyttöisten käyttäjälomakkeiden avulla. Suunnitteluprosessin tehostaminen ja optimointi ovat tämän hetkessä työympäristössä merkittävässä asemassa ja siten uudesta luettelosta saadut käyttäjäkokemukset palvelevat sekä nykyisten että tulevien menetelmien kehittämisessä ja kartoittamisessa.

Kehityshanke käynnistetään demoversion esityksellä. Tässä versiossa laaditaan vain muutosaineisto luettelon aiemmasta sisällöstä ja muut kehitystyön aikana tarpeelliseksi havaitut toiminnot lisätään tuotteeseen myöhemmässä vaiheessa. Jokaisen testausvaiheen kattava loppuunsaattaminen on olennaisessa asemassa, kun uusia ominaisuuksia ja käyttäjätoiveita lisätään sovellukseen testiryhmän palautteen myötä.

Viimeistellyn version automatiikan on sopeuduttava lähtötietojen eroavaisuuksiin sekä mahdollisten lisäominaisuuksien on toimittava luotettavasti. Kaikki uudet ominaisuudet tullaan testaamaan laajasti lisäämisvaiheessa, jotta tarpeettomilta ongelmatilanteilta vältyttäisiin. Työn kommunikaatioalustana toimii Microsoftin julkaisema Teams-sovellus, jonka kautta käyttöongelmat ja kehitysideat käsitellään. Sekä suljettu että julkinen testaaminen kestävät puoli vuotta ja esityksissä tuodaan esille taulukon uusimmat ominaisuudet. Työn lopussa tarkastellaan palautteen pohjalta luettelopohjan puutteita ja jatkokehitystarpeita. Lisäksi tavoitteena on arvioida automatisoitujen, yritykselle räätälöityjen ratkaisujen hyödyllisyyttä tulevaisuudessa.

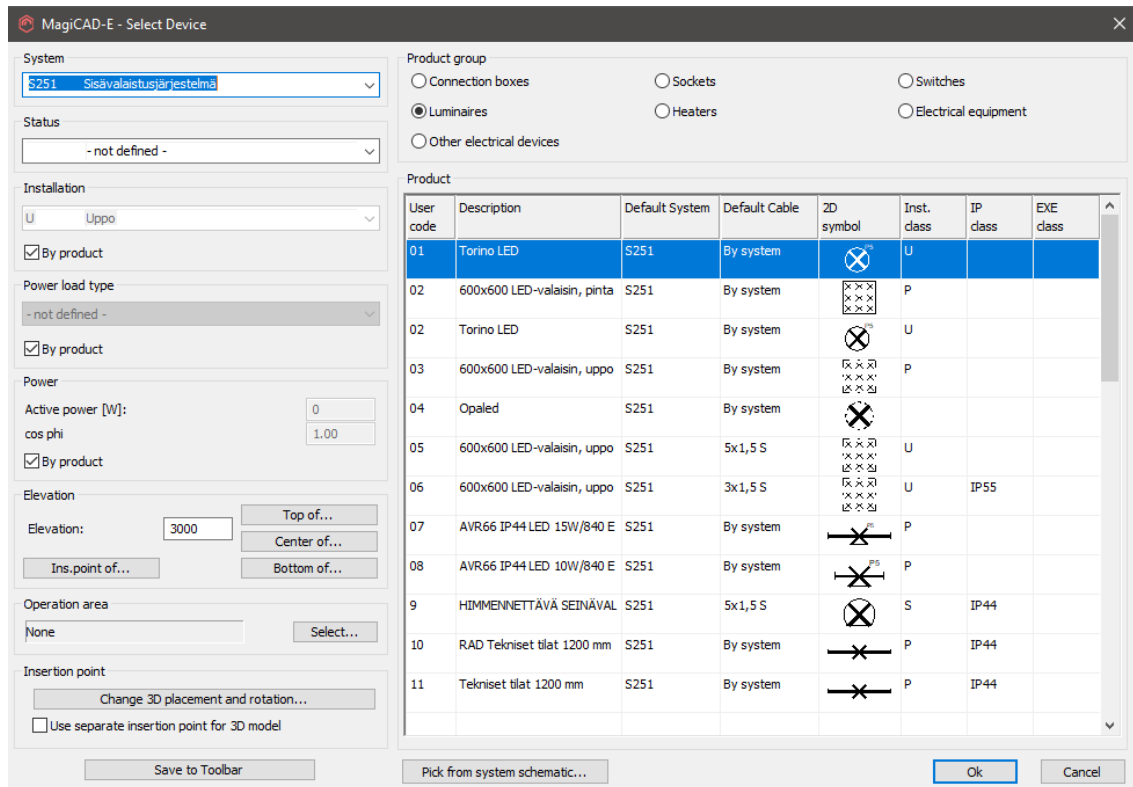
2 Suunnittelumenetelmät ja kehitystaustat

2.1 MagiCAD-suunnitteluohjelma

2.1.1 Valaisintietokanta

MagiCAD on Glodon-yhtiön julkaisema sovellus ja se toimii lisäosana Autodesk:n ylläpitämälle AutoCAD-ohjelmalle. Molemmat ovat laajasti käytössä alan toimijoiden keskuudessa ja tarjoavat kattavat ominaisuudet taloteknisten järjestelmien kaksi- tai moniulotteiseen suunnitteluun. MagiCAD-lisäosa tuo olennaisia apuvälineitä tietokantapohjaiseen suunnitteluun ja niihin pohjautuvat työkalut toimivat keskeisesti sekä nykyisen että uudistuvan valaisinluettelon pääosana. Valaisinsymbolit ja niihin tallennetut attribuutit toimivat lähtötietoina taulukolle sekä pohjana sen toiminnallisuudelle. Valaistus ja valostandardin SFS-EN 12464-1 mukainen suunnittelutyö on määrälaskennallisesti merkittävässä asemassa, mutta suunnitelmien määräystenmukaisuuden todentamiseen käytettävä DIALux-sovellus perustuu täysin erityyppiseen tietokantaan. Täten edellä mainittu sovellus sekä sen osuus suunnitteluprosessissa on rajattu ulos opinnäytetyöraportin aihealueesta, sillä sen merkityksellisyys luettelopohjan tuotekehityksen kannalta on mitätön. Kaikki luettelon kannalta keskeiset tiedot saadaan symboleihin tallennetuista attribuuteista ja valaistuslaskennallisesti keskeiset ominaisuudet, kuten luminanssikaumat ja valaistusvoimakkuudet eivät ole osana taulukkoa.

Suunnitteluohjelmisto käyttää piirtotekniikassa vektoreita sekä niistä koostuvia symboleita. AutoCAD-sovellus antaa automaattisesti näille tiettyjä ominaisuuksia niiden muotoon ja tyyppiin perustuen. Nämä tiedot tallennetaan vektorin attribuutteihin ja siten esimerkiksi ympyrän säde tai linjan pituus saadaan helposti ohjelmasta suunnitteluvaatimuksien mukaisesti. MagiCAD-lisäosa laajentaa näitä edellytyksiä ja tarjoaa kattavamman tietokannan sekä siihen pohjautuvat työkalut taloteknisten dokumenttien laadintaan. Tietoja, kuten valaisimen IP-luokitus, voidaan tallentaa sitä vastaavan symbolin attribuutteihin, mikä ei ole mahdollista AutoCAD-sovelluksessa. Kuvassa 1 on näkymä symboleista koostuvasta MagiCAD-valaisintietokannasta. Valintaikkuna avautuu lisäosan piirto toimintojen alla sijaitsevasta ”laitesymbolit”-alavalikosta tai vaihtoehtoisesti kirjoittamalla ohjelman komentoriville käskyn ”MAGIEDEVICE_ELUM”.



Kuva 1. Tele- ja sähköjärjestelmillä on erilliset tietokannat, joita hallinnoidaan ryhmittäin. Valaisimille on oma kategoria, joka mahdollistaa näille tuotteille tyypillisten ominaisuuksien määrittämisen.

MagiCAD-lisäosa mahdollistaa myös erilaisten kytkimien ja moottorien sekä sähkölaitteiden ja -kalusteiden ylläpidon tietokannan kautta. Symbolien yleisimmät attribuutit noudattavat yhtenäistä logiikkaa valaisimien kanssa riippumatta niiden tyypistä tai kategoriasta, mikä tulevaisuudessa mahdollistaa uuden luettelopohjan hyödyntämisen myös muissa taulukoissa. Valintaikkunan kautta symbolit sijoitetaan tasopiirustuksiin, joista määrät haetaan luettelon käyttöön myöhemmin käsiteltävän raporttityökalun avulla. Myös valaisimen tilaan (Status) voidaan vaikuttaa. Mikäli valaisin valitaan esimerkiksi olemassa olevaksi tai purettavaksi, ei sitä oteta automaattisesti huomioon määrälaskelmissa ilman erillistä määrittelyä. Suunnittelutehtävissä valintaikkuna on aktiivisessa käytössä, mutta valaisinluettelo koostuu pääasiassa symboleihin tallennetuista attribuuteista. Näitä tietoja päästään hallinnoimaan symbolikohtaisesti painamalla hiiren oikeaa painiketta ja valitsemalla ominaisuudet (Properties). Kuvassa 2 on tyypillinen näkymä valaisintiedoista, jonne tuotekohtaiset arvot lisätään.

Variable	Value
National code:	FI
Hyperlink:	
IfcExportAs:	IfcLightFixtureType
IfcType:	notdefined (IFC2X3, IFC4)
P1 yleinen:	
P2 val. as.tapa:	K
P3 val. lit.laite:	Dali
P4 yleinen:	

Kuva 2. Olennaisimmat tiedot valaisinluettelon toiminnallisuuden kannalta voidaan lisätä symbolikohtaisesti. Monet eri tyypeihin lajitellut symbolit sisältävät yhteisiä attribuutteja valaisimien kanssa.

Seuraavat attribuutit ovat hyödynnetty uudessa luettelopohjassa, ja monet näistä ovat osana myös nykyistä taulukkoa. Käytössä olevia tietoja pyritään hyödyntämään loogisemmin ja näin ollen myös helpottamaan tietokannan ylläpitoa. Tähän lähestymistapaan päädyttiin ensisijaisesti käyttäjäkokemusten myötä. Paras ratkaisu oli sisältää tulevassa luettelopohjassa mahdollisimman paljon nykyisiä tietoja identtissä järjestyksessä, sillä tämä helpotti huomattavasti uuden taulukon käyttöönottoa. Myös tietoisuus taulukon uudesta sopeutumismahdollisuudesta auttoi päätöksen teossa. Uuden luettelon oletusotsikot koostuvat tiedoista

- positiotunnus (user code)
- valmistaja (manufacturer)
- tuotenumero/-malli (product code)
- liitântäteho (active power)
- asennustapa (installation code)
- IP-luokitus (IP class)

- huomiotunnus (P2 product variable)
- liitäntälaite (P3 product variable)
- lisätiedot (product note)
- hyperlinkki (hyperlink)

ja ne kattavat pääosin kaiken tarvittavan tiedon hankinta- sekä vaihtotoimenpiteitä varten.

Positiotunnuksen tehtävänä on täsmentää valaisimien sijoitukset tasosuunnitelmissa. Täten se täydentää ensisijaisesti muuta suunnitteluaineistoa eikä suoranaisesti palvele valaisinluetteloa sisällöllisesti. Tuotenumerailla ilmaistaan hankintaa vastaava valaisinmalli suunnitelmien toteutusta varten. Tuotetta on käytetty myös määräyksien mukaisten laskelmien laatimiseen, sillä standardi edellyttää, että kaikista mallitiloista on laadittava laskelmat. Tästä johtuen, mikäli malli halutaan vaihtaa vastaavaan tuotteeseen, on sen täytettävä tilakohtaiset vaatimukset. Valmistaja havainnollistaa, kenen tuotevalikoimasta valaisin löytyy. Projektikohtaisesti voidaan myös sopia, että vain tietyn valmistajan valaisimien käyttö on sallittua.

Tarkempien teknillisten ominaisuuksien esittäminen herätti keskustelua henkilöstön keskuudessa. Tuotenumero täsmentää erinomaisesti valaisimen tarkemmat tiedot, mutta toisaalta joidenkin ominaisuuksien esittäminen luettelossa tuo tiettyjä etuja sekä suunnittelijalle että urakoitsijalle. Asennustavan perusteella voidaan huomioida mahdollinen kehyksen tarve mikäli valaisin sijoitetaan pintamateriaaliin tai vaihtoehtoisesti vaijeriripustukseen tarvittavat erikoisvarusteet kytetään ennakoimaan. IP-luokitus nousee usein esille hankintavaiheessa, kun valaisin halutaan vaihtaa eri tuotteeseen. Tuolloin on käytännöllistä nähdä luettelosta ensisilmäyksellä, onko tuotteen koteloitu erikoisvaatimuksia. Liitäntäteho on ehkä kiistanalaisin tieto, sillä LED-teknologia kehittyy varsin nopeasti. Siten pitempiaikaisissa projekteissa ei tämä tieto ole enää ajantasainen. Oletuksena tilalle ehdotettiin muun muassa valon määrää (lumen), joka tuo paremmin esille tekniset tarpeet määräystenmukaisuuden näkökulmasta. Valon määrän taustalla on kuitenkin muita teknisiä ominaisuuksia, kuten valojakauma, jotka vaikuttavat siihen. Täten uudesta pohjasta löytyy edeltäjänsä tavoin liitäntäteho. On mahdollista, että tulevaisuudessa symbolit tukevat attribuuteiltaan tehonkulutusta valon määrän suhteen (W/lm), mikä olisi muihin vaihtoehtoihin suhteutettuna käytännöllisin tieto.

Lisätietojen avulla voidaan tarkentaa valaisimen ominaisuuksia vapaamuotoisesti. On muun muassa tilanteita, joissa tuotevalikoimasta löytyy useita eri vaihtoehtoja samalle valaisimelle, mutta niiden poikkeuksellisia ominaisuuksia ei ole eroteltu tuotekoodissa. Esimerkiksi tapaus, jossa valaisimen väri on määritelty poikkeavaksi perusversiosta, voidaan tuoda esille lisätietojen kautta. Usein myös valaisimen erikoisvarusteet, kuten integroidut liiketunnistimet ja tuotekohtaiset liitännälaitteet kirjataan lisätietoihin. Uuteen luettelopohjaan lisättiin huomiokoodi helpottamaan näiden tietojen ylläpitoa. Koodi toimii tunnuksena toistuville ominaisuuksille, jotka on huomioitava yhdessä tai useammassa valaisimessa. Projekteissa on toistuvasti esillä tilojen erikoisvaatimuksia, joissa tämän tyylinen esitystapa on hyvin hyödynnettävissä. Esimerkiksi liikuntasalien valaisimet on varustettava pallosuojilla, vuokralaistiloissa on osittain käyttäjähankintaan kuuluvia tuotteita sekä eräisiin tiloihin pätee IK- ja ATEX-luokittelun vaatimuksia. Nämä tiedot ovat vaivattomasti esitettävissä tuotekohtaisilla huomiotunnuksilla, kuten ”KH = käyttäjähankinta”.

Hyperlinkit kuuluvat tulostusalueen ulkopuoleisiin tietoihin, jotka palvelevat vain suunnittelijoita. Niillä helpotetaan muun muassa valojakauma-tiedostojen lataamista valmistajan verkkosivuilta suunnittelua varten. Lisäksi valaisimien johdonsuojakohtaiset maksimimäärät tai optiikkavaihtoehdot suunnittelutyön helpottamiseksi löytyvät kätevästi hyperlinkin kautta. Taulukon sopeutuvuuden myötä on valaisinluettelon lopullinen sisältö täysin suunnittelijoiden ja projektien vastuuhenkilöiden päätettävissä. Useilla kollegoilla on eri näkemys valaisinluettelon sisällön tarpeista, ja uusi luettelo mahdollistaa joustavan käyttöönoton kaikissa tilanteissa. Osa luettelon tiedoista on päätetty kuitenkin pitää pysyvästi osana sen sisältöä, sillä ne ovat oleellinen osa hankinta-asiakirjaa.

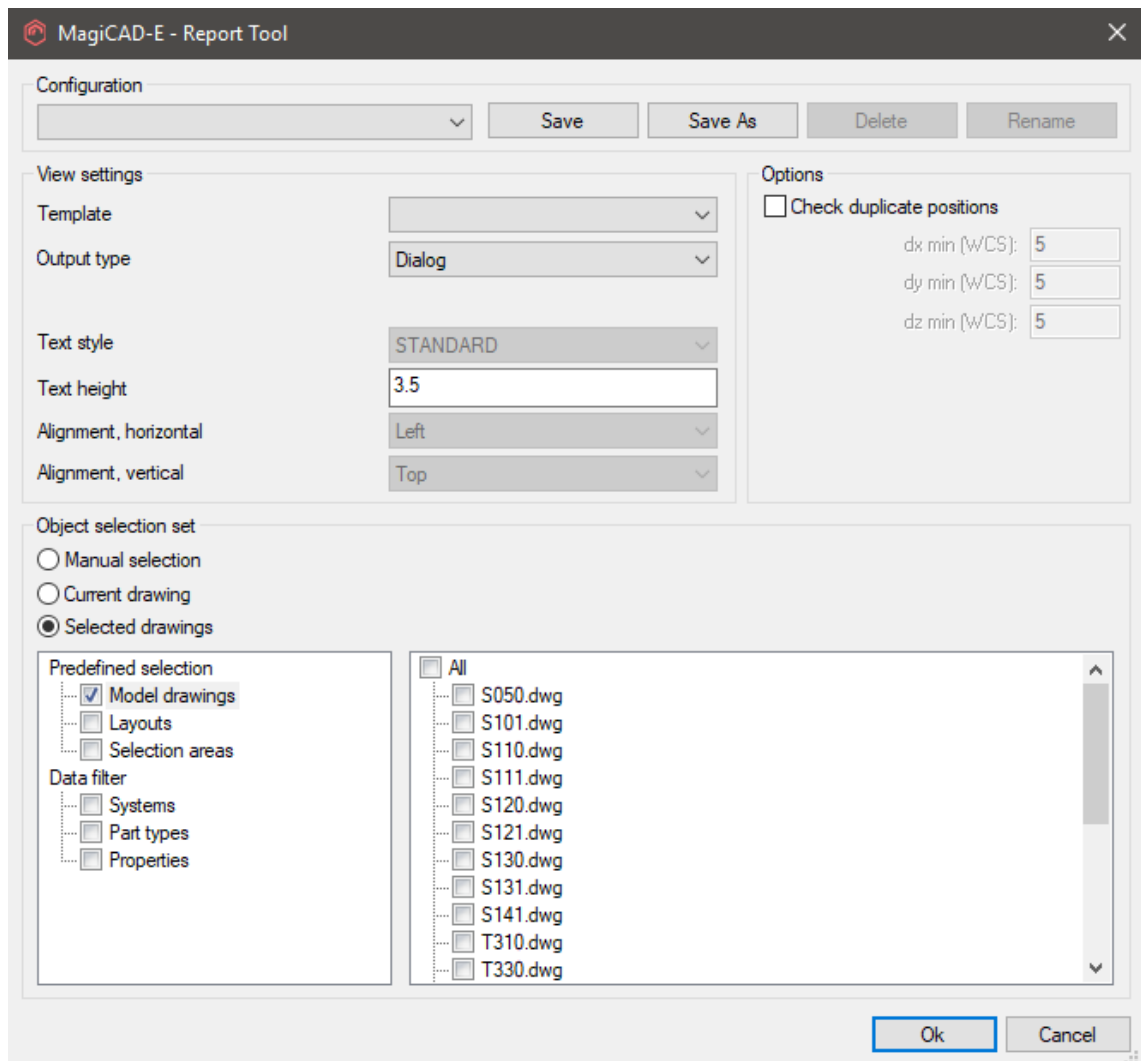
Seuraavassa alaluvussa käsiteltävä raporttityökalu käyttää kaikkien ominaisuuksien käsittelyyn valmista pohjaa (template) ja attribuutit haetaan symboleista tämän perusteella. Se on määritetty tietokantaan ja sitä voidaan käsitellä vapaamuotoisesti projekinhallinnan kautta. Taulukon testaamista varten luodaan oma tietopohja, jotta vanhan valaisinluettelon soveltaminen on edelleen mahdollista kehitysvaiheen aikana. Uuden valaisinluettelon virallisen käyttöönoton myötä korvataan käytössä oleva oletuspohja uudella versiolla, mutta sen edeltäjä säilytetään vielä toistaiseksi tietokannassa, sillä se on käytössä meneillään olevissa projekteissa.

2.1.2 Raportointityökalu

Jotta tietokantaan pohjautuva katalogisointi olisi mahdollista, on siihen perustuva raportointityökalu välttämättömyys. MagiCAD-lisäosa mahdollistaa symbolien ominaisuuksien tulostamisen monessa eri muodossa. Luettelon attribuuteista voi muun muassa liittää suoraan suunnitteluohjelmaan AutoCAD-sovelluksen blokkina tai vaihtoehtoisesti kopioida leikepöydälle erillisen ponnahdusikkunan kautta. Raportointiin on kaksi vaihtoehtoa, joista toista (Bill of Materials) käytetään pääasiassa vain sisäisten dokumenttien laadintaan, sillä sen tulostusomaisuus on rajattu ponnahdusikkunaan. Enemmän vaihtoehtoja tarjoava raportointityökalu (Report) mahdollistaa edellä mainittujen tapojen lisäksi symbolikohtaisien attribuuttien tallentamisen erilliseen xml-tiedostoon. Uusi valaisinluettelo pohjautuu tähän malliin, sillä se havaittiin luotettavimmaksi toimintatavaksi kehitysvaiheen aikana.

Vaikka raportointia käytetään pääasiassa määrälaskennan apuvälineenä, ei sen toiminnallisuus ole suinkaan rajoittunut vain tähän käyttötarkoitukseen. Työkalulla voidaan laatia projektikohtaisesti myös erityylisiä listauksia symbolien attribuuttien perusteella. Esimerkiksi keskuksista ja kalusteista voidaan tarpeen mukaan luoda luettelo tietyillä rajauksilla. Lisäksi yleiskaapelointijärjestelmän telepisteiden liitännät kytkimien portteihin tai paloilmoinjärjestelmän ilmaisimien osoitetiedot ohjauskeskuksittain ovat helposti raportoitavissa, mikäli lisäosan muut älyominaisuudet ovat käytössä. Täten työkalua soveltaessa on erityisen tärkeää, että tietokantaa ylläpidetään oikeaoppisesti.

Testivaiheessa käyttäjiä opastetaan attribuuttien täyttämässä, jotta ne saadaan tulostettua xml-tiedostoon taulukon tulkitsemassa muodossa. Ryhmää rohkaistaan myös kekeilemaan luettelon sopeutuvuutta opastamalla heitä henkilökohtaisien arvojen, kuten asennustilan lisäämisessä valaisinsymboliin. Kuvassa 3 on näkymä työkalun valintaikkunasta ennen raportin laadintaa. Käyttöliittymä on helppokäyttöinen ja rajaamisen päätteeksi voidaan asetukset tallentaa nimellä osaksi kokoonpanoa toistuvaa käyttöä varten.



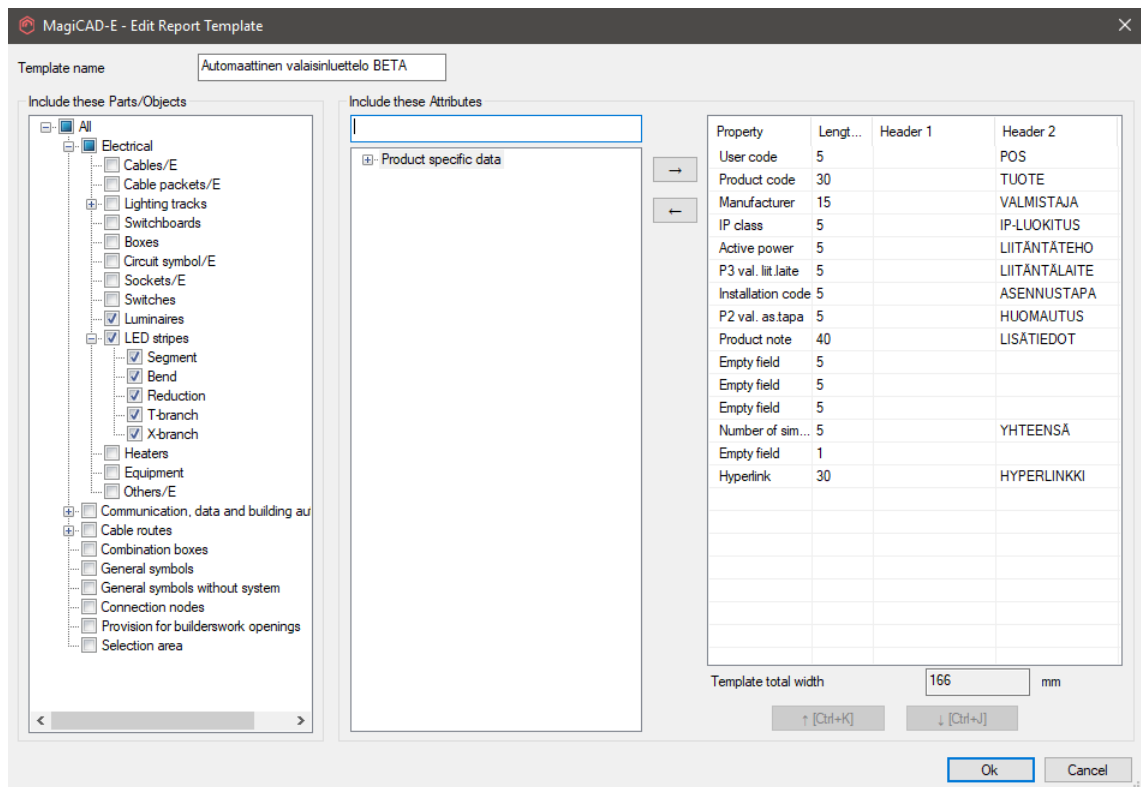
Kuva 3. MagiCAD-raporttityökalun avulla voidaan tuoda tietokannasta eri tavoin rajattuja tietoja. Asetukset voivat muodostua tasopiirustuksien symbolien tai ominaisuuksien rajauksesta sekä kaikkien näiden yhdistelmästä.

Raportin tuloksena tuotetun xml-tiedoston ulkoasu on merkityksetön, sillä uusi luettelopohja ei ollenkaan hyödynnä sen muotoiluja. Täten lähtötietoja tulostaessa ei juurikaan ole tarvetta kiinnittää huomiota sen tekstiasetuksiin. Ennalta määrätyt rajaukset (predefined selection) toteutetaan yleisesti dwg-tiedostojen mukaisesti, mutta lisärajoituksia voidaan toteuttaa näiden tulostusikkunoiden (layout) ja valinta-alueiden (selection area) avulla. Täsmennetty tarkkuus voi projektikohtaisesti olla tarpeellinen, mikäli rakennusosista tai vuokratiloista laaditaan erilliset luettelot. Taulukon toiminnallisuuden testaamiseksi se tukee lähtökohtaisesti vain yhtä xml-tiedostoa, mutta edellytykset osittaiselle määrälaskennalle listataan jatkokehitysmahdollisuuksia varten.

Suodattimella (data filter) voidaan eriyttää tasopiirustuksien sisältöä hyödyntäen symbolikohtaisia ominaisuuksia. Symbolit ovat jaettu tietokannan mukaisesti omiin järjestelmiinsä. Suunnitteluvaiheessa on syytä kuitenkin huomioida tilanteet, joissa valittu järjestelmä ei ole asianmukainen ja vaatii muutosta. Tällaiset tilanteet tulevat usein esille kiinteistöissä, joiden sähkönsyöttöä palvellaan monin eri tavoin. Mikäli käytössä on varmettu syöttöjärjestelmä, kuten UPS-laite tai generaattori, on näihin järjestelmiin liittyvät symbolit sijoitettava piirtoteknisesti oikealle järjestelmätasolle suunnitteluohjelmassa. Siten raporttia luodessa voidaan tasopiirustuksien sisältö rajata tarpeiden mukaisesti taulukon käyttöä ja dokumentaatiota varten. Lajittelulle voi olla myös ulkopuolisia perusteita. Esimerkiksi käyttäjän toiveesta voidaan eri rakennusosien tai vuokralaistilojen valaisimet luetteloida erikseen, mikä on hyvin tyypillinen käytäntö suuremman kokoluokan projekteissa. Myös tästä syystä uusi luettelo tukee vain yhtä lukumääräkenttää eikä siinä voida esittää eri osia rinnakkain. Tästä huolimatta luettelopohjan runko toteutetaan siten, että tällaisen esittämistavan lisääminen on tulevaisuudessa mahdollista.

