

Antti Rautapää

**AVOIMEN LÄHDEKODIN KAUPALLINEN  
HYÖDYNTÄMINEN**



JYVÄSKYLÄN YLIOPISTO  
TIETOJENKÄSITTELYTIETEIDEN LAITOS  
2015

## TIIVISTELMÄ

Rautapää, Antti

Avoimen lähdekoodin kaupallinen hyödyttäminen

Jyväskylän yliopisto, 2015, 22 s.

Tietojärjestelmätiede, kandidaatintutkielma

Ohjaaja(t): Makkonen, Pekka

Avoin lähdekoodi yleistyy IT-alalla. Avointa lähdekoodia on hyödynnetty aikojen saatossa eri lisenssien varjolla eri tavoin: toiset yritykset voivat julkaista tuotteen kahdella lisenssillä mahdollistaen avoimuuden ja samalla kaupallistamisen, toiset yritykset voivat sallivan lisenssin alla lisensoidun tuotteen ottaa itselleen pohjaratkaisuksi.

Tässä tutkielmassa tutkitaan kirjallisuuskatsauksen avulla, kuinka eri tekijät vaikuttavat lisenssivalintoihin mikäli avoimen lähdekoodin ratkaisua halutaan hyödyntää kaupallisesti. Monen pienen yrityksen kohdalla on järkevämpää löytää valmis ratkaisu, jota jatkokehittää, kuin aloittaa kehittäminen yrityksen kohdalta puhtaalta pöydältä. Tutkielmassa tutkitaan aihetta niin liiketoimintamallien, tuotteeseen liittyvien tekijöiden kuin ohjelmistolisenssien näkökulmasta.

Tutkimuksen tarkoituksena on selvittää millaiset vaikutukset on erilaisilla lisenssivalinnoilla kaupallisen hyödyntämisen mahdollisuuksiin kuin eri tekijöiden vaikutuksia lisenssivalinnan päätökseen. Tutkimuksen olettamuksena on se, että avoimen lähdekoodin kaupalliseen hyödyntämiseen on verrattava yrityksen liiketoimintamallia ennen kuin avoimen lähdekoodin ratkaisu otetaan yritykseen käyttöön tai jatkokehitettäväksi. Tutkimuksen tulokseksi saatiin se, että niin yrityksen koko kuin olemassa oleva liiketoimintamalli vaikuttavat mahdollisiin lisenssivalintoihin. Lisenssivalinnalla taas on suuri merkitys kuinka avoimen lähdekoodin ratkaisua voidaan yrityksessä hyödyntää.

Asiasanat: Avoin lähdekoodi, liiketoimintamalli, ohjelmistolisenssi, lisenssin valinta, GPL

## ABSTRACT

Rautapää, Antti

Exploiting of open source in commercial aspect

Jyväskylä: University of Jyväskylä, 2015, 22 p.

Information Systems, Bachelor's Thesis

Supervisor(s): Makkonen, Pekka

Open source is getting more common as a term in the IT business. IT companies have exploited open source in many ways: some can publish their software product with two licenses, keeping the product with limited functionality open and attract people to buy the propriety licensed product. Other companies can exploit permissive licenses and build their commercial product on an open source solution.

In this study the used research method is literature review. The main goal is to resolve factors that effect on the license choice if the open source solution is wanted to be exploited in commercial aspect. In many small-sized companies it is wiser to find an existing solution to be developed than start developing a new product from a scratch. In this study the subject is examined from business model, product-based factor and software license aspect.

The main point of this study is to resolve how license choice can have an effect on possibilities of exploiting open source solutions in commercial aspect. As well it is studied how different kind of factors effect on license decision. Hypothesis of this study is that company has to check their business model before company picks up an open source solution and starts to develop it further. Result of this study is that business model and external factors like company size have an effect on license choice. License choice has a great effect on exploiting the open source solution in commercial aspect.

Keywords: Open source, business model, software license, license choice, GPL

## SISÄLLYS

1	JOHDANTO.....	5
2	AVOIN LÄHDEKOODI.....	7
2.1	Avoimen lähdekoodin ohjelmistolisenssit.....	8
2.1.1	Rajoittavat lisenssit.....	8
2.1.2	Sallivat lisenssit.....	9
2.2	Avoimen ja suljetun lähdekoodin ohjelmistojen kehitysprosessien erot.....	10
2.2.1	Laadunhallinta.....	10
2.2.2	Julkaisu.....	11
3	LIIKETOIMINTAMALLIT.....	12
3.1	Perinteiset liiketoimintamallit.....	12
3.2	Avoimen lähdekoodin liiketoimintamallit.....	13
4	AVOIMEN LÄHDEKOODIN KAUPALLINEN HYÖDYNTÄMINEN ....	15
4.1	Lisenssin valintaan vaikuttavat tekijät.....	16
4.2	Liiketoimintamallin vaikutus avoimen lähdekoodin lisenssin valintaan.....	17
5	YHTEENVETO.....	19
	LÄHTEET.....	21

# 1 JOHDANTO

Avoin lähdekoodi on terminä esillä monessa paikassa IT-alalla. Monille yrityksille, varsinkin pienemmille, on helpompaa valita jatkokehitettäväksi avoimen lähdekoodin valmis ratkaisu kuin kehittää ohjelmisto alusta alkaen itse (Lindman, Rossi & Puustell, 2011). Avoimen lähdekoodin ratkaisuja pyritään hyödyntämään omiin liiketoimintamalleihin peilaten. Yrityksen on yleensä tärkeätä miettiä ratkaisua sen lisenssin kannalta. Väärällä lisenssivalinnalla voi olla yritystoiminnan kannalta kohtalokkaita vaikutuksia.

