



Aleksi Palosaari

YRITYKSEN TUOTEKEHITYSPROSESSIEN KEHITTÄMINEN

# **YRITYKSEN TUOTEKEHITYSPROSESSIN KEHITTÄMINEN**

Aleksi Palosaari

Opinnäytetyö

19.5.2011

Tietotekniikan koulutusohjelma

Oulun seudun ammattikorkeakoulu

Koulutusohjelma

Tietotekniikan koulutusohjelma

Opinnäytetyö

Insinöörityö

Sivuja

57

+

Liitteitä

+

6

Suuntautumisvaihtoehto

Sulautetut järjestelmät

Aika

2011

Työn tilaaja

M-Technology Oy

Työn tekijä

Aleksi Palosaari

Työn nimi

Yrityksen tuotekehitysprosessien kehittäminen

Avainsanat

ohjelmistotuotanto, ohjelmistosuunnittelu, tuotekehitys, dokumentointi

Tässä opinnäytetyössä perehdyttiin M-Technology Oy:n tuotekehitysprosesseihin sekä toimintaprosessien dokumentoimiseen. Tuotekehitystä tapahtuu M-Technology Oy:n omaan ohjelmistotuotteeseen WiseMasteriin. WiseMaster on toteutettu räätälöitynä useiden asiakkaiden erilaisiin tarpeisiin. Uusien ominaisuuksien myötä yrityksellä oli tarvetta kehittää suunnittelu-, testaus- sekä käyttöönottorutiineja.

Tämän insinöörityön tavoitteena oli tutkia ja yhtenäistää sekä edelleen kehittää jo M-Technologyssä olevia ohjelmistotuotannon eri osa-alueita. Työssä perehdyttiin kahteen todelliseen asiakastapahtumaan, joissa tehtiin asiakkaille heidän tarpeisiinsa sopivat mobiilitoiminnanohjausjärjestelmät. Asiakastapahtumat jakaantuivat vaatimustenmäärittely-, toteutus-, testaus- sekä käyttöönottovaiheisiin. Työssä tutustuttiin erivaiheiden olemassa oleviin ohjelmistoprosesseihin sekä peilattiin niitä M-Technologyn tarpeisiin.

Työ toteutettiin M-Technology Oy:lle, työ yhtenäisti yrityksen ohjelmistokehitysprosesseja ja työn tuloksena tulevaisuudessa kehitystyössä säästetään aikaa ja resursseja. Työ tuotti yritykselle valmiin ohjelmistokehitysprojektien dokumenttipohjan, joka otetaan käyttöön aina uusien ominaisuuksien kehitystyössä.

Degree programme

Thesis

Pages

+

Appendices

Information Technology

57

+

6

Line

Embedded Systems

Date

2011

Comissioned by

M-Technology Oy

Author

Aleksi Palosaari

Thesis title

The company's improvement in the processes in the field of product development

Keywords

Software Engineering, Software Design, Product Development

This thesis work explored product development processes and also documenting the action processes of M-Technology Oy. Product development occurs to M-Technology's own software product WiseMaster. WiseMaster is tailor produced to meet several needs of different customers. Through new features, the company had a need to develop designing-, testing- and introduction routines.

The aim of this engineer work was to research, integrate and further develop different sectors of M-Technology's existing software engineering. In the work the focus was on two real customer events where mobile resource planning systems were produced to meet each customers' needs. Customer events were divided to different phases. These were defining the demand, execution, testing and initialization stages. The work familiarized into the existing software processes of each stage and compared them to M-Technology's needs.

Thesis work was executed to M-Technology Oy which integrated company's software development processes. Through this work, development work will save both time and resources in the future. Thesis produced a ready software development projects' document base which can be implemented when developing new features.

# SISÄLLYS

TIIVISTELMÄ.....	3
ABSTRACT.....	4
SISÄLLYS.....	5
1 JOHDANTO .....	8
2 OHJELMISTOTEKNIikka .....	9
2.1 Ohjelmistotekniikka arjessa .....	9
2.2 Ohjelmistotyypit .....	10
2.3 Määrittely ja vaatimukset .....	11
2.3.1 Ohjelmiston määrittely .....	11
2.3.2 Ohjelmistoille asetetut vaatimukset .....	12
2.4 Ohjelmistotuotannon ongelmat.....	13
3 OHJELMISTOTUOTANTO .....	14
3.1 Ohjelmistotuotannon vaiheet .....	15
3.1.1 Esitutkimus .....	15
3.1.2 Määrittely.....	15
3.1.3 Suunnittelu .....	16
3.1.4 Toteutus .....	16
3.1.5 Testaus.....	16
3.1.6 Ylläpito.....	17
3.2 Elinkaarimallit .....	18
3.2.1 Vesiputousmalli .....	18
3.2.2 Evo-malli.....	19
3.2.3 Protoilumalli.....	19
3.3 Ketterä ohjelmistokehitys.....	20

3.3.1	Extreme Programming, XP .....	20
3.3.2	Scrum .....	21
3.3.3	RUP (Rational Unified Process) .....	21
4	JÄRJESTELMIEN MALLINTAMINEN UML-KIELELLÄ.....	23
4.1	UML.....	23
4.2	Käyttötapauskaavio .....	24
4.3	Luokkakaavio.....	25
4.4	Oliokaavio.....	26
4.5	Sekvenssikaavio.....	26
5	M-TECHNOLOGY OY:N WISEMASTER-TUOTTEET.....	28
5.1	WiseMaster .....	29
5.2	WiseMasterMobiili .....	29
5.3	WiseMasterPC .....	30
5.4	WiseMasterPCClient .....	31
5.5	WebWise.....	31
5.6	WiseMaster Huolto .....	32
5.6.1	Työmääräyksen luonti .....	33
5.6.2	Työmääräyksiä seuranta .....	34
5.6.3	Työnohjaus päätelaitteella.....	34
5.6.4	Kertymän hyväksyminen ja laskuttaminen .....	35
5.7	WiseMaster Tilaus.....	36
5.8	Mukautetut järjestelmät .....	38
6	OHJELMISTOPROSESSIN LUONTI .....	39
6.1	Vaatimusmäärittely .....	39
6.2	Toteutusvaihe .....	41
6.3	Testausvaihe .....	41

6.4	Käyttöönotto .....	42
6.5	Esimerkkitapaus .....	42
6.5.1	Vaatimusmäärittely .....	42
6.5.2	Toteutusvaihe .....	49
6.5.3	Testaus .....	51
6.5.4	Käyttöönotto .....	52
7	TYÖN ARVIOINTI .....	53
7.1	Käytetyt menetelmät .....	53
7.2	Esille tulleet ongelma tilanteet .....	54
8	YHTEENVETO .....	55
	LÄHTEET .....	56
	LIITTEET	
Liite 1.	Sekvenssikaavio esimerkki, Lujateko Oy	
Liite 2.	Esimerkki toteutusvaiheesta, Lujateko Oy	
Liite 3.	Esimerkki poikkeamista, Lujateko Oy	
Liite 4.	Esimerkki testaussuunnitelmasta, Lujateko Oy	

# 1 JOHDANTO

Tämän opinnäytetyön tavoitteena on kehittää oululaiselle ohjelmistoalan yritykselle M-Technologylle valmis ohjelmistoprosessi joka pitää sisällään dokumenttipohjia uusia tuotekehitysprojekteja varten. Yrityksessä ei ole aiemmin ollut tiettyä yksiselitteisesti määriteltyä tapaa, jolla uuden ominaisuuden ja kokonaisuuden tekemistä olisi lähdetty viemään eteenpäin. Työn aihe muotoutui yrityksen tarpeesta saada tietty toimintamalli projektien läpivientiin sekä dokumentaatio jo valmistuneista ohjelmistoprojekteista. Dokumenttipohjaa työstettiin useaan otteeseen yhdessä yrityksen ohjelmistoprojekteihin keskeisesti osallistuvien henkilöiden kanssa.

Tavoitteena työssä oli saada aikaiseksi mahdollisimman tehokas ja riittävän laaja malli tuotekehityksen neljälle eri osa-alueelle. Dokumenttipohjat toimivat suunnittelun, toteutuksen, testauksen sekä käyttöönoton runkona ja samalla muistilistana siitä mitä kaikkea uudessa ohjelmistoprojektissa pitää ottaa huomioon. Dokumentin jokainen osa-alue on muokattu jo tunnettuja ohjelmistojen suunnittelu-, toteutus- ja testausmalleja hyväksikäyttäen, kuitenkin niin että yrityksen resurssit ja tarpeet ovat huomioitu laajuudessa sekä eri osa-alueiden syvällisyydessä.

Toisena tavoitteena työssä oli oppia tuntemaan ohjelmistokehityksen koko elinkaari sekä tosielämän realiteetit tiukassa yritysmaailmassa. Työssä oppi tuntemaan syvällisesti kaikki yleisimmät ohjelmistokehitys mallit sekä pääsi tutustumaan tuotekehitysprosessin kaikkiin eri vaiheisiin, aina asiakkaan tapaamisesta määrittelyvaiheesta lopullisen ohjelmistoversion luovutukseen sekä koulutuksessa kouluttajana toimimiseen.



## 2 OHJELMISTOTEKNIikka

Ohjelmistotekniikan historia on verrattain nuori muihin tekniikan aloihin nähden. Historia alkaa vasta noin 1960-luvulta, jolloin tietokoneiden määrä oli hyvin pieni sekä ohjelmistot suppeita. Ohjelmistot, jos niitä enää nykyajan määritysten mukaan voidaan ohjelmiksi kutsua, olivat pieniä yhden käyttäjän sovelluksia. Sen jälkeen ohjelmistotekniikan kehitys on ollut hurjaa varsinkin muihin tekniikan aloihin verrattuna. Jo 1970-luvulle mentäessä tietokoneiden määrä oli moninkertaistunut, ohjelmistot laajentuneet useiden käyttäjien laajuisiksi sekä tietokantojen käyttö yleistynyt. Jopa ensimmäiset ohjelmistotalot aloittivat toimintansa. 1980-luvun loppuun mennessä tietokoneiden fyysisen koon pieneminen mahdollisti jo henkilökohtaisten tietokoneiden lisääntymisen. Käyttäjämäärien räjähdysmäinen lisääntyminen aiheutti ohjelmistopuolella laatuvaatimusten nousun sekä järjestelmien monimutkaistumisen ja sen takia oli kehitettävä uusia tekniikoita. Ohjelmistotekniikan kasvava trendi on jatkunut nykypäivään saakka ja tulee jatkumaan myös tulevaisuudessa. Suorituskyky ja laatuvaatimukset kasvavat vuosi vuodelta, jopa niin nopeasti ettei kehitys pysy enää perässä. (1.)

### 2.1 Ohjelmistotekniikka arjessa

2000-luvulla ohjelmistotekniikan yleistymisen on tehnyt siitä ihmisille arkipäiväisen asian, eikä ohjelmistotekniikan läsnäoloa enää edes tiedosteta. Ohjelmistotekniikka hallitsee arkea kotitöistä työpaikalle ja harrastuksiin. Ohjelmistotekniikkaa löytyy niin kodinkoneista, autoista, ajanottolaitteista kuin melkeinpä kaikesta, missä vain sähkö liikkuu. (2, s. 15.)

Jos nykypäivänä jouduttaisiin luopumaan ohjelmistosovelluksista, lamauttaisi se ihmisten jokapäiväisen toiminnan. Ihmiset vaativat aina vain nopeammin toimivia järjestelmiä ja se aiheuttaakin suurimmat ongelmat ohjelmistotuotannon puolella. Kehitys ei tahdo pysyä vaatimusten perässä ja näin laatu kärsii.

## 2.2 Ohjelmistotyypit

Jokapäiväisestä elämästäkin löytyvät ohjelmistot voidaan jakaa eri ryhmiin. Ryhmät luokitellaan ohjelmiston rakenteen ja käyttötarkoituksen mukaan. (2, s. 17.)

Varus- ja työkaluohjelmistot kuuluvat yleiskäyttöisiin ohjelmistoihin. Niitä voidaan löytää käyttäessä tietokonetta, niin käyttöjärjestelmänä sekä taulukkolaskentaohjelmistona. (2, s.17.)

Teknis-tieteellisiin ohjelmistoihin kuuluvat erilaiset mallinnus- sekä suunnitteluohjelmistot. Esimerkkeinä voidaan pitää CAD-ohjelmistoa, jolla mallinnetaan esineitä ja kokonaisia järjestelmiä. (2, s. 17.)

Asiantuntijaohjelmistoihin syötetään nimensä mukaan asiantuntevaa tietoa, jonka pohjalta ohjelma toimii. Ohjelmaa ja sen ”älyä” käytetään hyväksi esimerkiksi, kun halutaan määrittää asioita lainsäädäntöön ja muihin vallitseviin ominaisuuksiin pohjautuen. (2, s. 17.)

WiseMaster on hyvä esimerkki kaupallis-hallinnollisista ohjelmistoista. WiseMaster automatisoi yritysten toimintoja sekä kerää tietoa yrityksen sisällä raportoitavaan muotoon. Ohjelmistojen avulla yritysten toiminta nopeutuu ja kannattavuus kasvaa. (2, s. 17.)

Prosessinohjausjärjestelmät ovat teollisuuden käyttämiä tuotantoprosessien valvontaan ja ohjaamiseen tehtyjä ohjelmistoja. Näiden ohjelmistojen avulla pystytään automatisoimaan ja ylipäänsä suorittamaan erittäin monimutkaisia prosesseja. Esimerkiksi Outokummun Tornion tehtaalla on maailman pisin metallinjalostuslinjasto, joka toimii prosessinohjausjärjestelmien avulla. (2, s. 17.)

Sulautettujen järjestelmien ohjelmistoihin kuuluvat jo pelkästään lähes kaikki ihmisten arkikäytössä käyttämät elektroniset laitteet, kuten matkapuhelimet, pyykinpesukoneet ja autot. Sulautetut järjestelmät ovat ohjelmistotyypeistä näkyvin osa ihmisten arjessa. (2, s. 18.)

## **2.3 Määrittely ja vaatimukset**

Kaikille edellä mainituille ohjelmistotyypeille asetetaan monenlaisia vaatimuksia, jotka riippuvat ohjelmiston käyttötarkoituksesta ja vaatimustasosta. Ohjelmistot pyritään myös määrittelemään ainakin koon ja käsiteltävän tiedon määrän perusteella. (2, s. 18.)

### **2.3.1 Ohjelmiston määrittely**

Kun lähdetään määrittelemään ohjelmistoja, selvitetään ensin ohjelmiston tyyppi sekä se, miten esimerkiksi ohjelmiston koko voidaan kyseisessä järjestelmässä määrittää. Koko voidaan määrittellä mahdollisesti koodirivien määrällä, tietokantojen laajuudella tai vaikka näyttöjen lukumäärän perusteella. (2, s. 18.)

Ohjelmistojen koot vaihtelevat hyvin paljon sovellusten välillä, aina muutamista sadoista koodiriveistä miljooniin riveihin. Yleisin tapa ohjelmiston koon määrittämiseksi on lähdeohjelman rivimäärien laskeminen. Muita tapoja ovat esimerkiksi ohjelmiston loogisten lauseiden laskeminen dokumenttisivujen määrään perustuva koon arviointi tai toimintopistelaskenta. Esimerkiksi kaupallishallinnollisissa ohjelmistoissa laajuutta voi paremmin kuvata esimerkiksi näyttöjen määrä tai tietokantojen koko tai monimutkaisuus. (2, s. 18; 3, s. 14–15.)

