



Ville Koivumäki

**MIFARE DESFIRE EV1 -TUNNISTEIDEN AUTOMAATTISEN TESTAUS- JA  
ALUSTUSOHJELMAN SUUNNITTELU JA TOTEUTUS IDESCO OY:N  
TUOTANTOROBOTTIIN**

**MIFARE DESFIRE EV1 -TUNNISTEIDEN AUTOMAATTISEN TESTAUS- JA  
ALUSTUSOHJELMAN SUUNNITTELU JA TOTEUTUS IDESCO OY:N  
TUOTANTOROBOTTIIN**

Ville Koivumäki

Opinnäytetyö

Kevät 2012

Tietotekniikan koulutusohjelma

Oulun seudun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu  
Tietotekniikan koulutusohjelma

---

Tekijä(t): Ville Koivumäki

Opinnäytetyön nimi: MIFARE DESFire EV1 -tunnisteiden automaattisen testaus- ja alustusohjelman suunnittelu ja toteutus Idesco Oy:n tuotantorobottiin.

Työn ohjaaja(t): Kari Jyrkkä (OAMK), Anu-Leena Arola (Idesco)

Työn valmistumislukukausi ja -vuosi: Kevät 2012

Sivumäärä: 40

---

Tämä opinnäytetyö tehtiin Idesco Oy:lle. Idesco Oy on tunnistusratkaisuihin erikoistunut toimittaja ja on yksi alansa vanhimmista yrityksistä. Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa MIFARE DESFire EV1 -tunnisteiden automaattinen testaus- ja initialisointiohjelma PC-ohjattuun automaattiseen tuotantojärjestelmään.

Ohjelmaa käytetään MIFARE DESFire EV1 -tunnisteiden automaattiseen ohjelmointiin tuotantorobotissa. Se kirjoittaa etätunnisteen siruun ennalta määriteltyä dataa ja esimerkiksi jatkaa etätunnisteen sarjanumeroa seuraavaan tunnisteeseen. Jos etätunnistin on viallinen, järjestelmä siirtää sen hylkyyn ja seuraavaan tunnisteeseen siirretään hylätyn tunnisteen tiedot. Ohjelma toteutettiin C#-ohjelmointikielellä.

Työn tavoitteet saavutettiin ja työn lopputuloksena syntynyt ohjelma otetaan Idesco Oy:ssä käyttöön kevään 2012 aikana.

---

Asiasanat: MIFARE DESFire EV1, etätunniste, RFID

# ABSTRACT

Oulu University of Applied Sciences  
Information Technology and Telecommunications

---

Author(s): Ville Koivumäki

Title of thesis: MIFARE DESFire EV1- tags automatic testing and initialization software design and implementation to Idesco Ltd's manufacturing robot.

Supervisor(s): Kari Jyrkkä (OUAS), Anu-Leena Arola (Idesco)

Term and year when the thesis was submitted: Spring 2012

Number of pages: 40

---

This Bachelor's thesis was done for Idesco Ltd. Idesco is one of the oldest and most experienced companies in the field of RFID technology. The aim of the thesis was to design and implement testing and initialization software for MIFARE DESFire EV1 -tags to PC-controlled automated manufacturing system.

The software writes predefined data into the tags chip and continue serial number to next one. If the tags are faulty then the system will move it to the wreck place and the next tag identification information transferred from the abandoned.

The result of this thesis meet the objectives set in the beginning. The work requirements were completed and the software of the final result will end into production during the spring 2012.

---

Keywords: MIFARE DESFire EV1, tag , RFID

## ALKULAUSE

Työn toimeksiantajana toimi Idesco Oy. Opinnäytetyön ohjelmisto tehtiin Idesco Oy:n tiloissa ja opinnäytetyön teoria sekä kirjoitettu osuus tekijän omissa tiloissa. Työn tarkoitus oli kehittää MIFARE DESFire EV1 -tunnisteiden automaattinen alustusohjelma Idesco Oy:n tuotantorobottiin. Ohjelmisto kehitettiin Idesco Oy:n sisäiseen käyttöön.

Esitän kiitokset työn toimeksiantajalle ja Idesco Oy:n henkilöstölle, jotka olivat mukana projektin eri vaiheissa sekä loivat innovatiivisen ja rennon työilmapiirin työn tekemisen aikana. Työn valvojalle lehtori Kari Jyrkälle esitän kiitokset työn valvomisesta. Esitän myös kiitokset Idesco Oy:n tuotekehitysjohtaja Anu-Leena Arolalle työn ohjaamisesta. Haluan esittää kiitokset insinööriopiskelija Arto Hurskaiselle hyvin sujuneen yhteistyön johdosta. Hän teki samaan aikaan opinnäytetyötään tässä projektissa keskittyen mm. robotin ohjaukseen. Erityiskiitokset ansaitsee avovaimoni, joka on ollut kannustava sekä ymmärtäväinen opintojeni eri vaiheilla.

Oulussa 15.2.2012

Ville Koivumäki

# SISÄLLYS

TIIVISTELMÄ .....	3
ABSTRACT .....	4
ALKULAUSE .....	5
SISÄLLYS .....	6
LYHENNELUETTELO .....	8
1 JOHDANTO.....	9
1.1 Yritys.....	9
1.2 Työ.....	9
1.3 Käyttötarkoitus .....	10
2 RADIOTAAJUINEN ETÄTUNNISTUS .....	11
2.1 Historia.....	11
2.2 Toiminta .....	11
2.3 Tulevaisuus.....	13
2.4 MIFARE DESFire EV1 .....	14
2.4.1 Kortin rakenne.....	15
2.4.2 Kortin salaus .....	16
3 TUOTANTOROBOTTI.....	17
3.1 Kohdejärjestelmä .....	17
3.2 LINX 7300.....	17
3.3 Access 9CD .....	18
4 SUUNNITTELU .....	20
4.1 Vaatimusmäärittely .....	20
4.2 Asiantuntijahaastattelu.....	21
5 TOTEUTUS .....	23
5.1 Ohjelman kehitysympäristö.....	23
5.2 Ohjelman yleiskuvaus.....	23
5.3 Ohjelman toiminnallisuus.....	24
5.3.1 Kirjautuminen .....	25
5.3.2 Pääikkuna .....	26

5.3.3 Tilaus-välilehti .....	27
5.3.4 Formaatti-välilehti .....	28
5.3.5 Asetukset-välilehti .....	29
5.3.6 Loki-välilehti .....	30
5.3.7 Luku-välilehti .....	31
5.3.8 Kirjoitus-välilehti .....	31
6 TESTAUS .....	34
7 POHDINTA .....	35
LÄHTEET .....	37
LIITTEET .....	40

## LYHENNELUETTELO

AES	Advanced Encryption Standard, lohkosalausmenetelmä.
CRC	Cyclic Redundancy Check, tarkisteavaimen luontiin tarkoitettu tiivistealgoritmi.
DES	Data Encryption Standard, salausmenetelmä.
EEPROM	Electrically Erasable Programmable Read-Only Memory, pysyvä puolijohdemuisti.
EPC	Electronic Product Code, elektroninen tuotekoodi.
HF	High Frequency, korkea taajuus.
IEC	International Electrotechnical Commission, kansainvälinen sähköalan standardisoimisjärjestö.
ISO	International Organization for Standardization, kansainvälinen standardisoimisjärjestö.
NFC	Near Field Communication, RFID:hen pohjautuva lähitunnistustekniikka.
UHF	Ultra High Frequency, ultrakorkea taajuus.
UID	Unique Identification Number, laitteen sarjanumero.



# 1 JOHDANTO

## 1.1 Yritys

Toimeksiantajana tälle opinnäytetyölle toimi Idesco Oy, joka on oululainen vuonna 1989 perustettu yritys. Idesco Oy suunnittelee, valmistuttaa ja markkinoi RFID (Radio Frequency Identification, radiotaajuuksien etätunnistaminen) -teknologiaan perustuvia lukijoita, lukijamoduuleita, etäkortteja ja tunnisteita. Näitä sovelletaan mm. teollisuusautomaatiossa, kulunvalvonnassa, tuotemerkinnässä, joukkoliikenteen rahastusjärjestelmissä ja muissa haastavissa ympäristöissä. Haastava ympäristö voi olla esimerkiksi kaivosteollisuudessa, jossa tunnisteilta ja lukijoilta vaaditaan luotettavaa toimintaa mm. pölyisissä tai räjähdysalttiissa ympäristöissä.

## 1.2 Työ

Työn tavoitteena oli kehittää MIFARE DESFire EV1 -tunnisteiden automaattinen ohjelmointikäyttöliittymä yrityksen tuotantorobottiin. Tuotantorobotille tehtävän ohjelman avulla voidaan ohjelmoida tunnisteita automaattisesti määrittämällä alussa tunnisteiden tilaustiedot ja tunnisteen formaatti. Ohjelman tekniset ominaisuudet käytiin läpi vaatimusmäärittelyssä projektin alkuvaiheessa.

Työn edellytyksenä oli saada tuotantorobottiin kuuluvat laitteet toimimaan käyttöliittymän kontrollissa. Tuotantorobotin keskeisimmät laitteet ovat MIFARE DESFire EV1 -tunnisteita ohjelmoitaessa Access 9CD -lukijalaite, jota käytetään tunnisteen ohjelmointiin, ja Linx-tulostin, jolla tulostetaan haluttu tieto tunnisteen pintaan. Tuotantorobotti sekä siihen kuuluvat laitteet ovat kytkettynä sarjaporttien avulla tietokoneeseen, johon insinööriyössä suunniteltiin ja toteutettiin käyttöliittymä tunnisteiden automaattista ohjelmointia varten.