Kandidaatintutkielmassa käsitellään, mitä tulee ottaa huomioon mikäli halutaan hyödyntää avointa lähdekoodia yrityksen omassa liiketoiminnassaan. Avointa lähdekoodia on aiheena tutkittu jo vuosien ajan, mutta suomenkielistä tutkimusta ei aiheesta ole. Avointa lähdekoodia kaupallisena mahdollisuutena on jonkin verran tutkittu englanninkielellä. Tässä kandidaatintutkielmassa vastaan seuraaviin tutkimuskysymyksiin:

- Miten avointa lähdekoodia voidaan hyödyntää kaupallisesti?
- Miten eri tekijät vaikuttavat lisenssivalintapäätökseen?

Tutkimuskysymyksiin vastataan perehtymällä ensiksi käsitteisiin avoin lähdekoodi sekä liiketoimintamalli. Näiden käsitteiden perusteella voidaan pohtia kuinka avointa lähdekoodia voidaan hyödyntää sovittaen erilaisiin ohjelmistoalan liiketoimintamalleihin.

Tutkielma on tehty kirjallisuuskatsauksena. Lähteenä on käytetty avointa lähdekoodia tutkivia tieteellisiä artikkeleita sekä väitöskirjoja. Tiedonhankinnassa on pääasiallisesti käytetty IEEE:n Software ja Empirical Software Engineering -lehtiä sekä Googlen Scholar-tietokantaa. Aihe on ollut kauan tutkimuksen kohteena, mutta se ei ole silti kovin suosittu tutkimuskohde IT-alan keskuudessa. Tuoreimmat lähteet ovat vuodelta 2012 ja vanhimmat vuodelta 1999.

Tutkielman ensimmäinen luku käsittelee avointa lähdekoodia yleisesti ja sen tarkoitus on avata avoimen lähdekoodin syntyä ja historiaa, sekä liiketoiminnalliseen näkökulmaan liittyen avoimen lähdekoodin erilaisia lisenssejä. Toisessa pääluvussa käsitellään liiketoimintamalleja yleisellä tasolla ja vertaillaan liiketoimintamallien eroja puhtaasti kaupallisen alan ja avoimen

lähdekoodin osalta. Määritelmien selventämisen jälkeen tutkitaan kuinka eri tekijät vaikuttavat lisenssien valintaan ja lopuksi miten liiketoimintamalli vaikuttaa lisenssien valintaan.

## 2 AVOIN LÄHDEKOODI

Avointa lähdekoodia käytetään sovellusmarkkinoilla yleisesti. Avoimen lähdekoodin yleisyydelle on monta syytä. Vapaasti levitettävät ilmaiset ohjelmat mahdollistavat käyttöönoton helppouden. Avoimen lähdekoodin ohjelmia voidaan hyödyntää omaan käyttöön ohjelmien räätälöintimahdollisuuksien takia. Myös yritykset voivat hyödyntää ohjelmia ja ratkaisuja kaupallisesti julkaisemalla niitä omista nimissään ja omilla lisensseillään. (Khanjani & Sulaiman, 2011).

Avoimen lähdekoodin merkittävä merkkipaalu on vuodelta 1974, jolloin BSD (Berkeley Software Distribution) - UNIX-järjestelmän eri distribuutiot julkaistiin. Avoin lähdekoodi sai täten alkunsa, vaikkei sitä käsitteenä tunnettu (Bretthauer, 2001). Avoin lähdekoodi on elänyt käsitteenä vuodesta 1984 Richard Stallmanin perustaessa Free Software Foundationin rahoittaakseen GNU Projectia. (Wu & Lin, 2001). Free Software Foundationin avulla oli tarkoitus varmistaa sovelluskehityksen rahoitus, jotta kehittäjät eivät joutuisi kehittämään ohjelmistoja ilmaiseksi, mahdollistaen loppukäyttäjälle maksuttoman sovelluksen käyttöönoton (Wu & Lin, 2001). Stallmanin visiona oli mahdollistaa ohjelmistojen käyttö ja räätälöinti omiin tarpeisiin kuin myös räätälöintien levitys eteenpäin omina distribuutioinaan. Tunnetuin avoimen lähdekoodin tuote suurelle yleisölle on Linus Torvaldsin alulle saattama UNIX-pohjainen Linux-käyttöjärjestelmä. (Wu & Lin, 2001). Linux on Torvaldsin omana projektina alkanut järjestelmäprojekti. Linux sai 1990-luvun alkupuolella useita maineikkaita hahmoja taakseen uusiin distribuutioprojekteihin kuten 1993 Ian Murdockin aloittaman Debian Linuxin ja 1994 Mark Ewingin Red Hat Linuxin muodossa. (Hars & Ou, 2001) Linux kasvatti suosiotaan 1990-luvulla: vuonna 1999 7,5 miljoonaa ihmistä käytti Linuxia (Hars & Ou, 2001).

Avoimen lähdekoodin tärkein periaate on läpinäkyvyys. Esimerkiksi avoimen lähdekoodin järjestelmän lähdekoodia on päästävää katsomaan käyttäjän toimesta. Mikäli tuote on lisensoitu avoimen lähdekoodin lisensseillä, on järjestelmän sisässä tapahtuvia prosesseja päästävää muokkaamaan käyttäjän toimesta omiin tarpeisiinsa (Godfrey & Tu, 2000). Avoimen lähdekoodin vastakohtana on suljettu lähdekoodi. Suljetun lähdekoodin ohjelmisto on

ohjelmisto, joka on lisensoitu yksinoikeudella ohjelmiston omistajan taholle. (Khanjani & Sulaiman, 2011). Siinä missä avoimelle lähdekoodille on tärkeintä vapaa levitettävyyttä ja muokattavuutta, suljetussa lähdekoodissa on taustalla yleensä kaupalliset intressit. Suljetun lähdekoodin ratkaisussa pyritään tuomaan ratkaisun omistajalle taloudellista hyötyä. (Khanjani & Sulaiman, 2011). Avoimen lähdekoodin ohjelmiston on avoimen lähdekoodin etujärjestön OSI:n (Open Source Initiative) sertifioiman lisenssin saadakseen täytettävä edellä mainittujen kriteerien lisäksi myös seuraavat kriteerit: avoimen lähdekoodin ohjelmiston jakelussa ei saa tapahtua ryhmä- tai ihmiskohtaista syrjintää, lisenssi ei saa olla tuotekohtainen eikä myöskään estää muiden tuotteiden jakelua tai toimintaa ja ohjelmistolisenssin täytyy olla myöskin riippumaton käytettävästä teknologiasta. (Ueda, 2004).

