

ODOO ERP -JÄRJESTELMÄN RÄÄTÄLÖINTI



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutusohjelma
Hämeenlinnan korkeakoulukeskus

Kevät, 2020

Jenni Heikkilä

Tietojenkäsittelyn koulutusohjelma
Hämeenlinnan korkeakoulukeskus

Tekijä	Jenni Heikkilä	Vuosi 2020
Työn nimi	Odoo ERP -järjestelmän räätälöinti	
Työn ohjaaja	Lasse Seppänen	

TIIVISTELMÄ

Tämä opinnäytetyö on tehty toimeksiantona yritykselle SV Maksujärjestelmät Oy. Työn tavoitteena oli tutkia Odoo ERP -järjestelmän räätälöintiä moduulien avulla, sekä millaisia liiketoimintamahdollisuuksia tämä antaa toimeksiantajalle. Tutkimuskysymyksiä oli kaksi. Ensin tutkittiin, miten moduuleja ohjelmoidaan. Toisena tutkimuskysymyksenä oli, miten moduuleista saadaan toimeksiantajan liiketoimintaa.

Työn teoriaosuudessa käydään läpi yleisesti toiminnanohjausjärjestelmiä ja mitä avoin lähdekoodi tarkoittaa. Lisäksi käydään tarkemmin läpi Odoo ERP -järjestelmän arkkitehtuuria, Odoon kahta mallia (Community ja Enterprise), sekä Odoon ympärille rakennettavia liiketoimintamahdollisuuksia. Empiirisessä osassa on tutkittu ja toteutettu moduulien ohjelmointia kolmella eri tavalla. Ensimmäisenä tutkittiin moduulin ohjelmointia kokonaan alusta alkaen, toisessa Odoo Studio -työkalun käyttöä ja kolmannessa olemassa olevan moduulin räätälöintiä. Näiden pohjalta on tutkittu moduulien ympärille mahdollistettuja liiketoimintamahdollisuuksia pohjaten myös teoriaan.

Tuloksista voidaan todeta, että perusmoduulien ohjelmointi ei itsessään ole kovin vaikeaa, mutta jotta siitä saadaan todellista lisähyötyä asiakkaalle, tulee koodaamisen lisäksi olla ymmärrys käyttäjän liiketoimintaprosesseista. Näin järjestelmä saadaan tukemaan ja ohjaamaan käyttäjän toimintaa. Moduulien ohjelmointiin on tutustuttu työssä vain päällisin puolin ja varsinainen järjestelmän liiketoimintalogiikka koodin näkökulmasta jäi vielä kovin pinnalliseksi. Näistä voidaan todeta, että Odoo luo toimeksiantajalle monia liiketoimintamahdollisuuksia, joista tulee valita resurssien mukaan tietyt osat, joihin keskittyä.

Avainsanat ERP, Odoo, moduuli, avoin lähdekoodi

Sivut 28 sivua

Degree Program in Business Information Technology
Hämeenlinna University Centre

Author	Jenni Heikkilä	Year 2020
Subject	Odoo ERP customization	
Supervisor	Lasse Seppänen	

ABSTRACT

This thesis is done as an assignment for the company named SV Maksujärjestelmät Oy. The main goal of this thesis was to research how to customize Odoo ERP system with modules and what kind of business opportunities that gives for the assignment company. There were two research questions. The first one was how to program Odoo modules. The second one was how these modules can be assigned to the company's business.

The theory part deals with enterprise resource planning systems, what open source means and more closely the Odoo ERP system like architecture of Odoo, differences between Odoo Community and Odoo Enterprise and different kind of business solutions in the Odoo system. The empirical part studies and implemented module programming in three ways. The first one was to program a new module in the beginning, the second one to use Odoo Studio and the third one was to find out how to customize an existing module by programming. On the basis of these, business opportunities around Odoo have been explored.

The results show that programming a basic module is not very difficult, but if you want some real extra value for the customer, you need to understand also business processes and how to combine software and processes. Based on this, it can be said that there are many kind of business opportunities for the Odoo ERP system, that can be implemented.

Keywords ERP, Odoo, module, open source

Pages 28 pages

SISÄLLYS

1	JOHDANTO.....	1
2	TOIMINNANOHJAUSJÄRJESTELMÄT.....	2
2.1	Historia	2
2.2	Avoimen lähdekoodin toiminnanohjausjärjestelmä.....	3
3	ODOO.....	4
3.1	Odoon lisensointi	4
3.2	Odoon liiketoimintamahdollisuudet	5
3.3	Odoon arkkitehtuuri.....	7
3.4	Odoon Studio ja kehittäjätila	8
4	MODUULIN RAKENNE.....	11
4.1	Mallit (models) ja näkymät (views).....	13
4.2	Mallien ja näkymien periytyminen.....	14
4.3	QWeb	15
5	MODUULIN OHJELMOINTI.....	17
5.1	Uuden moduulin ohjelmointi.....	17
5.2	Olemassa olevan moduulin räätälöinti	18
5.2.1	Räätälöinti Studiolla	19
5.2.2	Räätälöinti moduulin ohjelmoinnilla	21
6	TULOKSET JA PÄÄTELMÄT	23
7	YHTEENVETO	25
	LÄHTEET.....	26

SANASTO

Avoin lähdekoodi	Open source. Ohjelmisto, joka on vapaasti levitettävissä ja muokattavissa. Ohjelmistoa kehittää yhteisö
ERP	Enterprise Resource Planning eli toiminnanohjausjärjestelmä, joka sisältää yhdessä ohjelmistossa yrityksen toimintoja kuten CRM, tuotanto, kirjanpito ja varasto
Moduuli	ERP-järjestelmän yksi osa, kuten CRM, laskutus, ostot
MVC	Model-View-Controller -arkkitehtuurimalli, jossa ohjelmiston koodi jaetaan malliin, näkymään ja kontrolleriin
Odoo	Belgialainen avoimen lähdekoodin ERP-järjestelmä
Odoo Studio	Odoon enterprise moduuli, jolla voidaan tehdä räätälöintejä Odoon moduuleihin
ORM	Object-Relational Mapping, kirjasto, joka automatisoi usein käytettyjä SQL-lausekkeita. Kirjoitettavan koodin määrä vähenee
SaaS	Software as a Service, ohjelmiston hankinta palveluna. Ei vaadi ohjelmiston asennusta tai omaa palvelinta, vaan tämä kuuluu lisenssihintaan
Räätälöinti	Ohjelmiston muokkaus käyttäjän tarpeisiin, muutos tehdään koodiin
XML	Extensial Markup Language eli rakenteellinen merkintäkieli

1 JOHDANTO

Yritykset tarvitsevat erilaisia ohjelmistoja ohjaamaan prosessejaan ja toimintaansa. Pienellä yrityksellä ohjelmistoiksi saattavat riittää Excel ja laskutusohjelma, sekä tilitoimiston käyttämä kirjanpito-ohjelma. Mitä suurempi yritys on, sitä enemmän ohjelmistoja yleensä on käytössä. Jotta jokaiseen toimintoon ei tarvita omaa ohjelmaansa, on olemassa toiminnanohjaus- eli ERP-järjestelmiä (Enterprise Resource Planning). ERP-järjestelmät keräävät yhteen ohjelmaan yrityksen toimintoja, joihin kuuluvat esimerkiksi asiakkuudenhallinta, markkinointi, myynti ja laskutus, kirjanpito, tuotanto, varastonhallinta, verkkokauppa ja työajanseuranta. Näin tieto löytyy helposti yhdestä paikasta, mikä säästää työaikaa ja tuo pitkällä aikavälillä säästöjä.

ERP-järjestelmiä on paljon erilaisia ja ne ovat usein keskittyneet tietyille toimialoille. Samoilla toimialoilla liiketoimintaprosessit ovat yleensä samankaltaisia, mutta usein yritykset tekevät asioita hieman eri lailla. Siitä syystä järjestelmä ei välttämättä vastaa sellaisenaan yrityksen tarpeisiin, vaan sitä on muokattava jollain tapaa. Kun järjestelmää muokataan käyttäjän tarpeisiin siinä valmiiksi olevilla asetuksilla, kutsutaan sitä konfiguroinniksi. Jos tämä ei riitä, muutokset tehdään kooditasolle. Tätä taas kutsutaan räätälöinniksi.

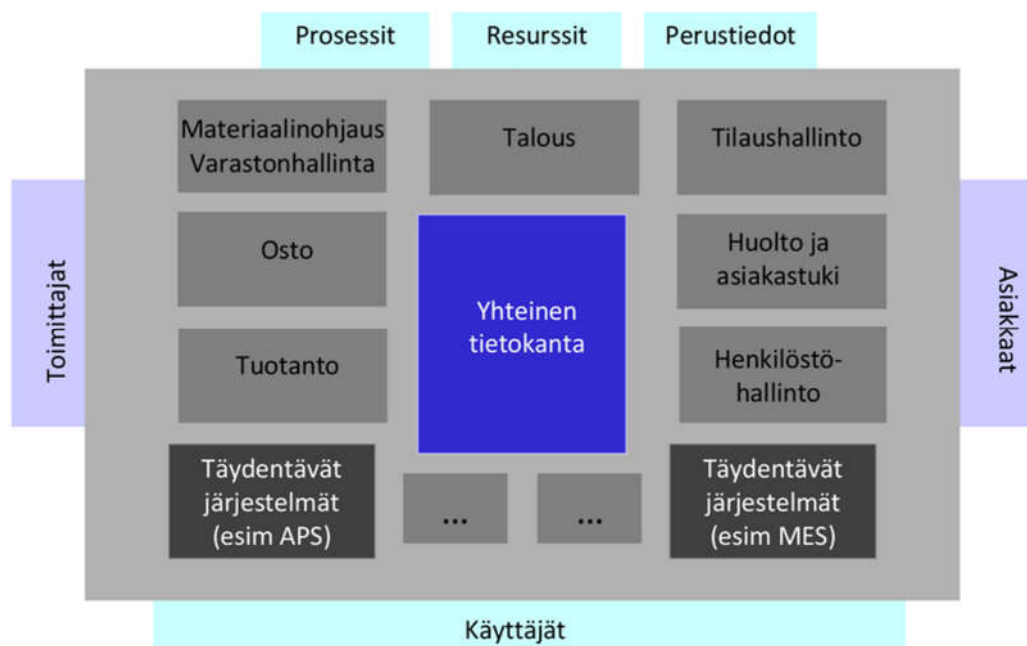
Opinnäytetyö tehdään toimeksiantona lahtelaiselle yritykselle nimeltä SV Maksujärjestelmät Oy (myöhemmin SV). Yritys jälleenmyy erilaisia myynnin ja maksamisen järjestelmiä ja se on ottanut itselleen käyttöönsä jokin aika sitten Odoo ERP -järjestelmän sekä on itse Odoo Learning -partner. Toimeksiantajayritys on jo ottanut Odoon osaksi valikoimaansa, mutta lähinnä vain perustoiminnallisuuksilla ja konfiguroinnein. Opinnäytetyön tarkoituksena on selvittää miten Odoota räätälöidään moduulien avulla, sekä Odoon liiketoimintamahdollisuuksista toimeksiantajalle.