### 1.3 Käyttötarkoitus

Etätunnistekorteilla on monenlaisia käyttötarkoituksia. Yleensä korteille kirjoitetaan sarjanumeroa tai vakiodataa, jota hyödynnetään monenlaisissa sovelluksissa etätunnistekortteja tukevassa järjestelmässä. Etätunnistekorttien hyvinä puolina voidaan nähdä mm. luotettava autentikointi, helppo muokattavuus sekä luotettavuus.

Jos yritys tilaa esimerkiksi 10 000 kpl järjestelmänsä yhteensopivaa MIFARE DESFire EV1 -tunnistetta yhteen tiedostoon kirjoitetuilla sarjanumeroilla 10—10 000, ne voivat asettaa järjestelmän tukemaan tällöin 10 000 erilaista tunnistautumista pelkästään sarjanumeron perusteella. Autentikointitasot määritetään yrityksen tarpeiden mukaan ja tällöin käytännössä erilaisen tason autentikointeja on vähempi. Eri sarjanumeroille voidaan asettaa pääsy esimerkiksi eri tiloihin, ja samoja etätunnistekortteja voidaan käyttää myös muissa sovelluksissa. Autentikoiminen voidaan myös määrittää kortin salausavaimien perusteella, mutta tällöin järjestelmän ylläpidosta tulee työläämpää. Tällöin järjestelmän pitää testata se, onko kortilta luku onnistunut tietyllä avaimella, ja järjestelmään syötetään jokaista autentikointia varten oma avain. Yritys voi myös itse ohjelmoida kortteja siihen tarkoitetulla DESCoder-ohjelmalla. Suuremmat korttitilaukset on tosin helpompi hoitaa tuotantorobotilla, joka kykenee tekemään kortteja määritetyillä sarjanumeroilla nopeasti, automaattisesti ja luotettavasti.

Järjestelmä voidaan nopeasti konfiguroida eri asetuksin toimivaksi koskettamalla järjestelmän RFID-lukijaa siihen tarkoitetulla konfigurointikortilla. Henkilö, joka tämän toimenpiteen tekee, ei tällöin myöskään pääse näkemään muutettavia salausavaimia, PIN-koodeja tai muita turvallisuuden kannalta kriittisiä tietoja.

## **2 RADIOTAAJUINEN ETÄTUNNISTUS**

### **2.1 Historia**

Radiotaajuisen etätunnistuksen eli RFID-tekniikan juuret liittyvät tutkan keksimiseen 1935. Toisen maailmansodan aikana Sir Robert Alexander Watson-Wattin keksimää tutkaa käytettiin varoittamaan lähestyvistä lentokoneista. Pian brittiläiset keksivät lisätä lentokoneisiin lähettimen, joka osasi automaattisesti vastata tutkasignaaliin tunnistussignaalilla. Järjestelmää kutsuttiin nimellä "identify friend or foe", IFF-järjestelmä, ja se oli ensimmäinen RFID-järjestelmä maailmassa. (1, s. 1.)

1990-luvulla otettiin käyttöön ensimmäiset RFID-tekniikkaan perustuvat tietullijärjestelmät moottoritienopeuksissa ja sovellukset laajenivat logistiikasta myös kulunvalvontaan, henkilön tunnistukseen sekä muihin monimutkaisempiin sovelluksiin. Oulussa toteutettiin ensimmäisenä kaupunkina maailmassa joukkoliikenteen maksujärjestelmä RFID-tekniikkaan perustuen. Tämän ensimmäisen kaupallisen LF-radiotaajuudella toimivan joukkoliikenteen maksujärjestelmän toteuttivat yhteistyönä Idesco Oy ja Buscom Oy. (2, s. 24.)

2000-luvulla RFID-tekniikka on laajentunut mm. maksutoimintojen hoitamiseen NFC-tekniikan myötä. NFC-tekniikassa laite toimii lukijana sekä tunnisteena.

### **2.2 Toiminta**

RFID tulee sanoista Radio Frequency Identification. Normaalisti RFID-tunniste voidaan jakaa passiivisiin, puolipassiivisiin ja aktiivisiin tunnisteesiin. Tämä jako tehdään tunnisteen virtalähteen perusteella. Etätunniste on heijastusperiaatteeseen perustuva kommunikointimenetelmä, jossa lukulaite lähettää kantoaallon tai moduloidun kantoaallon ja kuuntelee samaan aikaan tai välittömästi moduloinnin jälkeen etätunnisteen moduloimaa signaalia. (3, s. 9.)

Passiivisissa etätunnistimissa ei ole sisäistä virtalähdettä eli akkua. Tällöin tunnistin ottaa käyttösähkään vaadittavan erittäin pienen sähkövirran laitteen antennista, johon indusoituu virta antennin ympärillä muuttuvan magneettikentän takia. Tuon muuttuvan magneettikentän ja sähkökentän saa aikaan lukijalaite, joka tuo antenninsa säteilemän sähkömagneettisen säteilyn tunnisteen antennin lähistölle. Tämän avulla laite pystyy lähettämään vastauksen, joka on tyypillisesti jonkinlainen sarjanumero tai laitteen ID eli identification number. Passiivisten etätunnistimien etuja ovat edullisuus, laitteen pieni koko, käyttöikä ja kestävyys. Passiivisten tunnisteen lukuetaisyudet vaihtelevat 10 millimetrin ja 5 metrin välillä. Huonoina puolina voidaan mainita yleensä lyhyt luku- ja kirjoitusetäisyys. (2, s. 10.)

Puolipassiivisten etätunnisteen ominaisuuksiin kuuluu se, että laite sisältää virtalähteen, mutta ei lähetintä. Tällä saavutetaan suurempi luku- ja kirjoitusetäisyys sekä mahdollistetaan laajempi toiminnallisuus tunnisteen omassa muistissa. Puolipassiiviset tunnisteen käyttävät akkua mikrosirun voimantähteenä, mutta kommunikoiivat ottamalla virtaa lukulaitteelta. Tosin uusimmat passiiviset tunnistetyypit ovat saaneet kiinni puolipassiivisia toiminnallisuudessaan, jolloin puolipassiiviset tunnisteen ovat pikku hiljaa korvautumassa passiivisilla. (2, s. 10.)

Aktiiviset RFID-tunnisteen sisältävät virtalähteen ja lähetimen. Tällöin saavutetaan huomattavasti pidempi toimintasäde. Haittapuolina voidaan mainita virtalähteen ehtyminen, suurempi koko sekä laitteen korkeampi hinta verrattuna passiivisiin etätunnistimiin. (2, s. 10.)

Tunnisteen käyttämät radiotaajuusalueet voidaan jakaa neljään eri alueeseen:

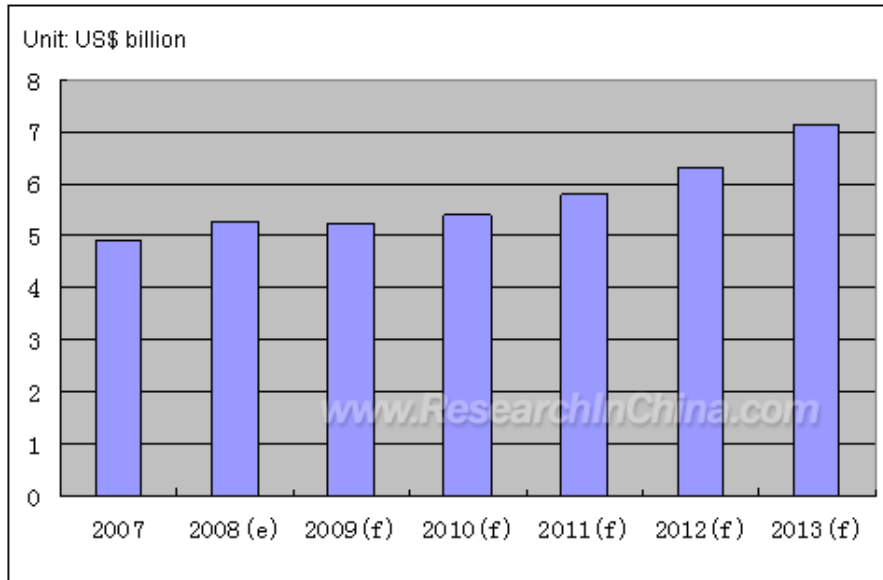
- matalan taajuuden tunnisteen (125–134 kHz)
- korkean taajuuden tunnisteen (13,56 MHz)

- UHF-tunnisteet (865 MHz Eurooppa, 956 MHz Japani)
- mikroaaltotunnisteet (2,45 GHz) (2, s. 10).

## 2.3 Tulevaisuus

Tärkeimmät ja kasvavimmat teknologiat tällä hetkellä ovat passiiviset korkeataajuiset High Frequency- (13,56 MHz) ja Ultra High Frequency -teknologiat (865 MHz Eurooppa) sekä NFC. HF- ja UHF-alueella toimivia etätunnistimia koskevat viestintäviranomaisten määrittelemät säännöt. Standardointi vaikuttaa yritysten kehitykseen ja vuorovaikutukseen. HF:n ja UHF:n kehityksen takana voidaan nähdä se, että niille molemmille etätunnistusteknologioille on rakennettu standardeja. (3, s. 10.)

Tällä hetkellä suurina erinä tilattuna passiivisen etätunnistekortin markkinahinta on noin 0,05 €. Edullinen hinta ja mahdollisuus toimia ilman erillistä teholähdettä ovat syyt siihen, miksi RFID-teknologian merkitys on niin suuri ja voimakkaasti kasvava lyhyen etäisyyden kommunikoinnissa. Vuotuisten markkinoiden (kuva 1) arvellaan kasvavan RFID-teknologiassa n. 0,5 mrd. € seuraavien vuosien aikana.



