

Jussi Saari

Ohjaustyökalujen suunnittelu ja kehitys

Opinnäytetyö

Kevät 2015

SeAMK Tekniikan yksikkö

Bio- ja elintarviketekniikan koulutusohjelma

SeAMK 

SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikka

Koulutusohjelma: Bio- ja elintarviketekniikka

Suuntautumisvaihtoehto: Lihateknologia

Tekijä: Jussi Saari

Työn nimi: Ohjaustyökalujen suunnittelu ja kehitys

Ohjaaja: Jarmo Alarinta

Vuosi: 2015

Sivumäärä: 32

Liitteiden lukumäärä: 0

Opinnäytetyössä käydään raportinomaisesti läpi vaihe vaiheelta Excel-pohjaisten suunnittelutyökalujen valmistaminen yhteistyöyrityksen tuotannonohjauksen käyttöön. Työn tavoitteena oli paitsi täyttää yrityksen toivomukset työkalujen suhteen myös luoda pohja muille suunnittelijoille vastaavanlaisten työkalujen valmistamiseksi. Lisäksi työssä esitellään Microsoft Excel-funktioiden käyttöä sekä VBA-ohjelmointia suunnittelijan näkökulmasta.

Varsinainen yritykselle tehty tuotannonohjauspaketti sisälsi päiväkohtaiset valmistusmäärät kaikille tuotteille 4 viikkoa eteenpäin. Näiden valmistusmäärien avulla raaka-ainetilauksia ja konekapasiteetteja saatiin tarkennettua entisestään. Työ koostui pääasiassa erilaisten ongelmien ratkomisesta Microsoft Excel –ympäristössä.

Avainsanat: tuotannonohjaus, Excel, Visual Basic for Applications

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: Engineering

Degree programme: Biotechnology and Food Engineering

Specialisation: Meat technology

Author/s: Jussi Saari

Title of thesis: Creating and developing Microsoft Excel-based resource planning tools

Supervisor(s): Jarmo Alarinta

Year: 2015 Number of pages: 32 Number of appendices: 0

This thesis presents in the form of a report on the process of creating resource planning tools for the company that ordered the work. The objective of the work was not only to fulfil the requirements that the company had for this work but also to create a basis for the other resource planners at the company for creating similar tools. This thesis also covers the use of Microsoft Excel functions and VBA-programming.

The main work that was carried out for the company was a resource planning tool that included daily production amounts for all the company's products for 4 weeks ahead. These production amounts can be used to improve the company's resource handling and machine capacity. The work was completed with the help of Microsoft Excel.

Keywords: resource planning, Excel, Visual Basic for Applications

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ	4
Käytetyt termit ja lyhenteet	6
1 JOHDANTO.....	7
2 TAUSTATIETOA	8
2.1 Tuotannonsuunnittelu.....	8
2.2 Tietovarastot	10
2.3 Microsoft Excel –funktiot	11
2.3.1 PHAKU ja VHAKU.....	11
2.3.2 SUMMA.JOS.....	13
2.3.3 INDEKSI	14
2.3.4 VASTINE	15
2.3.5 Matriisikaavat.....	16
2.4 Microsoft Excel - VBA	17
2.4.1 Esimerkkiohjelmointi	17
3 TYÖN TOTEUTUS	22
3.1 Lähtökohdat	22
3.2 Toimintaan perehtyminen.....	22
3.3 Työn suunnittelu.....	23
3.4 Suunnittelutyökalujen valmistus	24
3.4.1 Tuotetiedot.....	24
3.4.2 Myynti- ja ennustetietojen haku tietovarastoraporteista	25
3.4.3 Varastosaldon laskeminen saldoraportista	27
3.4.4 Tuotantosuunnitelma.....	28
3.4.5 Tuotantolistat	29
4 TYÖN TARKASTELU.....	30
LÄHTEET	31
Kuvio- ja taulukkuuettelo	

Kuvio 1. Tilaus-toimitusprosessi.....	9
Kuvio 2. Tuotannonsuunnittelun pääkohdat	10
Kuvio 3. Tietovarastoraportin luominen eri vaiheet	11
Kuvio 4. PHAKU	12
Kuvio 5. SUMMA.JOS	14
Kuvio 6. INDEKSI	15
Kuvio 7. VASTINE	16
Kuvio 8. PHAKU-funktion korvaus INDEKSI- ja VASTINE-funktioiden avulla	16
Kuvio 9. Esimerkki tavallisen kaavan korvaamisesta matriisikaavalla	17
Kuvio 10. Tuotantosuunnitelma edellisenä iltana ennen makron käyttöä	18
Kuvio 11. Tuotantosuunnitelma seuraavana aamuna makron käyttämisen jälkeen.....	18
Kuvio 12. Esimerkki valmiista ohjelmointikoodista	21
Kuvio 13. Asiakassuhteet.....	23
Kuvio 14. Hierarkia	24
Kuvio 15. Tuotetiedot.....	25
Kuvio 16. Tietovarastoraportin esimerkinäkymä.....	26
Kuvio 17. Tietovarastoraportin esimerkkimuunto	27
Kuvio 18. Saldolaskenta	28
Kuvio 19. Vähimmäisvalmistuksen laskenta	28
Kuvio 20. Tuotantosuunnitelma.....	29
Kuvio 21. Tuotantolista, jossa ennuste korjataan tarvittaessa lisätilauksella	29

Käytetyt termit ja lyhenteet

VBA Visual Basic for Applications. Microsoftin kehittämä ohjelmointikieli.

Lean Ajattelufilosofia, joka pyrkii arvoa tuottamattomien toimintojen minimointiin.

1 JOHDANTO

Opinnäytetyöni sai alkunsa yrityksen tuotannonohjauksen kehitystarpeesta, jossa haluttiin laajentaa suunnittelun aikaikkunaa muutamasta päivästä useisiin viikkoihin. Erityisesti uusilla työkaluilla haluttiin kehittää yrityksen materiaali-, raaka-aine- ja resurssihallintaa kohti Lean-ajattelumallin johtamisfilosofiaa.

Aloitin työn suorittamisen noin 3-4 kuukauden tuotannonohjaukokemuksen jälkeen. Tätä ennen olin tutustunut ohjattavan osaston esimies- ja tuotantotehtäviin useiden vuosien ajan työskennellessäni osastolla. Ensitöikseni tutustuin perinpohjaisesti jo olemassa oleviin työkaluihin ja suunnittelin tarkasti, millaisia muutoksia näihin täytyisi tehdä, että yrityksen toivomukset uusien työkalujen suhteen täyttyisivät. Tämän jälkeen alkoi useita kuukausia kestänyt VBA-ohjelmoinnin ja taulukkolaskentaohjelmien opetteleminen esimerkkien, teorian ja testaamisen avulla. Vasta tämän jälkeen aloitin varsinaisten työkalujen valmistamisen. Työkalujen valmistamisen viimeinen vaihe oli niiden sisäänajo tuotantoon.

