Jasu Koponen

AVOIMEN LÄHDEKOODIN HYÖDYT OHJELMISTO-YRITYKSISSÄ

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2020

TIIVISTELMÄ

Koponen, Jasu
Avoimen lähdekoodin hyödyt ohjelmistoyrityksissä
Jyväskylä: Jyväskylän yliopisto, 2020, 24 s.
Tietojärjestelmätiede, kandidaatin tutkielma
Ohjaaja: Seppänen, Ville

Vapaat ja avoimen lähdekoodin ohjelmistot sekoitetaan helposti tavallisiin ilmaisohjelmistoihin. Vaikka avoimen lähdekoodin ohjelmistot ovat usein käyttäjilleen täysin ilmaisia, on niillä useita ominaispiirteitä jotka erottavat ne omisteisista ilmaisohjelmistoista. Toisin kuin markkinoita vallitsevat omisteiset ohjelmistot, jotka ovat usein tiukasti lisensoitu kaupallisiin tarkoituksiin, on avoimen lähdekoodin ohjelmistot lisensoitu kaikkien jaettavaksi, ladattavaksi ja muokattavaksi. Käyttäjillä on myös vapaa pääsy avoimen lähdekoodin ohjelmistojen lähdekoodiin, joka mahdollistaa ohjelmiston toiminnan tarkan tutkimisen. Avoin lähdekoodi on kehittynyt ilmiöksi, jonka kehitykseen osallistuvat myös suuret ohjelmistoyritykset, kuten Facebook, Google ja Microsoft. Avoin lähdekoodi on mahdollistanut useita ennenäkemättömiä ohjelmistokehitysmetodeja ja liiketoimintamalleja, joista voivat hyötyä jopa suuret ohjelmistoyritykset. Ohjelmistoyritykset voivat säästää kehitykseen, testaukseen ja ylläpitoon tarvittavia resursseja käyttämällä avoimen lähdekoodin ohjelmistoja ja ohjelmistokirjastoja. Jotkut ohjelmistoyritykset, kuten Red Hat, ovat jopa onnistuneet perustamaan liiketoimintansa täysin avoimen lähdekoodin ympärille tarjoamalla ammattimaista tukea ja ylläpitoa avoimen lähdekoodin ohjelmistokokonaisuuksille.

Asiasanat: avoin lähdekoodi, vapaat ohjelmistot, ohjelmistotuotanto, ohjelmistoliiketoiminta

ABSTRACT

Koponen Jasu
Advantages of open source in software organizations
Jyväskylä: University of Jyväskylä, 2020, 24 p.
Information Systems, Bachelor's thesis
Supervisor: Seppänen, Ville

Free and open source software is often misconceived as software that is merely free of charge. While such software is often provided without a fee, it does have many significant characteristics of its own. As opposed to the currently dominant proprietary software, which is strictly licensed with commercial interest in mind, open source software unveils its whole code base for everyone to inspect, modify and share as they see fit. Open source has grown to become a phenomenon that even large software organizations like Facebook, Google and Microsoft are contributing to. Open source has enabled new software development practices and business models that even large software organizations can benefit from. Software organizations can save resources in development, testing and maintenance by using freely available open source solutions. Some software organizations, such as Red Hat, have managed to establish a business revolving entirely around open source by offering professional support and maintenance for open source solutions.

Keywords: open-source, free software, software development, software business

KUVIOT

SISÄLLYS

1 JOHDANTO

Avoin lähdekoodi ei ole nykyään vain harrastelijoiden ja hakkereiden käytössä. Avoin lähdekoodi kasvattaa jatkuvasti suosiotaan suurissa yrityksissä ja avoimen lähdekoodin ohjelmistoja hyödynnetään entistä enemmän yritysten tietotekniikka- ja atk-osastoilla. (Watson, Boudreau, York, Greiner & Wynn, 2008). Myös suuret ohjelmistoyritykset ovat lähivuosina kasvattaneet kiinnostustaan avointa lähdekoodia kohtaan, ja avoimen lähdekoodin käyttö sekä tuotanto ovat yleistyneet ohjelmistoyrityksissä. Jopa suuret, omisteisista ohjelmistoistaan tunnetut, ohjelmistojätit, kuten Facebook, Google ja Microsoft, panostavat jatkuvasti enemmän avoimen lähdekoodin kehitykseen ja ylläpitävät useita suosittuja avoimen lähdekoodin projekteja, kuten Reactia, Flutteria ja VSCodea.

1.1 Aiheen kuvaus ja tutkimusongelma

Vaikka avointa lähdekoodia on usein pidetty vapaaehtoistyönä, Kolassan, Riehlen, Riemerin ja Schmidtin (2014) tutkimuksen mukaan avoimen lähdekoodin tuotannosta noin puolet on nykyään palkallista työtä. Ohjelmistoyritykset hyödyntävät projekteissaan avoimen lähdekoodin ohjelmistokomponentteja ja osallistuvat avoimen lähdekoodin kehitykseen (Ebert, 2008).

Avointa lähdekoodia pitkään suosivien ohjelmistoyritysten, esimerkiksi Red Hatin, lisäksi omisteisista ohjelmistoistaan tunnetut ohjelmistojätit, kuten Facebook, Google ja Microsoft, ovat 2010-luvulla alkaneet panostamaan avoimen lähdekoodin projekteihin. Ymmärtääksemme ohjelmistoyritysten kiinnostusta avoimesta lähdekoodista, tutkielmassa tutustutaan avoimen lähdekoodin keskeisimpiin käsitteisiin, kehitykseen sekä sen ohjelmistoyrityksille tarjoamiin hyötyihin ja mahdollisuuksiin seuraavien kysymyksien avulla.

- Mitä tarkoittaa avoin lähdekoodi ja miten se on muovautunut ilmiöksi, joka se on nykyään?
- Miten ohjelmistoyritykset voivat hyödyntää avointa lähdekoodia?

Seuraavassa luvussa käsitellään avoimen lähdekoodin historiaa, kehitystä ja ominaispiirteitä. Avoimen lähdekoodin kehitystä käsitellään kronologisessa järjestyksessä tärkeimpien projektien ja ilmiöiden avulla. Luvussa tutkitaan myös avoimen lähdekoodin ja omisteisten ohjelmistojen välisiä eroja, jotka korostavat ohjelmistoyritysten kohtaamaa muutosta avoimen lähdekoodin yleistyessä sekä helpottavat avoimen lähdekoodin hyötyjen tutkimista ohjelmistoyrityksissä.

Kolmannessa luvussa perehdytään avoimen lähdekoodiin hyödyntämiseen ohjelmistoyrityksissä. Luvussa keskitytään avoimen lähdekoodin hyödyntämiseen sekä ohjelmistokehityksessä että -liiketoiminnassa. Ohjelmistoyritykset voivat käyttää valmiita avoimen lähdekoodin ohjelmistoja ja ohjelmistokomponentteja ohjelmistokehityksessä, tai hyödyntää avointa lähdekoodia liiketoiminnassaan erilaisten lisenssien ja liiketoimintamallien avulla.

1.2 Tutkimusmenetelmä

Tutkielma on kirjallisuuskatsaus, jonka lähdekirjallisuus on löydetty suurimmaksi osaksi Google Scholarin avulla. Kirjallisuushauissa on käytetty hakusanoja ”open source”, ”open source software” sekä ”free software”. Avoimen lähdekoodin hyötyjä tutkiessa kirjallisuushakuja on jatkettu ohjelmistoyrityksille oleellisilla hakusanoilla, kuten ”product”, ”organization”, ”business”, ”development” sekä ”innovation”. Lähteiden valintaan on vaikuttanut muun muassa julkaisuvuosi, lähteen alkuperäinen julkaisufoorumi sekä viittausten määrä. Suuri osa lähteistä on julkaistu vertaisarvioidussa IEEE Software -tiedejulkaisussa. Vaikka vanhempia lähteitä on käytetty avoimen lähdekoodin historiaa, kehittymistä ja toimintatapoja tutkiessa, on tuoreempia lähteitä pyritty käyttämään kartoittaessa avoimen lähdekoodin yrityksille tarjoamia hyötyjä.

