



Kai K. Kimppa
Lehtori
Turun Yliopisto
Informaatioteknologian laitos

Voidaanko tietojärjestelmiä tehdä toisin?

Voisiko vapaa ja avoin ohjelmistokehitys-paradigma tuottaa parempia järjestelmiä? Artikkelissa pohditaan yhteistyön ja uudelleenkäytön tarjoamia mahdollisuuksia järjestelmäkehityksessä. Tämä parantaisi pienten ja keski suurten ohjelmistotuottajien mahdollisuuksia toimia kentällä helpottamalla pääsyä markkinoille ja mahdollistaisi organisaatioille niiden käyttämien järjestelmien kehittämisen kilpailuttamisen vapaasti.

Uudelleenkäytön lupaukset eivät ole toteutuneet

Olio-ohjelmoinnin lupaukset moduulien uudelleenkäytettävyydestä ovat toteutuneet vain osittain. Järjestelmiä kehitettäessä edelleen 'keksitään pyörää uudestaan', vaikka tarvetta sille ei modulaarisen järjestelmäkehityksen tultua yhä yleisemmäksi olisi. Ongelman muodostaa takeruminen vanhaan immateriaalioikeusmalliin joka sulkee jo kehitetyt moduulit yrityskohtaisiksi sen sijaan, että niitä voitaisiin käyttää hyödyksi uusien järjestelmien kehittämisessä. Koska tekijänoikeus suojaa erityisesti käännettyä koodia lähdekoodin pysyessä salaisuutena, on lähdekoodin lisensointi hankalaa. Jos lähdekoodin julkaisee, ei kilpailijan niin tarvitse tehdä, jolloin on vaikea todistaa onko kilpailijan käännetty, ajonaikainen koodi perustunut julkaistuun vai itse kirjoitettuun lähdekoodiin. Samankaltaisia ratkaisuja tulee helposti, ja toista kääntäjää käyttämällä on käännetty koodi kuitenkin erilaista, vaikka lähdekoodi olisikin ollut sama.

Immateriaalioikeus vääristää kilpailua myös antamalla monopolin kirjoitettuun koodiin. Vaikka immateriaalioikeuksien tarkoituksena on antaa rajoitettu monopoli, on monopoli tiettyyn koodiin käytännössä rajoittamaton. Viimeisen tekijän kuolinvuodesta seitsemänkymmenen vuoden kuluttua vapautuva koodi on itsestään selvästi ikuisen suojan alla – koodin käytettävyyden yli seitsemänkymmenen vuoden päästä on käytännössä mahdotonta. Toistaiseksi meillä ei ole mitään koodia, joka olisi näin vanhaa¹, ja ottaen ohjelmistotalan kehitysvauhdin huomioon, mikään edes

parikymmentä vuotta vanha koodi ei juurikaan ole käyttökelpoista enää tänään. Puhtaat algoritmit ovat tietysti luku erikseen. Niille ei perinteisesti ole immateriaalioikeussuojaa voinut saada koska niiden katsotaan olevan matemaattisia yhtälöitä jotka ovat tieteellisiä lakeja ja siten immateriaalioikeuksien ulkopuolella. Rajoittamattomia monopoleja pyritään estämään kaikilla muilla aloilla paitsi immateriaalioikeuden alaisilla. Jostain syystä ne siellä katsotaan normaaleiksi vaikka monopolit muuten katsotaan kilpailua vääristäviksi tekijöiksi.

Vapaan ja avoimen ohjelmistokehityksen avulla uudelleenkäyttämään?

Jos nyt modulaarisen ohjelmoinnin aikakaudella siirryttäisiin käyttämään vapaan ohjelmistokehityksen idean mukaista kehitysmenetelmää (jonka puolesta väitöskirjassani esitän muita argumentteja) tulisi ohjelmistokoodin laajempi uudelleenkäyttäminen mahdolliseksi. Organisaation tietojärjestelmät itsessään täytyy kuitenkin tehdä organisaatiokohtaisesti toimiviksi; mitään "hopealuotia" joka ratkaisisi edes jonkun alan kaikkien organisaatioiden tarpeet tuskin koskaan löydetään. Tällöin ohjelmointityö sinällään ei toki loppuisi eikä tarve sovittaa moduulit organisaation tarpeisiin poistuisi, mutta järjestelmäkehitystä ei aina tarvitsisi aloittaa uudestaan vaan jo kehitetyt moduuleita voitaisiin hyväksikäyttää laajalti.