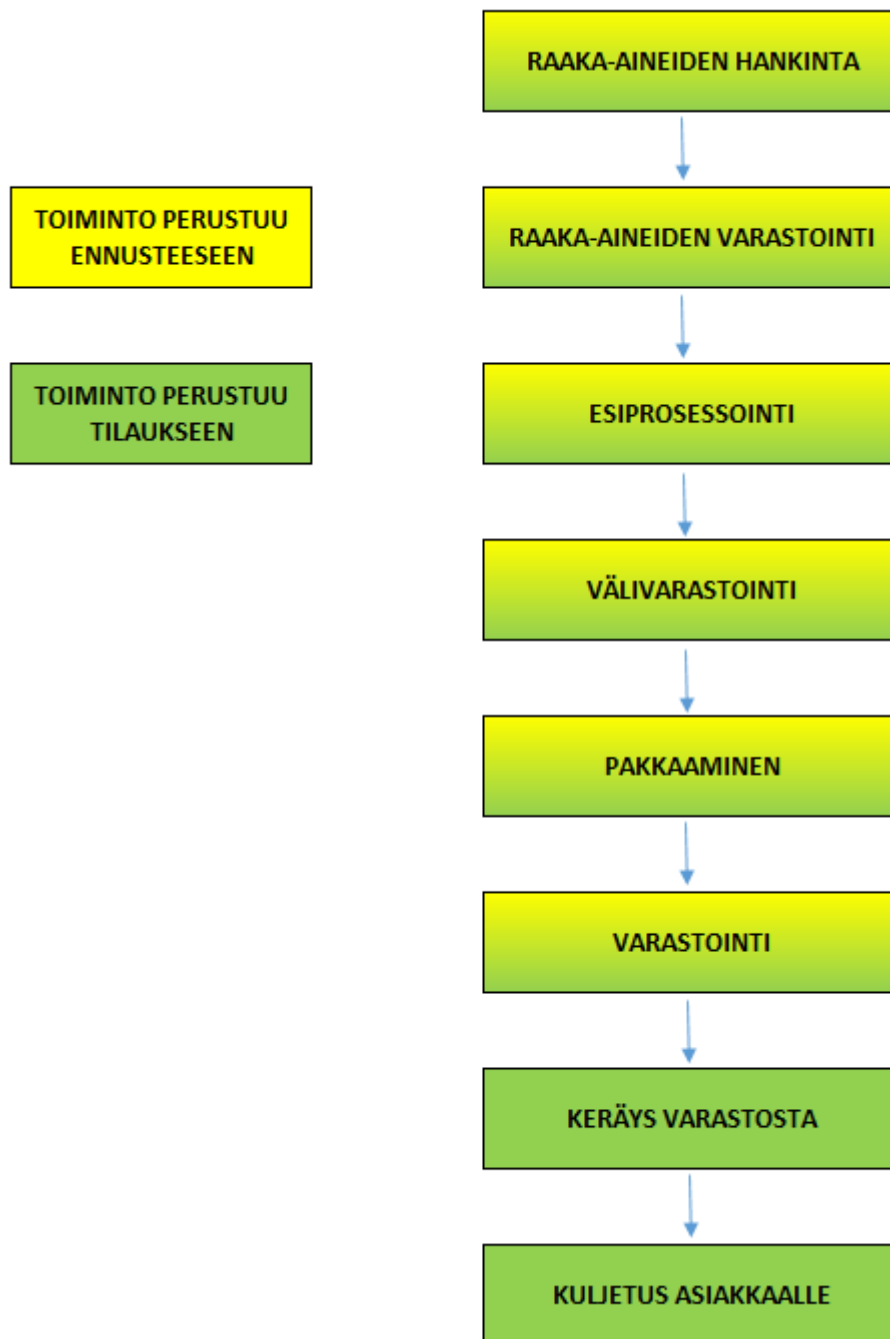
Tämän opinnäytetyön tarkoitus on antaa lukijalle selkeä yleiskuva tuotannonohjauksen työkalujen valmistamisesta sekä toimia yritykselle ohjeena työkalujen tarkempaan tarkasteluun. Työ koostuu lyhyestä tuotannonohjauksen esittelystä, työssä käytettyjen Excel-funktioiden sekä VBA-ohjelmoinnin esittelystä, ohjelmallisesta raportista työkalujen valmistamisesta ja tulososiosta.

2 TAUSTATIETOA

2.1 Tuotannonsuunnittelu

Hyvin toimiva tuotanto on monien tekijöiden summa, ja yksi tärkeimmistä tekijöistä on tuotannonsuunnittelu. Tuotannonsuunnittelu lähtee liikkeelle jo tuotteen kehitysvaiheessa, jolloin on osattava ottaa kantaa kehitettävän tuotteen ominaisuuksiin, kuten esimerkiksi säilyvyyteen ja valmistusprosessiin.

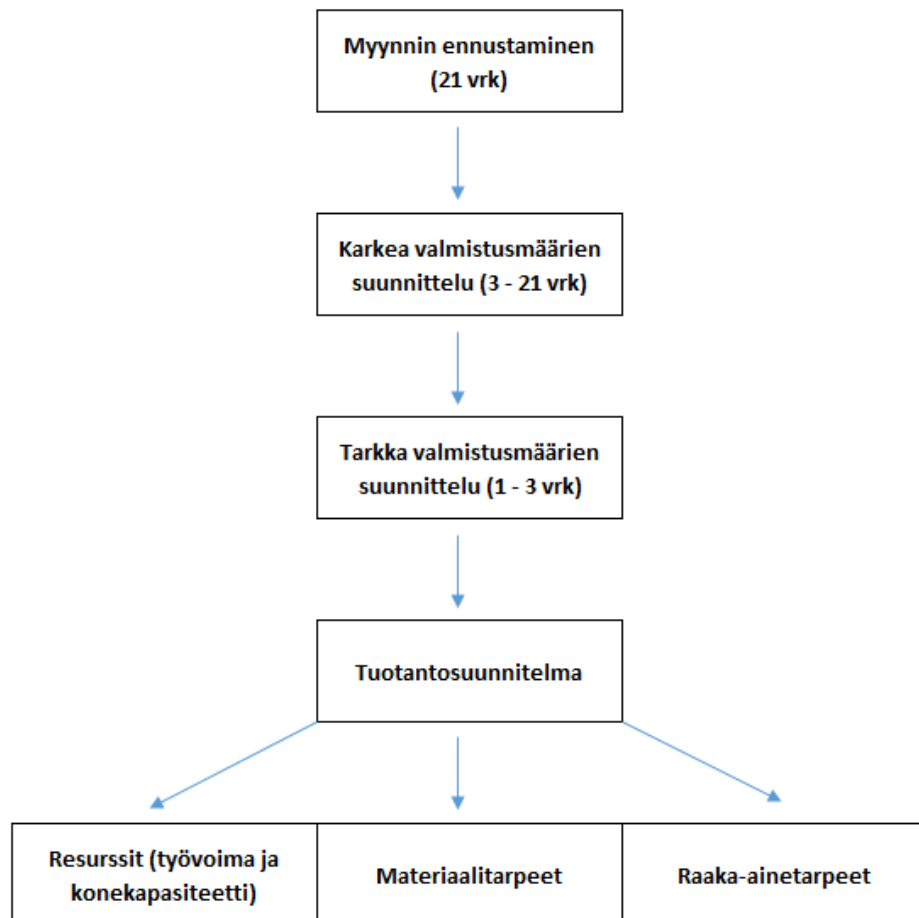
Tilaus-toimitusprosessi voidaan usein yksinkertaistaa tavarantoimittajan, yrityksen ja asiakkaan väliseksi vuorovaikutukseksi (Sakki 2014). Tässä vuorovaikutuksessa siirtyy tavaran ja rahan lisäksi myös tietoa (Sakki 2014). Tilaus-toimitusprosessin käynnistymisen impulssi voi olla joko tilaus tai ennuste riippuen yrityksen tuotteiden ominaisuuksista ja sovitusta toimitusrytmistä. Tämän vuoksi rajaveto ennusteeseen ja tilaukseen perustuvaan toimintoon on hankala tehdä, sillä ei ole olemassa yhtä oikeaa toimintamallia.



Kuvio 1. Tilaus-toimitusprosessi

Kysynnän ennustaminen tehdään jokaiselle tuotteelle erikseen päiväkohtaisesti huomioiden mahdolliset kysyntäpiikit. Jos halutaan tarkentaa ennustetta vielä tästä, on tuotteen päiväkohtainen kysyntä purettava vielä asiakas- tai ketjutasolle. Ennustaminen voidaan tehdä esimerkiksi edellisten viikkojen keskiarvo-laskuna (Sakki

2014). Kysynnän ennustamisen jälkeen voidaan alkaa suunnittelemaan päiväkohtaisia valmistusmääriä tuotteille. Näin saadaan aikaan tuotantosuunnitelma, johon materiaali-, raaka-aine- ja resurssitarpeet pohjautuvat.

