



UNIVERSITY  
OF TAMPERE

This document has been downloaded from  
Tampub – The Institutional Repository of University of Tampere

*Publisher's version*

Authors: Vainio Niklas  
Name of article: Vapaus ja yhteisöllisyys vapaiden ohjelmistojen liikkeen  
lähtökohtina  
Name of work: Rajoja ylittävä innovointi  
Editors of work: Koivisto Tapio, Mikkonen Teemu, Vadén Tere, Valkokari Katri,  
Ahonen Mikko, Vainio Niklas  
Year of  
publication: 2011  
ISBN: 978-951-44-8427-8  
Publisher: Tampere University Press  
Pages: 124-151  
Discipline: Natural sciences / Computer and information sciences  
Language: fi

All material supplied via TamPub is protected by copyright and other intellectual property rights, and duplication or sale of all part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorized user.

# VAPAAUS JA YHTEISÖLLISYYS VAPAI DEN OHJELMISTOJEN LIIKKEEN LÄHTÖKOHTINA

Niklas Vainio

Avoimen ohjelmistokehityksen (open source) peruseriaatteenä on ohjelmointi- ja suunnitteluprosessin avoimuus. Ohjelmistoa kehitetään avoimessa ja läpinäkyvässä yhteisössä, jotta lopputulos olisi paras mahdollinen. Toiveena on, että prosessin avoimuus tuo mukanaan sellaisia resursseja ja sellaista laatua, joka ei olisi mahdollista suljetussa prosessissa.

Ohjelmistoja on kehitetty avoimissa yhteisöissä alkaen 1960-luvulta, jolloin syntyi ns. hakkerikulttuuri ja sen myötä vapaiden ohjelmistojen liike. Erityisesti akateemisessa maailmassa ohjelmistojen avoimuus on ollut pääsääntö. Suljettujen, kaupallisten ohjelmistojen merkitys alkoi kasvaa 1980-luvulla. 2000-luvulla avoin malli on tullut jälleen suosituksi. Kamppailu *avoimen* ja *suljetun* tai *vapaan* ja *omistetun* mallin välillä on alkujaan ideologinen. Taustalla ovat kysymykset muun muassa informaation omistamisesta, käyttäjän vapaudesta ja yhteisöllisten arvojen turvaamisesta.

Ohjelmistojen omistamista koskeva kiista 1980-luvulla synnytti *vapaiden ohjelmistojen liikkeen*, joka korosti käyttäjän oikeutta päästä käsiksi tietokoneohjelmistoihin ja kontrolloida niitä. Vapaiden ohjelmistojen liikkeen pragmaattisempi siipi, *avoimen lähdekoodin ohjelmistojen liike*, korostaa ohjelmistojen vapautta ja prosessin avoimuutta edelly-

tyksenä hyvien ohjelmistojen synnylle. Tarkastelen seuraavassa niitä moraalij- ja yhteiskuntafilosofisia lähtökohtia, joiden varaan vapaiden ohjelmistojen liike on syntynyt ja jotka heijastuvat heikompiin myös avoimen lähdekoodin liikkeessä. Erityisessä tarkastelussa on Richard Stallman, joka vapaiden ohjelmistojen liikkeen johtohahmona on kiteyttänyt monia liikkeen ideologisia lähtökohtia.

## Varhainen hakkerikulttuuri vapausihanteen taustana

Tietokoneiden kehitys ja niiden merkityksen kasvu 1960-luvulta eteenpäin nosti esille uusia yhteiskunnallisia ja moraalisia kysymyksiä. Esimerkiksi tekoälytutkimukseen ja tietokoneiden tuomaan yhteiskunnalliseen muutokseen liittyi erilaisia pelkoja ja toiveita. Tietokoneellisuuden kehittymisen myötä myös kysymys ohjelmistojen omistamisen oikeutuksesta nousi esille.

Ensimmäiset tietokoneet ja niiden ohjelmistot rakennettiin 1940- ja 50-luvuilla pääasiassa länsivaltojen puolustusvoimien ja luonnontieteellisen yliopistotutkimuksen tarpeisiin. Eräs varhaisista tietotekniikkaa tutkivista yliopistoista oli Massachusetts Institute of Technology (MIT). MIT:n vuonna 1958 perustetusta tekoälylaboratoriosta tuli eräs tietojenkäsittelytutkimuksen ja samalla myös tietokonekulttuurin syntysijoista.

Steven Levy on kuvannut kirjassaan *Hackers* (1984/1996) tekoälylaboratorion tietokoneiden ympärille 1950- ja 60-luvuilla syntyneitä alakulttuuria. Koneista olivat innostuneita nuoret miespuoliset elektroniikkaharrastajat, jotka omistivat aikansa koneiden tutkimiselle ja ohjelmoinnille. Tämän alakulttuurin jäsenet alkoivat kutsua itseään *hakkereiksi* (engl. hacker), millä he tarkoittivat harrastajaa, joka nauttii tietokoneiden asettamista haasteista ja osaa ratkaista niitä nokkelilla ja yllätyksellisillä tavoilla. Hakkerille tietokone ei ole vain tekninen väline vaan myös päämäärä itsessään, sen ohjelmointiin liittyy oma estetiikkansa ja käsityöläismäinen tarkkuus ja huolellisuus. (Levy 1984/1996;

Turkle 1982; Hafner & Lyon 1996.) Myöhemmin 1980-luvulta alkaen tiedotusvälineet alkoivat käyttää hakkeri-sanaa tarkoittamaan tietojärjestelmiin murtautujaa (Suominen 1997). Hakkerikulttuurissa murtautujia kutsutaan hakkerin sijasta *kräkkereiksi* (engl. cracker). Tässä työssä käytän hakkeri-käsitettä aiemmassa, tietokonekäsitteellisen merkityksessään.