KUVA 1. Maailmanlaajuiset RFID-markkinat ja arvio tulevasta (f) 2007—2013 (4)

## 2.4 MIFARE DESFire EV1

MIFARE-teknologiat ovat NXP Semiconductors -yhtiön omistamia tavaramerkkejä. Näitä käytetään laajasti ilman kontaktia toimivissa älykorteissa ja etälukukorteissa. MIFARE-teknologiat pohjautuvat ISO/IEC 14443 standardiin. (5, s. 1.) MIFARE DESFire EV1 julkaistiin vuonna 2008 (6.)

MIFARE DESFire EV1- kortin käyttökohteita ovat

- julkisen liikenteen maksujärjestelmät
- kulunvalvonta
- valtioiden internet-palvelut
- maksupalvelut
- tunnistautumisohjelmat (5, s. 1).

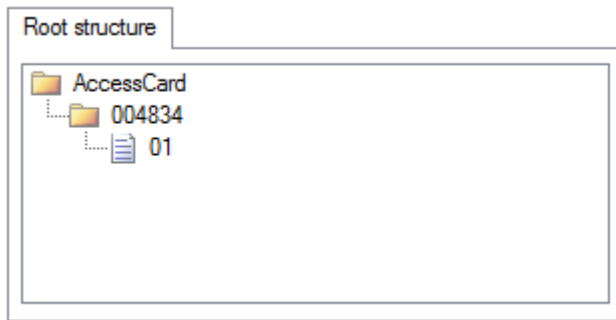
MIFARE DESFire EV1- kortin ominaisuudet ovat seuraavat:

- täysin ISO / IEC 14443 A 1-4 yhteensopiva
- 2/4/8 kB EEPROM

- tiedonsiirtonopeus 848 kbit/s asti
- joustava tiedostorakenne
- tuki DES/2K3DES/3K3DES/AES-salausalgoritmeille elektroniikkatasolla
- yksityisyyden suojaus
- uniikki 7-tavuinen sarjanumero
- voi sisältää maksimissaan 28 applikaatiota
- applikaatio voi sisältää max. 32 tiedostoa
- CRC-tarkistus ja bit-laskenta fyysisellä kerroksella (5, s. 1).

### 2.4.1 Kortin rakenne

MIFARE DESFire EV1 -tunniste voi sisältää enimmillään 28 applikaatiota ja yksi applikaatio voi sisältää enimmillään 32 tiedostoa. Käytännössä applikaatio on kansio, jolle voidaan asettaa 14 eri avainta kontrolloimaan kirjoitus-, luku-, luku-kirjoitus- ja tiedoston muutosoikeuksia. Tällä voidaan luoda melko joustava ja monipuolinen hakemistorakenne. Data voi olla heksadesimaali- tai ASCII-muodossa, salattuna tai salaamattomana. Lisäksi jokainen kortti sisältää UID:n (Unique Identification Number), joka on kovakoodattuna korttiin ja jota ei voi ohjelmallisesti vaihtaa. Rakennekuvassa (kuva 2) applikaation nimi on "004834" ja tiedoston nimi on 01.



*KUVA 2. Rakennekuvaus*

### **2.4.2 Kortin salaus**

Tunniste sisältää PICC-master-avaimen ja jokainen applikaatio voi sisältää 14 applikaatioavainta. Täten voidaan muodostaa monipuolisia autentikoimisratkaisuja eri sovelluksiin. MIFARE DESFire EV1 -tunniste tukee 56-bit DES-, 112-bit 3DES-, 158-bit 3DES- ja AES-salausalgoritmeja. Avaimet ovat pituudeltaan 128-bittisiä. Avaimet ovat 16-tavuisia. Ainoastaan kortin data-alue salataan ja käyttämättömälle alueelle voi silti kirjoittaa dataa, vaikka käyttäjä ei tietäisi PICC-master-avainta. (5, s. 1.)



## **3 TUOTANTOROBOTTI**

### **3.1 Kohdejärjestelmä**

Ohjelman asennuskohde oli Microsoft Windows XP -käyttöjärjestelmää käyttävä tietokone, joka oli kytketty sarjaporttien kautta Access 9CD -laitteeseen sekä Linx 7300 -kirjoittimeen. Tietokoneeseen piti asentaa myös Microsoft Office- sekä Microsoft Server MSSQL -ohjelmistot, koska ohjelma käytti Excel-tiedostojen tekemiseen Excel-ohjelman kirjastoja ja ohjelmaan kirjautumisessa käytettiin MSSQL-tietokantaa, jolloin vaadittiin Microsoft Server MSSQL -ohjelman asennus.

### **3.2 LINX 7300**

Tuotantorobotti käyttää Linx 7300 -sarjan mustesuihkutulostinta (kuva 3) sillä tehtävien tunnisteiden merkitsemiseen. Linx 7300 voidaan asettaa kirjoittamaan tunnisteita kahdella eri tavalla, automaattisesti tai sille lähetettävän datan määrittämänä. Automaattisella tavalla tehtynä se kirjoittaa tunnisteisiin samaa merkkisarjaa tai kasvattaa heksadesimaalimuodossa olevaa lukua tunnisteista seuraavaan.



*KUVA 3. LINX 7300 -mustesuihkutulostin (7, s. 1)*

Silloin kun automaattista moodia ei voi käyttää, robottia ohjaavan käyttöjärjestelmän on lähetettävä Linxille kirjoitettava merkkisarja jokaiselle tunnisteelle. Tällaisia tilanteita muodostuu esimerkiksi silloin, kun halutaan tehdä tunnisteita niin sanotuista hyppysarjoista. Näitä satunnaisien lukujen hyppysarjoja joudutaan tekemään esimerkiksi silloin, kun tunnisteita menee rikki.

### **3.3 Access 9CD**

Idesco Oy:n tekemällä Access 9CD -laitteella (kuva 4) on mahdollista lukea ja kirjoittaa MIFARE DESFire EV1- ja MIFARE Classic -tunnisteita. Laite käyttää tunnisteiden tiedonsiirtoon 13,56 MHz:n taajuutta. Laite kytkettiin sarjaportin kautta tuotantorobottiin kytkettyyn tietokoneeseen.

Tuotantomalli laitteelle on hieman erilainen kuin tuotantorobotissa käytettävä malli. Kotelointi on erilainen, koska tuotantorobotin mallin pitää voida operoida tunnisteiden kanssa, jotka ovat liukuhihnalla, ja taas Access 9CD -laitteen tuotantomallissa ohjelmoitava tunniste asetetaan laitteeseen käsin. Access 9CD -laitteen tiedonsiirto tapahtuu tuotantorobotissa suoraan tietokoneen sarjaporttiin ja tuotantomallissa USB-porttiin. Tuotantorobotin käyttämä Access 9CD -laitteen piti olla noin 1,3 cm:n etäisyydellä kirjoitettavasta tunnisteesta.



*KUVA 4. Access 9CD -tuotantomalli (8, s. 1)*

## 4 SUUNNITTELU

Työn suunnittelu käynnistyi heti projektin alussa. Oikeastaan ennen suunnittelua tehtäväni oli tutustua MIFARE DESFire EV1:n spesifikaatioihin, jotta ymmärtäisin DESFire-tunnisteen tiedostorakenteen, salausavainten toiminnan sekä valmistusprosessin. Tämä oli projektin läpi viemiseksi välttämätöntä. Vaatimusmäärittelyn lisäksi tehtiin projektin puolesta välissä PC Software Plan -niminen dokumentti helpottamaan niiden asioiden tekemistä, jotka vaatimusmäärittelyssä jäivät kirjaamatta ja jotka huomattiin projektin kannalta välttämättömäksi.

### 4.1 Vaatimusmäärittely

Vaatimusmäärittely oli työn suunnittelussa ensimmäisiä asioita. Palaverissa listattiin projektille seuraavanlaisia välttämättömiä ominaisuuksia:

- Ohjelmaan piti voida kirjautua normaalikäyttäjänä ja Administratorina. Vain Administratorin oikeuksin pystyi luomaan uuden käyttäjätunnuksen sekä luomaan ja aukaisemaan formaatin, jota käytetään tunnisteelle ohjelmoitaessa.
- Ohjelman piti pystyä tekemään kirjoituksesta lokia, josta pystyi tarkistamaan kirjoituksen onnistumisen tunnisteelle.
- Ohjelman piti kyetä kirjoittamaan juoksevaa sarjanumeroa vähintään kahteen valittuun tiedostoon sekä mahdollisesti vakiodataa muihin tiedostoihin.
- Ohjelmalla piti pystyä avaamaan formaatti eli xml-tiedosto, joka sisältää tunnisteen tiedostorakenteet sekä salausavaimet.

- Ohjelmalla piti voida lukea ja formatoida tunniste.
- Ohjelma piti pystyä sulkemaan niin, että tunnisteiden kirjoitusta voi jatkaa viimeisestä kirjoitetusta sarjanumerosta, jos tilaus jää kesken.
- Ohjelman piti kyetä ottamaan automaattisesti yhteys Access 9CD -laitteeseen ja Linx-kirjoittimeen ohjelman käynnistyksen yhteydessä.
- Ohjelmalle määritettiin, että tilauksen luonnin yhteydessä syötetään tunnisteiden lukumäärä ja sarjanumeron alkunumero.
- Peruskäyttäjälle ei näy tunnisteiden salausavaimia.
- Virheellinen tunniste koodataan automaattisesti uudelleen.
- Hyppysarjojen eli erikseen määrätyn datan kirjoitus on mahdollista. Hyppysarjat luetaan Excel-tiedostosta.
- Epäonnistuneen tunnisteiden virhetyyppi ilmaistaan lokitiedostossa.