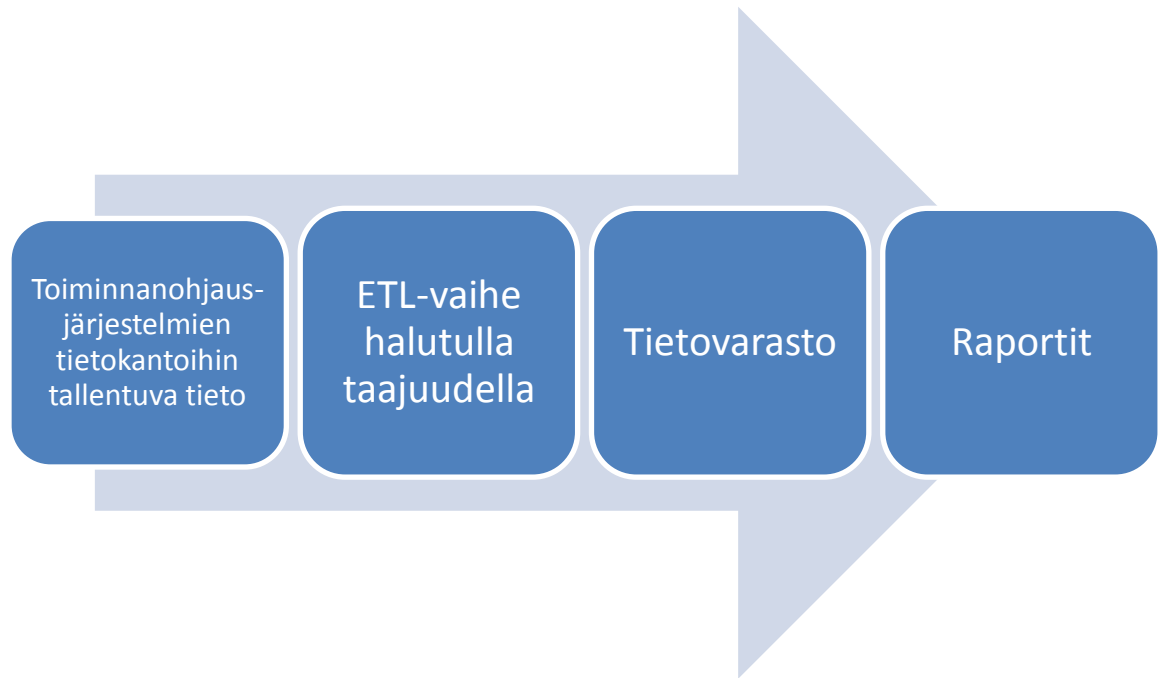


Kuvio 2. Tuotannonsuunnittelun pääkohdat

2.2 Tietovarastot

Toiminnanohjausjärjestelmää käytettäessä syntyy jatkuvasti lisää informaatiota, jonka seuraaminen antaa paitsi työntekijälle myös yritykselle arvokasta lisätietoa. Tämä tieto muutetaan ETL-vaiheessa (Extract – Transform – Load) tietovarastokannan vaatimaan muotoon (Hovi, Hervonen & Koistinen 2009, 14). ETL-vaihe voidaan

toteuttaa paitsi halutulla taajuudella myös reaaliaikaisesti (mts. 29). Tietovarastoraporttien kautta tietovarastokantaan siirretty tieto saadaan työntekijän ulottuville joko valmiissa tai taulukkolaskentaohjelmalla jatkojalostettavassa muodossa (mts. 6-7).



Kuvio 3. Tietovarastoraportin luominen eri vaiheet

2.3 Microsoft Excel –funktiot

Microsoft Excelin hallitseminen antaa tuotannosuunnittelijalle mahdollisuudet käsitellä yhtiön tuottamaa dataa paitsi nopeasti ja luotettavasti, myös toistettavasti. Merkitys korostuu vielä entisestään, jos tuotannosuunnittelun käyttämät työkalut ovat Excel-pohjaisia. Työn toteutuksen isoin osa oli sopivien funktioiden etsiminen ja niiden soveltaminen työkaluissa. Tässä osiossa esitellään suunnittelutyökaluissa käytettyjä funktiota sekä teorian että havainnollistavien esimerkkien kautta.

2.3.1 PHAKU ja VHAKU

PHAKU-funktion avulla voidaan etsiä määrittelyn matriisin ensimmäisestä sarakkeesta hakuarvoa vastaavia arvoja ja palauttaa samalta riviltä halutun sarakkeen

arvon. Funktio muodostetaan neljästä argumentista, jotka ovat hakuarvo, matriisi, sarakkeen indeksinumero ja totuusarvo. Kaava kirjoitetaan tällöin muotoon PHAKU (hakuarvo ; matriisi ; sarakkeen indeksinumero ; totuusarvo). Matriisiin ensimmäinen sarake toimii aina hakuarvon etsintäsarakkeena ja määrittää rivin, jolta palautettava arvo löytyy. Sarakkeen indeksinumero puolestaan kertoo, mistä sarakkeesta palautettava arvo löytyy. Totuusarvon avulla määritetään halutaanko etsiä hakuarvon tarkka vastine vai lähin vastaava. (Microsoft, PHAKU-funktio.)

Kuviossa 4 on esitetty esimerkki PHAKU-funktion käytöstä. Tässä esimerkissä halutaan soluun G1 palauttaa arvo, joka löytyy samalta riviltä kuin hakuarvo B-sarakkeesta ja matriisin kolmannesta sarakkeesta. Sana ”tiistai” löytyy B-sarakkeesta riviltä 2, ja matriisin kolmas sarake on D, jolloin soluun G1 palautuu arvo, joka löytyy solusta D2 eli viikon toinen päivä.

	A	B	C	D	E	F	G
1		maanantai		Viikon ensimmäinen päivä		tiistai	Viikon toinen päivä
2		tiistai		Viikon toinen päivä			
3		keskiviikko		Viikon kolmas päivä			
4		torstai		Viikon neljäs päivä		HAKUARVO	
5		perjantai		Viikon viides päivä			
6		lauantai		Viikon kuudes päivä			
7		sunnuntai		Viikon seitsemäs päivä			
8							PHAKU(F1;B:D;3;0)
9		1	2	3			
10							
11							
12		SARAKKEIDEN INDEKSIINUMEROT					
13		MATRIISI (KELTAINEN ALUE)					
14							

Kuvio 4. PHAKU

PHAKU-funktio on erityisen hyödyllinen, kun hakumatriisin ensimmäinen sarake sisältää jokaista arvoa vain yhden. Jos ensimmäinen sarake sisältää hakuarvoa vastaavia arvoja enemmän kuin yhden, palauttaa funktio pienemmältä riviltä löytyvän arvon.

VHAKU-funktio on hyvin samankaltainen funktio kuin PHAKU. Erotuksena näillä kahdella on se, että VHAKU etsii hakuarvoa matriisin ylimmältä riviltä, jolloin saadaan selville, mistä sarakkeesta palautettava arvo löytyy. Rivinumero määritetään rivin indeksinumeron avulla. (Microsoft, VHAKU-funktio.)

Hyvä esimerkki PHAKU- ja VHAKU-funktioiden käytöstä suunnittelutyökaluissa on tiedon tai ominaisuuden tuotekoodin viereiseen soluun laskentaa tai informaatiota varten.

2.3.2 SUMMA.JOS

SUMMA.JOS-funktion avulla saadaan laskettua yhteen sarakkeen arvoja määriteltujen ehtojen avulla. Syntaksin argumentit ovat hakualue, hakuehto ja summa-alue. Kaava kirjoitetaan tällöin muotoon SUMMA.JOS(hakualue ; hakuehto ; summa-alue. Hakuehtona ei tarvitse välttämättä olla yksi sana tai numero, vaan ehtona voidaan käyttää myös suurempi kuin- ja pienempi kuin-lauseita. (Microsoft, SUMMA.JOS-funktio.)

Kuviossa 5 käsitellään SUMMA.JOS-funktiota, jolla halutaan summata kaikki summa-alueen arvot, jotka löytyvät samalta riviltä kuin hakuarvot hakualueesta. Sana tiistai löytyy B-sarakkeen riveiltä 2 ja 9, jolloin summataan yhteen D-sarakkeen arvot, jotka löytyvät vastaavilta riveiltä. Näin summaksi saadaan 4 (2+2).