Levyn mukaan hakkeriyhteisö synnytti oman alakulttuurinsa perinteineen ja normistoineen. Yhteisössä tärkeitä arvoja olivat vapaus, älykkyys, tekniset taidot ja tietokoneiden mahdollisuuksien kunnioittaminen. Hakkerit puolestaan halveksivat byrokratiaa, hierarkkisuutta, salailua ja matemaattisen älykkyyden puutetta. Levy kiteyttää *hakkerietiikan* ytimen kuuteen sääntöön<sup>1</sup>:

- Tietokoneisiin – ja ylipäätään kaikkeen, joka voi opettaa jotakin maailman toiminnasta - täytyy olla rajoittamaton ja täysi pääsy. Kunnioita aina periaatetta ”kokeile itse”!
- Kaiken informaation tulee olla vapaata.
- Epäile auktoriteetteja – edistä vallan hajauttamista.
- Hakkereita tulee arvioida hakkerointinsa perusteella, ei hömppäkriteereillä kuten suoritetuilla tutkinnoilla, iällä, rodulla tai asemalla.
- Tietokoneella voi luoda taidetta ja kauneutta.
- Tietokoneet voivat parantaa elämää. (Levy 1984/1996, 40-45)<sup>2</sup>

Tekoälylaboratoriossa tietokoneohjelmiin suhtauduttiin kuten muuhunkin tieteellisen yhteisön tuottamaan tietoon: ohjelmien tuli olla vapaita, kaikkien käytettävissä, koeteltavissa ja paranneltavissa. Yhden ohjelmoijan tekemään ohjelmaan sai ja tuli tehdä parannuksia. Ohjelmia ei omistanut kukaan ja toisaalta ne olivat yhteisön yhteistä omaisuutta. Vähitellen hakkerikulttuuri siirtyi uusille opiskelija- ja ohjelmoijasukupolville ja toisiin yhdysvaltalaisiin yliopistoihin ja joihinkin teknisesti orientoituneisiin yrityksiin.

- 
1. Jossain määrin poikkeavan käsityksen hakkerietiikasta muotoilee Himanen (2001).
  2. Suom. Vainio ja Vadén (2006, 344).

1970- ja 80-luvun vaihteessa tietokoneet alkoivat yleistyä, ensimmäiset kotitietokoneet tulivat saataville ja alkoi syntyä ohjelmistoteollisuutta. Ohjelmien omistamista koskevissa asenteissa alkoi tapahtua muutos kun aiemmin yhteisenä pidettyjä tietokoneohjelmia alettiin käsitellä liikesalaisuuksina ja omaisuutena. Ajattelutavoissa tapahtunutta muutosta kuvaavat hyvin seuraavat kaksi konfliktia.

Kotitietokoneiden tultua saataville, niiden ympärille syntyi omia hakkeripiirejään ja käyttäjäkerhojaan, joissa harrastajat tapasivat toisiaan. Eräs varhaisista käyttäjäkerhoista oli vuonna 1975 toimintansa aloittanut Homebrew Computer Club. Kerhon piirissä oli tapana, kuten aiemmassa akateemisessa hakkeriyhteisössä, että harrastajat jakoivat tekemänsä tai hankkimansa tietokoneohjelmat toistensa kanssa tekemällä niistä kopioita. (Levy 1984/1996.)

Nuoret hakkerit, yliopisto-opiskelijat Bill Gates ja Paul Allen olivat tehneet ohjelman, joka oli harrastajapiirissä hyvin suosittu ja sen kopioita vaihdettiin paljon. Gates ja Allen eivät kuitenkaan olleet tehneet ohjelmaa tarkoituksenaan jakaa se kaikkien kanssa vaan he olivat käyttäneet sen tekemiseen paljon aikaa ja rahaa, ja tarkoituksena oli myydä ohjelmaa. Heidän näkökulmastaan oli väärin, että ohjelmaa levitettiin harrastajapiireissä ilman että tekijät saivat siitä korvauksen. Bill Gates kirjoitti harrastajille avoimen kirjeen, jossa hän suorasanaisesti tuomitsi ohjelmiston levittämisen:

Kuten suurin osa harrastajista tietää, suurin osa teistä varastaa ohjelmistonsa. Koneista maksetaan, mutta ohjelmistoja jaetaan. Kuka välittää siitä, saavatko ohjelmistojen kehittäjät korvauksen? [...]

Kenellä on varaa tehdä ammattimaista työtä korvauksetta? Kuka harrastaja pystyy käyttämään kolme miestyövuotta ohjelmointiin, kaikkien vikojen löytämiseen, tuotteensa dokumentointiin ja jakamaan sitä ilmaiseksi [for free]? [...] (Gates 1976)

Hakkerikulttuurissa tietokoneohjelmien tekemistä kauppatavaraksi vierastettiin. Hakkerietiikan lähtökohtana oli tiedon vapaus ja ohjelmointiin liittyvä luovuus ja kunnioitus. Vaikka osa hakkereista sai palk-

kansa ohjelmoinnista, työn taustalla oli pikemminkin hakkerieettinen innostus tietokoneisiin kuin protestanttisen työetiikan velvollisuusajattelu (vrt. Himanen 2001). Gates vetosi kahteen seikkaan omaisuutensa perusteena. Toisaalta hän vetosi luonnolliseen omistusoikeuteensa työn perusteella (“kuka välittää siitä, saavatko ohjelmistojen kehittäjät korvauksen?”). Toisaalta perusteena on utilitaristinen argumentti, jonka mukaan suuresta panostuksesta on saatava korvaus, sillä muuten ohjelmaa ei olisi tehty eikä vastaavan ohjelman tekeminen ole edes mahdollista harrastelijavoimin.