Tutkimuskysymyksinä ovat:

- Miten moduuleita rakennetaan Odooseen?
- Miten moduuleista saadaan SV Maksujärjestelmien liiketoimintaa?

2 TOIMINNANOHJAUSJÄRJESTELMÄT

Toiminnanohjaus- eli ERP-järjestelmät (Enterprise Resource Planning) ovat yritysohjelmistoja, jotka ohjaavat yrityksen liiketoimintaprosesseja yhden järjestelmän sisällä. ERP-järjestelmät ovat modulaarisia, joka tarkoittaa sitä, että ne sisältävät yrityksen toiminnan tarpeen mukaan erilaisia palasia, eli moduuleja. Näitä ovat muun muassa CRM, tuotanto, toimitusketju, henkilöstöhallinto ja taloushallinto. Näillä yksittäisillä moduuleilla on sama tietokanta, jolla ne muodostavat yhden kokonaisen järjestelmän. Järjestelmän tarkoituksena on tehostaa yrityksen toimintaa, kun kaikki löytyy yhden ohjelmiston alta. Näin vältetään monen järjestelmän päällekkäisiltä kirjauksilta, parannetaan asiakaspalvelua ja pidemmällä aikavälillä säästetään kustannuksissa. Myös tiedon jakaminen koko yrityksen sisällä ja datan analysointi helpottuu, kun kaikki löytyy yhden ohjelmiston sisältä. Reaaliaikaisen tiedon avulla johdon on helpompi vastata ja mukautua muuttuviin tilanteisiin. (Rouse, 2019) Kuvassa 1 on eräänlainen kuvaus ERP-järjestelmästä.



Kuva 1. Kuvaus ERP-järjestelmästä (Logistiikan maailma, 2019)

2.1 Historia

ERP-järjestelmien historia alkaa 60-luvulta, jolloin yritykset käyttivät varastojen hallintajärjestelmiä. Tarve on siis alun perin lähtenyt yritysten tarpeesta hallita varastoaan. 70-luvulla järjestelmät kehittyivät materiaali- ja tuotannosuunnittelun (MRP) tarpeisiin, jotka vastasivat lähinnä

tuotteiden ja varaosien vaatimuksiin tuotantoaikataulun mukaan. 80-luvulla järjestelmät alkoivat painottaa tuotantoprosessien optimointia, jossa materiaalien käyttöä suunniteltiin tuotannon mukaan. Tähän kuului lisäksi myymälä- ja jakeluhallinto, projektihallinta, taloushallinto ja henkilöstöhallinto. 80- ja 90-lukujen vaihteessa alettiin puhua ensimmäisen kerran ERP-järjestelmistä, kun yhteen järjestelmään integroitiin muun muassa tuotanto, jakelu, kirjanpito, henkilöstöhallinto, varastonhallinta ja projektihallinta. 2000-luvulla on alettu puhua laajennetusta ERP:stä, kun järjestelmään lisättiin muita moduuleja yrityksen ydintoiminnan lisäksi. (Rashid, Hossain, & Patrick, 2002)

Vanhat ERP-järjestelmät asennettiin yrityksen tiloihin omalle palvelimelle, joka vaati kustannuksia laitteistoon ja osaamista palvelimen ylläpidosta talon sisällä. Nykyisin yhä useammin ohjelmiston mukaan ERP-järjestelmät pyörivät pilvessä, joka muun muassa pienentää käyttönoton kustannuksia, sekä helpottaa järjestelmän päivitystä ja ylläpitoa. Yrityksen ei tarvitse investoida kerralla suuria summia palvelimeen, kun tämä ostetaan palveluna. (Rouse, 2019)

2.2 Avoimen lähdekoodin toiminnanohjausjärjestelmä

Avoimella lähdekoodilla (open source) tarkoitetaan ohjelmistoa, jonka lähdekoodi on avoimesti käytettävissä, jaettavissa, muokattavissa ja kopioitavissa ilman lisenssimaksuja. Ohjelmiston kehitykseen osallistuu yhteisö, johon lukeutuu sekä yrityksiä että yksityisiä henkilöitä. Avoimen lähdekoodin ohjelmistoissa ohjelmistovirheet löydetään ja korjataan usein nopeasti laajan kehittäjäyhteisön ansiosta. Tämä myös lisää ohjelmiston tietoturvaa ja laatua. Avoimen lähdekoodin hyötyihin kuuluu erityisesti riippumattomuus toimittajasta. Kun koodi on avointa, ei ohjelman käyttäjällä ole riskiä joutua yhden toimittajan varaan. (COSS ry, 2019)

ERP-järjestelmät tunnetaan yleisesti suljetuista ja kalliista ohjelmistoista kuten SAP, Oracle ja Microsoft Dynamics. Ohjelmiston tuottaa, toimittaa ja tukee yksityinen yritys. ERP-järjestelmistä löytyy myös avoimen lähdekoodin ratkaisuja, joissa ohjelmiston käyttäjä ei ole toimittajariippuvainen. Tällaisia ohjelmistoja ovat esimerkiksi Odoo, ERP Next ja Dolibarr. (Cybrosys Technologies, 2018)

3 ODOO

Odoo on belgialainen avoimen lähdekoodin ERP-järjestelmä. Odoo S.A., alkuperäiseltä nimeltään TinyERP on perustettu vuonna 2005. Sittemmin ohjelmisto muutettiin OpenERP-nimiseksi ja vuonna 2014 otettiin käyttöön nykyinen nimi Odoo. Odoon vakiomoduuleihin kuuluvat muun muassa CRM, verkkosivut, verkkokauppa, ostot, myynti, varastonhallinta, tuotanto, kassajärjestelmä ja henkilöstöhallinto. Lisäksi Odoon sovelluskau-pasta on ladattavissa ja ostettavissa tuhansia lisämoduuleita. Yrityksen pe-rustaja Fabien Pinckaers on sanonut perustaneensa Odoon erityisesti kil-pailemaan SAPia vastaan. (Pinckaers, 2013; Nurmi, 2018)

Odoo jakaantuu kahteen osaan: avoimen lähdekoodin Odoo Communityyn ja maksulliseen lisenssimaksujen takana olevaan Odoo Enterpriseen. Com-munity-versio on varsinainen avoimen lähdekoodin ratkaisu, joka asenne-taan omalle halutulle palvelimelle ja jota voidaan tarvittaessa räätälöidä halutulla tavalla. Enterprise-versiossa on sama Community-version pohja, jossa lisenssit ostetaan vuosittain käyttöön Odoolta. Ohjelmisto asenne-taan joko omalle palvelimelle tai vaihtoehtoisesti Odoon ylläpitämälle Odoo.sh palvelimelle. Community-versioon on asennettavissa rajattu määrä Odoon vakiomoduuleita. Enterprise-versiossa vakiomoduuleita on enemmän ja se toimii myös mobiilissa toisin kuin Community. Enterprise-versiota on myös mahdollista räätälöidä lisenssiehtojen mukaisesti kuten Communityä. Lisäksi Enterprise-versio on mahdollista ottaa käyttöön Odoon SaaS-mallilla. Tästä puhutaan nimellä Odoo.com, joka on ilmainen niin kauan kun käytössä on vain yksi moduuli. Hostauksesta ei tule kuluja. Tähän ratkaisuun on mahdollista ladata ainoastaan Odoon omia vakiomo-duuleita ja räätälöinti ei ole mahdollista. (Odoo S.A., 2019b; Reis Daniel, 2018; Odoo S.A., 2019a)

Odoo julkaisee vuosittain uuden version. Tätä kirjoitettaessa uusin jul-kaistu versio on 13. Kun Odoota räätälöidään moduuleilla, tehdään ne aina tiettyyn versioon. Moduulin käyttöönotossa tulee olla tarkkana mihin ver-sioon se on tehty, sillä eri versioiden moduulit eivät ole suoraan yhteenso-pivia toisiinsa. Odoo ylläpitää itse aina kolmea viimeisintä versiota. Tätä kirjoitettaessa tuettuja versioita ovat siis 11, 12 ja 13. Mikäli aiemmista versioista löytyy bugeja tai tietoturvaongelmia, niitä ei enää korjata. (Reis Daniel, 2018)

3.1 Odoon lisensointi

Vaikka avoimen lähdekoodin ohjelmistot ovat vapaasti käytettävissä ja muokattavissa, myös niiden käyttöä sääntelevät erilaiset lisenssit. Odoon Community 12 -versiossa pätee GNU LGPL versio 3:n lisenssi, eli GNU Les-ser General Public License. Aiempi GNU GPL eli GNU General Public License antaa vapauden käyttää ja tutkia ohjelmaa ja jakaa sitä edelleen, kunhan lähdekoodi ja lisenssi annetaan mukana. Ohjelmaa saa muuttaa ja levittää

eteenpäin, kunhan kyseisistä muutoksista on ilmoitettu selkeästi. Odoo Communityn käyttämä GNU LGPL on muutoin samanlainen kuin GNU GPL, mutta se sallii lisäksi toisten ohjelmien liittämisen, jotka eivät ole saman lisenssin alaisia. Tämä tarkoittaa sitä, että Odoon kanssa on mahdollista käyttää yhdessä suljettuja ohjelmistoja. (Linux.fi-wiki, 2015; Linux.fi-wiki, 2017)