Raymondin teos ”The Cathedral and the Bazaar” on keskeisessä osassa tutkielmaa, sillä siinä esitettyjä ohjelmistokehityksen malleja käytetään tutkielmassa sekä suuressa osassa tutkielman lähdekirjallisuutta. Stallmanin esseet, jotka ovat koottu teokseen ”Free Software, Free Society”, sekä Perensin ”Open Source Definition” -asiakirja ovat myös merkityksellisiä lähteitä, koska ne ovat vaikuttaneet valtavasti avoimen lähdekoodin kehittymiseen. GNU-projekti, Free Software Foundation ja vapaiden ohjelmistojen käsite perustuvat Stallmanin esseissä esitettyihin ideoihin sekä niissä paljastettuihin omisteisten ohjelmistojen ongelmiin. Perensin asiakirja on puolestaan perusta Open Source Initiativelle sekä Open Source -brandille.

Tutkielmassa tutkitaan avoimen lähdekoodin kehitystä kronologisessa järjestyksessä ilmiön tärkeimpien projektien ja tapahtumien kautta. Tutkielmassa esitetyt projektit ja tapahtumat esiintyvät suuressa osassa lähdekirjallisuutta ja erottuvat selvästi joukosta relevantteina ja ilmiöön eniten vaikuttaneina tapauksina. Aikajaksot ovat sidottu projekteihin ja tapauksiin vuosilukujen sijaan, koska tutkielmassa esitetyt projektit ja tapaukset ovat muovanneet ilmiötä ja mahdollistaneet ilmiön suurimmat muutokset.

Toiseen tutkimuskysymykseen pyritään löytämään vastaus tutkimalla lähteitä, jotka keskittyvät avoimen lähdekoodin hyötyihin tai sen hyödyntämiseen ohjelmistoyrityksissä. Lähteistä pyritään tunnistamaan avoimen lähdekoodin keskeisimmät hyödyt ja mahdollisuudet. Miksi voittoa tavoittelevat ohjelmistoyritykset käyttävät tai tuottavat avointa lähdekoodia ja miten avoin lähdekoodi voi tukea niiden liiketoimintaa?

2 AVOIN LÄHDEKOODI

Tässä luvussa esitellään avoimen lähdekoodin keskeisimmät käsitteet sekä niiden eroavaisuudet. Luvussa tutkitaan myös avoimen lähdekoodin ominaispiirteitä, historiaa ja kehittymistä sitä ympäröivien ilmiöiden, yhteisöjen ja projektien avulla.

2.1 Lähdekoodin määritelmä

Tyypillisesti ohjelmistoja jaetaan valmiiksi käännettyinä binääritiedostoina, jotka käyttäjien tarvitsee vain asentaa laitteilleen. Ohjelmistojen binääritiedostot ovat kuitenkin alun perin käännetty ohjelmistojen lähdekoodista. Harmanin (2010) sanoin lähdekoodi on jo yli kymmenen vuoden ajan määritelty ”täysin suoritettavana kuvauksena ohjelman toiminnasta”. Käytännössä ”täysin suoritettava kuvaus ohjelman toiminnasta” tarkoittaa ohjelmointikieltä, jonka kääntäjä (compiler) tai tulkki (interpreter) kykenee kääntämään konekieleksi.

Vaikka ohjelmistoja voidaan suorittaa ja käyttää pelkkien binääritiedostojen avulla, ei niiden toiminnan tutkiminen tai muokkaaminen ole mahdollista ilman takaisinmallinnusta (reverse-engineering). Jos ohjelmistoa jaetaan vain binääritiedostoina, on ohjelmiston lähdekoodi suljettua, eivätkä sen käyttäjät voi varmuudella tietää ohjelmiston toiminnasta.

Ohjelmistoja, joiden lähdekoodi on suljettua, voivat muokata ja päivittää vain ohjelmiston kehittäjät. Mikäli ohjelmisto kaipaa uusia ominaisuuksia, korjauksia tai tietoturvapäivityksiä, on käyttäjien odotettava ohjelmiston seuraavaa versiota. Jos kehittäjät ovat hylänneet ohjelmiston julkaisematta sen lähdekoodia, on käyttäjien usein tyydyttävä saatavilla olevaan versioon tai etsittävä vaihtoehtoja kyseiselle ohjelmistolle.

Ohjelmistot, joiden lähdekoodi on suljettua, ja joita jaetaan vain binäärimuodossa, ovat tyypillisesti omisteisia ohjelmistoja (Proprietary Software). Omisteisten ohjelmistojen lisenssit rajoittavat käyttäjien oikeuksia ohjelmiston käyttöön ja jakamiseen. Omisteisten ohjelmistojen lisenssit saattavat rajoittaa

esimerkiksi binääritiedostojen jakamista tai ohjelmiston muokkaamista. (von Hippel & von Krogh, 2003).

Omisteisuus ja suljettu lähdekoodi ovat yleisiä piirteitä ilmaisohjelmistojen (Freeware) keskuudessa. Ilmaisohjelmistoille ei ole yleistä määritelmää, mutta termiä käytetään usein ohjelmistoista, joiden binääritiedostot ovat käyttäjien saatavilla ja jaettavissa. Ilmaisohjelmistoja ei tule sekoittaa vapaisiin ohjelmistoihin tai avoimeen lähdekoodiin. (Stallman, 2002, s. 73).

2.2 Avoimen lähdekoodin määritelmä

Omisteisten ohjelmistojen vastakohtana pidetään usein avoimen lähdekoodin ohjelmistoja. Avoimen lähdekoodin ohjelmistot (Open Source Software) ovat omisteisten ohjelmistojen tapaan usein saatavilla binäärimuodossa. Binääritiedostojen lisäksi avoimen lähdekoodin ohjelmistot tarjoavat käyttäjilleen mahdollisuuden tutkia ja muokata ohjelmiston lähdekoodia. Lähdekoodin vapaa saatavuus ei kuitenkaan ole ainoa kriteeri avoimen lähdekoodin ohjelmistoille. Avoimen lähdekoodin ohjelmistojen lisenssin tulee sallia lähdekoodin vapaa käyttö, muokkaaminen, uudelleenjakelu sekä muokattujen versioiden (fork, modification) ja muiden lähdekoodia hyödyntävien tuotteiden jakelu alkuperäisen lisenssin ehdoilla (Perens, 1999).

Lähdekoodin vapaa käyttö, muokkaaminen ja jakelu ovat tyypillisiä piirteitä avoimen lähdekoodin ohjelmistojen lisäksi myös vapaille ohjelmistoille (Free Software). Käsitteiden väliset erot ovat suurimmaksi osaksi filosofisia – vapaiden ohjelmistojen painottaessa käyttäjän vapautta ja ohjelmiston moraalisuutta, avoin lähdekoodi keskittyy avoimen ohjelmistokehityksen hyödyntämiseen käytännössä (von Hippel & von Krogh, 2003). Käsitteiden samankaltaisuuden vuoksi suuri osa avoimen lähdekoodin tutkimuksesta käsittelee myös vapaita ohjelmistoja. Molempia tutkittaessa käytetään usein sateenvarjotermiä vapaat ja avoimen lähdekoodin ohjelmistot (Free/Libre and Open Source Software, FLOSS) (Cowston, Wei, Howison & Wiggings, 2012).