Gatesin kirjoitti kirjeensä itse asiassa altavastaajan asemassa. Vuonna 1976 tietokoneohjelmien tekijänoikeudellinen status oli vielä epäselvä ja siten Gatesin oli vedottavakin moraalisiin argumentteihin yksinoikeutensa puolesta. Gatesin ja muiden ohjelmistotuottajien kanta kuitenkin voitti ja ohjelmat tuotiin tekijänoikeuden piiriin vuonna 1980 (Kitsz 1980; Rosen 2005, 17). Sitten Gates ja Allen ovat tulleet tunnetuiksi menestyneen ohjelmistoyhtiö Microsoftin perustajina.

Keskeinen ohjelmistojen omistamisen kriitikko on Richard Stallman (1953–), itse ohjelmoija ja MIT:n aiempi työntekijä. Richard Stallman selittää oman tekijänoikeusfilosofiansa syntyä kuvaamalla tapahtumia 1980-luvun alussa MIT:n tekoälylaboratoriossa. Eräät laboratorion työntekijöistä perustivat Symbolics-nimisen yrityksen, joka myymiin ohjelmistoihin liittyi sopimusehto, jonka mukaan ohjelmat ovat liikesalaisuuksia eikä asiakas saa kopioida niitä eteenpäin. Suurin osa laboratorion työntekijöistä hyväksyi tämän ja siirtyi Symbolicsille töihin.

Työntekijöiden lähdettyä Stallman koki, että aiempi yhteisössä vallinnut jakamisen kulttuuri oli jäänyt uuden ohjelmistokulttuurin varjoon. Stallmanin mukaan Symbolicsin vaatimus olla jakamatta ohjelmistoa kollegoiden kanssa oli tuhonnut koko yhteisön ja sen elämäntavan. Itseään hän piti “kuolleen kulttuurin viimeisenä elonjääneenä” (Levy 1984/1996, 427; ks. myös Williams 2002).

Stallman piti ohjelmistojen omistamista eettisenä ongelmana. Tekoälylaboratoriossa oli vallinnut vahva yhteistyön ja jakamisen ilmapiiri. Ohjelmakoodi oli ollut siellä eräänlainen sosiaalisen vuorovaikutuksen

väline. Siten kaikki koodin käyttämisen rajoitukset rajoittivat samalla ihmisten mahdollisuuksia auttaa toisiaan. Stallman kuvaa joutuneensa moraalisen dilemman eteen:

Kun olin menettänyt yhteisöni, en voinut jatkaa entiseen malliin. Jouduin ankaran moraalisen valinnan eteen. Helppo ratkaisu olisi ollut liittyä omistettujen ohjelmistojen maailmaan, allekirjoittaa salassapitosopimuksia ja luvata olla auttamatta muita hakkereita. Todennäköisimmin tämä tarkoittaisi sellaisten ohjelmistojen tekemistä, jotka julkaistaisiin salassapitosopimusten alaisina, mikä lisäisi myös muiden paineita pettää toisiaan. Olisin voinut ansaita rahaa näin, ja ehkä viihdyttää itseäni kirjoittamalla koodia. Mutta tiesin, että urani lopussa olisin tarkastellut vuosia, joina rakensin ihmisiä erottavia muureja, ja olisin tuntenut käyttäneeni elämäni maailman huonontamiseen. (Stallman 1999/2002, 17)

Stallmanille ohjelmiston jakaminen kollegoidensa kanssa oli moraalinen imperatiivi; olisi väärin olla päästämättä toista hakkeria käsiksi sellaiseen tietoon, joka auttaisi häntä ongelman ratkaisussa tai josta olisi muuten vain jotain hyötyä – tai huvia. Vastatoimena *omisteisten ohjelmistojen* (proprietary software) vaikutukselle hän kirjoitti “GNU-manifestin”, jossa ilmoitti eroavansa MIT:stä ja ryhtyvänsä kehittämään vapaasti levitettävää GNU-nimistä käyttöjärjestelmää. Manifestissa Stallman vastasi kysymykseen GNU:n välttämättömyydestä seuraavasti:

Mielestäni kultainen sääntö vaatii, että jos pidän jostakin ohjelmistosta minun pitää jakaa se niiden ihmisten kanssa, jotka pitävät siitä myös. Ohjelmistojen myyjät haluavat hajoittaa ja hallita käyttäjiä, ja pakottaa jokaisen käyttäjän suostumaan olemaan jakamatta. Kieltäydyn rikkomasta solidaarisuutta tällä tavalla. En voi hyvällä omallatunnolla allekirjoittaa salassapitosopimusta tai ohjelmiston lisenssisopimusta. Tein vuosikausia työtä tekoöylaboratoriossa vastustaakseni näitä vihamielisiä pyrkimyksiä, mutta lopulta ne menivät liian pitkälle: en voinut jatkaa instituutiossa, jossa tällaisia asioita tehtiin vasten tahtoani.

Jotta voin jatkaa tietokoneiden käyttöä ilman häpeää, olen päättänyt rakentaa riittävän vapaiden ohjelmistojen kokonaisuuden, niin että pärjään ilman epävapaita ohjelmistoja. Olen eronnut tekoälylaboratoriosta, jotta MIT:lla ei olisi oikeudellista tekosyytä estää minua jakamasta GNU-järjestelmää. (Stallman 1984/2002, 32.)

GNU-manifestissa esittämiensä periaatteiden mukaisesti Stallman ryhtyi GNU-järjestelmän kehittämiseen ja perusti sitä varten vuonna 1985 Free Software Foundation -säätiön. Stallmanin näin synnyttämä *vapaiden ohjelmistojen liike* alkoi saada hakkerien parissa lisää kannatusta ja vapaasti levitettävien ohjelmistojen määrä alkoi kasvaa. 1990-luvun puolivälissä vapaat ohjelmistot alkoivat olla varteenotettavia kilpailijoita omisteisille, kaupallisesti tuotetuille ohjelmistoille, mikä herätti monien yritystenkin kiinnostuksen ja siten kasvatti vapaiden ohjelmistojen käyttäjä- ja kehittäjäjoukkoa.