## **2.1 Avoimen lähdekoodin ohjelmistolisenssit**

Ohjelmistolisenssi on lainmukainen sopimus, jolla määritellään mitä ohjelmistotuotteella voidaan tehdä (Lindman, Paajanen & Rossi, 2010). Avoimen lähdekoodin ohjelmistolisenssit voidaan jakaa kahteen vastakkaiseen kategoriaan OSI:n sertifioimina: rajoittaviin lisensseihin (*restrictive licenses*) sekä salliviin lisensseihin (*permissive licenses*) kaupallisessa aspektissa (Lindman ym., 2010).

### **2.1.1 Rajoittavat lisenssit**

Avoimen lähdekoodin rajoittava lisenssi on ohjelmistolisenssi, joka ei mahdollista kaupallista julkaisua modifioitavalla lopputuotteelle. Avoimen lähdekoodin lopputuotteen on oltava myös avointa lähdekoodia kuten alkuperäisen ohjelmiston tai tuotteen, joka on kehityksen kohteena. (Engelfriet, 2010).

GPL (GNU General Public License) on yleisin avoimen lähdekoodin rajoittava-tyyppinen ohjelmistolisenssi. GPL on vuorovaikutteinen ja rajoittava ohjelmistolisenssi, joka velvoittaa muokatun ohjelmistotuotteen julkaisun saman lisenssiehdon alla. Richard Stallman lanseerasi GPL:n perustettua Free Software Foundationin 1980 -luvulla (German, Di Penta & Davies, 2010). GPL on joissain tapauksissa todettu liian rajoittavaksi, joten joitakin poikkeuksia GPL -lisenssit yhteyteen on tehty. Esimerkiksi Perl -ohjelmointikieli oli alun perin lisensoitu GPL-lisenssillä, jonka myöhemmin huomattiin estävän kaupallisen Perl-sovelluksen kehityksen. Perlin versiossa 4.0 lanseerattiin sallivampi lisenssi nimeltään Artistic License, joka mahdollisti kaupallisten Perl-sovellusten kehityksen. (Lerner & Tirole, 2003). Myös käyttöjärjestelmien



kohdalla on jouduttu tekemään alkuperäiseen GPL:ään poikkeuksia. Esimerkiksi suurten internet-yhtiöiden kaupalliset palvelut ovat toimineet Linuxin päällä. Linux on kuuluisin yksittäinen GNU:lla lisensoitu tuote. (Välimäki, 2005).

LGPL (Lesser General Public License) on GPL:n johdannaislisenssi GPL:n rajoittavuuden johdosta. LGPL on GPL:ää sallivampi ohjelmistolisenssi, joka mahdollistaa kaupallisen jatkokehityksen, mutta suorat modifikaatiot avoimen lähdekoodin ohjelmistoon täytyy lisensoida LGPL tai GPL-lisenssillä. (Lindman, Rossi & Puustell, 2011). Esimerkiksi monet avoimen lähdekoodin ohjelmakomponenttikirjastot, jotka linkataan epäsuorasti kokonaiseen ohjelmistoon, eivät estä kaupallista julkaisua eri lisenssiehdon alla (Engelfriet, 2010).

MPL (Mozilla Public License) on NetScapen luoma ohjelmistolisenssi vuodelta 1998, kun Netscape päätti julkaista lähdekoodinsa selaimestaan (Lindman ym., 2011). MPL on myös GPL-johdannainen ohjelmistolisenssi, joka vaatii jatkokehityksessä syntyvän tuotteen lisensoinnin lähdekoodin välittämisen takaisin sovelluskehitysyhteisölle. MPL on suosittu lisenssiehto suuryritysten avustamien avoimen lähdekoodin sovelluskehitysprojektien osalta. (Lindman ym., 2011)

### 2.1.2 Sallivat lisenssit

Sallivat lisenssit ovat akateemisten tahojen kirjottamia ohjelmistolisenssejä kuten MIT:n ja BSD:n yliopistojen tuottamat lisenssit. Myös Apache Foundationin omat lisenssit kategorioidaan sallivien lisenssien joukkoon (Lindman ym., 2010). Salliva ohjelmistolisenssi mahdollistaa lopputuotteen kaupallistamisen. Ainoana vaatimuksena on antaa tunnustusta alkuperäisen tuotteen tekijälle. (Engelfriet, 2010).

BSD (Berkeley Software Distribution) - lisenssi on kaikkein sallivin kaupallisessa levityksessä. BSD on sekä vanhin että parhaiten tunnettu salliva lisenssi. BSD-lisenssi sallii muttei vaadi lähdekoodin julkaisemista. Jatkojulkaisu BSD-lisenssin alla voidaan tuoda myös binäärimuodossa ilman lähdekoodia. Kuitenkin BSD-lisenssissä on muutamia pieniä vaatimuksia lisensointiin ja tekijänoikeuksiin liittyen. (Lindman ym, 2011).

MIT:n (Massachusetts Institute of Technology) lisenssi on ehdoiltaan yksinkertaisin sallivista lisensseistä. MIT:n ohjelmistolisenssi muistuttaa yleisesti ottaen hyvin paljon BSD:n ohjelmistolisenssiä, mutta on BSD:tä yksinkertaisempi ja riisutumpi versio ja sopii näin kaupalliseen jatkokäyttöön. (Lindman ym, 2011).