Odoo Enterprise -versio on Odoon oman lisenssin alainen ohjelmisto. Ohjelmistoa on mahdollista käyttää voimassaolevalla tilauksella ja sillä käyttäjämäärällä, joka tilauksella on mainittu ja maksettu. Lisäksi partnereilla on lupa käyttää Enterprise-versiota testaus- ja kehitysympäristössä, kunhan partnersopimus on voimassa. Enterprise-version kopiointi, jakelu ja myynti on kielletty, mutta omien Odoo-moduulien julkaisu on sallittu, jos moduuli on yhteensopiva Odoon Enterprise-lisenssin kanssa. Sallittuihin lisensseihin kuuluu muun muassa yllä mainittu LGPL, jota Odoo Community käyttää. (Odoo S.A., 2018c)

3.2 Odoon liiketoimintamahdollisuudet

Community- ja Enterprise-versioiden kehittämisen lisäksi Odoo myy asiakkailleen suoraan omia palveluitaan, kuten järjestelmän käyttöönottoa ja ylläpitopalveluita. Odoolla on lähes kymmenen toimipistettä ympäri maailmaa, jonka lisäksi Odoolla on oma partneruusohjelma. Virallisia partnereita yli sadassa eri maassa. (Odoo S.A., 2018a)

Odoo-partneruudessa on neljä eri tasoa. Ensimmäinen on Learning-partner, joka ei ole vielä virallinen partneri, eli ei saa näkyvyyttä Odoon sivuilla eikä kaikkia partneruuden etuja. Tämän jälkeen tulevat viralliset partneritasot, jotka ovat Ready, Silver ja Gold. Kaikilla partnereilla on pääsy Enterprise-version lähdekoodiin, Odoon teknisiin ja toiminnallisiin dokumentteihin sekä koulutusmateriaaliin, mahdollisuus antaa asiakkaille Enterprise kokeilukäyttöön, mahdollisuus saada Odoolta apua implentointiin, sekä komissio myydyistä Enterprise-lisensseistä. Tämän lisäksi virallisilla partnereilla on Odoolta nimetty yhteyshenkilö, näkyvyys Odoon virallisilla sivuilla, sekä partnereilla on pääsy Odoon kautta kyseiseen maahan tulleisiin liideihin. Suomessa on työn tekohetkellä viisi virallista partneria, kolme Ready-partneria ja kaksi Gold-partneria. (Odoo S.A., 2018a)

Odoo koostuu Odoon omista vakiomoduuleista, joihin kuuluu muun muassa CRM, myynti ostot, varastonhallinta, tuotannonohjaus, kassajärjestelmä, nettisivut ja verkkokauppa. Kuvassa 2 näkyy Enterprise-version työpöytä, jossa on suuri osa Odoon vakiomoduuleista. Tämän lisäksi Odoota voidaan räätälöidä kolmansien osapuolien moduuleilla, joita on sekä ostettavissa/ladattavissa Odoon omilta sivuilta tai tilattavissa IT-taloilta, kuten virallisilta partnereilta. Moduuli voi olla yksi pieni lisätoiminnallisuus jo olemassa olevaan moduuliin, tai suurempi kokonaisuus, joka laajentaa ohjelmistoa merkittävästi. (Nurmi, 2018)



Kuva 2. Odoo 12 Enterprise moduuleja Odoon Dashboard näkyssä (Yasir Muhamed, 2018)

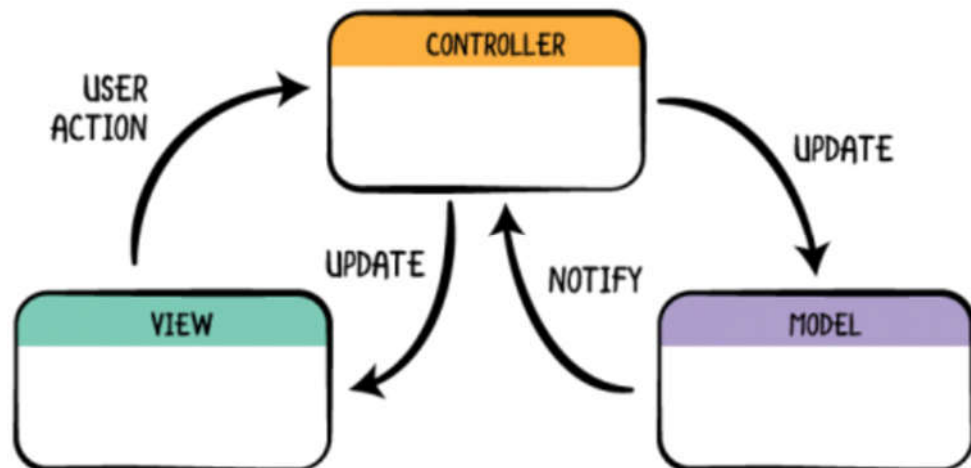
Odoo.comista löytyy reilu 20 000 erilaista moduulia, joista suosituimpiin kuuluvat muun muassa ulkopuolisen verkkokaupan ja Odoon yhdistävä Odoo-Woocommerce liittymä ja Odoo-Shopify liittymä, erilaiset raportti- ja dashboard näkymät sekä toimialakohtaiset moduulit. Hinnat vaihtelevat pääosin ilmaisista muutamaan sataan euroon. Viralliset Odoo-partnerit voivat ladata moduuleitaan Odoo.comiin myyntiin. Odoo ei varsinaisesti valvo ladattuja moduuleita, mutta moduuleita arvostellaan ja mikäli Odoolle tulee tietoon väärinkäyttöksiä, he ryhtyvät välittömästi toimiin. Moduulin julkaisija saa jokaisesta Odoo.comissa myydystä moduulista 70%. (Odoo S.A., 2019c) Tämän lisäksi kuka vain voi avoimen lähdekoodin mukaisesti tehdä omia moduuleitaan ja myydä niitä eteenpäin LGPL-lisenssiehtojen mukaisesti.

Community-version puolella Odoon ympärillä oleva liiketoiminta on siis lähinnä Odoo Community -version käyttöönottoon liittyviä projektipalveluja, moduulien koodaamista ja ylläpitopalveluja. Partnerit saavat lisäksi jatkuvaa tuloa myydystä Enterprise-lisensseistä ja Odoo.comissa myydystä moduuleista.

Odoon itselleen projektin käyttöönotto ja asiakkaan liiketoiminnan syvälinen ymmärtäminen ovat erityisen tärkeitä asioita, jotta projekti saadaan onnistumaan. Odoon kouluttaa partnereitaan implementointiin ja heidän metodologiansa mukaan projekti tulee pitää niin yksinkertaisena kuin mahdollista. Ylimääräistä räätälöintiä tulee välttää niin kauan, kun sille on oikeasti tarvetta. Projektin prioriteettina on aina tyytyväiset asiakkaat, jossa projekti on tehty ajallaan ja budjetin sisällä. Tämä kun ei ole ERP-projekteissa mikään itsestäänselvyys. Odoon oman toimintaperiaatteen mukaan projektilla on aina oma projektipäällikkö, joka huolehtii projektin yksinkertaistamisesta ja asiakasyrityksen liiketoimintaprosessien sujuvasta kulusta yhdessä Odoon-ohjelmiston kanssa. Odoon antaa partnereilleen oman osaamisensa projektien implementointiin nimetyn yhteyshenkilön kautta. (Odoon S.A., 2019b)

3.3 Odoon arkkitehtuuri

Odoon arkkitehtuuri perustuu kolmiosaiseen MVC-malliin. MVC tulee sanoista malli (model), näkymä (view) ja kontrolli (controller). Malli sisältää ohjelmiston tietokannan ja datan, sekä ohjelmiston logiikan. Malli on vastuussa tietojen ylläpidosta. Näkymä on ohjelmiston käyttöliittymä, eli mallin näkymä. Näkymä ja malli ovat itsenäisiä osia. Jos käyttöliittymään eli näkymään tehdään muutoksia, ne eivät vaikuta malliin, eli tietokantaan. Samoin tietokannan tietoja voidaan järjestää uudelleen muuttamatta käyttöliittymää. Malli ja näkymä kommunikoivat kontrollerin avulla. (Nizarudheen, 2018; Odoon S.A, 2013) Tämä on kuvattu kuvassa 3.



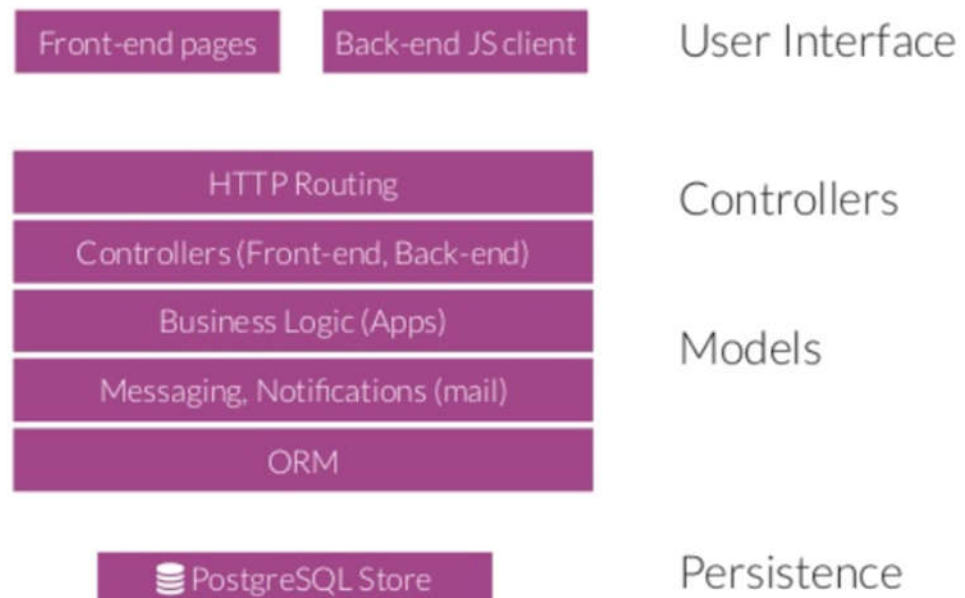
Kuva 3. MVC arkkitehtuuri (PT, 2018)

MVC-malli toimii Odoossa seuraavasti taulukon 1 mukaisesti.