## **4.2 Asiantuntijahaastattelu**

Työssä kävi ilmi suunnittelun ja dokumentaation tärkeys projektin onnistumiselle, joten päätin etsiä haastatteleamalla enemmän informaatiota epäselviksi jääneisiin asioihin. Tarkoituksena oli kartoittaa tarkemmin Idesco Oy:n laatujärjestelmää ja hyvää suunnittelukäytäntöä sellaisen henkilön avulla, joka oli näitä työssään pidempään hyödyntänyt.

Esitin kirjallisena neljä kysymystä Erno Väyryselle, joka toimi Idesco Oy:ssä opinnäytetyön kirjoitushetkellä Software Engineer -työnimikkeellä. Haastattelua varten pyysin luvan käyttää vastauksia ja henkilötietoja opinnäytetyössäni. Asiantuntijahaastattelu on liitteenä 1. (9.)

Haastattelussa ilmeni, että ohjelmistosuunnittelu on vain yksi osa Idescon laatujärjestelmäkokonaisuutta. Tällöin dokumentaatiovaiheet noudattavat tiettyjä periaatteita, jotta ohjelmistosuunnittelu palvelisi projektien etenemistapaa suunnittelusta tuotantoon. Hyvästä suunnittelukäytännöstä nousi mielestäni tärkeimpänä modulaarisuus eli tapa ohjelmoida niin, että koodi on jatkossa helposti uudelleen käytettävä. Tämä edellyttää mm. riittävää kommentointia koodissa ja ohjelman pilkkomista komponentteihin.

Ohjelmistoa koskevat laatujärjestelmävaiheet:

- Vaatimusmäärittely (asiakkaan kanssa määritellyt tuotteen ominaisuudet)
- Embedded SW plan / PC SW plan (yksityiskohtainen suunnitelma ohjelmiston rakenteesta aikatauluarvioineen ja -seurantoineen)
- Ohjelmiston nimeäminen
- Testisuunnitelma (räätälöidään valmiin testisuunnitelman päälle tuotteen ominaisuuksien mukaisesti)
- Ohjelmistokatselmointi (käydään ohjelmisto läpi toisen ohjelmistosuunnittelijan kanssa ja kirjataan katselmointipöytäkirjaan havainnot kustakin katselmoitavasta aiheesta). (9).

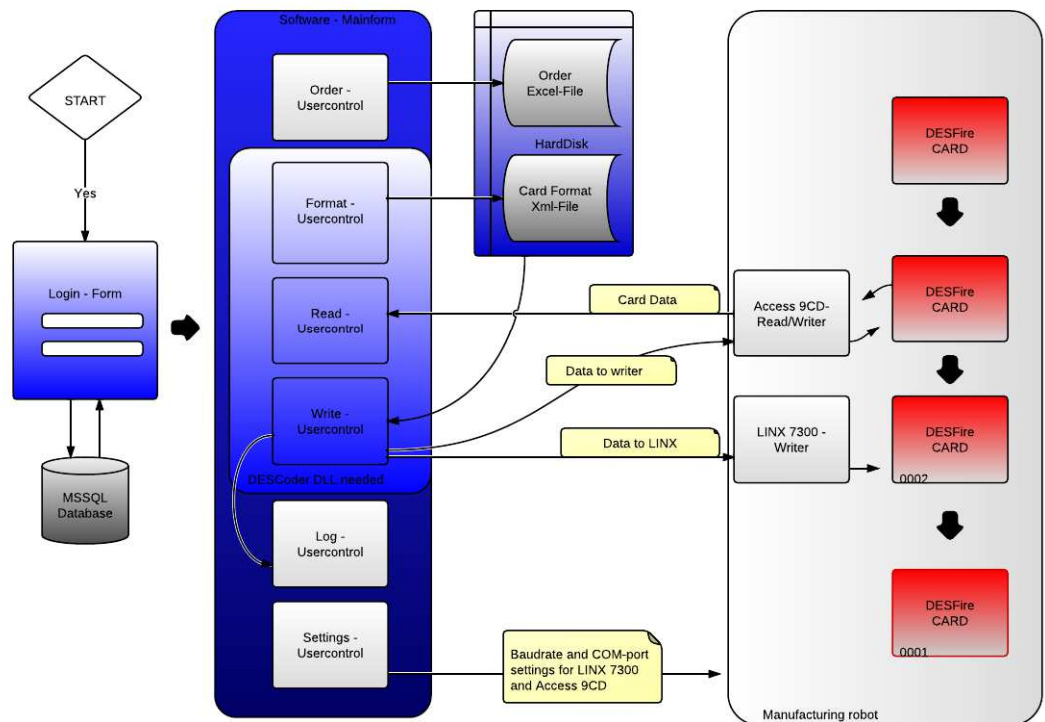
## **5 TOTEUTUS**

### **5.1 Ohjelman kehitysympäristö**

Ohjelmiston kehitysympäristönä toimi Microsoft Visual Studio 2008, .Net Framework 3.5 ja ohjelmointikielenä C-Sharp eli C#. C# on oliopohjainen ohjelmointikieli. Se on suunniteltu täyttämään edellisten (mm. C-kieli) ohjelmointikielten puutteet ja lisäämään hyödyllisiä ominaisuuksia (C++:n tehokkuus ja Java-kielten helppokäyttöisyys). (10, s. 5.) Ohjelman käyttämät DESCoder-kirjastot olivat C++-pohjaisia. DESCoder-kirjastot sisälsivät osan kirjoitus- ja lukufunktioiden toiminnalle välttämättömästä toiminnallisuudesta, sekä toimivat lukijalaitteen ja käyttöliittymän rajapinnassa yleisesti. DESCoder-kirjastot hoitivat myös formaatin alustamisen sellaiseen muotoon, että kortille kirjoitus on mahdollista ohjelman toiminnan kannalta.

### **5.2 Ohjelman yleiskuvaus**

Ohjelman pääkäyttötarkoitus on initialisoida MIFARE DESFire EV1 -tunnisteita automaattisesti Idesco Oy:n tuotantorobotilla. Tämä tarkoittaa käytännössä sitä, että tunnisteelle luodaan ensiksi formaatti. Formaatti sisältää MIFARE DESFire EV1 -spesifikaation mukaisen juurirakenteen, jonka päälle luodaan halutut applikaatiot sekä tiedostot. Formaattia luotaessa päätetään myös tunnisteen salausavaimet, joita ei näytetä enää ohjelman muissa suoritusvaiheissa. Tämän jälkeen formaatin sisältämiin tiedostoihin kirjoitetaan vakiodataa, juoksevaa sarjanumeroa tai hyppysarjoja. (kuva 5.)



KUVA 5. Yleiskuvaus ohjelman toiminnasta

### 5.3 Ohjelman toiminnallisuus

Käynnistettäessä ohjelma aukaisee kirjautumisikkunan, johon syötetään käyttäjätunnus ja salasana. Autentikoinnin onnistuessa ohjelmaan avautuu pääikkuna, jossa on tuki seuraaville toiminnoille:

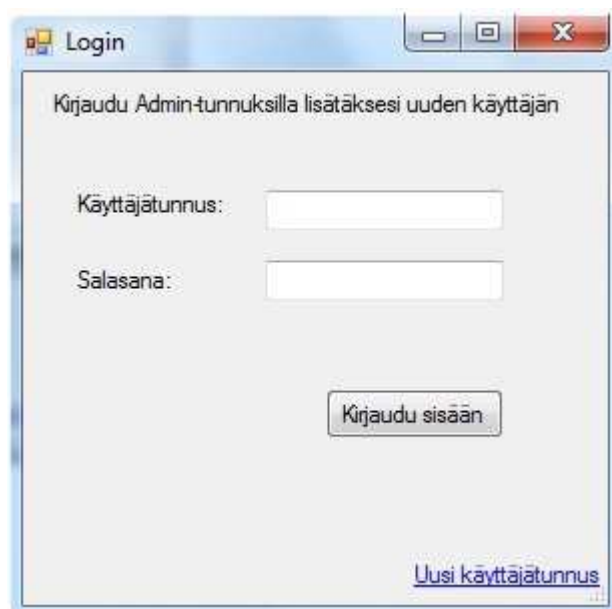
- tilauksen luonti
- formaatin luonti
- asetusten muuttaminen
- tunnisteen initialisoinnissa syntyneiden lokien tarkistus
- tunnisteen lukeminen
- tunnisteen kirjoittaminen.



Autentikoinnin jälkeen pääikkunan avautuessa ohjelma ottaa yhteyden tuotantorobotin Access 9CD -laitteeseen ja ilmoittaa yhteyden statuksen, COM-portin numeron ja tiedonsiirtonopeuden (ns. baudinopeus) alareunassa olevassa status-kentässä.

### 5.3.1 Kirjautuminen

Ohjelman käynnistymisen jälkeen ohjelma avaa kirjautumisikkunan (kuva 6). Kirjautumisikkunassa on mahdollisuus kirjautua käyttöliittymään pääkäyttäjänä tai normaalikäyttäjänä. Pääkäyttäjänä on mahdollisuus avata ja tehdä uusia formaatteja. Mikäli pääkäyttäjä haluaa lisätä MSSQL-tietokantaan uuden normaalitason käyttäjän ohjelmalle, alareunassa on linkki Uusi käyttäjätunnus, jota valitsemalla ohjelma pyytää kirjautumaan pääkäyttäjän tunnuksilla. Tämän jälkeen avautuvat tarvittavat kentät uuden käyttäjän luomiselle ja hyväksymiselle.