Apachen ohjelmistolisenssi on Apache Foundationin luoma akateeminen avoimen lähdekoodin ohjelmistolisenssi. Apachen ohjelmistolisenssi sallii kaupallisen jatkokehityksen kuten muutkin akateemiset ohjelmistolisenssit, mutta Apachessa on yksi poikkeus MIT:n ja BSD:n lisensseihin verrattuna. Apachen lisenssin alla olevan tuotteen jatkokehityksen lopputuotos ei saa käyttää Apachen nimeä markkinoinnissa. (Lindman ym, 2011).

## **2.2 Avoimen ja suljetun lähdekoodin ohjelmistojen kehitysprosessien erot**

Avoimen lähdekoodin kehityksessä on merkittäviä eroja suljetun lähdekoodin kehittämiseen verrattuna. Avoimen lähdekoodin ratkaisun kehitys alkaa monesti yhden ihmisen visiosta. Tyypillisesti yhden ihmisen tarkoitus on kehittää ratkaisua omaan tarpeeseensa joko tyhjästä tai perustuen vanhaan ratkaisuun ja sen mahdollisiin puutteisiin. Esimerkkejä visioista ovat Linus Torvaldsin Linux-järjestelmä pohjautuen Minix-järjestelmään, sekä Bram Moolenaarin *stvie*-tekstieditori pohjautuen vanhaan *vi*-editoriin. (Godfrey & Tu, 2000). Avoimen lähdekoodin kehitysprosessi voi alkaa kehittäjän halusta luoda vastine suurten toimijoiden tuotteille esim. IBM:n ja Microsoftin kaltaisten toimijoiden tuotteiden rinnalle. (Khanjani & Sulaiman, 2011). Itse kehitysprosessin vaiheissa on merkittäviä eroja. Avoimen lähdekoodin järjestelmien aikataulujen pitävyys ja kaupallisen julkaisun paineet eivät ole niin suuressa osassa kuin kaupallisten ohjelmistojen toteutuksessa. (Khanjani & Sulaiman, 2011).

### **2.2.1 Laadunhallinta**

Laadunhallintaprosessit eroavat avoimen ja suljetun lähdekoodin ohjelmistojen kehityksissä. Khanjanin ja Sulaimanin (2011) mukaan avoimen lähdekoodin ohjelmistojen kehityksessä suunnittelu ja dokumentointi on epäorganisoidumpaa kuin suljetun lähdekoodin ohjelmistoissa. Koska suljetun lähdekoodin ohjelmistojen kehityksessä on mukana taloudelliset intressit ohjelmiston omistajataholla, projektia hallitaan asiantuntijatiimin voimin keskitetysti. (Khanjani & Sulaiman, 2011).

Khanjanin ja Sulaimanin (2011) mukaan avoimen lähdekoodin ohjelmiston laatu voidaan mitata eri tavoin kuin suljetun lähdekoodin ohjelmiston laatu. Siinä missä suljetun lähdekoodin ohjelmistossa laatu tarkoittaa projektinhallinnan onnistumista kuin vaatimusten toteutumista, avoimessa lähdekoodin ohjelmistoissa laatu merkitsee eri asioita: Avoimen lähdekoodin laatu tarkoittaa kehitysyhteisön omien intressien täyttymistä. (Khanjani & Sulaiman, 2011).

Avoimen lähdekoodin ohjelmistojen dokumentointi on usein sekavaa ja epäselvää riippuen projektinhallinnan tasosta. Koska avoimen lähdekoodin projektien hallinta on hyvin eroavaista verrattuna suljetun lähdekoodin projekteihin, dokumentaation taso vaihtelee projektien välillä. Avoimen lähdekoodin ohjelmistojen dokumentointi on usein vapaaehtoista ja tämän takia puutteellista. Projektien vastuuhenkilöiden puuttuessa dokumentointiin ei

kohdistu valvontaa ja täten dokumentoinnissa ei ole rakennetta. Suljetun lähdekoodin ohjelmistojen dokumentointi on usein projektinhallinnan resurssien johdosta paremmin valvottua ja rakenteeltaan järjestelmällisempää. (Khanjani & Sulaiman, 2011).

### **2.2.2 Julkaisu**

Avoimen ja suljetun lähdekoodin ohjelmistojen julkaisutavoissa on merkittäviä eroja. Avoimen lähdekoodin sovellusten julkaisuissa ei yleensä ole tarkkoja julkaisupäivämääriä vaan julkaisupäivä on silloin kuin projektin omistajat hyväksyvät ratkaisun olevan tarpeeksi vakaa ja hyvä julkaisua varten. Julkaisuissa on myös muita merkittäviä eroja. Suljetussa lähdekoodissa ei erillisiä versioita julkaista käyttäjille kokeiltaviksi. Avoimen lähdekoodin ratkaisu voidaan jakaa kokeellisiin (*unstable*) ja vakaisiin (*stable*) julkaisuversioihin. (Godfrey & Tu, 2000).

### 3 LIKETOIMINTAMALLIT

Liiketoimintamalli on taloudellinen laskelma, jossa otetaan huomioon asiakkaan tuomat kassavirrat kuin myös asiakkaiden miellyttämiseen tarkoitetut kustannukset. Liiketoimintamallia luodessa voidaan soveltaa viitekehystä, jonka avulla vastataan keskeisiin kysymyksiin: kuka maksaa, kenelle tuotetaan ja minkä takia. (Muselli, 2008).

Perinteisessä ohjelmistoalan liiketoiminnassa on useita erilaisia liiketoimintamalleja, joilla pyritään tuottamaan rahaa yrityksille. (Välimäki, 2003). Liiketoimintamallin tarkoitus on auttaa yritystä käsittämään rahansaannit strategisina valintoina ja viemään toiminnan kehitystä eteenpäin (Schief & Buxmaan, 2012).