Myös kilpailuasetelma paranisi. Uusien toimijoiden aloittaminen helpottuisi – voisivathan ne näin käyttää aiemmin ohjelmoituja moduuleita hyväkseen – joka lisäisi alan kilpailua. Samaten korjaukset, lisäykset ja uudet osat olemassa oleviin järjestelmiin voitaisiin kilpailuttaa erikseen, eivätkä organisaatiot olisi sidoksissa yhteen järjestelmätoimittajaan. Tämä kannustaisi järjestelmätoimittajia hyvään palveluun ja huolelliseen ohjelmointiin. Tietysti mikäli normaalisti muilla aloilla edellytetty palvelualltius olisi ohjelmistotalallakin normi, käyttäisi organisaatio varmasti mieluiten järjestelmän alkuperäistä toimittajaa jatkokehityksenkin tekijänä; onhan heillä selke-

ästi järjestelmän kehittäjänä myös paras ymmärrys sen toiminnasta. Organisaatio ei kuitenkaan olisi sidottu näin tekemään, jos palvelu ei pelaisi tai organisaation tarpeita ei otettaisi riittävästi huomioon.

Markkinatilanteen muuttuessa myös markkinointistrategioiden täytyisi tietysti sopeutua uuteen tilanteeseen. Koska ensimyyntin (ks. Kuva 1) täytyisi kattaa ainakin merkittävä osa kehityskustannuksista – seuraavien myyntien ollessa selkeästi kilpailumpia muidenkin voidessa hyödyntää kehitettyjä moduuleita – olisi sekä kehittäjällä että ostavilla organisaatioilla nykyistä selvempi motiivi yhteistyöhön toisten kehittäjien ja organisaatioiden kanssa järjestelmiä tilattaessa. Myös kustannusten korvaus siirtyisi enemmän tehdyn työn määrästä riippuvaksi ”lopullisen” tuotteen uudelleenmyyntin sijaan. Koska olen kuitenkin enemmän yhteiskuntafilosofi kuin markkinoinnin ammattilainen, jätän uusien mallien kehittämisen, sikäli kuin ne ajankohtaisiksi tulevat, markkinoinnin ammattilaisille: yritysten markkinointiasetoille ja kauppaakkeakouluihin.

Kuvassa 1 on malli ensimmäisen myyntin osuus järjestelmän tuomasta tulosta. Seuraavien myyntien osuus on mallissa selvästi laskeva koska saman järjestelmän osia tai kokonaista järjestelmääkin on kauppaamassa myös kilpailijoita.

Ehkä paras uudistus esittämästäni mallista seuraisi kuitenkin tietojärjestelmän syvällisen oppimisen mahdollistumisena. Jos organisaatioissa olisi halua ymmärtää järjestelmän toimintaa syvällisemmin, ei mikään estäisi sen omia toimijoita, esimerkiksi IT-osaston työntekijöitä tai vaikka edistyksellisiä käyttäjiäkään tutustumasta järjestelmän toimintaan ohjelmakoodin tasolla. Näin ymmärrys järjestelmän toiminnasta olisi parempi myös organisaation sisällä, eikä kaikkia korjauksia tarvitsisi odottaa sormi suussa ilman mahdollisuutta vaikuttaa niiden tapahtumiseen tai toteutustapaan.