	A	B	C	D	E	F	G	
1		maanantai		1		tiistai	4	
2		tiistai		2				
3		keskiviikko		3				
4		torstai		4		HAKUARVO		
5		perjantai		5				
6		lauantai		6				
7		sunnuntai		7				
8		maanantai		1			SUMMA.JOS(B:B;F1;D:D)	
9		tiistai		2				
10		A L U E		S U M M A - A L U E				
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								

Kuvio 5. SUMMA.JOS

SUMMA.JOS-funktio on erittäin hyödyllinen, jos halutaan esimerkiksi summata yrityksen kaikkien tuotteiden eri materiaalien päivä- tai viikkokohtaiset kulutukset.

2.3.3 INDEKSI

INDEKSI-funktion avulla voidaan palauttaa halutusta hakumatriisista rivi- ja sarakenumeron perusteella solun arvo. Funktion argumentit ovat hakumatriisi, rivinumero ja sarakenumero. Kaava kirjoitetaan muotoon INDEKSI(hakumatriisi ; rivinumero ; sarakenumero). (Microsoft, INDEKSI-funktio.)

Kuviossa 6 on havainnollistettu INDEKSI-funktion käyttöä yksinkertaisella esimerkillä. Solussa O3 olevan kaavan avulla etsitään hakumatriisista L3:M5 arvoa, joka löytyy matriisin ensimmäisen sarakkeen ensimmäiseltä riviltä. Näin ollen kaava palauttaa arvon A1.

	K	L	M	N	O	P
1						
2						
3		A1	B1		A1	
4		A2	B2			
5		A3	B3			
6					INDEKSI(L3:M5;1;1)	
7						
8						
9	Matriisi keltaisella pohjalla					

Kuvio 6. INDEKSI

2.3.4 VASTINE

VASTINE-funktion avulla voidaan etsiä määritellystä hakumatriisista hakuarvoa vastaavan arvon suhteellinen sijainti matriisissa. Funktion argumentit ovat hakuarvo, hakumatriisi ja vastinelaji. Kaava kirjoitetaan muotoon VASTINE(hakuarvo ; hakumatriisi ; vastinelaji). Vastinelajin avulla määritellään halutaanko etsiä hakuarvon tarkkaa vastinetta (0 tai puuttuu), lähintä pienempää tai yhtä suurta vastinetta (-1) vai lähintä suurempaa tai yhtä suurta vastinetta (1). (Microsoft, VASTINE-funktio.)

Kuvion 7 solun E5 kaavassa etsitään solun A5 arvon suhteellista sijaintia hakumatriisista 1:1. Arvo F löytyy hakumatriisin kuudennesta sarakkeesta, joten funktio palauttaa arvon 6. Vastinelaji ei tässä tapauksessa voi olla mikään muu kuin 0, sillä vastinelajit 1 ja -1 toimivat halutulla tavalla vain numeroita käytettäessä.

	A	B	C	D	E	F	G	H	I	J	K	L
1	A	B	C	D	E	F	G	HAKUMATRIISI KELTAISELLA POHJALLA				
2												
3												
4												
5	F				6							
6												
7												
8												
9	HAKUARVO				VASTINE(A5;1;1;0)							

Kuvio 7. VASTINE

VASTINE-funktio on erinomainen kaava käytettäväksi toisten funktioiden kanssa. Esimerkiksi korvaamalla INDEKSI-funktion rivinumero argumentin VASTINE-funktiolla, saadaan aikaan hyvin pitkälti PHAKU-funktion kaltainen kaava. Tällä yhdistelmällä pystyy hakemaan arvoja jopa PHAKU-funktiota joustavammin, sillä lukuja voidaan hakea myös vasemmalle päin edeten, kuten kuvion 8 esimerkissä on esitetty.

	A	B	C	D	E	F	G	H		
1		Viikon ensimmäinen päivä		maanantai		tiistai	Viikon toinen päivä			
2		Viikon toinen päivä		tiistai						
3		Viikon kolmas päivä		keskiviikko						
4		Viikon neljäs päivä		torstai						
5		Viikon viides päivä		perjantai						
6		Viikon kuudes päivä		lauantai						
7		Viikon seitsemäs päivä		sunnuntai						
8		<div style="text-align: center;"> <p>1 2 3</p> <p>↑ ↑ ↑</p> <p>SARAKKEIDEN NUMEROT</p> <p>MATRIISI (KELTAINEN ALUE)</p> </div>								
9										
10										
11										
12										
13										
14										

Kuvio 8. PHAKU-funktion korvaus INDEKSI- ja VASTINE-funktioiden avulla

2.3.5 Matriisikaavat

Matriisikaavojen avulla voidaan suorittaa hankalia laskutoimituksia, joiden ratkaisu tavanomaisia kaavoja käyttämällä vaatisi useita eri välivaiheita, sillä matriisikaava osaa suorittaa useita eri laskutoimituksia samaan aikaan. Matriisikaavan kirjoittami-

nen jälkeen on muistettava aina painaa CTRL+VAIHTO+ENTER, sillä muutoin kirjoitettu kaava jää tavalliseksi kaavaksi eikä toimi halutulla tavalla. (Microsoft, Matriisikaavoihin liittyviä ohjeita ja esimerkkejä.)

Kuvion 9 sarakkeessa D näkyvät kaavat, joilla sarakkeiden A-C osoittamat laskutoimitukset voidaan suorittaa tavanomaisia kaavoja käyttämällä. Sarakkeessa E on puolestaan esitetty samat laskutoimitukset matriisikaavoja käyttäen.

	A	B	C	D	E
1				Tavallinen kaava	Matriiskaava
2	10	x	20	=A2*C2	=A2:A12*C2:C12
3	11	x	21	=A3*C3	=A2:A12*C2:C12
4	12	x	22	=A4*C4	=A2:A12*C2:C12
5	13	x	23	=A5*C5	=A2:A12*C2:C12
6	14	x	24	=A6*C6	=A2:A12*C2:C12
7	15	x	25	=A7*C7	=A2:A12*C2:C12
8	16	x	26	=A8*C8	=A2:A12*C2:C12
9	17	x	27	=A9*C9	=A2:A12*C2:C12
10	18	x	28	=A10*C10	=A2:A12*C2:C12
11	19	x	29	=A11*C11	=A2:A12*C2:C12
12	20	x	30	=A12*C12	=A2:A12*C2:C12

Kuvio 9. Esimerkki tavallisen kaavan korvaamisesta matriisikaavalla

2.4 Microsoft Excel - VBA

VBA-ohjelmointia voidaan suorittaa joko nauhoittamalla toimenpidesarja tai kirjoittamalla itse ohjelmointikoodia. Nauhoittamalla kirjoitettujen ohjelmointikoodien rajoitteet tulevat vastaan suhteellisen nopeasti, kun taas itse kirjoitettaessa nuo rajat ovat huomattavasti laajemmat. Ohjelmointiin tutustuvat on syytä muistaa, että ohjelmointi on itseasiassa enemmän ongelmanratkaisua kuin ohjelmoinnin opettelua. (Taanila 2013, 1.) Hyvä tapa opiskella ohjelmointikieltä on seurata nauhoitettuja koodinpalasia (mts. 6).

2.4.1 Esimerkiohjelmointi

Seuraavassa esimerkissä esitellään ohjelmointikoodi, jonka avulla voidaan toteuttaa haluttuja toimenpiteitä määritettyjen ehtojen täytyessä. Koodi on erityisen hyödyllinen korvaamaan muutoin käsityönä tehtävää virhealtista ja toistuvaa työtä.

Kuviossa 10 on esitetty kahden tuotteen tuotantosuunnitelmat, jotka halutaan siirtää seuraavan päivän aamuna vastaamaan eilen tehtyjä suunnitelmia. Kuviossa 11 on sama taulukko koodin suorittamisen jälkeen.

