

ICT8TD001-4

Tiedonrakentelun loppuraportti

Sisältö

Avoimeen lähdekoodiin siirtyminen julkishallinnossa.....	1
<u>1 Johdanto.....</u>	<u>1</u>
<u>2 Avoimeen lähdekoodiin siirtymisen haasteet.....</u>	<u>2</u>
<u>3 OpenOfficen käyttöönoton positiiviset puolet.....</u>	<u>4</u>
<u>4 JUHTA:n suositukset.....</u>	<u>5</u>

Avoimeen lähdekoodiin siirtyminen julkishallinnossa

1 Johdanto

Ryhmän aihe; avoimeen lähdekoodin siirtyminen julkishallinnossa oli ja on valtavan laaja ja monisyinen ja aivan mahdoton selvittää tyhjentävästi tämän lyhyehkön kurssin puitteissa.

Ryhmä ottikin tarkempaan selvitykseen muutaman näkökohdan: yleiset avoimeen lähdekoodiin siirtymisen haasteet, OpenOfficen hyvät puolet ja JUHTA:n suositukset.

2 Avoimeen lähdekoodiin siirtymisen haasteet

Avoimeen lähdekoodin siirtyminen julkishallinnossa ei missään nimessä ole ongelmatonta tai haasteetonta.

Huomionarvoista on, että avoimen lähdekoodin ohjelmistoista on liikkeellä runsaasti näkemyksiä, mikä johtunee siitä että kokemuksetkin ovat yleensä varsin rajalliset. Kuten yleensäkin, osa näistä teorioista ja näkemyksistä on vain helposti tunnistettavia virhekäsityksiä, osa tykee yleistä muutosvastaisuusteoriaa ja jotkut osuvat totuuden ytimeen.

Ryhmämme löysi esitetyistä teorioista useita haasteita ja huomioita. Kävimme läpi niistä muutamia:

- **Osaajien niukkuus.**

Avoimen lähdekoodin osaajat voivat olla harvassa.

Tämä koskee etenkin harvinaisempia tuotteita, esim. OpenOffice-tuki ei todennäköisesti tuottaisi ongelmia, kun taas käyttöjärjestelmäpuolella osaajien joukko painottuu vahvasti Microsoft Windowsin puoleen. Linux-puolen palvelinosaajia tosin saisi helposti räätälöityä vanhoista Unix-osaajista, kiitos POSIX-standardin.

- **Tietoturvariskit.**

Harva avoimen lähdekoodin tuotteita tekevä painottaa tietoturvaa riittävästi. Toisaalta ei suljettu puolikaan ole tietoturvattomuudelta vapaata.

Avoimen lähdekoodin etulyöntiasema tietoturvassa on se, että ongelmat ovat periaatteessa kenen tahansa todettavissa. Suljetussa koodissa ongelmat pysyvät jopa vuosia, jos toimittaja niin haluaa, ja esimerkkitapauksiakin on olemassa. Myöskin käsitys suljetun koodin tuottajien korkeammasta tietoturvan arvostuksesta on yleinen käsitys, jota tukee mm. erot toteutustapojen luonteessa, mikä on omiaan ruokkimaan mielikuvia. (halpa, lmainen ja vapaachtoisvoimin tuotettu = ala-arvoista)

- **Ohjelmistoja hankittaessa on erittäin tärkeää kiinnittää huomiota jatkuvuuteen.**

On hyvä tutustua tuotteen versiohistoriaan ja kehittäjäyhteisöön. Jos tuote on ollut markkinoilla kymmenen vuotta se on todennäköisesti seuraavtkin kymmenen vuotta.

Tästä hyviä esimerkkejä löytyy mm. kotimaisten kirjanpito-ohjelmien saralta, missä kehityspakkoa ei ole markkina-asemien ansiosta. Tämä on johtanut esimerkiksi siihen, että ohjelmistoja ei ole päivitetty yli vuosikymmeneen. Samanlaisia esimerkkejä löytyy myös erikoisammattien koulutusohjelmien tuottajien valikoimista.

Tällaiset tilanteet johtavat useimmiten siihen, että asiakkaan on pakko käyttää vanhentunutta järjestelmää voidakseen ajaa tarvitsemaansa sovellusohjelmaa,

sillä esim. toimittajan konkurssin jälkeen kukaan ei pysty jatkamaan tämän työtä, joka on tehty suljetun ohjelman parissa.

- **Kannattaa valita tuote, jolla on kaupallinen tuki ja sen kehittäjäyhteisö on laaja ja toimiva.**

Huomio, joka pätee niin suljettuun kuin avoimeenkin koodiin. Takeita yritysten tai muiden yhteisöjen pystyessä pysymisestä harvoin on, ja tässäkin tapauksessa tulee muistaa mahdollisuus kehittää avoimen lähdekoodin ohjelmia vapaasti – kehitys ei käytännössä voi loppua niin kauan kuin siihen halukkaita riittää.