Taulukko 1: MVC-malli Odoossa

Malli:	PostgreSQL tietokanta
Näkymä:	XML- ja HTML mallit
Kontrolli:	Python-luokat ja funktiot

Käytännössä tämä toimii niin, että painettaessa Odoon käyttöliittymässä jotakin haluttua toimintoa, kyseinen kontrolleri käynnistyy ja tämä vie tiedon eteenpäin kyseiselle python-toiminnolle. Tämä toiminto kommunikoi mallin (tietokannan) kanssa, hakee pyydetyt tiedot ja näyttää lopputuleman käyttäjän havaitsemassa muodossa käyttöliittymässä XML- tai HTML-mallien avulla. (PT, 2018)



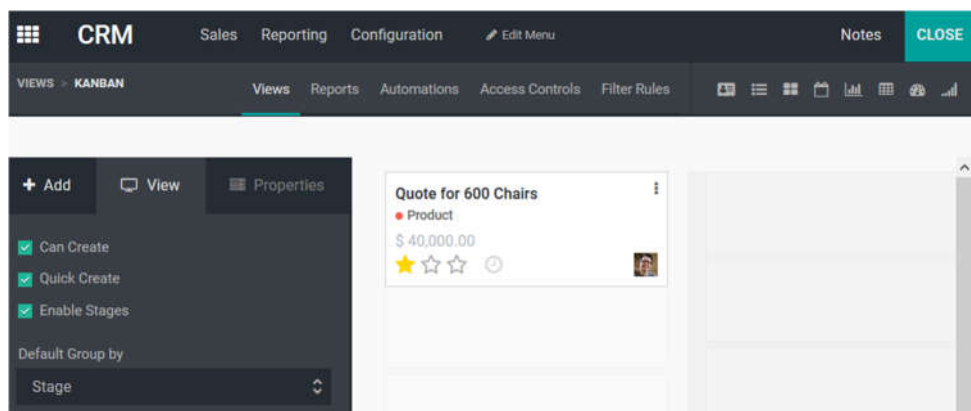
Kuva 4. Odoon arkkitehtuuri (Dony, 2014)

Odoon arkkitehtuuri on kuvattu vielä tarkemmin yllä olevassa kuvassa 4. Kuten kuvasta näkyy, PostgreSQL-tietokannan yläpuolelta löytyy ORM-kerros eli Object-Relational Mapping. ORM on eräänlainen relaatiotietokannoille tarkoitettu kirjasto, joka automatisoi usein käytettyjä SQL-lausekkeita. (Makai Matt, 2012) Näitä ovat muun muassa tietojen lukeminen, luominen, päivittäminen, poistaminen, sekä tietojen tuonti ja vienti (Reis Daniel, 2018).

3.4 Odoo Studio ja kehittäjätila

Odoota on mahdollista räätälöidä käytännössä kahdella tapaa; erillisillä moduuleilla ja Odoo Studiolla. Moduuli on oma Odooseen liitettävä osa,

joka on mahdollista asentaa eri tietokantoihin. Odoo Studio on Odoon Enterprise-versioon saatavilla oleva moduuli, jolla muutokset tehdään suoraan Odoon käyttöliittymässä. Työkalulla on mahdollista tehdä täysin uusi moduuli tai vain pieniä muutoksia olemassa oleviin moduuleihin. Studion käyttäjän ei tarvitse tietää koodista mitään, mutta jonkinlainen ymmärrys ohjelmiston toiminnasta on suotavaa, sillä Studiolla pystyy myös tekemään vahinkoa, jos ei tiedä mitä tekee. Tämä on hyvä työkalu, jos halutaan nopeasti yksittäisiä muutoksia moduulin toimintaan. Mikäli Odoo päivitetään uudempaan versioon, Studiolla tehdyt muutokset katoavat. Kyseiseen tietokantaan tehdyt muutokset Studiolla on ladattavissa zip-tiedostona. (Odoo S.A., 2019a) Kuvassa 5 näkyy miltä Odoo Studio-työkalu näyttää käytössä.



Kuva 5. Odoo Studio-työkalu

Odoossa on rakennettuna lisäksi 'debug mode' eli kehittäjätila (developer mode). Debug modessa on mahdollista tehdä joitakin muutoksia näkymiin, mutta lähinnä se on tarkoitettu kertomaan määrittämiin tarvittavia tai muita teknisiä tietoja. Kuvassa 6 näkyy miten Mahdollisuus-kentän lisätiedot näkyvät, kun hiiri laitetaan kyseisen kentän päälle debug modessa. (Ijas, 2019)

CRM Myynti Liidit Raportointi Asetukset

Myyntiputki / test

TALLENNA HYLKÄÄ

UUSI TARJOUS MERKITSE VOITETUKSI MERKITSE HÄVITYKSI

Mahdollisuus

- o Kenttä: name
- o Objekti: crm.lead
- o Tyyppi: char
- o Muuttajat: {required:true}

Mahdollisuus< Esime isuus

Odotetut tulot	Todennäköisyys
0,00	€ ~ 10

Asiakas _____

Sähköposti _____

Puhelin _____

Myyjä Jenni Heikkilä _____

Myyntitiimi Myynti _____

Sisäiset kommentit Seuranta

Kuva 6. Mahdollisuus-kenttä debug modessa

4 MODUULIN RAKENNE

Odoo koostuu siis moduuleista, joita ovat Odoo kehittäjäyhteisön viralliset moduulit, Enterprise-version maksulliset moduulit sekä kolmansien osapuolien moduulit. Perusasennuksen mukana tulee jo useampi sata moduulia, sekä Odoo.comista on ostettavissa/ladattavissa tuhansia moduuleita. Moduuli voi olla joko yksi suuri kokonaisuus kuten CRM, tai yksi toiminnallisuus jo olemassa olevaan moduuliin, kuten suomalainen viitenumero las-kulle.

Moduuli sisältää aina tietyn hakemistorakenteen. Näitä tutkimalla selvää moduulin liiketoimintalogiikka. Hakemisto on kuvattu taulukossa 2.

Taulukko 2. Moduulin rakenne (Odoo S.A., 2018f)

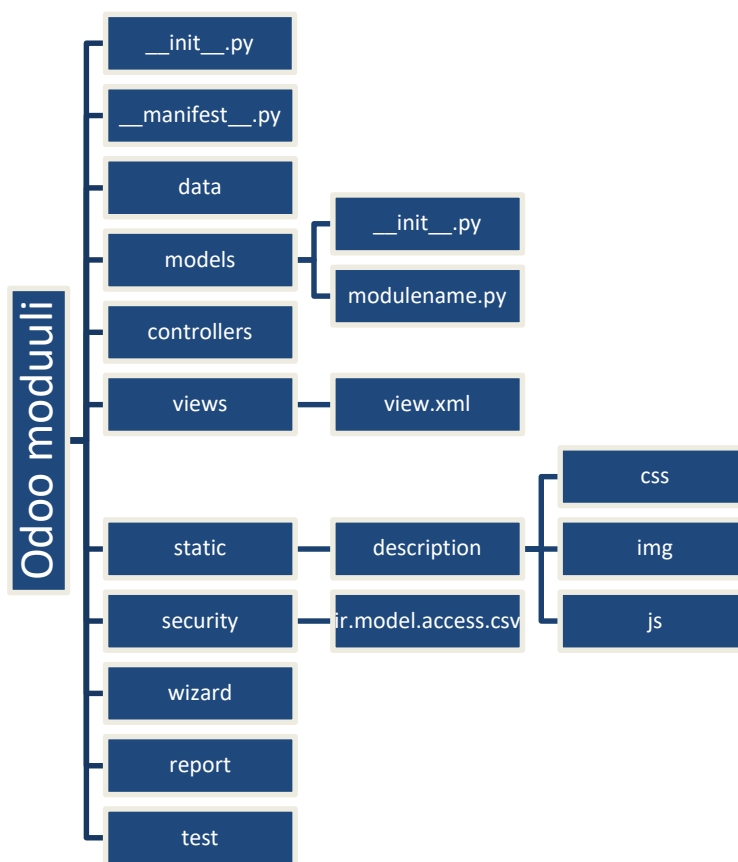
Hakemisto	Sisältö
data	demo ja data xml-tiedostot
model	mallien määritelmät (python tiedostot)
controllers	kontrollerit (http reitit)
views	näkymät ja XML-mallit (xml-tiedostot)
static	käyttöliittymään liittyvät tiedostot erotellen css-, javascript- ja kuvatiedostohakemistoihin

Lisäksi valinnaisia hakemistorakenteita ovat muun muassa taulukossa 3 kuvatut.

Taulukko 3. Valinnaisia hakemistoja (Odoo S.A., 2018f)

Hakemisto	Sisältö
security	käyttöoikeuksien hallinta käyttäjille ja ryhmille
wizard	ohimenevien mallien uudelleenryhmittely ja niiden näkymät
report	sisältää tulostettavat raportit, jotka perustuvat SQL-näkymiin
test	python-testit

Kuvassa 7 koko moduulin rakenne on kuvattu vielä selkeämpänä hierarkisena mallina.