	A	B	C	D	E	F	G
1							
2	TUOTEKODI	TUOTENIMI		20.10.2014	21.10.2014	22.10.2014	23.10.2014
3	1	A	VARASTOSALDO (KG)	5	3	1	1
4			ENNUSTE (KG)	5	6	5	6
5			SUUNNITELMA (KG)	3	4	5	
6							
7							
8	TUOTEKODI	TUOTENIMI		20.10.2014	21.10.2014	22.10.2014	23.10.2014
9	2	B	VARASTOSALDO (KG)	11	7	3	1
10			ENNUSTE (KG)	6	7	6	7
11			SUUNNITELMA (KG)	2	3	4	
12							
13							
14							
15							
16							
17							
18							
19							
20							

Kuvio 10. Tuotantosuunnitelma edellisenä iltana ennen makron käyttöä

	A	B	C	D	E	F	G
1							
2	TUOTEKODI	TUOTENIMI		21.10.2014	22.10.2014	23.10.2014	24.10.2014
3	1	A	VARASTOSALDO (KG)	3	1	1	-5
4			ENNUSTE (KG)	6	5	6	0
5			SUUNNITELMA (KG)	4	5		
6							
7							
8	TUOTEKODI	TUOTENIMI		21.10.2014	22.10.2014	23.10.2014	24.10.2014
9	2	B	VARASTOSALDO (KG)	7	3	1	-6
10			ENNUSTE (KG)	7	6	7	0
11			SUUNNITELMA (KG)	3	4		
12							
13							
14							
15							
16							
17							
18							
19							
20							

Kuvio 11. Tuotantosuunnitelma seuraavana aamuna makron käyttämisen jälkeen

Kuviosta 12 löytyy valmis ohjelmointikoodi, joka suorittaa esimerkissä mainitut tehtävät. Alla on selitetty eri osioiden suorittamat tehtävät.

```
Application.ScreenUpdating = False
```

```
Application.Calculation = xlCalculationManual
```

- Ruudun päivityksen ja laskennan poistaminen käytöstä laskennan nopeuttamiseksi

```
Windows("Mallityökalu.xlsm").Activate
```

```
Sheets("TUOTANTOSUUNNITELMA").Select
```

```
Range("C1").Select
```

- Valitsee auki olevista työkirjoista Mallityökalu.xlsm:n TUOTANTOSUUNNITELMA-välilehden C1-solun

```
Do
```

```
  If Range("C" & ActiveCell.Row) = "SUUNNITELMA (KG)" Then
```

```
    ActiveCell.Offset(0, 2).Resize(, 2).Select
```

```
    Selection.Copy
```

```
    Range("D" & ActiveCell.Row).Select
```

```
    ActiveSheet.PasteSpecial Format:=3, Link:=1, DisplayAsIcon:=False, _
```

```
    IconFileName:=False
```

```
    ActiveCell.Offset(0, 2).Select
```

```
    Selection.ClearContents
```

```
    Range("C" & ActiveCell.Row).Select
```

```
  End If
```

```
  ActiveCell.Offset(1, 0).Select
```

```
Loop Until (ActiveCell.Address = "$C$500")
```

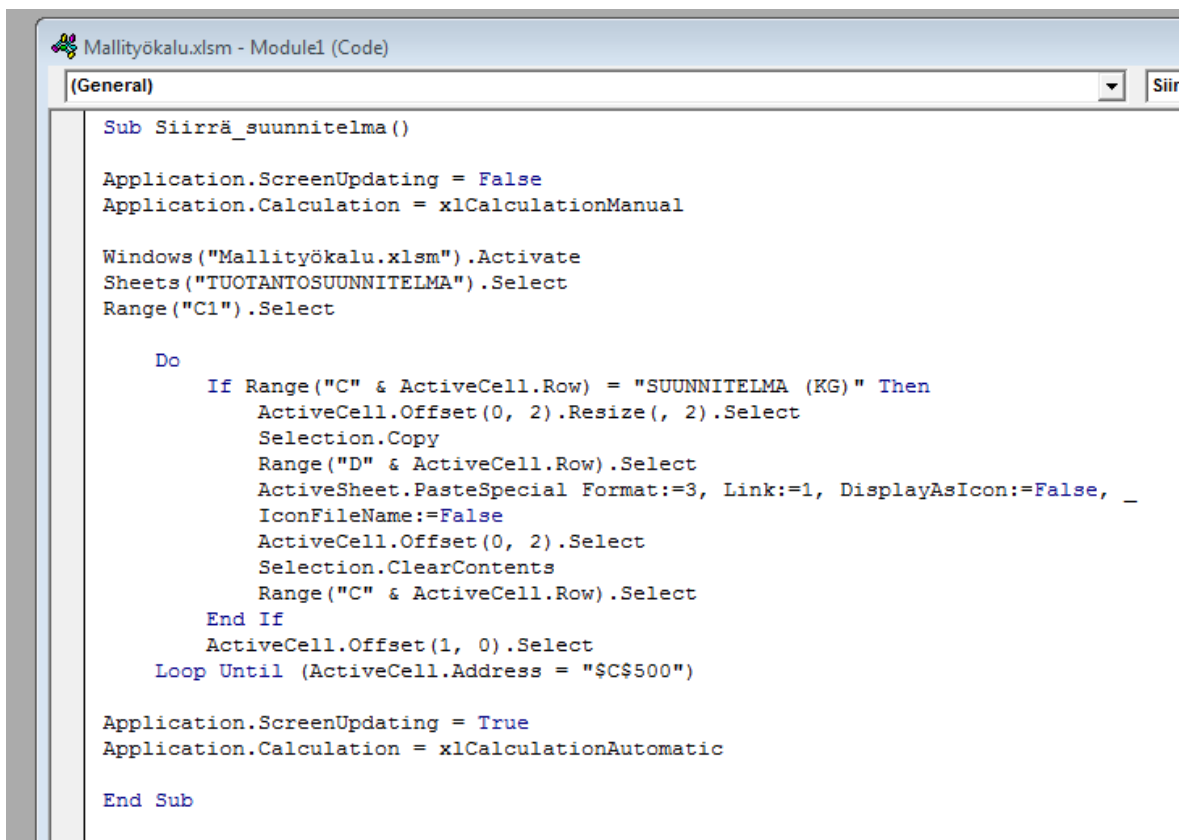
- Koodin ydinosa, joka käy läpi alueen C1:C500 ja etsii tekstiä SUUNNITELMA (KG). Jos teksti löytyy, siirrytään solusta 2 solua oikealle

ja suurennetaan valintaa yhden solun verran vasemmalle. Tämän jälkeen palataan saman rivin D-sarakkeeseen, johon valittu alue liitetään. Seuraavaksi siirrytään vielä tästä 2 solua vasemmalle ja tyhjennetään valitun solun sisältö. Lopuksi palataan saman rivin C-sarakkeeseen. IF-lause päättyy tähän. Tämän jälkeen Siirrytään C-sarakkeessa seuraavalle riville ja jatketaan seuraavaa SUUNNITELMA (KG)-tekstin etsintään

Application.ScreenUpdating = True

Application.Calculation = xlCalculationAutomatic

- Ruudun päivityksen ja laskennan palauttaminen käyttöön

The image shows a screenshot of the Microsoft Excel VBA code editor. The title bar at the top reads "Mallityökalu.xlsm - Module1 (Code)". Below the title bar, there is a tab labeled "(General)" and a "Siir" button. The main area contains the following VBA code:

```
Sub Siirrä_suunnitelma()  
  
Application.ScreenUpdating = False  
Application.Calculation = xlCalculationManual  
  
Windows("Mallityökalu.xlsm").Activate  
Sheets("TUOTANTOSUUNNITELMA").Select  
Range("C1").Select  
  
    Do  
        If Range("C" & ActiveCell.Row) = "SUUNNITELMA (KG)" Then  
            ActiveCell.Offset(0, 2).Resize(, 2).Select  
            Selection.Copy  
            Range("D" & ActiveCell.Row).Select  
            ActiveSheet.PasteSpecial Format:=3, Link:=1, DisplayAsIcon:=False, _  
            IconFileName:=False  
            ActiveCell.Offset(0, 2).Select  
            Selection.ClearContents  
            Range("C" & ActiveCell.Row).Select  
        End If  
        ActiveCell.Offset(1, 0).Select  
    Loop Until (ActiveCell.Address = "$C$500")  
  
Application.ScreenUpdating = True  
Application.Calculation = xlCalculationAutomatic  
  
End Sub
```