KUVA 6. Kirjautumisikkuna.

### 5.3.2 Pääikkuna

Onnistuneen kirjautumisen jälkeen ohjelma avaa pääikkunan. Käyttöliittymä avaa välilehdet pääikkunan (kuva 7) sisällä olevaan paneeliin. Pääikkunassa on seuraavat välilehdet:

- Tilaus
- Formaatti
- Asetukset
- Loki
- Luku
- Kirjoitus.

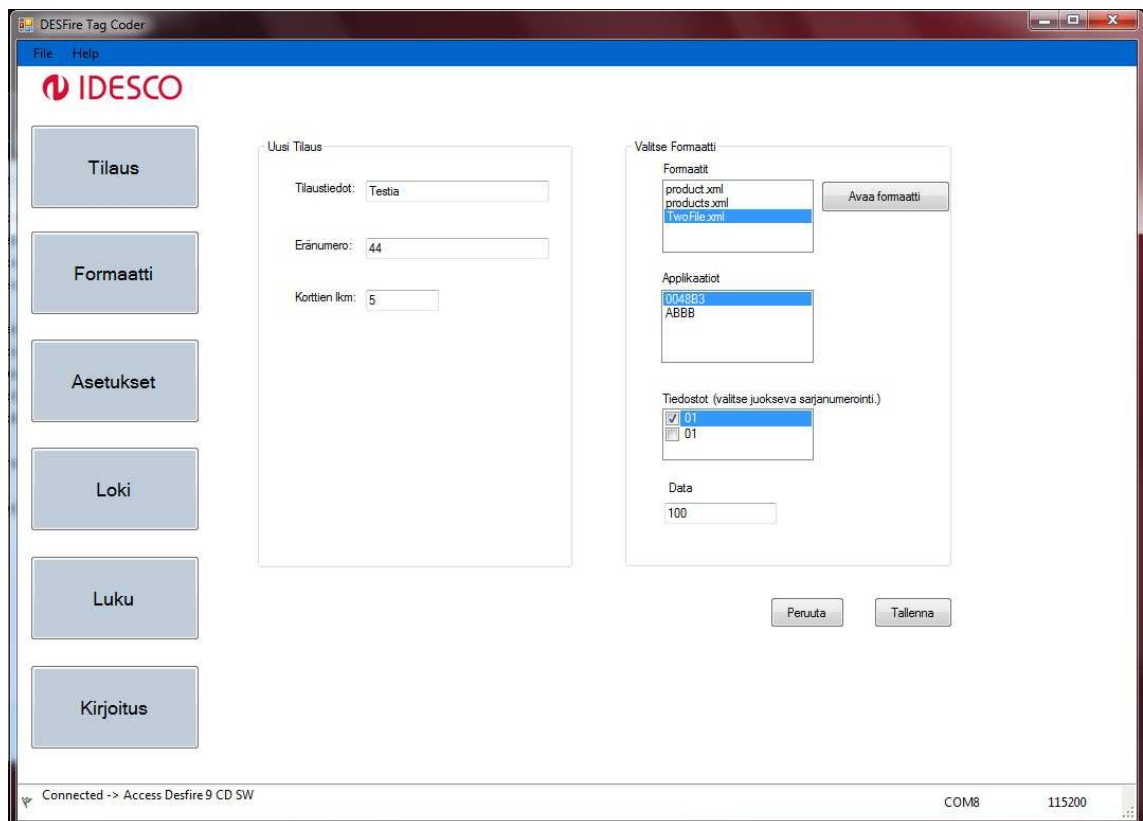


*KUVA 7. Pääikkuna*

### 5.3.3 Tilaus-välilehti

Tilaus-välilehdellä (kuva 8) siirretään uuden tilauksen tiedot Excel-tiedostoon. Excel-tiedostoon tallennetaan seuraavat muuttujat:

- tilaustiedot (samalla myös Excel-tiedoston tiedostonimi)
- eränumero
- kirjoitettavien tunnisteen lukumäärä
- kirjoituksessa käytettävä formaatti
- juoksevan sarjanumeron tai hyppysarjojen kohdetiedostot.

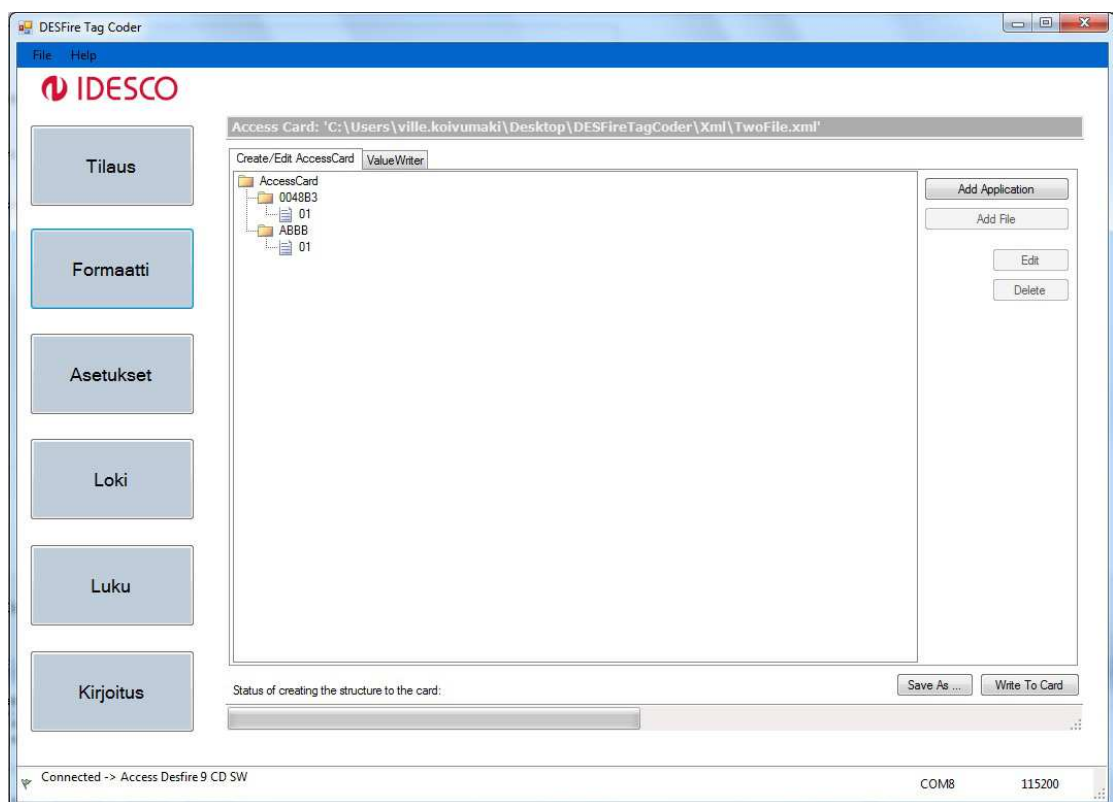


KUVA 8. Tilaus-välilehti

### 5.3.4 Formaatti-välilehti

Formaatti-välilehdellä (kuva 9) voidaan avata formaatteja tai luoda uusi formaatti. Ohjelma päästää Formaatti-välilehdelle vain pääkäyttäjän. Täällä rakennetaan tunnisteelle juurirakenne sekä lisätään applikaatiot ja tiedostot. Välilehdellä on mahdollista kirjoittaa tiedostoille vakiodataa, joka voi olla HEX- tai ASCII-merkkejä.

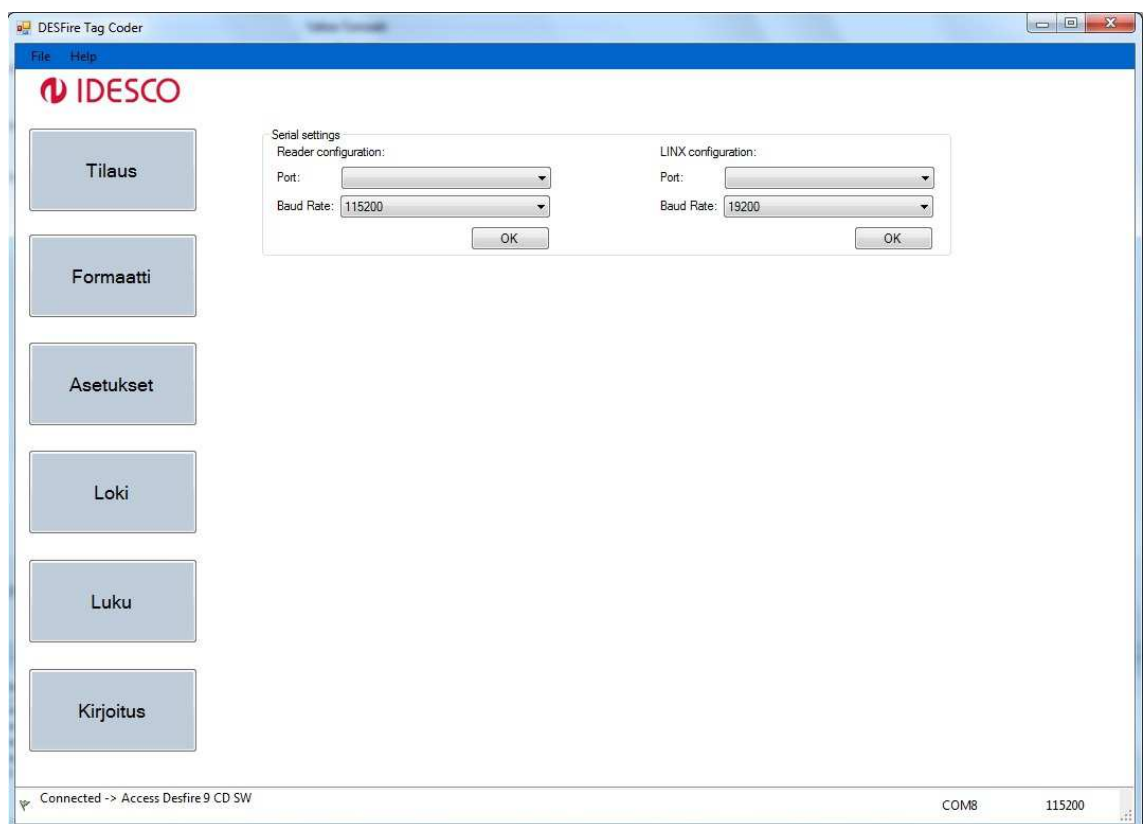
Uuden formaatin luonnissa määritetään PICC-master-avain tunnisteelle sekä valitaan käytettäväksi joko 3K3DES- tai AES-salausalgoritmi. Applikaatioille voidaan antaa 14 avainta sekä määrittää niiden salausalgoritmi, joka on riippumaton tunnisteiden PICC-master-avaimen salauksesta. Tunnisteiden pääavaimen salaus voi siis olla AES:llä tehty ja applikaatioiden 3K3DES:llä tai toisin päin.