Avoin lähdekoodi nähdään usein ohjelmistojen, lähdekoodin ja lisenssien sijaan liikkeenä ja/tai ilmiönä, joka haastaa markkinoita hallitsevien omisteisten ohjelmistojen tyypilliset toimintatavat (Freeh, Madley & Tynan, 2002). Feller (2005) kuvaa avointa lähdekoodia ”kokoelmana työkaluja ja prosesseja, joiden avulla ihmiset tekevät, vaihtavat ja hyödyntävät ohjelmistoja ja tuntemusta”. Tässä tutkielmassa avointa lähdekoodia tutkitaan vapaiden- ja avoimen lähdekoodin ohjelmistojen ympärille muodostuneena ilmiönä. Turhien sekaannusten ja käännösten välttämiseksi tutkielmassa käsitteellä ”avoin lähdekoodi” tarkoitetaan vapaita ja avoimen lähdekoodin ohjelmistoja, ohjelmistoprojekteja, niihin liittyviä toimintatapoja (mm. ohjelmistotuotantoa, lisensointia ja liiketoimintamalleja) sekä niitä ympäröiviä yhteisöjä.

2.3 Avoimen lähdekoodin kehitys

Stallmanin (2002, s. 9) ja Perensin (1999) mukaan ohjelmistoja ja lähdekoodia jaettiin jo ensimmäisten tietokoneiden aikaan. Ohjelmistojen ja lähdekoodin jakaminen oli hyvin yleistä vielä 1960- ja 1970-luvuilla, jolloin suuri osa ohjelmistokehityksestä tapahtui akateemisessa ympäristössä ja suurissa yrityksissä (von Hippel & von Krogh, 2003).

Käsitteenä vapaat ohjelmistot ja avoin lähdekoodi syntyivät kuitenkin myöhemmin omisteisten ohjelmistojen jo hallitessa markkinoita. Avoin lähdekoodi yleistyi sitä hyödyntävien projektien, kuten Berkeley Software Distributionin (BSD), GNU-projektin ja Linuxin kehittyessä (Bretthauer, 2002). Vaikka omisteiset ohjelmistot ovat hallinneet ohjelmistomarkkinoita jo 1980-luvulta, on avoin lähdekoodi onnistunut kasvattamaan suosiotaan vuosikymmenien ajan.

Lähivuosina avoimen lähdekoodin suosio on kasvanut entistä enemmän omisteisista ohjelmistoistaan tunnettujen suuryritysten – kuten Microsoftin ja Googlen – hyödyntäessä avointa lähdekoodia omissa projekteissaan. Microsoft hyödyntää avointa lähdekoodia muun muassa Azure, .NET Core ja Linux Subsystem for Windows -projekteissa, Googlen kehittäessä Androidia – Linux-pohjaista mobiilikäyttöjärjestelmää. Avoimen lähdekoodin suosion kasvua kehittäjien ja käyttäjien keskuudessa taas heijastaa muun muassa Github-verkkopalvelu, jota käytetään ohjelmavarastona (repository) useille avoimen lähdekoodin projekteille.

Vaikka suurin osa käyttämästämme ohjelmistosta on nykyään omisteista, käyttävät monet avointa lähdekoodia – mahdollisesti tietämättään – päivittäin. Tunnettuja avoimen lähdekoodin ohjelmistoja ovat muun muassa GNU/Linux käyttöjärjestelmä, Apache Web Server -palvelinohjelma, sekä Mozilla Firefox -verkkoselain. Erityisesti avoin lähdekoodin on kerännyt suosiota IT-alan ammattilaisten keskuudessa ja suuri osa suosituista kehitystyökaluista, kuten Git ja Node.js, on avointa lähdekoodia.

Avoimesta lähdekoodista on vuosien aikana muodostunut huomattava ekonominen ja sosiaalinen ilmiö (von Hippel & von Krogh, 2002), joka nähdään usein omisteisten ohjelmistojen haastajana. Avoin lähdekoodi on mahdollistanut muun muassa uusien tuotantomenetelmien, kommunikaatiotapojen, lisenssien ja liiketoimintamallien synnyn (Fritzgerald, 2006). Avointa lähdekoodia tutkiessa on tärkeä ymmärtää sitä ympäröiviä yhteisöjä ja projekteja sekä sen historiaa ja kehitystä. Tässä tutkielmassa tutkin avoimen lähdekoodin historiaa sen huomattavimpien tapahtumien ja projektien avulla. Tutkielmassa esitetyt projektit ovat oleellisia avoimen lähdekoodin kehitykselle, koska niiden toimintatavat ovat eronneet aikakaudelleen tyypillisistä avoimen lähdekoodin toimintatavoista, vaikuttaen niitä seuraavien avoimen lähdekoodin projektien toimintatapoihin. Projektit ovat kiinnostaneet eniten myös muita avoimen lähdekoodin tutkimuksia (Crowston ym., 2012).

2.3.1 Alkuperäiset hakkeriyhteisöt

Vaikka avoimen lähdekoodin käsitettä ei tunnettu vielä 1960- ja 1970-luvuilla, oli ohjelmistojen ja lähdekoodin jakelu hyvin yleistä tietokoneiden parissa työskentelevien ammattilaisten keskuudessa. Valmiiksi paketoituja ohjelmistoratkaisuja ei juuri ollut tarjolla, joten ohjelmistoja jouduttiin kehittämään itse tai niiden kehitykseen palkattiin ulkopuolinen ammattilainen. Aikakauden ohjelmistokehittäjät kokivat ohjelmistojen ja koodin jakamisen, sekä muiden jakamien ohjelmistojen muokkaamisen osaksi ”hakkerikulttuuria”. (von Hippel & von Krogh, 2003). Hakkerit ja hakkerikulttuuri ovat olennainen osa avoimen lähdekoodin kehitystä. Nykyään hakkerit leimataan helposti kyberrikollisiksi, mutta alun perin käsitteellä tarkoitettiin henkilöä, joka on kiinnostunut tietotekniikasta ja haluaa edistää alan kehittymistä. Taitava hakkeri käyttää kaikkia saatavilla olevia työkaluja (esim. muiden jakamaa koodia) saavuttaakseen haluamansa lopputuloksen. (Raymond, 1999).

1980-luvulla tilanne muuttui huomattavasti, MIT:n lisensoidessa vapaasti tuotettua lähdekoodia kaupalliselle yritykselle ja rajoittaessa lähdekoodin saatavuutta eväten osalta koodin tuottajista oikeudet lähdekoodin lukemiseen, muokkaamiseen ja jakamiseen (von Hippel & von Krogh, 2003). Omisteisten ohjelmistojen yleistyessä, suurin osa ammattilaisista palkattiin tuottamaan suljettua lähdekoodia. Yritykset näkivät ohjelmistojen lähdekoodin liikesalaisuutena ja alkoivat suojata sitä lisenssein ja salassapitosopimuksin, tuhoten 1970-luvulle tyypillisen hakkerikulttuurin. (Stallman, 2002, s. 9-11).

2.3.2 GNU ja Free Software Foundation

Suljetun lähdekoodin ja salassapitosopimusten yleistyessä, MIT:n entinen työntekijä nimeltä Richard Stallman kehitti konseptin vapaista ohjelmistoista, joiden tulisi kunnioittaa käyttäjän vapautta, sallien ohjelmistojen vapaan tutkimisen, muokkaamisen ja jakamisen (Perens, 1999). Stallman (2002) kaipasi vapaasti muokattavia ohjelmistoja ja 1970-luvun hakkerikulttuuria, joten omisteisten ohjelmistojen tuottamisen sijaan hän halusi hyödyntää taitojaan edistääkseen vapaita ohjelmistoja.

UNIX-pohjaiset käyttöjärjestelmät olivat suuressa suosiossa 1970- ja 1980-luvuilla. Myös kaupalliset UNIX versiot olivat yleisiä, jonka takia UNIX oli hajaantunut useiden yhteensopimattomien ja omisteisten versioiden välille. (Newman, 1999). Stallman (2002, s. 11) ajatteli vapaan ja UNIX-yhteensopivan käyttöjärjestelmän yhdistävän käyttäjiä ja luovan yhteisön, jossa ohjelmistoja ja lähdekoodia jaettaisiin 1970-luvun tapaan. Kuten 1970- ja 1980-luvun käyttöjärjestelmille tyypillistä, vapaan käyttöjärjestelmän tulisi Stallmanin (2002, s. 11) mukaan sisältää useita työkaluja kuten kääntäjiä, debuggereja ja tekstieditoreja. Stallmanin idea johti GNU-projektiin (GNU’s Not Unix), jota varten Stallman kirjoitti muun muassa GNU Compiler Collectionin (GCC), GNU Debuggerin (GDB) ja GNU Emacsin, jotka ovat vielä tänäkin päivänä oleellinen osa GNU/Linux-käyttöjärjestelmää. GNU-projektin kasvaessa Stallman (2002, 16)

perusti Free Software Foundationin – voittoa tavoittelemattoman yhdistyksen – rahoittaakseen GNU-projektia, tukemaan vapaiden ohjelmistojen kehitystä ja yleistääkseen vapaiden ohjelmistojen käyttöä. Free Software Foundation toimii aktiivisesti vielä nykyäänkin.