- **On hyvä selvittää miten suljettu ja avoin ympäristö toimisivat yhdessä.**

Molemmista parhaat -politiikka voi säästää rahaa ja aikaa.

Kyse on yhteensopivuudesta, ja syitä kompromisseihin voi löytyä niukan tarjonnan erikoisohjelmista, joiden toimittajat eivät välttämättä ole valmiita laajentamaan yhteensopivuutta yhden asiakkaan toiveiden mukaan.

-

Siirtymistä avoimeen lähdekoodiin käynnistettäessä tarjouspyynnöt tulee rakentaa niin, että niihin voi antaa tarjouksen sekä avoimen lähdekoodin, että suljetun koodin tuotteilla.

Kilpailutus tulee aina hoitaa kunnolla varsinkin julkishallinnossa, jossa yhteisiä varoja käytetään yhteisiin hankintoihin.

- **Lisenssiividakko.**

Avoimen lähdekoodin lisenssiin kannattaa tutustua ennen käyttöönottoa, ne eroavat suurestikin toisistaan.

Tosiasia, joka voi olla ystävällinen tai vihollinen. Kun organisaatiolla on tarve muuttaa ohjelmia ja suojata muutokset, on ensiarvoisen tärkeää, että lisenssi sallii koodin sulkemisen. Avointa lisenssiä haluttaessa varmin vaihtoehto on useimmiten GPL 2.0. Ohjelmahankintoja miettiessä tulee tarkistaa, voiko lisenssiä vaihtaa, sillä avoin lisenssi harvoin sallii sulkemisen.

- **Yleinen muutosvastarinta.**

Nykytilan säilyttäminen on helpompaa kuin uuteen siirtyminen.

Tämä vanha totuus pätee etenkin loppukäyttäjiin, joilla muutos käyttöliittymässä saattaa aiheuttaa muutosvastarintaa. Tällöin on johdon ja IT-hallinnon tehtävä varmistaa, että muutoksen hyödyt tuodaan esille kaikkien nähtäväksi.

- **Mahdolliset säästöt eivät hyödytä henkilöstöä vaan katovat valtion tai kunnan pohjattomaan nieluun.**

Väite lienee suurelta osin tapauskohtainen, mutta säästöt kuin säästöt hyödyttävät tällöin joka tapauksessa organisaatiota laajemmalla tasolla, vaikkakin todennäköisesti yksittäisen työntekijän kannalta vähemmän näkyvästi.

Nimenomaan taloudellisen määräämisoikeuden pidätys organisaation ylemmällä tasolla saattaa olla ratkaisevaa; säästöt voidaan tällöin kohdistaa sinne missä niitä tarvitaan.

3 OpenOfficen käyttöönoton positiiviset puolet

Ryhmä otti avoimen lähdekoodin tuotteista erityistarkkailuun OpenOfficen. Seuraavassa kymmenen OpenOfficen käyttöä tukevia argumentteja:

- 1) Mahdollisuus laskea tietotekniikkakuluja. Avoimen lähdekoodin ohjelmistot ovat usein ilmaisia, jolloin syntyy selvää säästöä, jos (ja kun) lisenssimaksuihin menevät rahat eivät hupene lisääntyneeseen IT-tukeen.
- 2) Laaja tuki eri ohjelmistoalustoille.
- 3) Hankitusta ohjelmapaketista ja käyttöjärjestelmästä riippumaton, lähes universaali kielituki
- 4) Openoffice muistuttaa käyttöliittymältään enemmän MS Office 2003:a kuin Microsoftin uusi Office 2007. 2003:sta siirtyminen OpenOffice 3.0:aan voi olla loppukäyttäjien kannalta helpompaa kuin siirtyminen uuteen MS Officeen.
- 5) Toimiva PDF-formaattituki. PDF:a pidetään monesti ylivoimaisena formaattina lähes kaikenlaisiin dokumenttijulkaisuihin, sillä se säilyttää muotonsa sovelluksesta riippumatta. PDF on äärimmäisen tärkeä myös painotaloille.
- 6) Tuki useimmille avoimille sekä de jure -ja de facto -tiedostostandardeille.
- 7) Avoimia standardeja noudattavat tiedostot tulisi lain mukaan saada avatuksi julkishallinnossa. Tämä ei vielä ole kaikkialla toteutunut.
- 8) OpenOfficea voidaan käyttää MS Officen rinnalla helposti vaikkapa muistitikulta.
- 9) Kuntaliitto ja valtiovarainministeriö ovat tehneet valtion virastoille ja kunnille tarkoitetun OpenOffice-paketin. Paketti on ilmainen ja entisestään helpottaa OpenOfficen käyttöönottoa.
- 10) OpenOfficea on käytetty pohjana monille muille enemmän tai vähemmän avoimille toimisto-ohjelmapaketeille, kuten StarOffice ja KOffice. Yhteiset sukujuuret ja yhtäläinen formaattituki helpottaa ohjelmistotoimittajan vaihtamista tarvittaessa.