KUVA 9. Formaatti-välilehti.

### 5.3.5 Asetukset-välilehti

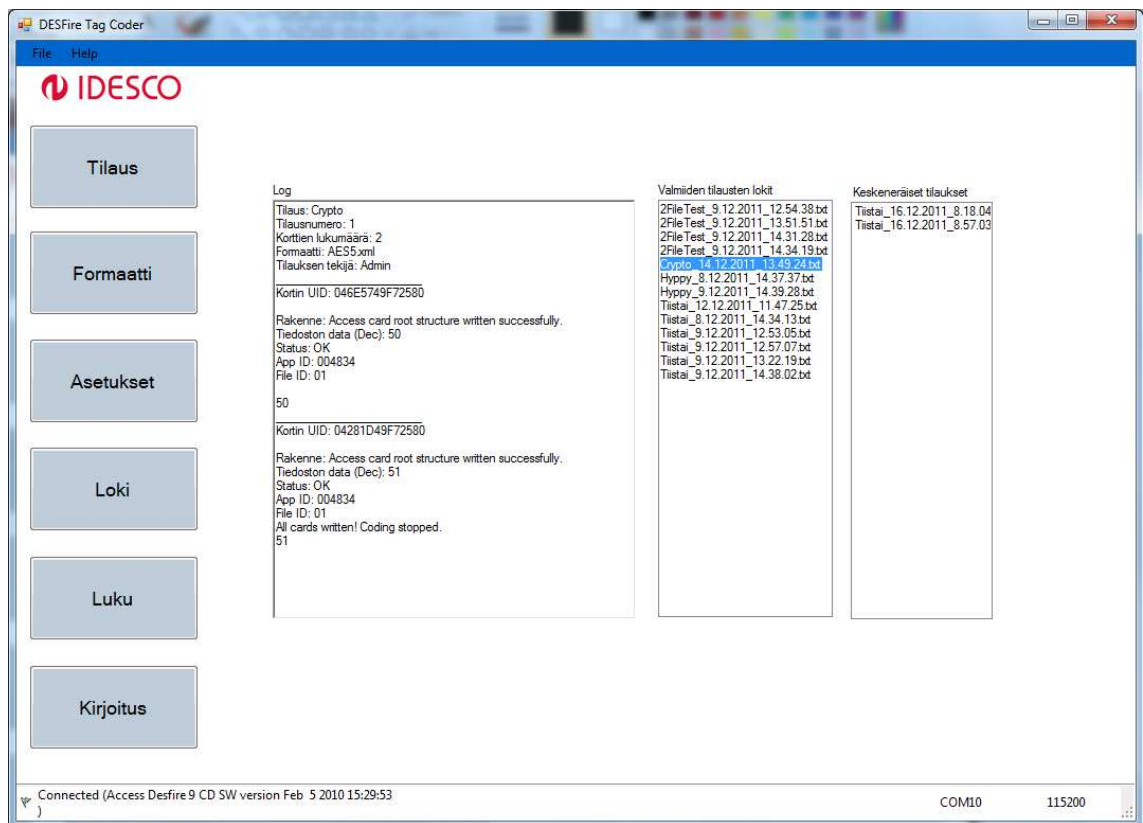
Asetukset-välilehdeltä voidaan muuttaa Access 9CD -laitteen sekä Linx-kirjoittimen sarjaportti tai tiedonsiirtonopeus manuaalisesti. Ohjelma ottaa käynnistyessään yhteyden Access 9CD -laitteeseen määritettyillä asetuksilla, mutta mikäli ne ovat vaihtuneet tai käytetään eri tiedonsiirtonopeudella toimivaa lukijalaitetta, ne on mahdollista myös muuttaa manuaalisesti.



KUVA 10. Asetukset-välilehti

### 5.3.6 Loki-välilehti

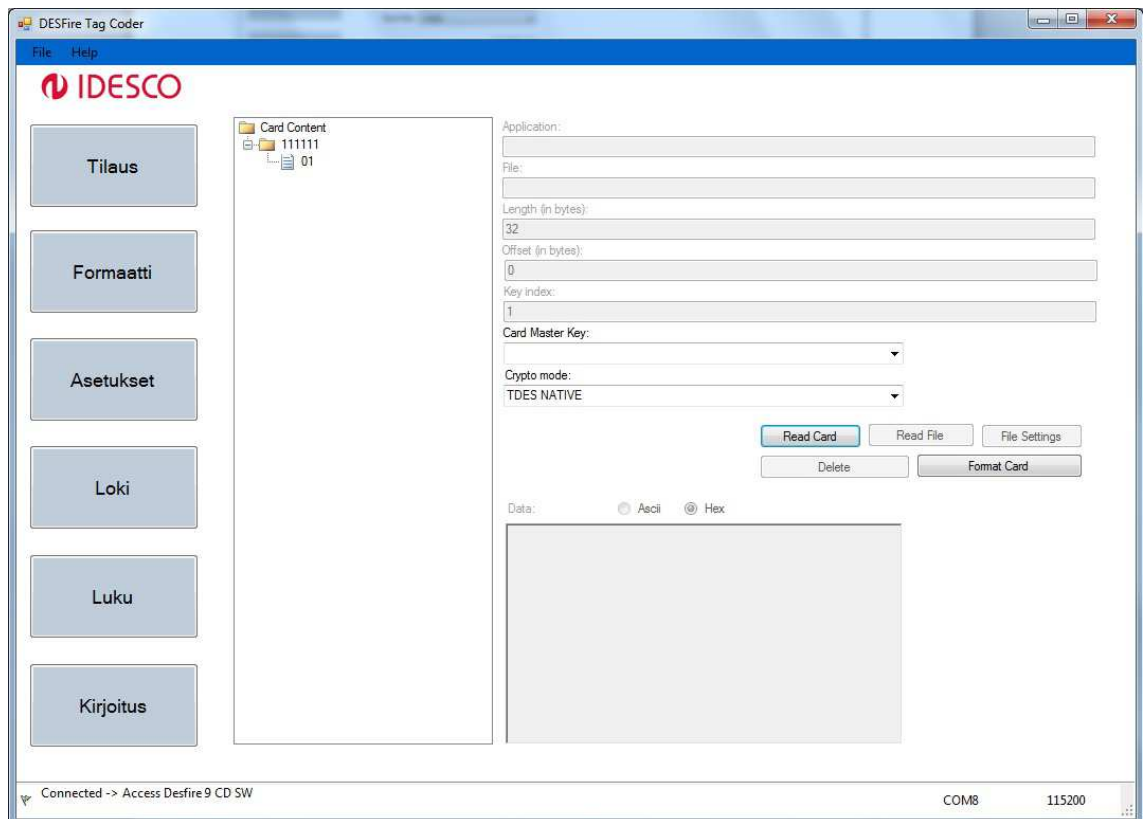
Vaatimusmäärittelyssä kirjattiin, että ohjelman pitää voida kirjoittaa lokia sessioista, joissa tunnisteita kirjoitettiin. Lokiin kirjoitettiin tieto kirjoituksen onnistumisesta sekä kirjoitettiin tunnisteiden UID-numero, kirjoitetun applikaation nimi, kirjoitettu data sekä tiedoston nimi. Mikäli ohjelma tai tietokone sammutetaan kesken tunnisteiden kirjoituksen, ohjelma tarkastaa lokista kesken jääneet tilaukset ja jatkaa tunnisteiden tekoa viimeisestä kirjoitetusta sarjanumerosta tai hyppysarjasta. (kuva 11.)



KUVA 11. Loki-välilehti

### 5.3.7 Luku-välilehti

Luku-välilehdeltä (kuva 12) voidaan lukea tunniste yksi kerrallaan. Lukeminen vaatii, että käyttäjä tietää tunnisteiden pääavaimen ja luettavan applikaation avaimen ja avain-indeksin. Avain-indeksillä määritetään avaimelle kuuluvat oikeudet. Luku-välilehdeltä on myös mahdollista formatoida tunniste. Formatointi poistaa kortilta applikaatiot ja tiedostot. Lisäksi se alustaa tunnisteiden PICC-master-avaimen ja juurirakenteen.