#### 3.1 Perinteiset liiketoimintamallit

Perinteiset liiketoimintamallit voidaan Välimäen (2003) mukaan jakaa karkeasti kahteen eri kategoriaan: palveluorientoituneisiin liiketoimintamalleihin (*service oriented business model*) ja tuotepohjaisiin liiketoimintamalleihin (*product-based business model*).

Tyypillisimmät liiketoimintamallit ohjelmistoalalla ovat palveluorientoituneet liiketoimintamallit, joiden keskiössä ei ole itse ohjelmisto vaan esimerkiksi ohjelmointityö tai tukitoiminnot (Käkölä, 2003). Palveluiden myynti tapahtuu useimmiten myytynä ohjelmistoprojektina, jossa kaupataan ohjelmisto-osaaminen palveluna valmisohjelmiston sijaan. (Välimäki, 2003). Palveluorientoitunut malli Käkölän (2003) mukaan sisältää pienemmän riskin ja kestää suhdannevaihtelut paremmin syvempien asiakassuhteiden ansiosta. Palveluorientoitunut malli vetoaa niin asiakkaisiin kuin palveluntuottajiin, koska palvelu itsessään on hyödyntuottaja pelkän teknologian ja ohjelmiston sijaan. (Käkölä, 2003).

Perinteisen ohjelmistoalan toisissa liiketoimintamalleissa eli tuotepohjaisissa liiketoimintamalleissa voidaan projektien sijaan myydä kokonaista valmisohjelmistoa lisenssien avulla (Välimäki, 2003).

Tuotepohjaisia malleja ovat tuotteen julkaisuun pohjautuva malli *software publishing*, sekä kuukausittaisiin maksuihin pohjautuva *software subscription*. (Välimäki, 2003). Tuotepohjaisissa liiketoimintamalleissa tarkoituksena on myydä ohjelmistolisenssejä mahdollisimman monelle asiakkaalle ennen kilpailijoiden ilmaantumista markkinoille (Käkölä, 2003).

### 3.2 Avoimen lähdekoodin liiketoimintamallit

Avoimen lähdekoodin julkaisijatahon täytyy kaupallisessa aspektissa erityisesti huomioida, mistä mahdolliset tulot saadaan. (Muselli, 2008). Musellin (2008) mukaan liiketoimintamallit avoimen lähdekoodin hyödyntämisessä voidaan jakaa kahteen eri kategoriaan: *bundle* eli paketteina tai *dual licensing* eli kahden lisenssin strategialla. Kaikkein yleisin liiketoimintamalli avoimen lähdekoodin kaupallisessa hyödyntämisessä on palveluiden ja ohjelmistojen pakettimalli. Liiketoimintamallissa julkaistaan ohjelmisto avoimen lähdekoodin lisenssillä, mutta yritys pyrkii myymään palveluina avoimen lähdekoodin ohjelmiston käyttöön- ja integraatiopalveluita. (Muselli, 2008). Yritykset voivat myös eriyttää avoimen ja kaupallisen lisenssin ohjelmistot erilleen. Tässä liiketoimintamallissa perusohjelmisto julkaistaan avoimen lähdekoodin ohjelmistolisenssin alla, mikä täyttää normaalin käyttäjän perustarpeet. Yritys saa tässä tapauksessa rahavirtansa *advanced*-tason ohjelmistoista, joissa on enemmän toiminnallisuuksia parannettuina perustason ohjelmistoon verrattuna. (Muselli, 2008).

Tyypillisimmät avoimen lähdekoodin hyödyntämiseen perustuvat liiketoimintamallit perustuvat hyvin pitkälti perinteisiin liiketoimintamalleihin. Ohjelmistoyritykset tarjoavat tyypillisimmin palveluina avoimen lähdekoodin ratkaisun käyttöön- ja integrointipalveluita. (Hecker, 1999). *Support sellers* eli tuenmyynti-liiketoimintamallissa yritys tarjoaa koulutusta ja konsultointia avoimen lähdekoodin ratkaisua varten. (Hecker, 1999). Käkölän (2003) mukaan perinteisessä, puhtaasti kaupallisessa liiketoimintamallissa keskiössä ovat palvelut ja niiden myynnit. *Service Enabler* eli palvelunmahdollistamis-liiketoimintamallissa yritys pyrkii avoimen lähdekoodin ohjelmiston tarjonnalla saaman asiakasta integroitumaan valmiin kaupallisen järjestelmän kanssa, jolloin yritys voi myydä kaupallisen lisenssin. Tällaisessa tapauksessa avoimen lähdekoodin ohjelmisto on usein välikappaleena kassavirran tuottamisessa (Hecker, 1999). Välimäen (2003) mukaan yritykset voivat myös pitää ratkaisunsa ilmaisena, jolloin voidaan lisensoida ohjelmistotuote kahteen kertaan (*dual licensing*), jolloin avoimen lähdekoodin lisenssillä lisensoitu tuote on yleensä kevyempi ja pelkistetympi. Esimerkiksi MySQL AB lisensoi

tuotteensa kahteen kertaan, jossa pelkistetympi tietokantaratkaisu on ilmainen GPL-lisenssillä lisensoituna. MySQL Enterprise on yrityksille tarkoitettu tuote, joka sisältää täyden tuen ja dokumentaation tietokantatuotteeseen liittyen. (Välimäki, 2003).

## 4 AVOIMEN LÄHDEKOODIN KAUPALLINEN HYÖDYNTÄMINEN

Monet suuret ohjelmistoyritykset IBM:n, Novellin tai Googlen lailla ovat hyödyntäneet omassa liiketoiminnassaan avoimen lähdekoodin ratkaisuja. Esimerkiksi Novell ja Sun ovat rakentaneet omat liiketoimintamallinsa avoimen lähdekoodin ympärille, kun taas internet-jättien Facebookin ja Googlen ratkaisut pohjautuvat räätälöityinä avoimen lähdekoodin ratkaisuihin. (DeKoenigsberg, 2008). Avoimen lähdekoodin hyödyntäminen kaupallisesti ei myöskään rajoitu pelkästään ohjelmistotuotteeseen. Avoimen lähdekoodin osaamista voidaan hyödyntää myös tietotaidon perusteella. Suuret ohjelmistoyritykset, kuten Red Hat, arvostavat henkisen pääoman arvoa rekrytoinneissaan. (DeKoenigsberg, 2008).