Omisteisten UNIX-käyttöjärjestelmien käyttäessä ja sulkiessa lisensoimattoman X ikkunointiympäristön koodia, Stallman (2002, s. 70-71) halusi GNU-projektin vapaiden ohjelmistojen myös pysyvän vapaana tulevaisuudessa. Suojellakseen vapaiden ohjelmistojen lähdekoodia omisteisilta lisensseiltä, Stallman loi ensimmäisen avoimen lähdekoodin lisenssin – GNU General Public Licensen (GPL). GPL:n mukaan kaikkien lähdekoodia hyödyntävien teosten tulee käyttää GPL-lisenssiä, jotta ohjelmisto pysyisi vapaana myös tulevaisuudessa. (Fritzgerald, 2006). GPL:n tapaisista lisensseistä käytetään usein termiä käyttäjänoikeus (copyleft), sillä lisenssit kumoavat teoksen tekijänoikeuden (copyright), sallien kaikkien halukkaiden muokkaavan ja jakavan työtä vapaasti lisenssin määräämin ehdoin.

2.3.3 Linux ja Basaarimalli

UNIXille ei ollut vielä moneen vuoteen täysin avointa vaihtoehtoa, sillä osa BSD:n lähdekoodista pysyi suljettuna eikä GNU-projektin HURD-ytimen kehitys edennyt toivotulla vauhdilla. Vuoden 1991 lokakuussa suomalainen opiskelija Linus Torvalds kuitenkin julkaisi ohjelmoimansa – GNU-ohjelmistojen kanssa yhteensopivan – Linux-ytimen. (Bretthauer, 2002).

Linuxin ansiosta hakkereilla oli viimein täysin ilmainen ja vapaasti muokattava käyttöjärjestelmä, jollaisen Stallman oli visioinut jo GNU-projektin alkuvaiheessa. Vapaa vaihtoehto UNIXille kannusti hakkereita osallistumaan käyttöjärjestelmän kehitykseen, mahdollistaen Linuxille tyypillisen kehitysmallin, jonka Raymond (1999) nimesi arvostetussa esseessään ”The Cathedral and the Bazaar” basaarimalliksi. Kuten Torvaldsin aloittamalle basaarityyliselle ohjelmistokehitykselle tyypillistä, Linuxia kehittää suuri maailmanlaajuinen yhteisö, joka kommunikoi internetin välityksellä. Linuxin kehitykseen voi osallistua kuka tahansa asuinpaikasta ja taustasta huolimatta.

Basaarityylisen ohjelmistokehityksen synty on tärkeä osa avoimen lähdekoodin kehitystä. Linuxia edeltäviä avoimen lähdekoodin projekteja (esim. GNU ja BSD) kehitettiin omisteisten ohjelmistojen tapaan pienissä ja tarkkaan organisoiduissa ryhmissä – Raymondin (1999) termein, katedraalimallilla, joka eroaa huomattavasti Linuxille ja muille yhteisökeskeisille projekteille tyypillisestä basaarimallista. Katedraali- ja basaarimallit ovat oleellinen osa avoimen lähdekoodin ohjelmistokehitystä ja eroavat toisistaan huomattavasti. Malleja ja niiden eroavaisuuksia tutkitaan tarkemmin luvussa 3.1 – Avoimen lähdekoodin ominaispiirteet.

2.3.4 Open Source Initiative ja Mozilla Firefox

Avoin lähdekoodi keräsi valtavasti huomiota 1990-luvun loppupuolelta 2000-luvun alkupuolelle ”täysin uutena tapana kehittää ohjelmistoja” ja markkinoita hallitsevien omisteisten ohjelmistojen ”haastajana” (Mockus, Fielding, Herbsleb, 2002). Myös akateeminen maailma kiinnostui avoimesta lähdekoodista 2000-luvun alussa ja avoimen lähdekoodin tutkimus yleistyi hurjasti 2000-luvun alussa (Crowston ym., 2012). Samaan aikaan myös yritykset kiinnostuivat avoimesta lähdekoodista ja halusivat hyödyntää sitä omassa liiketoiminnassaan.

Linux ja avoin lähdekoodi olivat jo saavuttaneet suurta suosiota akateemisessa ympäristössä, mutta Raymondin mielestä Free Software Foundationin käyttäjälähtöinen määritelmä vapaista ohjelmistoista ja ilmaisuutta implikoiva käsite ”Free Software” kuitenkin hidastivat Linuxin sekä avoimen lähdekoodin yleistymistä erityisesti kaupallisten yritysten keskuudessa. Erimielisyyksien takia Eric Raymond, Bruce Perens ja useat muut tunnetut hakkerit perustivat Open Source Initiative -järjestön, jonka tarkoitus on Free Software Foundationin tapaan edistää avoimen lähdekoodin ohjelmistojen käyttöä ja kehitystä. (von Hippel & von Krogh, 2003).

Open Source Initiativen perustamisen aikoihin Netscape (joka oli kehittänyt yhden aikansa suosituimmista internetselaimista) oli osoittanut kiinnostusta Raymondin ”The Cathedral and the Bazaar” -esseetä kohtaan ja halusi hyödyntää basaarimallia uudessa selaimessaan. Netscape otti yhteyttä Raymondiin, joka auttoi heitä kehittämään Netscapelle sopivan avoimen lähdekoodin lisenssin. (Perens, 1999). Netscapen kehittämä Mozilla Public License (MPL) oli avoimen lähdekoodin historian kannalta tärkeä, koska se mahdollisti kaupallisten ohjelmistojen siirtymisen avoimeen lähdekoodiin. GPL ei sopinut tilanteeseen, koska se olisi vaatinut kaiken Netscapen alkuperäiseen sovellukseen yhdistetyn koodin muuntamisen GPL:ksi (Fritzgerald, 2006).

Netscapen kaupallinen selain oli menettänyt markkinaosuuttaan Microsoftin kehittämälle Internet Explorerille ja avointa lähdekoodia hyödyntämällä Netscape toivoi jäljittelevänsä Linuxin yllättävää menestystä. Netscapen aloittama Mozilla-projekti on kehittänyt Firefox-selaimen lisäksi useita kehitystyökaluja ja suositun Mozilla Thunderbird -sähköpostiohjelman. (Mockus ym., 2002).

2.3.5 OSS 2.0

2000-luvun alussa Netscapen lisäksi myös muut IT-alan suuryritykset, kuten IBM ja Oracle, näkivät uusia mahdollisuuksia avoimessa lähdekoodissa, sillä avoimen lähdekoodin ohjelmistot vähensivät yritysten tarvetta toisten – usein kilpailevien – yritysten ohjelmistoratkaisuille, koska ohjelmistoja voitiin räätälöidä yritysten sisällä ilman ohjelmiston kehittäjän lupaa tai apua. Avoimen lähdekoodin ohjelmistot eivät myöskään rajoittaneet laitteisto- ja ohjelmistovalikoimaa yritysten luomiin ekosysteemeihin, toisin kuin omisteiset ohjelmistot. (Koenig, 2004).