### 2.3.2 Ohjelmistoille asetetut vaatimukset

Ohjelmistojen vaatimukset poikkeavat paljolti toisistaan sen mukaan, mihin ohjelmistotyyppiin ohjelma kuuluu. Esimerkiksi prosessinohjaus- ja sulautettujen järjestelmien ohjelmistoille vaatimuksia ovat reaaliaika-, vasteaika- ja luotettavuusvaatimukset. Vaatimukset ovat näille ohjelmistoille erityisen tärkeitä, jotta niitä pystytään käyttämään esimerkiksi vaativissa tehdasprosesseissa. (2, s. 18–19.)

Nykypäivän ohjelmistot ovat jakautuneet useita eri tyyppisiä koskeviksi järjestelmiksi. Kuvassa 1 on esitetty eri koko- ja vaatimusmäärittelyjä kuvitteellisista järjestelmistä. (2, s. 19.)

ominaisuus sovellus	koko	tietokan- nan koko	vasteaika, reaaliaikai- suus	hajautus	luotetta- vuus	tuotteistus- aste
GSM- puhelin	ROM- muistin koko 1MB	-	kovia reaa- liaikavaati- muksia	2 väylällä yhdistettyä prosessoria	-	massatuote
matkapu- helinke- kus	15M koo- diriviä	useita tie- tokantoja, kymme- niä mega- tavuja	paikoitel- len kovia reaaliaika- vaatimuk- sia	hajautettu monipro- sessorijär- jestelmä	kahdennet- tuja kom- ponentteja	tuotteis- tettu, asiakaskoh- taisia piir- teitä
televisio- vastaanotin	ROM- muistin koko 1MB	-	kovia reaa- liaikavaati- muksia	2 proses- soria, useita sig- naalipro- sessoreita	-	massatuote
kotitalou- den sähkö- mittari	ROM- muistin koko 64kB	-	kovia reaa- liaikavaati- muksia (virtakat- kos)	sähköver- kon kautta komentoja valvomosta	laskutustie- doissa ei saa olla vir- heitä	massatuote
suurehko hissijärjes- telmä	50 000 koodiriviä	-	kovia reaa- liaikavaati- muksia	hajautettu, CAN-väylä	virhetoimin- not estetty erillisellä elektro- niikalla	tuotteis- tettu, konfi- guroitava ko. raken- nukseen
vakuutus- ten hoito- järjestelmä	300 000 koodiriviä, 100 näyt- töä	100 relaa- tietieto- kannan taulua	vasteaika alle sekun- nin	keskitetty järjes- telmä, satojen päätteiden verkko	käyttökä- tkokset hait- taavat asiakaspal- velua	räätälöity

KUVA 1. Esimerkki vaatimuksista ja luokittelusta (2, s. 20)

## 2.4 Ohjelmistotuotannon ongelmat

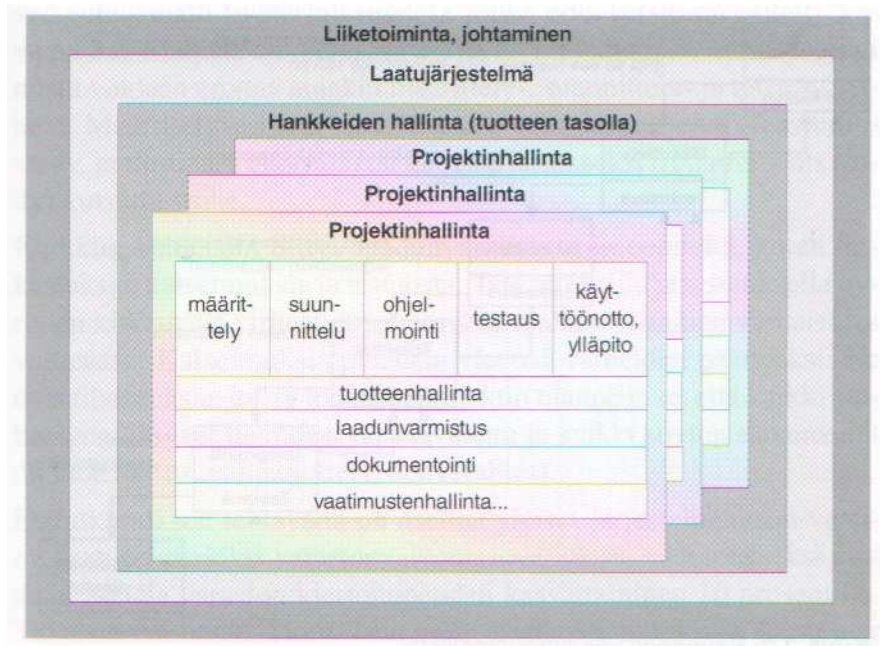
Ohjelmistotuotannon ongelmat johtuvat suurelta osin alan nuoruudesta sekä ohjelmistojen räjähdysmäisestä kasvusta. Ohjelmistojen jatkuva nopea kasvu sekä monimutkaistuminen aiheuttavat kehitystyössä suuria ongelmia, koska ohjelmistojen tekeminen vielä tänäkin päivänä perustuu liikaa improvisointiin ja teellisen pohjan puuttumiseen kehitystyöstä. (2, s. 24; 4; 5.)

Lisää ongelmia sekä työtä aiheuttaa jo aikaisemmin tehdyn työn alhainen uudelleen käyttöprosentti 10–20 %, joka johtuu suurilta osin standardoinnin puutteesta sekä jakelukanavien kehittymättömyydestä. WiseMaster-tuotteen kehityksessä käytetään huomattavasti suurempaa osuutta vanhaa koodia kuin lähteestä ilmenee. Myös ohjelmistojen määrittelystä löytyy puutteita ja on hyvin yleistä, että asiakas ei tiedä, mitä haluaa. Josta hyvä esimerkki löytyy M-Technologyssä tehdystä ohjelmisto projektista. (2, s. 25; 4; 5.)

Ohjelmistoprojektien kannattavuus on yleisellä tasolla melko heikkoa, joka johtuu projektien aikataulutuksen vaikeudesta ja näin ollen projektien venymisistä. Toinen suuri kustannus projekteissa koostuu ohjelmiston ylläpidosta. Ylläpito-kustannukset nousevat ohjelmiston elinkaaren aikana moninkertaisiksi tai jopa monikymmenkertaisiksi ohjelmiston kehityskustannuksiin verrattuna ja voivat näin aiheuttaa yllätyksiä kokemattomille toimijoille. (2, s. 25; 4; 5.)

### 3 OHJELMISTOTUOTANTO

Ohjelmistoprojekteille on olemassa monenlaisia esitystapoja. Kuvassa 2 on yksi esimerkki eri vaiheista. Vaiheet projektinhallinnan alapuolella ovat yrityksen määrittämiä liiketoiminnallisia sekä laadunhallintaan ja hankkeidenhallintaan kuuluvia osioita. Projektinhallinnan vaiheisiin yleisesti kuuluvat ainakin seuraavat vaiheet: määrittely, suunnittelu, ohjelmointi, testaus, käyttöönotto ja ylläpito. Edellä mainittujen vaiheiden alapuolella olevat tukitoiminnot kestävät koko ohjelmiston elinkaaren ajan. Yleisimpiin ja tärkeimpiin tukitoimintoihin kuuluvat laadunvarmistus, tuotteenhallinta ja dokumentointi. Tukitoimintojen määrä ja laajuus vaihtelee projektien ja yritysten tapojen mukaan. Pienemmät yritykset voivat yhtenäistää eri tukitoimia, kun taas suuremmat käsittelevät tukitoimet laajemmin omina kokonaisuuksina. (2, s. 36.)



KUVA 2. Esimerkki ohjelmistotuotannon eri osa-alueista (2, s. 35)

## **3.1 Ohjelmistotuotannon vaiheet**

Ohjelmistojen suunnittelu jakaantuu eri vaiheisiin. Vaiheiden määrä vaihtelee ohjelmistoprojektin laajuuden sekä yrityksen sisällä olevien tapojen mukaan. Seuraavissa luvuissa läpi käydään ohjelmistotuotannon päävaiheet, jotka kuuluvat tai joiden tulisi kuulua jokaisen projektin suunnittelurunkoon.

### **3.1.1 Esitutkimus**

Esitutkimusvaiheessa selvitetään järjestelmän yleisiä vaatimuksia, jotka saadaan suoraan asiakasvaatimuksista. Esitutkimus on tärkeä osa projektin onnistumisen kannalta. Jos asiakasvaatimuksia ei oteta huomioon heti kehityksen alussa, ei ohjelmasta voida saada hyvää. (6.)

Esitutkimuksen perusteella tehdään päätös onko järjestelmää mahdollista tehdä sekä onko järjestelmän toteuttaminen kannattavaa. Käytännön asiakasprojekteissa esitutkimus suoritetaan tehtäessä projektin tarjouslaskentaa. (6.)

### **3.1.2 Määrittely**

Kun esitutkimuksen perusteella projektille on saatu vihreää valoa, suoritetaan tarkempi vaatimustenmäärittelyvaihe. Määrittelyvaiheessa asiakasvaatimuksia tarkennetaan ja kootaan yhteen yhdessä asiakkaan kanssa. Jos tässä luistetaan ja määrittelyssä ei huomioida asiakkaan tarpeita eikä kommunikoida asiakkaan kanssa, on hyvin todennäköistä että valmis tuote ei vastaa tilattua ja aiheuttaa näin huomattavia lisäkustannuksia. (6.)

Kun asiakkaan kanssa yhteistyössä tehty vaatimusmäärittely on valmis, johdetaan siitä ohjelmistovaatimukset. Ohjelmistovaatimuksista ilmenee, millainen ohjelmisto täyttää asiakkaan vaatimukset. (6.)

### **3.1.3 Suunnittelu**

Suunnitteluvaiheessa muunnetaan määrittelyvaiheiden suunnitelmat ja toiminnot tekniselle kielelle. Suunnitteluvaiheessa suunnitellaan koko ohjelmistototeutus. Eri vaiheita ovat ohjelmiston-, käyttöliittymän-, tietokantojen, testauksen- ja käyttöönoton suunnittelu. (6.)

Suunnittelu koostuu kahdesta päävaiheesta, arkkitehtisuunnittelusta ja moduulisuunnittelusta. Arkkitehtisuunnittelussa järjestelmä jaetaan mahdollisimman itsenäisiin moduuleihin ja määritellään moduulien rajapinnat. Arkkitehtuurisuunnittelusta koostuvaa dokumenttia kutsutaan tekniseksi määrittelyksi. Moduulisuunnittelussa paneudutaan arkkitehtuurisuunnittelussa jaettuihin moduulien rakenteisiin. (2, s. 40; 6.)

Suunnittelun tuloksena ohjelmoitsijan pitäisi pystyä toteuttamaan määritysten mukainen järjestelmä suunnitteluvaiheen dokumenttien mukaan.

### **3.1.4 Toteutus**

Toteutusvaiheessa kirjoitetaan itse koodi. Ohjelmoitsija toteuttaa ohjelmiston aikaisemmissa vaiheissa luotujen dokumenttien perusteella ja huolehtii yksikkötestauksesta. Ohjelmoitsija myös suunnittelee moduulien sisäiset liitännät, tietorakenteet ja algoritmit. (6.)

### **3.1.5 Testaus**

Testauksen määritelmä on virheiden järjestelmällinen etsiminen ohjelmaa suorittamalla. Testausta suoritetaan monella tasolla, alkaen moduulitestauksesta ja päättyen ohjelman laajamittaiseen testiin esimerkiksi päätelaitteessa. (6.)



Hyvin suunniteltu ja toteutettu testaus säästää suurilta kustannuksilta ohjelmiston käyttöönoton jälkeen. Virheiden korjaus vielä testausvaiheessa on jopa kymmeniä kertoja halvempaa kuin ohjelmiston julkaisemisen jälkeen. Hyvin testattu ohjelmisto lisää myös asiakastytyvyyttä. (2, s. 40; 6.)

Testauksen suunnittelu aloitetaan testitapauksien suunnittelulla. Kaksi yleisintä lähestymistapaa ovat lasilaatikko- sekä mustalaatikkotestaus. Lasilaatikkotestauksessa testaus tapahtuu käyttämällä ohjelman lähdekoodia hyväksi ja mustalaatikkotestauksessa testitapaukset valitaan tuntematta ohjelman rakennetta ja sisäistä toimintaa. Testitapauksien määrittämisen jälkeen ohjelmaa testataan niin kauan, että testitapaukset on käyty riittävällä tarkkuudella virheettömästi läpi. (6.)

### **3.1.6 Ylläpito**

Ylläpitoon kuuluvat kaikki ohjelmaan käyttöönoton jälkeen tehtävät muutos- tai korjaustyöt. Suurimmat ylläpitotoimet aiheutuvat asiakkaalla ilmenneistä ongelmista ja testistä läpi päässeiden bugien korjauksesta. Asiakkaalla voi myös taustajärjestelmä muuttua tai koko taustaohjelmisto ja laitteisto uusiutua, jolloin joudutaan tekemään muutoksia myös toimitettuun järjestelmään. Myös asiakkaan muutospyynnöt kuuluvat ylläpidollisiin tehtäviin. (6.)

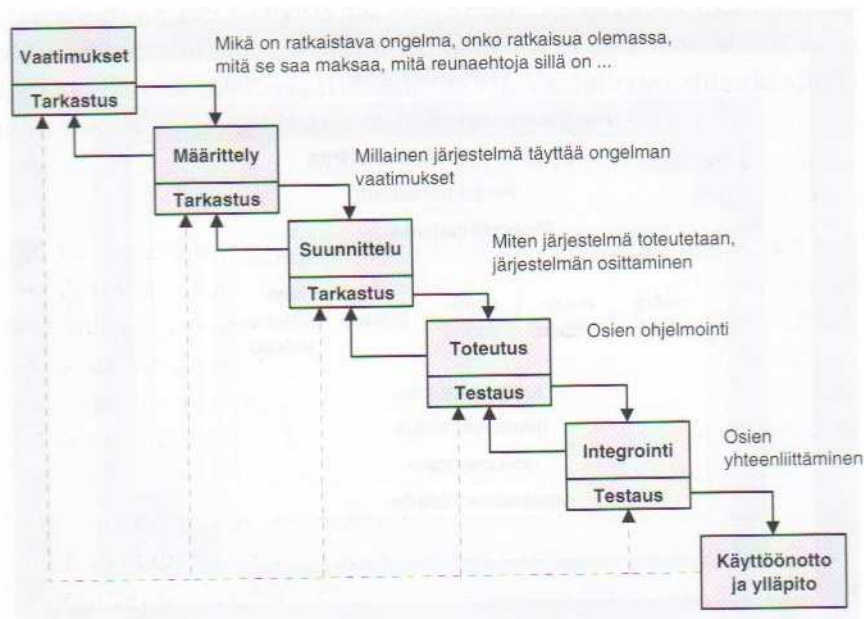
Asiakasprojektien ylläpitokustannukset ovat vuosittain noin 5–30 % kehityskustannuksista. Koko ohjelmiston elinkaaren aikana kertyvät ylläpitokustannukset voivat nousta jopa 80 %. (6.)