Avoimen lähdekoodin kaupallinen hyödyntäminen on monessa tapauksessa monen eri tekijän yhteisvaikutusta. Kaupallisessa hyödyntämisessä yritys joutuu tutkimaan hyödyntämistapaa monesta eri näkökulmasta. Kaupallisen hyödyntämisen näkökulmaan vaikuttavat yrityksen koko sekä liiketoimintamalli, minkä kautta yritys valitsee avoimen lähdekoodin lisenssin. (Lindman ym., 2011).

Avointa lähdekoodia voidaan hyödyntää monilla tavoin. Pienten yritysten osalla avoimen lähdekoodin kaupallista hyödyntämistä siivittää tuki innovoinnissa. Pienen yrityksen ei tarvitse tuhlata niin paljon resurssejaan tutkimustyöhön. Kehittäjäyhteisön tuki auttaa pientä yritystä löytämään ongelmakohtia kehitettävästä ohjelmistosta. Suurten yritysten kohdalla taas arvostetaan avoimen lähdekoodin ohjelmistojen laatua. Kaiken kokoisissa yrityksissä arvostetaan IT-työntekijöiden ammattitaitoa, sillä avoimen lähdekoodin kehittäjäyhteisön kehittäjien kiinnostus alaa kohtaan on korkeampi kuin puhtaasti kaupallisten ohjelmistojen kehittäjien. (Bonaccorsi & Rossi, 2003).

## 4.1 Lisenssin valintaan vaikuttavat tekijät

Lindmanin ym. (2011) mukaan tekijöillä avoimen lähdekoodin kaupallisessa hyödyntämisessä tarkoitetaan kehitysyhteisön omia päätöksiä, kehittäjien motivaatiota, kehitysyhteisön johtajuutta sekä yrityksen omaa kokoa. Näillä yksittäisillä tekijöillä on usein vaikutusta lisenssivalintoihin kuten myös yrityksen liiketoimintamallilla (Lindman ym., 2011). Liiketoimintamalli on tekijänä suurempi kokonaisuus ja se käsitellään tutkielmassa omana alalukunaan.

Useimmiten avoimen lähdekoodin kehitysprojektit lähtevät liikkeelle ennen kuin yritykset pääsevät vaikuttamaan niihin. Noin 95 prosentissa projekteista lisenssivalinta on jo tehty (Lindman ym., 2011). Tämä voi tuottaa hyödyntävälle yritykselle useimmiten ongelmia esimerkiksi valitun GPL-lisenssin tapauksessa, jossa GPL-lisensoidun komponentin yhdistäminen järjestelmään vaatii koko järjestelmän lisensointia GPL-lisenssin alla. (Lerner & Tirole, 2003). Tällaisessa tapauksessa voi olla yrityksen kannalta hankalaa, mikäli tiettyä järjestelmänosaa yritetään sulauttaa valmiiseen olemassa olevaan asiakkaan järjestelmään, sillä asiakkaan valmis järjestelmä joudutaan lisensoimaan uudelleen GPL-lisenssillä (Lindman ym., 2011).

Kehittäjäyhteisön motivaatiolla on merkitystä ohjelmiston laatuun sekä lisenssin valintaan. Mikäli halutaan motivoida kehittäjiä entisestään, valitaan lisenssiksi GPL tai LGPL-lisenssi, jolloin viestitään kehittäjäyhteisölle halusta pitää ohjelmisto tai ohjelmistokomponentti avoimena. (Lindman ym., 2011). Yksittäistä ohjelmistokehittäjää tämä motivoi myös rekrytoinnissa, sillä usein avoimen lähdekoodin ohjelmistoprojekti toimii rekrytointikanavana yrityksen ja yhteisön välillä. Kehittäjän motivaation nousu tätä kautta heijastaa myös ohjelmistotuotteen laatuun. (Bonaccorsi & Rossi, 2003). Vahvasti rajoittavalla GPL-lisenssillä voidaan laajentaa kehitysyhteisöä laajemmaksi, sillä usein kehittäjän kiinnostus kasvaa mikäli loppukomponentti pysyy jatkossakin avoimena (Lindman ym., 2011).

Kehitysyhteisön johtamisella tarkoitetaan projektinhallintaan liittyviä seikkoja. Useimmiten isot yritykset haluavat vaikuttaa avoimen lähdekoodin projektiin, mikäli kyseinen projekti on heidän liiketoiminnallisesta näkökulmastaan tärkeä. (Lindman ym., 2011). Avoimen lähdekoodin projektien hallinnassa on riskinsä. Liiallisella kontrollilla on mahdollista haitallisesti vaikuttaa kehitysyhteisön motivaatioon. Pahimmassa tapauksessa tuotoksen laatu on huonoa ja projekti voi hajota useaan osaan. (Välimäki, 2005). Toisaalta sallivien ohjelmistolisenssien projekteissa vahva projektinhallinta on keskeisessä osassa. Sallivien lisenssien tapauksessa projektin aloittaja saa kunnian ohjelmiston tuottajana, vaikka kyseinen instituutio jättäytyisi pois ohjelmistoprojektin alkumetreillä. Sallivien lisenssien projektinhallinta on vahvassa roolissa myös siinä tapauksessa, mikäli jokin ulkopuolinen taho jatkokäyttää sallivan lisenssin alla julkaistua ohjelmistotuotetta omalla kaupallisella lisenssillään. (Lindman ym., 2011).