Kuva 7. Moduulin rakenne kuvattuna edellisten taulukoiden pohjalta

Moduulin juuressa on aina tiedostot `__manifest__.py` ja `__init__.py`. Nämä kertovat Odoolle, että kyseessä on moduuli. Manifest-tiedostossa on moduulin metatiedot. Se kertoo muun muassa moduulin nimen, version, tekijän, kuvauksen ja lisenssin. (Odoo S.A., 2018d) Tiedosto `__init__.py` sisältää tuontiohjeet moduulin eri Python-tiedostoille (Odoo S.A., 2018b). Manifest-tiedoston esimerkki löytyy kuvasta 8.

```

{
    'name': "A Module",
    'version': '1.0',
    'depends': ['base'],
    'author': "Author Name",
    'category': 'Category',
    'description': """
Description text
""",
    # data files always loaded at installation
    'data': [
        'views/mymodule_view.xml',
    ],
    # data files containing optionally loaded demonstration data
    'demo': [
        'demo/demo_data.xml',
    ],
}

```

Kuva 8. Esimerkki `__manifest__.py` tiedostosta (Odoo S.A., 2018d)

4.1 Mallit (models) ja näkymät (views)

Mallit ovat liiketoimintaobjekteja, jotka kertovat moduulin datastruktuurin. Malli on linkki sovelluspalvelimen ja tietokannan välillä ja se sisältää moduulin niin sanotun ”bisneslogiikan”. Mallin luokissa kerrotaan attributit, joita se käyttää. Kuvassa 9 yksinkertainen esimerkki mallista.

```

from odoo import models, fields

class LessMinimalModel(models.Model):
    _name = 'test.model2'

    name = fields.Char()

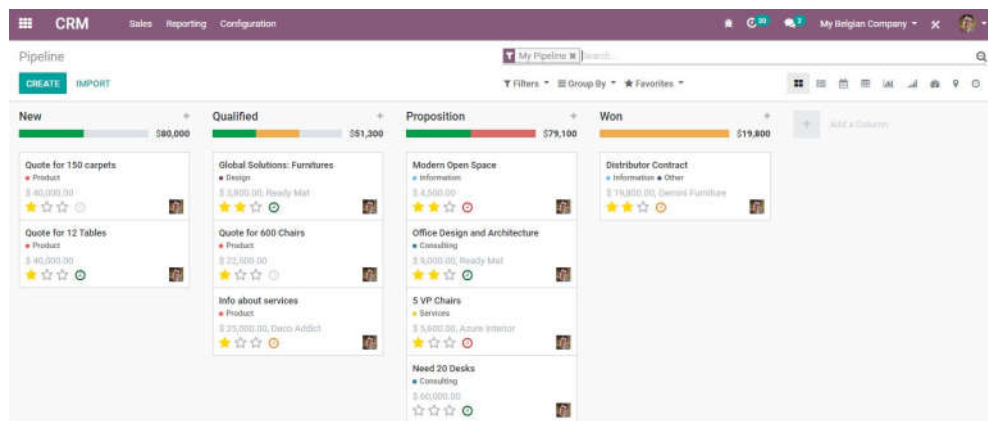
```

Kuva 9. Esimerkki mallista, python-tiedosto (Odoo S.A, 2019)

Mallit ovat itsenäisiä osia ja jos niiden halutaan käyttävän ja lukevan toisensa tietoja, on sitä varten käytettävä relaatiokenttiä. Suhteita on kolmenlaisia, jotka ovat monen suhde yhteen (Many2one), yhden suhde moneen (One2many) ja monen suhde moneen (Many2many).

Mallissa monen suhde yhteen viittaa siihen, että siinä kyseisellä mallilla on viittauksia toisen mallin yhteen kenttään. Vastaavasti yhden suhde moneen on päinvastainen – se luettelee mallit ja niiden kentät, jotka ovat yhteydessä tähän kenttään. Monen suhde monessa taas on molemmin puolin monen välinen suhde. (Reis Daniel, 2018)

Odoon näkymät määrittelevät sen, miten mallit näytetään käyttäjälle eli siis käyttäjän näkemä käyttöliittymä. Perusnäkyymiä on kolmenlaisia. Nämä ovat lista, lomake ja haku. Tämän lisäksi on olemassa esimerkiksi Kanban- ja kalenterinäkymät. Näkymät määritellään xml-tiedostoina. (Odoo S.A, 2019) Kuvassa 10 näkyy CRM-moduuli kanban-näkymässä ja kuvassa 11 sama CRM-moduuli listanäkymässä.



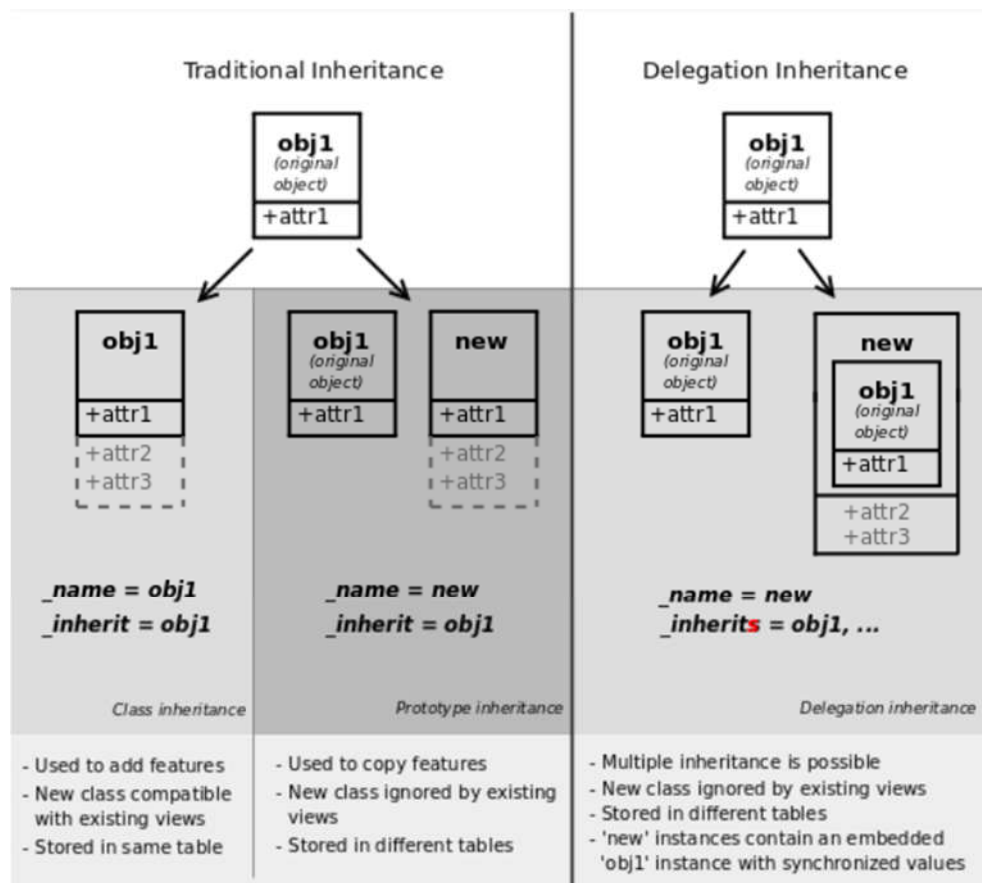
Kuva 10. CRM-moduuli kanban-näkymässä (Odoo tricks and tips, 2019)

Quotation Number	Create Date	Delivery Date	Expected Date	Customer	Website	Salesperson	Company	Total	Status
900037	12/18/2019 09:20:37		12/17/2019 21:23:31	Ready Mat		Mitchell Admin	Demo Company	\$ 44,275.00	Sales Order
900036	12/18/2019 09:20:37		12/17/2019 21:23:31	Deco Addict		Mitchell Admin	Demo Company	\$ 18,400.00	Sales Order
900007	12/18/2019 09:20:37		12/17/2019 21:21:17	Gemini Furniture		Mitchell Admin	Demo Company	\$ 14,981.00	Sales Order
900006	12/18/2019 09:20:37		12/17/2019 21:21:17	Lumber inc		Mitchell Admin	Demo Company	\$ 750.00	Sales Order
900004	12/18/2019 09:20:37		12/17/2019 21:21:16	Gemini Furniture		Mitchell Admin	Demo Company	\$ 2,240.00	Sales Order
900003	12/18/2019 09:20:37		12/18/2019 11:12:38	Ready Mat		Mitchell Admin	Demo Company	\$ 1,127.50	Quotation
900019	12/18/2019 09:20:37		11/17/2019 21:21:17	YourCompany, Joel Willis		Mitchell Admin	Demo Company	\$ 2,947.50	Sales Order
900018	12/18/2019 09:20:37		12/18/2019 11:12:38	YourCompany, Joel Willis		Mitchell Admin	Demo Company	\$ 9,705.00	Quotation Sent
900002	12/18/2019 09:20:37		12/18/2019 11:12:38	Ready Mat		Mitchell Admin	Demo Company	\$ 2,947.50	Quotation

Kuva 11. CRM-moduuli listanäkymässä (Odoo tricks and tips, 2019)

4.2 Mallien ja näkymien periytyminen

Mallien puolella Odoossa on kaksi periyntymismekanismia. Ensimmäinen periyntymismekanismi sallii moduulin muuttaa toisen moduulin toimintaa esimerkiksi lisäämällä tai ohittamalla jonkin kentän, lisäämällä metodin malliin tai ohittamalla jonkin metodin. Toinen periyntymismekanismi on delegointi, jossa jokainen mallin kenttä voidaan linkittää päämalliin. Tämä tarjoaa läpinäkyvän pääsyn päämallin kenttiin. Nämä kaksi periyntymistä-paa on kuvattu kuvassa 12. (Odoo S.A, 2019)



Kuva 12. Mallien periytyminen (Odoon S.A, 2019)

Myös näkymien puolella on Odoossa perinnöllisyys, jolloin näkymiä ei tarvitse kirjoittaa kirjoittamalla vanhojen yli. Tämä toteutetaan lapsinäkymillä, jotka voivat lisätä tai poistaa tietoa päänäkymiltä. Yksittäisen näkymän sijaan käytetään haluttu määrä xpath-elementtejä, joilla päänäkymää muutetaan. (Odoon S.A, 2019)

4.3 QWeb

Odoon raportit kirjoitetaan HTML/QWeb-muodossa, kuten Odoon kaikki tavalliset näkymät. Templatet tehdään samoin XML-tiedostoina. Käyttäjä saa raportit käyttöönsä pdf-muodossa, jotka muodostuvat wkhtmltopdf-työkalulla. Raportit ovat todellisuudessa omia sivujaan, joista muodostetaan PDF-tiedosto raportin oman URL-osoitteen perusteella. (Odoon S.A., 2018g)