## 3.2 Elinkaarimallit

Elinkaarimalleilla kuvataan ohjelmiston elinkaaren eri vaiheita, ohjelmiston kehittämisen aloittamisesta sen poistamiseen käytöstä. Elinkaarimallit mahdollistavat ongelmien systemaattisen tarkastelun. Elinkaarimalleja on olemassa lukuisia ja erilaiset mallit ovat tarpeen, koska työ- ja kehitysprosessit poikkeavat toisistaan. (6.)

### 3.2.1 Vesiputousmalli

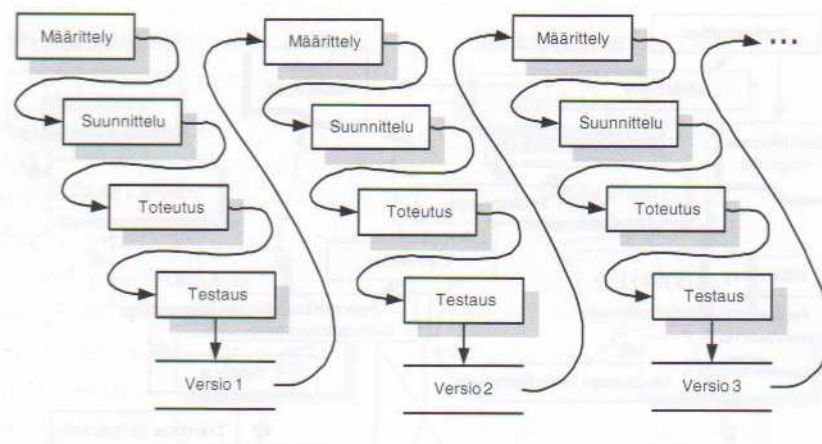
Tyypillisin vaihejakomalli on vesiputousmalli. Yleensä se sisältää ainakin määrittely-, suunnittelu- ja toteutusvaiheet. Vesiputousmallin vaiheisiin kuuluu myös laadunvarmistustoimenpiteitä, tarkastuksia, katselmuksia ja testausta. Kuva 3 on esimerkki yleisestä vesiputousmallista. (6.)



KUVA 3. Esimerkki vesiputousmallista (1, s. 36)

### 3.2.2 Evo-malli

Evo-mallissa rakennetaan ensimmäisessä projektissa ydinjärjestelmä, jota kehitetään seuraavissa projekteissa eteenpäin. Evo-malli rakentuu peräkkäisistä vesiputousmalleista ja jokaisen mallin tuloksena syntyy uusi versio järjestelmästä. Uutta versiota lähdetään kehittämään edellisen version palautteen perusteella tai jopa aikaisempien versioiden palautteen perusteella. Esimerkki evo-mallista on kuvassa 4. (2, s. 45; 6.)



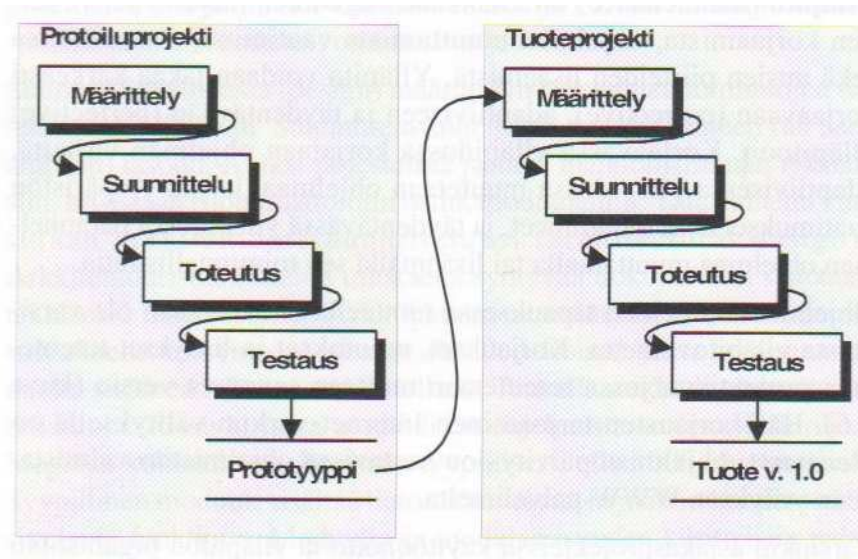
KUVA 4. Esimerkki Evo-mallista (1, s. 44)

Evo-mallin tapaisia muunnosmalleja sanotaan inkrementaaliksi malleiksi. Tällä tarkoitetaan mallia, jossa lopputuotetta kehitetään pienehköinä osina projektin sisällä. Inkrementtikierroksien lopputuloksena syntyy toimiva järjestelmä. (2, s. 45; 6.)

### 3.2.3 Protoilumalli

Protoilumallilla voidaan tarkoittaa kaikkia malleja, joissa jotain tuotteen piirrettä kokeillaan ennen varsinaisen tuotteen rakentamista. Prototyypit soveltuvatkin hyvin uusien ratkaisujen kokeiluun ja epäselvien asiakasvaatimusten määrittämiseen. (2, s. 42; 6.)

Prototyypin valmistumisen jälkeen projektin jatkamiselle on kaksi päävaihtoehtoa. Prototyypin perusteella määritetään tehtävä järjestelmä ja toteutetaan se alusta alkaen uudestaan tai prototyypin kehittämistä jatketaan valmiiksi järjestelmäksi asti. Välimuodotkin ovat mahdollisia. Kuvassa 5 on yleinen protoilumalli. (2, s. 42; 6.)



KUVA 5. Esimerkki protoilumallista (2, s. 42)

### 3.3 Ketterä ohjelmistokehitys

Ketterät ohjelmistokehitystyökalut ovat joukko ohjelmistotuotantoprojekteissa käytettäviä menetelmiä, joille yhteistä on toimivan ohjelmiston ensisijaisuus, suora viestintä ja nopea reagointi muutoksiin. Ketterissä menetelmissä pyritään myös järkevään dokumentaation, suunnittelun ja ohjelmoinnin suhteeseen. Pääpaino asetetaan asiakkaan tyytyväisyydelle ja toimivalle ohjelmistolle. (7; 8.)

#### 3.3.1 Extreme Programming, XP

XP on tunnetuin ketteristä menetelmistä. XP:n tunnuspiirteitä ovat jatkuva testaaminen ja pariohjelmointi, jossa ohjelmoitsijat parina työskentelevät erittäin tiiviisti kommentoiden koko ajan tuotosta. (2, s. 47; 8.)

Tärkeimpiä asioita XP:ssä ovat kommunikointi ihmisten välillä sekä rakenteiden yksinkertaisuus, palautteen kerääminen ja rohkeus. Varsinkin kohdatessa virheitä tulee uskaltaa tarvittaessa jopa vaatia koodin kokonaan uudelleenkirjoittamista. (8.)

### **3.3.2 Scrum**

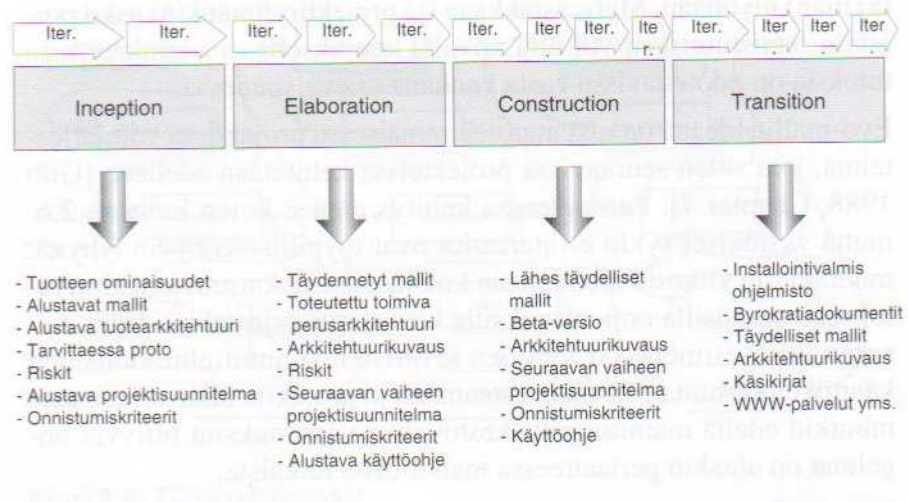
Scrumissa tavoitteena olevaa tuotetta rakennetaan pikkuhiljaa täydellisemmäksi ja valmiimmaksi useiden toteutuskierrosten aikana. Toteutuskierroksia kutsutaan sprinteiksi. Sprintti kestää n. 2–4 viikkoa tyypillisesti noin 30 päivää. Sprinttien lopuksi järjestettävässä demotilaisuudessa jokainen projektitiimi esittelee, mitä sprintin aikana on saavutettu. (9.)

Scrum-menetelmässä vastuu toteutuksesta on tiimeillä. Tiimit päättävät sisäisesti kunkin sprintin tavoitteet ja tehtävät ja vastaavat yhdessä siitä, että tavoitteet saavutetaan. Scrumin ohjelmistokehitysprosessi koostuu seuraavista vaiheista: Suunnitteluvaiheessa määritetään vaatimukset ja vaatimusten perusteella arvioidaan aikataulutus ja kustannukset. Arkkitehtuurisuunnitteluvaiheessa luodaan vaatimuksia vastaava järjestelmän arkkitehtuuri. Peräkkäisissä toteutusvaiheissa toteutetaan valitut vaatimukset. Projektin päätöksessä käydään toteutukset läpi ja luodaan tarvittava dokumentaatio. (9.)

### **3.3.3 RUP (Rational Unified Process)**

Evo- ja spiraalimallien sovelluksena voidaan pitää RUP-nimistä prosessia. RUP ei varsinaisesti ole ketterä menetelmä, mutta soveltuvin osin käytettynä siitä voidaan ketteriäkin piirteitä löytää. (2, s. 45; 8.)

RUP perustuu vesiputousmallin kaltaisiin perättäisiin iteraatioihin. Ohjelmistonkehitys jakaantuu RUPissa neljään päävaiheeseen: inception, elaboration, construction ja transition. Vaiheiden tärkeimmät tulokset on esitetty kuvassa 6. (2, s. 45.)



KUVA 6. RUPin vaiheet jaoteltuina (2, s. 46)

## 4 JÄRJESTELMIEN MALLINTAMINEN UML-KIELELLÄ

Olioperustaisuus mahdollistaa systemaattisen ohjelmistokehityksen lähtien vaatimusmäärittelystä ja päättyen suoritettavaan koodiin. Olioperustaisuus ei ole kuitenkaan sovelluskehitysautomaatti eikä tee huonosta suunnittelijasta hyvää. Yhteiset yleisperiaatteet suunnittelijan ja ohjelmoitsijan tukena helpottavat kyllä käyttökelpoisten ratkaisujen löytämistä. Täysin automaattista sovelluskehitystä ei ole vielä keksitty ja ei tulla luultavasti keksimäänkään. (10, s. 123.)

### 4.1 UML

UML (Unified Modeling Language) syntyi, kun kolme keskeistä oliosuunnittelu menetelmien kehittäjää (Booch, Rumbaugh ja Jacobson) alkoivat 90-luvulla yhdistää omia kaavioesityksiään universaaliksi notaatioksi. Myöhemmin vastuun kehittämisestä on ottanut OMG (Object Management Group)-yrityskonsortio, jonka tavoitteina on olioteknologioiden edistäminen ja standardointi. Tänä päivänä lähes kaikki CASE-työkalut ja menetelmät perustuvat UML:ään. (10, s. 124.)

UML koostuu erilaisista kaavioista, erityyppisiä kaavioesitystapoja on yhdeksän. Kaavioita käytetään kuvaamaan järjestelmän toiminnallisuutta, järjestelmän staattista ja dynaamista rakennetta sekä järjestelmän dynaamista käyttäytymistä. Osa kaavioista on toistensa kanssa osittain päällekkäisiä. (10, s. 125.)

Kaaviotyyppejä voidaan menetelmistä riippuen soveltaa eri ohjelmistokehityksen vaiheissa. Kuvassa 7 on esitetty kaaviotyypit siten että ohjelmistokehityksen alkuvaiheen kaaviot ovat ylhäällä ja loppuvaiheessa käytettävät kaaviot alhaalla. (10, s. 124.)

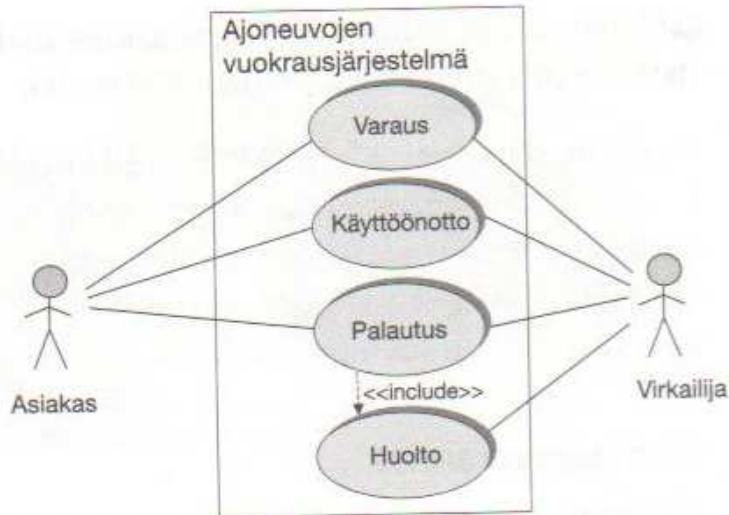


KUVA 7. UML:n kaavio tyypit (10, s. 126)

## 4.2 Käyttötapauskaavio

Käyttötapauskaaviolla kuvataan järjestelmän käyttötapausten väliset suhteet ja käyttötapauksiin osallistuvat järjestelmän ulkoiset toimijat. Käyttötapauskaavion perusteella saadaan jo suunnitteluvaiheessa hyvin hahmoteltua ohjelmiston eri toimijat ja toiminnot. Kuvassa 8 on esimerkki käyttötapauskaaviosta. (10, s. 127.)

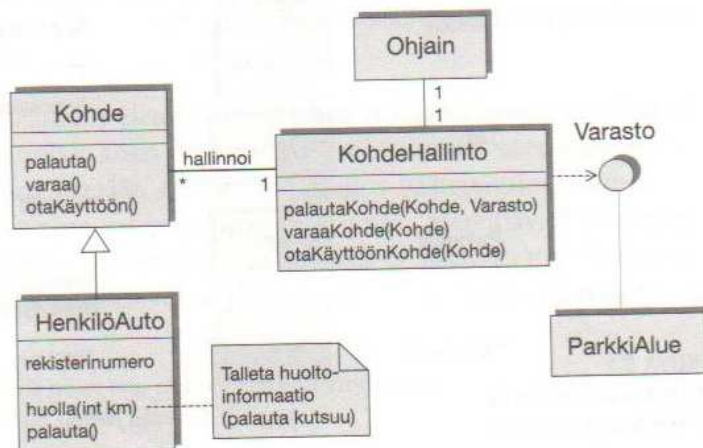




KUVA 8. Esimerkki käyttötapauskaaviosta (10, s. 128)

### 4.3 Luokkakaavio

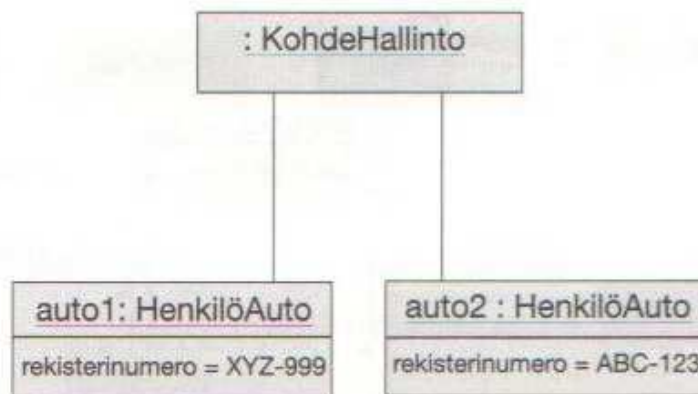
Luokkakaavion avulla kuvataan olio-ohjelmointikielen luokkia ja niiden välisiä suhteita. Luokkakaaviolla kuvataan järjestelmän staattista rakennetta. Jotkin ohjelmistoympäristöt voivat jopa suoraan osata tehdä luokkakaaviosta ohjelma-koodia. Kuvassa 9 on esimerkki luokkakaaviosta. (10, s. 128.)