## Avoimen prosessin pragmaattiset hyödyt

Vuonna 1991 Linus Torvalds, joka tuolloin opiskeli tietojenkäsittelytiedettä Helsingin yliopistossa, ilmoitti Internetin keskusteluryhmässä tehneensä hovin vuoksi alkeellisen Unix-tyyppisen käyttöjärjestelmän ytimen ja antoi sen muiden harrastajien käyttöön (Torvalds 1991). Lähinnä tietojenkäsittelytieteen ammattilaisista ja opiskelijoista koostunut yleisö otti ohjelmiston mielenkiinnolla vastaan ja alkoi tehdä siihen omia lisäyksiään kuten Stallmanin järjestelmän kohdalla aikaisemmin. Torvaldsin ohjelma ei ollut kuitenkaan kokonainen järjestelmä, vaan olakseen käyttökelpoinen se tarvitsi tuekseen GNU-projektissa kehitetyt ohjelmistot, joten Linuxiin perustuvia järjestelmiä kutsutaan toisinaan nimellä GNU/Linux (ks. Stallman 2000/2002, 51-53).

Torvaldsin tekemän ytimen ympärille rakennettu käyttöjärjestelmä alkoi saavuttaa suosiota ensin ammattilaispiireissä, mutta muutaman vuoden kuluessa se havaittiin tehokkaaksi ja laadukkaaksi ratkaisuksi



myös niiden ulkopuolella. Yritykset alkoivat ottaa vapaisiin ohjelmistoihin perustuvia järjestelmiä käyttöön. Vuonna 1995 GNU/Linux-käyttäjiä arvioidaan olleen jo miljoona, vuonna 1997 viisi miljoonaa ja vuonna 1999 kymmenen miljoonaa. Vuosikymmenen loppua kohden, dotcom-buumin keskellä, perustettiin ohjelmistoyrityksiä, joiden liiketoimintamalli ei perustunutkaan ohjelmakoodin omistamiseen vaan siihen, että tekemällä ohjelmistoja yhteistyössä yritykset ja yksityiset voivat kehittää itselleen laadukkaita työvälineitä, joihin perustuvia ratkaisuja ja osaamista ne voivat myydä. Linuxista tuli yksi uuden talouden hurmosvaiheen ikoneista ja Linus Torvalds päätyi niin talouslehti Forbesin kuin Timenkin kansijuttuun vuonna 1999. Tällä hetkellä vapaat ohjelmistot ovat omisteisten ohjelmistojen vakavia kilpailijoita ja eräiden arvioiden (esim. Asay 2005) mukaan ohjelmistoteollisuudessa on niiden johdosta menossa suuri muutos.

Vapaiden ohjelmistojen kehitysmallia seurannut Eric Raymond (1999a) huomasi 1990-luvun puolivälissä, että Linuxin kehitystyössä oli jotain erityistä. Linuxin ympärille oli kertynyt vajaan vuosikymmenessä aktiivinen ja laaja kehittäjäjoukko joka sen sijaan, että olisi yrittänyt pitää kaikki oivalluksensa ja kehitystyönsä itsellään, jakoi sen muiden kanssa. Oli huomattu, että tämä kehitysmalli tuotti usein laadukkaampia ohjelmistoja kuin ohjelmistoyritysten suosima ”katedraalimalli” – hierarkisesti järjestetty, julkaisuaikatauluihin, tarkkoihin suunnitelmiin ja salailuun perustuva tuotantotapa. Linuxin ”basaarimalli” sen sijaan oli alusta lähtien avoin ja suunnitelmaton. Kun katedraalimalli perustuu tarkkaan rajanvetoon yrityksen sisäpuolisen maailman ja ulkopuolisen kuluttajien maailman välillä, basaarissa tällaisia muodollisia rajoja ei ole. Linus Torvalds on ytimen pääkehittäjä ja hän on delegoinut joistakin sen osista vastuuta muille kehittäjille, mutta muuten kuka tahansa voi tarjota tekemäänsä muutosta liitettäväksi ytimeen.

Koska muodollinen jako kehittäjiin ja käyttäjiin on lähes olematon, basaarimainen ohjelmistokehitys on lähempänä käyttäjiään. Raymondin mukaan basaarimallin tärkeä sääntö on: ”Julkaise varhain. Julkaise usein. Kuuntele asiakkaitasi” (Raymond 1999a). Ideana on tarjota ohjelmaan tehdyt uudet ideat ja kokeilut muille kehittäjille ja käyttä-

jille tutkittavaksi, testattavaksi ja kommentoitavaksi mahdollisimman varhaisessa vaiheessa, kun taas katedraalimallissa tällaisia ohjelman ”keskeneräisiä” versioita ei missään nimessä näytetä käyttäjille ennen virallista julkaisua, joka tapahtuu kerran 1-2 vuodessa. Tämä avoimuus lienee yksi tärkeimpiä syitä basaarimallin menestykseen. Antamalla uudet ominaisuudet kenen tahansa kokeiltaviksi ja olemalla avoin palautteen suhteen ohjelmistoprojekti saa käyttöönsä testaajajoukon, jollainen ei ole katedraalimallin mukaisesti rakennetuissa ohjelmistoissa mahdollinen. Kun laaja, heterogeeninen käyttäjäjoukko kokeilee ohjelmaa ja sen uusia ominaisuuksia kukin omassa toimintaympäristössään ja suhteessa omiin tarpeisiinsa ja raportoi eteen tulleista ongelmista, ohjelmiston viat löydetään pikaisesti ja ohjelmiston laatu paranee.<sup>3</sup> Toisaalta avoimuus palvelee myös koodiin liittyvän asiantuntijuuden leviämistä ja kasvamista: usein projektiin liittyvät uudet kehittäjät ovat olleet ensin sen käyttäjiä, jotka ovat ilmoittaneet löytämistään vioista mutta myöhemmin ovat alkaneet myös korjata löytämiään vikoja. Matala kynnys kehittäjien ja käyttäjien välillä mahdollistaa ohjelmistoon tutustumisen pikkuhiljaa ja kehitystyöhön liittymisen ymmärryksen karttuessa.<sup>4</sup>