Ensin raportti määritellään attribuutteineen <report> elementissä. Tämän jälkeen raportin näkymä määritellään omassa template-elementissään. Näistä näkyy esimerkki kuvissa 13 ja 14. Raportille voidaan määritellä oma paperiformaatti. Mikäli raportin tulee hakea tietoja myös muista malleista, tarvitsee sitä varten luoda oma malli.py -tiedosto. (Odoon S.A., 2018g)

```
<report
  id="account_invoices"
  model="account.invoice"
  string="Invoices"
  report_type="qweb-pdf"
  name="account.report_invoice"
  file="account.report_invoice"
  attachment_use="True"
  attachment="(object.state in ('open','paid')) and
              ('INV'+(object.number or '').replace('/','')+'.pdf')"
/>
```

Kuva 13. Report elementti ja sen attribuutit (Odoo S.A., 2018g)

```
<template id="report_invoice">
  <t t-call="web.html_container">
    <t t-foreach="docs" t-as="o">
      <t t-call="web.external_layout">
        <div class="page">
          <h2>Report title</h2>
          <p>This object's name is <span t-field="o.name"/></p>
        </div>
      </t>
    </t>
  </t>
</template>
```

Kuva 14. Raportin template <template> elementin sisällä (Odoo S.A., 2018g)

5 MODUULIN OHJELMOINTI

Opinnäytetyön käytännön osuudessa selvitetään toimeksiantajayritykselle, miten moduuleja ohjelmoidaan ja tästä tehdään kirjallinen dokumentaatio. Tämän lisäksi Odoon mahdollistamia liiketoimintamahdollisuuksia peilataan toimeksiantajayrityksen mahdollisuuksiin tarjota entistä laajempia palveluita Odoon ympärillä.

Toimeksiantajalla on itsellään käytössä Odoo Enterprise versio 12, joka sijaitsee Odoon omalla Odoo.sh palvelimella. Odoo.sh toimii selaimella ja siinä on itsessään developer-työkaluja, kuten editorina Jupyter Notebook, Python konsoli ja Odoo Shell. Odoo.sh:n käyttö vaatii github tunnuksen. Sh:ssa voi luoda suoraan eri brancheja eri tiloissa (development, staging, production), sekä tehdä näistä buildseja. Odoo.sh:ssa toimii suoraan githubiin yhdistettynä gitin perustoimintoja kuten push, pull, merge, fork ja clone. Lisäksi opinnäytetyön tekijällä on omalla koneellaan asennettuna Odoo Community versio 12 ja Odoo.sh:ssa oleva projekti, jonne voi lisätä omia Odoo-asennuksia development-tilaan. Tähän työhön tehdyt moduulit ja räätälöinnit on tehty Odoo.sh ympäristössä olevilla työkaluilla, jolloin mitään asennuksia ei ole ollut tarve tehdä. (Odoo S.A., 2018e)

5.1 Uuden moduulin ohjelmointi

Ensimmäisenä moduulin ohjelmointiin lähdettiin tutustumaan täysin uuden moduulin luomisesta. Tällä tavalla alkuun päästiin tutustumalla moduulin rakenteeseen ilman, että oli vielä tarvetta ymmärtää toisen moduulin toimintalogiikkaa tai koodia. Toimeksiantajayrityksellä ei ollut omasta takaa tarvetta täysin uuden moduulin luonnille, joten tätä lähdettiin tutkimaan Odoon oman sivuston ohjeiden sekä Daniel Reisin Odoo 12 Development Essentials – Fourth Edition -kirjan ohjeistuksien mukaan.

Uuden moduulin hakemistorakenne tehdään tiettyyn polkuun, joka on määritelty aina jokaisen Odoo-instanssin määrittelyssä. Kun tietyt tiedostot on luotu, moduuli on jo asennettavissa käyttöliittymästä, vaikka moduulissa ei itsessään ole vielä mitään toimintaa. Kuvassa 15 näkyy Odoon käyttöliittymästä ladattavissa oleva Library Management -moduuli, joka ei tässä vaiheessa vielä sisältänyt muuta kuin pakolliset hakemistot ja tiedostot, sekä SV:n logon.

Library Management

By Jenni Heikkilä

SV.

INSTALL UPGRADE

Information

Technical Data

Website

Category Other

Summary

Technical Name library

License LGPL Version 3

Latest Version 12.0.1.0

Kuva 15. Asennettavissa oleva uusi moduuli

Kun moduuli on asennettu, sitä pääsee muokkaamaan Odoosh:n editorissa, josta moduulin pääsee päivittämään suoraan tietokantaan. Toiminnot pystytään tämän jälkeen testaamaan heti käyttöliittymässä sivun päivityksellä.

Perusmallien ja näkymien ohjelmoinnin logiikkaan pääsi nopeasti kiinni. Kun moduuliin halusi perustoimintojen jälkeen lisätoiminnallisuuksia, täytyi tutustua syvällisemmin Odoon omaan ORM Apiin. Tämän työn ohessa tähän päästiin tutustumaan vasta pintapuolisesti. Tässä vaiheessa tuli erityisesti ilmi se, kuinka paljon moduulin opetteluun tulee käyttää aikaa, jotta asioita alkaa ymmärtämään syvällisesti ja tietää mitä tekee. Myös raporttien laatiminen on vielä oma osansa, johon ei tämän työn ohessa vielä ehditty tutustua.

5.2 Olemassa olevan moduulin räätälöinti

Toisessa vaiheessa tutustuttiin moduulin räätälöintiin, eli olemassa olevan moduulin muokkaamiseen. Tässä on kaksi tapaa, joilla asiaa voi lähestyä. Ensimmäinen on Odo Studio, joka on siis Odoon oma moduuli räätälöintien tekemiseen. Toimeksiantajayrityksellä oli tämä käytössään, joten tässä työssä päästiin tutustumaan tähän vaihtoehtoon. Toinen tapa on moduulin ohjelmointi kooditasolla, jolloin räätälöity moduuli on asennettavissa myös muihin tietokantoihin.

5.2.1 Räätelöinti Studiolla

Toimeksiantajayrityksellä oli joitain yksittäisiä tarpeita oman järjestelmän käyttöön liittyen, joita varten ei ollut järkevää lähteä rakentamaan omaa moduulia. Näitä muutoksia lähdettiin tekemään kokeilemalla Studio-työkäluä.

Toimeksiantajayritys pyörittää maksupäätteiden vuokraustoimintaa Odoon avulla ja tätä varten muun muassa tarjous- ja tilausvahvistuksiin, sekä lähetteisiin tarvitaan vuokrien alku- ja loppupäivämäärät. Odooseen oli jo tehty valmiiksi päivämääräkentät Start Date (Alkupvm) ja End Date (Loppupvm) Studiolla. Nämä näkyvät kuvassa 16.

Tarjous # SO215

Tarjouksen päiväys: 09.09.2019 11.36.25
Myyjä: Jenni Heikkilä
Viimeinen voimassaolo: 16.09.2019

Kuvaus	Alkupvm	Loppupvm	Määrä	Yksikköhinta	Verot	Summa
[N] Nets iisi 1kk iWL250 3G Viking	23.09.2019	22.10.2019	1,00 Kpl	75,00	VAT 24%	75,00 €
Välisumma						75,00 €
Verot						18,00 €
Yhteensä						93,00 €

Kuva 16. Tarjouksella näkyvät kentät

Kun tarjous on vahvistettu tilaukseksi, luo Odoo automaattisesti toimitustilauksen, jossa näkyvät nyt samat päivämäärät. Aiemmin päivämäärä piti tarkistaa muualta, jotta voitiin varmistua, milloin laite tulee toimittaa eteenpäin asiakkaalle (kuva 17). Samat päivämäärät näkyvät myös asiakkaalle tulostettavassa läheteessä, joka näkyy kuvassa 18.

Toiminnot			
Lisätietoa			
Muistiinpano			
Tuote	Start date	End date	Alkuperäinen tarve
IWL250 3G VIKING	23.09.2019	22.10.2019	1,00

Kuva 17. Käyttöliittymässä näkyvät kentät varaston toimitustilauksella

WH/OUT/00171

Tilaus	Päivä
SO215	23.11.2019 21.42.32

Tuote	Alkupvm	Loppupvm	Määrä
IWL250 3G VIKING	23.09.2019	22.10.2019	1,00 Kpl

Kuva 18. Lähetete, joka tulostetaan fyysisen laitteen mukaan asiakkaalle

Odoo Studiolla tehdyt muutokset ovat vietävissä zip-tiedostona ja palautettavissa takaisin tietokantaan suoraan käyttöliittymästä. Nykyiset muutokset on viety omalle koneelle, josta pääsi näkemään kooditasolla tehtyjä muutoksia. Tässä tapauksessa muutoksia oli tehty näkyviin, jolloin zip-tiedostosta löytyi muutama xml-tiedosto. Asiakaskortille on tehty yksi kenttä, johon voidaan laittaa suoran linkki kyseisen asiakkaan Vainu-kortille. Kyseisen kentän xml-koodi näkyy alla olevassa kuvassa 19.

```
- <record context="{ 'studio': True}" model="ir.model.fields" id="studio_customization.new_teksti_res_partn_2936ab6c-c368-4fcd-8238-2e7f60d9a979">
  <field eval="False" name="complete_name"/>
  <field eval="False" name="compute"/>
  <field eval="False" name="copied"/>
  <field eval="False" name="depends"/>
  <field name="domain">[]</field>
  <field name="field_description">Vainu linkki</field>
  <field eval="[(6, 0, [])]" name="groups"/>
  <field eval="False" name="help"/>
  <field eval="False" name="index"/>
  <field name="model">res.partner</field>
  <field name="model_id" ref="base.model_res_partner"/>
  <field name="name">x_studio_vainu_linkki</field>
  <field eval="False" name="on_delete"/>
  <field eval="False" name="readonly"/>
  <field eval="False" name="related"/>
  <field eval="False" name="relation"/>
  <field eval="False" name="relation_field"/>
  <field eval="False" name="required"/>
  <field eval="True" name="selectable"/>
  <field eval="False" name="selection"/>
  <field eval="False" name="size"/>
  <field name="state">manual</field>
  <field eval="True" name="store"/>
  <field eval="False" name="track_visibility"/>
  <field eval="False" name="translate"/>
  <field name="ttype">char</field>
</record>
```

Kuva 19. Odoo Studiolla tehty kenttä asiakaskortille xml muodossa

Versiopäivityksistä Odoo Studiolla tehtyihin muutoksiin ei löytynyt Odoon omilta sivuilta tietoa, mutta yleisesti tästä on saanut sen käsityksen, että päivittäessä versiota muutokset häviävät. Odoo Studiolla tehdyt muutokset ovat vietävissä ja tuotavissa zip-tiedostona käyttöliittymässä, jolloin todennäköisesti muutokset olisivat säilytettävissä myös versiopäivityksissä. Tätä ei kuitenkaan ole tässä työssä testattu.