Mikäli erityiselle jaottelulle ei ole tarvetta, voidaan kaikki valaisimet sisällyttää yhteen raporttiin. Tämä voidaan toteuttaa yksinkertaisesti käyttämällä symbolin tyyppi-attribuuttia. Kaikille symboleille on oma valikoima tietokannassa, joka määrittelee, onko kyseessä esimerkiksi sähkökeskus vai -moottori tai vaihtoehtoisesti kaluste, kuten pistorasia tai kytkinlaite. Tämä tieto määritetään tyyppi-attribuuttiin, joka löytyy myös valaisimista. Suunnitteluohjelmalla voidaan piirtää myös LED-profiileita, jotka kuuluvat omaan alavaikoihin. Nämä saadaan raporttiin muiden symbolien tavoin, mutta kappalemäärän sijaan ohjelma esittää sen kokonaispituuden. Siten on asiallista tuoda esille luettelossa, että kyseisen valaisinposition määrä on esitetty metreinä sekä liitäntäteho suhteessa sen pituuteen.

Kaiken rajaamisen päätteeksi voidaan asetukset tallentaa nimellä kokoonpanoksi (configuration), jolloin yhdenmukaisen raportin laadinta onnistuu vaivattomasti. Uusi luettelo hyödyntää lähtötietona xml-tiedostoa, joka voidaan valita tulostuksen tyyppiä (output type). Sen otsikkokentät näkyvät poikkeavalla taustavärillä sekä fontin tyyppi voidaan määrittellä raporttia laatiessa. Uuden luettelon muotoilu on eroteltu täysin lähtötiedoista ja siten xml-tiedoston ulkoasu ei vaikuta siihen. Sarakkeiden leveys asetetaan kuvan 4 mukaisesti tietopohjassa, jossa on esitettyä myös uuden taulukon käyttämät attribuutit.



Kuva 4. Symbolityyppien sisältämät attribuutit voidaan suodattaa raportin tulosteeseen projektin tietokannan kautta. Oikeassa laidassa on esitetty luetteloa vastaava otsikkokenttä sekä vasemmalla näitä vastaavat ominaisuudet, joihin tiedot täytetään.

2.2 Nykyinen luettelo

Olen toiminut tilaajayrityksen palveluksessa sähkösuunnittelutehtävissä lähes neljä vuotta. Vaikka ajallinen määrä on suhteellisen lyhyt, olen saanut mahdollisuuden työskennellä usean projektipäällikön kanssa ja siten saanut vankan käsityksen toimiston toimintatavoista ja -malleista. Kehitettävän luettelon tavoin on toimipisteen käytössä MagiCAD-lisäosan raporttityökaluun pohjautuva Excel-taulukko, joka kopioidaan alkuvaiheessa projektikohtaiseen hakemistoon mallikansiosta. Pohja on yksinkertainen eikä siihen ole integroitu automatisoituja tai tuoteälyä vaativia ratkaisuja. Rajoitetun toiminnallisuutensa vuoksi AutoCAD-sovelluksen sisäinen laskentataulukko olisi hyvin epäkäytännöllinen eikä se saavuttaisi merkittäviä hyötyjä Excel-pohjaiseen ratkaisuun verrattuna. Siten valmiiksi laadittu luettelopohja on saavuttanut suosion suunnitteluryhmän keskuudessa. Kuvassa 5 on näkymä toimipisteen nykyisestä valaisinluettelosta.

1	A	B	C	D	E	F	G	H	I	J	K	L
2	REJLERS		Rakennuskohteen nimi ja osoite		Laatinut		Päiväys	Sivut				
3	www.rejlers.fi		KOHDE		Piirtäjä		Muutos päiväys	Muutos				
4	Yrjökatu 2 B		Osoite		Työ Nro		Piirustus Nro					
5	01650 Vantaa, Finland		Osoite		3310xxx							
6	Puh. +358 207 520 700		Sisäilma		Piirustuslaji		Tiedosto					
7	etunimi.sukunimi@rejlers.fi		VALAISINLUETTELO		SÄHKÖPIIRUSTUS		rejlers_valaisintaulukko_fi.xlsx					
8												
10												
11	POS	VALAISINTYYPPI	VALAISIN	VALMISTAJA	VALONLÄHDE	TYYPPI JA TEHO	ASENNUSTAPA	LIITÄNTÄLAITE	HUOM	YHT	REV	
12												
13	HUOMAUTUKSET		VALONLÄHDELYHENNYKSET					ASENNUSTAPAMERKINNÄT				
14	1)	xxxxxxxx	T = loistelamppu					K = kattoasennus				
15	2)	xxxxxxxx	T5 = loistelamppu					S = seinäasennus				
16	3)	xxxxxxxx	TC = pienoistoistelamppu					V = väjieriasennus				
17			TC-SE = pienoistoistelamppu E27					U = uppoasennus				
18			T-R = rengasloistelamppu					R = ripustusasennus				
19			LED = puoli johdevalonlähde					RK = ripustuskoasennus				
20			A = vakioheijonlampu					KK = kosketinkoasennus				
21			PAR = puuristeiskohdelamppu					RP = ripustusputkiasennus				
22			QT = halogeenilamppu					KL = kalustossa				
23			QM = etohopeahöyrylampu					RV = väjieripustus				
24			HS = suurpainatriumlampu					EK = erikoiskiinnittimin				
25			HIT = monimetallilamppu					KR = ketjuriipustus				
26								P = pylväsasennus				
27												
28												
29												
30												
31												
32												
33												
34												
35												
36												
37												
38												
39												
40												
41												
42												
43												
44												
45												
46												
47												
48												
49												
50												
51												

Kuva 5. Nykyistä ulkoasua tullaan jäljittelemään uudessa tuotteessa. Muutoksina nimiö korvataan valaisinluettelosta erotetulla etulehdellä sekä aputaulukkoa käytetään muutoksien ylläpitoon.

Manuaalisen käsittelyn vuoksi pohjan kopiointi tapahtuu usein kuitenkin edellisestä projektista ja mallikansiossa sijaitsevaa pohjaa hyödynnetään harvakseltaan. Koska taulukko on ollut aina vapaasti muokattavissa, on jokaisessa luettelossa useiden kollegoiden projektikohtaisia muutoksia ja lisäyksiä. Tämän vuoksi uuden taulukon muotoilu-mahdollisuuksia on rajoitettu huomattavasti, ja sen sopeutuvuus kaikkien käyttäjien tarpeisiin on otettu huomioon jo kehityksen alkuvaiheessa. Lisäksi huomiota kiinnitetään taulukon saatavuuteen suunnitteluohjelman kautta. Yhtenäisten toimintatapojen puuttuessa, nämä muunnokset tuovat omanlaisia haasteita suunnittelu ympäristöön, mitkä heijastuvat etenkin suunnitteluohjelman ja taulukon väliseen täsmällisyyteen. Usein taulukon esitäyttö suoritetaan raporttityökalun avulla, mutta jälkityö tehdään risteävästi sekä tietokannassa että luettelossa. Tämä ongelma korostuu etenkin tilanteissa, joissa taulukkoon on lisätty henkilökohtaisia tietoja tai muotoiluja, sillä dokumentointi tietokannan ylläpidosta on usein puutteellista.

Nykyinen taulukkopohja tuo luonteensa vuoksi myös muita ongelmia suunnittelu ympäristöön. Valaisinluettelo asiakirjana vaatii aina päivittämistä muun aineiston rinnalla, mikäli siihen liittyviä muutoksia on laadittu muissa suunnitelmissa. Vaikka raporttityökalun luomat lähtötiedot ovat hyödyllisiä taulukon käyttöönotto vaiheessa, ei se kykene itsenäisesti dokumentoimaan tapahtuneita muutoksia, jonka myötä asianmukainen ylläpito on täysin käyttäjän vastuulla. Määrälaskennasta johtuvat muotoilut ovat usein yksinkertaiset, mutta niitä joudutaan tekemään valaisinkohtaisesti. Muutosten lukumäärän kasvaessa samankaltaisten toimenpiteiden määrä kasvaa ja virheiden todennäköisyys lisääntyy. Aika ajoin havaitaan myös tilanteita, joissa valaisin on poistunut aineistosta kokonaan, mutta se on unohdettu hävittää luettelosta, mikä aiheuttaa ongelmia hyvityksissä. Tällaisten huolimattomuusvirheiden ilmeneminen on lähes väistämätöntä laajemmissa hankkeissa ja tilanteista seuraavat selvitystyöt ovat usein hyvin raskaita ja rahallisesti arvokkaita prosesseja.

2.3 VBA-ohjelmointikieli

2.3.1 Valintaperusteet

Visual Basic for Applications on Microsoft Office-tuotteissa käytetty, ja kehitetyssä tuotteessa hyödynnetty, isäntäohjelman sisällä toimiva ohjelmointikieli. Kieli perustuu makroihin, jotka kutsuvat ja ohjaavat isäntäohjelman toimintoja. Excel-sovellus on vallitsevassa asemassa tilaajayrityksen kaikkien toimialojen taulukkolaskennassa. Sovellusta hyödynnetään myös kattavasti luetteloiden pohjana. Suunnitteluohjelmistojen osalta käytössä on useampia sovelluksia, mutta valtaosa näistä vaikuttaa enimmäkseen LVI-suunnittelijoiden työtehtäviin. Esimerkiksi nykyisin CADMATIC Oy:ta osana olevan suomalaisen Kyndata Oy:n kehittämä CADS-ohjelmisto (CADMATIC) on yleinen työkalu talotekniikan suunnittelu ympäristöissä. Kyseistä sovellusta käytetään myös toimeksiantajan toimipisteessä, mutta sen soveltaminen sähkösuunnittelijoiden keskuudessa rajautuu pääosin ulkopuolisilta saatujen tiedostojen käsittelyyn. Näistä esimerkkinä ovat sovelluksella laaditut keskuksien kokoonpanopiirustukset, jotka ovat osana luovutuspiirustusaineistoa, mutta tulevat usein muilta toimijoilta.

Myös suunnitteluohjelmat tarjoavat erilaisia ohjelmointiympäristöjä. AutoCAD-ohjelma tukee muun muassa alla listattuja API-mahdollisuuksia, joilla voidaan toteuttaa vuorovai-
kutteisia rajapintoja eri kolmannen osapuolen sovelluksien välille hyödyntäen valmiita
objekteja sekä työkaluja.

- AutoLISP
- Visual LISP
- VBA
- VB.NET
- ObjectARX

MagiCAD-lisäosa ei laajenna isäntäsovelluksen ohjelmointikapasiteettia, mutta ole-
massa olevia ratkaisuja voidaan hyödyntää myös laajennusosassa. Koska MagiCAD tar-
joaa oletuksena kattavat työkalut tietokantapohjaiseen raportoimiseen ja niiden käyttö
on luontevaa henkilöstölle, ei tarvetta suunnitteluohjelmistopuolen tuotekehitykselle kar-
toitettu tämän työn aikana. Näin ollen kehitystyön pääpainona on uuden luettelon auto-
matisoidut toiminnot sekä käyttäjäystävällisyys. Työn alkuvaiheessa toteutetulla rajauk-
sella halutaan taata taulukon riittävä toiminnallisuus ja luotettavuus, sillä kyseessä on
toimipisteen ensimmäinen automatisoitu suunnitteluprosessi. Lisäksi rajauksella vahvis-
tetaan tuotteen valmistumista suunnitellussa aikataulussa, johon sisältyy testausvai-
heen käyttäjäpalautteen jäsentely.

Excel-tilukkolaskentasovellus on yleinen työkalu sekä työympäristöissä että koulutus-
ohjelmissa (1). Yrityksen koosta, liikevaihdosta tai käsiteltävästä tietosisällöstä riippu-
matta viedään aineisto useimmiten taulukkoon tarkasteltavaksi (2). Siten voidaan pitää
todennäköisenä, että suurin osa tämänhetkisistä sekä tulevista työntekijöistä omaa vähin-
tään ohjelmiston perustaidot, jotka riittävät uuden valaisinluettelon käyttöön. Lisäksi Ex-
cel-sovellus on saatavilla useammalle käyttöjärjestelmälle, se tukee kattavasti aiemmillä
versioilla laadittua aineistoa (3) sekä sen ylläpidosta vastaa suuri ohjelmistotalo. Siten
on oletettavaa, että ohjelmiston hyödyntäminen pysyy vähintään samassa käyttölaajuu-
dessa seuraavan 10 vuoden ajan.

Suoritettujen nettihakujen sekä sovelluksiin perehtymisen myötä voidaan linjata, että
kaupallisten ratkaisujen saatavuus on erittäin rajallista. Suurimpana rajaavana tekijänä

on suunnitteluun sekä luettelointiin käytettävät eri ohjelmistot. Moni pienempi yritys tarjoaa räätälöityjä ratkaisuja eri sovelluksien rajapintoihin, mutta näiden hyödyntäminen kansainvälisessä työyhteisössä on haasteellista. Lisäksi useat ohjelmistot tarjoavat mahdollisuuden sisäiseen raportointiin sekä tietojen katalogisointiin, mutta näiden soveltamismahdollisuudet suoraan asiakasyrityksen suunnitteluympäristössä ovat vähäiset. Myös AutoCAD-sovelluksesta löytyy ominaisuus, jolla luodaan taulukoitu listaus suoraan dwg-tiedostoon käyttäen lähtötietona Excel-tiedostoa ja ne voidaan synkronoida toisiinsa päivittämistä varten. Kyseisestä toiminnosta ei kuitenkaan ole merkittäviä etuja tämän hetkessä ylläpitoprosessissa. Edellä mainittuihin tekijöihin vedoten, on yrityksen sisäiselle tuotekehitykselle perusteita, sillä

- markkinoilta ei löydy luettelon ylläpitoon suoranaista kaupallista ratkaisua
- kaupallinen ratkaisu vaatisi erillisinvestointeja tai –lisenssimaksuja
- itse kehitetty sovellus voidaan muokata laajemmin yrityksen käyttötarpeisiin
- työympäristössä voidaan järjestää helpommin henkilökohtaista koulutusta
- luettelon käyttöohjeiden laadinnassa kyetään huomioimaan käyttäjäkunta
- luettelon voidaan liittää osaksi muita kehityshakkeita tai käyttää vaihtoehtoisesti niiden perustana.

Lisäksi kyseessä on toimipisteen ensimmäinen automatisoitu suunnitteluratkaisu, jolle asetettiin selvästi painoarvoa tuotekehityksen lähestymistapojen valinnassa. Kattavasti testatun lopputuotteen lisäksi halutaan kerätä riittävästi käyttäjäkokemuksia jatkokehittämisen kartoittamiseen.

2.3.2 Heikkoudet

Vaikka automatisoitu luettelo tuo monipuolisia etuja suunnitteluprosessiin, on sen kehittämisessä omat haasteensa. Kuten johdannossa on linjattu, käsitellään tässä työssä tuotekehitykseen sekä koodikieleen liittyvää turvallisuutta, ylläpitoa ja kilpailukyvykkyyttä vain merkittävimmiltä osin. Nämä aiheet ovat sisällytetty työhön toimeksiantajan tahdosta, sillä ne ovat osana tuotekehityksen kokonaisuutta ja vaativat siten huomiota jatkokehittämisen näkökulmasta. Lisäksi työn lopputavoitteena on tuottaa yrityksen kaupalliseen käyttöön tarkoitettu sovellus ja siten Excel-/VBA-pohjaisen ratkaisun potentiaalisia

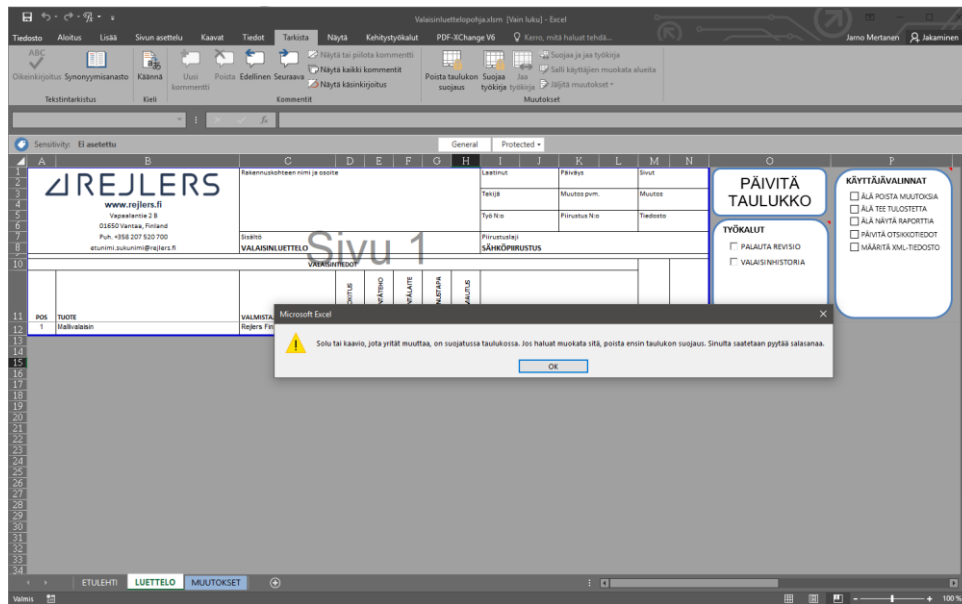
ongelmia tulee käsitellä työssä. Mikäli käyttäjäpalaute johtaa laajempaan ohjelmistokokonaisuuteen, on näihin tekijöihin kiinnitettävä erityistä huomiota

Luonnollisesti yrityksen sisäisen sovelluksen luominen vaatii ohjelmointikieltä hallitsevan henkilön. Mikäli kyseistä työnantoa ei ulkoisteta erilliselle palveluntarjoajalle, on ohjelmiston kehityksestä vastaava henkilö löydettävä yrityksen henkilöresursseista. Tämä ei ole konsulttialalla itsestäänselvyys, sillä useat käyttäjät hallitsevat vain sovelluksen perustoiminnot ja siten kehittyneemmät toiminnot jäävät usein hyödyntämättä tai niistä ei olla tietoisia. Toisaalta etuna on VBA-ohjelmointikielen ja sen isäntäohjelman laaja-alainen soveltaminen (1), jonka myötä kiinnostuneet löytävät runsaasti aineistoa verkosta, kuten toteutusohjeita, valmiita esimerkkejä sekä ammattilaisten tarjoamaa avustusta. Uuden valaisinluettelon kehittämisestä vastaa tässä projektissa opinnäytetyön laatija. Tämän valinnan taustalla ovat henkilökohtainen kiinnostus ja taustatieto ohjelmointikielistä sekä halu kehittää ja automatisoida suunnittelutehtäviä. Vaikka VBA on ohjelmointikielenä laadittu helposti lähestyttäväksi, vaatii suurempien kokonaisuuksien rakentaminen aikaa ja henkilökohtaista investointia. Siten kieltä hallitsevia henkilöitä ongelmatilanteiden ratkaisemiseen sekä teknisen tuen antamiseen saattaa olla rajallisesti.

Projektin jälkeinen ylläpitovaihe on rajattu ulos opinnäytetyön sisällöstä, muttei sitä voida jättää täysin huomioimatta. Laskentataulukkosovelluksen rinnalla uudistuvat myös suunnitteluun hyödynnettävät työkalut jatkuvasti. On tyypillistä, että käytössä olevista ohjelmista julkaistaan uusi versio vuosittain. Tämä tekee uudesta tuotteesta niin sanotusti irrallisen liikkuvan kohteen, jonka ympärillä sekä isäntäohjelmaa että muita rinnakkaissovelluksia päivitetään taukoamatta ulkopuolisten ohjelmistokehittäjien toimesta (4). Kyseinen ympäristö luo mahdollisesti tilanteita, joissa aiemmin ongelmitta toiminut ominaisuus tai tietty osuus koodista lopettaa toimintansa osittain tai lamauttaa koko tuotteen toiminnan. Vaikka ulkopuoliset ohjelmistokehittäjät tekevät parhaansa aiempien versioiden välisen yhteensopivuuden säilyttämiseksi, ei tämä takaa yritykselle kehitetyn tuotteen jatkuvaa toimivuutta. Lisäksi on mahdollista, että sisäiseen tarpeeseen kehitetyn tuotteen ominaisuudet löytyvät jonain päivinä osana erillistä suunnitteluohjelmaa, mikä tekee siitä tarpeettoman. Pidän kuitenkin edellä mainittua epätodennäköisenä, sillä muutoksien vertailun lisäksi, on sovelluksen kyettävä muotoilemaan aineisto ja suorittamaan dokumentaatio yrityksen käyttämän luettelopohjan mukaisesti.

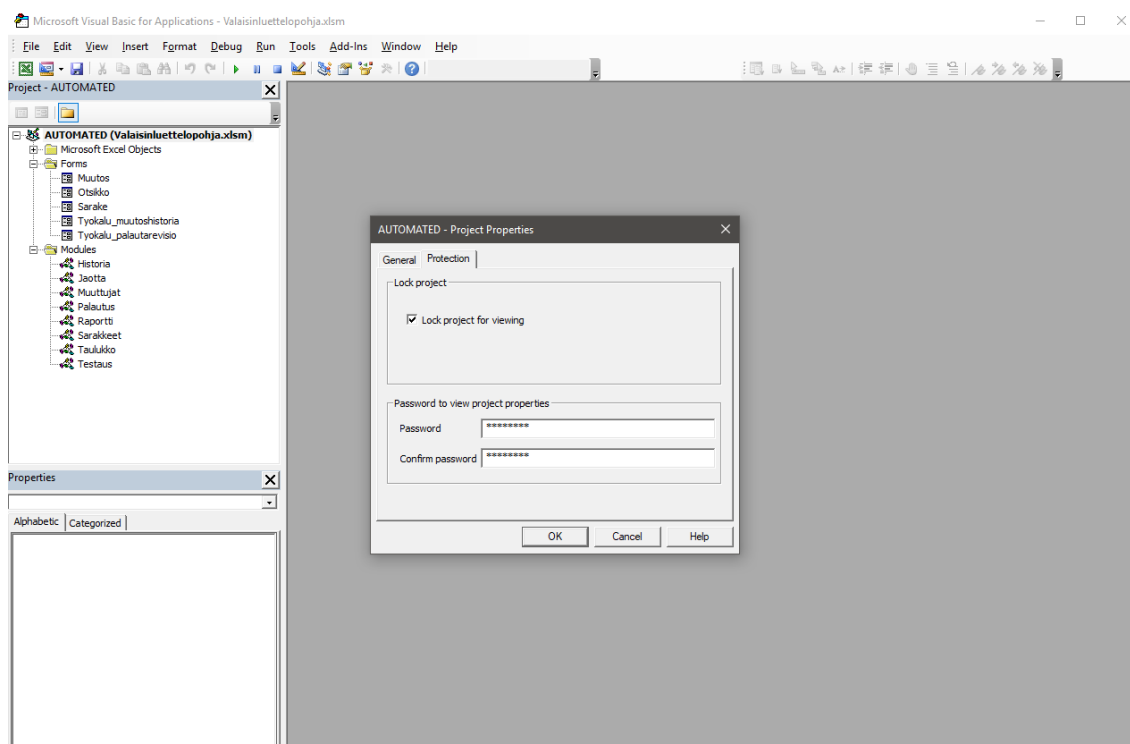
VBA-ohjelmointikielellä toteutettuun ratkaisuun liittyy myös tietoturvariskejä, joista käyttäjien sekä kehittäjien on oltava tietoisia. Kuten edellä on pohjustettu, on VBA isäntäohjelmasta riippuvainen koodikieli. Siten luettelon toiminnot ovat väistämättä osana Excel-taulukkolaskentasovellusta tai tiedostotyyppiä, jonka isäntäohjelma tunnistaa xlsx-päätteestä (macro enabled). Tämän vuoksi sen toiminnallisuutta ei voida eriyttää itsenäiseksi sovellukseksi, mikä mahdollistaa kopion kulkeutumisen toisiin yrityksiin henkilöstövaihdoksen yhteydessä ilman toimeksiantajan suostumusta. Opinnäytetyössä laaditun tuotteen päätehtävänä on automatisoida muutosaineiston vaatimat muotoilut käyttäen yrityksen luettelopohjaa, mikä hankaloittaa ja kyseenalaistaa hyödyntämismahdollisuudet muissa sovelluksissa. Tästä huolimatta ovat sovelluksen koodiosuudet sopeutettavissa osaavan henkilön toimesta, minkä johdosta osaa toiminnoista voidaan käyttää myös kilpailijoiden sovelluksissa.

Edellä käsitelty ongelma pohjautuu pääasiassa Excel-/VBA-suojauksiin, jotka ovat ensisijaisesti laadittu välttämään yrityksen sisäisiä ongelmatilanteita rajoittamalla vuorovaiikutusta käyttäjien ja sovelluksen välillä. Excel mahdollistaa kahdentasoisen suojauksen, joista toinen vaikuttaa taulukoihin. Tämän avulla käyttäjille asetetaan muokkausrajoituksia, joiden tavoitteena on estää taulukon toiminnallisuuteen vaikuttavat tahattomat muutokset. Muun muassa toimintoja, välilehtiä sekä soluja voidaan lukita ja ajantasaisimmat lukitusohjeet löytyvät aina Microsoftin verkkosivuilta. Kuvassa 6 on esitetty ilmoitus, mikäli käyttäjä pyrkii muuttamaan lukittua sisältöä.



Kuva 6. Kun käyttäjä yrittää suorittaa muokkaustoimenpidettä lukittuun sisältöön, hän saa virheilmoituksen, jossa viitataan suojauksen poistamiseen. Suojauksen tavoitteena on estää käyttäjien ei-toivotut muutokset sovelluksessa.

Mikäli tiedosto sisältää VBA-pohjaisia makroja, voidaan myös nämä suojata. Tämä tapahtuu Visual Basic Editorin kautta, jota käsitellään tarkemmin seuraavassa kappaleessa. Kehittäjä voi suojata koko projektin tai sen osan. Salasana lisätään kuvan 7 mukaisesti lukittavan kohteen ominaisuudet-ikkunasta. Sovelluksen käytön kannalta VBA-sisällön suojaaminen ei ole yhtä tärkeää taulukon suojaamisen kanssa ja voidaan vaihtoehtoisesti jättää tekemättä, sillä loppukäyttäjät eivät käsittele tätä osuutta sovelluksesta. Lisäksi salaus ei palvele tietoturvaa merkittävästi. Excelin molemmat salasanat ovat helposti poistettavissa pakkausohjelmalla ja binaarieditorilla. Tähän löytyy kattavasti ilmaisia sovelluksia ja ohjeita jopa VBA-ohjelmointiin liittyviltä verkkosivustoilta (5). Tämä selkeästi vaikeuttaa VBA-pohjaisen sovelluksen kaupallista käyttöä, sillä suojaus ei ole riittävän vahva ja tekee tuotteesta haavoittuvaisen.



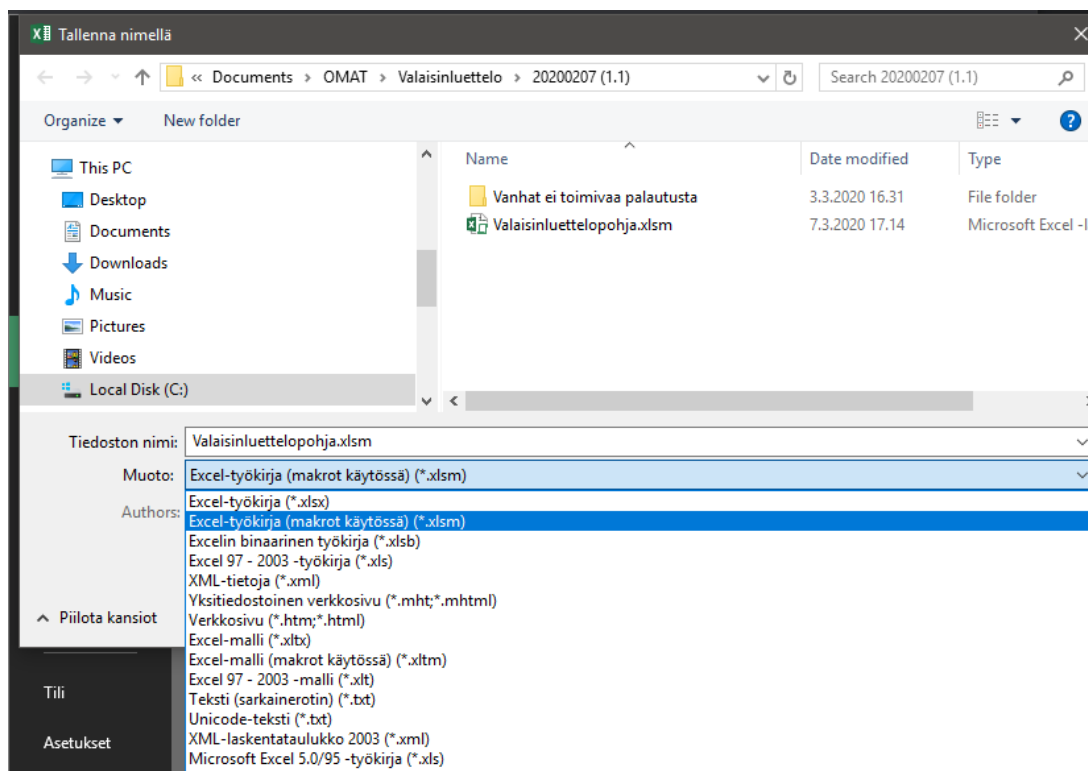
Kuva 7. VBA-pohjaisen projektin sisältö on suojattavissa joko täysin tai osittain Visual Basic Editorin kautta. Salasana asetetaan valitun osan ominaisuudet-ikkunassa.