KUVA 9. Esimerkki luokkakaaviosta (10, s. 128)

## 4.4 Oliokaavio

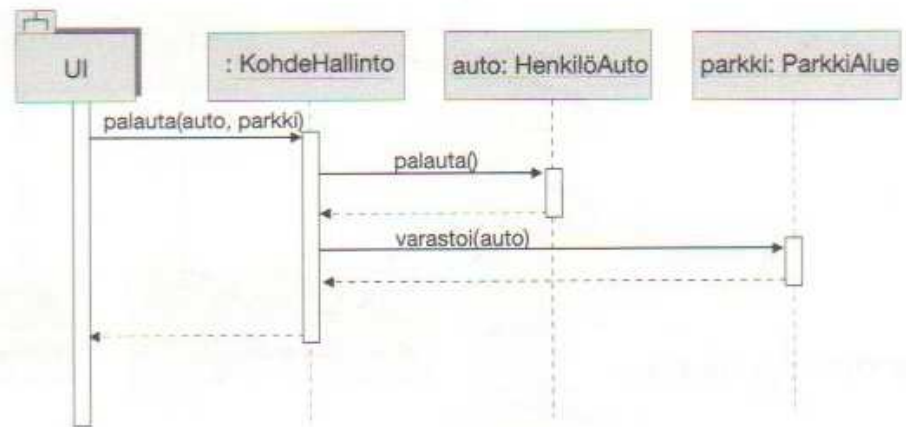
Oliokaavioilla kuvataan olioiden välisiä suhteita. Oliokaavio esittää luokkakaavio-  
on ajonaikaista ilmenemismuotoa ja oliot kuvataan suorakaiteina luokkien ta-  
voin. Oliolle voidaan kirjoittaa näkyviin arvoja, ja olioiden yhteyksiä kuvataan  
suorakaiteiden välisillä viivoilla. Kuvassa 10 on esimerkki oliokaaviosta (10, s.  
129.)



KUVA 10. Esimerkki oliokaaviosta (10, s. 129)

## 4.5 Sekvenssikaavio

Sekvenssikaaviolla kuvataan oliojoukkojen välistä vuorovaikutusta. Kuvaus  
koostuu olioista ja niiden välisistä sanomista. Sanoma voi olla esimerkiksi ope-  
raatiokutsu, jolloin kutsuja on nuolen alkupäässä ja kutsuttava nuolen loppu-  
päässä. Esimerkki sekvenssikaaviosta on kuvassa 11. (10, s. 129–130.)



KUVA 11. Esimerkki sekvenssikaaviosta (10, s. 130)

## 5 M-TECHNOLOGY OY:N WISEMASTER-TUOTTEET

M-Technology Oy on vuonna 2001 perustettu oululainen mobiiliteknologiaan erikoistunut ohjelmistoyritys. M-Technology Oy on erityisesti keskittynyt liikkuvan työn mobiilitekniikoihin ja erityisesti sen hyödyntämiseen käytännön yritystoiminnassa. Yrityksen ensimmäiset ohjelmistotuotteet TimeMaster sekä DriveMaster tuotiin markkinoille vuonna 2001. Ohjelmistot olivat ensimmäiset Windows Mobile -pöytälaitteille tehdyt kaupalliset suomalaiset ohjelmistoratkaisut. (11.)

Vuonna 2003 julkaistiin WiseMaster-tuoteperhe, joka oli tarkoitettu vastaamaan yritysten kasvaviin vaatimuksiin mobiilintoiminnan kentällä. WiseMaster-ohjelmiston keskeinen tavoite on tuoda asiakkaille säästöä mobiilintoiminnanohjauksen avulla liikkuvan työntekijöille. WiseMaster-ohjelmisto on modulaarinen ohjelmistoratkaisu, jonka eri osa-alueet soveltuvat niin huolto- ja ylläpitotoimintaan, kenttämyyntiin, tuotteiden jakeluun kuin julkisten palvelujen vaatimuksiin. Ohjelmiston perusominaisuuksiin kuuluu sähköinen työnohjaus, laskutustiedon keruu mobiilisti, asiakas- ja kohdehistoriat sekä kalustonhallinta ja kunnossapitoratkaisut. (11.)

WiseMaster-mobiiliohjelmisto on toteutettu client-server-tekniikalla ja näin ollen WiseMaster on myös liitettävissä asiakkaiden omiin ja kolmannen osapuolen taustajärjestelmiin. WiseMasterin käyttäjiä on satoja ympäri Suomen, aina yhden miehen yrityksistä suuriin konserneihin. Voidaan sanoa, että WiseMaster on vielä tänä päivänäkin ainutlaatuinen ohjelmistoratkaisu Suomessa. Joitain web-pohjaisesti toteutettuja ohjelmistoratkaisuja Suomestakin toki löytyy mutta WiseMasterin kaltaista kokonaisuutta ei. (11.)

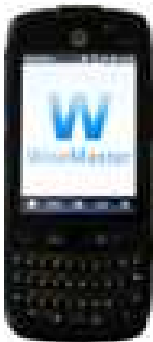
## 5.1 WiseMaster

WiseMaster-tuoteperheeseen kuuluu neljä eri ohjelmistoa, PC-tietokoneelle tarkoitettu WiseMasterPC, PC-tietokoneelle tarkoitettu suppeampi ohjelmisto WiseMasterPCClient, PDA-laitteelle tarkoitettu WiseMasterMobiili ja selainpohjainen WebWise. (12.)

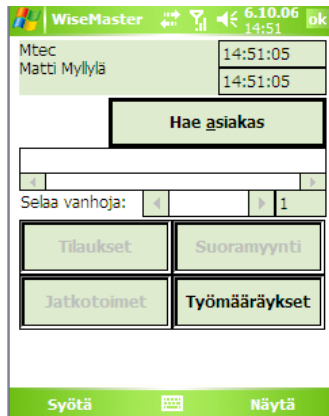
Näistä neljästä ohjelmistosta lähdetään rakentamaan asiakkaan vaatimuksia vastaava järjestelmä. (12.)

## 5.2 WiseMasterMobiili

WiseMasterMobiili on WiseMaster-tuoteperheen ”tärkein” ohjelmisto, joka tekee WiseMasterista ainutlaatuisen muihin laskutus- ja tiedonkeruuratkaisuihin verrattuna. Mobiilijärjestelmä jakautuu kolmeen eri pääluokkaan: huolto- ja ylläpito-toiminta, kenttämyynti ja tuotteiden jakelu sekä julkiset palvelut. WiseMasterMobiili-ohjelmisto on laajasti muokattavissa kirjavan asiakaskunnan käyttöön. Lisäominaisuutena WiseMasterMobiiliin on liitettävissä esimerkiksi GPS-paikannusominaisuus sekä erilaisten ulkoisten Bluetooth-laitteiden (esim. tulostimien) hyväksi käyttäminen. Kuvissa 14 ja 15 on esimerkit mobiilipäätelaitteesta sekä WiseMasterMobiilin näkymästä. (12.)



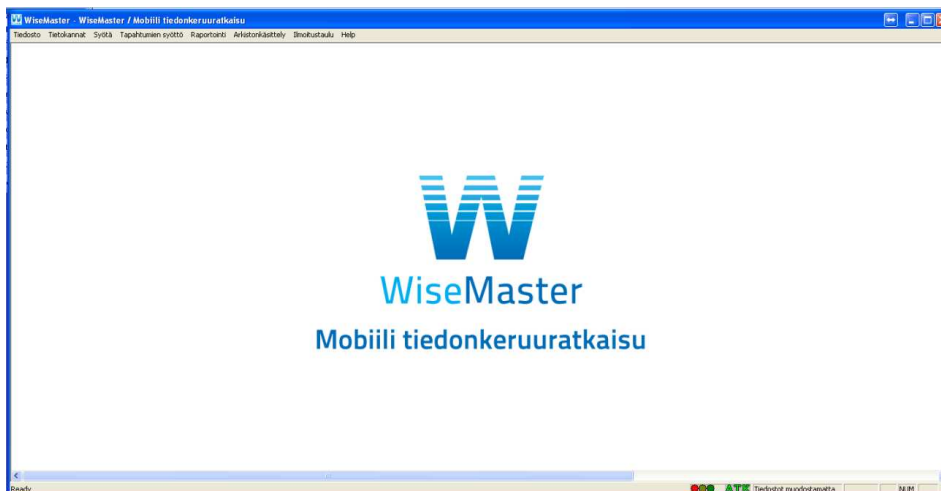
*KUVA 14. Pätelaite (MotorolaES400) (12)*



KUVA 15. WiseMasterMobiili (12)

### 5.3 WiseMasterPC

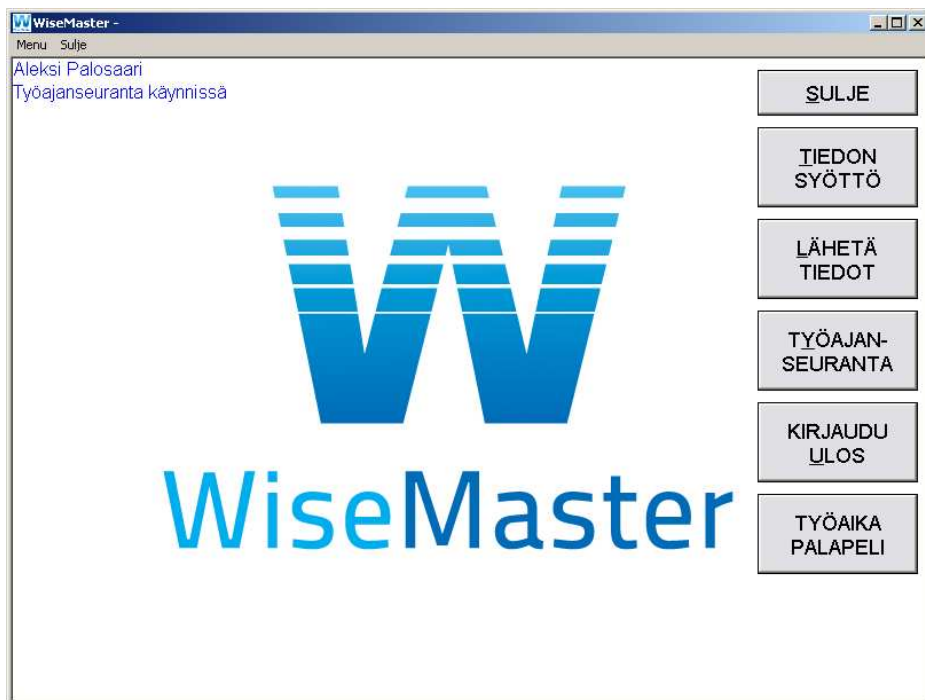
WiseMasterPC-ohjelmisto on toimistolle tarkoitettu tiedonylläpito- sekä käsittelyohjelma. Toimisto-ohjelmalla hoidetaan niin asiakas-, tuote- kuin työntekijäkantojen ylläpito sekä huoltopuolella työmääräyksien luonti, työnohjaus ja mobiililaitteilta tulleen kertymän laskutus. Tilauspuolella tärkeimmät toiminnot ovat myyjien mobiili-sovelluksesta tulleiden tilausten laskutus sekä laajamittainen myynninraportointi. Kuvassa 12 on WiseMasterPC-ohjelman päänäkymä. (12.)



KUVA 12. WiseMasterPC:n päänäkymä (12)

## 5.4 WiseMasterPCClient

WiseMasterPCClient on WiseMasterMobiilia vastaava tietokonesovellus. WiseMasterPCClientistä löytyy samat ominaisuudet kuin huollon mobiili-sovelluksesta sekä työnjohtajan tunnuksilla myös WiseMasterPCstä löytyvä työmääräyksien hallinta näkymä. Ohjelmisto on vaihtoehto PDA-laitteelle, työn tapahtuessa esimerkiksi aina samassa toimipisteessä tai työ vaatiessa kannettavan tietokoneen kuljettamisen mukana. PCClient-sovellus toimii myös TimeMaster-työajanseurannan hallintatyökaluna. Kuvassa 13 WiseMasterPCClientin päänäkymä. (12.)



KUVA 13. WiseMasterPCClientin päänäkymä (12)

## 5.5 WebWise

WebWise-ohjelmisto on WiseMaster-perheen uusin tulokas. WebWise kehitettiin Hiven Oy:n tarpeisiin. Ohjelmistoa käyttävät ympäri Suomea olevien myyjien lisäksi toimiston työntekijät.

Myyjät määrittävät WebWisessä katteet omille tuotteille, seuraavat tilausten etenemistä reaaliaikaisesti sekä näkevät kaikki historiatapahtumat. Toimistolla pystytään seuraamaan edustajien myyntiä reaaliaikaisesti sekä tekemään tuotannosuunnittelua. Kuvassa 16 Hivenen WebWisen päänäkymä.



KUVA 16. WebWise päänäkymä(Hiven Oy)

WebWise-ohjelmisto tullaan muuntamaan muidenkin asiakkaiden tarpeita vastaavaksi sekä ottamaan käyttöön halukkaille asiakkaille. WebWise ohjelmisto on ensimmäinen selainpohjainen WiseMaster-sovellus ja tulevaisuudessa toimintojen hyödyntäminen selaimella tulee lisääntymään.

## 5.6 WiseMaster Huolto

WiseMaster-ohjelmiston huoltapuolen-sovellus tarjoaa työkalut sähköiseen työnohjaukseen mobiilipäätelaitteita hyväksikäyttäen sekä laskutustietojen keräämisen suoraan kentältä. (12.)

Huoltapuolella työnohjaajan päätehtävät WiseMasterPC:llä ovat työmääräysten luominen ja muokkaaminen, töiden etenemisen seuranta ja reaaliaikainen tilaseuranta sekä töiden aikataulutuksen ja aikataulutuksen toteutuksen seuraaminen. (12.)



Kentällä työntekijät näkevät työnohjaajien osoittamat työmääräykset PDA-laitteilla tai kannettavilla tietokoneilla WiseMasterPCClient-ohjelmistolla. Työntekijöiden tehtäviin kuuluu kuitata työmääräykset tilojen mukaan aloitetuksi, keskeytetyksi ja valmiiksi sekä laskutustietojen kerääminen ja työselostuksen täyttäminen. (12.)