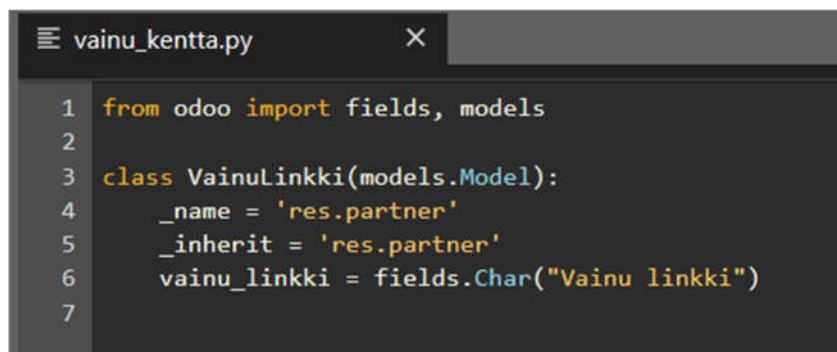
Studiota pystyy käyttämään hyvin intuitiivisesti, kunhan tuntee Odoon perustoiminnan. Tällä on helppo tehdä yksittäisiä muutoksia, kuten lisätä yksittäisiä kenttiä asiakaskortille. Studio-työkalu kuuluu siis Enterprise-versioon ja kyseisen moduulin hinta on 48€/kk. Muiden moduulien hinnat pyörivät pääosin 10-20 euron kuukausihinnoissa. Työn tekohetkellä vain Amazon Connector- ja Ebay -moduulit olivat tätä kalliimpia. Hintaa tuleekin

verrata siihen, kuinka paljon Studiolle on oikeasti tarvetta, kuin että kyseiset muutokset tilattaisiin ohjelmointityönä kertahinnalla. Toisena näkökulmana on myös se, että muutokset pystytään tekemään itse, eikä tilaamaan ulkopuolelta, jolloin käytäntöönpano on huomattavasti nopeampaa.

5.2.2 Räätelöinti moduulin ohjelmoinnilla

Olemassa olevaa moduulia voidaan siis räätälöidä Studion lisäksi ohjelmalla moduuli alusta alkaen samalla logiikalla kuin täysin uusi moduuli. Tässä tapauksessa kaikkia malleja ja näkymiä ei luoda alusta, vaan siinä käytetään olemassa olevia moduuleja pohjalla. Ohjelmoinnin näkökulmasta luodaan siis samanlaiset hakemistorakenteet tiettyyn polkuun, mutta koodissa määritellään mitä olemassa olevia Odoon moduuleja se perii.

Tätä lähdettiin tutkimaan hyvin yksinkertaisella esimerkillä. Aiemmin Studiolla oli tehty asiakaskortille kenttä nimeltä Vainu-linkki. Tämä sama kehitettiin nyt toteuttaa erillisenä moduulina, jossa pääsi tutustumaan Odoon periytymismekanismiin. Moduulissa siis vain laajennetaan olemassa olevaa moduulia ilman, että aiempaan koodiin tehdään mitään muutoksia. Kuvassa 20 näkyy tehdyn moduulin malli python-tiedosto, jossa luokan VainuLinkki attribuutit `_name` ja `_inherit` kertovat mistä mallista se periytyy. Tässä tapauksessa siis moduulista `res.partner`, eli kontakteista.



```
1 from odoo import fields, models
2
3 class VainuLinkki(models.Model):
4     _name = 'res.partner'
5     _inherit = 'res.partner'
6     vainu_linkki = fields.Char("Vainu linkki")
7
```

Kuva 20. Malli, joka periytyy `res.partner` moduulista

Tämän lisäksi kyseinen kenttä tuli lisätä nykyiseen olemassa olevaan kontaktinäkökuvään, jotta se olisi näkyvässä ja käytettävissä käyttöliittymässä. Näkymät määritellään `views`-hakemistossa, jonne luotiin kuvan 21 mukainen `views.xml` tiedosto.

```

1 <odoo>
2   <record id="view_vainu_id" model="ir.ui.view">
3     <field name="name">"Vainu linkki"</field>
4     <field name="model">res.partner</field>
5     <field name="inherit_id"
6     ref="base.view_partner_form"/>
7     <field name="arch" type="xml">
8       <xpath expr="//field[@name='website']"
9       position="after">
10        <field name="vainu_linkki" widget="url"/>
11      </xpath>
12    </field>
13  </record>
14 </odoo>

```

Kuva 21. Näkymän määrittely views.xml tiedostossa

Azure Interior

Address: 3404 Edgewood Road
Jonesboro Arkansas (US) 72401
United States

Phone: (870)-931-0505

Mobile:

Email: azure.interior24@example.com

Website: http://www.azure-interior.com

Vainu linkki: www.vainulinkkiexample.com

Language: English

Tags: [Services](#)

Kuva 22. Vainu-linkki käyttöliittymässä kontakti-kortilla

Kuvassa 22 näkyy kyseinen kenttä käyttöliittymän asiakaskortilla Website-kentän alapuolella, kuten tiedostossa views.xml oli määritelty. Tämä on tehty nyt moduulina, joka on asennettavissa ja ladattavissa yksistään haluttuun tietokantaan. Kun moduuli on tehty toisen moduulin päälle, se käyttää kyseisen moduulin tietoja, jolloin kaikkea ei tehdä alusta asti. Tässä tapauksessa nykyiseen res.partner Kontakti-moduuliin lisättiin vainu_linkki attribuutti ja se määriteltiin näkymään omassa näkymässään, joka ei muutoin muuta mitään nykyistä toimintaa.

6 TULOKSET JA PÄÄTELMÄT

Opinnäytetyössä on tutkittu, millaisia erilaisia tapoja on räätälöidä Odoo ERP -järjestelmää ja millaisia liiketoimintamahdollisuuksia se antaa toimeksiantajayritykselle, joka on ottanut Odoon osaksi tuotevalikoimaansa. Tässä työssä on käyty Odoota läpi yleisesti lähinnä teknisestä näkökulmasta ja miten tämä vaikuttaa liiketoimintamahdollisuuksiin sen ympärillä. Yleisesti voidaan todeta, että Odoon ympärille on mahdollista rakentaa monenlaista liiketoimintaa, jossa yritys voi tarjota palvelujaan niin liiketoiminnan kehittämiseen koodaamiseen asti. Yrityksen tulee oman strategian ja resurssien mukaisesti tehdä päätös mihin asiaan se keskittyy.

ERP-järjestelmät ovat laajoja kokonaisuuksia kuten myös Odoo. Sen halluunotto ei onnistu vain yhdeltä tai kahdelta henkilöltä, vaan siinä tulee jakaa osaamista henkilöiden ja tiimien kesken. Toimeksiantajayrityksen nykyinen osaaminen on vahvasti myynnin ympärillä toimivien ohjelmistojen ympärillä, joita Odoossa ovat muun muassa CRM, kassa, myynti, varasto ja kirjanpito moduulit. Näiden ja muutaman näihin linkittyvän moduulin syvällinen osaaminen olisi varmasti parempi, kuin koko ohjelmiston osaaminen päällisin puolin.

Moduulien ohjelmoinnin näkökulmasta näiden tehtyjen töiden perusteella voidaan todeta, että Odoon ympärillä on valtavasti mahdollisuuksia. Perusmoduulien ohjelmointiin kykenee perusteet osaava, mutta jotta tästä saadaan oikeasti lisäarvoa loppuasiakkaalle, vaatii se panostusta ja keskittymistä koodin lisäksi myös asiakkaan toimintaympäristöön ja prosesseihin. Tässä työssä ehdittiin tutustumaan moduulien ohjelmointiin vasta pintapuolisesti ja varsinainen koodin kautta syntyvän ohjelmiston liiketoimintalogiikkaan ei vielä ehditty syventyä enempää. Lisäksi vuosittain julkaistavan versiopäivityksen johdosta joka vuosi jokin asia hieman muuttuu ja näissä muutoksissa ajan tasalla pysyminen vaatii aikaa sekä perehtymistä.

Liiketoiminnan näkökulmaa on tutkittu tämän opinnäytetyön osalta melko suppeasti. Pääpaino työssä oli selvittää moduuleja teknisestä näkökulmasta ja liiketoiminnallista puolta onkin käyty läpi enemmän niiltä osin, mitä teknisen osuuden puolelta on tullut ilmi. Työssä käytyjen asioiden pohjalta nousi kuitenkin kolme erilaista näkökulmaa liiketoiminnalliseen osuuteen. Ensimmäinen näkökulma on keskittyminen Odooseen projekti-johtamisen ja asiakkaan liiketoiminnan kehittämisen kannalta. Tällöin voidaan myydä omaa asiantuntijapalvelua, jossa Odoo on yksi osa liiketoiminnan kehittämistä, sillä ERP on yrityksen tärkeimpiä työkaluja. Odoon räätälöintiä voidaan toteuttaa pienissä osin esimerkiksi Odoo Studiolla tai mahdollisesti ostaa erillisenä projektina toiselta yritykseltä, joka on keskittynyt Odoon räätälöintiin.