Vuonna 1998 Raymond ja joukko muita hakkereita perustivat Open Source Initiative -järjestön<sup>5</sup>. Heidän pyrkimyksensä oli saada yritysmaailman kiinnostumaan vapaista ohjelmistoista painottamalla niitä teknisiä ja taloudellisia hyötyjä, joita ohjelmistojen vapaudesta seuraa: teknistä paremmuutta, tehokkuutta ja kustannussäästöjä. Liike ei puhunut ohjelmistojen vapaudesta eikä poissulkemisen moraalista kuten Stallman, vaan heidän julkilausuttu tavoitteensa oli ”markkinoiden valloittaminen”. Helmikuussa 1998 Raymond lähetti vapaiden ohjelmistojen kehittäjille viestin otsikolla ”Goodbye, ’free software’; hello, ’open source’”, jossa hän selitti nimeen ”vapaa ohjelmisto” (free software) liittyviä ongelmia:

3. Väite laadun paranemisesta näyttäisi pitävän paikkansa myös tutkimusten valossa, ks. Mockus et al. (2000); Reasoning (2003).
4. Avoimen lähdekoodin ohjelmistokehitykseen liittyvän asiantuntijuuden erityisluonteesta ks. Vainio ja Vadén (2006).
5. <http://www.opensource.org>

Termin ongelma on kahtalainen. Ensinnäkin se aiheuttaa väärinkäsityksiä; termi ”free” on epämääräinen [...] Tarkoittaako ”free” samaa kuin ”maksuton”? Vai tarkoittaako se ”vapaasti kenen tahansa muokattavissa”, vaiko jotain muuta?

Toiseksi, termi saa yhtiöväen hermostumaan. Vaikka se ei henkilökohtaisesti haittaa minua tippaakaan, meillä on selvä käytännöllinen syy käännyttää heidät sen sijaan että näyttäisimme heille pitkää nenää. Meillä on nyt mahdollisuus isoihin saavutuksiin yritysmaailmassa ilman että joudumme luopumaan ihanteistamme tai pyrkimyksestämme tekniseen erinomaisuuteen – on aika vaihtaa asemaa. Tarvitsemme uuden, paremman nimen. (Raymond 1998)

Raymond ehdotti uudeksi nimeksi ”avoin lähdekoodi” (open source). Kaikki eivät suinkaan hyväksyneet uutta nimeä ja siihen liittyviä konnotaatioita, mutta nimi levisi ja tietoisien julkisuuskampanjan seurauksena avoimen lähdekoodin liike sai julkisuutta ensin tietokone- sitten talouslehdissä. Suuryritykset alkoivat tehdä omia avoin lähdekoodi-strategioitaan ja esimerkiksi IBM, Novell, Sun, Apple ja Nokia alkoivat tukea vapaita ja avoimen lähdekoodin ohjelmistoja tai osallistua niiden kehittämiseen.

Avoimen lähdekoodin liike aiheutti aiemmin yhtenäisessä vapaiden ohjelmistojen yhteisössä ideologisen jaon niihin, joille ohjelmistojen vapaus oli eettinen imperatiivi ja niihin, jotka eivät pitäneet omisteisia ohjelmistoja ongelmana sinänsä vaan yhtenä vaihtoehtoisena kehitys- ja lisensointimallina, jolla on omat etunsa. Jakautuminen oli kuitenkin lähinnä ideologinen eikä aiheuttanut ohjelmistojen tai yhteisön itsensä jakautumista.

Eri ideologisista lähtökohdista toimivat hakkerit kehittävät edelleen yhteistyössä samoja ohjelmistoja. Ohjelmistojen ympärille on kuitenkin syntynyt järjestöjä, yhteisöjä ja hankkeita, joiden tavoite on laajempi kuin pelkkä tekninen ohjelmistokehitys. Esimerkiksi Open Source Initiative (OSI) toimii edelleen yhteistyössä yritysten kanssa vapaan ja avoimen kehitysmallin edistämiseksi, kun taas Free Software Foundation

(FSF) pyrkii tuomaan julkiseen keskusteluun ohjelmistojen omistamisen ongelmia. Debian GNU/Linux puolestaan on parin tuhannen hakkerin ohjelmistoprojekti ja yhteisö, jolla on selkeästi julkilausutut periaatteet ohjelmistojen vapauden edistämiseksi.

## Vapaus yhteisöä ohjaavana tekijänä

Vapaiden ohjelmistojen filosofian keskeinen käsite on *vapaus*. Taustalla on klassinen liberalistinen käsitys ihmisestä vapaana yksilönä, joka on riippumaton muiden määräysvallasta. Free Software Foundationin Richard Stallman kertoo inspiraatiokseen Ranskan vallankumouksen arvot: vapauden, yhteisön ja vapaaehtoisen yhteistyön (Stallman 2001) ja vertaa vapaiden ohjelmistojen liikettä niihin. Avoimen lähdekoodin liikkeen keulakuva Eric Raymond taas tunnetaan libertaristina ja anarkistina, joka puolustaa laajaa sananvapautta ja vapaata asenkantooikeutta (Raymond 1999b; Raymond 2006).