Avoimen lähdekoodin yleistyessä yrityksissä, teknisen tuen tarve kasvoi. Avointa lähdekoodia oli jo aiemmin moitittu tuen puutteesta, sillä tukea avoimen lähdekoodin ohjelmistojen ongelmiin sai hakea yleensä käyttäjäfoorumeilta tai ohjelmistojen sähköpostilistoilta. Yritykset kuitenkin kaipasivat virallisempaa tukea avoimen lähdekoodin käyttöön, mahdollistaen avoimelle lähdekoodille yleisen – tukeen perustuvan liiketoimintamallin. Liiketoimintamallia hyödyntävät esimerkiksi Linuxiin keskittyneet yritykset kuten Novell ja Red Hat. (Koenig, 2004).

Netscapen kehittämään MPL-lisenssiin perustuvat yrityskeskeiset lisenssit ja liiketoimintamallit nostivat avoimen lähdekoodin suosiota yritysten keskuudessa ja aloittivat avoimen lähdekoodin muutoksen kohti Fritzgeraldin (2006) esittelemää OSS 2.0:aa. OSS 2.0 on Fritzgeraldin (2006) artikkelissa ”The Transformation of Open Source Software” esitetty käsite, jonka avulla hän vertaa nykyaikaisempia ja kaupallisesti kannattavampia avoimen lähdekoodin toimintatapoja, liiketoimintamalleja ja lisenssejä niiden ”perinteisiin” vastineihin.

Fritzgeraldin (2006) mukaan OSS 2.0:n yrityksille sopivammat toimintatavat eroavat huomattavasti perinteisistä avoimen lähdekoodin toimintatavoista. OSS 2.0:lle on tyypillistä tarkka ja ammattimainen ohjelmistosuunnittelu ja – kehitys avoimen lähdekoodin perinteisen – Raymondin (1999) esittämän – ”rapsuttamisen arvoisen kutinan” (”an itch worth scratching”) sijaan. OSS 2.0:lle tyypilliset toimintatavat yhdistelevät niin basaari- kuin katedraalimalleja mahdollistaen avoimen lähdekoodin hyödyntämisen yrityksen tarkoitukseen sopivalla tavlla. (Fritzgerald, 2006).

2.4 Avoimen lähdekoodin ominaispiirteet

Avoin lähdekoodi ei ole saavuttanut suosiotaan ainoastaan sen vapaiden lisenssien ja arvojen ansiosta. Avoimen lähdekoodin ohjelmistot eroavat huomattavasti omisteisista ohjelmistoista muun muassa avoimen ohjelmistokehityksen ja ainutlaatuisten liiketoimintamalliensa ansiosta. Avoin lähdekoodi tarjoaa erilaisia hyötyjä ja mahdollisuuksia niin käyttäjille kuin kehittäjille.

Raymond (1999) tutkii esseessään ”The Cathedral and the Bazaar” avoimelle lähdekoodille tyypillisiä piirteitä, keskittyen erityisesti Linuxille tyypilliseen yhteisöpainoitteiseen ohjelmistokehitykseen. Raymond (1999) vertaa Linus Torvaldsin Linux-projektissa hyödyntämiä avoimen ohjelmistotuotannon menetelmiä kuhisevaan basaariin, jota rakentaa joukko toisilleen tuntemattomia ihmisiä erilaisin päämäärin. Basaarimallin vastakohtana Raymond (1999) esittää perinteisemmän tuotantomenetelmän, jota harjoitetaan pienissä ja ulkopuolisilta suljetuissa ryhmissä, joiden jäsenillä on yhteinen päämäärä. Perinteisemmästä ja suljetummasta ohjelmistokehityksestä Raymond (1999) käyttää katedraalimetaforaa.

Raymond (1999) tutkii avoimen lähdekoodin ominaispiirteitä kahden avoimen lähdekoodin ohjelmistoprojektin, Torvaldsin Linux-käyttöjärjestelmäytimen ja Raymondin oman Fetchmail-sähköpostityökalun,

avulla. Raymond (1999) hyödynsi Linuxille tyypillistä basaarimallia omassa projektissaan, pyrkien tunnistamaan Torvaldsin hyödyntämän ohjelmistokehityksen vahvuudet. Raymond (1999) tiivisti tutkielman tulokset yhdeksääntoista ”oppituntiin”, jotka kuvaavat hyvin avoimen lähdekoodin ohjelmistokehitystä ja sen tarjoamia mahdollisuuksia.

Raymond (1999) kiinnitti erityisesti huomiota muun muassa Torvaldsille tyypilliseen julkaisuaikatauluun. Torvalds julkaisi Linuxia tihein väliajoin, jotta vapaaehtoiset käyttäjät voisivat toimia ohjelmiston testaajina. Tiheän julkaisuaikataulun riskit, kuten mahdolliset ohjelmistovirheet, eivät olleet omisteisten ohjelmistojen tapaan ongelma, koska Torvaldsin kehittämä vapaa käyttöjärjestelmäydin ei varsinaisesti kilpaillut kaupallisten käyttöjärjestelmien kanssa. Torvalds piti Linuxin käyttäjäkuntaa resurssina, joka auttoi ohjelmiston kehityksessä ja testaamisessa. Raymondin (1999) sanojen mukaan Torvalds ”delegoi kaiken, minkä kykeni” ja piti projektin mahdollisimman avoimena kaikille siitä kiinnostuneille.

Raymondin (1999) tutkimuksen keskeisin väite tunnetaan nimeltä ”Linuksen laki” (Linus’s law). Linuksen lain mukaan laaja käyttäjäkunta, jolla on mahdollisuus tutkia ohjelmiston lähdekoodia sekä osallistua ohjelmiston kehitykseen ja testaukseen helpottaa ja nopeuttaa ohjelmistovikojen löytymistä ja korjaamista. Laajan käyttäjäkunnan ansiosta erilaiset ohjelmistovirheet löytyvät nopeammin ja käyttäjien joukosta löytyy usein joku, joka ymmärtää virheen, osaa korjata sen tai auttaa ohjelmistokehittäjiä korjaamaan sen (”given enough eyeballs, all bugs are shallow”). (Raymond, 1999).

Ohjelmistokehityksen lisäksi avoimen lähdekoodin ohjelmistot eroavat omisteisista ohjelmistoista myös liiketoiminnaltaan ja ansaintamalleiltaan. Vaikka avoimen lähdekoodin ohjelmistot ovat usein ilmaisia, voi niitä hyödyntää monipuolisesti liiketoiminnassa ja jopa myydä perinteisen ohjelmiston tavoin. Jo GNU-projektin alkuvaiheessa Stallman (2002, 16) keräsi projektille rahoitusta myymällä vapaita ohjelmistoja CD-levyillä.

Ohjelmistojen ja ohjelmistolisenssien myyminen perinteisten ohjelmistojen tapaan voi kuitenkin olla haaste avoimen lähdekoodin ohjelmistoille, koska asiakkailla on oikeus jakaa ohjelmistoa ja sen lähdekoodia vapaasti eteenpäin. Koska avointa lähdekoodia on vaikea myydä perinteisin tavoin, on avoimen lähdekoodin tuottaminen nähty usein sopimattomana vaihtoehtona ohjelmistoyrityksille.

Avoimen lähdekoodin hyödyntäminen oli vielä 90-luvun alussa harvinaista ohjelmistoyrityksissä, koska avoimen lähdekoodin projektit nähtiin usein amatöörimäisinä ja tukea niiden käyttöön jouduttiin hakemaan sähköpostilistoilta ja käyttäjäfoorumeilta. (Hecker, 1999). 2000-luvulla avoimen lähdekoodin kehitys on kuitenkin yleistynyt kaupallisissa ohjelmistoyrityksissä niiden investoidessa jatkuvasti enemmän rahaa ja työvoimaa avoimen lähdekoodin projekteihin. Myös tuki avoimen lähdekoodin käyttöön on parantunut huomattavasti vuosien aikana, ja monien avoimen lähdekoodin projektien koetaan nykyään tarjoavan jopa laajemmin tukea kuin niiden omisteisten vaihtoehtojen. (Ven, Verelst & Mannaert, 2008).