KUVA 12. Luku-välilehti

### 5.3.8 Kirjoitus-välilehti

Kirjoitus-välilehdellä (kuva 13) suoritetaan tunnisteiden automaattinen ohjelmointi. Avaa tilaus –painiketta painamalla valitaan Excel-tiedosto, joka

sisältää halutun tilauksen tiedot. Tiedot siirtyvät Kirjoitus-välilehden kenttiin, jonka jälkeen on mahdollista alkaa kirjoittaa tunnisteita.

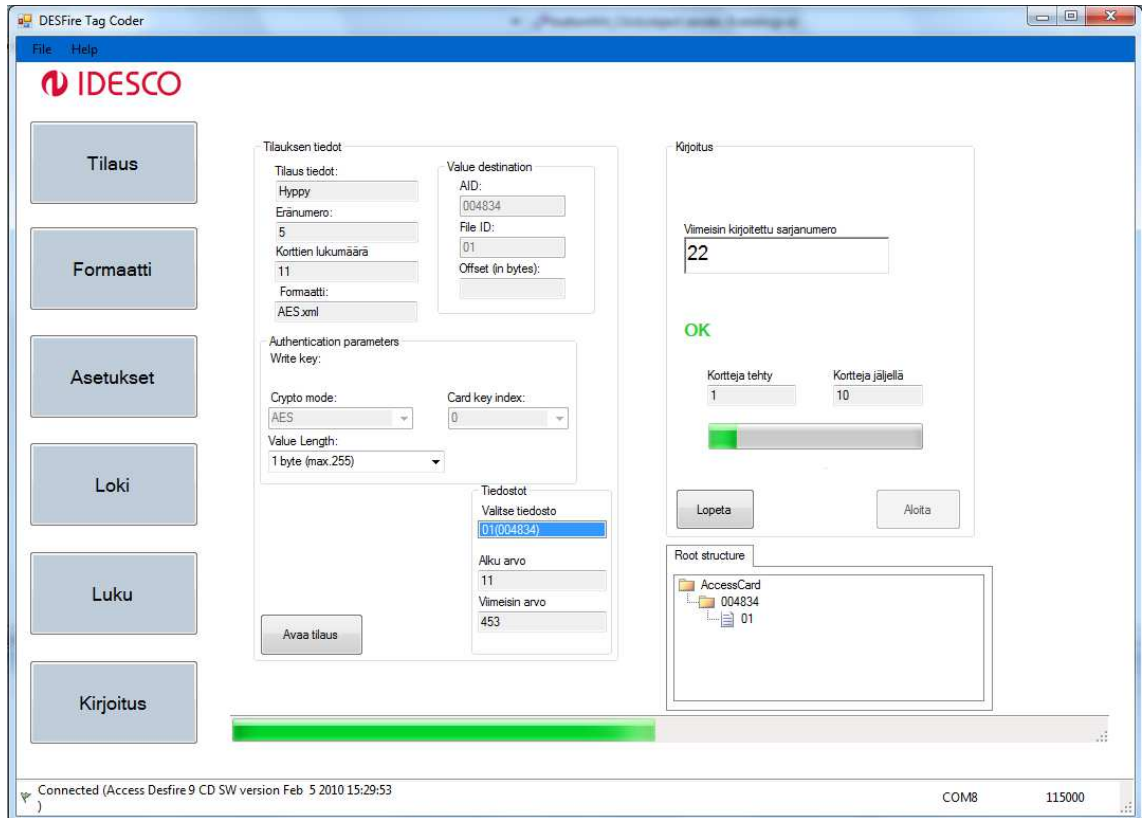
Mikäli tunnisteiden ohjelmoinnissa tapahtuu virhe, ohjelma siirtää tunnisteiden virhetiedot lokiin. Tuotantorobotti siirtää puolestaan tunnisteiden hylkyyn, mikäli tunnisteiden ohjelmointi ei onnistu tuotantorobotissa valitun ajan kuluessa. Jos tunnisteelle kirjoittaminen onnistuu, ohjelma lähettää signaalin tuotantorobotille, joka siirtää tunnisteiden liukuhihnalla mustesuihkutulostimelle, joka merkitsee tunnisteelle sarjanumeron.

Mikäli kirjoitetaan hyppysarjoja, ohjelma lähettää tunnisteelle ohjelmoitavan hyppysarjan mustesuihkutulostimelle, joka kirjoittaa puolestaan sen eikä automaattisesti sarjanumeroa. Hyppysarjat sijaitsevat tilauksen Excel-tiedostossa.

Jos ohjelma lopetetaan tai pysäytetään, ohjelmalla on mahdollista jatkaa keskeneräisten tilausten tekoa aukaisemalla sitä vastaava lokitiedosto. Kun tilauksen viimeinen kortti on onnistuneesti tehty, ohjelma ilmoittaa siitä aukaisemalla MessageBox-ikkunan. Tilauksen sekä senhetkisen tunnisteiden ohjelmoinnin edistymistä ohjelma ilmaisee ProgressBar-kentissä.

Tilauksen luonnin aikana muut välilehdet ovat lukittuina, kunnes tilaus keskeytetään tai kaikki tunnisteet ovat valmiita. Mikäli ohjelmalla formatoidaan tunnisteita automaattisesti, pitää käyttäjän tietää tunnisteiden PICC-master-avain sekä käytetty salausalgoritmi.





KUVA 13. Kirjoitus-välilehti

## 6 TESTAUS

Ohjelman testausta suoritettiin tuotantorobotilla sekä ohjelmistokehitystyökalut sisältävällä tietokoneella, jossa oli Access 9CD -lukija. Isoimmat muutokset testattiin tuotantorobotilla ja ohjelmoinnissa syntyneitä pieniä muutoksia tietokoneella, johon oli kytkettynä pelkästään Access 9CD -lukija.

Tuotantorobotilla tehty testaus oli hieman laajempaa, koska silloin pystyttiin testaamaan ohjelman kaikki ominaisuudet toisin kuin tavallisella tietokoneella tehdyissä testeissä. Tuotantorobotti sisälsi mm. eri sarjaporttiasetukset sekä mahdollisuuden testata myös Linx-kirjoitinta.

Testauksessa selvisi muutamia tapoja toteuttaa asioita eri tavalla kuin olimme vaatimusmäärittelyssä suunnitelleet. Esimerkkinä mainittakoon, että robotilta tuleva signaali, joka ilmoitti sen tiedon, milloin tunnisteelle voi kirjoittaa tai milloin se on luku- ja kirjoitusetäisyyden päässä Access 9CD -laitteesta, osoittautui hyödyttömäksi. Ohjelma sai tiedon jo laitteelta itseltään, milloin tunniste on kirjoitusetäisyyden päässä Access 9CD:stä, jolloin robotilta tulevan signaalin kuuntelua ei tarvittu. Tämä itse asiassa paransi ohjelman toimintaa ja luotettavuutta.

Testausvaiheessa saatiin myös idea toteuttaa tunnisteiden automaattisen kirjoitustoiminnon lisäksi tunnisteiden automaattinen formatointi. Testauksessa kirjoitetut testitunnisteet piti aina formatoida yksi kerrallaan, ja se osoittautui vaivalloiseksi ja aikaa vieväksi. Automaattinen formatointiominaisuus mahdollisti tunnisteiden formatoinnin nopeasti.

## 7 POHDINTA

Työn alkaessa tietoni RFID-tekniikasta olivat perustasolla. En ollut perehtynyt MIFARE DESFire EV1- spesifikaatioon ennen projektin alkua. Työn aihe vaikutti kuitenkin erittäin mielenkiintoiselta ja haastavalta, joten päätin tarttua projektiin. Olin edellisessä projektissa tehnyt käyttöliittymää RFID-tekniikkaa käyttävään kosketusnäytölliseen lukijaan, joten siitä oli valtavasti apua myös tämän projektin käyttämän teknologian ja sen toiminnan ymmärtämisessä.

Työ jakautui siten, että insinööriopiskelija Arto Hurskainen suunnitteli, toteutti ja auttoi liittämään ohjelmaan ne komennot, funktiot ja kirjastot joilla ohjattiin robotin toimintoja. Puolestaan minun osuuteni oli ohjelmoida käyttöliittymä sekä liittää siihen robotin ohjaus. Hän oli jatkuvasti apuna ohjelman testauksessa, suunnittelussa ja toteutuksessa. Mikäli projektia ei olisi jaettu tällä tavalla, olisi työmäärä ollut aivan liian suuri ja projektin suunnittelu ja toteutus olisi hidastunut merkittävästi.

Ensimmäiset 40 tuntia projektin alussa kuluivat projektin suunnitteluun ja tekniikan sekä MIFARE DESFire EV1 -spesifikaation opiskeluun. Nyt jälkeenpäin miettiessä voi todeta, että suunnitteluun olisi pitänyt käyttää vieläkin enemmän aikaa, sillä hyvin suunniteltu on puoliksi tehty etenkin ohjelmistokehityksessä. Suurimmat ongelmat liittyivätkin suunnittelussa ja vaatimusmäärittelyssä tekemättömiin asioihin. Suunnittelussa huomioimatta jääneet asiat ohjelmistossa hidastavat toteutusta merkittävästi, koska ohjelman toimintaa pitäisi suunnitella kokonaisuutena ja se olisi paras tehdä ohjelman suunnitteluvaiheessa projektin alussa.