Toinen näkökulma on keskittyä nimenomaan Odoon räätälöintiin, jolloin myytävä palvelu on lähinnä koodaustyötä. Tämä vaatii taas hyvin erilaista osaamista ja resursseja, kuin ensimmäinen näkökulma. Tässä työssä ei ehditty vielä perehtymään versioylläpitoon, joka on huomioonotettava asia nopeasti kehittyvässä ohjelmistossa. Eli vaikka moduuleja saataisiin asiakkaille tiuhaan tahtiin ulos, toinen asia on, miten vanhojen moduulien päivitys järjestetään systemaattisesti. Kaikkia asiakkuuksia ei ole syytä päivittää vuosittain ilman validia syytä, mutta viimeistään tämä tulee eteen, kun päivitykset lakkaavat Odoon puolelta kyseiseen versioon.

Kolmas näkökulma on näiden kahden välimuoto. Tässä keskityttäisiin enemmän asiakkaan liiketoiminnan kehittämiseen Odoon avulla ja moduuleja tehtäisiin kaikkein kriittisimpiin tarkoituksiin. Moduuleista tehtäisiin tuotteistettuja paketteja, jolloin ylläpito ja versiopäivitys olisivat helpommin hallittavissa.

Tässä työssä on keskitytty Odoon toiminnallisen puolen räätälöintiin moduulien avulla ja sen liiketoimintamahdollisuuksiin. ERP-järjestelmään kuuluu myös oleellisesti muun muassa palvelimen ylläpito, sekä mahdolliset integraatiot muihin järjestelmiin, mutta nämä asiat on jätetty tämän työn ulkopuolelle. Lisäksi työn teoriaosuudessa on käyty läpi Odoon raportointia, mutta sitä ei ole tutkittu tarkemmin työn käytännönsuudessa.

7 YHTEENVETO

Tässä oppinnäytetyössä oli kaksi tutkimuskysymystä, jotka ovat:

- Miten moduuleita rakennetaan Odooseen?
- Miten moduuleista saadaan SV Maksujärjestelmien liiketoimintaa?

Moduuleja voidaan tehdä Odooseen muutamalla eri tavalla. Odoon Enterprise-versioon on saatavilla Odo Studio -työkalu, jolla voidaan tehdä yksittäisiä muutoksia tai kokonaisia moduuleita suoraan käyttöliittymässä. Tämä on käyttäjäystävällinen tapa, jolloin käyttäjän ei tarvitse välttämättä ymmärtää koodista mitään, mutta ymmärrys ohjelmiston toiminnasta ja logiikasta on lähes välttämätön, jotta asiat tehdään oikein. Tämä on myös erittäin nopea tapa tehdä joitakin uusia toiminnallisuuksia ohjelmistoon ja ne ovat heti käytettävissä. Tämä antaa kuitenkin hyvin rajalliset mahdollisuudet mitä kyetään tekemään.

Toinen vaihtoehto on ohjelmoida moduuli alusta alkaen, jolloin puhutaan jo jonkinlaisesta ohjelmistoprojektista. Odo on itsessään jo todella laaja ohjelmisto ja sen toiminnan ymmärtämiseen niin liiketoiminnan kuin koodin näkökulmasta vaatii aikaa jo isommallekin organisaatiolle. Moduulin ohjelmoinnin näkökulmasta perusteiden opettelu ja selvitys vei jonkin verran aikaa. Minkään monimutkaisemman toiminnon tekeminen taas vaatisi paljon enemmän aikaa ja perehtymistä Odoon logiikkaan. Tällöin voidaan vasta sanoa, että moduulista on oikeasti lisäarvoa yritykselle.

Odoon ympärille on mahdollista luoda liiketoimintaa erilaisista näkökulmista. Tällaista ohjelmistoa jälleenmyydessä organisaatiossa on oltava myös teknistä osaamista, keskittyipä enemmän konsultointiin tai koodaamiseen. Yhteenvetona toimeksiantajayritykselle voidaan kuitenkin todeta, että keskittyminen tiettyihin asioihin Odoon ympärillä on suositeltavaa, sillä mahdollisuuksia Odoon ympärillä on paljon liiketoiminnan konsultoinnista ohjelmistoprojekteihin ja koulutuksiin.

LÄHTEET

COSS ry. (2019). Avoin lähdekoodi - COSS.fi. Haettu 21.9.2019 osoitteesta: <https://coss.fi/avoimuus/avoin-lahdekoodi/>

Cybrosys Technologies. (2018). A Beginners guide to Odoo (Cybrosys Technologies) by CybrosysTechnologies - issuu. Haettu 16.9.2019 osoitteesta: <https://issuu.com/cybrosys2016/docs/odoo-book-by-cybrosys-technologies>

Dony, O. (2014). Improving the performance of Odoo deployments. Haettu 9.9.2019 osoitteesta: <https://www.slideshare.net/openobject/performance2014-35689113>

Ijas. (2019). DEVELOPER MODE IN ODOO | Activating Developer Mode- Technaureus. Haettu 1.10.2019 osoitteesta: <https://www.technaureus.com/developer-mode-in-odoo/>

Linux.fi-wiki. (2015). GNU LGPL – Linux.fi. Haettu 22.9.2019 osoitteesta: https://www.linux.fi/wiki/GNU_LGPL

Linux.fi-wiki. (2017). GNU GPL – Linux.fi. Haettu 22.9.2019 osoitteesta: https://www.linux.fi/wiki/GNU_GPL

Logistiikan maailma. (2019). Toiminnanohjausjärjestelmä – Logistiikan Maailma. Haettu 21.9.2019 osoitteesta: <http://www.logistiikanmaailma.fi/logistiikka/ohjausjarjestelmat/toiminnanohjausjarjestelma/>

Makai, M. (2012). Object-relational Mappers (ORMs) - Full Stack Python. Haettu 24.9.2019 osoitteesta: <https://www.fullstackpython.com/object-relational-mappers-orms.html>

Nizarudheen, M. (2018). MVC Architecture of Odoo | MVC in Odoo | Odoo Architecture- Technaureus. Haettu 9.9.2019 osoitteesta: <https://www.technaureus.com/mvc-architecture-of-odoo/>

Nurmi, R. (2018). Odoo | ite wiki digitalisoinnin opas. Haettu 8.9.2019 osoitteesta: <https://www.itewiki.fi/opas/odoo/>

Odoo S.A. (2013). Architecture. Haettu 22.9.2019 osoitteesta: https://doc.odoo.com/trunk/server/02_architecture

Odoo S.A. (2018a). Become a Partner | Odoo. Haettu 7.10.2019 osoitteesta: <https://www.odoo.com/become-a-partner>

Odoo S.A. (2018b). Building a Module — odoo 12.0 documentation. Haettu 23.9.2019 osoitteesta:
<https://www.odoo.com/documentation/12.0/howtos/backend.html>

Odoo S.A. (2018c). Licenses — Odoo 12.0 documentation. Haettu 22.9.2019 osoitteesta:
<https://www.odoo.com/documentation/user/12.0/legal/licenses/licenses.html>

Odoo S.A. (2018d). Module Manifests — odoo 12.0 documentation. Haettu 23.9.2019 osoitteesta:
<https://www.odoo.com/documentation/12.0/reference/module.html#reference-module-manifest>

Odoo S.A. (2018e). Odoo.sh — Odoo 12.0 documentation. Haettu 10.12.2019 osoitteesta:
https://www.odoo.com/documentation/user/12.0/odoo_sh/documentation.html

Odoo S.A. (2018f). Odoo Guidelines — odoo 12.0 documentation. Haettu 23.9.2019 osoitteesta:
<https://www.odoo.com/documentation/12.0/reference/guidelines.html>

Odoo S.A. (2018g). QWeb Reports — odoo 12.0 documentation. Haettu 4.12.2019 osoitteesta:
<https://www.odoo.com/documentation/12.0/reference/reports.html>

Odoo S.A. (2019a). Create & Customize Your Own Apps with Odoo Studio. Haettu 30.9.2019 osoitteesta:
<https://www.odoo.com/page/studio>

Odoo S.A. (2019b). Implementation Methodology.

Odoo S.A. (2019c). Odoo Apps FAQ | Odoo Apps. Haettu 22.9.2019 osoitteesta: <https://www.odoo.com/apps/faq>

Odoo S.A. (2019d). Odoo Enterprise vs Community | Odoo Editions Comparison. Haettu 8.9.2019 osoitteesta:
<https://www.odoo.com/page/editions>

Odoo S.A. (2019e). Odoo Pricing | Odoo. Haettu 8.9.2019 osoitteesta:
<https://www.odoo.com/pricing#>

Odoo S.A. (2019). Building a Module — odoo 12.0 documentation. Haettu 21.9.2019 osoitteesta:
<https://www.odoo.com/documentation/12.0/howtos/backend.html>

Odoo tricks and tips (2019). Hettu 30.12.2019 osoitteesta:
<https://odootricks.tips/about/building-blocks/odoo-view-types/>

Pinckaers, F. (2013). The Odoo story. Haettu 8.9.2019 osoitteesta: <https://www.odoo.com/blog/odoo-news-5/post/the-odoo-story-56>

PT, S. (2018). MVC Architecture & MVC in Odoo ERP | LinkedIn. Haettu 9.9.2019 osoitteesta: <https://www.linkedin.com/pulse/mvc-architecture-odoo-erp-shereef-pt/>

Rashid, M. A., Hossain, L., & Patrick, J. D. (2002). The Evolution of ERPSystems: A HistoricalPerspective. Haettu 8.9.2019 osoitteesta: <https://faculty.biu.ac.il/~shnaidh/zooloo/nihul/evolution.pdf>

Reis, D. (2018). Odoo 12 Development Essentials - Fourth Edition. Haettu 30.9.2019 osoitteesta: <https://subscription.packtpub.com/book/business/9781789532470/1>

Rouse, M. (2019). What is ERP (Enterprise Resource Planning) and Why is It Important? Haettu 8.9.2019 osoitteesta: <https://searcherp.techtarget.com/definition/ERP-enterprise-resource-planning>

Yasir, M. (2018). Odoo 12 Features | Features Of Odoo 12 | Odoo 12- Technaureus. Haettu 22.9.2019 osoitteesta: <https://www.technaureus.com/odoo-12-new-features/>