VB.NET tarjoaa kehittäjille tehokkaammat työkalut suojaukseen ja kyseinen ohjelmointiympäristö on muutoinkin kehittyneempi VBA-ympäristöön verrattuna. Lisäksi AutoCAD-ohjelmistossa oli aiemmin sisäänrakennettu VBA-ympäristö, jota voitiin käyttää Excelin Visual Basic Editorin tavoin. AutoCAD tukee edelleen VBA-ohjelmointia, mutta tätä varten on ladattava erillinen maksuton moduuli valmistajan verkkosivuilta (6). Autodesk suosittelee .NET-ympäristöön siirtymistä, mutta ylläpitäjä ei ole julkaissut virallista lausuntoa VBA-ympäristötuen päättymisajankohdasta. Lisäksi Microsoft on edistänyt mittavasti JavaScript-ohjelmointikielen käyttöä Office-tuoteperheessään korvatakseen VBA-pohjaiset ratkaisut. Tämä ei ajankohtaisesti vaikuta kehitettävään tuotteeseen, mutta ohjelmistotalojen toiminnoista voidaan johtaa viitteitä VBA-ohjelmointikielen heikkenevästä asemasta. VBA-pohjaiset ratkaisut ovat edelleen vahvassa asemassa Office-tuotteiden laajan suosion vuoksi ja virallisen tuen päättymisestä ei ole annettu lausuntoa. Mikäli VBA-ohjelmointikielen tuki päättyy, on edessä useamman vuoden siirtymisaika. Siten kieleen pohjautuvia sovelluksia voidaan pitää ajankohtaisina vähintään seuraavan vuosikymmenen ajan.

Kaikista heikkouksista huolimatta laaditaan uusi luettelo VBA-pohjaisena xlsx-tiedostona. Excel-tilukkolaskentasovellus on yleinen työkalu toimipisteessä ja tilukon käyttöprosessia ei haluta monimutkaistaa tarpeettomasti erillisellä sovelluksella. Kyseessä on lisäksi henkilöstön ensimmäinen automatisoitu työkalu, joten tuotekehityksen rinnalla on käyttäjäkokemusten kerääminen tärkeässä asemassa. Työssä käytetään toimeksiantajan päätöksen myötä olemassa olevia sovellusmahdollisuuksia eikä toteutusta varten tehdä uusia erillisinvestointeja. Valinnan taustalla ovat myös työn laatijan ohjelmointitausta sekä VBA-ohjelmointikielen helppo lähestyttävyyden. Projektin riskit tiedostetaan yrityksessä ja ne huomioidaan tuotteen käyttäjäpalautteen myötä johdetuissa jatkokehityshankkeissa sekä tämän opinnäytetyöraportin yhteenvedossa.

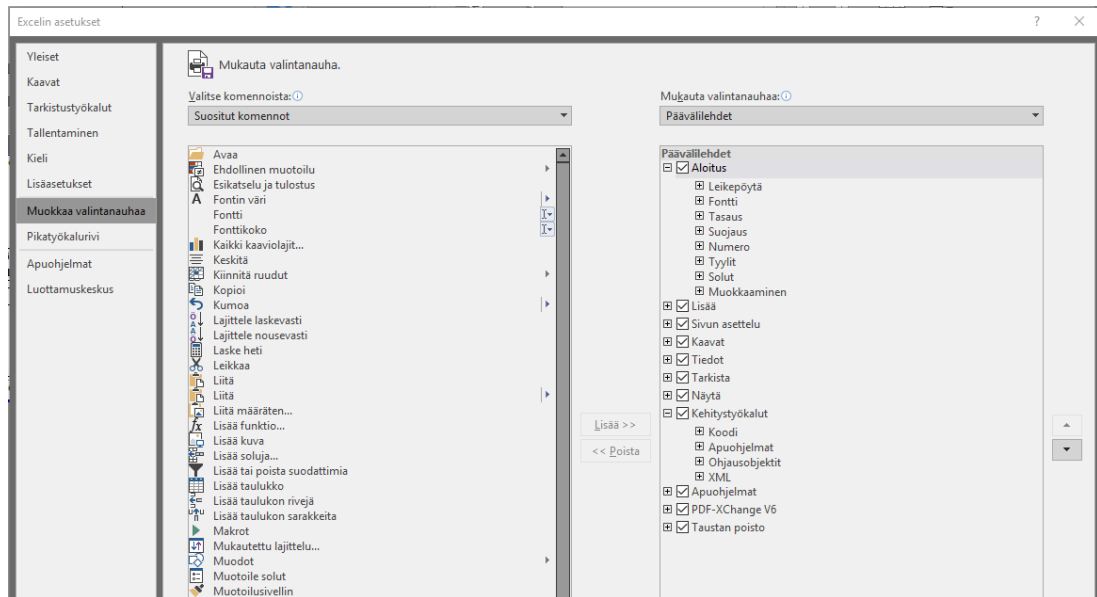
2.3.3 Visual Basic Editor

Microsoftin julkaisema Excel-sovellus tarjoaa mahdollisuuden komentojen automatisointiin VBA-ohjelmointikielen avulla. Kuten aiemmassa luvussa esiteltiin, on kieli isäntäohjelmaan pohjautuva ja se suorittaa jonomaisesti olemassa olevia toimintoja. Ohjelmointi tapahtuu siihen sisältyvällä editorilla, joka on osana myös muita Office-tuoteperheen sovelluksia. Visual Basic Editorin saa käyntiin näppäinyhdistelmällä Alt + F11, mutta sen laukaiseminen vaatii hieman pohjustusta. Aluksi on varmistettava, että tiedostotyyppi on oikea. Mikäli tyyppi on valittu väärin, eikä se ole ”xlsx”-päätteinen (macro-enabled), ohjelmoitu koodi ei tallennu tiedostoon. Valinnan voi tehdä helposti tallentamalla tiedoston nimellä kuvan 8 mukaisesti.

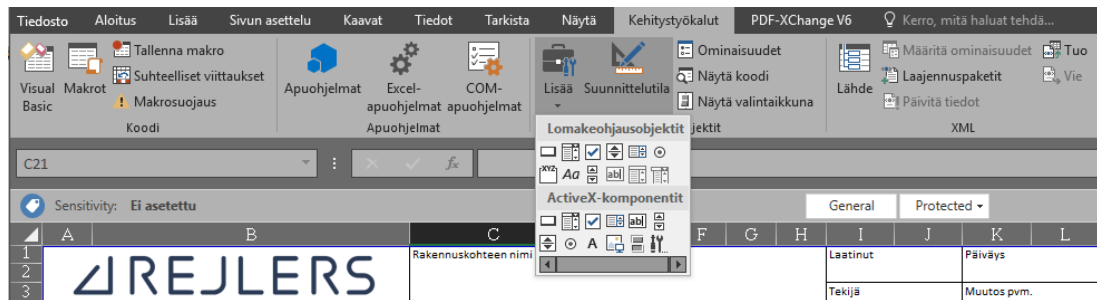


Kuva 8. Makroja tukevan taulukon tallentaminen onnistuu käyttämällä ”Tallenna nimellä”-ominaisuutta ”Tiedosto”-valikosta. Näin voidaan varmistaa, että tiedosto tukee siihen laadittua VBA-pohjaista sisältöä.

Jotta VBA-pohjaisia objekteja, kuten painonappeja voidaan lisätä taulukkoon, on otettava käyttöön sovelluksen kehitystyökalut. Nämä näkyvät normaalisti ylhäällä olevassa valintanauhassa erillisenä välilehtenä, mutta oletuksena ne ovat poissa käytöstä. Käyttöönotto tehdään valitsemalla ”Tiedosto”-välilehti ja ”Asetukset”. Avautuvasta ikkunasta löytyy valinta ”Muokkaa valintanauhaa”, jonka esittämältä listalta työkalut voidaan aktiivoida kuvan 9 mukaisesti. Kehitystyökalut mahdollistavat muun muassa makrojen nauhoittamisen, mikä on hyödyllistä etenkin Excelissä käytettävien värikoodien selvittämiseen. Painonappien sekä valintaruutujen sijoittaminen taulukkoon tapahtuu ”Lisää”-alavalikosta. Ne ovat jaettu kahteen kategoriaan, joista lomakeohjausobjektit toimivat yksinkertaisimpina vaihtoehtoina kehittyneimmille ”Active-X”-komponenteille. Nämä komponentit kykenevät objektitoimintojen lisäksi varastoimaan VBA-koodia. Siten esimerkiksi muiden valintaruutujen käytöstä poistaminen voidaan automatisoida, mikäli vain yksi niistä sallitaan. Ne mahdollistavat joustavamman toteutuksen ilman suoranaisia haittoja, mutta asettavat joitain rajoitteita kokonaisuudelle, joista kehittäjän on oltava tietoinen (7). Kuvassa 10 on kehitystyökalujen valintanauhanäkymä.



Kuva 9. Kehitystyökalut ovat oletuksena poissa käytössä ja ne on aktivoitava Excelin asetuksista. Tämän jälkeen ne ovat saatavilla valintanauhasta seuraavan sivun kuvan mukaisesti.



Kuva 10. Kehitystyökalujen valintanauha tarjoaa pääsyn kaikkiin tarvittaviin VBA-ominaisuuksiin. Editori voidaan laukaista vasemmalla olevasta painikkeesta tai vaihtoehtoisesti näppäinyhdistelmällä Alt + F11.

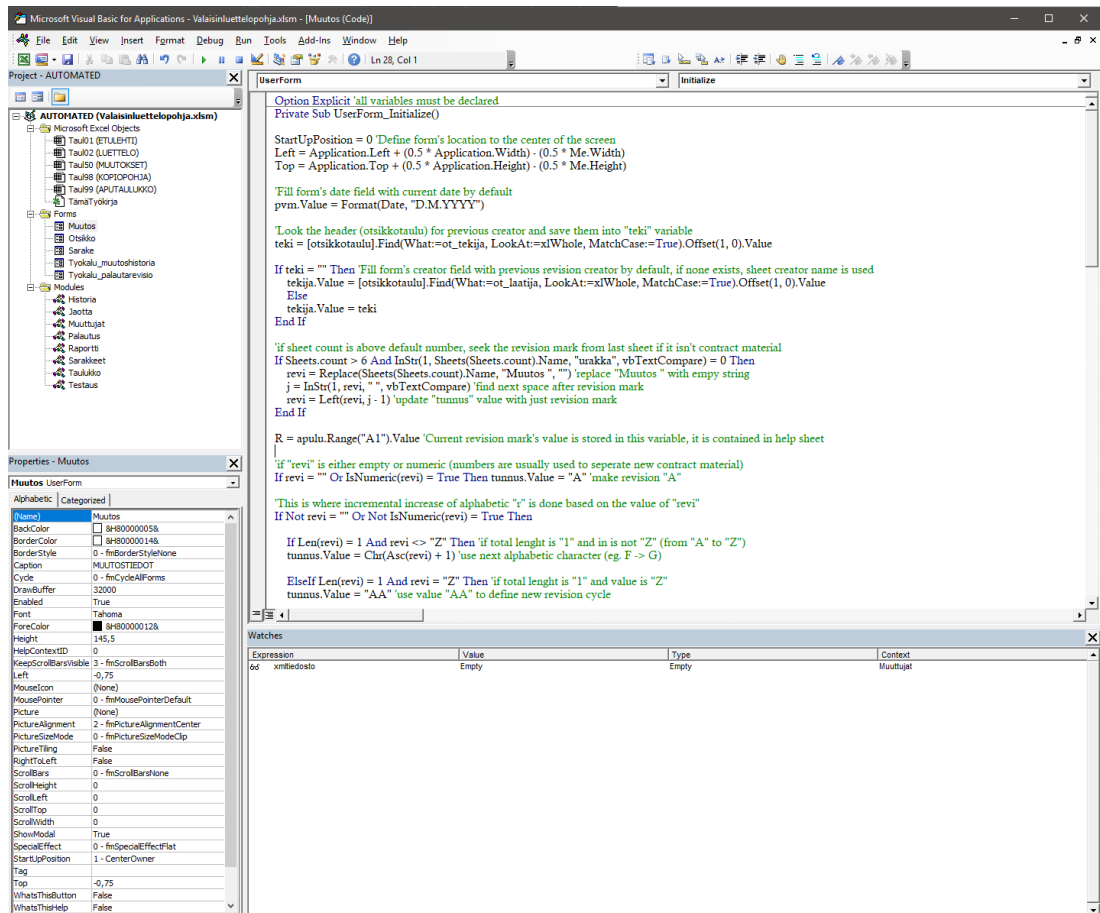
Editori tarjoaa kattavasti välineitä koodin laatimiseen ja hallintaan sekä elementtien ominaisuuksien muokkaamiseen. Aikataulullisista syistä tässä opinnäytetyössä käsitellään vain kehitettävän tuotteen kannalta olennaisimmat työkalut, jotka ovat

- Ominaisuudet (Properties Window, F4)
- Projektihallinta (Project Explorer, Ctrl + R)
- Objektiselain (Object Browser, F2)
- Koodinäkö (Code, F7)
- Katseluikkuna (Watch Window)

ja ne ovat saatavilla myös View-valikosta. Ominaisuuksien kautta voidaan muuttaa aktiivisena olevan objektin attribuutteja. Objektina voi olla esimerkiksi käyttäjälomake tai sen sisällä oleva tekstikenttä. Projektihallinta tarjoaa kehittäjälle puumaisen tiedostorakenteen aktiivisena olevan VBA-projektin sisällöstä. Objektiselain mahdollistaa kaikkien toimintojen ja objektien selailun sekä näyttää niihin liittyvät metodit ja ominaisuudet. Esimerkiksi valitsemalla listasta AutoFilter-toiminnon näyttää selain sen yhtenä ominaisuutena FilterMode sekä metodina ShowAllData, joka poistaa suodatuksen. Lisäksi toiminnon ollessa aktiivisena valikossa, voidaan F1-painikkeella laukaista sitä vastaava Microsoftin Help-sivusto.

Koodin kirjoittaminen tapahtuu koodinäkyvässä ja katseluikkuna mahdollistaa muuttujien arvojen seuraamisen reaaliajassa. Tämä tuo huomattavia etuja koodin ongelmien selvittämiseen. Muuttuja voidaan lisätä tarkasteltavaksi maalaamalla se koodista ja painamalla näppäinyhdistelmää Shift + F9 tai vaihtoehtoisesti klikkaamalla hiiren oikeaa ja valitsemalla "Add Watch...". Myös manuaalinen lisääminen kirjoittamalla on mahdollista. Selvitettävä arvo voidaan määrittää esimerkiksi testimuuttujaan ja tutkia toistuvasti, kuinka siihen kohdistuvat metodit vaikuttavat sen arvoon.

Suoritettavan koodin työkalut löytyvät pääosin Edit-, Run-, ja Debug-valikoiden alta. Näistä ensimmäisessä on muun muassa etsi-komento, jonka saa käynnistetty Windows-käyttäjille tutusta näppäinyhdistelmästä (Ctrl + F). Run- ja Debug-valikosta löytyvät enemmän kehittäjäluonteiset toiminnot. Koodiin voidaan asettaa pysähdyskohta (Breakpoint, F9), joka seisauttaa koodin suorittamisen sen osoittamalle riville. Näin uuden koodiosuuden testaus tai ongelmakohdan saavuttaminen on yksinkertaista. Tästä eteenpäin voidaan koodia edistää rivi kerrallaan painamalla F8-näppäintä ja koko koodiosuuden suoritus voidaan laukaista painamalla F5. Kuvassa 11 on näkymä opinnäytetyön soveluksen laadinnassa käytetystä asetelmasta.

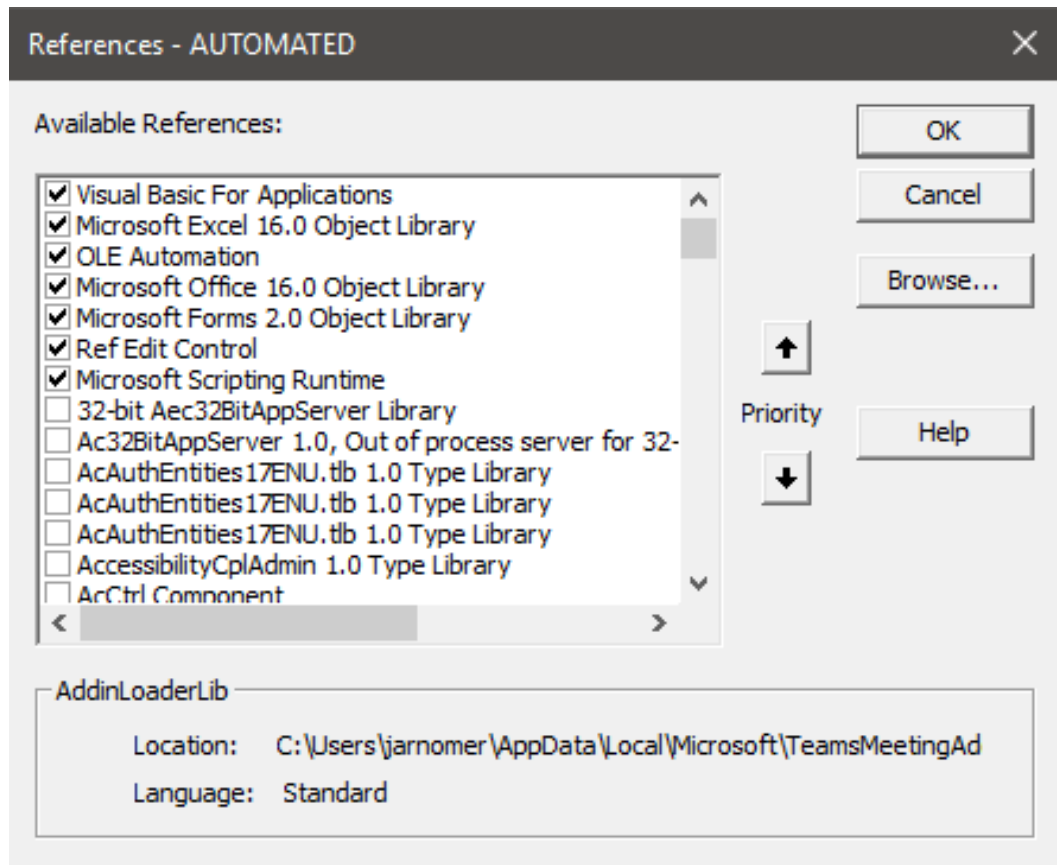


Kuva 11. Koodinäkyssä olevat vihreät rivit ovat kommentteja. Mikäli ikkunassa halutaan näyttää edemmän koodisisältöä, voidaan työkaluja eritellä editorista raahaamalla.

Kuvan koodinäkyssä on esitelty muutosaineiston lomakkeen käynnistyessä suoritettavaa koodiosuutta. Lisäksi on hyvä tiedostaa, että VBA-editori tukee lukuisia komentoja aiempien esille tuotujen kommentojen lisäksi (8). Esimerkiksi Alt + F11 keskittää myös aktiivisen ikkunan takaisin editorista Exceeliin. On suositeltavaa, että kehittäjän omaksuu kaikki tarpeelliset työkalut ja näppäinyhdistelmät jo projektin alkuvaiheessa, sillä näiden muodostama kokonaisuus edistää jokaisen koodiosan luontivaihetta ja vianmääritystä.

Ennen ohjelmoinnin aloittamista on lopuksi ladattava tarvittavat referenssikirjastot. Niiden tehtävänä on sanakirjan tavoin "kääntää" olemassa olevia "sanoja" VBA-kielen hyödyntämään muotoon (9). Aktiivisena olevien kirjastojen määrittäminen voidaan tehdä editorin "Tools"-valikosta ja valitsemalla "References". Ponnahdusikkuna esittää voimassa olevat referenssit valintaruuduilla ja nämä arkistoituvat suoraan VBA-projektitie-

dostoon. Siten niitä ei tarvitse ylläpitää työasemakohtaisesti, mutta muuttuvan kokonaisuuden vuoksi on mahdollista, että asetuksia joudutaan vaihtamaan jatkokehityksen myötä. Kuvassa 12 on esitetty tässä työssä tarvittavat referenssikirjastot.



Kuva 12. VBA-projektin referenssikirjastot voidaan määrittellä "Tools"-alasettovalikon kautta. Referenssit tallentuvat suoraan tiedostoon ja niitä voidaan muuttaa kehitystyön aikana.

2.3.4 Läheteitä ohjelmointiin

VBA on hyvin laajasti hyödynnetty ja helposti lähestyttävässä oleva ohjelmointikieli. Tämä on ymmärrettävä toteutustapa, sillä suurin osa kuluttajista on peruskäyttäjiä eikä heillä ei ole aiempaa kokemusta ohjelmoinnista. Siksi suurimmat hakukoneet palauttavat runsaan määrän osuvia haettuun ongelmaan. Kielen kirjoittaminen on alussa hyvin pitkälti valmiina löytyvien koodiosuoksien kopioimista, testaamista ja omaksumista. Ajan myötä kuitenkin toteutuksien pääpaino muuttuu kehittäjän omiin ratkaisuihin ja verkkosivustojen

käyttö siirtyy enemmän malliesimerkkien selailuun. Huomasin henkilökohtaisesti soveltavani tämän tyylistä toimintatapaa noin puolen vuoden aktiivisen kehitystyön jälkeen. Alla on lista eniten käyttämistä nettisivustoista hakukoneiden rinnalla.

- <https://docs.microsoft.com/en-us/>
- <https://stackoverflow.com/>
- <https://www.thespreadsheetguru.com/>
- <https://trumpexcel.com/excel-vba/>
- <https://www.wallstreetmojo.com/category/vba/>
- <https://excelmacromastery.com/vba-articles/>

Microsoftin sivusto sisältää enemmän virallista teknillistä tietoa esimerkiksi sovelluksen metodeissa ja objekteista sekä niiden argumenteista. Stackoverflow-sivustolta löytyy laajasti malliratkaisuja, mutta ne ovat usein kokeneempien VBA-ohjelmoijien esittämiä ja siksi aloittelijan voi olla vaikea sisäistää niitä. Näiden sijaan muut esitetyt sivustot tarjoavat kaiken tasoista tietoa myös VBA-ohjelmointikielestä kiinnostuneille peruskäyttäjille. Moni oppii parhaiten lukemisen ja kirjoittamisen lisäksi kuuntelemalla. Siten alla on kaksi henkilökohtaisesti suosittelemaani videosarjaa.

- <https://www.youtube.com/user/WiseOwITutorials/playlists>
- <https://www.youtube.com/user/ExcelVbalsFun/playlists>

Kaikki edellä mainitut lähteet ovat ilmaisia ja ne löytyvät verkosta, mutta markkinoilla on myös kattavasti maksullista sisältöä kirjallisuudesta opetuskursseihin. Nämä eivät kuitenkaan palvelleet opinnäytetyön rajaamien päämäärien saavuttamisessa, sillä täsmälliset ratkaisut ovat paremmin saatavilla verkkolähteistä (10).

3 Tuotekehitys ja testausprosessi

3.1 Toimintatapamuutokset

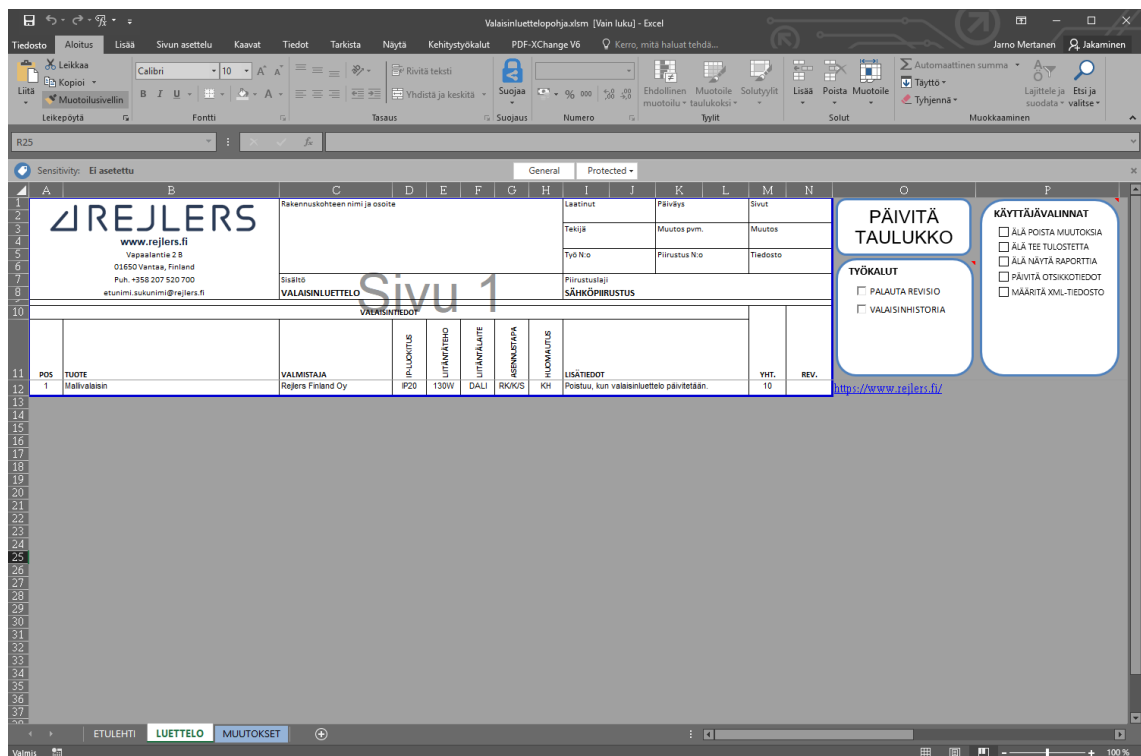
Aiemmin käsitellyt luettelossa olevat heikkoudet tarjoavat lähtökohdat myös toimintatapojen muutoksille. Suunnitteluohjelman hallinnointi tulee tapahtumaan edelleen käsin,

sillä sen toiminnallisuutta ei voida suoranaisesti rajata käyttäjien keskuudessa. Siten väärin tai puutteellisesti täytetyt symbolikohtaiset tiedot ovat yhä osana suunnitteluprosessia. Lisäksi tasopiiruksissa tapahtuneet erehdykset ja puutteet heijastavat aineiston paikkansapitävyyttä myös jatkossa. Tästä huolimatta voidaan uuden taulukon rajauksilla vaikuttaa ohjelmistopuolen käytäntöihin. Luettelon kaikki solut ovat lukittuna ja siten käyttäjät eivät kykene aiheuttamaan ristiriitoja taulukon sisällön ja projektin tietokannan välillä. Tämän myötä käsikäyttö on korvattu kokonaisuudessaan painonapilla ja tietomuu-
tosten päivittäminen tapahtuu käyttäjälomakkeiden kautta. Ainoa muokattava elementti on etulehdellä sijaitseva tekstikenttä, johon käyttäjät voivat lisätä koko luetteloa koskevia tietoja. Muut komponentit sekä muotoilu säilyvät lukittuna koko käytön ajan ja koodi käsittelee nämä suunnittelijan sijasta.

Alkuvaiheessa tehdyt rajaukset helpottavat taulukon kehitysvaiheiden läpiviemistä ja selkeyttävät sen kanssakäymistä ohjelmiston kanssa. On hyvin yleistä, että sovelluskäytössä kuluttajan mahdollisuuksia vaikuttaa tuotteen toimivuuteen rajoitetaan, eikä periaatteellisesti kyseinen konsepti juurikaan poikkea uudesta luettelosta. Ennalta määrätty tietoisuus käyttäjän vuorovaikutusmahdollisuuksista taulukon kanssa vähentää koodissa huomioitavien tilanteiden määrää ja virtaviivaistaa kokonaisuutta. Vastapainoisesti tämä vaatii erinomaista kommunikointia kehittäjän ja käyttäjien välillä.