Yhtiön koolla on merkitystä lisenssin valintaan. Pienten yritysten motivaatio avoimen lähdekoodin hyödyntämiseen on usein innovaatioiden jakaminen ja valmiiden ratkaisujen jatkokäyttö, sillä pienellä yrityksellä harvoin on resursseja aloittaa kehittämistä alusta (Bonaccorsi & Rossi, 2003). Pienillä yrityksillä on useimmiten suuremmat riskit avoimen lähdekoodin ratkaisun jatkokäytössä johtuen erilaisista lisenssiratkaisuista. Toisaalta pienillä yrityksillä on samat kyvykkyydet avoimen lähdekoodin ratkaisujen hyödyntämiseen kuin suurilla yrityksillä. (Lindman ym., 2011). Suurilla yrityksillä on monesti myös suuremmat järjestelmät, joten palveluorientoitunut liiketoimintamalli on yleisempi kuin tuotepohjainen (Välimäki, 2005). Suurten yritysten osalta lisenssivalinnalla ei ole niin suurta merkitystä kuin pienemmällä yrityksillä. Suuret yritykset lisenssien sijaan pyrkivät vaikuttamaan avoimen lähdekoodin kehitysyhteisöön hanakammin kuin pienemmät yritykset. (Lindman ym., 2011).

## **4.2 Liiketoimintamallin vaikutus avoimen lähdekoodin lisenssin valintaan**

Yrityksen liiketoimintamallilla on suurin yksittäinen vaikutus avoimen lähdekoodin ohjelmiston lisenssivalintaan. Palveluorientoituneessa liiketoimintamallissa ei tukipalveluiden myynnin takia ole suurta merkitystä lisenssivalintaan. (Lindman ym., 2011). Suurten ohjelmistoyritysten kohdalla yritys voi käyttää GPL-lisenssiä, mutta myydä tähän liittyviä tuki- ja konsultointipalveluita asiakkailleen, vaikka ohjelmistotuote on asiakkaalle maksuton. (Välimäki, 2005). Mikäli ohjelmistotuotetta halutaan saada sellaisenaan kaupaksi rajoittavan lisenssin varjolla, on ainoana vaihtoehtona kaksoislisensoinnin liiketoimintamalli (Välimäki, 2005). MySQL AB:n tapauksessa tietokantaratkaisu on lisensoitu kahteen otteeseen, jolloin edistyneempää MySQL Enterprise tuotetta pyritään myymään GPL-lisensoidun ilmaisversion vaillinaisuuden avulla (Välimäki, 2005). Muunlaisissa tuotepohjaisissa liiketoimintamalleissa rajoittavat lisenssit rajoittavat liikaa yrityksen liiketoimintaa (Lindman ym., 2011).

Tuotepohjaisissa liiketoimintamalleissa on suuressa roolissa oikean lisenssin valinta. Monesti valmiin komponentin ylläpidossa piilee omat liiketoiminnalliset riskinsä. Hyvillä päätöksillä voi toisaalta olla liiketoiminnallista hyötyä, mikäli avoimen lähdekoodin ratkaisu osataan valita oikeanlaisesti. (Lindman ym., 2011). Applen OS X-käyttöjärjestelmä on hyvä esimerkki akateemisen eli sallivan lisenssin jatkohyödyntämisestä. Applen OS X-käyttöjärjestelmä perustuu UNIX-pohjaiseen FreeBSD-käyttöjärjestelmään. Aikaisempi itsekehitetty käyttöjärjestelmäratkaisu hylättiin toimimattomana ja

Apple siirtyi BSD-lisensoidun käyttöjärjestelmän hyödyntämiseen. (Välimäki, 2005). Apple otti kehitystyössään tietoisesti riskin, mutta se on myöhemmin kannattanut liiketoiminnallisesta näkökulmasta (Lindman ym., 2011). Vaikka Apple on hyödyntänyt BSD-lisenssin sallivuutta ja lisensoinut tuotteensa omilla ehdoillaan, Apple on toisaalta myös julkaissut avoimella lähdekoodilla käyttöjärjestelmänsä ytimen eli kernelin omalla avoimen lähdekoodin lisenssillään pitääkseen hyvät välit kehittäjäyhteisöihin. (Välimäki, 2005).

Lisenssien valintaa pitää liiketoimintamallin lisäksi peilata myös muiden tekijöiden perusteella. Joissain tapauksissa yhteisön lisenssivalinta ja yrityksen hätiköity päätös komponentin tai ohjelmiston valinnasta voi aiheuttaa sen, että pienehkö yritys joutuu miettimään omaa liiketoimintamalliaan uusiksi (Lindman ym., 2011). Engelfrietin (2010) mukaan palveluorientoitunut yritys voi esimerkiksi ehkäistä sallivien lisenssien riskit julkaisemalla ratkaisun rajoittavalla GPL-lisenssillä.

## 5 YHTEENVETO

Tämän tutkielman tarkoituksena oli selvittää, kuinka avointa lähdekoodia voidaan hyödyntää kaupallisessa mielessä. Tutkielman alussa perehdyttiin avoimeen lähdekoodiin yleisessä mielessä sekä liiketoimintamalleihin ohjelmistoalalla. Näillä tiedoilla pyrittiin vertailemaan liiketoimintamallien ja muiden tekijöiden vaikutusta avoimen lähdekoodin lisenssien valintaan.

Avoimen lähdekoodin kaupallisessa hyödyntämisessä on otettava huomioon monia erilaisia näkökulmia yrityksen kannalta. Avoimen lähdekoodin kaupallisessa hyödyntämisessä tärkeintä on oikean lisenssin valinta ja peilata sitä yrityksen liiketoimintamalliin. Lisenssien ehdot eivät välttämättä jokaiselle yritykselle sellaisenaan sovi vaan yrityksen johdon on mietittävä tarkkaan erilaisten tekijöiden vaikutus mahdolliseen lopulliseen ohjelmistotuotteeseen.