4 JUHTA:n suositukset

JUHTA eli julkisen hallinnon tietohallinnon neuvottelukunta on julkaissut suosituksensa - JHS 169 - avoimen lähdekoodin käytöstä julkishallinnossa. Pähkinänkuoressa suositus on seuraavanlainen:

1) huomioi ohjelmistojen jakaminen ja uudelleenkäyttö hankinnoissa.

Ohjelmisto tulisi hankkia siten, että se voidaan tarvittaessa jakaa hallinnon ja kansalaisten keskuudessa. Avoimen lähdekoodin ohjelmistot varmistavat tämän.

Näin ei aina ole, ja lopputilanne riippuu lisenssistä. Tulee myös aina muistaa, että useimmiten lisenssiä ei saa muuttaa lennossa.

2) julkaise muokatut ohjelmistot avoimella lisenssillä

Avoimeen ohjelmistoon tehdyt tai teetetyt muutokset tulisi myös julkaista avoimen lähdekoodin lisenssillä ellei kehitystyön sulkemiseen ole painavia perusteita, esim. tietoturva- tai tietosuojasyistä.

Tässä päästään juuri siihen, että ohjelmisto tulee valita huolella myös lisenssin perusteella, jos erikoistarpeita löytyy. Useimpiin avointa lisenssiä vaativiin tilanteisiin sopii GPL 2.0.

3) käytä avoimia rajapintoja ja standardeja

Avoimet standardit ja rajapinnat varmistavat suuren liikkumavaran myös tulevaisuudessa.

4) tarkista lisenssien avoimuus

Ohjelmistolisenssien tulee täyttää Open Source Initiativen määritelmä avoimesta lähdekoodista. Määritelmän mukaisen ohjelmiston pitää täyttää seuraavat vaatimukset:

- Vapaa levitys.
- Lähdekoodin saatavuus.
- Vapaus muutoksiin ja niiden levitykseen samalla lisenssillä.
- Mahdollisuus rajoittaa muunnellun lähdekoodin levittämistä, jos alkuperäisen lähdekoodin mukana toimitetaan muutoksen mahdollistavat tiedostot.
- Ei henkilöiden tai ryhmien syrjintää lisenssissä.
- Ei toimialojen syrjintää lisenssissä.
- Lisenssin pätevyys kaikkien käyttäjien kohdalla ilman erillisiä lisenssejä.
- Lisenssin tuoteriippumattomuus.
- Lisenssi ei saa rajoittaa muiden ohjelmien levittämistä.
- Lisenssin teknologiariippumattomuus.

Koske erilaisia lisenssejä on useita ja ne kaikki ovat erilaisia, tämä määritelmä ei täsmää läheskään kaikkien kanssa, mutta määritelmän ulkopuolelle jäävistä voi kuitenkin olla organisaatiolle hyötyä tilanteesta riippuen. Tältä osin suositus ei ole täysin yhtenäinen itsensä kanssa.

5) varmista riittävä asiantuntijuus käyttöönotossa

Ohjelmiston omatoimisessa käyttöönotossa on huolehdittava, että käyttäjäorganisaatiolla on riittävästi omaa osaamista ja resursseja evaluoida vaihtoehtoiset ratkaisut ja suorittaa käyttöönotto. Useimmissa tapauksissa ulkopuolisen asiantuntijapalvelun hankkiminen on suositeltavaa.

Syyt ulkopuolisen avun tarpeeseen on havaittavissa nykyisten tietoympäristöjen homogeenisyydessä, josta voidaan päätellä, että osaaminen on myös keskittynyt yhteen alueeseen.

6) kartoita ja vertaile vaihtoehtoja

Tutustu huolellisesti ja syvällisesti eri vaihtoehtoihin. Voit ottaa mallia muiden tekemistä, onnistuneista valinnoista tai etsiä yhteistyökumppania jakamaan kustannukset.

7) selvitä tietoturvariskit

Kiinnitä eritoten huomiota avoimen lähdekoodin tietoturvariskeihin, kun ohjelmistoja käytetään kriittisten toimintojen kanssa.