Automatisoidun luettelon tuomat edut eivät suinkaan päädy inhimillisten virheiden minimointiin vaan tavoitteena on kehittää kyseistä suunnitteluprosessia myös hyväksymisen jälkeen. Valaisinluettelo on usein osana muutosaineistoa ja siten se työllistää lähes poikkeuksetta työryhmää näissä tilanteissa. Projektin laajuudesta riippuen voi käytössä olla satoja valaisinmalleja sekä suunnitelmia päivittäviä henkilöitä useita. Nykyinen luettelo edellyttää suunnittelijoilta muutoksien manuaalista läpikäymistä sekä tekemään vertailut ja muotoilut taulukkoon käsin. Luonnollisesti tämä ei ole pelkästään työllistävä, mutta myös virheherkkä työvaihe. Siten kehitettävän luettelon päämääränä on virtaviivaistaa muutosprosessi painonappiin sekä mahdollistaa käyttöönotto hyödyntäen samoja periaatteita. Lisäksi käyttäjien hyödyllisiksi havaitsemia sekä luettelon käyttöä palvelevia toimintoja lisätään taulukkoon kehitysvaiheittain.

Taulukon rajoitetulla muokattavuudella pyritään kiinnittämään huomiota myös laadunhallinnallisiin tekijöihin. Nykykäytännöt mahdollistavat suunnittelusovelluksen ja valaisinluettelon väliset ristiriidat, joita halutaan vähentää. Uuden luettelon soveltaminen vaatii pakollisten attribuuttien asettamisen ennalta määritetyllä periaatteella. Tämä tuo yhdenmukaisuutta tilanteisiin, joissa henkilöresursseissa muuttuvat tai käyttäjät haluavat tuoda valmiiksi täytettyjä symboleita edellisistä projekteista. Kuvassa 13 on esitetty erään kehitysvaiheen "LUETTELO"-välilehti, joka jäljittelee aiempaa ulkoasua luontevampaa käyttöönottoa varten. Kaikki taulukon uudet ominaisuudet ovat dokumentoitu käyttöohjeisiin, jossa suunnittelijoita ohjeistetaan jokaisen käyttövaiheiden läpiviennissä.



Kuva 13. Uuden luettelon käyttö pohjautuu täysin painonappiin, mutta sen ulkoasultaan tyyllittelee edeltäjäänsä. Kaikki uudet ominaisuudet ovat sijoitettu keskitetysti tulostusalueen ulkopuolelle, taulukon oikeaan reunaan.

3.2 Luettelon rakennemuutokset

Ennen kehityshankkeen aloittamista on toimipisteen valaisinluettelo perustunut mallikansiosista löytyvään taulukkopohjaan. Sen otsikkotiedot ovat osana taulukkoa ja ne on jaettu kolmeen osaan. Vasen laita sisältää yrityksen logon ja yhteystiedot. Keskellä sijaitsevat

projektikohtaiset tiedot ja viimeisenä esitetään dokumentin sisältö. Pohjasta löytyy erillinen nimiö-välilehti, jonka tehtävänä on toimia taulukon etulehtenä. Lisäksi luettelon lopussa on aputaulukko mahdollisia muistiinpanoja varten. Tätä rakennetta pidetään uuden luettelon lähtökohtana ja aiemmat toimintaperiaatteet halutaan säilyttää.


Nimiö vaihdetaan kuvan 14 mukaisesta pystysuorasta rakenteesta vaakasuuntaiseksi, jotta se on yhtenäisempi muiden välilehtien kanssa, mutta on edelleen toimiva luettelon etulehtenä. Otsikkotaulu muutetaan vastaamaan kuvassa 15 esitettyä luetteloa, jotta muutostietojen välittäminen etulehdelle olisi yksinkertaisempaa. Etulehti on täysin lukituna yhtä tekstielementtiä lukuun ottamatta. Siihen voidaan listata yleisiä projektikohtaisia tietoja, joiden ennalta määrittäminen on hankalaa ja siten käyttäjille annetaan täydet vapaudet sen muokkaamiseen.

Muut etulehden soluarvot päivitetään koodin avulla. Ensimmäisellä käyttökerralla luettelo pyytää projektitiedot lomakkeelle, josta se syöttää ne hyväksynnän myötä koko taulukoon. Näiden muokkaaminen on mahdollista myös myöhemmässä vaiheessa käyttäjävalintojen avulla. Olemassa olevan taulukkopohjan etulehdeltä löytyy erillinen tietokenttä tapahtuneille muutoksille. Käytännössä tämä on kuitenkin harvoin tarpeellinen, sillä nykyisin muutokset esitetään erillisessä muutosluettelossa. Siten tämä kenttä poistetaan uudesta taulukosta ja sen sijalle siirretään luettelossa olleet tiedot liitännälaitteista ja valonlähteistä. Nämä tiedot esitetään usein lyhenteillä, jotka sijoitetaan valaisinsymbolien attribuutteihin. Ne koskevat kaikkia luettelossa esitettyjä valaisimia ja siten niiden esittäminen "Luettelo"-välilehdellä on epäkäytännöllistä. Lisäksi tämän siirron myötä käyttäjät voivat lisätä vapaamuotoisesti toivomiaan tietoja määrättyyn kohtaan. Näin koodissa vältytään lukuisien eri tilanteiden huomioimiselta ja muiden toimintojen toteutuksesta saadaan johdonmukaisempi.

Rev.	Muutos tai täydennys	Suunn.	Pvm.

K.osa/Kylä	Kortti/Tila	Tontti/Rn:o	Viranomaisen merkintöjä varten		
Rakennustoimenpide			Piirustuslaji		
			SÄHKÖPIIRUSTUS		
Rakennuskohteen nimi ja osoite			Piirustuksen sisältö	Mittakaavat	
KOHDE			VALAISINLUETTELO		
Osoite					
Osoite					
 www.rejlers.fi			Vapaalantie 2 B 01650 VANTAA Puh. +358 207 520 700 etunimi.sukunimi@rejlers.fi	Suunnitteluala Työn n:o	Piir n:o
			SÄH		
Pvm	Suunn.	Hyv.	Piir. Tiedosto		Muutos
			rejlers_valaisintaulukko.fi.xlsx		

Kuva 14. Käytössä oleva etulehti jäljittelee dwg-tasopiirustuksien otsikkotaulua, jonka nimiön yläpuolella esitetään muutoshistoria. Tämä uudistetaan täysin vastaamaan nykykäytäntöjä uudessa luettelossa.

 www.rejlers.fi Vapaalantie 2 B 01650 Vantaa, Finland Puh. +358 207 520 700 etunimi.sukunimi@rejlers.fi	Rakennuskohteen nimi ja osoite	Laatinut	Päiväys	Sivut
		Tekijä	Muutos pvm.	ETULEHTI
		Työ N:o	Piirustus N:o	Muutos
	Sisältö	Piirustuslaji		
	VALAISINLUETTELO	SÄHKÖPIIRUSTUS		

Kuva 15. Nykyinen etulehden otsikkotaulu korvataan luettelon kanssa yhdenmukaisella vaakasuoralla versiolla. Muutoksien dokumentointi tapahtuu erilliseen muutosluetteloon, joten revisiohistorian esitys on tarpeeton.

Luettelon otsikkotaulun tiedoissa ei ole muutostarpeita, sillä ne ovat edelleen ajankohdattaiset, mutta muita tietoja on päivitettävä nykyaikaisemmaksi. Liitäntälaitteissa ja valonlähteissä on useita eri tyyppisiä, joita ei käytetä enää nykyaikaisissa ratkaisuissa. On edelleen kohteita, joissa on käytössä vanhentunutta valaistustekniikkaa, mutta näiden käsittelyyn toteutetaan täysin uusi ratkaisu. Esimerkiksi magneettikuristimet ovat korvautuneet elektronisilla liitäntälaitteilla sekä elohopeahöyry- ja loistelamppujen tilalle on ke-

hittynyt puolijohdevalonlähteet (LED). Siten etulehdeltä löytyvät jatkossa vain asennustapamerkinnot sekä liitäntälaitetiedot kuvan 16 mukaisesti. Valonlähdeyhene on korvattu monipuolisemmat tarpeet kattavalla ”Huomautus”-sarakkeella, joka on osana luettelon valaisintietoja. Tämän avulla voidaan tuoda esille vanhat valonlähteet sekä liitäntälaitteet. Lisäksi saraketta voidaan käyttää useampaa valaisinta koskevan huomion osoittamiseen, kuten ”KH = Valaisin käyttäjähankinnassa”.

HUOMAUTUKSET	
ELLEI TOISIN MAINITTU:	
VALAISIMIEN IP-LUOKAT OLTAVA VÄHINTÄÄN TAULUKOSSA ESITETTYJEN ARVOJEN MUKAISET.	
KAIKKI LIITÄNTÄLAITTEET OVAT ELEKTRONISIA.	
KAIKKI VALAISIMET SISÄLTÄVÄT VALONLÄHTEEN (LED).	
KAIKKIEN VALAISIMIEN VÄRILÄMPÖTILA 4000K.	
HUOMAUTUKSET:	
KH = KÄYTTÄJÄHANKINTA	

ASENNUSTAPA	
K	= kattoasennus
S	= seinäasennus
U	= uppoasennus
V	= vaijeriasennus
R	= ripustusasennus
RK	= ripustuskiskoasennus
KK	= kosketinkiskoasennus
RP	= ripustusputkiasennus
KL	= kalustossa
VR	= vaijeriripustus
EK	= erikoiskiinnittimin
KR	= ketjuripustus
P	= pylväsasennus

LIITÄNTÄLAITE	
HF	= Elektroninen liitäntälaitte, On/Off
DALI	= DALI-yhteensopiva, säädettävä
DSI	= DSI & DALI-yhteensopiva, säädettävä
DMX	= DMX-yhteensopiva
1-10V	= 1-10V ohjattava

Kuva 16. Projektin kaikkia valaisimia koskevat tiedot ovat tästä lähtien etulehdellä. Huomautukset-kenttä on täysin käyttäjien muokattavissa ja attribuuttimerkinnoissa noudetaan taulukossa olevia arvoja.

Toimipisteessä suoritetun suullisen kyselyn perusteella on aputaulukon käyttö erittäin harvinaista. Aluksi välilehti poistettiin kokonaan, mutta myöhempien kehitysvaiheiden myötä se tuotiin takaisin ja korvattiin uudella käyttötarkoituksella. Jatkossa muutokset kirjataan nykyiseen aputaulukkuun ja näitä tietoja hyödynnetään muun muassa valaisinkohtaisen historian tarkastelussa. Tämä päätös mahdollistaa myös muutoshistorian esittämisen etulehdellä, mikäli tällainen ratkaisu halutaan jatkossa integroida taulukkuun, mutta toistaiseksi sitä käytetään vain sisäiseen dokumentointiin. Luettelossa on myös kaksi piilotettua välilehteä, jotka toimivat käytön tukena. ”Kopiopohja”-välilehdelle tuodaan usein leikepöydällä olevaa tai joukkojen sisältämää tietoa jatkokäsittelyä varten. Tämä osa ei sisällä mitään muotoiluja ja se tyhjennetään jokaisen koodiosuuden yhtey-

dessä, jolloin sen vaikutus tiedostokokoon pysyy hallittuna. Tämän rinnalla olevaa apu-
taulukkoa käytetään olennaisten tietojen säilyttämiseen. Esimerkiksi käyttäjän valitsema
XML-tiedostopolku tallennetaan sinne ja se voidaan aina hakea soluun tallennettuna ar-
vona koodia suorittaessa. Nämä piilotetut välilehdet eivät näy lainkaan käyttäjille ja nii-
den hyödyntämiseen palataan tarkemmin ohjelmointiosuudessa.

3.3 Ulkoasu ja visualisointi

Rakenteen ja funktionaalisen toteutuksen rinnalla graafisella ulkoasulla on suunnaton
merkitys tämän päivän sovelluskäytössä. Vanhanaikainen visualisointi antaa epäluotta-
muksellisen mielikuvan tuotteesta ja luo viimeistelemättömän käsityksen sovelluksesta.
Sen myötä tuotteen toiminnallisuus herättää helposti epäluuloja ja hyötyjen laajuutta pi-
detään arveluttavana. Oli siis tärkeää luoda miellyttävä ja moderni kokonaisuus, jossa
käyttöliittymät ovat jäsennellyt ja vaivattomasti tulkittavissa. Selkeyttä lisäävät muun mu-
assa solujen reunaviivojen rajaaminen, selvästi erotettava ja luettava fontti sekä havain-
nollinen ja ohjattu värien käyttö.

Taulukon alkuperäiset tavoitteet laajenivat kehitystyön aikana ja nyt luettelo tukee myös
muutosaineiston palauttamista sekä valaisinhistorian tarkastelua. Nämä sekä muut eril-
lään käytettävät ominaisuudet ovat lajiteltu eri kategoriaan käyttäjävalinnoista ja ne löy-
tyvät työkalukehyksen sisältä. Näin saadaan selkeä luokittelu taulukon toimivuuteen vai-
kuttavien valintojen ja erillistoimintojen välille. Taulukon päivittämiseen käytetty painike
löytyy kuvan 17 mukaisesti näiden valintojen läheisyydestä. Painike seuraa käyttäjäva-
lintojen taustalla olevaa visualisointia, mutta saa sisäisen heijastuksen erottuakseen
muista vaihtoehdoista. Yleinen väriteema on sinertävä yrityksen logosta johtuen. Viimei-
sen kehityssyklin aikana työkalujen väriteemaksi valittiin selkeyden vuoksi vihreä sekä
näiden laukaisemista varten taulukkoon sijoitettiin erillinen painonappi.



Kuva 17. Taulukon toiminnot on keskitetty oikeaan laitaan tulostusalueen ulkopuolelle. Solujen sisältämät kommentit, jotka ovat esitetty punaisilla kolmioilla, viestittävät lisätietoja käyttäjille valintojen vaikutuksista.

Kaikki uuden taulukon toiminnallisuudet löytyvät keskitetysti "Luettelo"-välilehden tulostusalueen ulkopuolelta. Tämä on selkeä sijoituspaikka, sillä muiden välilehtien käyttö on vähäistä ja ensisijaisesti tapahtuneet muutokset näkyvät käyttäjälle luettelon puolella. Käyttöohjeita ei kyetty tiedostokoon vuoksi sisällyttämään taulukkoon, joten avustukseksi sijoitetaan kehyksien ylänurkkauksiin kommentit, jotka esittävät tärkeimpiä ohjeista löytyviä tietoja. Ne ovat asetettu takana oleviin soluihin ja niiden avulla tiedot ovat saatavilla reaaliajassa suunnittelijoille. Valitettavasti vain kommenttien fonttia voidaan muokata ja muutoin niiden ulkoasu on kuvan 18 mukainen, mutta muun muassa Excel 365 tarjoaa miellyttävämmän ja kattavamman muistiinpano-ominaisuuden (11).

"Muutokset"-välilehti korostetaan yleistä tyyliä mukaillen sinisellä taustavärillä. Lisäksi sen otsikkotietojen vieressä on punaisella painottava teksti, joka ilmoittaa tämän osan taulukosta olevan vain sisäiseen dokumentointiin. Lisäksi koodilla on estetty tämän välilehden osallistuminen tulostettavaan aineistoon, jottei se päädy vahingossa osaksi pdf-tiedostoa manuaalisissa tulostustilanteissa (12). Tämä ei kuitenkaan estä tulostenotto-prosessia vaan käyttäjää tiedotetaan, että välilehti on poistettu tulostussarjasta. Sisäisen raportin vienti eri tiedostomuotoihin sekä paperille on edelleen mahdollista, mutta tämä onnistuu vain vastaavan välilehden ollessa osana tulostusvalintaa.

Tieto tarpeellisista taulukon tapahtumista välitetään käyttäjille viesti-ikkunoiden avulla ja tiedonkeruu suoritetaan käyttäjälomakkeilla. Kommenttien tavoin ei ponnahdusikkunan ulkoasuun voida juurikaan vaikuttaa ja se on aina identtinen kuvan 20 kanssa. Teknisten ominaisuuksien lisäksi voidaan joillain ikonivalinnoilla muuttaa viestin merkitystä visuaalisesti (13). Myös käyttäjälomakkeiden oletusgrafiikka on erittäin vanhanaikainen, mutta niiden visualisointiin löytyy kattavasti valintoja. Vaikka elementtikohtaisia muotoiluvaihtoehtoja on vain muutama, kollektiivinen muutos useampaan näistä luo erittäin modernin näkymän. Kuvassa 18 on esitetty uusi ulkoasu, jota kaikki taulukon käyttäjälomakkeet noudattavat. Sen oikealla puolella näkyy oletusgrafiikalla luotu lomake, joka oli käytössä kehitysprojektin alkuvaiheessa. Muotoilu olisi mahdollista suorittaa myös koodilla, mutta lomakkeiden vähäinen määrä ei tuonut tälle tarvetta. Lisäksi käytössä on valitun painikkeen korostus, joka on hyvin yleistä nykyisessä sovelluskäytössä (14). Sekä viesti-ikkunoiden että käyttäjälomakkeiden soveltamista käsitellään laajemmin ohjelmointiosuudessa.

Kuva 18. Kollektiivisilla väri- ja fonttimuutoksilla sekä efektejä vaihtamalla käyttäjälomakkeisiin saadaan modernimpi ulkoasu. Valitun painikkeen korostaminen visuaalisesti on yleistä nykyisessä sovelluskäytössä.

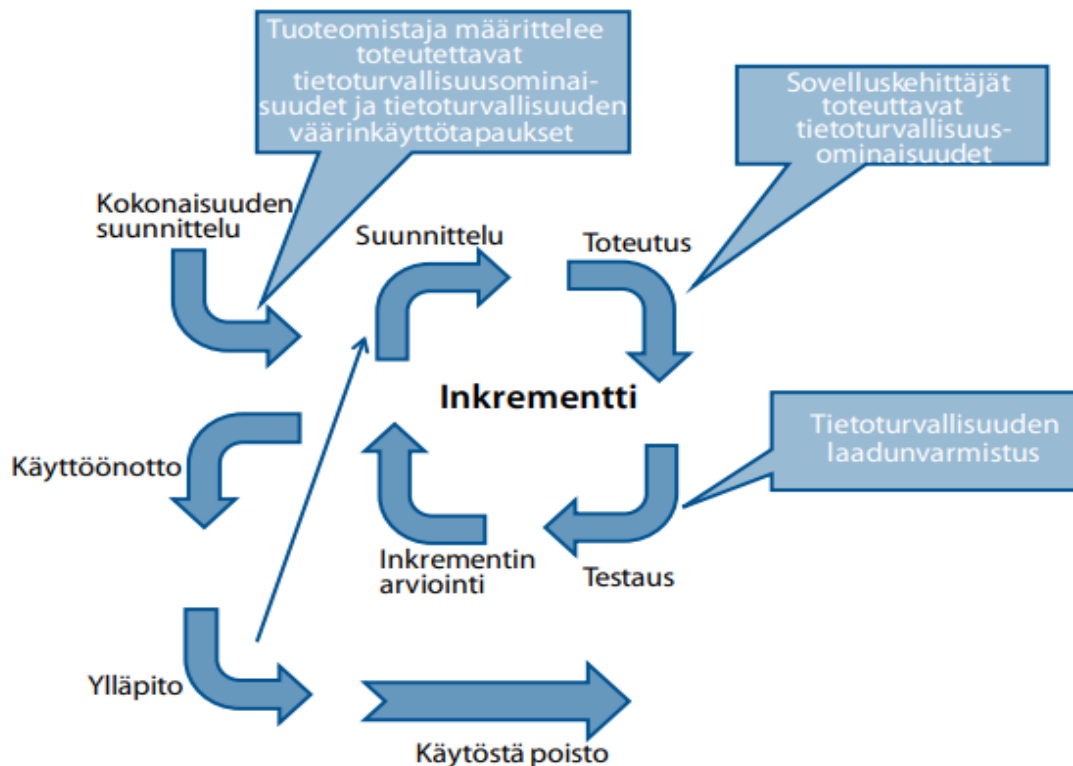
3.4 Automatisoinnin toteutus

3.4.1 Kehittämismenetelmän valinta

Tuotekehitys toteutetaan alusta saakka käyttäen ketterää sovelluksen kehittämismenetelmään. Finlex kuvailee virallisessa sovelluskehityksen turvallisuusohjeessaan kyseistä menetelmää seuraavin sanoin:

Ketterissä menetelmissä sovellusta kehitetään lyhyissä vaiheissa. Jokaiseen vaiheeseen valitaan toteutettavat ominaisuudet, jotka siinä vaiheessa (inkrementissä) toteutetaan. Vaiheita toistetaan, kunnes kaikki halutut ominaisuudet on toteutettu (kuva 2). Sovellus voidaan ottaa käyttöön, vaikka sovellukseen toteutettaisiinkin uusia ominaisuuksia seuraavassa vaiheessa (15).

Lisäksi kuvassa 19 on esitetty julkaisun graafinen esitys menettelytavasta. Lähtökohtaisesti määritetään, että taulukon on kyettävä automaattisesti tekemään tarvittavat toimenpiteet muutosaineiston laadinnan yhteydessä. Koska toiminnallisuuden laajuus on projekti alkuvaiheessa vielä epäselvä, tarjoaa valittu kehittämismenetelmä erinomaisen ympäristön eri työvaiheiden toteutukselle. Toinen yleinen sovelluskehityksen lähestymistapa on lineaarinen menetelmä, jossa kehityksen jokainen työvaihe saatetaan loppuun saakka. Tätä kehittämismenetelmää kutsutaan myös ”vesiputousmalliksi”, ja siitä on olemassa myös joustavampia, eri tasoilla iteroivia versioita. Oletuksena menetelmä vaatii projektin reunaehtojen mukaisen rajauksen, jota tämän työn alkuvaiheessa ei voida suorittaa perusteellisesti. Ketterällä menetelmävalinnalla pyritään parantamaan lopputuotteen toimintojen rajaamista sekä kykyä reagoida käyttäjätoiveisiin. Koska kyseessä on konsulttialan henkilön koordinoima yksilöllinen ohjelmistokehityshanke, on kehittämismenetelmien vaihtoehtoihin syytä palata, mikäli automatisoituja suunnitteluprosesseja kehitetään laajemmalti yrityksen käyttöön.



Kuva 19. Ketterässä kehittämismenetyssä sovelluksen eri työvaiheet jaetaan lyhyisiin iteraatioihin, jolloin sisältömuutokset käyvät läpi tarvittavat kehitystasot. Iteraation pituus on yleensä muutaman viikon mittainen (15).

Opinnäytetyön rajauksen myötä ei kehittämismenetyksiä käsitellä tässä työssä yksityiskohtaisemmin. Lisäksi tietyt menetelmän tasot, kuten ylläpito ja käytöstä poisto, on linjattu aihealueen ulkopuolelle. Kehityssyölien pituudeksi sovitaan toimeksiantajan kanssa viisi kuukautta, joka on menetelmän kannalta poikkeuksellisen pitkä. Pitkityllä iteraatiolla pyritään koodin rakenteellisen laadun parantamiseen sekä uusien ominaisuuksien kattavaan testaamiseen ennen julkaisua. Lisäksi tämä mahdollistaa kehittäjän perehtymisen eri ohjelmointiratkaisuihin.

3.4.2 Sovelluksen tarvekartoitus

VBA-projekti koostuu kahdesta perusosasta: taulukoihin tallennetuista komponenteista, kuten painonapeista, sekä projektin sisäisistä elementeistä, joita ovat esimerkiksi käyttäjälomakkeet. Mikäli koodi ei sisälly kumpaankaan edellä mainituista, on se esitettävä

moduulissa. Myös luokkien luominen on mahdollista, mutta opinnäytetyön aikarajan puitteissa, ei näitä hyödynnetty laadittavassa sovelluksessa. VBA-projekti on jatkuvasti muuttuva kokonaisuus ja siten sovelluksen tarpeiden laajentuessa on vaikea ylläpitää selkeää ja määrättyä lopputulosta. Työssä kehitettävän tuotteen lähtökohtina olivat

- taulukossa sijaitseva painonappi rajattua muokattavuutta varten
- taulukossa sijaitsevat valintaruudut käyttäjävalinnoille
- käyttäjälomakkeet eriävien lähtötietojen hakemiseen
- edellä mainittujen komponenttien tarvitsemat moduulit.

Näiden pohjalta sijoitettiin ensimmäiset valintaruudut sekä luettelon käyttöpainike taulukoon. Objekteille voidaan määrittää vapaamuotoinen nimi (16). Siten käyttäjävalinnat saivat etuliitteen ”opt_” (optio), ja myöhemmin lisättyjen työkalujen valintaruudut noudattavat myös tätä tyyliä. Useammalle päivytyspainikkeelle ei nähty tarvetta demo-version jälkeen ja siten käyttöpainikkeen nimeämiseen ei ollut tarvetta kiinnittää erityishuomiota. Tämän jälkeen siirryttiin jäsentelemään VBA-projektin sisäisiä elementtejä. Lähtötietoihin perustuen nähtiin tarve moduulille, joka

- toimii päämoduulina ja kutsuu muita aliohjelmia
- käsittelee kaikki tarpeelliset tekniset toteutukset
- sisältää kaikki toiminnallisuuden vaatimat muuttujat.

Moduulit voivat sisältää aliohjelmia (subroutine) tai funktioita (function). Näiden merkittävien ero on se, että funktioita käytetään arvon, kuten booleanin palauttamiseen (17). Esimerkiksi etsittävän arvon löytyessä joukosta, palauttaa funktio arvon tosi. Myöhemmässä vaiheessa havaitsin tarpeen myös erilliselle testausmoduulille, jonka käyttötarkoituksena oli luoda täysin eristetty ympäristö verkkolähteistä löytyvien ratkaisujen tarkasteluun. Viimeisimpänä luotiin moduulien ympärille toiminnallisuutta täydentävät käyttäjälomakkeet. Alustavasti suunnittelijoilta kerättäviä tietoja olivat

- muutosaineiston tunnus, päivämäärä sekä laatija
- luontivaiheessa tarpeelliset projektikohtaiset tiedot.

Myöhemmin lisätyt toiminnot kuitenkin laajensivat tätä osiota ja työkalulomakkeet eriteltiin normaalin käytön ympärille sidotuista lomakkeista etuliitteellä "Tyokalu_". Näille luotiin myös omat moduulinsa yhdenmukaista toteutusta varten. VBA-projektin lopullinen rakenne näkyy aiemmassa kuvassa 11.

3.4.3 Tuotteen kehitysvaiheet

Uuden luettelon kehittämistä ja toiminnallisuutta pohjustettiin työnantajan kanssa jo ennen yhteishakuprosessia. Tuloksien julkaisun myötä demo-version rakentaminen alkoi kesällä 2018 ja ensiesitys toimeksiantajalle pidettiin syksyllä. Tämän demonstraation tarkoituksena oli näyttää muutosaineiston koodilla toteutettua automatiikkaa. Vastaanotto oli sekä positiivinen että ennakkoluuloinen. Huolenaiheeksi nostettiin muun muassa projektien moninaiset tarpeet, koska taulukon sisältö oli tuolloin vielä muuttumaton. Alkuperäinen taulukko tuki rakennusosasta jaottelua, mutta tämä ominaisuus poistettiin yksimielisesti projektin alkuvaiheessa, jotta lopputuotteesta saataisiin laadukkaampi opinnäytetyön rajaamassa aikataulussa. Tällöin sovittiin myös taulukon kokonaisvaltaisesta lukituksesta, sillä luettelon sisältö oli vielä esityshetkellä osittain muokattavissa. Esimerkkikoodissa 1 ja 2 on poimintoja demo-version koodista, jotka ovat osittain käytössä myös nykyisessä versiossa.

```

For x = 1 To UBound(kt, 1)
    koplu.Range("A1", Cells(lr, 1)).AutoFilter field:=1, Criteria:=kt(x, 1)
    For y = 1 To UBound(rol) - 1
        rot(x, y) = koplu.Range(Cells(1, klc - 1), Cells(lr, klc - 1)). _
            SpecialCells(xlCellTypeVisible).Find _
            (what:=rol(y), lookat:=xlWhole, _
            MatchCase:=True).Offset(0, -1).Value
    Next y
    rot(x, y + 1) = Application.Sum _
        (Columns(klc - 2).SpecialCells(xlCellTypeVisible))
Next x

```

Esimerkkikoodi 1. Testikoodi joukkoihin (array) perustuvasta rakennusosien valaisinmäärien summauksesta, joka ei tällä hetkellä ole käytössä. Kopiovälilehden sisältö suodatetaan rakennusosan tunnuksen mukaisesti ja vastaava otsikko etsitään valaisinluettelosta yhteislukumäärän esittämistä varten.

```

On Error GoTo xmlvirhe:
Set xmltiedosto = Workbooks.OpenXML
  (Filename:=ThisWorkbook.Path & "\ " & valu.Name & ".xml", _
  LoadOption:=xlXmlLoadImportToList)

xmltiedosto.Sheets(1).Range("A1", Range("A1"). _
SpecialCells(xlCellTypeLastCell)).Copy

xmltiedosto.Close
On Error GoTo 0

```

Esimerkkikoodi 2. Pyydetään käyttäjää avaamaan xml-tiedosto lähtötietoja varten. Jos tiedosto valitaan, kopioidaan sen sisältö leikepöydälle ja jatketaan koodin suorittamista. Mikäli tiedostoa ei valita, edetään virheenkäsittelyyn, sillä "xmltiedosto" muuttuja on tyhjä ja siihen kohdistuva kopiointi palauttaa virheen.