Vapaiden ohjelmistojen liikkeen ajattelussa keskeistä on ohjelmistojen – oikeastaan välillisesti ihmisten – vapaus, jonka esteeksi nähdään tekijänoikeudelliset rajoitukset. Richard Stallmanin määritelmän mukaan ohjelmisto on vapaa, kun se toteuttaa seuraavat neljä erityisvapautta:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this. (Stallman 1996a/2002, 41).

Oikeastaan ohjelmistojen vapaudessa on kyse *käyttäjän* henkilökohtaisista vapausoikeuksista. Oikeuksista ensimmäisenä, ikään kuin lähtökohtaoikeutena (“vapaus nolla”), on oikeus käyttää ohjelmaa mihin tahansa tarkoitukseen, riippumatta siitä, mihin tarpeeseen ja minkä tavoitteiden täyttämiseksi ohjelmaa käytetään. Seuraavaksi oikeus laajennetaan pelkästä käyttämisestä oikeuteen tutkia ohjelmaa, jotta käyttäjä voisi itse olla perillä ohjelman tarkasta toiminnasta ja jotta hän voisi myös oppia siitä. Jotta käyttäjä voisi tutkia ohjelman toimintaperiaatteita, hänellä on käytännössä oltava pääsy ohjelman lähdekoodiin, jota ei normaalisti levitetä ohjelman mukana. Vapaudet 2 ja 3 edellyttävät oikeutta jakaa ohjelman kopioita niitä tarvitseville sellaisenaan tai muutettuna. Taustalla on ajatus siitä, että jakaminen sinänsä on arvokasta ja edistää yhteisöllisiä arvoja.

Usein korostetaan myös erottelua englannin kielen *free*-sanana merkitysten *vapaa* ja *ilmainen* välillä. Ilmauksella ‘free’ tarkoitetaan nimenomaan vapautta (esimerkiksi sananvapauden mielessä) riippumatta siitä, liittyykö vapauteen myös ilmaisuus. Vapaita tietokoneohjelmia voi myydä, mutta myyjä ei saa asettaa ostajalle jakamiseen liittyviä velvoitteita, toisin sanoen ostajalla on oltava oikeus jakaa ostamansa ohjelma eteenpäin haluamallaan tavalla. (Stallman 1996a/2002; Stallman 1996b/2002.)

Avoimen lähdekoodin liike on laatinut oman avoimuuden määritelmänsä (Open Source Definition), joka monessa suhteessa muistuttaa Stallmanin neljän vapauden määritelmää. Open Source Definitionin mukaan ohjelmistoa voi kutsua avoimeksi, jos se täyttää seuraavat ehdot:

- 1) Ohjelmiston on oltava vapaasti levitettävissä.
- 2) Ohjelmiston mukana tulee lähdekoodi.
- 3) Ohjelmistosta on sallittua tehdä ja levittää muutettuja versioita.
- 4) Muutettujen versioiden levittämistä voidaan rajoittaa vain siinä tapauksessa, että erillisten muutostiedostojen levittäminen on kuitenkin sallittua.

- 5) Lisenssin on sallittava ohjelmiston käyttö kaikille ihmisille ja ihmisryhmille.
- 6) Lisenssin on sallittava ohjelmiston käyttö mihin tahansa tarkoitukseen.
- 7) Levitys- ja käyttöoikeuden on siirryttävä automaattisesti uusille käyttäjille ohjelmiston mukana.
- 8) Lisenssi ei saa olla tuotesidonnainen.
- 9) Lisenssi ei saa asettaa ehtoja, jotka rajoittavat sen levittämistä yhdessä muiden ohjelmistojen kanssa.
- 10) Lisenssin on oltava teknologiasta riippumaton. (Open Source Initiative 2004)

Debian-projekti puolestaan on laatinut ”yhteisösopimuksen” (Debian Social Contract). Yhteisösopimuksessa määrätään Debian-projektin tärkeimmäksi arvoksi levitettävien ohjelmistojen pysyminen vapaana. Yhteisösopimukseen sisältyy ohjelmistojen vapauden määritelmä, joka vastaa läheisesti Open Source Definitionia. (Debian 2004.)

Free Software Foundationin vapauden määritelmän ja Open Source Initiativen ja Debian-projektin määritelmien välillä on se ero, että FSF:n määritelmä on moraalifilosofinen: FSF katsoo, että kaikkien ohjelmistojen pitäisi täyttää mainitut ehdot. OSI:n ja Debianin määritelmät puolestaan ovat teknisluontoisia ja kuvailevia: ne määrittelevät, mitä ohjelmiston vapaudella tarkoitetaan OSI:n ja Debianin toiminnan puitteissa, mutta eivät ota kantaa siihen, pitäisikö projektin ulkopuolisten ohjelmistojen olla vapaita.

## GNU General Public License oikeudellisena apuvälineenä

Tekijänoikeus antaa tietokoneohjelman tekijälle yksinomaisen oikeuden päättää, kuka hänen tekemänsä ohjelmaa saa käyttää tai levittää. Tekijänoikeusjärjestelmä ei varsinaisesti pakota ketään sellaiseen salailuun ja ”pihtaamisen logiikkaan” (Ingo 2005, 15), jonka torjuminen

oli pontimena Stallmanin GNU-projektille. Syynä on pikemminkin kulttuurin muutos: 1980-luvulla hakkerien yhteisö alkoi kasvaessaan ja kaupallistuessaan etäännyä alkuperäisestä hakkerietiikasta, johon kuului tiedon avoin jakaminen, ja tilalle tuli yksinoikeuksiin perustuva markkinaetiikka.