3 AVOIN LÄHDEKOODI YRITYSKÄYTÖSSÄ

Vaikka avoimen lähdekoodin ei uskoisi sopivan yrityskäyttöön sen suosimien vapaiden arvojen ja lisenssien vuoksi, on avoimen lähdekoodin käyttö yleistynyt yrityskäytössä huomattavasti 2000-luvulla (Ebert, 2008). Tässä luvussa keskitytään avoimen lähdekoodin käyttöön ja hyödyntämiseen yrityksissä. Miten yritykset voivat hyötyä avoimen lähdekoodin käytöstä, ja miten avoimen lähdekoodin projektin voi lisensoida tai tuotteistaa?

3.1 Avoimen lähdekoodin hyödyntäminen

3.1.1 Modulaarinen ohjelmistokehitys

Perinteinen ohjelmistokehitys ilman kolmannen osapuolen ohjelmistokehyksiä (framework) tai -kirjastoja (library) on nykyään erittäin harvinaista (Ebert, 2008). Avoimen lähdekoodin ohjelmistokehysten, -kirjastojen ja -työkalujen laajan tarjonnan ansiosta yritysten ei tarvitse aloittaa suuria ohjelmistoprojekteja täysin tyhjästä. Sen sijaan suuret projektit voidaan rakentaa saatavilla olevia avoimen lähdekoodin komponentteja hyödyntäen.

Koska kolmannen osapuolen komponenteilla on ollut aikaa kehittyä ja kypsyä jo ennen projektin alkua, ovat ne usein laadukkaampia, paremmin dokumentoituja ja sisältävät ne usein laajemmin ominaisuuksia kuin projektin resursseilla kehitetyt ja testatut ratkaisut. Avoimen lähdekoodin ansiosta valmiita komponentteja voidaan myös räätälöidä projektin tarpeisiin, kartuttaen niiden ominaisuuksia entisestään (Spinellis & Szyperski, 2004).

Avoimen lähdekoodin ohjelmistokehysten ja -kirjastojen mahdollistama modulaarinen ohjelmistokehitys helpottaa sekä ohjelmistojen kehitystä että niiden ylläpitoa. Ohjelmistoyritysten tuottaessa entistä laajempia ja monimutkaisempia ohjelmistokokonaisuuksia, helpottavat kolmannen osapuolen komponentit ohjelmistojen ylläpitoa, koska kolmannen osapuolen komponentteja ylläpitää ja kehittää basaarimainen yhteisö, joka koostuu komponentin kehittäjistä ja käyttäjistä. Kolmannen osapuolen komponenttien käyttö helpottaa myös

vianmääritystä, sillä komponentin muut käyttäjät ovat todennäköisesti törmänneet samanlaisiin ongelmiin. (Ebert, 2008).

Resursseja säästyy ylläpidon lisäksi myös ohjelmiston kehitysvaiheessa, koska valmiiden ohjelmistokomponenttien hyödyntäminen ja muuntaminen projektin vaatimusten mukaan vie usein vähemmän aikaa, kuin täysin uuden komponentin kirjoittaminen. Koska ohjelmistokomponentteja ei ole kehitetty vain yhden projektin tarpeisiin, ovat ne helposti uudelleen käytettävissä, säästäen resursseja myös tulevissa projekteissa, jotka vaativat samankaltaisia ratkaisuja. (Bonarcossi & Rossi, 2002).

Modulaarinen ohjelmistokehitys helpottaa myös koordinointia projektin ohjelmistokehittäjien keskuudessa, sillä komponentit ovat usein toisistaan riippumattomia (Osterloh & Rota, 2007). Modulaarinen ohjelmistokehitys helpottaa myös koordinointia, erityisesti projekteissa, joissa on useita ohjelmistokehittäjiä. Ohjelmistokehittäjien kehittäessä eri komponentteja, vältytään helposti ”astumasta toisten varpaille” (Bonarcossi & Rossi, 2002).

Avoimen lähdekoodin komponenttien käyttö voi kuitenkin vaikeuttaa ohjelmiston kaupallistamista komponenttien lisensseistä riippuen. Osa avoimen lähdekoodin ”copyleft” lisensseistä vaatii myös niitä hyödyntävien ohjelmistojen lisensoinnin samoin termein. (Spinellis & Szyperski, 2004). Avoimen lähdekoodin lisenssien aiheuttamia ongelmia on kuitenkin mahdollista lieventää esimerkiksi kaksoislisensoinnin avulla, palvelupainotteisilla liiketoimintamalleilla (Software as a Service, SaaS) tai paketoimalla avoimen lähdekoodin komponentit erikseen. Avoimen lähdekoodin lisenssejä ja tuotteistamista käsitellään lisää luvussa 3.3.

3.1.2 Omavaraisuus

Erilaiset ohjelmistot ja ohjelmistokomponentit tukevat nykyään monien yritysten liiketoimintaa, ja ovat joskus jopa kriittisessä asemassa yritysten ydinliiketoimintaa. Yrityksen liiketoiminnan perustuessa kolmannen osapuolen tarjoamaan ohjelmistoon, on yritys mahdollisesti riippuvainen ohjelmistontarjoajasta (vendor lock-in), koska ohjelmiston vaihtaminen voi koitua yritykselle erittäin kalliiksi (Ven ym., 2008).

Avoin lähdekoodi nähdään usein ratkaisuna riippuvuuteen ohjelmistontarjoajasta, koska se mahdollistaa ulkopuolisten tarjoajien tutkivan ohjelmiston toimintaa ja jatkavan sen kehitystä. Vaikka ohjelmistoa käyttävä yritys joutuisi lopettamaan yhteistyön aikaisemman tarjoajan kanssa, voi se jatkaa ohjelmiston käyttöä ja kehitystä sisäisen tiimin tai uuden tarjoajan kanssa. (Ruffin & Ebert, 2004; Ven ym., 2008).

Mikäli yritys haluaa rakentaa yhteensopivan ohjelmiston tai korvata vanhan ohjelmiston, on yhteensopivuus ohjelmistojen välillä helpompi toteuttaa, jos ohjelmisto seuraa avoimia standardeja kolmannen osapuolen omisteisten ratkaisujen sijaan. (Ruffin & Ebert, 2004; Ven ym., 2008).

3.1.3 Muut hyödyt

Taloudellisten hyötyjen lisäksi avoin lähdekoodi mahdollistaa kehittäjien ja opiskelijoiden kompetenssikehityksen myös koulussa ja vapaa-ajalla. Ohjelmistojen ollessa työntekijöille entuudestaan tuttuja, on aikaisemman osaaminen hyödyntäminen työtehtävissä mahdollista. (Ruffin & Ebert, 2004).

Avoin lähdekoodi kannustaa myös innovaatioihin. Ebertin (2007) mukaan avoin lähdekoodi on ohjelmistokehittäjien ja tutkijoiden mielestä esimerkiksi patentteja tärkeämpi lähde uusille ideoille. Avoimen lähdekoodin kannustaessa yrityksiä hyödyntämään käyttäjien ja muun yhteisön kontribuutioita kehityksessä ja dokumentoinnissa, mahdollistaa avoin lähdekoodi myös käyttäjien innovaatioiden hyödyntämisen projekteissa. Avoin kommunikaatio loppukäyttäjien kanssa helpottaa yrityksiä myös ymmärtämään loppukäyttäjien tarpeet ja vaatimukset, edistäen innovaatioita. (Von Hippel, 2001).

Avoimen lähdekoodin tarjoamien ohjelmistojen laatu ja turvallisuus mainitaan usein tutkiessa sen tarjoamia hyötyjä (Raymond 1999; Ven ym., 2008). Vaikka monet avoimen lähdekoodin ominaispiirteet, kuten modulaarisuus ja käyttäjien mahdollisuus koodikatselmointiin, voivat vaikuttaa positiivisesti ohjelmiston laatuun, riippuu niiden vaikutus yhteisön koosta (Aberdour, 2007). Vaikka avoimen lähdekoodin projektitkin voivat olla huonolaatuisia tai sisältää tietoturva-aukkoja, on niitä hyödyntävillä tahoilla kuitenkin mahdollisuus kartoittaa mahdolliset riskitekijät projektin lähdekoodin avulla (Ruffin, Ebert, 2004).