Demoversion työvaihe edusti sovelluskehityksen ensimmäistä iteraatiota ja tähän perustuen oli seuraavan version

- kaikki solut oltava lukittuna ja käyttäjien sallittaisiin muokata vain valittuja elementtejä
- kyettävä toimimaan täysin yhden painonapin avulla, joka suorittaisi tarvittavat toimenpiteet tiettyjä kriteerejä kunnioittaen
- pystyttävä laatimaan muutosaineiston lisäksi urakkalaskentamateriaali
- joustettava tulevien käyttäjävalintojen mukaisesti.

Joukkomenetelmää (array) testattiin jo ensimmäisessä iteraatiossa ja se tarjosi huomattavia parannuksia luettelon suorituskykyyn verrattuna solupohjaiseen ratkaisuun. Siten seuraavan version tiedonkäsittely muutettiin perusteellisesti joukkopohjaiseksi, mikä vaati monen koodiosuuden täydellistä uudelleenkirjoitusta. Teknillisesti metodi käyttää tietokoneen keskusmuistia, joten suurempienkin tietomäärien hallinnointi tapahtuu hetkessä. Tämä muutos oli merkittävä taulukossa, sillä sen sisällön laajuutta ei voitu tarkalleen määrittää. Demoversion sarakelevydet ja otsikkotiedot olivat muokattavissa projektin tarpeita ajatellen, mutta tämä menettelytapa haluttiin muuttaa luotettavamman toimivuuden takaamiseksi. Siten kaikki taulukon muokattavuus lukittiin. Tämä päätös ei kuitenkaan vaikuttanut seuraavaan kehityssykliin, sillä koodi ei suorittanut vielä sopeutumiseen liittyviä muotoiluja.

Taulukon alustavana käyttäjävalintana lisättiin optio automaattisen tulosteenoton käytöstä poistamiselle, jonka avulla myös tulevat vaihtoehdot voitaisiin huomioida. Lisäksi demoversio sisälsi omat painikkeet sekä muutosaineiston että urakkalaskentamateriaa-

lin laatimiselle, mutta toinen näistä poistettiin ja valintaperusteet siirrettiin koodin käsiteltäväksi. Uusi versio otettiin heti testikäyttöön suljettuun ympäristöön, jonka käytöstä vastasi työn ohjaava projektipäällikkö. Lisäksi toimeksiantajan toiveesta lähdettiin perehtymään tuotteen tietoturvariskeihin välittömästi demo-version esityksen jälkeen. Eräitä virheetoimintoja korjattiin testaavan suunnittelijan toimesta, mutta uusia ominaisuuksia ei lisätty käytön aikana. Täten taulukko hyväksyttiin laajempaan julkiantoon ja kymmenhenkiselle testiryhmälle lähetettiin kutsu esittelyyn osallistumista varten. Esimerkkikoodeissa 3, 4 ja 5 ratkaisuja, joita sovelletaan myös käytössä olevassa versiossa ja niitä käsitellään laajemmin ohjelmointiosuudessa.

```
Function uusi(pos As Variant, vtv As Variant) As Boolean
    For i = LBound(vtv) To UBound(vtv)
        If vtv(i, 1) = pos Then
            uusi = True
            Exit Function
        End If
    Next i
    uusi = False
End Function
```

Esimerkkikoodi 3. Edellä mainittu funktio, joka palauttaa boolean arvon (uusi). Pos-muuttuja pitää sisällään lähtötietojen valaisinposition (ktv) jota verrataan olemassa olevien valaisimien array(vtv) arvoon. Jos arvot ovat yhtä suuret, asetetaan Boolean todeksi (uusi = true) ja lopetetaan funktio; muutoin "uusi" on epätosi.

```
If tapahtuivirhe = True Or opt_akt = 1 Then GoTo eitulostetta

tiedosto = Application.GetSaveAsFilename(InitialFileName:=tiedostonimi, _
    FileFilter:="PDF Files (*.pdf), *.pdf", Title:"Tallenna nimellä")

If tiedosto <> "False" Then

    ThisWorkbook.Sheets(printsheets).Select

    ActiveSheet.ExportAsFixedFormat _
        Type:=xlTypePDF, Filename:=tiedosto, _
        Quality:=xlQualityStandard, _
        IncludeDocProperties:=True, _
        IgnorePrintAreas:=False, _
        OpenAfterPublish:=False

End If
eitulostetta:
```

Esimerkkikoodi 4. Mikäli virhettä ei tapahtunut ja optio ei ole aktiivinen, pyydetään käyttäjää tallentamaan taulukko PDF-muodossa. GetSaveAsFilename tarjoaa ikkunan, jossa tiedostopolku voidaan valita. Tämä tallennetaan "tiedosto" muuttujaan ja ExportAsFixedFormat suorittaa tallennuksen. Jos käyttäjä sulkee valintaikkunan, saa muuttuja arvon "False" ja tiedostoa ei tallenneta.


```

For x = 1 To UBound(ktv, 1)
    If uusi(ktv(x, 1), vtv) Then GoTo eiuusi
    ktv(x, lc - 1) = "UUSI" & " (" & revisio
eiuusi:
Next x

```

Esimerkkikoodi 5. Yksi päämoduulin for-silmukka, joka kutsuu alafunktiota "uusi" tarkastamaan onko päivittyneiden lähtötietojen valaisinpositio olemassa vai ei. Mikäli positiotietoa ei löydy, täytetään array(ktv) arvo esimerkiksi tekstillä "UUSI (A". Jos funktio palauttaa toden (uusi = true), mennään koodiriville "eiuusi:"

Esitys testiryhmälle järjestettiin keväällä 2019. Osallistujiin kuului projektipäälliköitä sekä heidän kanssa runsaasti yhteistyötä tehneitä suunnittelijoita. Esityksen tarkoituksena oli testiprojektien lisäksi jakaa yleistä tietoutta kehitteillä olevasta tuotteesta. Vastaanotto oli yleisesti myönteinen, mutta eräät taulukon yksityiskohdat nostivat kiivasta keskustelua ja esille nostettiin muun muassa taulukon valaisintiedot. Usealla oli henkilökohtaisiin kokemuksiin perustuvia kriteerejä valaisinluettelon sisällön tarpeisiin, mikä on tyypillistä konsulttialalla. Testiryhmälle kuitenkin painotettiin, että sopeutuvuuden lisäys oli osana seuraavan kehityssyklin työlistaa. Tämän myötä tuote vastaanotettiin testikäyttöön meillä oleviin pienimuotoisiin hankkeisiin, kuten liiketilamuutoksiin.

Esille tuotiin myös ehdotuksia yleiseen muotoiluun sekä esitystapoihin liittyen ja nämä lisättiin seuraavan version tarveluettelon. Testiryhmä oli erityisen tyytyväinen uuden luettelon samankaltaisuuteen edeltäjänsä kanssa. Täten voidaan sanoa, että päätös ulkoasuvalinnasta oli oikea. Lisäksi yksikertaista siirtymisprosessia pidettiin etuna tulevissa käyttöönotoissa. Kaikki osallistujat ohjeistettiin uuden luettelon toimintaan, joka käynnistyi kopioimalla tarvittavat tiedostot yrityksen mallihakemistosta projektin työvaihekansioon. Nämä tiedostot sisälsivät taulukon lisäksi säännöt valaisimen lisäämisestä tietokantaan sekä ohjeistuksen luettelon soveltamiseen. Tämän jälkeen käyttöprosessi demonstritiin konkreettisesti suunnitteluohjelman avulla. Esitys kesti kokonaisuudessaan noin tunnin, josta viimeiset viisitoista minuuttia olivat yleistä keskustelua kehitettävästä tuotteesta.

Käyttäjävälintoina esiteltiin taulukon otsikkotietojen päivittäminen sekä aloitusvaiheessa että käyttöprosessin aikana, mikä oli taulukon lukituksen vuoksi tehtävä käyttäjälomakkeen kautta. Tällöin ehdotettiin ominaisuutta, joka mahdollistaisi aiempien muutoksien säilyttämisen tilanteita varten, joissa päivityksiä joudutaan tekemään poikkeuksellisen usein. Lisäksi toivottiin, että taulukko ilmoittaisi tapahtuneet muutokset välittömästi, sillä

se avustaisi laajempien projektien virheiden havaitsemista. Nämä molemmat lisättiin tehtävälistalle ja seuraava kokoontuminen sovittiin syksylle. Henkilökohtaisesti olin mielisäni testiryhmän aktiivisuudesta ja osallistumisesta esitykseen sekä kiinnostuksesta kehitteillä olevaan tuotteeseen. Useat olivat vakuuttuneita, että varsinkin sopeutuvuuden lisäyksen myötä luettelosta tulisi aidosti projekteissa hyödynnettävä työkalu.

Tämä motivoi seuraavan kehityssyklin päämäärien saavuttamisessa, joiden mukaan tulevan julkaisun tuli

- sopeutua automaattisesti xml-tiedostossa oleviin valaisintietoihin
- sisältää käyttäjävalinta aiempien muutoksien säilyttämiseen
- näyttää tapahtuneet muutokset välittömästi päivityksen jälkeen.

Vaikka ensimmäinen näistä olikin merkittävin uusi ominaisuus, oli se myös työtaakkana raskain. Siten oli tärkeämpää taata muiden toimintojen onnistunut suorittaminen ennen sopeutuvuuteen siirtymistä, sillä sarakkeiden poistaminen tai lisääminen vaikuttaisi merkittävästi muun koodin läpikäyntiin. Esimerkkikoodissa 6, 7 ja 8 on osuuksia tuolloisista toteutuksista, jotka ovat käytössä myös tällä hetkellä.

```
If uusiurakkakyllavaiei = False And tapahtuivirhe = False Then
  For i = 0 To UBound(r_post)
    rtxt = rtxt & vbNewLine & "-----"& _
      vbNewLine & r_post(i) & vbTab & r_mt(i) & vbTab & _
      r_tt(i) & vbTab & r_pt(i) & vbTab & r_ut(i)
  Next i
  MsgBox Join(r_ot, vbTab) & rtxt, vbOKOnly, "REPORT OF REVISION: " &
    revisio & " / DATE: " & paivam & " / CREATED BY: " & teki
End If
```

Esimerkkikoodi 6. Jos ei olla laatimassa urakkalaskenta-aineistoa ja virhettä ei ole tapahtunut, kootaan kaikki muutokset "rtxt" joukkoon. vbNewLine ja "---" toimivat positiokohtaisena rivinvaihtona sekä muissa joukoissa olevat arvot ketjutetaan käyttäen tabulaattoria (vbTab) erottimena. "r_or" sisältää raportin otsikkotiedot. Alla olevasta kuvasta 20 nähdään suoritettua koodin lopputulos.

POS	LKM	TIETO	POIS	UUSI
4	+20			
5	-30			
6	+50			
8	-15			
23	+20			
53				X

Kuva 20. Taulukon tekemistä muutoksista näytetään välittömästi ponnahdusikkuna, joka kertoo käyttäjälle poistuneet ja uudet valaisimet sekä päivittyneet lukumäärä- ja tietomuutokset.

With valu

```

lr = .Cells(Rows.Count, 1).End(xlUp).Row
If opt_apm = 1 Then GoTo alapoistarev

.Range(Cells(a - 1, lc - 1), Cells(lr, lc - 1)). _
    AutoFilter field:=1, Criteria1:="*POIS*"

On Error Resume Next
.Range(Cells(a, lc), Cells(lr, 1)). _
    SpecialCells(xlCellTypeVisible).Rows.EntireRow.Delete

    .ShowAllData
    .AutoFilterMode = False
On Error GoTo 0

lr = .Cells(Rows.Count, 1).End(xlUp).Row
.Range(Cells(a, lc - 1), Cells(lr, lc - 1)).ClearContents

alapoistarev:

vtt = .Range(Cells(a, 1), Cells(lr, lc))
.Range(Cells(a, 1), Cells(lr, 1)).Rows.EntireRow.Delete

End With

```

Esimerkkikoodi 7. Oletuksena taulukko (valu) laskee viimeisen rivi (lr) ja hävittää aluksi viime revisiossa poistuneet valaisimet suodatuksella. Tämän jälkeen tyhjennetään muutossarake, jolloin sisältö on valmis siirrettäväksi joukkoon (vtt). Tätä osuutta ei suoriteta, mikäli optio muutoksien säilyttämiselle on valittuna.

```

For x = 2 To UBound(vt, 2) - 6
  If kt(rev, x) <> vt(rev, x) Then

    rt(rev) = revisio & " "

    valu.Cells(a - 1 + rev, 1).Font.Bold = True
    valu.Cells(a - 1 + rev, x).Font.Bold = True
    valu.Cells(a - 1 + rev, lc - 1).Font.Bold = True

    r_tieto = True

  End If
Next x

```

Esimerkkikoodi 8. Tämä for-silmukka tarkistaa valaisimen muuttuneilta tiedoilta. Jos arvot ovat erisuuret, muutetaan vastaava revisiojoukon (rt) arvo, tehdään tarpeelliset muotoilut taulukkoon (valu) sekä boolean (r_tieto) muutetaan todeksi. Määrätarkastelun jälkeen kutsutaan erillistä alamoduulia, mikäli tietoja on muuttunut. Valaisinposition on tallennettu "rev" muuttujaan.

Välittömästi käyttöönoton jälkeen testiryhmä raportoi virheilmoituksia. Ensisijaisesti näiden alkuperänä oli vääränlainen raporttipohjan ja xml-tiedoston käsittely, minkä vuoksi luettelon käyttöohjeita voitiin pitää toistaiseksi puutteellisina. Lisäksi ongelmia ilmeni tilanteissa, joissa xml-tiedosto sisälsi vain yhden valaisinposition. Tämä johtui siitä, että luettelo rakensi automaattisesti uuden urakkalaskenta-aineiston, mikäli luettelossa oli vain yksi valaisinpositio. Koska vastaava tilanne on erittäin harvinainen, ei luettelon toimintaa muutettu tältä osin ja esitys uusista ominaisuuksista pidettiin suunnitellusti syksyllä 2019.

Paikalle kutsuttiin samainen testiryhmä, koska tuotteen ei ajateltu olleen riittävän laajassa testauksessa ennen esittelytilaisuutta. Joukolle painotettiin, etteivät käyttöohjeet olleet vielä ajantasaiset ja siten taulukon käyttöä demonstroitiin kauttaaltaan kiinnittäen huomiota testauksessa esille tulleisiin ongelmiin. Uudet ominaisuudet saivat myönteisen vastaanoton ja eräitä parannusvaihtoehtoja tuontiin esille. Erityisesti taulukon testaushetkellä laadittu tiedostopohjainen varmuuskopio aiheutti ennalta näkemättömiä rajoitteita. Tämä toimi ongelmitta toimipisteen sisäisessä käytössä, mutta aiheutti vaikeuksia etätyöskentelijöille. Siten vastaavalle ominaisuudelle toivottiin vaihtoehtoista ratkaisua. Alustavasti päätettiin, että se korvattaisiin taulukon sisäisellä kopiolla. Yleisesti sisäisten varmuuskopioiden ongelmana on kuitenkin lineaarinen tiedostokoon kasvaminen, mikä aiheuttaa käyttövaikeuksia etenkin laajemmissa hankkeissa. Tästä huolimatta testaamisen myötä tehtiin johtopäätös, ettei pelkkien arvojen säilyttämien aiheuta mittavaa muutosta tiedoston koossa ja ominaisuuden muuttaminen lisättiin seuraavaan kehityssykliin.

Lisäksi testauksen ympäristövalinnoissa havaittiin puutteita. Urakkalaskentamateriaalin laadinnassa ei testiryhmän mukaan ollut ongelmia, mutta testihankkeiden tarpeet huomioon ottaen, olivat muutosaineistojen rakentamistilaisuudet mitättömät. Täten käytössä olevia luetteloita pyydettiin päivitettäväksi suunnittelijoiden toimesta, jotta varmistuttaisiin virheettömästä toiminnallisuudesta. Muutoksien laadinta tapahtui ongelmitta, joten taulukon ydintoiminnot eivät tarvitse päivittämistä. Myös sopeutuvuutta testattiin vain yhdessä käyttöympäristössä ilman hankaluuksia. Siten oli todennäköistä, että tämä ominaisuus vaatisi vielä parannuksia myöhemmässä kehitysvaiheessa.

Varmuuskopioinnin muuttamisen myötä ehdotin kehittäjänä, että myös valaisinkohtaisen historian selailu olisi mahdollista toteuttaa taulukon sisäisesti tallennettuja tietoja hyödyntäen. Tämä ehdotus sai yksimielisen hyväksynnän ja lisäksi toivottiin, että varmuuskopioinnin sijaan voitaisiin mikä tahansa muutos palauttaa muotoiluineen taulukkoon. Siten seuraavan kehitysvaiheen jälkeen tulisi taulukon

- luoda automaattisesti palautettavissa oleva varmuuskopio laadittavasta muutosaineistosta
- sisältää työkalu, jolla voidaan tarkastella valaisinkohtaista historiaa.

Tämän syklin työlista oli poikkeuksellisen lyhyt, mutta silti toteutukseltaan kaikista vaativin. Pelkästään päämoduuli, joka oli vastuussa kaikista päivittämiseen liittyvistä teknillisistä toimenpiteistä, sisälsi tässä vaiheessa yli 500 riviä koodia. Täten ongelmia tuottivat muun muassa uusien toimintojen oikeanlainen toteutus ja sijoitus koodiosuuksissa. Useissa testeissä tietty suoritustapa oli täysin esteetön siihen saakka, kunnes eräs käyttäjävuorovaikutus rikkoi vähintään osan taulukon toiminnallisuudesta. Tämä johti laajaan testausprosessiin, jotta kokonaisuus toimisi ilman ongelmia. Kehitysvaihetta lähdettiin edistämään varmuuskopioinnin laatimisella ja palautustoiminnon toteutuksella, sillä valaisinhistorian tarkastelua ei voitu suorittaa ilman niitä. Lisäksi tietoisuus siitä, ettei historian selaaminen liittyisi muun koodin suorittamiseen, vaikutti päätöksentekoon.

Uusia ominaisuuksia ohjelmoidessani havaitsin tarpeen myös luotettavuutta parantavalle tarkastustoiminnolle. Näin taulukon päivittyminen kyettiin estämään, mikäli lähtötietojen ja luettelosisällön välillä ei havaittu eroavaisuuksia. Lisäksi huomasin eräitä virheetoimintoja, jotka olivat jääneet raportoimatta tai ne olivat ilmenneet muiden kehitysvaiheiden aikana. Näiden ohella tein eräitä ulkoasuun liittyviä muotoiluja ja viimeistellystä

versiosta pidettiin suljettu esitys toimialajohtajalle sekä ohjaavalle suunnittelu- ja projektipäällikölle. Heidän hyväksymänä viimeinen esittely järjestettiin huhtikuun alussa 2020 ja kutsu lähetettiin koko toimipisteen suunnitteluhenkilöstölle. Esimerkkikoodissa 9, 10 ja 11 on eräitä tämän kehityssyklin ohjelmointitoteutuksia, jotka ovat osana myös viimeistä tässä työssä laadittua versiota.

```
If eirevisiomuutosta = False Then
    valu.Activate
    lr = Cells(Rows.count, 1).End(xlUp).Row
    valu.Range(Cells(a, 1c), Cells(lr, 1)).SpecialCells _
        (xlCellTypeVisible).Rows.EntireRow.Delete
Else
    returnrev = Sheets(Sheets.count).Name
    previousrev = Sheets(Sheets.count - 1).Name
    Palautarevisio = True
End If
```

Esimerkkikoodi 9. Koodiosa, jota käytetään havaitsemattomien muutoksien aineistopalautukseen. Boolean on epätosi (eirevisiomuutosta), ellei muutoksia ole tapahtunut. Tuolloin voidaan poistaa luettelon sisältö, sillä ollaan suorittamassa työkalulla tehtävää palautusta. Muutoin tallennetaan viimeisin ja sitä edeltävä muutos muuttujiin (returnrev, previousrev) sekä muutetaan boolean arvo jatkokprosessointia varten (Palautarevisio).

```
Do
    Set hisnomark = .Range("A1", Cells(lr, 1)).Find(What:="EIVALITTU",
        LookAt:=xlWhole, MatchCase:=True, LookIn:=xlValues)

    If Not hisnomark Is Nothing Then
        hisnomark.EntireRow.Delete
    Else: Exit Do
End If
Loop
```

Esimerkkikoodi 10. Toinen Do-silmukka, jolla käydään läpi valitsemattomat muutokset. Jos muutosaineisto ei ole osana historiavalintaa, on sen tunnus korvattu tekstillä "EIVALITTU". Silmukka pyörii niin kauan kuin etsittävää arvoa ei löydy ja solun asetusta ei voida tehdä (hisnomark = Nothing).

```
valhisnotfound = True

Do
    If InStr(1, muutostiedot(y, 3), "POS " & hispos & ":", _
        vbTextCompare) <> 0 Then valhisnotfound = False
    y = y + 1
Loop While y <> UBound(muutostiedot, 1) + 1
```

```
If valhisnotfound = True And _
    InStr(1, muutostiedot(2, 3), "urakka", vbTextCompare) = 0 Then _
GoTo historiavirhe
```

Esimerkkikoodi 11. Do-silmukka, joka katsoo läpi kaikki ”Muutokset”-välilehden tapahtumat (muutostiedot). Jos valittu positio löytyy joukon arvonn sisältä, muutetaan boolean epätodeksi (valhisnotfound). Mikäli boolean pysyy totena ja urakka-laskenta-aineistoja on laadittu vain yksi, ei historian tarkastelua suoriteta, sillä muutoksia ei ole tapahtunut.

Ennen huhtikuun esitystä lisäsin pienten toiminnallisten muutosten ohella eston ”Muutokset”-välilehden tulostamiselle. Välilehden voi edelleen tallentaa vapaamuotoisesti yksittäisenä valintana, mutta sen tulostaminen muiden välilehtien kanssa on estetty, sillä kyseessä on vain sisäiseen dokumentointiin tarkoitettu aineisto. Tämän myötä luettelo oli valmis viimeiseen esitykseen, joka pidettiin huhtikuun toisella viikolla. Yhteisen päätöksen myötä tilaisuuteen kutsuttiin aiemmin testaukseen osallistunut ryhmä, sillä heidän tiedossaan oli uusia lähiaikoina käynnistyviä projekteja. Valitettavasti testaaminen oli jäänyt ennakoitua vähäisemmäksi etenkin muutosaineistojen laadinnan osalta. Siten taulukon toimintaa esiteltiin raportin luomisesta saakka, jotta ryhmälle palautuisi mieleen luettelon hyödyntäminen käyttöönnotosta lähtien.

Testiryhmä lähestyi alkuosiota tuttavallisesti ja sisäiseen varmuuskopiointiin oltiin tyytyväisiä, sillä aiemmassa toteutuksessa oli todettu ongelmia. Olin säilyttänyt aiemman testiversion, jolla kykenin demonstroimaan hallitusti kasvavaa tiedostokokoa, koska tämä huolenaihe nostettiin välittömästi esille. Lisäksi tarpeettoman päivittämisen esto sai myönteisen vastaanoton. Hieman ennen esityksen järjestämistä minulle oli ilmoitettu tämänkaltaisesta ongelmatilanteesta, joka oli johtanut varmuuskopion palautukseen. Ongelma ratkaistiin ja kyseinen ominaisuus tulisi olemaan osana seuraavaa versiota. Vaikka tiedoston palauttaminen on yksinkertaista, oli arvokasta tietää, että ongelmatilanteeseen oli kyetty ennakoimaan oikein ja estotoiminnolle nähtiin tarvetta jo ennen esitystä.

Muutosaineiston luomisen jälkeen siirryttiin työkalujen käsittelyyn. Revision laadinta oli tuttu prosessi koko testiryhmälle, eikä tähän toimintatapaan esitetty lisätoiveita. Aiemmin käyttäjävalinnoissa olleet otsikkotietojen päivitys sekä XML-tiedoston määrittäminen siirrettiin työkaluosiin. Päätöstä pidettiin loogisena, sillä näitä käytettäisiin todennäköisimmin erillisenä ominaisuutena, eikä taulukkoa haluttaisi päivittää näiden toimintojen rinnalla.

Ryhmä oli erittäin yllättynyt ja samalla tyytyväinen uusien työkalujen tarjoamista mahdollisuuksia.

Ennen vapaamuotoista loppukeskustelua ilmoitin keskittyväni seuraavaksi opinnäytetyöraportin kirjoittamiseen. Sovelluksen jatkokehitys jäi näin ollen tauolle, mutta käyttöohjeet olivat ajan tasalla ja koko työryhmän saatavilla. Testiryhmä ilmaisi kuitenkin taulukossa olleet kommentit paljon hyödyllisemmiksi, mikä oli luonnollista, sillä ohjeet oli laadittu ensisijaisesti uusille käyttäjille. Uudet testiympäristöt saatiin kartoitettua keskustelun aikana ja totesin, että uusien työkalujen myötä myös kiinnostus käyttöä kohtaan oli aiempaa vahvempi. Lopussa ehdotettiin myös muutamia kehitysideoita. Muun muassa automaattiseen tulostenottoon toivottiin parannuksia, sillä esimerkiksi muutosaineiston palautus ei esityshetkellä laukaissut tätä toimintoa. Kokonaisuudessaan taulukko arvioitiin täysin testikelpoiseksi myös laajempiin projektihankkeisiin ja sen seuraavat kehitystoimenpiteet olisivat osana tämän opinnäytetyön ulkopuolista iteraatiota.

3.5 Ohjelmointiosuus

3.5.1 Muuttujien määrittelyt

Muuttujat ovat olennainen osa jokaista ohjelmointikieltä, ja ne tuovat merkityksellisiä etuja koodin käsittelyyn. VBA-kielessä muuttujien tyyppimäärittely tapahtuu ”Dim”-sanalla (dimension), joka antaa muuttujalle niin sanotun ”ulottuvuuden” (18). Tämän lisäksi muuttujalle on asetettava arvo käyttämällä ”Set”-komentoa, mikäli kyseessä on referenssiobjekti. Tällaisia ovat muun muassa esimerkikoodissa 12 esitetyt Range- ja Worksheet-muuttujat, mutta esimerkiksi numeerisia arvoja ei tarvitse asettaa erikseen, sillä editori kykenee käsittelemään niitä automaattisesti.

```
Option Explicit 'all variables must be declared

Public valu As Worksheet
    Set valu = Taul02 'catalog sheet

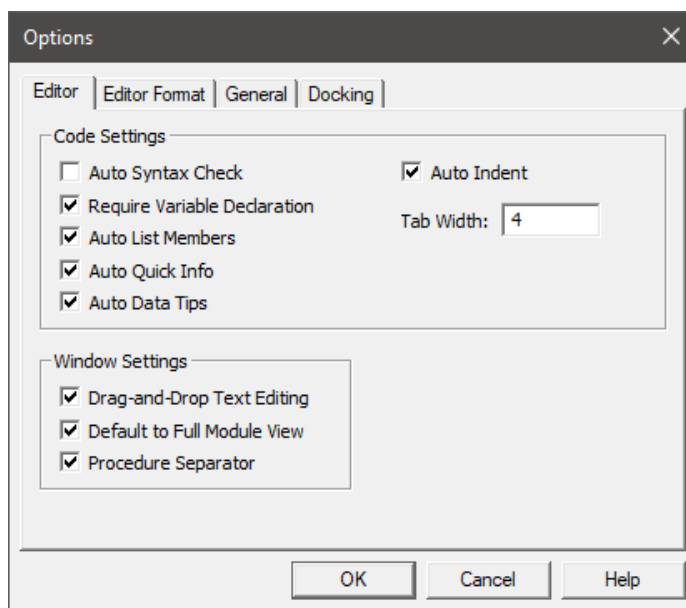
Public b, lc as long
    b = 11 'row index of lamp information labels
    lc = valu.Cells(b, Columns.count).End(xlToLeft).Column 'last column
```



```
Public otsikkotaulu As Range
    Set otsikkotaulu = valu.Range("A1", Cells(b, lc).Offset(0, -1))
```

Esimerkkikoodi 12. Option Explicit pakottaa kehittäjän määrittämään kaikki muuttujat. Koodissa esitettyjen muuttujien avulla saadaan asetettua taulukon otsikon alue. Kommentit erotetaan koodissa ’-merkillä.