### 5.6.1 Työmääräyksen luonti

Työmääräyksen luonti tapahtuu WiseMasterPC-ohjelmistolla tai WiseMasterPCClientin työnohjaajan versiolla. Mikäli asiakkaalla on käytössä erillinen järjestelmä jossa työmääräysten luominen tapahtuu, voidaan työmääräykset tuoda rajapintojen avulla suoraan WiseMasteriin. Kuvassa 17 on esimerkki työmääräyksen lisäämisestä. (12.)

Vaahutu	Tuote	Käytetty	Rivinä	T
1	MattiTuote			1

KUVA 17. Työmääräyksen lisäysnäky (12)

## 5.6.2 Työmääräysten seuranta

Työmääräysten tilaseuranta tapahtuu WiseMasterPC-ohjelmiston työmääräysnäköymästä. Tilan seuranta tapahtuu reaaliaikaisesti väri koodein. Kuvassa 18 on esimerkki työmääräysnäköymästä. (12.)

Tekijä:	Asiakas:	Selite:	Kommentti:
Aloittamatta: 1 Matti Myllylä Alku aika: 26.09.07 08:00 Loppu aika: 26.09.07 12:00 Kesto: 04:00 Sisältää tuotteita		Keittion IV-suodattimen vaihto: prior: 3 Vaihdetaan suodatin ipx-23	
Aloittamatta: Kohdentamaton Alku aika: 26.09.07 08:00 Loppu aika: 26.09.07 10:00 Kesto: 02:00		Viemärin tukkeuman poisto: prior: 3 Keittion viemärisä tukkeuma, poistettava. Asiakas toivoi ajan.	
Aloittamatta: 2 Niko Honkonen Alku aika: 26.09.07 09:00 Loppu aika: 26.09.07 10:30 Kesto: 01:30		Tarkasta valaistus: prior: 3 Muutama lamppu palanut, vaihda. 3. krs	
Aloitettu: 1 Matti Myllylä Alku aika: 25.09.07 08:00 Loppu aika: 25.09.07 11:00 Kesto: 03:00 Sisältää tuotteita		Oulun Komac Oy esittely: prior: 3 Järjestelmän esittely ja ympäristömittaritely.	
Lopetettu: 1 Matti Myllylä Alku aika: 26.09.07 13:00 Loppu aika: 26.09.07 16:00 Kesto: 03:00 Sisältää tuotteita		IV-koneen kuukausihuolto: prior: 3 Tarkista moottori ja suodattimet. Laakeri sääntää ilmoituksen mukaan.	

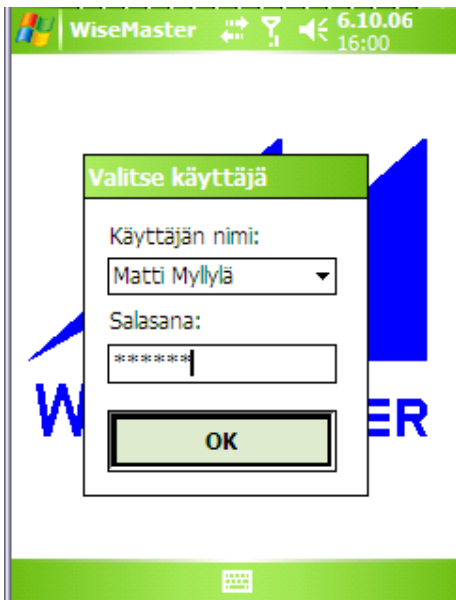
Right side controls:  
Suodata tilan perusteella: Aloittamattomat, Keskeneräiset, Käynnissä olevat, Valmit, Poistu  
Hae ajan perusteella: Hae vain alla olevan ajanjakson työmääräykset (23.9.2007 - 14.10.2007)  
Suodata työntekijän perusteella: Kaikki, Ei mitään  
List: 0 Kohdentamaton, 1 Matti Myllylä, 2 Niko Honkonen, 3 Peim Eriklund, 99 Tekijä avoin  
Päivitä Lista, Asetukset  
Hae asiakasta / osoitetta: Hae, Alä huomioi asiakasluussa muuta suodatinta.  
Yhteensä: Aloittamattomat: 4 kpl (11 h 30 min), Käynnissä olevat: 1 kpl (3 h 0 min), Kesken: 0 kpl (0 h 0 min), Valmit: 1 kpl (3 h 0 min)

KUVA 18. Työmääräysnäköymä (12)

## 5.6.3 Työnohjaus päätelaitteella

Työntekijä kirjautuu WiseMasterMobiili-sovellukseen päätelaitteella (kuva 19). Huoltoapuolella kirjaukset tapahtuvat työmääräysten kautta. Käyttäjä näkee oman listan sekä kaikille tarkoitetut avoimet työt työmääräysnäköymästä (kuva 20). (12.)

Työmääräyksiä kuitataan reaaliaikaisesti päälle, pois ja kirjataan jokaiselle työmääräykselle tapahtunut laskutettava kertymä. Työmääräykset sisältävät kaiken tarvittavan tiedon töiden toteuttamiseen, kuten osoitetiedot, työohjeen, yhteystiedot sekä työn tekemiseen tarkoitetun ajankohdan. (12.)



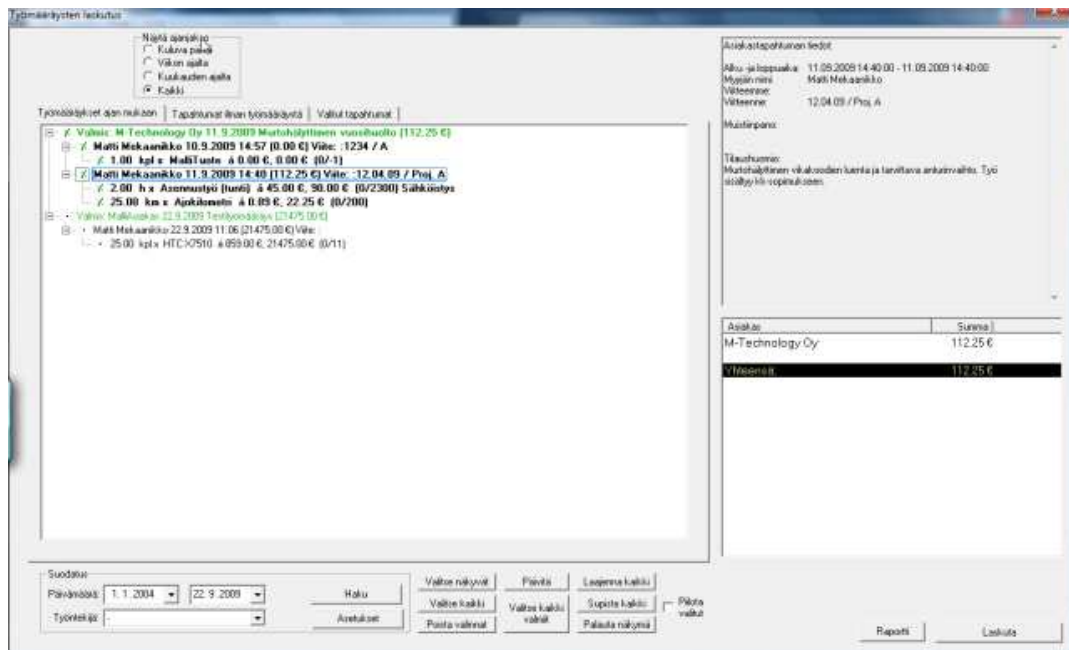
KUVA 19. WMMobiilin sisäänkirjautuminen (12)



KUVA 20. WMMobiilin työmääräysnäkyvä (12)

#### 5.6.4 Kertymän hyväksyminen ja laskuttaminen

Työntekijän PDA-laiteella keräämät ja toimistolle lähettämät tiedot ovat toimisto ohjelmistolla hyväksyttävissä ja siirrettävissä eteenpäin laskutukseen tai erilliseen taustajärjestelmään. Esimerkki laskutusnäkyvästä on kuvassa 21. (12.)



KUVA 21. WiseMasterPCn laskutusnäkymä (12)

## 5.7 WiseMaster Tilaus

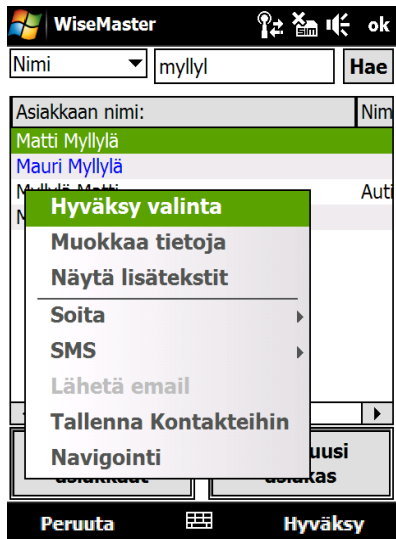
WiseMaster-ohjelmiston tilauspuolen sovellus tarjoaa työkalut asiakas- ja tuoterekisterien reaaliaikaiseen käyttöön, sekä tilauksien keräämisen suoraan kentältä mobilipäätelaitteita hyväksikäyttäen. (12.)

WiseMasterin tilauspuoli toimii pääasiassa liikkuvien myyntiedustajien mobiilina tilaustenkirjausjärjestelmänä. Myyjät saavat ajantasaiset tuote-, asiakas- ja historiatiedot suoraan PDA-laiteelle ja pystyvät kirjaamaan tilaukset suoraan asiakkaan luona. Tilaukset siirtyvät mobililaitteelta palvelimelle ja sieltä edelleen toimistolle jossa tilaukset hyväksytään ja laskutetaan. (12.)

Käyttäjä aloittaa käynnistämällä ohjelman ja kirjautumalla sisään omalla nimellä. Asiakastapahtuma aloitetaan hakemalla asiakas, asiakasrekisteristä (Kuva 23). Asiakas- ja tuoterekisterit päivittyvät jokaisen tiedon synkronoinnin yhteydessä ja rekisterit tallentuvat päätelaitteen muistiin jolloin ne ovat nopeasti ja aina yhteyksistä riippumatta käyttäjän käytettävissä. Kuvassa 22 on tilauspuolen perusnäkymä. (12.)

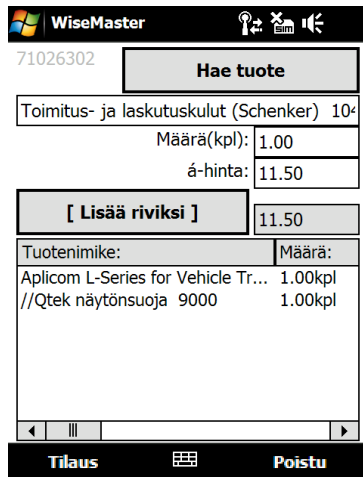


KUVA 22. Tilauspuolen nappiruutu (12)



KUVA 23. Asiakkaan haku WMMobiilissa (12)

Kun tilaukselle on haettu asiakas, syötetään itse tilaus. Tilauksen kirjaaminen tapahtuu tilausnäköymästä (Kuva 24), tuote haetaan tuotetietokannasta ja määritetään tilaukselle määrä sekä tuotteen kappalehinta. Tilaukselle voidaan määrittää myös tilausviite, sekä muuttaa maksuehtoa tarvittaessa. Kun tilaus on kokonaisuudessaan syötetty ja tarvittavat viitteet ja tiedot lisätty tulostetaan tilausvahvistus. Tilausvahvistuksen tulostus onnistuu käyttäjän mukana olevalla Bluetooth-printterillä. (12.)



KUVA 24. Tuotteiden syöttö tilaukselle WMMobiilissa. (12)

## 5.8 Mukautetut järjestelmät

Esimerkki WiseMasterin huolto- ja tilaussovelluksista mukautetusta järjestelmästä on Jokilaaksojen Pelastuslaitoksen käyttämä Palotarkastusjärjestelmä. Jokilaaksojen Pelastuslaitoksen WiseMasterMobiiliin on rakennettu aivan omanlainen toteutus juuri tarkastus toimintaa silmälläpitäen. (12.)

WiseMasterin tuomia etuja palotarkastukselle on tarkastustenhallinta sähköisesti, toiminnan suunnittelu, seuranta sekä raportointi. Kentällä käyttäjää helpottaa ohjeiden ja historiatietojen reaaliaikainen saatavuus, raporttien tulostus suoraan asiakkaalle sekä mobiilipäätelaitteen tuoma nopeutus tarkastusrutiineihin. (12.)

## 6 OHJELMISTOPROSESSIN LUONTI

Opinnäytetyön aiheena oli olemassa olevien ohjelmistotuotteiden tuotekehitysprosesseihin perehtyminen ja toimintaprosessien dokumentoiminen. Työ toteutettiin osallistumalla kahdessa todellisessa asiakasprojektissa tapahtuviin uusiin ominaisuuksien toteutusprosesseihin. Toteutettavat uudet ominaisuudet olivat osa pieniä, toiset taas isompia kokonaisuuksia, jotka toteutettiin M-Technologyn tuotekehitysprosessin mukaisesti. Osallistuin näiden ominaisuuksien kehitysprosessiin alkaen aina asiakkaan tarvekuvauksesta päättyen ominaisuuden käyttöönottoon ja koulutukseen.

Työssä laadittiin alustava projektisuunnitelma, jota noudatettiin projektien edetessä soveltaen. Suunnitelma koostuu neljästä ohjelmistokehityksen vaiheesta, vaatimusmäärittely-, toteutus-, testaus- ja käyttöönottovaihe.

Projektisuunnitelma tehtiin yhteistyössä M-Technologyn työntekijöiden kanssa. Muodostin oman näkemykseni suunnitelman rungosta, jota sitten vielä hiottiin henkilöstön kanssa M-Technologyn tarpeita vastaavaksi. Suunnitelma muokkautui vielä hieman toteutusvaiheessa, mutta runko ja pääpiirteet pysyivät kutakuinkin samoina.

### 6.1 Vaatimusmäärittely

Vaatimusmäärittelyn tavoitteena on selvittää ohjelmistolle asetettavat vaatimukset, mitä ohjelmiston tulee pitää sisällään ja mitä ei. Vaatimusmäärittelystä muodostuu dokumentti jonka perusteella haluttu ohjelmisto pitää olla tuotettavissa.

Tämä vaatimusten määrittely koskee pieniä ominaisuuksia, ei suuria kokonaisuuksia. Muodostetaan pohjadokumentaatio vaatimusten määrittelyn tekemiseksi. Dokumentissa huomioidaan vähintään seuraavat vaatimustenmäärittelyn osa-alueet:

- Määritellään, mitä on myyty ja kenelle ja mitä asioita pitää huomioida.
- Käyttökuvaus (mitä, miksi ja miten)
- Asiakkaan erityisvaatimukset
- Kytkeä olemassa olevaan järjestelmään (moduulit, toiminnallisuudet, ympäristöt)
  - WMMobiili, PCClient, PC-ohjelmisto, www-liityntä ym.
- Raportointi / rajapinta vaikutukset
- Käyttöliittymäkuvaus
  - Alustava suunnitelma käyttöliittymän toteutuksesta ja muutoksista
- Käyttötapausten määrittely
- Käyttötapauskäytännöt
- Toiminnon tuotos:
  - Tiedon tallennus (tiedosto, tietokanta, rajapinta jne.)
  - Raportit (ominaisuuden tuottamat raporttimuutokset ym.)
  - Uuden ominaisuuden kytkennät järjestelmän tuottamassa datassa



## 6.2 Toteutusvaihe

Toteutusvaiheessa dokumentoidaan määrittelyvaiheessa ilmenneet epäkohdat ja ratkaisut sekä kirjataan tarkat tiedot muutoksista. Dokumentointi auttaa tarkentamaan vaatimusmäärittelyä, saadaan lisättyä puutteelliset tiedot ja asiat jotka toteutus vaiheessa tulee ottaa huomioon. Dokumentti ohjelmallisesta toteutuksesta sekä muutoksista helpottavat ohjelman muokattavuutta jälkikäteen.