Tarjotakseen vaihtoehdon omisteisille ohjelmistoille, Stallman aloitti GNU-projektin tuottaakseen kilpailevan ja vapaan käyttöjärjestelmän. Koska Stallman oli omien tietokoneohjelmiansa tekijänoikeuden haltija, mikään ei olisi estänyt häntä luopumasta tekijänoikeudestaan ja luovuttamasta ohjelmiaan ”julkisen piiriin” (in the public domain), kaikkien käyttöön ilmoittaen, että tekijänoikeus ei koske niitä ja että niitä voi vapaasti käyttää, muuttaa ja levittää. Nämä ohjelmistot olisivat olleet vapaita Stallmanin vapauden määritelmän mukaan. Tämän tapaisilla ehdoilla esimerkiksi Berkeleyn yliopisto alkoi levittää omaa Unix-käyttöjärjestelmäänsä (Berkeley Software Distribution, BSD) vuonna 1989 (McKusick 1999). Berkeleyn ohjelmiston lisenssi sallii ohjelmistoa käytettävän ja levitettävän hyvin vapaasti. Levityksen ainoa ehto oli, että Berkeleyn nimi säilytetään näkyvissä ohjelmiston lähdekoodissa ja ohjeteksteissä. Siten Berkeleyn yliopisto toteutti ohjelmiston jakelussa samoja periaatteita kuin tieteellisessä julkaisutoiminnassa: pääsy tietoon on vapaata ja tiedon lähde on tunnustettava, mutta tiedon myöhemmälle käytölle ei aseteta mitään rajoituksia. Tieteellistä tietoa voidaan käyttää niin ei-kaupalliseen, kaupalliseen kuin sotilaalliseenkin tarkoitukseen. Lisenssi ei sanele, pidetäänkö tutkimustiedon jatkokehittämät salaisina vai julkistetaan ne tiedeyhteisölle. Tällaista lisenssityyppiä voidaan kutsua *akateemiseksi lisenssiksi* (Rosen 2005, 69–70).

Akateemisella lisenssillä levitettyä ohjelmaa voi kuitenkin levittää tavalla, joka rajoittaa käyttäjien vapautta, esimerkiksi olemalla levittämättä lähdekoodia ohjelman mukana. Näin alun perin vapaa ohjelmisto olisi taas tullut epävapaaksi (non-free). Kun tekijänoikeuden tarkoituksena oli suojella tekijän oikeutta määrätä kopioista (copyright), Stallman halusi luoda vastapainoksi ”käyttäjän oikeuden” (copyleft; suomenkielinen termi Vadén 2002, 27). Käyttämällä tekijänoikeuden hänelle antamaa oikeutta määrätä omien teostensa käytöstä Stallman

kehitti GNU General Public License -lisenssin (GPL), jolla suojattaisiin erityisesti käyttäjien oikeuksia (Stallman 1989; Stallman 1991). GPL on luonteeltaan *vastavuoroinen lisenssi* (Rosen 2005, 70).

GPL:n ehtojen mukaan kuka tahansa saa käyttää, muuttaa ja levittää ohjelmaa vapaasti mutta vain sillä ehdolla, että siirtää saamansa vapauden myös seuraaville käyttäjille. Lisenssillä on siten virusmainen vaikutus: viruksen tapaan se tartuttaa vapaat käyttöehdot sellaisiin ohjelmiin, joiden osaksi copyleft-ohjelma on liitetty. Tarkoitus on estää ohjelmien sulauttaminen osaksi omisteisia ohjelmistoja ja siten pitää vapaiksi tarkoitettujen ohjelmien vapaina ja toisaalta laajentaa vapaiden ohjelmistojen piiriä. (Stallman 1999/2002, 20–21; Stallman 1996b/2002b, 89–90.) Hakkeriyhteisön eettiset periaatteet saivat näin konkreettisen muodon oikeudellisena tekstinä. Lopputuloksena on ikään kuin tekijänoikeuden kääntäminen nurin: kun alkuperäinen pyrkimys oli kumota tekijänoikeuden yhteistyötä tukahduttava vaikutus, käyttämällä tekijänoikeuden suomia oikeuksia ja oikeusjärjestystä apunaan Stallman rakensi ikään kuin vaihtoehdoisen tekijänoikeusjärjestelmän, jonka tavoitteena on varmistaa ohjelmistojen mahdollisimman laaja vapaa käyttö, eräänlainen hakkerien “jokamiehen oikeus” (vrt. Vadén 2002, 9–10).

Osin Stallmanin tarjoama innostava esimerkki, osin copyleft-periaatteen toimivuus ohjelmistojen vapauden suojelijana on vaikuttanut siihen, että GPL-lisenssillä ja muilla vastaavilla lisensseillä julkaistuja ohjelmia on olemassa nykyisin tuhansia. Myös useat informaatioteknologiayritykset näkevät vapaissa ohjelmistoissa liiketaloudellisia etuja. Siten vapaiden ohjelmistojen filosofia on saanut runsain joukoin kannattajia praktisen muotoilunsa, GPL-lisenssin, kautta. GPL, oikeudellisena dokumenttina, luo oikeudelliset ja sosiaaliset puitteet, joissa vapaiden ohjelmistojen liikkeen edustama vaihtoehtoinen tekijänoikeusajattelu on mahdollista. Mielenkiintoista on, että GPL-lisenssin käyttäjiksi on tullut paljon sellaisia tahoja, jotka eivät sinänsä hyväksy vapaiden ohjelmistojen liikkeen tekijänoikeuskritiikkiä tai eivät pyri yleisellä tasolla edistämään tiedon vapautta. Monet yritykset, esimerkiksi Nokia tai IBM, ovat kiinnostuneita ohjelmistojen vapautteen perustuvan ohjel-



mistotuotannon liiketaloudellisista mahdollisuuksista. Myös vapaiden ohjelmistojen kehitykseen on liittynyt yhä useampi yritys mukaan, kun aiemmin niiden kehitys on ollut lähinnä vapaaehtoisten harteilla (Mikkonen ym. 2007).