Kuvio 12. Esimerkki valmiista ohjelmointikoodista

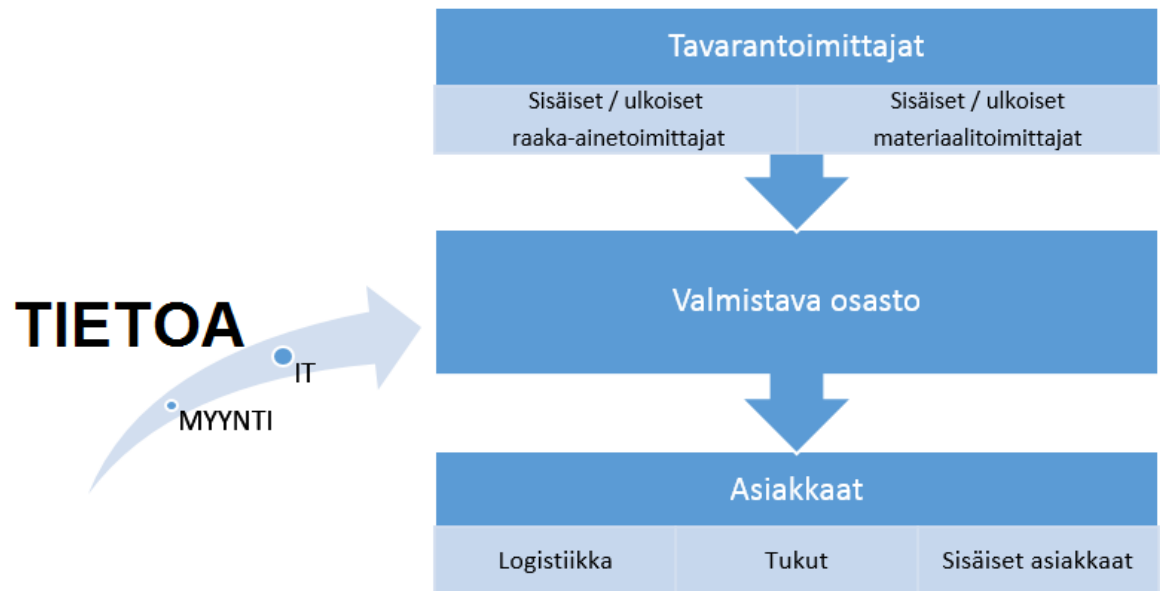
3 TYÖN TOTEUTUS

3.1 Lähtökohdat

Microsoft Excel -pohjaisten suunnittelutyökalujen lähtökohtana on usein toiminnanohjausjärjestelmän hienosuunnittelun ja muokattavuuden tarve. Esimerkiksi reaaliaikaisen varastosaldon käyttö tilausohjautuvilla lyhyen varastointiajan omaavilla tuotteilla voi olla hankalaa ja joissain tapauksissa jopa mahdotonta. Lisäksi aloilla, joilla on laaja ja monimuotoinen tuotevalikoima, ulkoistettu toiminnanohjausjärjestelmä ei aina tarjoa sitä muokattavuutta, jota tuotannonohjausjärjestelmä tarvitsisi.

3.2 Toimintaan perehtyminen

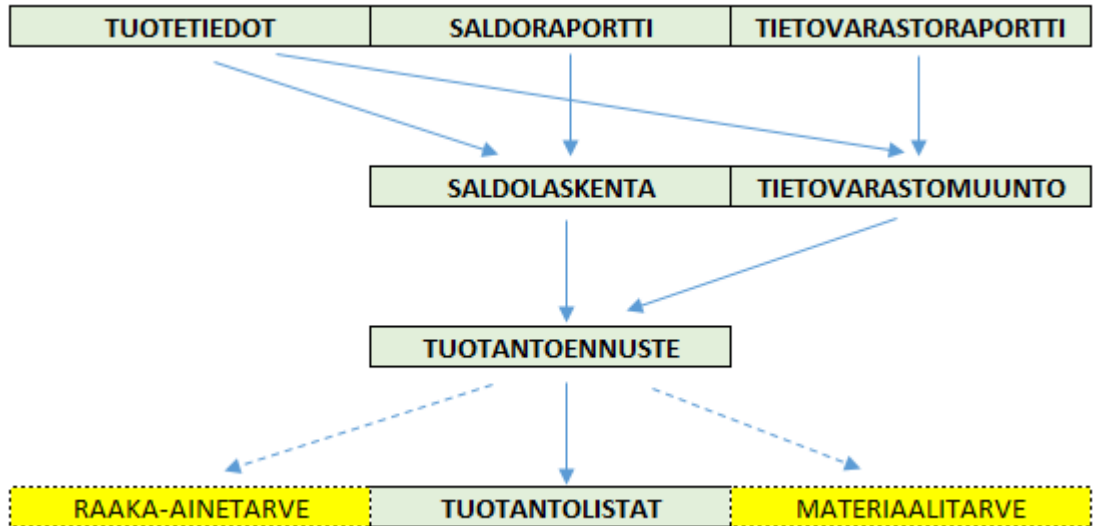
Ennen kuin uusien suunnittelutyökalujen suunnittelu voidaan aloittaa, on suunnittelijan syytä perehtyä läpikotaisin valmistavan osaston toimintaan. Etenkin prosesseista, käytettävistä raaka-aineista ja lopputuotteista on oltava erittäin hyvä käsitys. Lisäksi on tunnettava tavarantoimittajien ja asiakkaiden luomat vaatimukset ja rajoitukset. Perehdyttämiseen saattaa kulua pitkiäkin aikoja, mutta ilman sitä suunnittelutyökalujen luontia on erittäin hankala edes aloittaa.



Kuvio 13. Asiakassuhteet

3.3 Työn suunnittelu

Valmistettaessa Microsoft Excel -pohjaisia työkaluja, on työjärjestys ja tietohierarkia suunniteltava hyvin, sillä tällä vähennetään puutteellisen hierarkian riskiä olennaisesti. Puutteellinen hierarkia saattaa hankalimmissa tapauksissa tarkoittaa työn aloittamista jopa kokonaan uudelleen. Hyvä perusperiaate on, että tietoa ei voi olla liikaa hierarkian eri tasoilla. On kuitenkin huomioitava, että tiedon määrän lisääntyessä kokonaisuuden hahmottaminen voi vaikeutua. Siksi työkalujen valmistuksessa onkin alusta alkaen pyrittävä mahdollisimman selkeään ulosantiin.



Kuvio 14. Hierarkia

3.4 Suunnittelutyökalujen valmistus

Tässä osiossa käydään läpi suunnittelutyökalujen valmistuksen eri vaiheet pääpiirteittäin. Kappaleen tarkoituksena on antaa hyvä yleiskuva suunnittelutyökalujen toimintaperiaatteesta, joita hyväksikäyttämällä saadaan rakennettua yrityksen tarpeisiin räätälöity tuotannonohjausjärjestelmä.

3.4.1 Tuotetiedot

Ensimmäinen työvaihe on koota tuotteen olennaiset ominaisuudet ja tiedot. Koonti on hyvä aloittaa tuotekoodien ja näitä vastaavien nimien syöttämisellä. Tämän jälkeen tuotteen ominaisuudet kerätään samalle riville ominaisuudet selkeästi otsikoiden.

TUOTEKOODI	TUOTENIMI	OMINAISUUS 1	OMINAISUUS 2	OMINAISUUS 3	OMINAISUUS 4
1	A				
2	B				
3	C				
4	D				
5	E				
6	F				
7	G				
8	H				
9	I				
10	J				

Kuvio 15. Tuotetiedot

Elintarvikealan yritykselle olennaisia ominaisuuksia ovat muun muassa:

- Säilyvyysaika
- Asiakkaan hyllyaika
- Valmistusaineet ja niiden osuus tuotteesta
- Hävikit
- Rasian sisältämä kappalemäärä
- Rasiapaino
- Myyntierä-, laatikko- ja lavakoko
- Pakkausmateriaalit ja -koodit
- Tuotantolinja

3.4.2 Myynti- ja ennustetietojen haku tietovarastoraporteista

Käytettävän tietovarastoraportin on sisällettävä vähintään suunnittelijan ohjaamien tuotteiden myynti- ja ennustetiedot päiväkohtaisesti. Lisäksi jossain tapauksissa on tuotteen päiväkohtainen myynti purettava vielä asiakastasolle, jolloin pystytään ohjaamaan tuotannon valmistamia tuotteita useille eri asiakkaille suoraan tuotanto-osastolta käsin ilman erillisen logistiikkakeskuksen apua.