Projektin läpivientiä helpotti merkittävästi se, että MIFARE DESFire EV1 -tunnisteiden ohjelmoimiseen projektissa käytettävälle Access 9CD -laitteelle oli jo valmiina ohjelmisto, jolla pystyi lukemaan ja kirjoittamaan tunnisteita yksi

kerrallaan. Koodia joutui tietysti muokkaamaan melko paljon, mutta silti se nopeutti projektia todella paljon, kun useita toiminnallisuuksia ei tarvinnut alusta lähtien itse suunnitella.

Työtä suorittaessani opin myös todella paljon C#:lla tehtävästä ohjelmistokehityksestä. Käyttötaitoni parani ohjelmistokehitystyökalujen käyttämisessä, joka mahdollisti niiden nopeamman ja monipuolisemman käytön.

Oppiminen ei jäänyt pelkästään ohjelmoinnin tai kehitystyökalujen käytön parempaan ymmärtämiseen, sillä MIFARE DESFire EV1 -tekniikka sisälsi paljon myös tietoturvaan liittyviä asioita, jotka projektissa oli pakko ottaa huomioon ja käsiteltäväksi. Tämä edesauttoi myös oppimista tietoturvasta ja johti syvempään alan tekniikoiden ymmärtämiseen, mistä olenkin erityisen tyytyväinen projektin oppimiskokemuksena.

Projektin aikana syntyi myös muutamia jatkokehitysideoita MIFARE DESFire EV1 -tunnisteiden käyttöä tai niiden kehitystä varten. Osa niistä ominaisuuksista, joita vaatimusmäärittelyssä ei alun perin esitetty, otettiin jopa käyttöön ohjelmassa. Esimerkkinä tällaisesta on tunnisteiden automaattinen formatointi. Kokonaisuutena olen projektiin erittäin tyytyväinen ja projektin tulokset syntyivätkin paljon helpommin kuin alussa näytti. Ne tuskastumisen hetket, joita syntyi projektin toteutusvaiheessa, kun yritin keksiä ratkaisua joihinkin ongelmiin, ratkesivat onneksi melko nopeasti ja projektissa pääsi etenemään ilman pahoja viiveitä.

Tulevaisuus RFID-tekniikkaan perustuvien sovellusten ja ratkaisujen käytössä näyttää positiiviselta. Ala oli vuonna 2011 edelleenkin kasvava ja markkinoille tulee jatkuvasti entistä monipuolisempia sovelluksia. Tämä tarjoaa kehittäjille varmasti uusia mahdollisuuksia sekä haasteita RFID-tekniikan parissa.

## LÄHTEET

1. RFID- tekniikan historia. 2012. Saatavissa: <http://www.rfidlab.fi/rfid-tekniikan-historia>. Hakupäivä 1.1.2012.
2. Seppä, Heikki 2009. Etätunnistusteknologian kehitys meillä ja maailmalla. Saatavissa: [www.tekes.fi/fi/document/27018/rfid\\_pdf.pdf](http://www.tekes.fi/fi/document/27018/rfid_pdf.pdf) Hakupäivä 16.01.2012.
3. Seppä, Heikki 2011. RFID- etätunnistus – mahdollisuudet ja uhat. Eduskunnan tulevaisuus valiokunnan julkaisu 9/2011. Saatavissa: [http://www.eduskunta.fi/triphome/bin/thw.cgi/trip?\\${BASE}=erekj&\\${HTML}=akxpdf&\\${SNHTML}=akxeiloydy&tunniste=TUVJ+9/2011](http://www.eduskunta.fi/triphome/bin/thw.cgi/trip?${BASE}=erekj&${HTML}=akxpdf&${SNHTML}=akxeiloydy&tunniste=TUVJ+9/2011) Hakupäivä 10.2.2012.
4. China Market and Industry Report 2008-2009. 2009. Saatavissa: <http://www.pday.com.cn/Uploads/ReportAttach/20090122023237.pdf> Hakupäivä 4.12.2011.
5. MIFARE DESFire EV1. 2010. Datalehti. Saatavissa: <http://www.nxp.com/documents/leaflet/75015782.pdf>. Hakupäivä 22.11.2011.
6. MIFARE Milestones. 2011. Saatavissa: [www.mifare.net/overview/history](http://www.mifare.net/overview/history). Hakupäivä 22.11.2011.
7. Linx 7300 Datasheet. 2009. Datalehti. Saatavissa: [http://www.linxglobal.com/uploadeddocuments/Datasheets/Linx\\_7300\\_Data\\_sheet\\_Final.pdf](http://www.linxglobal.com/uploadeddocuments/Datasheets/Linx_7300_Data_sheet_Final.pdf). Hakupäivä: 10.2.2012.

8. Access-9-CD---Ddatasheet. 2011. Datalehti. Saatavissa: [www.idesco.fi](http://www.idesco.fi). Hakusana: Access 9 CD. Hakupäivä 21.11.2011.
9. Väyrynen, Erno 2012. Software Engineer, Idesco Oy. Sähköpostihaastattelu. 19.01.2012.
10. Syrjälä, Santtu 2005. C# Pikakurssi. Kandidaattitutkielma. Saatavissa: [http://users.jyu.fi/~vesal/kurssit/winohj/csharp/seminaari05/CS\\_Pikakurssi.pdf](http://users.jyu.fi/~vesal/kurssit/winohj/csharp/seminaari05/CS_Pikakurssi.pdf). Hakupäivä 22.11.2011.

## **LIITTEET**

Erno Väyrysen sähköpostilla 19.01.2012 toteutetun asiantuntijahaastattelun kysymykset ja vastaukset:

### **Mistä asioista Idesco Oy:n laatu järjestelmä koostuu silloin kun tuotetaan uutta ohjelmaa?**

Ohjelmistosuunnittelu on osa Idescon laatu järjestelmäkokonaisuutta. Ohjelmistoihin liittyvä laatuajattelu tarkoittaa sitä, suunnitteluun käytetään laatu järjestelmässä kuvattua etenemistapaa määrittelystä tuotantoon viemiseen.

Ohjelmistoa koskevat laatu järjestelmävaiheet:

- Vaatimusmäärittely (asiakkaan kanssa määritellyt tuotteen ominaisuudet)
- Embedded SW plan / PC SW plan (yksityiskohtainen suunnitelma ohjelmiston rakenteesta aikatauluarvioineen ja -seurantoineen).
- Ohjelmiston nimeäminen
- Testisuunnitelma (räätälöidään valmiin testisuunnitelman päälle tuotteen ominaisuuksien mukaisesti).
- Ohjelmistokatselmointi (käydään ohjelmisto läpi toisen ohjelmistosuunnittelijan kanssa ja kirjataan katselmointipöytäkirjaan havainnot kustakin katselmoitavasta aiheesta).

### **Mitä laatu järjestelmällä saavutetaan ohjelmistotuotannossa?**

Laatu järjestelmä on hyvä työkalu pitämään huoli siitä, että laadun kannalta tärkeät asiat käydään aina läpi. Laadukasta työskentelyä ei kuitenkaan saavuteta pelkillä järjestelmillä, vaan työntekijöiden on ymmärrettävä, sisäistettävä ja motivoitettava laatuajatteluun. Laatu järjestelmän aktiivinen käyttö palvelee tätä. Aktiivinen käyttö tarkoittaa sitä, että menetelmiä noudatetaan sekä kannustetaan antamaan kehitysehdotuksia mikäli huomataan, että asioita voidaan hoitaa jotenkin paremmalla tavalla.

**Mitä asioita sisältyy hyvään suunnittelukäytäntöön?**

Hyvä suunnittelukäytäntö ja laatuajattelu ovat pitkälti sama asia. Oma suosikkini on kokonaiskuvan hahmottaminen. Eli kun uutta tuotetta suunnitellaan, on tärkeää nähdä miksi tuote on olennainen ja määrittellä sen ominaisuudet käyttötarkoituksen mukaan. Idescolla on usein sellainen tilanne, että olemassaolevia ratkaisuja eli valmiiksi toteutettua ohjelmakoodia on paljon saatavilla. Tuote kannattaa luonnollisesti suunnitella siten, että kerran tehtyä työtä eli koodia voisi mahdollisimman vaivattomasti käyttää muuallakin. Hyvään ohjelmointikäytäntöön kuuluvat mm. modulaarisuus eli monikäyttöisyys, runsas kommentointi ja selkeät rakenteet.

**Miten vaatimusmäärittelystä saadaan riittävän tarkka ja millä tavoin se helpottaa jatkovaiheita?**

Vaatimusmäärittely saadaan tarkaksi kysymällä asiakkaalta yksityiskohtaisia kysymyksiä niin kauan, että vaatimusmäärittely on riittävän tarkka. On myös olennaista saada asiakkaan hyväksyntä kootuille ominaisuuksille lähettämällä valmis vaatimusmäärittely asiakkaalle. Idescolla on käytössä vaatimusmäärittelypohja, jonka jokainen kohta täyttämällä saadaan useimmiten riittävän tarkka kuvaus tuotteen ominaisuuksista. Mikäli tarvetta tarkennuksille on, on vaatimusmäärittelyä syytä laajentaa ja jatkaa asiakkaan kanssa keskustelua.

Vaatimusmäärittelyvaiheessa on hyvä käyttää useampia eri toimenkuvissa työskenteleviä henkilöitä mahdollisimman hyvän kokonaiskuvan saavuttamiseksi.

Kun vaatimukset ovat aikaisessa vaiheessa selkeät, työn suunnittelun aikana ei tarvitse käyttää aikaa ominaisuuksien pohtimiseen.