Tyypimäärittäykseen voidaan käyttää myös ”Public”-komentoa. Tällöin siitä tulee julkinen ja jokainen moduuli kykenee hyödyntämään sitä. Tämä on oleellisesti tärkeää sovelluksen toteutuksen kannalta, sillä kaikki muuttujat sijaitsevat itsenäisessä moduulissa. Ilman muuttujien yksilöllistä määrittelyä, antaa VBA-kieli niille sisältöön perustuvan oletustyyppin, mikä voi johtaa ongelmatilanteisiin esimerkiksi vertailuja tehdessä. ”Option Explicit” pakottaa kehittäjän antamaan ”ulottuvuuden” jokaiselle muuttujalle ja ilmoittaa heti puutteellisista tilanteista. Tämän käytäntö on suositeltavaa ja se voidaan kirjoittaa jokaisen aliohjelman alkuun manuaalisesti tai aktivoimalla sen kuvan 21 mukaisesti ”Tools”-valikosta.



Kuva 21. ”Require Variable Declaration”-valintaruutu lisää automaattisesti ”Option Explicit”-tekstin kaikkiin koodiosuuksiin. Lisäksi muuhun editorin käyttäytymiseen voidaan vaikuttaa eri vaihtoehtoja muuttamalla.

Julkisten muuttujien ongelmana on se, että niiden arvot eivät tyhjene automaattisesti niin kuin moduulikohtaiset muuttujat tekevät. Siten sovelluksen luotettavan toimivuuden takaamiseksi on vastaavien muuttujien arvot nollattava manuaalisesti uudelleenkäyttöä varten. Tätä varten moduulissa on kaksi aliohjelmää. Toinen näistä suorittaa muuttujien

määritykset, toinen tyhjentämisen. Näin eri moduuleja ajaessa voidaan molempia kutsua tarpeen vaatiessa. Tyhjennys toteutetaan antamalla muuttujalle nolla-arvo esimerkikoodin 13 tavoin, joka on riippuvainen sen tyyppimäärityksestä.

```
Sub tyhjenna() 'reset all variables for continued usage

On Error Resume Next

Erase printsheets `printable sheet names
lc = Empty `used to count last column
valu = Empty `catalog worksheet
otsikkotaulu = Nothing `header range
xmlpolku = vbNullString `xml-filepath

On Error GoTo 0

End Sub
```

Esimerkkikoodi 13. Muuttujat-moduulin tyhjenna-aliohjelmassa muuttujien arvot nolataan tarkoituksenmukaisen toiminnallisuuden takaamiseksi. "On Error"-osat estävät virheet, jotka tapahtuvat tilanteissa, joissa muuttujalla ei ole koodiosuuden antamaa lähtöarvoa. Joukot (array) tyhjennetään Erase-komennolla.

3.5.2 Päämoduulit

Tuoreimmassa versiossa on yhteensä kahdeksan moduulia, joista kaksi on vastuussa ydintoimintojen suorittamisesta jättäen

- kaksi tukemaan päämoduulien toteutusta
- yhden käsittelemään kaikki muuttujat
- kaksi avustamaan työkalujen käyttöä
- yhden testaamista varten.

Moduulin aliohjelma tunnistetaan Sub-komennolla ja sille annetulla nimellä. Ohjelma päättyy End Sub -komentoon, mutta se voidaan päättää aiemmin Exit Sub -komennolla. Tätä käytetään muun muassa virhetilanteiden käsittelyyn, jolloin koodiosa sijoitetaan edellä mainittujen komentojen väliin (19). Goto-komennolla voidaan annetuilla ehdoilla hypätä koodiosuuksia, jotta vältetään tarpeettomien koodirivien suorittamiselta. Myös If-komentoa voidaan käyttää ehdolliseen toimintojen läpikäymiseen. Yksityiskohtaisen käsittelyn vuoksi päämoduulien esimerkikoodit 14-30 on eroteltu liitteeseen 1. Koodiosuudet on käsitelty johdonmukaisesti esitellen merkittävimmät päämoduulien. Periaatteet

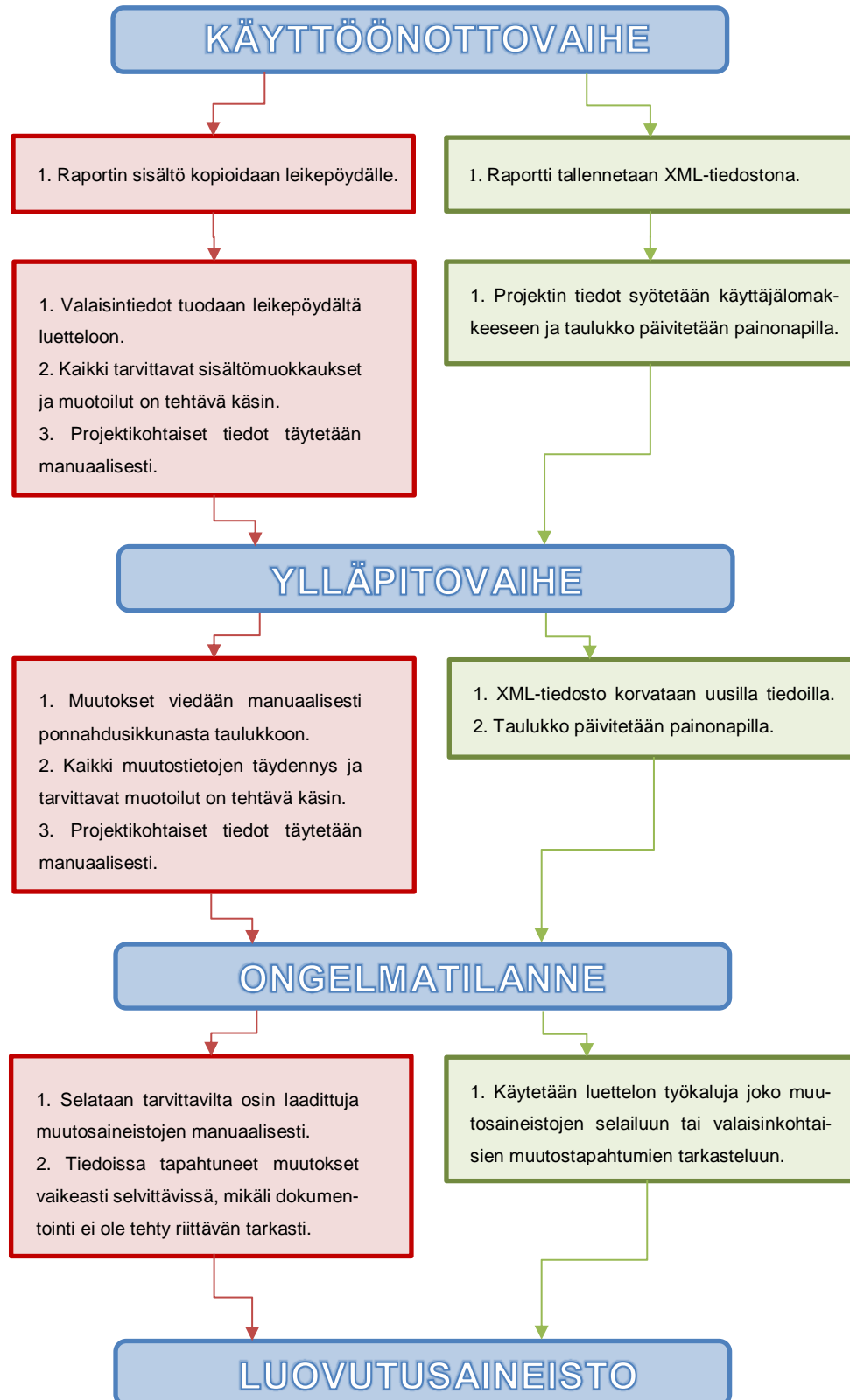
muuttujien ja testaamisen merkityksestä sovelluksen toteutuksessa on esitelty aiemmin eikä näihin liittyviä moduuleja tarkastella liitteen esimerkkikoodeissa.

3.5.3 Lisämoduulit

VBA-editori mahdollistaa vapaamuotoisen aliohjelmien ja moduulien luomisen. Soveltaessa tämä ei vaikuta koodin suoritukseen, sillä niitä kutsutaan yhtenevillä komennoilla sijainnista riippumatta (20). Siten laadittu rakenne palvelee enemmän kehittäjän mieltymyksiä ja käytäntöjä. Siitä huolimatta on suositeltavaa, että nimeäminen vastaa tarkoitusta sekä varmistaa, ettei samannimisiä moduuleja ja aliohjelmiä ole projektissa (21). Lisäksi yhteiseen käyttöön tarkoitetut koodiosuudet ovat hyödyllistä erottaa omiksi kokonaisuuksiksi. Tämän projektin kahta päämoduulia tukevat 4 lisämoduulia ja 5 käyttäjälomaketta, joista molemmista kaksi palvelee työkaluja. Myös näiden muodostama kokonaisuus käsitellään perusteellisesti liitteen 2 esimerkkikoodeilla 31-36. Viimeisenä esitellään työkalujen osalta valaisinkohtaisen historian tarkastelun koodisisältöä.

3.6 Tuotekehityksen tulokset

Viimeistely tuote vastaa sekä työn alkuperäisiä tavoitteita että kehitystyön aikana esille tulleita erikoisvaatimuksia. Testiryhmä osallistui aktiivisesti jokaiseen kehityssykliin ja useat alkuperäisessä toimintakartoituksessa huomioimatta jääneet ominaisuudet ovat osana nykyistä tuotetta käyttäjäpalautteen myötä. Kuvassa 22 on esitetty työvaiheittainen projektin suunnitteluprosessi, jossa verrataan automatisoidun valaisinluettelon vaikutuksia aiempiin käytäntöihin. Samat työtehtävät kohdistuvat edelleen molempiin käytötapoihin ja oikeassa laidassa vihreällä on näytetty uuden taulukon virtaviivaistama prosessikokonaisuus.



Kuva 22. Projektin työvaiheet ovat esitetyt sinisellä. Oikeassa laidassa vihreällä kuvaillaan automatisoidun taulukon virtaviivaistama suunnitteluprosessi.

Työn aikana taulukkoa sovellettiin pienemmissä hankkeissa, jotka vahvistivat luettelon virheettömät päätoiminnot. Viimeisessä testiryhmälle pidetyssä esityksessä taulukon todettiin olevan valmis laajempiin hankkeisiin, ja luettelo otetaan työn päätteeksi käyttöön kahteen satoja valaisimia sisältävään projektiin. Ainoana loppuvaiheen ongelmana esille tuotiin taulukon täydellinen lukitus, sillä käyttäjät kokivat pienimuotoisien täsmennyksien laadinnan käytännöllisemmäksi manuaalisen muokkauksen kautta. Suunnittelijana ja tuotteen kehittäjänä ymmärrän ryhmän esittämän näkökulman. On kuitenkin huomiotava, että uuden luettelon on tarkoituksenaan palvella jatkokehityksen myötä koko konsernin tarpeita. Tällä hetkellä projektitietokannat ja valaisinluettelot sisältävät usein manuaalisen käytön myötä ristiriitaisia tietoja, jotka tuottavat laadunhallinnallisia ongelmatilanteita etenkin resurssivaihdoksien yhteydessä. Tosin taulukko ei kykene estämään suunnitteluohjelmassa tapahtuneita virheitä, mutta lukumäärien laskentaan liittyvä virheherkyys pohjautuu osittain myös käsin tehtävään tietojen syöttöön. Siten automatisoidun luettelon kollektiivisesti tuomia etuja voidaan pitää merkityksellisempänä manuaalisen käytön aiheuttamien ongelmien rinnalla. Esimerkiksi jo suunnitelmissa tapahtuneita erehdyksiä pyritään hallinnoimaan taulukon antamalla välittömällä muutosraportilla.

Koska toimeksiantajayritys toimii osana kansainvälistä Rejlers-konsernia, sovelluksen hakutoimintoihin käytetyt tekstit on tallennettu muuttujiin ja koodiosuoksien kommentit on laadittu englanniksi. Nämä helpottavat taulukon kansainvälisen jatkokehityksen lisäksi tuotteen sopeuttamista muille kielille. Maiden välillä on kuitenkin poikkeavat käytännöt, mikä voi tuoda haasteita tuotteen saattamisessa laajempaan käyttöympäristöön. Täten esimerkiksi luettelopohja ei välttämättä vastaa muotoilullisesti tai sisällöllisesti kansainvälisen käytön tuomia haasteita. Siten näitä tarpeita on syytä tarkastella yksityiskohtaisemmin ennen uusien ominaisuuksien lisäämistä.

Opinnäytetyön aikana testiryhmä esitti myös muita käytössä olevia taulukoita, joita voitaisiin kehittää automatisoinnin avulla. Näistä merkittävimpana esimerkkinä on massaluettelo, jota on vaadittu eräissä projekteissa ja joka helpottaa hankintojen tekemistä. Siinä esitetään asennuskalusteiden määrät ja MagiCAD-lisäosan "selection area" -toiminto mahdollistaa niiden jaottelun esimerkiksi kerroksittain tai rakennusosittain raporttia laatiessa. Tätä varten on kuitenkin ensin syytä lisätä rakennusosittainen jaottelu kehitettyyn valaisinluetteloon, josta myöhemmin automatisoitava massaluettelo voidaan rakentaa.

Näin voidaan todeta, että nopean ja yksinkertaistetun päivitysprosessin tarjoamia hyötyjä toivotaan kehitettävän myös muihin suunnitteluprosesseihin.

Virallisen kehitystyön ohessa perehdyttiin osittain AutoCAD:n ohjelmointimahdollisuuksiin ja etenkin edellytyksiin liittää valaisinluettelo osaksi suunnitteluohjelman käyttöliittymään. Verkkolähteisiin tutustumisen sekä epävirallisen testaamisen myötä voitaisiin ohjelman työkalupaletteja (tool palette set) käyttää integroimiseen. Näin kyettäisiin yleistämään sekä luettelon että muiden laadunhallinnallisten dokumenttien käyttöä työympäristössä, sillä välittömällä saatavuudella on merkittävä vaikutus käyttöasteeseen. Myös projektin lopetusvaiheen viimeistely on vielä tarkasteltava yksityiskohtaisemmin tulevaisuudessa, sillä yksikään testaushanke ei päättynyt kehitystyön aikana. Normaalitylanteessa automatisoidun luettelon ominaisuudet poistettaisiin ja taulukko tallennettaisiin tyyppillisessä xlsx-muodossa luovutusaineistoa varten. Koska taulukkoa voidaan vaatia lähetettäväksi muokattavana, on tämä prosessi syytä automatisoida ja ohjelmointiosuus suoritetaan, kun yksi meneillään olevista asiakasprojekteista saatetaan päätökseen.

Kokonaisuudessa kaksi vuotta kestäneen kehitystyön tuloksiin ollaan tyytyväisiä sekä johdon että kehitystiimin osalta. Tuotteeseen liittyvät riskitekijät on kartoitettu ja toimiva sovellus on otettu käyttöön todellisiin asiakashakkeisiin. Opinnäytetyön jälkeisen käyttöjakson ja laajemman taustaselvityksen myötä voidaan lähteä resursoimaan seuraavia samankaltaisia kehitysprojekteja.

4 Yhteenveto

Opinnäytetyön tavoitteena oli kahden vuoden aikana luoda Rejlers Finland Oy:n toimipisteen käyttöön automatisoitu suunnitteluratkaisu. Automatisoitavaksi prosessiksi valittiin valaisinluettelo sekä sen ylläpito ja käyttöönotto, mikä pysyi sovelluksen päämääränä työn loppuun saakka. Työvaiheittain kartoitettujen kehityssykkien lopputuloksena muodostui täysin itsenäisesti painonapilla toimiva Excel-taulukkoon liitetty kokonaisuus.

Käyttäjystävällisyyttä mukaillen tuotekehityksessä jäljiteltiin toimipisteen aiempia toimintatapoja. Käyttäjät voivat edelleen vapaamuotoisesti lisätä projekti- ja valaisinkohtaisia tietoja sekä vaikuttaa taulukon toimintoihin valintaruutujen avulla. Lisäksi tuotteessa

on työkalut muutosaineiston palautukselle sekä yksittäisen valaisinhistorian tarkastelulle. Sovelluksen graafisella toteutuksella savutettiin ammattimainen käyttöliittymä ja viimeistely lopputulos. Taulukon lähtötiedot pohjautuvat MagiCAD-lisäosan raporttityökalulla luotuun tietokantapohjaiseen xml-tiedostoon. Tällaiset ratkaisut ovat olleet osana suunnitteluohjelmistoja jo vuosien ajan ja tietokantapohjaiset ratkaisut kasvattavat yhä suosiotaan alan ympäristöissä. Lisäksi valitun ohjelmointikielen käyttöasteen ja virallisen ohjelmistotalon tuen myötä tuotteen arvioidaan pysyvän keskeisenä vähintään seuraavat kymmenen vuotta.

Sovelluskehityksen rinnalla työssä käsiteltiin myös muita päivityksenalaisia luettelon osia sekä tietoturvallisuutta. Näiden tuoma tietoisuus heijastuu laadittuun ohjelmointiosuuteen, jonka tarkoituksena on palvella myös muita tämän työn pohjalta aloitettuja kehityshakkeita. Useat taulukon toteutukset kokivat tämän opinnäytetyövaiheen aikana merkittäviä parannuksia, ja tarkoituksena oli antaa seuraavalle kehittäjälle tarvittavat lähtötiedot uuden projektin perustamiseen. Ohjelmointikielen ja isäntäohjelmariippuvuuden tuomat heikkoudet on käsitelty työssä ja luettelon haavoittuvuuteen on neuvoteltu ratkaisua yrityksen IT-osaston kanssa.

Toimipisteen valaisinluettelon laadinta ja ylläpito suoritetaan edelleen pääsääntöisesti käyttäen vapaasti muokattavia taulukkopohjia joko mallihakemistosta tai aiemmista projekteista, mutta kehitetty sovellus on otettu käyttöön jo todellisiin asiakashankkeisiin. Automatisoidun suunnitteluprosessin tuomista laadunhallinnallisista ja kilpailukyvyllisistä eduista on keskusteltu Suomen yritysjohdon kanssa. Lisäksi testiryhmä sekä tämän ulkopuolella työskentelevät suunnittelijat ovat osoittaneet kiinnostusta kehitetystä tuotteesta.

Valaisinluettelo edustaa toimipisteen ensimmäistä erikseen automatisoitua prosessia. Sen avulla saavutettuihin tuloksiin ollaan tyytyväisiä yritystasolla, vaikka kehitettävää on edelleen sekä kehitetyssä tuotteessa että muissa suunnittelun osa-alueissa. Siten uuden taulukon päämääränä ei ole ainoastaan avustaa henkilöstöä suunnittelutehtävissä, vaan toimia myös käyttäjäkokemusten pohjalta lähtökohtana yrityksen sisäisille ohjelmistokehityshakkeille tulevaisuudessa. Seuraava vaihe, jota on valmisteltu aktiivisesti jo ennen työn päättymistä, onkin resursoitu jatkokehitys sekä jalkauttaminen laajempaan käyttöympäristöön.

Lähteet

- 1 Walhs, Anne. 2017. SEVEN REASONS WHY EXCEL IS STILL USED BY HALF A BILLION PEOPLE WORLDWIDE. Verkkoaineisto. Irish tech News. <<https://irishtechnews.ie/seven-reasons-why-excel-is-still-used-by-half-a-billion-people-worldwide>>. Luettu 17.9.2019.
- 2 Puls, Ken. 27 Microsoft Excel Experts Predict The Future Of Excel In Business Intelligence. Investintech.com Inc. <<https://investintech.com/resources/blog/archives/5718-experts-predict-the-future-of-excel-in-business-intelligence.html>>. Luettu 17.9.2019.
- 3 Support, Microsoft. Use Excel with earlier versions of Excel. Microsoft Corporation. Verkkoaineisto. Microsoft Corporation. <<https://support.office.com/en-us/article/use-excel-with-earlier-versions-of-excel-2fd9ffcb-6fce-485b-85af-fecfd651a5ac>>. Luettu 17.9.2019.
- 4 Walkenbach, John. Advantages and Disadvantages of Excel VBA. Verkkoaineisto. <<https://www.dummies.com/software/microsoft-office/excel/advantages-and-disadvantages-of-excel-vba/>>. Luettu 13.10.2019.
- 5 Uygar Y. 2011. Removing the password from a VBA project. Verkkoaineisto. Stackoverflow. <<https://stackoverflow.com/questions/272503/removing-the-password-from-a-vba-project/272664>>. Luettu 22.10.2019.
- 6 Support, Autodesk. 2019. VBA support for AutoCAD. Verkkoaineisto. <<https://knowledge.autodesk.com/support/autocad/learn-explore/caas/sfdcarticles/sfdcarticles/VBA-support-for-AutoCAD.html?st=vba>>. Luettu 23.10.2019.
- 7 Storiknow. 2013. What is the difference between “Form Controls” and “ActiveX Control” in Excel 2010. Verkkoaineisto. Stackoverflow. <<https://stackoverflow.com/questions/15455179/what-is-the-difference-between-form-controls-and-activex-control-in-excel-20>>. Luettu 12.4.2019
- 8 Better Solutions Limited. Shortcut Keys, Also known as Accelerator Keys or Hot Keys. Verkkoaineisto. <<https://bettersolutions.com/vba/visual-basic-editor/shortcut-keys.htm>>. Luettu 8.5.2019
- 9 Office Dev Center. 2018. Check or add an object library reference. Verkkoaineisto. Microsoft Corporation. <<https://docs.microsoft.com/en-us/office/vba/language/how-to/check-or-add-an-object-library-reference>>. Luettu 4.4.2019
- 10 Morgado, Flavio. 2016. Programming Excel with VBA: A Practical Real-World Guide. E-kirja. MetCat Finna.

- 11 Support, Microsoft Office. Insert comments and notes in Excel. Verkkoaineisto. Microsoft Corporation. <<https://support.office.com/en-us/article/insert-comments-and-notes-in-excel-bdccc9f5d-38e2-45b4-9a92-0b2b5c7bf6f8>>. Luettu 9.1.2020.
- 12 Chris. 2016. Preventing Users From Printing Your Excel Workbook Tabs With VBA. Verkkoaineisto. TheSpreadsheetGuru. <<https://www.thespreadsheetguru.com/blog/preventing-users-from-printing-excel-workbook-tabs-vba-disable-macro>>. Luettu 14.1.2020.
- 13 Bansal, Sumit. Excel VBA MsgBox [Message Box] – All You Need to Know. Verkkoaineisto. TrumpExcel. <<https://trumpexcel.com/vba-msgbox/>>. Luettu 18.1.2020.
- 14 Chris. 2019. How To Build A Modern-Looking VBA Userform. Verkkoaineisto. TheSpreadsheetGuru. <<https://www.thespreadsheetguru.com/blog/build-modern-vba-userforms>>. Luettu 20.1.2020.
- 15 Valtionhallinnon tietoturvallisuuden johtoryhmä. 2013. Sovelluskehityksen tietoturvaohje. Verkkoaineisto. Valtiovarainministeriö. <http://www.finlex.fi/data/normit/41655/VAHTI_1_Sovelluskehityksen_tietoturvaohje_NETTI.pdf>. Luettu 2.12.2019.
- 16 Office for Mere Mortals. 2019. Named Ranges and Objects in the Excel Name Box. Verkkoaineisto. Office Watch. <<https://office-watch.com/2019/named-ranges-and-objects-in-the-excel-name-box/>>. Luettu 1.4.2019
- 17 Free Excel Help. Function and Sub. Verkkoaineisto. Easy Excel. <<https://www.excel-easy.com/vba/function-sub.html>>. Luettu 12.2.2019
- 18 Kelly, Paul. 2018. VBA Dim – A Complete Guide. Verkkoaineisto. Excel Macro Mastery. <<https://excelmacromastery.com/vba-dim/>>. Luettu 7.6.2019
- 19 Kelly, Paul. 2019. VBA Error Handling – A Complete Guide. Verkkoaineisto. Excel Macro Mastery. <<https://excelmacromastery.com/vba-error-handling/>>. Luettu 7.6.2019
- 20 Office Dev Center. 2019. Calling Sub and Function procedures. Verkkoaineisto. Microsoft Corporation. <<https://docs.microsoft.com/en-us/office/vba/language/concepts/getting-started/calling-sub-and-function-procedures>>. Luettu 10.5.2019
- 21 Ekin, Well-known Member. 2014. Store Macros in 1 module or in multiple modules. Verkkoaineisto. MrExcel. <<https://www.mrexcel.com/board/threads/store-mcros-in-1-module-or-in-multiple-modules.81551/>>. Luettu 14.3.2020.