- Kuvaus muutoksen toteutustavasta
  - Moduulit ja osiot joihin muutos on tehty
  - Viittaukset versionhallintaan ja muutoslogeihin
  - Versionmerkinnät
- Käyttökuvaus toteutetusta ominaisuudesta
- Mahdolliset poikkeamat vaatimustenmäärittelystä ja käyttökuvauksesta
- Testauksessa huomioitavat asiat, sekä mahdolliset viitteet testaussuunnitelmiin

## 6.3 Testausvaihe

Testauksen dokumentointi aloitetaan tekemällä testaussuunnitelma. Testaus-suunnitelmasta selviää testauksen laajuus, aikataulu ja se, millaisia tuloksia testaukselta odotetaan. Testauksen suorittamisen jälkeen tehdään testausraportti, jonka avulla käydään läpi virheiden laatu, virheiden syntyminen syy sekä tapa jolla virheet olisi voitu estää.

- Testaussuunnitelman laadinta
- Testauksen suorittaminen ja testauksen dokumentointi
- Testaustulosten sekä testiaineiston ja raporttien arkistointi

## 6.4 Käyttöönotto

Käyttöönottovaiheessa luodaan valmiista ominaisuudesta käyttöohje sekä vedetään projekti yhteen ja arkistoidaan projektin aikana syntynyt dokumentaatio.

- Käyttöohje dokumentin tekeminen
- Koulutuksessa huomioitavat asiat
- Tekninen dokumentti
  - Asennusympäristö
  - Rekisterimerkinnät
- Käyttöönoton materiaalin arkistointi

## 6.5 Esimerkitapaus

Työssä läpikäytiin kaksi asiakasprojektia. Ensimmäinen projekti toteutettiin oululaiselle metallialan yritykselle Lujateko Oy:lle, ja toinen ympäri Suomen toimivalle Hiven Oy:n myyntiorganisaatiolle. Käytän työn esittelyssä esimerkkejä Lujateko Oy:n projektista.

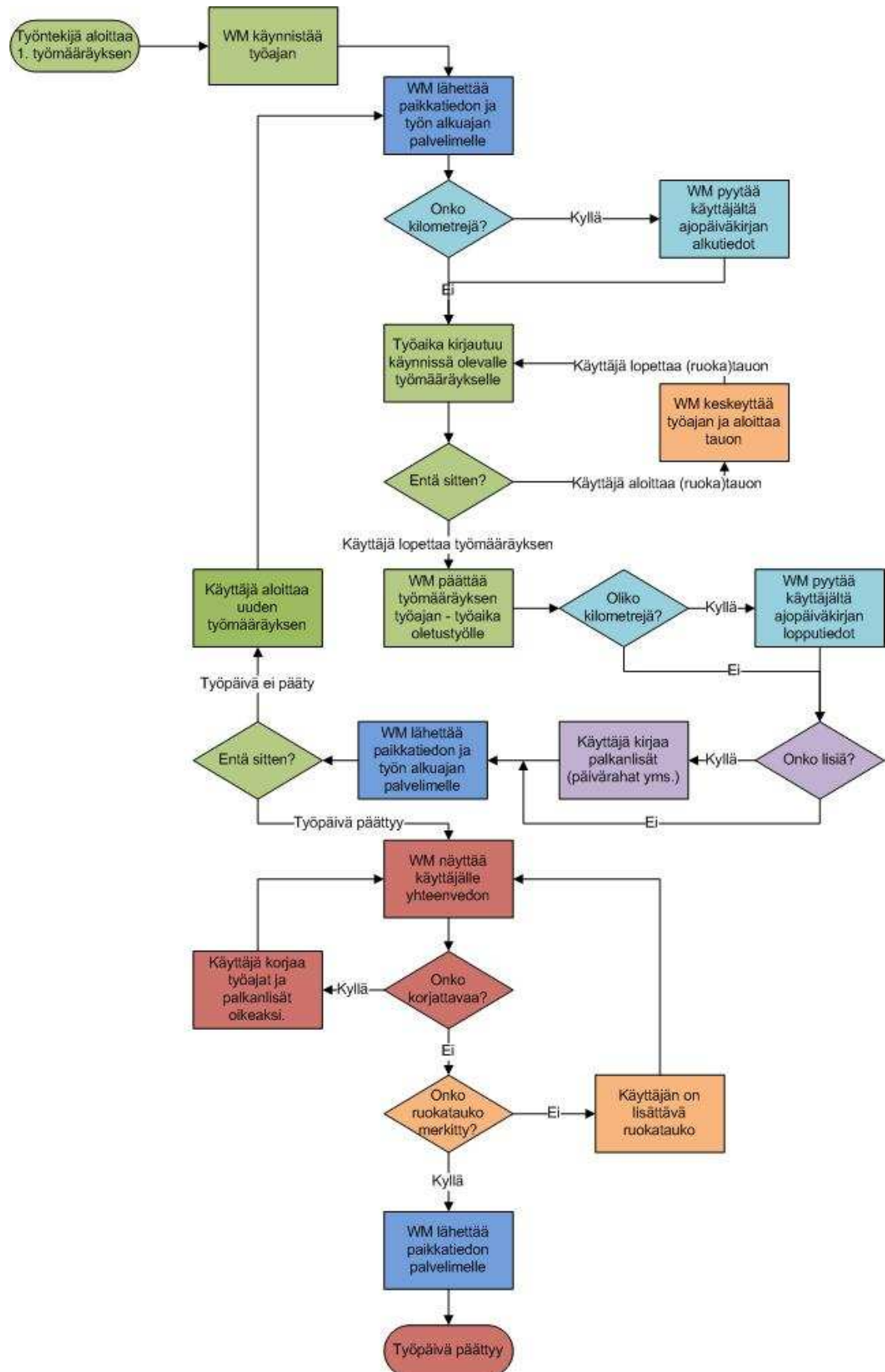
### 6.5.1 Vaatimusmäärittely

Lujateko Oy:n tarve oli työajanseurannan parantaminen. WiseMasterin avulla Lujateolla saadaan kerättyä liikkuvien työntekijöiden työaika reaaliajassa. Lisäksi WiseMasterilla piti pystyä keräämään myös päivärahat yms. lisät ja pitämään ajopäiväkirjaa. Alussa määritettiin Lujateon WiseMasterista poikkeavat vaatimukset, esimerkki alla Lujateon projektista. Ohjelmiston vaatimustenmäärittely tapahtui asiakkaan kanssa yhteistyössä. Alla listattu Lujateon projektin vaatimuksia.

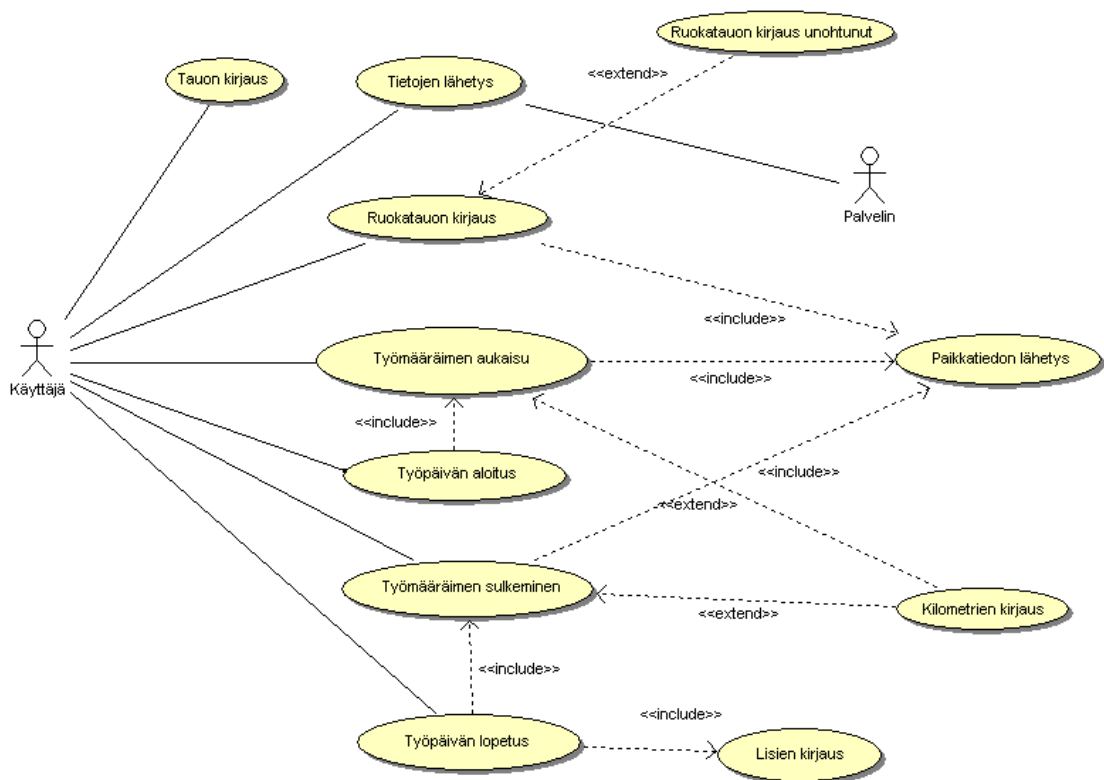
Työntekijöillä on kuukausipalkat. Päivärahat ja kilometrit maksetaan työajan ja ajopäiväkirjan perusteella.

- Työaika pitää pystyä keräämään reaaliajassa (kellokortti).
- Joka työpäivä pitää olla pakotettu palkaton ruokatauko.
- Työn aloittamisesta, lopettamisesta sekä ruoka tauosta pitää saada paikka tiedot.
- Työ pitää olla aina kohdistettu johonkin työmääräimeen.
- Työajan lisät kohdistetaan työmääräimeen.

Lujateon projektissa suurimmat muutokset kohdistuivat mobiiliin käyttöjärjestelmään. Vaatimusmäärittelyssä seuraavaksi määriteltiin toimintakaavio (Kuva 25) mobiilille, sekä käyttötapauskaavio (Kuva 25). Toimintakaavion värit kertovat mitä osioita WiseMasterista löytyy jo valmiina, mitkä ovat uusia ominaisuuksia ja mitkä ominaisuudet tulevat rekisterimerkin taakse.



KUVA 25. Lujateko Oy, toimintakaavio.



KUVA 26. Lujateko Oy, käyttötapauskaavio

Kaavioilla saatiin hahmotettua järjestelmän toiminnallisuutta (toimintakaavio), sekä kuvattua järjestelmän väliset suhteet ja käyttötapauksiin osallistuvat ulkoiset toimijat (käyttötapauskaavio). Toimintakaavio oli myös hyvä havainnollistamisväline, kun asiakkaan kanssa käytiin määrittelyvaiheessa ohjelmiston toimintaa läpi. Käyttötapauskaavion avulla saatiin kuvattua järjestelmän eri osalueiden suhteita, sekä järjestelmän ja käyttäjien välistä vuorovaikutusta. Käyttötapauskaavio helpotti myös sekvenssikaavioiden luontia.

### 6.5.1.1 Kytkenöt olemassa oleviin järjestelmiin

Määrittelyvaiheessa käydään läpi myös kytkökset muihin jo olemassa oleviin järjestelmiin, kuitenkin tässä vaiheessa niitä enempää määrittelemättä. Seuraavassa on esimerkki Lujateon projektista.

Sähköinen työajanseuranta toimii WiseMasterMobiilin lisänä. Muutoksia tulee myös PCWiseMasteriin sekä WiseMasterPCClientiin.

PC Wisemaster:

- Työluokkatietojen ylläpito
- Raportointi

(PC client:)

- Samat toiminnot kuin PDA

Mobiili:

- Muutokset projektikuvauksen kappaleessa yksi

WMFileservice

- Seriaalisaatiot muuttuneet tiedostojen sisällä

WMWeb

- Työaikatietojen näyttö (www-heijastus)

Paikannus palvelin

- Paikkatietoon lisää tietoa, palvelimen pitää osata käsitellä (työpäivän aloitus, jne)

MT seuraaja

- Osattava näyttää käyttäjälle paikannuspalvelimen data

### **6.5.1.2 Raportointi ja rajapinta vaikutukset**

Raportointi ja rajapintavaikutukset tulee mainita aina dokumentissa vaikka niitä ei olisikaan. Lujateon projektissa rajapinta vaikutuksia ei ollut, koska lujateolla ei ollut järjestelmää johon WiseMaster olisi liitetty. Raportointiin tuli muutoksia, raporteille tuli uutta dataa sekä suodatusta piti muuttaa Lujateolle sopivaksi.

### 6.5.1.3 Käyttöliittymäkuvaus

Käyttöliittymäkuvauksessa määritetään uudet ja muokattavat näkymät. Näkymistä tehdään alustavat käyttöliittymä kuvat, ja kerrotaan näkymien toiminnallisuus. Esimerkkinä katsotaan Lujateon projektista yksi muokattavan näkymän ja yhden kokonaan uuden näkymän määrittely.

Tauko-näkymä tuli Lujateon projektissa kokonaan uutena näkymänä. Näkymän toimintoja on taukojen aloitus ja lopetus, näkymässä myös näkyy työnkesto sekä taukojen yhteiskesto. Näkymää suunniteltaessa tavoite oli tehdä näkymästä mahdollisimman yksinkertainen, ja näkymän napeista mahdollisimman kookkaat jotta niitä olisi mahdollista käyttää jopa työkalusineet kädessä.

Näkymien suunnittelu lähti liikkeelle vaatimusten pohjalta. Vaatimuksina oli töissä- tai poissa tiedon esittäminen käyttäjälle sekä Lujateon tapauksessa olevan pakollisen ruokatauon helppo käyttö. Piirsin näkymät Microsoftin Visual C++ -ohjelmointiympäristöllä, näin ollen näkymistä sai autenttiset ja niitä pystyi mallintamaan päätelaiteella. Esimerkkikuvat näkymistä kuvat 27 ja 28.