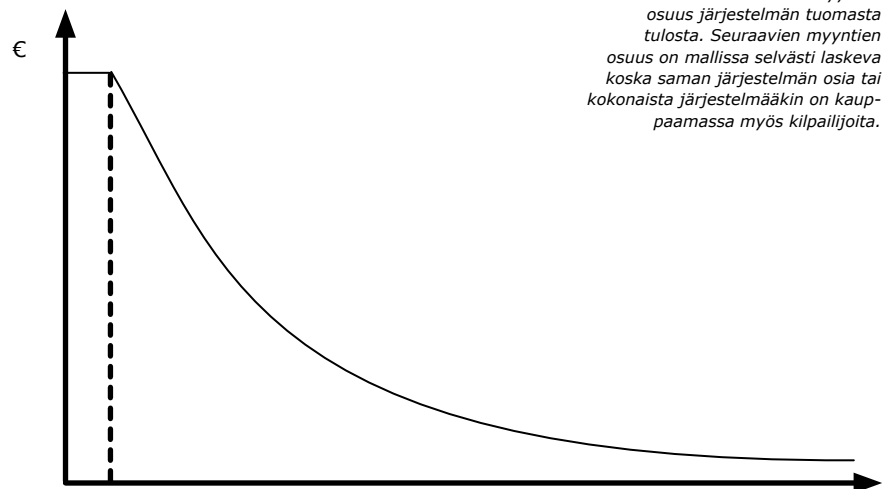
Esittämästäni mallista on joitain esimerkkejä vapaan ja avoimen ohjelmistokehityksen (FLOSS²) piirissä jo nyt³. Ongelmaksi on muodostunut yleisen ilmapiirin vastaisuus vapaan ja avoimen ohjelmakoodin kehityksen kautta tuotetuille järjestelmille. Jopa valtion ja kuntien organisaatioita vaaditaan – yleensä toisistaan erillään – kilpailuttamaan

”vapaila” markkinoilla tarjotut järjestelmäratkaisut. Tämä estää, tai ainakin haittaa järjestelmien hankkimista yhteistyössä, joka taas yleensä sulkee tarvittavat investoinnit FLOSS-kehityksen toteuttamiseksi pois – kun järjestelmän kehittäjä ei voi luottaa uudelleenmyynteihin tai monopoliin järjestelmän jatkotoimittajana ja huoltajana vaan heidän täytyy tehdä alkuperäinen tarjous kokonaiskulut mielessään. Tuntuu ristiriitaiselta lukea monilta tahoilta kuinka kuntien ja valtioiden organisaatioissa pitäisi pyrkiä avoimuuteen toimintatapojen suhteen ja samalla käytännössä tämä järjestelmien suhteen valtaosin estetään. Vanha sanonta siitä kuinka hallinnon oikea käsi ei tiedä mitä vasen tekee tulee väistämättä mieleen.

Lopuksi

FLOSS-malliin perustuva järjestelmäkehitys, niiltä osin kuin se on toteutunut, on muutamaa suurta kansallista projektia lukuun ottamatta kuitenkin jäänyt lähinnä pienten ja keskisuurten yritysten pienten järjestelmien kehityksen työkaluksi. Toivottavasti suuretkin organisaatiot voivat ja erityisesti kansallisella ja kunnallisella taholla saavat jatkossa kilpailuttaa tarvitsemansa tietojärjestelmät ottaen kaikki mahdolliset toteutustavat huomioon, olematta ”väkisin” ajettuja käyttämään nykyisin vallitsevaa suljettua ohjelmistokehitystä.

FLOSS
= vapaa ja avoin
ohjelmistokehitys
(Free/Libre/Open-
Source Software)



Kuva 1: Ensimmäisen myyntin osuus järjestelmän tuomasta tulosta. Seuraavien myyntien osuus on mallissa selvästi laskeva koska saman järjestelmän osia tai kokonaista järjestelmääkin on kauppaamassa myös kilpailijoita.

¹Ellei Ada Byronin Bernoullin lukuja laskevaa koodia Babbagen koneelle lasketa.

²FLOSS = vapaa ja avoin ohjelmistokehitys (Free/Libre/Open-Source Software)

³Ks. esim Kolab, <http://www.kolab.org/> tai JBoss, <http://labs.jboss.com/portal/index.html?ctrl=id=page.default.default>

Kirjallisuutta

Kimppa, Kai K. (2007), *Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media*, väitöskirja, TUCS Dissertations, No 83. Osittain ladattavissa: <http://www.tucs.fi/publications/insight.php?id=phdKimppa07a>