Esimerkkikoodeja päämoduulien toteutuksista

```
Sub paivita() 'main module launched by push button
```

```
Muuttujat.hae 'seek all necessary variables
```

```
'disable all features that hinder functionality
```

```
Application.ScreenUpdating = False
```

```
Application.DisplayStatusBar = False
```

```
Application.EnableEvents = False
```

```
Application.DisplayAlerts = False
```

Esimerkkikoodi 14. Taulukko-moduulin paivita-aliohjelma laukaistaan luettelon painonapista. Aluksi haetaan kaikki tarvittavat muuttujat, jonka jälkeen poistetaan käytöstä prosessia hidastavat Excel-toiminnot. Lopuksi poistetaan taulukon suojaus.

```
'check if any of the tools which require multiple revisions is active
```

```
If opt_nvh = True Or opt_per = True Then
```

```
  If Sheets.count <= 6 Then GoTo eityokalukaytto
```

```
  If opt_nvh = True Then 'check if user wants to review lamp history
```

```
    'change boolean, disable checkbox and show userform
```

```
    valaisinhistoria = True
```

```
    btn_nvh.Object.Value = False
```

```
    Tyokalu_muutoshistoria.Show
```

```
    GoTo tyokalukaytetty
```

```
  Else 'user wants to restore previous revision
```

```
    'change boolean, disable checkbox and show userform
```

```
    Palautarevisio = True
```

```
    btn_per.Object.Value = False
```

```
    Tyokalu_palautarevisio.Show
```

```
    GoTo tyokalukaytetty
```

```
  End If
```

```
End If
```

Esimerkkikoodi 15. Tarkastetaan työkalut, jotka tarvitsevat useamman muutosaineiston. Mikäli aineistoja ei ole riittävästi, siirrytään virheenkäsitelyyn. Muutoin suoritetaan tarvittava koodiosuus ja hypätään muiden työkaluoptioiden yli.

```
tyokalukaytetty: 'skip to this line after tool is used
```

```
Jaotta.paivita 'call module to update the catalog
```

```
'recall main module if there were no changes
```

```
If eirevisiomuutosta = True Then
```

```
  eirevisiomuutosta = False
```

```
  'this needs to be set false for reuse
```

```
  uusinrevisiovarmistettu = False
```

```
  Jaotta.paivita
```

```
End If
```

Esimerkkikoodi 16. Kaikkien työkalujen jälkeen kutsutaan päivitykseen käytettävän moduulin paivita-aliohjelmaa. Mikäli lähtötiedoissa ei ole havaittu muutoksia, tätä aliohjelmaan hyödynnetään uudestaan, jotta vanha aineisto voidaan palauttaa takaisin taulukkoon.

```
'if error occurred importing data from xml-file
If tapahtuixmlvirhe = True Then GoTo xmlvirhe

'if changes were discovered or any of the tools were used
If Palautarevisio = True Or Palautaurakka = True Or _
    otsikkopaivitys = True Or xmlmaaritys = True Or _
    valaisinhistoria = True Or eirevisiomuutosta = True _
    Then eitaulukkotoimintoja = True

'if option is disabled and script proceeded normally, show report window
If uusiurakkakyylla = False And eitaulukkotoimintoja = False And _
    opt_anr = False Then Call taulukko.report

'if option is disabled or utility boolean is false, ask for printout
If eitaulukkotoimintoja = False And opt_akt = False _
    Then Call taulukko.printout

'if utility boolean is false, created an entry into log sheet
If eitaulukkotoimintoja = False _
    Then Call taulukko.logentry

'Call subroutine to default workbook
Call taulukko.reset

Exit Sub 'if script proceeds normally, end the subroutine here
```

Esimerkkikoodi 17. Päivityksen jälkeen käsitellään raportin näyttämisen, automaattinen tuloste-
teenotto ja lokin kirjaus. Näiden toimintojen ohitusta varten muutetaan kol-
lektiivinen boolean todeksi, mikäli ehdot täyttyvät. Loppuosa suoritetaan nor-
maalisti kutsuen muita moduulin aliohjelmia tai siirrytään virheenkäsitelyyn.

```
xmlvirhe: 'skip to this line if error occurred with xml-file
taulukko.reset 'set workbook back to default and show error message
MsgBox "XML-tiedoston sisältö ei ole yhteensopiva!", _
    vbOKOnly + vbExclamation, "Virhe!"
Exit Sub
```

Esimerkkikoodi 18. Esimerkki virheenkäsitelystä. Jos lähtötietojen käsittelyssä on ongelmia,
palautetaan taulukon oletusasetukset ja näytetään virheilmoitus.

```
Sub printout() 'subroutine to create automatic PDF printout

tiedosto = Application.GetSaveAsFilename(InitialFileName:=tiedostonimi, _
FileFilter:="PDF Files (*.pdf), *.pdf", Title:="Tallenna nimellä") 'filepath

If Not tiedosto = "False" Then 'if filepath is selected, create file
    ThisWorkbook.Sheets(printsheets).Select
    ActiveSheet.ExportAsFixedFormat Type:=xlTypePDF, Filename:=tiedosto, _
    Quality:=xlQualityStandard, IncludeDocProperties:=True, _
    IgnorePrintAreas:=False, OpenAfterPublish:=False 'actual PDF file
End If

End Sub
```

Esimerkkikoodi 19. Yksi päämoduulin aliohjelmista, jota käytetään PDF-tulosteen ottoon. En-
simmäinen osia on vain polun määrittelyyn, jonka jälkeen varsinaisen tie-
doston tallennus tapahtuu.

```
Sub paivita() 'subroutine to update undivided catalog

'if tools which require no materials were used
If xmlmaaritys = True Or otsikkopaivitys = True Or _
    valaisinhistoria = True Then GoTo tyokalukaytetty
```

Esimerkkikoodi 20. Moduulin alussa tarkastetaan käytetyt työkalut.

```
On Error GoTo xmlvirhe: 'on error, proceed to error handling
Set xmldata = Workbooks.OpenXML(FileName:=xmlpolku, _
    LoadOption:=xlXmlLoadImportToList) 'XML-file
'count last column and store headers into array
'error occurs here if xml-file is not selected
klc = Cells(1, Columns.Count).End(xlToLeft).Column
kkk = Application.Transpose(Application.Transpose( _
    ActiveWorkbook.ActiveSheet.Range("A1", Cells(1, _ klc)).Value))

'if first XML header is not "POS", user selected wrong file
If Not txtcomp(kkk(1), "POS") Then GoTo xmlvirhe

'compare headers only if first contract material is build
lr = valu.Cells(Rows.Count, 1).End(xlUp).Row
If lr = a Then Call Jaotta.tarkastotsikot

'Copy XML data, including the headers
xmldata.Activate 'and close XML-file
xmldata.Sheets(1).Range("A1", _
    Range("A1").SpecialCells(xlCellTypeLastCell)).Copy
xmldata.Close
valu.Activate
On Error GoTo 0
```

Esimerkkikoodi 21. Tämän jälkeen aukaistaan xml-tiedosto, josta otsikon tiedot viedään joukkoon vertailua varten, mikäli ollaan laatimassa ensimmäistä materiaalia. Lopuksi tiedoston sisältö kopioidaan leikepöydälle.

```
If lr = a Then 'if true, build contract automatically
    uusiurakkakylla = True 'and empty catalog sheet
    valu.Range(Cells(a, 1), Cells(lr, 1)). _
        Rows.EntireRow.Delete
    GoTo uusiurakkamateriaali
Else 'setup for revision
    uusiurakkakylla = False
    muutos.Show
    'user may leave revision mark empty
    If revisio = "" Then
        'if so, rebuild contract material
        uusiurakkakylla = True
        valu.Range(Cells(a, 1), Cells(lr, 1)). _
            Rows.EntireRow.Delete
        GoTo uusiurakkamateriaali
    End If
End If
```

Esimerkkikoodi 22. Urakkalaskenta-aineisto rakennetaan automaattisesti, jos taulukossa on vain yksi valaisin. Muutoin näytetään muutoslomake. Lomakkeen revisio-kenttä voidaan jättää tyhjäksi, kun urakkalaskenta halutaan laatia uudelleen.

```

If Palautarevisio = True Then
'if previous revision was restored, store sheet data into array
'NOTE: klc is defined as equal to lc when boolean is true
Sheets(restorerev).Activate
lr = Cells(Rows.count, 1).End(xlUp).Row 'count last row
ktv = .Range("A1", Cells(lr, klc)) 'copy sheet data
.Activate 'activate copy sheet

```

Esimerkkikoodi 23. Jos ollaan palauttamassa aiempaa muutosaineistoa, viedään valitun välilehden sisältö ktv-joukkoon. Vaihtoehtoisesti lähtötietoina toimii xml-tiedosto.

```

'seperate new values with comma
vaihdettava = Array("TEHO")
lyhenne = Array("W")
etuliite = Array("N")

For i = LBound(vaihdettava) To UBound(vaihdettava)
'search for the titles from all headers
For x = 1 To UBound(kkk, 1) - 3
  If InStr(1, kkk(x), vaihdettava(i), vbTextCompare) <> 0 Then
    For y = 1 To UBound(ktv, 1)
      If IsEmpty(ktv(y, x)) = False Then
        'check for prefix or suffix, N = suffix
        If etuliite(i) = "N" Then
          ktv(y, x) = ktv(y, x) & lyhenne(i)
        Else
          ktv(y, x) = lyhenne(i) & ktv(y, x)
        End if
      End If
    Next y
  End If
Next x
Next i

```

Esimerkkikoodi 24. Joihinkin suunnitteluohjelman attribuutteihin ei voida laittaa numeroita ja kirjaimia, mutta niiden yhdistelmiä halutaan esittää taulukossa. Koodi etsii teho-otsikon (TEHO) ja lisää watin (W) tunnuksen sen arvoihin. Aliohjelma toimii dynaamisesti ja uusia otsikoita voidaan lisätä tarpeen mukaan.

```

.UsedRange.Clear 'insert ktv first, then vtv
.Range(Cells(1, 1), Cells(UBound(ktv, 1), UBound(ktv, 2))) = ktv
.Range(Cells(UBound(ktv, 1) + 1, 1), Cells(UBound(ktv, 1) _
  + UBound(vtv, 1), UBound(ktv, 2))) = vtv 'below ktv
With koplus.Range(Cells(1, 1), Cells(UBound(ktv, 1) _
  + UBound(vtv, 1), UBound(ktv, 2))) 'whole content
'sort all the lamps and remove duplicates
.Sort Key1:=Range("A1"), order1:=xlAscending, Header:=xlNo
.RemoveDuplicates Columns:=1, Header:=xlNo
End With
'recount last row and store data into kt array
lr = Cells(Rows.count, 1).End(xlUp).Row
kt = .Range("A1", Cells(lr, lc)) 'kt=ktv+vtv

```

Esimerkkikoodi 25. Lukumäärämuutoksien tarkastelua varten on joukot rakennettava uudelleen, kun uudet ja poistuneet valaisimet on käsitelty. Molempien näistä on oltava yhteneviä vertailun vuoksi. Tämä saavutetaan valitsemalla niiden järjestys ja käyttämällä Sort- ja RemoveDuplicates-komentoja.

```

x = x + 3 'adjust extra columns of additional information
'check if values in both arrays are the same
If vt(rev, x) = kt(rev, x) Then GoTo skiplkm
If vt(rev, x) < kt(rev, x) Then
'if value in vt (previous) is lower than kt, total amount increased
rt(rev) = revisio & " " & "+" & kt(rev, x) - vt(rev, x) 'calculate
valu.Rows(a - 1 + rev).EntireRow.Font.Bold = True 'bold entire row
r_lkm = True 'change boolean "true" for reporting

Else 'if amount wasn't increased, it was decreased
rt(rev) = revisio & " " & "-" & vt(rev, x) - kt(rev, x) 'calculate
valu.Rows(a - 1 + rev).EntireRow.Font.Bold = True 'bold entire row
r_lkm = True 'change boolean "true" for reporting

End If
skiplkm: 'if either specifications or amount changed
'call subroutine to update report
If r_tieto = True Or r_lkm = True Then
Raportti.paivita
End If
Next rev 'proceed to next lamp

```

Esimerkkikoodi 26. Muutosaineistojen lukumäärätarkastelu tehdään viimeisenä. Jos arvot ovat yhtä suuret, ei tarkistukselle ole tarvetta. Muuten suoritetaan laskutoiminto sekä tarvittavat muokkaukset. Lopuksi kutsutaan aliohjelmää raportin päivittämistä varten. Valaisinposition on tallennettu muuttujaan "rev".

```

'count total column width of additional information
For Each lisatietosakare In valu.Range _
    (valu.Cells(a, lc - 6), valu.Cells(a, lc - 3))
    aicw = aicw + lisatietosakare.ColumnWidth
Next lisatietosakare
valu.Columns(lc + 15).ColumnWidth _
    = aicw + (4 * 0.5) 'add frame width, 0.5 per column

'NOTE because merged cells cannot be autofitted
'copy additional information to help column
valu.Range(valu.Cells(a, lc - 6), valu.Cells(lr, lc - 6)).Copy
valu.Cells(a, lc + 15).PasteSpecial Paste:=xlPasteValues
'format help column to imitate catalog's merged cells
valu.Range(valu.Cells(a, lc + 15), valu.Cells(lr, lc + 15)).WrapText = True
valu.Range(valu.Cells(a, lc + 15), valu.Cells(lr, lc + 15)).Rows.AutoFit
valu.Columns(lc + 15).EntireColumn.Hidden = True 'hide help column
'merge catalog's columnd data per row and warp text
valu.Range(valu.Cells(a, lc - 6), valu.Cells(lr, lc - 3)).Merge (True)
.Range(valu.Cells(a, 1), valu.Cells(lr, lc - 1)).WrapText = True

```

Esimerkkikoodi 27. Excel ei tue yhdistettyjen solujen mukaista autofit-komennon muotoilua. Siinä on käytettävä apusaraketta, jonka myötä rivikorkeus saadaan oikein, kun tekstinrivitys on käytössä. Sarakkeen leveyttä määrää aicw-muuttuja.

```
'format all links into hyperlinks based on their cell value
valu.Range(valu.Cells(a, lc), valu.Cells(lr, lc))._
    VerticalAlignment = xlTop 'top all hyperlinks vertically
On Error Resume Next 'lamps with no link will give error
For Each hl In valu.Range(valu.Cells(a, lc), valu.Cells(lr, lc))
    hl.Parent.Hyperlinks.Add anchor:=hl, _
        Address:=hl.Value, TextToDisplay:=hl.Value
    'also format additional information to the left
    hl.Offset(0, -4).HorizontalAlignment = xlLeft
Next hl
On Error GoTo 0 'return errors

'Adjust print are the end of catalog sheet
valu.PageSetup.PrintArea = _
    Range(valu.Cells(1, 1), valu.Cells(lr, lc - 1)).Address
```

Esimerkkikoodi 28. Suunnitteluohjelmasta tuodut linkit on muokattava aktivoitaviksi hyperlinkeiksi solukohtaisesti käyttäen niissä olevaa arvoa. Tämä on ainoa taulukossa toimiva silmukka, sillä muut ratkaisut ovat yleisesti tehokkaampia. Muotoilujen loppuksi asetetaan tulostusalue luettelon loppuun.

```
Else 'apply form values from Muutos userform into both headers
[otsikkotaulu].Find(What:=ot_mpv, LookAt:=xlWhole, _
    MatchCase:=True).Offset(1, 0).Value = paivam 'revision date
etulu.Range([etutaulu].Find(What:=ot_mpv, LookAt:=xlWhole, _
    MatchCase:=True).Offset(1, 0).Address).Value = paivam
[otsikkotaulu].Find(What:=ot_tekija, LookAt:=xlWhole, _
    MatchCase:=True).Offset(1, 0).Value = teki 'revision creator
etulu.Range([etutaulu].Find(What:=ot_tekija, LookAt:=xlWhole, _
    MatchCase:=True).Offset(1, 0).Address).Value = teki
[otsikkotaulu].Find(What:=ot_muutos, LookAt:=xlWhole, _
    MatchCase:=True).Offset(1, 0).Value = revisio 'revision mark
etulu.Range([etutaulu].Find(What:=ot_muutos, LookAt:=xlWhole, _
    MatchCase:=True).Offset(1, 0).Address).Value = revisio
```

Esimerkkikoodi 29. Revisiota luodessa tuodaan muutoslomakkeen arvot etulehdelle ja luetteloon. Etsittävät arvot on tallennettu muuttujiin joustavuutta varten. Offset-komento sijoittaa lomakkeen arvon oikeaan soluun.

```
'check if backup sheet already exists, if it does, delete it
For x = Sheets.count To 5 Step -1
    If InStr(1, Sheets(x).Name, "Urakka " & paivam, _
        vbTextCompare) = 1 Or _
        InStr(1, Sheets(x).Name, "Muutos " & paivam, _
        vbTextCompare) = 1 Then
        Sheets(x).Visible = True
        Sheets(x).Delete
    Exit For
End If
Next x
```

Esimerkkikoodi 30. Yksilöllisiä varmuuskopioita säilytetään vain yksi. Tästä johtuen tarkastetaan kaikki välilehdet nimen perusteella. Välilehdet ovat piilotettu koodilla, joten niiden näkyvyys on asetettava todeksi ennen poistamista, muuten koodisuus palauttaa virheen

Esimerkkikoodeja lisämoduulien toteutuksista

```
Private Sub UserForm_Initialize()

'incrementally increase of alphabetic R
'is done based on the value of revi
If Not revi = "" Or Not IsNumeric(revi) = True Then
  'if total lenght is 1 and isn't Z (from A to Y)
  If Len(revi) = 1 And revi <> "Z" Then
    'use next alphabetic character (e.g. F -> G)
    tunnus.Value = Chr(Asc(revi) + 1)
    'if total lenght is 1 and value is Z
    ElseIf Len(revi) = 1 And revi = "Z" Then
      'use AA to define new revision cycle
      tunnus.Value = "AA"
      'ATTENTION save value "A" into help sheet
      'allows revision mark's last alphabet to increase incrementally
      'NOTE unless revision mark lenght is not 2, R is not used
      apulu.Range("A1").Value = "A"
      'if total lenght is 2, first letter is R and last isn't Z
      ElseIf Len(revi) = 2 And Left(revi, 1) = R And Right(revi, 1) <> "Z" Then
        'increment the last alphabet (e.g. AA -> AB)
        tunnus.Value = R & Chr(Asc(Right(revi, 1)) + 1)
      Else 'start new lenght 2 revision cycle
        R = Chr(Asc(R) + 1)
        'update help value into the sheet (e.g. A -> B)
        apulu.Range("A1").Value = R
        tunnus.Value = R & "A" ' (e.g. AZ -> BA)
      End If
    End If
  End If
End If
```

Esimerkkikoodi 31. Muutoslomakkeen avautuessa täytetään tunnuskenttä oletusarvolla. Tämä arvo määritetään muuttujalla "revi", jolla on vähintään arvo "A" koodiosuutta suorittaessa. Jos muuttujan arvo on esimerkiksi "H", vaihtuu se kirjaimeksi "I". Kierro jatkuu Z-kirjaimen saakka, jolloin vaihdetaan molempia kirjaimia, esimerkiksi "BZ" muuttuu tunnukseksi "CA". Kahden merkin mittaisen revisiotunnuksen ensimmäinen kirjain on tallennettu aputaulukkoon.

```
'call a function "uusi" to check if xml-header info is new
If uusitieto(kkk(x), vvv) Then GoTo eluusitieto
Sarake.Show

Private Sub UserForm_Activate()
'when userform is shown, seek header each time
Label1.Caption = vbNewLine & "Alla oleva otsikkotieto löytyi XML-tiedostosta!"
& vbNewLine & "-----" & _
  vbNewLine & vbNewLine & vbNewLine & vbNewLine & _
  "-----"
& vbNewLine & "Onko kyseessä vaaka- vai pystysuora otsikkotieto?"
Label2.Caption = WorksheetFunction.Trim(Replace(kkk(x), vbLf, " "))
End Sub

Private Sub ps_hvr_Click()
'check if new header is vertical
sps = True
Sarake.Hide
End Sub
```

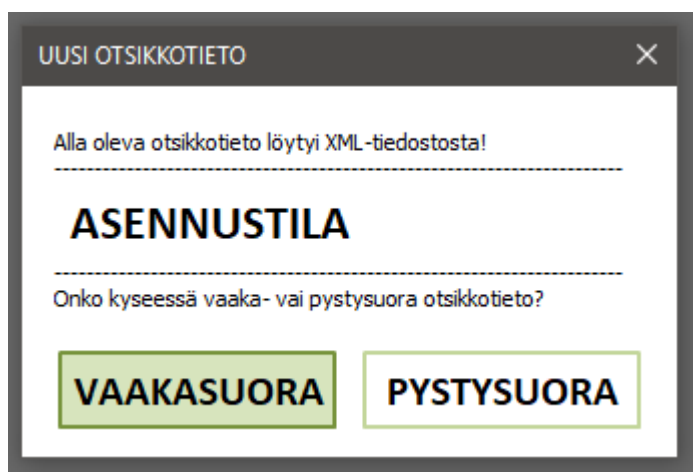


```

If sps = True Then 'if vertical
    .ColumnWidth = 5
    .Orientation = xlUpward
    .HorizontalAlignment = xlCenter
    'reset the boolean for next check
    sps = False
Else
    .ColumnWidth = 15 'if horizontal
    .Orientation = xlHorizontal
    .HorizontalAlignment = xlLeft
End If

```

Esimerkkikoodi 32. Kun muutos valaisintiedoissa havaitaan, kutsutaan vaadittavaa aliohjelmaa. Ohjelma hyödyntää aiemmin käytettyä funktiota uusien otsikoiden tarkastukseen. Mikäli otsikko on uusi, näytetään kuvan 23 mukainen käyttäjälomake, josta voidaan valita sen sarakkelevyys. Muokkaus tehdään boolean arvon mukaisesti.



Kuva 23. Kuvan mukaisella käyttäjälomakkeella päätetään uuden sarakkeen leveys ja tekstin suuntaus. Pystysuorana tietona voi olla esimerkiksi valaisimen IK-luokitus.

```

Private Sub peruuta_hvr_Click()
'if user selects cancel with PERUUTA
'hide the userform and end all subroutines
    valu.Activate
    taulukko.reset
    Unload Me
End
End Sub

Private Sub UserForm_QueryClose( _
Cancel As Integer, CloseMode As Integer)
'if user closes userform, forcefully end the macro
If CloseMode = 0 Then
    valu.Activate
    taulukko.reset
    Unload Me
End
End If
End Sub

```

```

Sub PERUUTA_MouseMove( _
    ByVal Button As Integer, ByVal Shift As Integer, _
    ByVal x As Single, ByVal y As Single)
'make PERUUTA appear red when hovered on
PERUUTA.Visible = False
OK.Visible = True
SUODATA.Visible = True
End Sub

```

Esimerkkikoodi 33. Yllä olevat koodiosuudet edustavat kaikkia käyttölomakkeita. Mikäli käyttäjä sulkee lomakkeen painikkeesta tai sulkee ikkunan, palautetaan taulukko oletusasetuksille ja kaikki makrot pysäytetään End-komennolla. Painikkeen korostus suoritetaan MouseMove-aliohjelmalla, joka löytyy editorin Events-alasvetovalikosta.

```

If r_upval = True Then
'if lamp is new or was removed
r_post(rcntr) = kt(rev, 1) 'pos value
'if lamp was removed, fill r_ut with x
'NOTE +2 is for the space and "("
rstr = Len(kt(rev, lc - 1)) - Len(revisio) - 2
If Trim(Right(kt(rev, lc - 1), rstr)) = "POIS" Then
r_pt(rcntr) = " X"
Else 'else fill r_ut with x
r_ut(rcntr) = " X"
End If
'set boolean for recheck
r_upval = False
End If

```

```

If r_tieto = True Then
'if specs changed, fill r_tt with x
r_post(rcntr) = kt(rev, 1) 'pos value
r_tt(rcntr) = " X"
'loop through all changed specs and concatenate
'NOTE uboud - 1 because last element is empty
For z = LBound(r_tiedot) To UBound(r_tiedot) - 1
r_ttt(rcntr) = r_tiedot(z) & " / " & r_ttt(rcntr)
Next z
'set boolean for recheck
r_tieto = False
End If

```

```

If r_lkm = True Then
'if amount changed, fill r_mt with count
r_post(rcntr) = kt(rev, 1) 'pos value
'NOTE +2 is for the space and "("
rstr = Len(rt(rev)) - Len(revisio) - 2
r_mt(rcntr) = Right(rt(rev), rstr)
'set boolean for recheck
r_lkm = False
End If

```

Esimerkkikoodi 34. Koodin 26 rinnalla toimii raporttimoduulin aliohjelma, joka suorittaa toimenpiteet päämoduulin boolean arvon perusteella. Jos tiedot ovat muuttuneet, täytetään joukko r_tt x-merkillä viesti-ikkunaa varten. Lisäksi kaikki muutokset ketjutetaan otsikkotietojen perusteella joukkoon r_ttt. Myös uudet ja poistuneet valaisimet sekä lukumäärämuutokset käsitellään aliohjelmassa.

```
Private Sub UserForm_Initialize()  
j = 0 'create array of all materials  
For i = 6 To Sheets.Count  
ReDim Preserve backupsheets(j)  
backupsheets(j) = ThisWorkbook.Sheets(i).Name  
j = j + 1  
Next i  
  
'insert created array into listbox  
kaikkimuutokset.List = backupsheets  
End Sub  
  
Private Sub UserForm_Activate()  
'if lamps have been filtered upon form activation  
If Jaotta.arrayontyhja(valitutvalaisimet) = False Then  
'replace revision list with filtered lamp list  
kaikkimuutokset.Visible = False  
kaikkipositiot.Visible = True  
'move filtering button outside the userform  
'because list of filtered lamps extends over it  
'this will also make it "invisible" for users  
SUODATA.Left = SUODATA.Left + 1000  
SUODATA_HVR.Left = SUODATA_HVR.Left + 1000  
'ensure single selection is enabled for the listbox  
kaikkipositiot.MultiSelect = fmMultiSelectSingle  
'insert filtered lamps into the listbox  
kaikkipositiot.List = valitutvalaisimet  
End If  
End Sub  
  
Private Sub suodata_hvr_Click()  
'when user confirms filtering, reset x and z before loop  
For y = 0 To kaikkimuutokset.ListCount - 1: x = 0: z = 0  
'loop all materials and add selected into valituthistoria  
If kaikkimuutokset.Selected(y) = True Then  
ReDim Preserve valituthistoria(x)  
valituthistoria(x) = kaikkimuutokset.List(y)  
x = x + 1: z = z + 1  
End If  
Next y  
  
If z <= 1 Then GoTo alasuodata  
'if enough materials were selected  
Hide 'then call subroutine for filtering  
Historia.suodatin  
Exit Sub  
  
alasuodata: 'if not enough materials are selected  
MsgBox "Vähintään kaksi muutosta on oltava valittuna!", _  
vbOKOnly + vbInformation, "Virhe!" 'show error message  
Unload Me 'unload and reshew form  
Tyokalu_muutoshistoria.Show  
End Sub
```

Esimerkkikoodi 35. Kun historiatyökalu käynnistetään, käyttäjälomakkeen lista täytetään luoduilla materiaaleilla. Tämän rinnalla lomake myös aktivoituu, jolloin tarkastetaan, onko suodatus tehty. Mikäli näin on, korvataan aineistot suodatetuilla valaisimilla. Jos aineistoja ei ole valittu riittävästi, "valitutvalaisimet" joukon luomisen sijaan annetaan käyttäjälle virheilmoitus sekä näytetään lomake uudelleen.

```

On Error Resume Next 'if reviewable lamp is last, InStr with "," returns error
For x = LBound(muutostiedot) To UBound(muutostiedot)
revisiovalittu = False 'set boolean false before next entry
'check if revision date was included in selection
For z = LBound(valituthistoria) To UBound(valituthistoria)
If InStr(1, valituthistoria(z), muutostiedot(x, y - 1), _
vbTextCompare) <> 0 Then
revisiovalittu = True
Exit For
End If
Next z
If revisiovalittu = False Then
muutostiedot(x, 1) = "EIVALITTU"
GoTo revisioeivalittu:
End If

'first check whenever the change was contract or revision
If InStr(1, muutostiedot(x, y), "urakka", vbTextCompare) = 0 Then
'next check if revision contained the lamp position
If InStr(1, muutostiedot(x, y), "POS " & hispos & ":", vbTextCompare) Then
'if it does, find location of reviewable lamp position (as string)
'within changes and add the length of search item
i = InStr(1, muutostiedot(x, y), "POS " & hispos & ":", _
vbTextCompare) + Len("POS " & hispos & ":") 'text after "POS x: "
j = InStr(i, muutostiedot(x, y), ",", vbTextCompare) 'find next comma
'if comma is not found, j is set to zero, then find next dot instead
If j = 0 Then j = InStr(i, muutostiedot(x, y), ".", vbTextCompare)
'first remove comma or dot and text after it, then remove pos value
muutostiedot(x, y) = Left(muutostiedot(x, y), j - 1) '-1 for comma or dot
muutostiedot(x, y) = Right(muutostiedot(x, y), j - i - 1) '-1 for space
after ":"
Else: muutostiedot(x, y) = "---" 'if selected but no changes
End If

Else 'if change is contract material

muutostiedot(x, 1) = "Urakka" 'insert "contract" into revision mark
Sheets(x + 5).Activate 'activate backup sheet (+5 for default sheets)
lr = ActiveSheet.Range("A" & Rows.count).End(xlUp).Row 'last row
'search column A for lamp position from contract material
Set hisvalurakkarivi = ActiveSheet.Range _
(Cells(1, 1), Cells(lr, 1)).Find(What:=hispos)
If Not hisvalurakkarivi Is Nothing Then
'if lamp was found and variable is not empty
muutostiedot(x, y) = hisvalurakkarivi. _
Offset(0, lc - 3).Value & " kpl" 'fill array with amount
Else 'else inform that lamp was not in material
muutostiedot(x, y) = "Ei urakka-aineistossa."
End If

End If 'skip here if not selected
revisioeivalittu:
Next x 'next revision

```

Esimerkkikoodi 36. Käydään läpi suodatettu materiaali. Mikäli muutosta ei valittu, täytetään muutoksen tunnus tekstillä "EIVALITTU" poistamista varten. Jos kyseessä on urakka, käydään hakemassa lukumäärät valintaa vastaavista välilehdistä. Muutosaineistojen osalta etsitään positiota lokikirjauksesta. Mikäli positio löytyy, poistetaan sitä edeltävä sekä seuraavan ", " tai "." jälkeinen teksti ja viedään muokattu sisältö joukkoon.

Valaisinluettelon rakenna ja ulkoasu

The screenshot shows a Microsoft Excel spreadsheet with a sidebar on the right. The spreadsheet contains a table with columns for 'POS.' (1-40), 'TUOTE' (1), 'VALAISTUSLAITTEEN NIMI JA SUUNNITTELU' (2), 'VALAISTUSLAITTEEN TUNNUS' (3), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (4), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (5), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (6), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (7), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (8), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (9), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (10), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (11), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (12), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (13), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (14), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (15), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (16), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (17), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (18), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (19), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (20), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (21), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (22), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (23), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (24), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (25), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (26), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (27), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (28), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (29), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (30), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (31), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (32), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (33), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (34), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (35), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (36), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (37), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (38), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (39), 'VALAISTUSLAITTEEN TUOTEMÄÄRÄ' (40).