Lisenssinvalintapäätökseen vaikuttavat monet erilaiset tekijät. Tärkein yksittäinen tekijä on yrityksen liiketoimintamalli, johon eivät kaikki lisenssit välttämättä sovi. Esimerkiksi erittäin rajoittavalla GPL-lisenssillä lisensoitua tuotetta ei voi tuotepohjaisissa liiketoimintamalleissa, lukuunottamatta kaksoislisensointimallia, juurikaan hyödyntää, vaikka kyseessä olisi yksittäinen komponentti järjestelmään. Rajoittava lisenssi rajoittaa myös muiden järjestelmän osien lisensointia. Yksittäinen GPL-lisensoitu komponentti tekee muutamia poikkeuksia lukuunottamatta koko siihen integroidusta järjestelmästä GPL-lisensoidun järjestelmän, jonka lähdekoodi on annettava maksutta.

Muita tekijöitä liiketoimintamallin lisäksi ovat yrityksen koko kuin myös kehitysyhteisön omat intressit. Yrityksen koolla voidaan vaikuttaa niin liiketoimintamalliin kuin suoraan lisenssin valintaan. Esimerkiksi pienen yrityksen on hankala tuottaa palveluita avoimen lähdekoodin ohjelmistoon pohjautuen vaan pieni yritys voi itse tuottaa avoimeen lähdekoodiin pohjautuvan tuotteen ja sitä kautta palveluita. Suuren ohjelmistoyrityksen kohdalla on helpompi tuottaa palveluita avoimen lähdekoodin ohjelmiston ympärille, koska yritys on todennäköisemmin ollut vaikuttamassa avoimen lähdekoodin ohjelmiston kehitykseen.

On olemassa muutamia erilaisia esimerkkiyrityksiä, joiden menestys on perustunut juurikin avoimen lähdekoodin hyödyntämiseen. Applen tapauksessa OS X -käyttöjärjestelmä perustuu FreeBSD- sallivan lisenssin ratkaisuun. Toinen esimerkki on MySQL (nykyään Oraclen ohjelmistotuote), jonka menestys on perustunut kaksoislisensointiin. Toinen ohjelmisto on samaan aikaan lisensoitu GPL-lisenssillä ja toinen hieman laajempi ohjelmisto kaupallisella lisenssillä.

Yhteenvetona voi todeta, että yritykset voivat avoimen lähdekoodin kanssa luoda menestystarinoita. Samalla myös yritykset ottavat tietoisia riskejä ja väärillä lisenssivalinnoilla voidaan aiheuttaa yrityskonkursseja varsinkin pienten yritysten tapauksissa. Monessa tapauksessa on kuitenkin järkevää jatkaa kehitystä avoimen lähdekoodin ratkaisun kanssa ja räätälöidä se omaan liiketoiminnalliseen tarpeeseen kuin kehittää ohjelmistoa alusta alkaen.

Jatkotutkimuksena voitaisiin tutkia esimerkiksi miten pienet ja keskisuuret yritykset hyödyntävät avoimen lähdekoodin ratkaisuja ja kuinka erilaisia lisenssejä tällaisessa yrityksessä hyödynnetään.

## LÄHTEET

- Bonaccorsi, A., Rossi, C. (2003). Why Open Source Software Can Succeed. *Research Policy*, 32(7), 1243-1253.
- Bretthauer, D. (2001). Open Source Software : A History. *UConn Libraries Published Works : Paper 7*.
- DeKoenigsberg, G. (2008). How Successful Open Source Projects Work, and How and Why to Introduce Students to the Open Source World. *IEEE Software Engineering Education and Training*. 274-276.
- Engelfriet, A. (2010). Choosing an Open Source License. *IEEE Software*, 27(1). 48-49.
- German, D., Di Penta, M. & Davies, J. (2010). Understanding and Auditing the Licensing of Open Source Software Distributions. *IEEE Program Comprehension*, 84-93.
- Godfrey, M. & Tu, Q. (2000). Evolution in open source software : a case study. *IEEE Software Maintenance*, 131-142.
- Hars, A. & Ou, S. (2001). Working for Free? Motivations of participating in open source projects. *IEEE System Sciences*.
- Hecker, F. (1999). Setting Up Shop: The Business of Open-Source Software. *IEEE Software*, 16(1), 45-51.
- Khanjani, A. & Sulaiman, R. (2011). The Aspects of Choosing Open Source versus Closed Source. *IEEE Symposiums on Computers & Informatics*, 646 - 649.
- Käkölä, T. (2003). Software business models and contexts for software innovation: key areas software business research. *IEEE System Sciences*.
- Lerner, J. & Tirole, J. (2003). Some Simple Economics of Open Source. *The Journal of Industrial Economics*, 50(2), 197-234.
- Lindman, J., Paajanen, A. & Rossi, M. (2010). Choosing an Open Source Software License in Commercial Context: A Managerial perspective. *IEEE Software Engineering and Advanced Applications*, 237-244
- Lindman, J., Rossi, M. & Puustell, A. (2011). Matching Open Source Software Licences with Corresponding Business Models. *IEEE Software*, 28(4). 31-35
- Muselli, L. (2008). Open source software publishers' business models: the strategic role of licenses. *IEEE Management of Innovation and Technology*. 1169-1174.
- Schief, M. & Buxmann, P. (2012). Business Models in the Software Industry. *IEEE System Science*. 3328-3337.
- Ueda, M. (2005). Licenses of Open Source Software and their Economic Values. *IEEE Applications and the Internet Workshops*, 381-383
- Välimäki, M. (2003). Dual Licensing in Open Source Software Industry. *Systemes d'Information et Management*, 8(1), 63-75.

- Välimäki, M. (2005). The Rise of Open Source Licensing: A Challenge to the Use of the Intellectual Property in the Software Industry. *Helsinki University of Technology*.
- Wu, M. & Lin, Y. (2001). Open Source Software Development : An Overview. *IEEE Computer Society*, 34(6), 33-38.