3.2 Avoimen lähdekoodin tuotteistaminen

Vaikka avoin lähdekoodi mielletään usein ilmaiseksi, voi sen avulla tavoitella myös taloudellista voittoa (Hecker, 1999). Ohjelmistoalan kilpailun kasvaessa ohjelmistoyritykset ovat alkaneet etsimään tapoja hyödyntää avointa lähdekoodia liiketoiminnassaan (Riehle, 2007). Avoimen lähdekoodin ohjelmistot voi tuotteistaa omisteisten ohjelmistojen tavoin, ja avoimen lähdekoodin ympärille on muodostunut useita, ohjelmistoyrityksille ennennäkemättömiä, liiketoimintamalleja. Ohjelmistoyrityksen liiketoiminta voi perustua täysin avoimeen lähdekoodiin. (Hecker, 1999).

3.2.1 Ohjelmistojen jakelu, ylläpito ja tuki

Avointa lähdekoodia voi hyödyntää ohjelmistoliiketoiminnassa esimerkiksi tukeen perustuvien liiketoimintamallien avulla. Vaikka avoimen lähdekoodin yhteisöt tarjoavat usein tukea projekteille esimerkiksi sähköpostilistoilla ja keskustelupalstoilla, saattavat yritykset kaivata virallisempaa tukea, jota omisteisten ohjelmistojen tarjoajat usein tarjoavat. (Krishamurty, 2005). Virallisen tuen puute on yleinen syy omisteisten ohjelmistoratkaisujen suosimiseen yritysten keskuudessa. Yritykset ovat valmiita investoimaan kattavaan tukeen ohjelmiston ollessa tärkeässä osassa niiden toimintaa. (Fritzgerald, 2004).

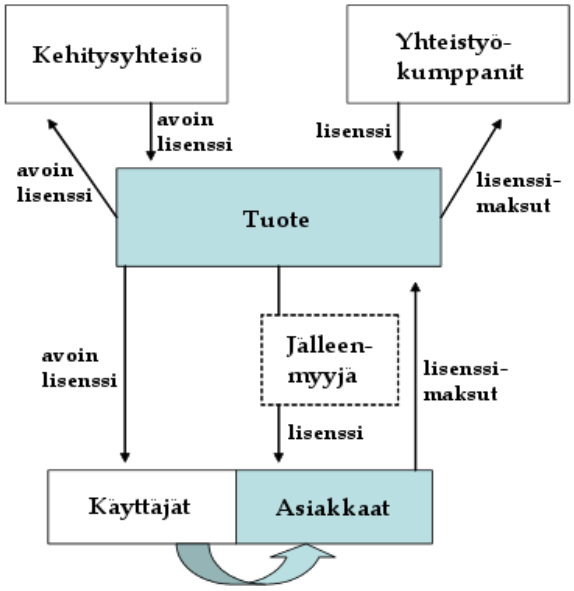
Useat avoimeen lähdekoodiin keskittyvät ohjelmistoyritykset, kuten Red Hat, pyrkivät tekemään avoimesta lähdekoodista mahdollisimman järkevän ja houkuttelevan vaihtoehdon siitä kiinnostuneille yritysasiakkaille. Red Hatin kaltaiset ohjelmistoyritykset tarjoavat asiakkailleen kattavia ja laadukkaita avoimen lähdekoodin ohjelmistokokonaisuuksia sekä omisteisten ohjelmistojen tarjoajille tyypillistä kattavaa tukea. (Watson, ym., 2008).

Vaikka Red Hatin kaltaisten yritysten tarjoamat ohjelmistokokonaisuudet ovat täysin avointa lähdekoodia ja usein saatavilla myös ilmaiseksi, maksavat asiakkaat ohjelmistolisensseistä, koska ohjelmistokokonaisuudet ovat luotetun ohjelmistoyrityksen jakamia, ylläpitämiä ja tukemia. Ohjelmistoyritysten tarjoamat ohjelmistokokonaisuudet ovat myös hyvin dokumentoituja ja ohjelmistolisenssi voi sisältää useita ohjelmistoja täydentäviä palveluita. (Watson, ym., 2008). Jakelun, ylläpidon ja tuen lisäksi ohjelmistoyritysten tarjoamat avoimen lähdekoodin ohjelmistokokonaisuudet voivat olla yrityksen brändäämiä. Avoimen lähdekoodin ohjelmistoille voidaan tarjota myös koulutusta, konsultointia, sertifikaatteja sekä asiakkaiden tarpeiden mukaista mukautettua kehitystä. (Hecker, 1999).

3.2.2 Avointen ja omisteisten lisenssien yhdistäminen

Avoimen lähdekoodin myyminen omisteisten ohjelmistojen tapaan onnistuu esimerkiksi kaksoislisenssin avulla. Ohjelmistoyritykset voivat myydä avoimen lähdekoodin ohjelmistoja omisteisilla lisensseillä. (Watson ym., 2008; Välimäki, 2003). Kaksoislisenssiä hyödyntävät esimerkiksi MySQL ja TrollTech. (Välimäki, 2003).

Kaksoislisensoidut ohjelmistot ja ohjelmistokomponentit ovat avoimen lähdekoodin tapaan vapaasti käytettävissä, muokattavissa ja jaettavissa. Kaksoislisensoitujen ohjelmistojen lisenssi kuitenkin rajoittaa lähdekoodin käyttöä omisteisissa ohjelmistoissa copyleft lisenssien tapaan, mutta tarjoaa käyttäjille mahdollisuuden ostaa lisenssi, joka sallii ohjelmiston käytön myös omisteisissa ohjelmistoissa. Kaksoislisensoitua ohjelmistoa hyödyntävät avoimen lähdekoodin projektit noudattavat avointa lisenssiä, kun taas omisteisen lisenssin ostaneet yhteistyökumppanit noudattavat omisteista lisenssiä (ks. kuvio 1). (Välimäki, 2003).



KUVIO 1 Vuorovaikutus kaksoislisenssillä

Yritykset voivat yhdistellä avointa lähdekoodia ja omisteisia ohjelmistoja myös julkaisemalla ohjelmiston ydintoiminnallisuuden avoimen lähdekoodin lisenssillä, mutta laajentaen ohjelmiston toiminnallisuutta omisteisien lisenssien avulla. Esimerkiksi Oraclen virtualisoimisohjelman, VirtualBoxin, ydintoiminnallisuus on lisensoitu avoimen lähdekoodin GPL-lisenssillä, mutta käyttökokemusta parantava laajennus, VirtualBox Extension Pack, on lisensoitu omisteisella PUEL-lisenssillä (Personal Use and Evaluation License). (Oracle, 2020).

4 YHTEENVETO

Avoin lähdekoodi on mielenkiintoinen ilmiö, joka kannustaa innovaatioon aktivoimalla käyttäjiä ja mahdollistamalla käyttäjien osallistumisen ohjelmistokehitykseen kaikissa sen vaiheissa. Oli kyse käyttöjärjestelmistä, internetselaimista tai palvelinohjelmista, ovat avoimen lähdekoodin projektit tarjonneet kaupallisten ohjelmistojättien tuotteille vapaita vaihtoehtoja jo vuosikymmenien ajan.

Vaikka avoin lähdekoodi nähtiin vielä 2000-luvun alussa amatöörimäisenä puuhasteluna, eivätkä ohjelmistoyritykset hyödyntäneet avointa lähdekoodia läheskään yhtä paljon kuin nykypäivänä, ovat suuret avoimen lähdekoodin projektit kasvattaneet yritysten luottoa ja kiinnostusta avointa lähdekoodia kohtaan jo yli vuosikymmenen ajan. Omisteisten ohjelmistojen sijaan, monet avoimen lähdekoodin ohjelmistot, kuten Linux-käyttöjärjestelmäydin ja Apache Web Server, ovatkin muodostuneet jo ”de facto” standardeiksi alan yrityksissä.