Kuviossa 16 on esitetty havainnollistava esimerkki tietovarastoraportin tietoalueesta, jotka on muutettu kuviossa 17 näkyvien kaavojen avulla muotoon, joita on helppo käyttää jatkossa hyväksi. Erityistä huomiota on syytä kiinnittää kuvion 17 A-

sarakkeen ketjutuskaavoihin, joiden lopputuleman on oltava muodossa "TUOTEKOODI_PVM", jolloin tätä saraketta voidaan käyttää hakusarakkeena haettaessa tuotteen päiväkohtaista kokonaismyyntiä tai –ennustetta tuotantoennusteeseen. Jos halutaan vielä jakaa päiväkohtainen myynti asiakastasolle, on tätä varten syytä tehdä omalle sarakkeelle ketjutuskaavat, joiden lopputulema on muotoa "TUOTEKOODI_PVM_ASIKASNUMERO". Ketjutukset tehdään omille sarakkeilleen, koska ketjutuksella "TUOTEKOODI_PVM_ASIKASNUMERO" ei voida hakea tuotteen päiväkohtaista kokonaismyyntiä ja ketjutuksella "TUOTEKOODI_PVM" puolestaan ei saada purettua tuotteiden kokonaismyyntiä asiakastasolle.

H	I	J	K	L
TUOTEKOODI	TUOTENIMI	PVM	ENNUSTE (kg)	TILAUS (kg)
1	A	20141018	1	1
1	A	20141019	2	2
1	A	20141020	1	1
1	A	20141021	2	2
2	B	20141018	1	1
2	B	20141019	2	2
2	B	20141020	1	1
2	B	20141021	2	2
3	C	20141018	1	1
3	C	20141019	2	2
3	C	20141020	1	1
3	C	20141021	2	2
4	D	20141018	1	1
4	D	20141019	2	2
4	D	20141020	1	1
4	D	20141021	2	2

TIEVARASTORAPORTTIALUE

Kuvio 16. Tietovarastoraportin esimerkinäkymä

	A	B	C	D	E	F
1	KETJUTUS	TUOTEKOODI	TUOTENIMI	PVM	ENNUSTE (kg)	TILAUS (kg)
2	=KETJUTA(\$B2;" ";\$D2)	=H2*1	=I2	=PÄIVÄYS(VASEN(\$J2;4);POIMI.TEKSTI(\$J2;5;2);OIKEA(J2;2))	=\$K2*1	=\$L2*1
3	=KETJUTA(\$B3;" ";\$D3)	=H3*1	=I3	=PÄIVÄYS(VASEN(\$J3;4);POIMI.TEKSTI(\$J3;5;2);OIKEA(J3;2))	=\$K3*1	=\$L3*1
4	=KETJUTA(\$B4;" ";\$D4)	=H4*1	=I4	=PÄIVÄYS(VASEN(\$J4;4);POIMI.TEKSTI(\$J4;5;2);OIKEA(J4;2))	=\$K4*1	=\$L4*1
5	=KETJUTA(\$B5;" ";\$D5)	=H5*1	=I5	=PÄIVÄYS(VASEN(\$J5;4);POIMI.TEKSTI(\$J5;5;2);OIKEA(J5;2))	=\$K5*1	=\$L5*1
6	=KETJUTA(\$B6;" ";\$D6)	=H6*1	=I6	=PÄIVÄYS(VASEN(\$J6;4);POIMI.TEKSTI(\$J6;5;2);OIKEA(J6;2))	=\$K6*1	=\$L6*1
7	=KETJUTA(\$B7;" ";\$D7)	=H7*1	=I7	=PÄIVÄYS(VASEN(\$J7;4);POIMI.TEKSTI(\$J7;5;2);OIKEA(J7;2))	=\$K7*1	=\$L7*1
8	=KETJUTA(\$B8;" ";\$D8)	=H8*1	=I8	=PÄIVÄYS(VASEN(\$J8;4);POIMI.TEKSTI(\$J8;5;2);OIKEA(J8;2))	=\$K8*1	=\$L8*1
9	=KETJUTA(\$B9;" ";\$D9)	=H9*1	=I9	=PÄIVÄYS(VASEN(\$J9;4);POIMI.TEKSTI(\$J9;5;2);OIKEA(J9;2))	=\$K9*1	=\$L9*1
10	=KETJUTA(\$B10;" ";\$D10)	=H10*1	=I10	=PÄIVÄYS(VASEN(\$J10;4);POIMI.TEKSTI(\$J10;5;2);OIKEA(J10;2))	=\$K10*1	=\$L10*1
11	=KETJUTA(\$B11;" ";\$D11)	=H11*1	=I11	=PÄIVÄYS(VASEN(\$J11;4);POIMI.TEKSTI(\$J11;5;2);OIKEA(J11;2))	=\$K11*1	=\$L11*1
12	=KETJUTA(\$B12;" ";\$D12)	=H12*1	=I12	=PÄIVÄYS(VASEN(\$J12;4);POIMI.TEKSTI(\$J12;5;2);OIKEA(J12;2))	=\$K12*1	=\$L12*1
13	=KETJUTA(\$B13;" ";\$D13)	=H13*1	=I13	=PÄIVÄYS(VASEN(\$J13;4);POIMI.TEKSTI(\$J13;5;2);OIKEA(J13;2))	=\$K13*1	=\$L13*1
14	=KETJUTA(\$B14;" ";\$D14)	=H14*1	=I14	=PÄIVÄYS(VASEN(\$J14;4);POIMI.TEKSTI(\$J14;5;2);OIKEA(J14;2))	=\$K14*1	=\$L14*1
15	=KETJUTA(\$B15;" ";\$D15)	=H15*1	=I15	=PÄIVÄYS(VASEN(\$J15;4);POIMI.TEKSTI(\$J15;5;2);OIKEA(J15;2))	=\$K15*1	=\$L15*1
16	=KETJUTA(\$B16;" ";\$D16)	=H16*1	=I16	=PÄIVÄYS(VASEN(\$J16;4);POIMI.TEKSTI(\$J16;5;2);OIKEA(J16;2))	=\$K16*1	=\$L16*1
17	=KETJUTA(\$B17;" ";\$D17)	=H17*1	=I17	=PÄIVÄYS(VASEN(\$J17;4);POIMI.TEKSTI(\$J17;5;2);OIKEA(J17;2))	=\$K17*1	=\$L17*1
18						
19	TIETOVARASTORAPORTIN MUUNTOALUE					

Kuvio 17. Tietovarastoraportin esimerkkimuunto

3.4.3 Varastosaldon laskeminen saldoraportista

Elintarvikkeiden varastosaldon laskemisessa on otettava huomioon, että kaikki varastossa olevat tuotteet eivät välttämättä enää täytä asiakkaalle luvattuja ehtoja. Näitä luvattuja ehtoja ovat muun muassa asiakkaalle taattu hyllyaika ja pakkauksen fyysinen kunto. Näin ollen varaston absoluuttinen arvo ei aina ole yhtä suuri kuin myytävissä oleva arvo. Lisäksi on otettava huomioon, että osa varaston sisältämistä tuotteista on jo varattu asiakkaalle, jotka on myös otettava huomioon.

Alla on esitetty yksi tapa sellaisen varastosaldon laskemiseen, joka sisältää asiakkaille varattuja tuotteita. Tässä tapauksessa asiakkaille varattujen tuotteiden määrä on yhtä suuri kuin seuraavan toimituspäivän tilauskanta. Tällöin varaston absoluuttisesta arvosta on vähennettävä tämä tilauskanta ja liian lyhyen hyllyajan omaavat tuotteet, jolloin saadaan selville myytävissä oleva saldo.