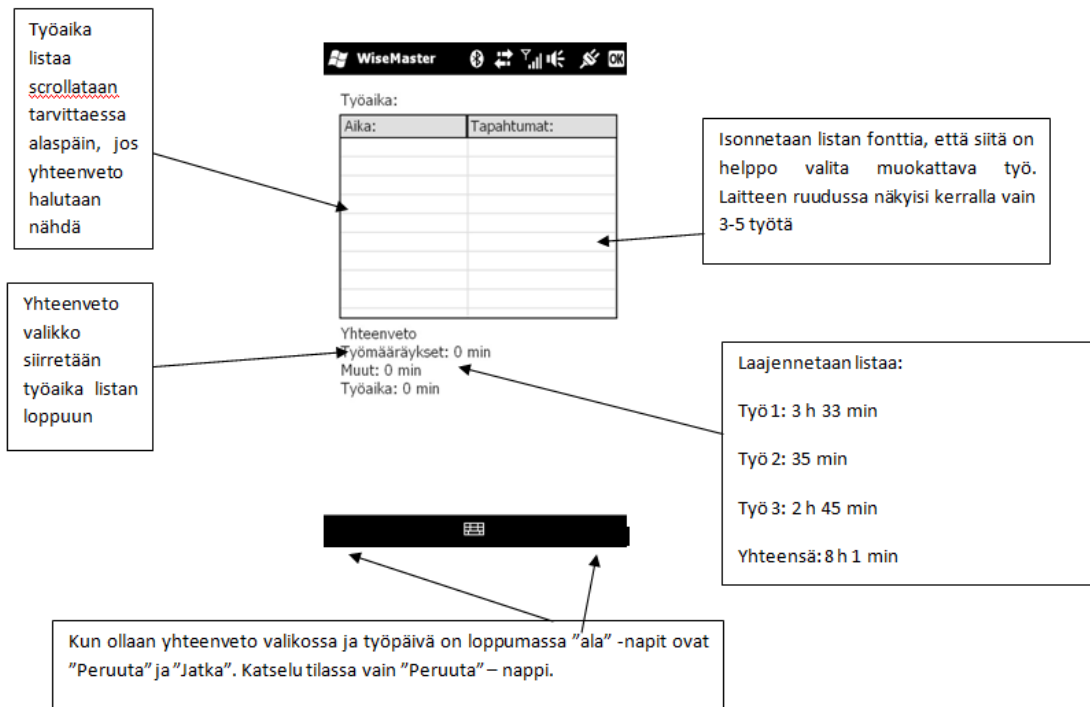


KUVA 27. Lujateko Oy, taukonäkymä 1



KUVA 28. Lujateko Oy, taukonäkymä 2

Työajanyhteenveto näkymä oli jo ennestään olemassa WiseMasterMobiilissa, mutta näkymä oli puutteellinen eikä näin soveltunut Lujateolle. Suunnittelussa pohjaksi otettiin jo olemassa oleva näkymä, määritettiin mitä tietoja näkymästä pitää ilmetä ja sen perusteella tehtiin muutokset ehdotukset. Kuvassa 29 kuva näkymästä ja muutokset ehdotuksista.



KUVA 29. Lujateko Oy, yhteenveton suunnittelu



#### **6.5.1.4 Käyttötapausten määrittely**

Käyttötapausten määrittelyä tehtiin sekvenssikaavioiden avulla. Jokaisesta käyttötapauksesta tehtiin sekvenssikaavio, josta ilmenivät ohjelmistojen väliset vuorovaikutukset. Sekvenssikaaviot tehtiin ohjelman havainnollistamiseksi ja näin helpottamaan ohjelmoitsijan työtä. Liitteessä 1 on esimerkit sekvenssikaavioista Lujateon projektista.

#### **6.5.2 Toteutusvaihe**

Toteutusvaiheessa listattiin ohjelmat joihin muutokset olivat kohdistuneet, Lujateon projektissa WiseMasterPC sekä WiseMasterMobiili. Käytiin toteutuksien vaikutukset läpi ohjelmistoon, liitteessä 2 esimerkki Lujateon projektista.

##### **6.5.2.1 Valmiit käyttöliittymät**

Seuraavaksi esiteltiin valmistuneet käyttöliittymät ja niiden poikkeavuus määrittelyvaiheessa suunniteltuihin. Lujateon projektin aikaisemmin läpi käyty taukonäkymä pysyi lähes samana kuin oli suunniteltukin. Kuvankaappaus valmiista näkymästä on kuvassa 30.



KUVA 30. Lujateko Oy, lopullinen taukonäkymä

### 6.5.2.2 Poikkeamat vaatimusten määrittelystä

Toteutusvaiheen lopuksi listattiin ja perusteltiin poikkeukset, joita toteutusvaiheessa tuli verrattuna määrittelyvaiheessa määriteltyihin ominaisuuksiin. Lujateon projektissa jouduttiin tekemään kaksi poikkeusta määrittelyyn. Liitteessä 3 ote Lujateon projektin poikkeuksista.

Poikkeamat jouduttiin tekemään osin aikataulullisista syistä ja osin WiseMasterin käytön aloittamisen helpottamiseksi. Ajopäiväkirjan käyttöönoton siirtäminen myöhemmälle ajankohdalle osoittautui hyväksi ratkaisuksi, koska käyttäjillä on ollut vaikeuksia jo kertymän kirjaamisen kanssa työmääräyksille. Käyttöönoton jälkeen teimme vielä helpotuksia kertymän kirjaamiseen jotta käyttäjät eivät unohda tai laista sitä toimintoa.

### 6.5.3 Testaus

Testaus lähdettiin toteuttamaan tekemällä testaussuunnitelma. Testaussuunnitelma laadittiin yhteistyössä testauksesta vastaavan Petri Eklundin kanssa. Testaus toteutettiin tiiviissä yhteistyössä Lujateon projektin vastaavan ohjelmoitsijan Sami Turpeisen kanssa. Liitteessä 4 on ote Lujateon projektin testaussuunnitelmasta, kuvat 31 ja 32 ovat esimerkkejä testauksesta syntyneestä dokumentaatiosta.

Päivämäärä: 07.02.2011			
Testi kierros: 1			
Testattava ominaisuus: Työajan yhteenveto	OK	Bugi	Bugi kuvaus:
Kellon aika	x		
Päivämäärä	x		
Tapahtumien oikeellisuus	x		
Työajan yhteenveto		x	Kellon ajat eivät täsmää yhteenvedossa
Taukojen yhteenveto		x	Eivät näy oikein yhteenvedossa
Lisien lisääminen	x		
Lisien tallentuminen		x	Lisä katoaa kun tallentaa ja menee yhteenvetoon
Ruokatauon kysely		x	Ei kysynyt ruokatauko vaikka ei ollut merkitty
Yhteenvedon varmistus ruokatauon lisäämisen jälkeen		x	Ruokatauko väärässä paikkaa
Peruuta napin toiminta		x	Tallensi tapahtumat vaikka peruutettiin

KUVA 31. Lujateko Oy, testausraportti 1

Päivämäärä: 07.02.2011			
Testi kierros: 2			
Testattava ominaisuus: Työajan yhteenveto	OK	Bugi	Bugi kuvaus:
Kellon aika	x		
Päivämäärä	x		
Tapahtumien oikeellisuus	x		
Työajan yhteenveto		x	Kellon aikoja edelleen päällekin
Taukojen yhteenveto		x	Näkyvät virheellisesti
Lisien lisääminen	x		
Lisien tallentuminen	x		
Ruokatauon kysely	x		
Yhteenvedon varmistus ruokatauon lisäämisen jälkeen		x	Ruokatauko väärässä paikkaa
Peruuta napin toiminta		x	Tallentaa vieläkin tapahtumat

KUVA 32. Lujateko Oy, testausraportti 2

#### **6.5.4 Käyttöönotto**

Käyttöönotto vaiheeseen kuuluu käyttöohjeen tekeminen, laitteiden vakioinnit, esiasennukset sekä asiakkaan koulutus. Käyttöohjeita tehtiin kaksi, WiseMasterPC-ohjelmalle sekä WiseMasterMobiilille. WiseMasterPC:lle tehtiin yleiskäyttöohje joka sisälsi WiseMasterin yleisenkäytön lisäksi uusina ominaisuuksina tulleiden raportoinnin sekä työajan muokkauksen ohjeet. WiseMasterMobiilille tehtiin käyttöohje koskemaan kattavasti koko mobiiliohjelmaa.

Laitteet vakioitiin eli asetettiin tarvittavat asetukset. Koska Windows Phone -käyttöjärjestelmä eroaa valtaosan käyttämistä järjestelmistä, laitteet on hyvä olla käyttövalmiina kun ne luovutetaan asiakkaalle. Motorola es400 -laitteisiin asennettiin WiseMasterMobiili-käyttöjärjestelmä ja jokaiseen laitteeseen ladattiin valmiiksi työntekijöiden omat profiilit.

Käyttöönotto tapahtui asiakkaan tiloissa 11.2.2011 pidetyn koulutuksen yhteydessä. Aamulla käytiin toimiston väen kanssa läpi WiseMasterPC:n toiminta, iltapäivä koulutettiin asiakkaan työntekijöitä WiseMasterMobiilin käyttöön.

## 7 TYÖN ARVIOINTI

### 7.1 Käytetyt menetelmät

M-Technology Oy:lle suoritetuissa ohjelmistoprojekteissa löytyi kaikki tunnusomaiset piirteet, mitä ohjelmistotuotantoon liittyy. Projektien kulusta erottui hyvin määrittely-, suunnittelu-, ohjelmointi-, testaus-, käyttöönotto- ja ylläpitovaihe sekä ainakin jossain määrin tuotteenhallinta, laadunvarmistus, dokumentointi ja vaatimustenhallinta. Kaikkien osa-alueiden löytyminen oli ehkä jopa hieman yllättävää, koska esimerkki projektien yhteydessä ei noudatettu ennalta määritettyä toimintamallia.

Pääosin ohjelmistotuotanto noudatti ihan perinteistä vesiputousmallia, joskin yrityksen koon ja sitä myöten aikataulutuksen myötä voidaan sanoa, että kaikki vaiheet toteutettiin testauksesta tutuksi tulleen XP:n mukaan. Pienehkön yrityksen ohjelmistoprojekteissa mukana olevien ihmisten kommunikointi keskenään sekä asiakkaan kanssa on ensiarvoisen tärkeää projektien onnistumisen takaamiseksi. Huolimatta tiukasta aikataulutuksesta M-Technologyssä kommunikointi toimi hyvin niin projektipäällikön, ohjelmoitsijan kuin testauksesta vastaavien henkilöiden kesken.

Pääosin kaikki vaiheet toteutettiin ketteriä ohjelmistomenetelmiä käyttämällä. Selvimmin ketterät menetelmät tulivat esille testauksessa, joka toteutettiin lähes kokonaan soveltamalla XP:tä. Myös ohjelmistokehityksestä löytyi viitteitä Scrum-menetelmästä, joskin kaikki Scrumin tunnusmerkit eivät täyty aikataulutuksen ja henkilöstömäärän rajallisuuden vuoksi.

UML-kielen käyttö työssä tuntui itsestä luontevalta tavalta kuvata ohjelmistoprosesseja. Niiden käyttö todettiin hyödylliseksi suunnitteluvaiheessa sekä toteutusvaiheessa ohjelmoitsijan työn kannalta. Ohjelmoitsijan puolelta käyttötapauskaavioiden sekä sekvenssikaavioiden lisäksi työtä nopeuttava merkitys oli valmiilla käyttötapauskuvauksilla.

## 7.2 Esille tulleet ongelmatilanteet

Esimerkissä läpikäydyn Lujateko Oy:n projektin suurimmat haasteet ja ongelmat olivat asiakkaan kanssa tapahtuvan ohjelmiston määrittelyn vaikeudessa. Määrittelyä suoritettiin tiiviissä yhteistyössä asiakkaan kanssa käymällä useita kertoja asiakkaan luona demoamassa ja esittelemässä ohjelmistoa ja suunnitelmia. Työn määrittely ja tekovaiheessa kaikki tuntuikin menevän hyvin ja kaikki suurimmat asiat oli otettu huomioon, lukuun ottamatta pieniä viilauksia. Joita jouduttiin tekemään vielä toteutusvaiheessa.

Kun käyttöönotto päivä koitti ja aloimme kouluttaa asiakkaan työntekijöille WiseMasterMobiilin käyttöä, huomasin asiakkaan projektin avainhenkilö ettei työmääräyksien käyttö voi heillä näin toimia. Asian julkitulo näinkin myöhäisessä vaiheessa aiheutti koulutuksessa ja projektin vetäjissä suurehkoa hämmennystä.

Asia ei ollut tullut ilmi koskaan aikaisemmissa määrittelyvaiheissa eikä edes valmiin tuotteen esittelyssä. Kun asia saatiin koulutuksessa asiakkaan kanssa hieman kiivaastikin puitua, saimme jatkettua koulutuksen loppuun. Valitettavasti asian selvittely vei tärkeää koulutus-aikaa työntekijöiltä.

Päätimme toteuttaa asiakkaan vaatiman ominaisuuden osana projektia, vaikka sopimus- tai määrittelyvaiheissa ei tällaisesta ominaisuudesta ole ollut puhetta. Tosiasia on, että kun pienehkö yritys toimii asiakasprojekteissa, tulee tilanteita jolloin on mukauduttava asiakkaan pyyntöihin, vaikka ne eivät edellä esitetyn tilanteen mukaan projektin määrittelyyn kuuluisikaan.

Työntekijöille järjestettiin myös lisäkoulutus M-Technologyn tiloissa kertymän kirjauksesta ja WiseMasterMobiilin yleisestä käytöstä.

## 8 YHTEENVETO

Työn tarkoituksena oli luoda M-Technology Oy:n ohjelmistoprojektien toteutus mallipohja, jota käytetään uusien ominaisuuksien toteutukseen. Malli koostui neljästä ohjelmistosuunnittelun vaiheesta, suunnittelu-, toteutus-, testaus- sekä käyttöönottovaiheista. Malli tulee toimimaan pohjana aina, kun lähdetään toteuttamaan uusia ohjelmistoprojekteja.

Työtä tehdessä M-Technologyssä paneuduin yleisimpiin ohjelmistosuunnittelumalleihin sekä mallintamismenetelmiin. Työssä en lähtenyt suoraan soveltamaan tiettyjä ohjelmistosuunnittelumalleja, vaan sovelsin useampia malleja yrityksen resurssit sekä jo aikaisemmat toimintatavat huomioon ottaen. Mallien yhdistäminen suoraan tiettyyn ohjelmistoprojektin vaiheeseen olisi ollut jopa mahdotonta. Projekteissa ilmenneissä toimintatavoissa löytyi viitteitä niin Scrumista kuin XP:stäkin sekä myös osittaisia toimintamalleja myös muista malleista.

Projekteissa suurimmat ongelmat syntyivät vaatimustenmäärittelyvaiheessa. Määrittely epäonnistui ensimmäisessä Lujateon projektissa, joka käytiinkin kappaleessa 7.2 läpi. Määrittelyn epäonnistuminen ja julkitulo näinkin myöhäisessä vaiheessa tulevat yritykselle kalliiksi ja näin heikentävät projektien kannattavuutta. Toisessa Hiven Oy:n projektissa määrittely sujui tiiviissä yhteistyössä asiakkaan kanssa aina ohjelmiston valmistumiseen asti. Määrittely onnistui Hivenen projektissa hyvin. Tosin määrittelyn rajaus ei ajoittain pysynyt sovittuna asiakkaan lisävaatimusten johdosta.

Yrityksen olisi hyvä huomioida ja tiedostaa ohjelmistosuunnittelumallien käyttö ja käyttää niitä enemmän hyväksi projektien toteutuksessa. Opinnäytetyössä syntyneen projektipohjan käyttö sekä edelleen kehittäminen tulevaisuudessa on M-Technologylle erittäin suositeltavaa ohjelmistoprojektien läpi viemisen sekä kannattavuuden parantamiseksi.