The sidebar on the right contains the following buttons and options:

- PÄIVITÄ TAULUKKO** (Update Table)
- KÄYTÄ TYÖKALU** (Use Tool)
- KÄYTTÄÄVAUVAUNNAT** (Use Templates)
- ALÄ POISTA MUUTOKSIA** (Remove Changes)
- ALÄ TEE TULOSTETTA** (Do Not Print)
- ALÄ NÄYTA RAPORTTIA** (Do Not Show Report)
- TYÖKALUT** (Tools)
- OTA POFF-TILOSTE** (Get POFF Summary)
- PÄIVITÄ OTSIKKOTIEDOT** (Update Section Info)
- UUDELLENNÄÄRITÄ KIVL** (Refresh KIVL)
- PALAUTA ANVEISTO** (Reset Anveisto)
- VALAISINLUETTLO** (Lighting Fixture List)

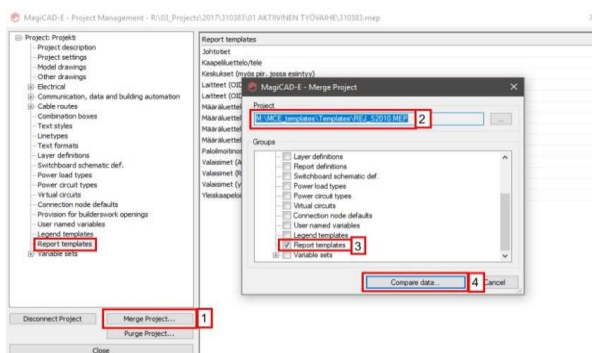
Valaisinluettelon käyttöohjeet

ESIVALMISTELUT

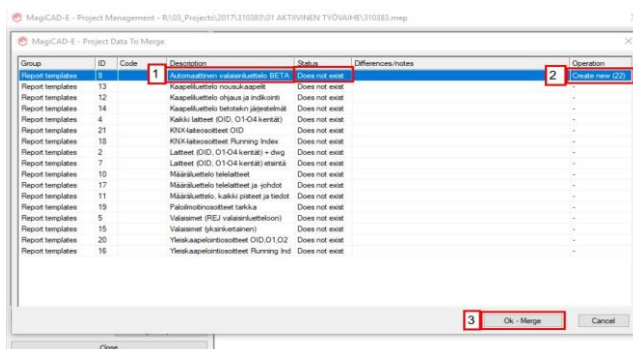
1. Ensimmäisenä on lisättävä tarvittavat tiedostot projektiin. Selaa tiedostonhallinnalla alla olevaan polkuun ja kopioi ne projektiin.

*M:\MALLIT JA VALMIIT POHJAT\01 TAULUKOT, LASKELMAT JA TEKSTIT\05
Valaisintaulukot\Epäviralliset\Automatoitu valaisinluettelo\versio 1.1*

2. Mene projektin "01 AKTIIVINEN TYÖVAIHE"- kansioon ja uudelleennimeä tiedosto "valaisinluettelopohja" projektin piirustusluettelon mukaisesti.
3. Tuo raporttipohja projektiin, mikäli sitä ei vielä löydy. Käynnistä projektinhallinta ja valitse "Merge Project..."
(1). Selaa MEP-mallitiedostoon (2) ja aktivoi "Report templates" (3). Tämän jälkeen paina "Compare data..."
(4). Pohjan nimi on "Automaattinen valaisinluettelo BETA".

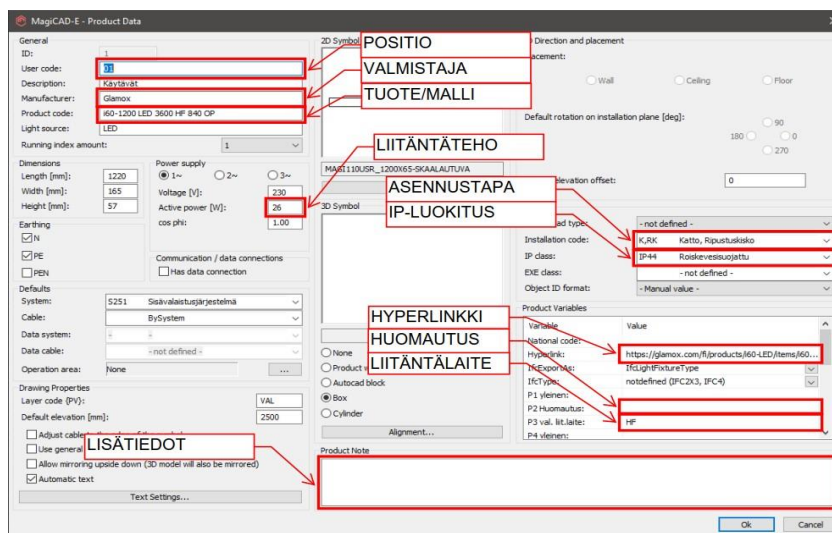


4. Valitse ensin kaikki raporttipohjat painamalla Ctrl + A, sitten klikkaa hiiren oikeaa ja valitse "Do not merge (use target project's value)". Tämän jälkeen valitse edelle mainittu raporttipohja aktiiviseksi, klikkaa jälleen hiiren oikeaa ja valitse "Create new..." (1). [Jos pohjan "Status" on muuta kuin "Does not exist", on se jo osana projektia eikä sen liittämislle on tarvetta \(2\)](#). Vahvista lopuksi "Ok - Merge" painikkeesta (3).

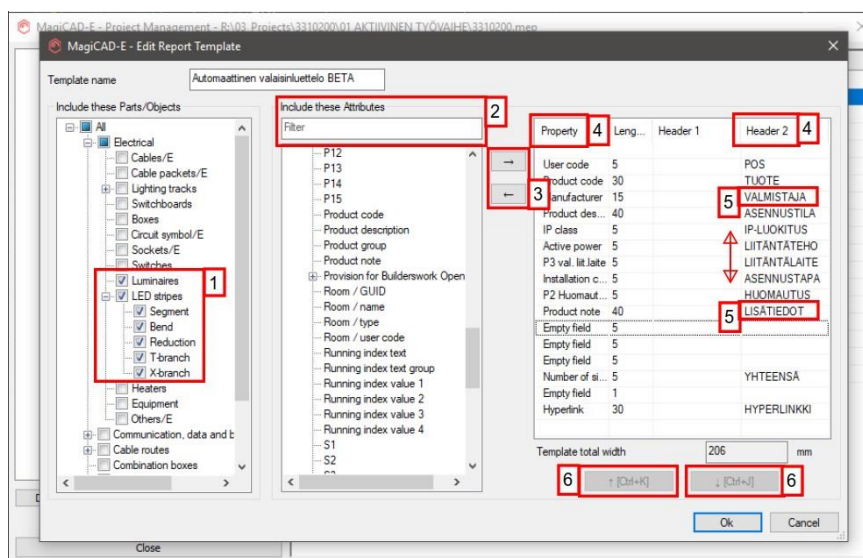


VALAISIMIEN LISÄYS

1. Taulukon hakemistosta löytyy tiedosto ”VALAISIMEN LISÄYS MEPPIIN”. Tässä on esitetty, mistä kentistä taulukko hakee oletustiedot, mutta sitä pystytään käyttämään myös projektikohtaisten tietojen täyttöohjeena. Esimerkiksi Description-kenttää voidaan käyttää asennustilietuna tai Px-kenttää IK-luokan esittämiseen. Tiedosto on vapaasti muokattavissa projektin ja käyttäjien tarpeiden mukaisesti.

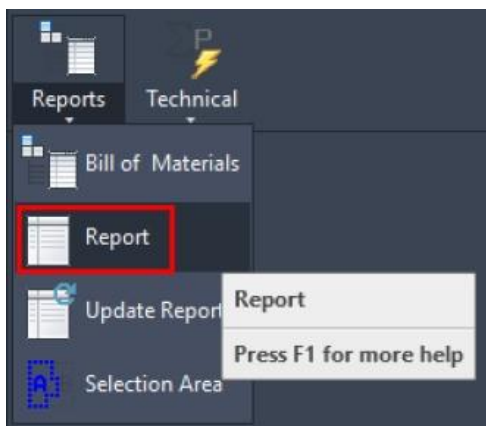


2. Raporttipohjaa päästään muokkaamaan projektihallinnasta valitsemalla pohja aktiiviseksi, klikkaamalla hiiren oikeaa ja valitsemalla ”Properties”. Vasemmasta laidasta voidaan valita luettelon laskemat laitteet (1). Keskellä olevia tietoja (property) (2) voidaan lisätä tai poistaa raportista näiden välissä sijaitsevilla nuolipainikkeilla (3). [Header 2 toimii tiedon otsikkona ja tämä tulee näkyviin taulukkoon \(4\)](#). [Muita ominaisuuksia, kuten Header 1, ei huomioida. Tietojen lisäys ja poistaminen on tapahduttava valmistajan ja lisätietojen välillä \(5\)](#). Näiden järjestys voidaan muuttaa alhaalla sijaitsevista painikkeista (6).

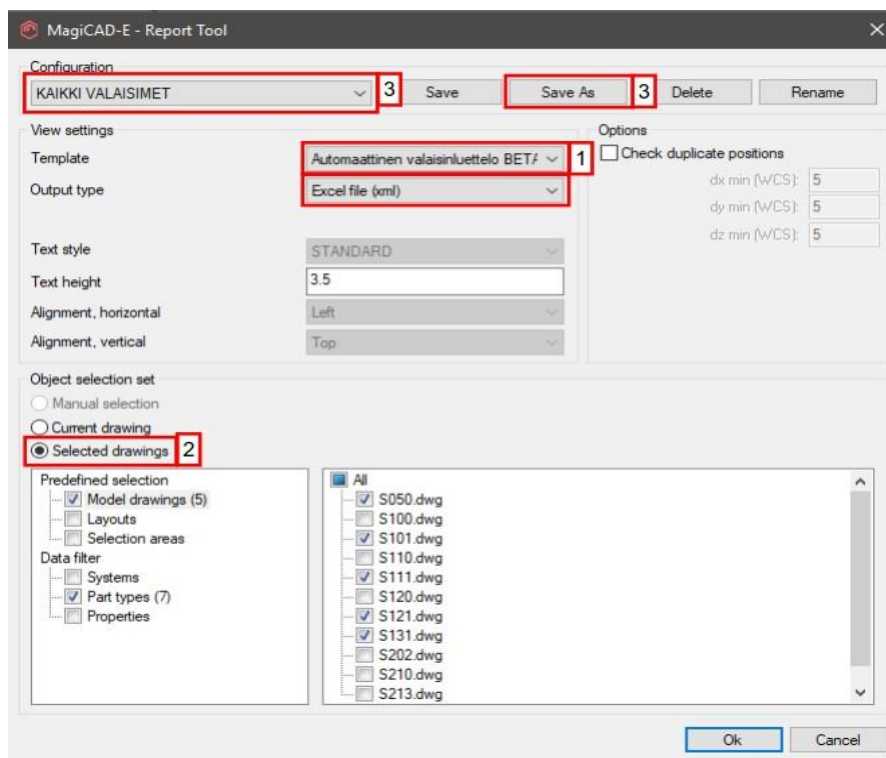


RAPORTIN LAATIMINEN

1. Taulukon lähtötietoina toimii "Report"-työkalulla luotu xml-tiedosto, jonka taulukko pyytää valitsemaan ensimmäisellä käyttökerralla. [Ei käytetä "BILL OF MATERIALS"-komentoa.](#)

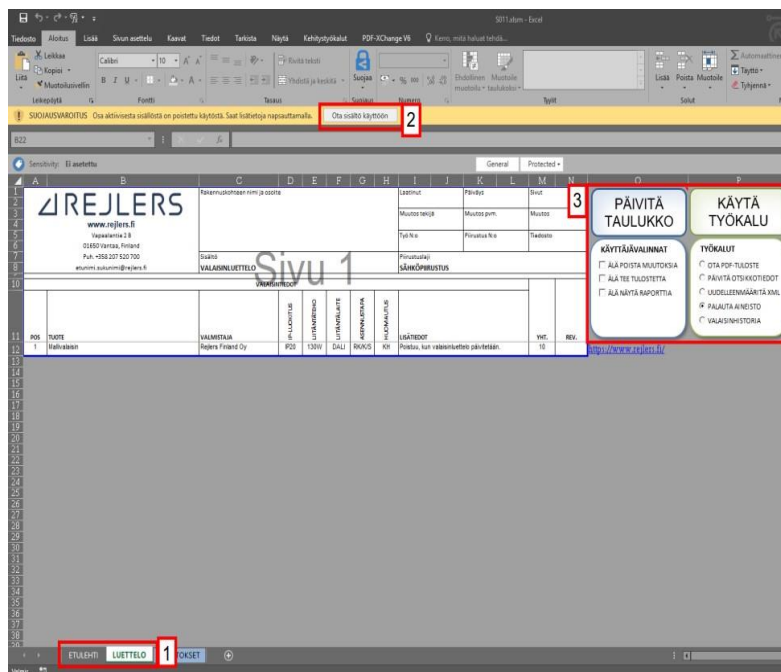
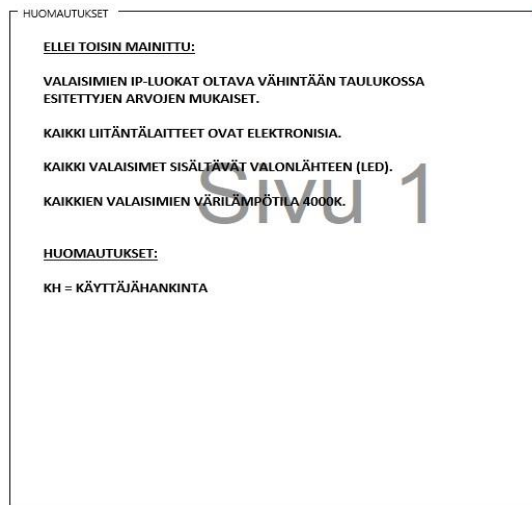


2. Valitse aukeavasta ikkunasta pohjaksi (template) "Automaattinen valaisinluettelo BETA" ja tulosteen tyyppi (output type) "Excel file (xml)" (1). Tämän jälkeen määritä raporttiin kuuluvat tasopiirustukset ja muut mahdolliset rajaukset (2). Lopuksi asetukset voidaan tallentaa nimellä uudelleenkäyttöä varten (3). [Anna tiedostolle selkeä nimi, esimerkiksi \[projektinumero\]_valaisimet, jolloin se erottuu varsinaisesta valaisinluettelosta.](#)



TAULUKON KÄYTTÖ

1. Avaa valaisinluettelon Excel-tiedosto ja ota muokkaus käyttöön. Taulukon toiminnot tahtuvat LUETTELO-välilehdeltä (1) ja ensimmäisellä käyttökerralla on sallittava makros-sisältö (2). Luettelon oikeasta laidasta, tulostusalueen ulkopuolelta löytyvät valintaruudut ja painonapit (3). Taulukon solut ovat täysin lukittuna, mutta käyttäjät voivat valintaruuduilla vaikuttaa taulukon toimivuuteen. Painikkeiden oikeassa ylänurkassa on punaiset kolmiot (kommentti), jotka tarjoat tiivistetyt ohjeet. Taulukon etulehden HUOMAUTUKSET-kenttä on muokattavissa käsin. Voit esimerkiksi lisätä kaikkia valaisimia koskevia yleistietoja tai projektiokohtaisia huomautustunnuksia, kuten "KH = KÄYTTÄJÄHANKINTA".



2. Kun päivität taulukon ensimmäistä kertaa painamalla "PÄIVITÄ TAULUKKO"-painiketta, pyytää taulukko täyttämään otsikkotaulun tiedot. Jos et tiedä kaikkia projektiin liittyviä tietoja tai haluat päivittää niitä myöhemmin, voit tehdä tämän käyttämällä työkaluoptiota "PÄIVITÄ OTSIKKOTIEDOT". Mikäli painat PERUUTA-nappia, ei tietoja viellä taulukoon. Ikkunan sulkeminen lopettaa päivityksen.

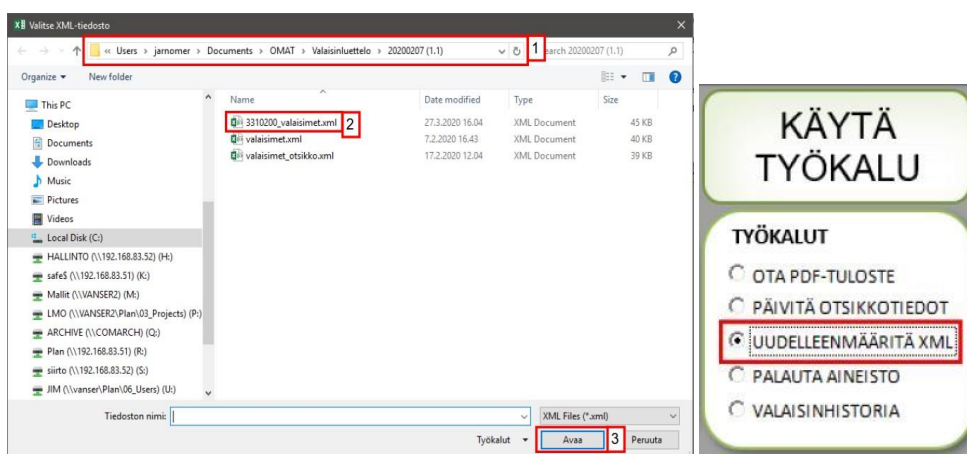
The image shows two parts of the software interface. On the left is the 'OTSIKKOTIEDOT' dialog box with the following fields:

- RAKENNUSKOHDE: [Empty]
- LÄHISOITE: [Empty]
- POSTINUMERO JA POSTITOIMIPAIKKA: [Empty]
- TYÖNUMERO: [Empty]
- PIIRUSTUSNUMERO: S011
- TIEDOSTONIMI: S011.xlsx
- LAATINUT: [Empty]
- PÄIVÄYS: 30.3.2020
- SISÄLTÖ: VALAISINLUETTELO

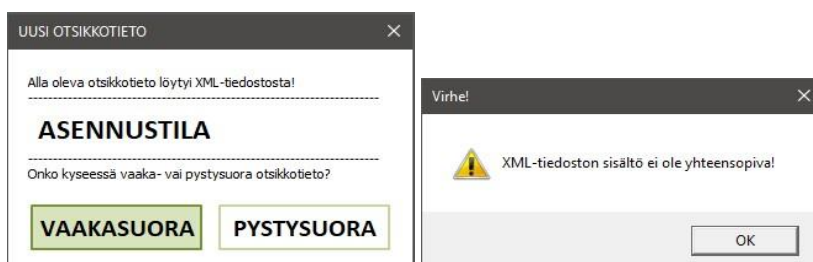
At the bottom of the dialog are 'OK' and 'Peruuta' buttons. On the right is the 'KÄYTÄ TYÖKALU' menu with the following options:

- OTA PDF-TULOSTE
- PÄIVITÄ OTSIKKOTIEDOT** (highlighted with a red box)
- UUDELLEENMÄÄRITÄ XML
- PALAUTA AINEISTO
- VALAISINHISTORIA

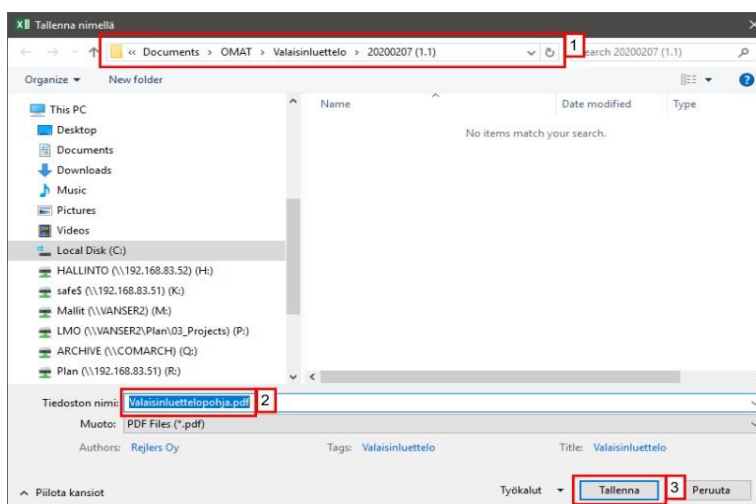
3. Tämän jälkeen taulukko pyytää määrittämään lähtötietojen sijainnin. Selaa kansioon, joka sisältää MagiCAD:sta ottamasi XML-tiedoston (1), valitse se aktiiviseksi (2) ja paina Avaa-painiketta (3). Mikäli haluat muuttaa lähtötietoina toimivaa tiedostoa, voit käyttää työkalua "UUDELLEENMÄÄRITÄ XML". Taulukon päivittäminen pysähtyy, jos suljet ikkunan tai valitset "Peruuta".



4. Jos olet lisännyt raporttipohjaan tietoja, jotka eivät löydy siitä oletuksena, pyytää taulukko sinua määrittämään onko uusi tieto vaak- vai pystysuorassa. [Pystysuorat otsikot sisältävät lyhyitä tietoja, kuten IP-luokituksen tai asennustavan](#). Muut otsikot, kuten asennustila tulee olla vaakasuorassa. Luettelossa olevat otsikkotiedot, joita ei löydy enää xml-tiedostosta poistuvat automaattisesti. [Tämä toimenpide suoritetaan vain ensimmäisellä käyttökerralla!](#) Mikäli huomaat, että taulukko pyytää sinua määrittämään virheellisen otsikon, voit pysäyttää päivityksen sulkemalla ikkunan. Mikäli XML-tiedoston sisältö ei ole yhteensopiva taulukon kanssa, pysäyttää taulukko päivityksen ja antaa alla olevan virheilmoituksen. [Taulukko ei huomio valaisimia, joilla ei ole positiotunnusta!](#)



5. [Tämän jälkeen taulukko rakentaa urakkalaskenta-aineiston automaattisesti](#). Kaikki tietojen päivitykset, muotoilut ja tulostusalueiden muokkaukset tapahtuvat automaattisesti. Päivityksen jälkeen taulukko pyytää käyttäjää tallentamaan PDF-tiedoston automaattisesti. Oletuskansiona käytetään taulukon sijaitsemiskansiota (1). Anna tiedostolle nimi (2) ja paina "Tallenna" (3). Voit poistaa tämän ominaisuuden käytöstä laittamalla ruksin käyttäjävalintaan "ÄLÄ TEE TULOSETTA". Lisäksi voit ottaa tulosteen manuaalisesti työkaluoptiolla "OTA PDF-TULOSTE".



6. Kun taulukko on päivitetty kerran ja urakkalaskenta-aineisto on rakennettu, pyytää taulukko seuraavan päivityksen yhteydessä antamaan muutostiedot. Taulukko olettaa ensimmäisen muutosaineiston olevan "A" (1), mutta voit muokata tätä vapaamuotoisesti. Esimerkiksi lisäkirjettä laatiessa voidaan tunnuksesi laittaa "1". [Jos haluat rakentaa urakkalaskenta-aineiston uudella päivämäärällä, jätä tunnuskenttä \(1\) tyhjäksi!](#) Mikäli taulukko ei havaitse muutoksia lähtötiedoissa, antaa se alla olevan virheilmoituksen ja palauttaa viimeisimmän muutosaineiston taulukkoon.

The image shows two windows. The first is a dialog box titled 'MUUTOSTIEDOT' with fields for 'TEKIJÄ' (JMe), 'PÄIVÄMÄÄRÄ' (31.3.2020), and 'TUNNUS' (A). The 'TUNNUS' field is highlighted with a red box and a '1' next to it. Below the fields are 'OK' and 'Peruuta' buttons. The second window is titled 'Virhel' and contains a red 'X' icon and the text 'XML-tiedostossa ei havaittu muutoksia!' with an 'OK' button below.

7. Päivityksen päätteeksi taulukko esittää raportti-ikkunan, josta näkyy yhteenveto tapahtuneista muutoksista. Kuten taulukon ensimmäisellä käyttökerralla, tapahtuvat kaikki muutokset automaattisesti. Seuraavan päivityksen yhteydessä aiemmat muutokset poistetaan automaattisesti, mutta käyttäjä voi vaihtoehtoisesti säilyttää edelliset muutokset käyttäjävalinnalla "ÄLÄ POISTA MUUTOKSIA". Lisäksi raportointia voidaan hallita optiolla "ÄLÄ NÄYTÄ RAPORTTIA". [Jokainen uudella päivämäärällä laadittu päivitystapahtuma kerätään MUUTOKSET-välilehdelle!](#)

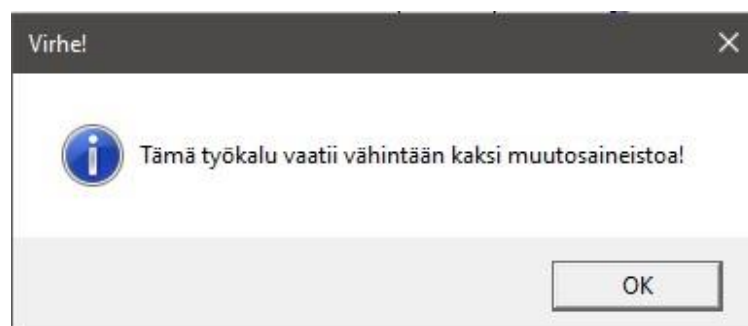
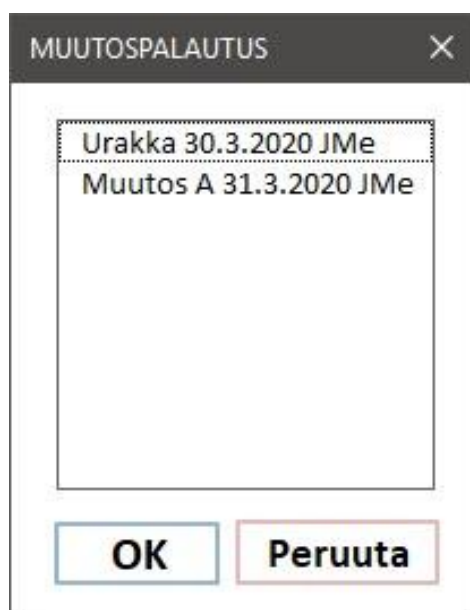
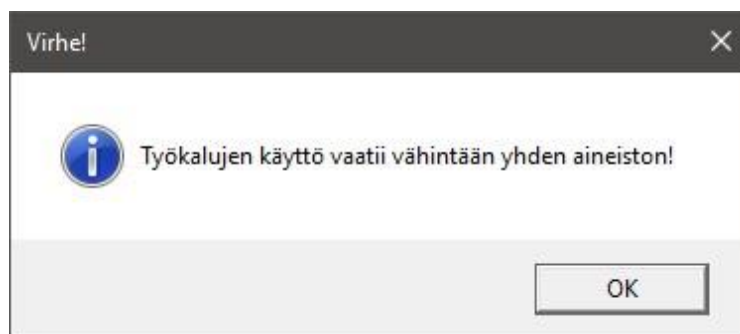
The image shows two windows. The first is a dialog box titled 'REPORT OF REVISION: A / DATE: 31.3.2020 / CREATED BY: JMe' with a table of changes. The second window is titled 'PÄIVITÄ TAULUKKO' and contains a section 'KÄYTTÄJÄVALINNAT' with three checkboxes: 'ÄLÄ POISTA MUUTOKSIA' (checked), 'ÄLÄ TEE TULOSTETTA' (unchecked), and 'ÄLÄ NÄYTÄ RAPORTTIA' (checked). Both windows have an 'OK' button.

POS	LKM	TIETO	POIS	UUSI
1	-20			
2	+40			
4			X	
7		X		
22				X

	A	B	C
1	MUUTOS	PÄIVÄMÄÄRÄ	TIEDOT
2		30.3.2020	Uusi urakkalaskenta-aineisto laadittu.
3	A	31.3.2020	POS 1: -20 kpl, POS 2: +40 kpl, POS 4: poistunut valaisin, POS 7: lisätiedot / liitântälaite / tuote muuttunut, POS 22: uusi valaisin.

TYÖKALUT

1. Edellä mainittujen työkalujen lisäksi taulukkoon on mahdollista palauttaa aiempi muutosaineisto käyttämällä optiota "PALAUTA AINEISTO". [Kaikkien työkalut vaativat vähintään yhden aineiston, palautustoiminto vaatii kaksi.](#) Taulukko antaa käyttäjälle virheilmoituksen, mikäli nämä kriteerit eivät täyty. Palautettava aineisto voidaan valita työkalun esittämältä listalta. [Työkalut eivät päivitä taulukon sisältöä! Päivitys suoritetaan aina käyttäen uusinta muutosaineistoa!](#)



2. Lisäksi valaisinkohtaisia muutoksia voidaan tarkastella käyttämällä työkalua ”VALAISINHISTORIA”. Työkalun esittämältä listalta voidaan vapaamuotoisesti valita aineistoja suodatusta varten, jonka jälkeen valitun valaisimen muutokset näytetään uudessa ikkunassa OK-napin painalluksella. Ilmoitus annetaan myös, kun valaisimessa ei ole tapahtunut muutoksia valittujen aineistojen aikana.

HISTORIATARKASTELU

Urakka 30.3.2020 JMe
Muutos A 31.3.2020 JMe

Poista kaikki valinnat.

SUODATA

OK Peruuta

HISTORIATARKASTELU

1
2
3
4
5
7
8
9
10
11
12
15
17
20

OK Peruuta

HISTORY REVIEW OF LAMP POSITION: 7

REVISIO	PÄIVÄMÄÄRÄ	MUUTOKSET
Urakka	30.3.2020	5 kpl
A	31.3.2020	lisätiedot / liitäntälaitte / tuote muuttunut

OK

Ei muutoksia!

i Tässä valaisimessa ei ole tapahtunut muutoksia!

OK