+	VARASTON ABSOLUUTTINEN ARVO	A
-	AIKAISEMPIIN TILAUKSIIN VARATUT TUOTTEET	B
-	LIIAN LYHYEN HYLLEYAJAN OMAAVAT TUOTTEET	C
SALDO =	A - (B+C)	X

Kuvio 18. Saldolaskenta

Tämän kaltaista saldolaskentaa käytettäessä saadaan päivän vähimmäisvalmistus laskettua seuraavasti:

+	SALDO	X
-	UUDET TILAUKSET	Y
-	VIRHEELLISET PAKKAUKSET	Z
VALMISTUS =	X - (Y+Z)	

Kuvio 19. Vähimmäisvalmistuksen laskenta

Käytännössä kuitenkin virheellisten pakkausten määrää on mahdotonta tietää tarkasti, joten useimmiten muuttuja Z (virheelliset pakkaukset) jätetään usein kokonaan huomiomatta tässä vaiheessa ja virheelliset pakkaukset otetaan huomioon huolehtimalla siitä, että muuttuja X (saldo) ei koskaan ole 0, vaan seuraavien myyntipäivien ennusteeseen suhteutettu positiivinen luku.

3.4.4 Tuotantosuunnitelma

Tuotantosuunnitelman avulla kootaan tuotteen tietovarastoraportilta haetut ennustetiedot (Kuvion 20 alue D4:G4) sekä varaston lähtötilanne (Kuvion 19 solu D3) yhteen. Tämän jälkeen voidaan alueen D5:F5 solujen avulla suunnitella seuraavan päivän lähtötilanne eli varastosaldo.

Suunnitteluajavälin määrittämiseen vaikuttavat lukuisat eri tekijät, joista etenkin materiaalien ja raaka-aineiden saatavuus ovat keskeisessä roolissa. Omakohtainen

kokemukseni sopivasta suunnitteluajavälistä liha-alan yrityksessä on vähintään 3 viikkoa tulevaisuuteen johtuen raaka-aineiden saatavuuden rajallisuudesta ja työ- määrän tasauksen tarpeesta.

	A	B	C	D	E	F	G
1							
2	TUOTEKOODI	TUOTENIMI		18.10.2014	19.10.2014	20.10.2014	21.10.2014
3	1	A	VARASTOSALDO (KG)	2	1	1	0
4			ENNUSTE (KG)	1	2	1	2
5			SUUNNITELMA (KG)	0	2	0	
6							
7							
8							
9					$D3+D5-D4$	$E3+E5-E4$	$F3+F5-F4$

Kuvio 20. Tuotantosunnitelma

3.4.5 Tuotantolistat

Tuotantolistoille kootaan tuotteet sellaiseen muotoon, että kaikki informaatio on hel- posti työntekijän luettavissa. Lisäksi on otettava huomioon, että jos tuotanto alottaa tuotteen valmistamisen ennusteeseen pohjautuen, voidaan ennustetta joutua kor- jaamaan tuotannon aikana, jos ennuste on ollut pielessä.

	A	B	C	D	E
1	PÄIVÄN VALMISTUS	20.10.2014			
2					
3	LINJA 1				
4					
5	TUOTEKOODI	TUOTENIMI	ENNUSTEMÄÄRÄ (KG)	LISÄTILAUSMÄÄRÄ (KG)	TEHTY (KG)
6	1	A	1	1	1+1

Kuvio 21. Tuotantolista, jossa ennuste korjataan tarvittaessa lisätilauksella

4 TYÖN TARKASTELU

Ohjaustyökalujen lopputulos oli mielestäni onnistunut, sillä laajentuneen aikaikkunan ansiosta tulevaisuuden raaka-aine- ja materiaalisuunnittelu helpottui merkittävästi. Kehittynyt raaka-aine- ja materiaalisuunnittelu puolestaan tarkoittavat usein paitsi varastojen ja hävikkien pienentymistä myös tuotannon tehostumista. Isoin haaste työssä oli työkalujen pitäminen riittävän yksinkertaisina ja käyttäjäystävällisinä. Yksi Microsoft Excel -pohjaisten suunnittelutyökalujen haasteista onkin ylläpidon vaatima työmäärä ja sen mukanaan tuoma epätarkkuus, sillä jokainen pienikin muutos voi tuoda mukanaan ei-toivottuja tuloksia. Tähän saattaisi olla ratkaisuna se, että ennen tuotteen siirtymistä tuotantoon toiminnanohjausjärjestelmiin syötettäisiin kaikki ne tiedot, joita tuotannonohjaus tarvitsee, jolloin näiden pohjalta voitaisiin mahdollisesti luoda tuotetietoraportti, joka korvaisi ainakin osittain suunnittelijan käsin ylläpitämän tuotetieto-tiedoston. Tämän muutoksen isoin hyöty olisi mielestäni se, että tällöin ajan tasalla olevat tiedot olisivat koko henkilökunnan käytössä.

LÄHTEET

- Hovi, A., Hervonen, H. & Koistinen H. 2009. Tietovarastot ja business intelligence. 1. painos. Porvoo: WS Bookwell.
- Karkea tuotannosuunnittelu. [Verkkójulkaisu]. [Viitattu 24.3.2015]. Saatavana: https://noppa.aalto.fi/noppa/kurssi/a35a00310/luennot/A35A00310_luento_14_-_kalvot.pdf
- Microsoft. Ei päiväystä. INDEKSI-funktio. [Verkkosivu]. [Viitattu 10.2.2015]. Saatavana: <https://support.office.com/fi-FI/Article/INDEKSIfunktio-a5dcf0dd-996d-40a4-a822-b56b061328bd>
- Microsoft. Ei päiväystä. Matriisikaavoihin liittyviä ohjeita ja esimerkkejä. [Verkkosivu]. [Viitattu 13.2.2015]. Saatavana: <https://support.office.microsoft.com/fi-fi/article/Matriisikaavoihin-liittyvi%C3%A4-ohjeita-ja-esimerkkej%C3%A4-3be0c791-3f89-4644-a062-8e6e9ecee523?CTT=3&CorrelationId=db4778a8-f63e-42aa-852f-ca61a2c9e33f&ui=fi-FI&rs=fi-FI&ad=FI>
- Microsoft. Ei päiväystä. PHAKU-funktio. [Verkkosivu]. [Viitattu 10.2.2015]. Saatavana: <https://support.office.com/fi-FI/Article/PHAKU-0bbc8083-26fe-4963-8ab8-93a18ad188a1>
- Microsoft. Ei päiväystä. SUMMA.JOS-funktio. [Verkkosivu]. [Viitattu 10.2.2015]. Saatavana: <https://support.office.com/fi-fi/article/SUMMAJOS-169b8c99-c05c-4483-a712-1697a653039b>
- Microsoft. Ei päiväystä. VASTINE-funktio. [Verkkosivu]. [Viitattu 10.2.2015]. Saatavana: <https://support.office.com/fi-FI/Article/VASTINE-e8dff45-c762-47d6-bf89-533f4a37673a>
- Microsoft. Ei päiväystä. VHAKU-funktio. [Verkkosivu]. [Viitattu 10.2.2015]. Saatavana: <https://support.office.microsoft.com/fi-fi/article/VHAKU-VHAKU-funktio-a3034eec-b719-4ba3-bb65-e1ad662ed95f?CorrelationId=aef443ea-f520-4421-8371-fb8452dff84&ui=fi-FI&rs=fi-FI&ad=FI>
- Sakki, J. 2014. Tilaus-toimitusketjun hallinta – Digitalisoitumisen haasteet. [Verkkokirja]. Vantaa: Jouni Sakki Oy. [Viitattu 11.5.2015]. Saatavana Ellibs – e-kirjakokoelmasta. Vaatii käyttöoikeuden

Taanila, A. 27.9.2013. EXCEL VBA-OHJELMOINTI. [Verkkajulkaisu].
[Viitattu 24.3.2015]. Saatavana:
<http://myy.haaga-helia.fi/~taaak/vba/vba.pdf>