Avoimen lähdekoodin pyöriessä 2000-luvun alussa pääosin palvelimilla ja ohjelmistokehittäjien koneilla, ovat suuret ohjelmistojätit alkaneet tuottamaan 2010-luvulla entistä enemmän avointa lähdekoodia myös loppukäyttäjilleen. Esimerkiksi Googlen Android- ja ChromeOS-käyttöjärjestelmät koostuvat suurimmaksi osaksi avoimen lähdekoodin ohjelmistokomponenteista.

Voittoa tavoittelevien ohjelmistoyritysten kiinnostus avointa lähdekoodia kohtaan tuskin juurtaa juuriaan silkkaan altruismiin, sillä avoin lähdekoodi tarjoaa myös yrityksille paljon potentiaalisia hyötyjä. Avoimen lähdekoodin ohjelmistoja ja ohjelmistokomponentteja hyödyntäessä yritykset voivat säästää resursseja niin ohjelmiston kehityksessä, testauksessa kuin ylläpidossa. Suuria avoimen lähdekoodin projekteja kehittää ja ylläpitää usein kolmas osapuoli. Yrityksen ei välttämättä tarvitse käyttää resursseja kyseisen ohjelmiston tai ohjelmistokomponentin kehitykseen tai testaukseen.

Avoimen lähdekoodin projektien ohjelmistokehitykseen osallistuu usein ylläpitäjien lisäksi käyttäjistä koostuva ”basaarimainen” yhteisö. Muiden yritysten ja osaajien osallistuminen ohjelmistokehitykseen vaikuttaa positiivisesti ohjelmiston laatuun. Erityisesti projektit, joilla on laaja käyttäjäkunta, hyötyvät avoimelle lähdekoodille tyypillisistä toimintatavoista.

Ohjelmistoyritykset ovat myös löytäneet tapoja pyörittää liiketoimintaa avoimen lähdekoodin ympärillä. Monet avoimeen lähdekoodiin keskittyvät

yritykset, kuten Red Hat, lieventävät avoimeen lähdekoodiin liittyviä huolia yritysasiakkaille tarjoamillaan ohjelmistokokonaisuuksilla. Red Hatin kaltaiset yritykset jakavat brändättyjä ohjelmistokokonaisuuksia asiakkailleen, ja tarjoavat kattavaa tukea ja dokumentaatiota ohjelmistoille.

Jotkut ohjelmistoyritykset, kuten TrollTech, yhdistelevät avoimen lähdekoodin ja omisteisten ohjelmistojen hyötyjä kaksoislisensseillä. Kaksoislisenssin avulla kehitysyhteisö ja muut avoimen lähdekoodin projektit saavat käyttää ohjelmistoa vapaasti avoimen lisenssin avulla, mutta myös omisteinen lisenssi on tarjolla, mikäli käyttäjä haluaa hyödyntää ohjelmistoa omisteisissa ohjelmistoissa.

Vaikka tutkimuksia avoimen lähdekoodin hyödyistä ohjelmistoyrityksissä on tehty todella kattavasti 2000-luvun alussa IBM:n panostaessa avoimeen lähdekoodiin, Netscapen kehittäessä Firefoxia ja Apache Web Serverin yleistyessä, on 2010-luvulla tutkimusten määrä vähentynyt huomattavasti. Avoimen lähdekoodin yleistyessä 2010-luvulla suurten, omisteisista ohjelmistoista tunnettujen, ohjelmistojättien panostaessa avoimeen lähdekoodiin ja avoimen lähdekoodin kehittäjä- sekä käyttäjäyhteisöjen kasvaessa moderneilla avoimen lähdekoodin jakoalustoilla, kuten GitHubissa ja GitLabissa, on avoimen lähdekoodin tutkimukselle syntynyt uusia mahdollisuuksia.

Monet 2000- ja 2010-luvuilla aloitetut avoimen lähdekoodin projektit, esimerkiksi Googlen Android-käyttöjärjestelmä, ovat olleet markkinoilla jo vuosien ajan. Vaikka 2000-luvun tutkimukset avoimen lähdekoodin tarjoamista hyödyistä selittää osin suurten ohjelmistoyritysten kiinnostuksen avoimesta lähdekoodista, voisi avoimen lähdekoodin tarjoamien hyötyjen ja siihen liittyvien odotusten toteutumista tutkia markkinoilla olevien, avointa lähdekoodia hyödyntävien, ohjelmistojen perusteella.

LÄHTEET

Aberdour, M. (2007). Achieving quality in open-source software. *IEEE software*, *24*(1), 58-64.

Bonaccorsi, A., & Rossi, C. (2003). Why open source software can succeed. *Research policy*, *32*(7), 1243-1258.

Bretthauer, D. (2002). Open source software: A history. *Information Technology and Libraries, 21*(1), 3.

Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2012). Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR), 44(2),* 7.

Feller, J. (Ed.). (2005). *Perspectives on free and open source software.* MIT press.

Fitzgerald, B. (2004). A critical look at open source. *Computer*, *37*(7), 92-94.

Fitzgerald, B. (2006). The transformation of open source software. *MIS quarterly*, 587-598.

Ebert, C. (2007). Open source drives innovation. *IEEE Software*, *24*(3), 105-109.

Ebert, C. (2008). Open source software in industry. *IEEE Software*, *25*(3), 52-53.

Harman, M. (2010, September). Why source code analysis and manipulation will always be important. In *Source Code Analysis and Manipulation (SCAM), 2010 10th IEEE Working Conference on* (pp. 7-19). IEEE.

Hars, A., & Ou, S. (2001, January). Working for free? Motivations of participating in open source projects. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on* (pp. 9). IEEE.

Hecker, F. (1999). Setting up shop: The business of open-source software. *IEEE software*, *16*(1), 45-51.

Von Hippel, E. (2001). Open source shows the way: Innovation by and for users–no manufacturer required. *Sloan Management Review*, *42*(4), 82-86.

Von Hippel, E., & von Krogh, G. V. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. *Organization science, 14(2),* 209-223.

Koenig, J. (2004). Seven open source business strategies for competitive advantage. *IT Manager's Journal, 14.*

Kolassa, C., Riehle, D., Riemer, P., & Schmidt, M. (2014) Paid vs. volunteer work in open source. In *System Sciences (HICSS), 2014 47th International Conderence (pp. 3286-3295).* IEEE.

Krishnamurthy, S. (2005). An analysis of open source business models.

Madey, G., Freeh, V., & Tynan, R. (2002). The open source software development phenomenon: An analysis based on social network theory. *AMCIS 2002 Proceedings*, 247.

Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM), 11*(3), 309-346.

Newman, N. (1999). The origins and future of open source software.

O'Reilly, T. (1999). Lessons from open-source software development. *Communications of the ACM, 42*(4), 32–37.

Oracle. (12.5.2020). Licensing: Frequently Asked Questions. Haettu osoitteesta https://www.virtualbox.org/wiki/Licensing_FAQ

Osterloh, M., & Rota, S. (2007). Open source software development – Just another case of collective invention?. *Research Policy*, *36*(2), 157-171.

Perens, B. (1999). The open source definition. *Open sources: voices from the open source revolution*, 1, 171-188.

Raymond, E. (1999). The cathedral and the bazaar. *Philosophy & Technology, 12*(3), 23.

Riehle, D. (2007). The economic motivation of open source software: Stakeholder perspectives. *Computer*, *40*(4), 25-32.

Ruffin, C., & Ebert, C. (2004). Using open source software in product development: A primer. *IEEE software*, *21*(1), 82-86

Spinellis, D., & Szyperski, C. (2004). How is open source affecting software development?. *IEEE software*, *21*(1), 28.

Stallman, R. (2002). *Free software, free society: Selected essays of Richard M. Stallman.*

Valimaki, M. (2002). Dual licensing in open source software industry.

Watson, R., Boudreau, M., York, P., Greiner, M., & Wynn, D. (2008). The business of open source. *Communications of the ACM*, *51*(4),