## LÄHTEET

1. Kollanus Sami 2003-2005. Johdatusta ohjelmistotekniikkaan. Saatavissa:  
<http://users.jyu.fi/~kolli/ohjelmistotekniikka/files/johdatustaOT.pdf>.  
Hakupäivä 19.6.2011.
2. Haikala Ilkka – Märijärvi Jukka 2004. Ohjelmistotuotanto. Helsinki: Talentum.
3. Forselius Pekka 1999. Ohjelmistojen koon mittaaminen eri tyyppisissä kehityshankkeissa. Saatavissa:  
<http://www.pcuf.fi/sytyke/lehti/kirj/st19991/14.pdf>. Hakupäivä 21.6.2011.
4. Taina Juha 2005. Ohjelmistotuotanto. Saatavissa:  
<http://www.cs.helsinki.fi/u/taina/ohtu/k-2005/2luku1.pdf>.  
Hakupäivä 21.6.2011.
5. Paakki Juha – Laine Harri. Ohjelmistotuotanto. Saatavissa:  
<http://www.cs.helsinki.fi/u/paakki/ohtuk03-luento1-bw.pdf>.  
Hakupäivä 21.6.2011.
6. Immonen Jarkko 2002. Johdatus Ohjelmistotuotantoon. Saatavissa:  
[http://cs.joensuu.fi/~jimmonen/jot\\_moniste/jot\\_moniste\\_121.html](http://cs.joensuu.fi/~jimmonen/jot_moniste/jot_moniste_121.html).  
Hakupäivä 21.6.2011.
7. Ketterä ohjelmistokehitys 2011. Wikipedia vapaa tietosanakirja. Saatavissa:  
[http://fi.wikipedia.org/wiki/Ketter%C3%A4\\_ohjelmistokehitys](http://fi.wikipedia.org/wiki/Ketter%C3%A4_ohjelmistokehitys).  
Hakupäivä 21.6.2011.
8. Lappalainen Vesa 2009. Agile. Saatavissa:  
<http://users.jyu.fi/~vesal/kurssit/ohjelmointi2009/materiaali/agile.html>.  
Hakupäivä 22.6.2011.
9. Scrum 2011. Wikipedia vapaa tietosanakirja. Saatavissa:



<http://fi.wikipedia.org/wiki/Scrum>. Hakupäivä 22.6.2011.

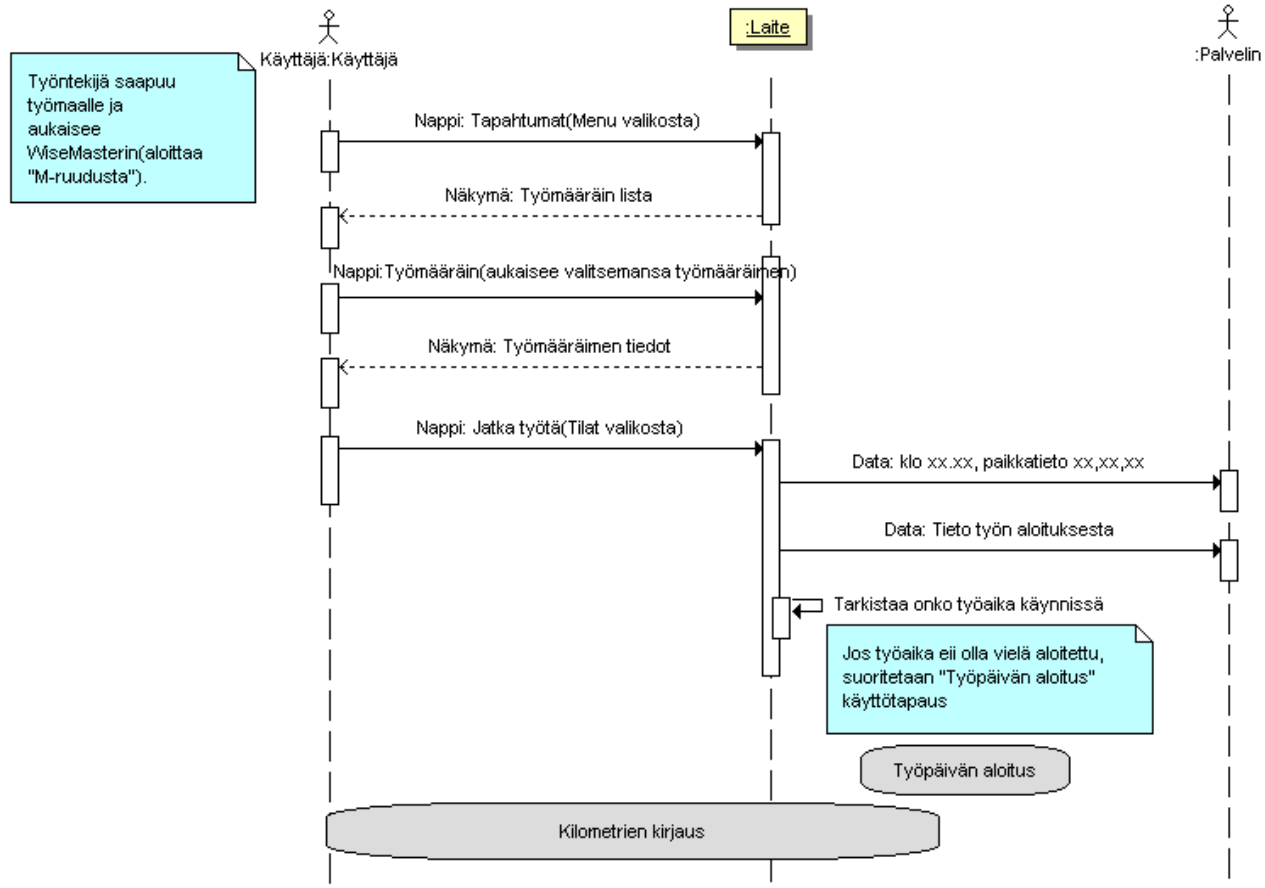
10. Koskimies Kai 2000. Oliokirja. Helsinki: Satku - Kauppakaari.

11. WiseMaster 2011. Saatavissa: [www.wisemaster.fi](http://www.wisemaster.fi).  
Hakupäivä 25.5.2011.

12. WiseMaster 2011. Yrityksen sisäinen dokumentaatio. M-Technology Oy.

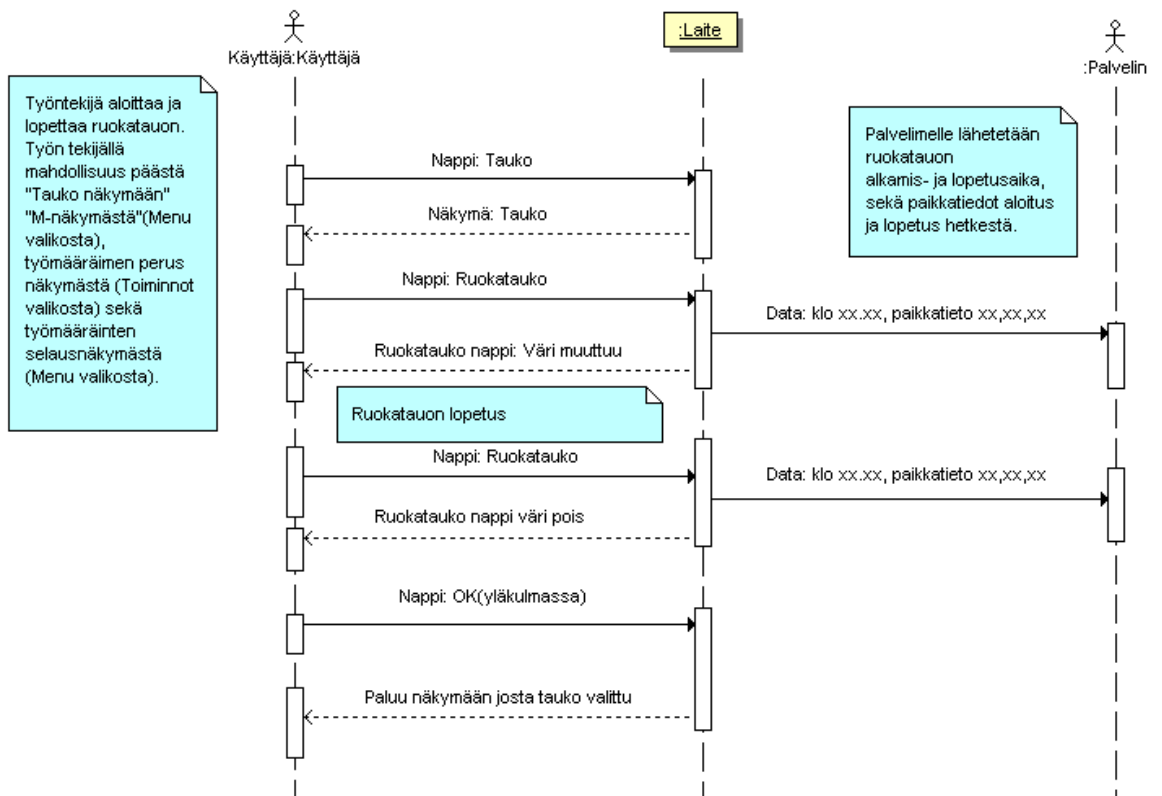
## Työmääräimen aukaisu/aloitus

Työntekijä aukaisee työmääräin listan, valitsee työmääräimen ja aloittaa työn.



## Ruokatauon kirjaus

Ruokatauko kuuluu pakollisena osana työntekijän päivään. Työntekijä kirjaa ruokatauon alkaneeksi kun aloittaa ruokatauon työmaalla, ja päättää ruokatauon kirjaamalla työnjatkuvaksi. Ruokatauon aloituksesta sekä loppumisesta lähetetään reaaliaikaisesti palvelimelle kellonajat sekä paikkatiedot.



## TOTEUTUSVAIHE

### Toteutustapa

Ohjelmisto muutokset toteutettiin kahteen eri ympäristöön, WiseMasterPC sekä WiseMaster mobiiliin. Pc ympäristön ohjelmointi kielenä on Visual C++ ja mobiilin puolella Windows *Embedded*. Toteutus suoritettiin käyttämällä mahdollisuuksien mukaan vanhaa koodia, mutta myös vanhaa toteutusta korvattiin ja tehtiin uutta jotta Lujateon muutokset saatiin hyödynnettyä myös muille asiakkaille.

### Toteutuksien vaikutus

Muutokset aiheuttivat suurimman työn mobiiliin puolella, jonne suurimmat kokonaan uudet ominaisuudet tulivat. Alla muutokset listattuna ja eriteltyinä PC:n ja mobiilin välillä.

### Wisemaster PC

WiseMasterPC:n muutokset:

- Työn raportointi:
  - o Työnraportointi tehtiin kokonaan uudestaan vastaamaan asiakkaan tarpeita.
- Työajan muokkaus
  - o Työajan muokkaus toiminto tehtiin näkymän puolelta kokonaan uudestaan sekä muokattiin edellistä toiminnallisuutta

### WiseMaster mobiili

WiseMaster mobiilin muutokset:

- Tauko näkymä
  - o Täysin uusi ominaisuus
- Työajan yhteenveto näkymä
  - o Paranneltiin käyttöliittymää sekä toiminnallisuutta
- Pakollinen ruokatauko
  - o Täysin uusi toiminnallisuus
- Lisien lisäys ja muokkaus näkymä
  - o Uusi näkymä sekä toiminnallisuus
- Lisien kyselyn wizardi
  - o Uusi näkymä sekä toiminnallisuus
- Automaattinen työajan lähetys

- Muokattiin vanhaa toteutusta halutunlaiseksi

### **WM Fileservice**

Fileserviceä ei otettu lujateolle käyttöön, koska kokonaisuus ei ole Lujateon tapauksessa niin suuri että se olisi järkevää.

### **WMWeb**

Web käyttöliittymää ei otettu vielä käyttöön, mahdollisesti kytketään myös Lujateolle myöhemmin päälle.

## **Poikkeamat vaatimusten määrittelystä**

Muutama poikkeus jouduttiin tekemään vielä suunniteltuun toteutusvaiheessa, johtuen asiakkaan muuttuneista vaatimuksista, aikataulutuksesta sekä käyttäjien WMMobiilin käytön aloituksen helpottamiseksi.

## **Ajopäiväkirja**

Ajopäiväkirja moduulin toteutus ja käyttöönotto päätettiin jättää myöhempään ajankohtaan, selventääkseen ohjelman käyttöönottoa. Ajopäiväkirja olisi hankaloittanut entisestään varsin kokemattomien ja jo iäkkäidenkin työmiesten ohjelmankäyttöä joten päädyimme jättämään sen pois liikkeelle lähtö versiosta. Päätökseen vaikutti myös se että Lujateko Oy ei ollut tilannut ajopäiväkirja toimintoa vaan se oli tulossa ns. lisänä mukaan kauppaan.

Ajopäiväkirja tullaan ottamaan käyttöön kun asiakas on omaksunut WiseMasterin käytön ja kun muut muutokset on toteutettu. Käyttöönotto tapahtuu automatisoidulla päivityksellä.

## **Paikannus**

Paikannus toimintoa ei saatu tehtyä valmiiksi määräpäivään, asian monimutkaisuuden sekä muiden projektien kiireellisyydestä johtuen. Paikannustoiminnon siirtämiseen muiden projektien tieltä vaikutti myös se että se olisi lisännyt käyttöönoton yhteydessä virheherkkyyttä sekä hidastanut järjestelmää.

Paikannus otetaan käyttöön myöhempänä ajankohtana. Se tullaan toteuttamaan verkkopaikannusta hyväksi käyttäen, verkkopaikannus valittiin GPS:n tilalle johtuen paikannustarkkuuden vaatimuksista sekä toteutukseen liittyvien seikkojen takia.

## TESTAUSSUUNNITELMA

### Toteutus

Testattava ohjelma: WiseMaster Mobiili, Lujateko Oy:n ympäristö

Päätelaite: Motorola es400

Aineisto: Asiakkaan aineisto

Testauksen suorittaa:

- Petri Eklund:
  - o Vastaa testauksen toteutuksesta sekä laajuudesta
  - o Suorittaa ympäristön kokonaisuuden tarkistamisen ja verifioimisen
- Alexi Palosaari:
  - o Testauksen suunnittelu ja suoritus

Vastaava ohjelmoitsija:

- Sami Turpeinen

Pääpaino testissä painottuu uusien ominaisuuksien sekä näkymien testaamiseen, testaus aloitetaan kokonaiskatselmuksella ohjelmasta. Kun näkymät ja niiden toiminnot ovat verifioitu oikeiksi, aloitetaan ominaisuuksien testaus. Testaus suoritetaan käyttämällä ketterää testausta, päätös ohjelman palauttamisesta takaisin toteutukseen suoritetaan Petrin ja Aleksin yhteispäätöksellä.

### Ominaisuudet

#### Testattavat ominaisuudet

- Työajan kirjautuminen päälle työmääräyksen käynnistämisen yhteydessä
- Taukojen toiminta
  - o Ruokatauko
  - o Muu tauko
- Tauko näkymän työajan ja tauko ajan oikeellisuus sekä nappien värin vaihdon toiminta
- Lisien lisäys näkymä
  - o Oikea työmääräimen nimi
  - o Oikeat päivämäärät ja kellon ajat
  - o Lisien lisäys näkymän sisältö
  - o Lisien määrän toiminta, myös erikoiset luvut
  - o Lisän poistaminen listalta