Vaikka sekä akateemiset että käyttäjänoikeudelliset lisenssit tekevät kohdeohjelmistoistaan vapaita, ne eroavat merkittävästi siinä, kuinka ne varmistavat ohjelmistojen vapauden. Akateeminen lisenssi sallii kenen tahansa käyttävän ohjelmistoa. Esimerkiksi ohjelmistoyritys voi käyttää tällaista ohjelmistoa hyväkseen ja liittää sen osaksi omaa omisteista ohjelmistoaan. Akateeminen lisenssi ei kuitenkaan edellytä, että yritys jakaa eteenpäin ohjelmiston lähdekoodin (joka mahdollistaisi sen muokkaamisen) tai että yrityksen tekemää uutta ohjelmistoa saisi kuka tahansa käyttää. Akateeminen lisenssi luo siten hobbeslaisessa mielessä “luonnollisen vapauden” tilan: kuka tahansa voi ottaa ohjelmiston käyttöönsä, mutta häneen ei kohdistu mitään velvoitteita. Sen sijaan käyttäjänoikeudellinen lisenssi edellyttää, että ohjelmistoa hyödyntävän yrityksen on annettava eteenpäin ne samat oikeudet, jotka se sai: vapaus käyttää, muokata ja levittää eteenpäin. Käyttäjänoikeudellinen lisenssi on siten eräänlainen yhteiskuntasopimus ohjelmistojen käyttäjyhteisön kesken: se takaa kaikille tietyn vapauden ohjelmistojen käyttöön, mutta ottaa pois yhteisön vapautta rajaavan, vapaamatkustajuuden mahdollistavan “luonnollisen vapauden”.<sup>6</sup>

## Mitä ohjelmistojen vapaudella tarkoitetaan?

Stallmanin esittämä neljän vapauden määritelmä ei sinällään vielä kuvaa tarkemmin vapauden käsitettä, vaan se on pelkkä luettelo tärkeimmistä yksittäisvapauksista eli -oikeuksista. Stallmanille ohjelmiston vapaus on itseasiassa ohjelmiston käyttäjän vapautta käyttää, tutkia, muuttaa ja levittää ohjelmaa. Näiden vapauksien Stallman näkee olevan vaa-

6. Yhteiskuntasopimuksen metaforaa hyödyntää *expressis verbis* muun muassa Debian-projekti. “Debianin yhteisösopimuksessa” osallistujat muun muassa sitoutuvat pitämään projektin tuotokset aina käyttäjien vapaasti hyödynnettävinä.

rassa nykyisessä yhteiskunnassa, jossa hänen mukaansa epäsosiaalisesta käytöksestä eli kilpailusta, ahneudesta ja ulos sulkemisesta palkitaan ja ihmisten keskinäistä yhteistyötä hankaloitetaan (Levy 1984/1996, 416; Stallman 1992/2002). Kuvaus muistuttaa Hobbesin kuvausta luonnontilasta, jossa kukin kamppailee oman etunsa puolesta ja jossa yhteistyö on vain käytännön pakon sanelemaa. Hakkeriyhteisössä kukin pyrkii jakamaan tietonsa ja ohjelmistonsa: tämän kieltäminen on vakava este hakkerin vapauden toteutumislle.

### Vapaus 0: käyttövapaus

Vapauksista ensimmäinen, ”vapaus nolla”, edustaa hakkerietiikan ydinperiaatetta: pääsyn kaikkeen tietoon ja kaikkiin ohjelmistoihin tulisi olla vapaata. Berlinin (1958/2001) jaottelussa käyttövapaus edustaa varsin puhtaasti vapauden käsitteen negatiivista muotoa: tietokonetta tai tietoverkkoa ei voi käyttää ilman ohjelmistojen käyttöoikeutta ja silloin tällaisen lähtökohtavapauden puuttuminen tarkoittaa myös sulkeutumista ulos tietokoneenkäyttäjien yhteisöstä. Nollavapauden taustalla on alkuperäisen hakkerietiikan vaatimus vapaasta pääsystä kaikkeen tietoon (Chopra ja Dexter 2007, 40), mukaan lukien tietokoneohjelmien kaltainen instrumentaalinen tieto.

Vaatus käyttövapaudesta laajenee kuitenkin myös hakkeriyhteisön ulkopuolelle, ihmiskuntaan. Tietoyhteiskunnassa potentiaalisten tietokoneenkäyttäjien yhteisö on yhtä kuin globaali yhteisö. Koska kehitysmailla ei ole taloudellisia mahdollisuuksia käyttää omisteisia tietokoneohjelmia, ne ovat vaarassa tulla suljetuiksi ulos globaalista tietoyhteiskunnasta. Puhutaan ”digitaalisista kuiluista”, joita tietoteknologian ulkopuolelle jääminen synnyttää rikkaiden teollisuusmaiden ja köyhien kehitysmaiden välille. Vapaita ohjelmistoja on pidetty keirona, jonka avulla myös kehitysmaat voisivat tasapuolisesti osallistua tietoyhteiskuntakehitykseen. (Rajani 2003; Vadén ja Vainio 2005.)

Käyttövapauteen kuuluu oikeus käyttää ohjelmistoa miten tahansa, mihin tarkoitukseen tahansa. Käyttäjä on tässä suhteessa radikaalisti vapaa: ohjelmistoa voi käyttää niin hyvään kuin pahaankin, esimerkiksi tekijää vastaan. Eteen tulee vanha teknologian ja tieteellisen tiedon neutraalisuutta koskeva filosofinen ongelma: onko teknologian kehittäjällä ja tieteellisen tiedon tuottajalla vastuu teknologian käytön seurauksista? Onko ohjelmiston tekijä velvollinen lisensoimaan ohjelmistonsa siten, että sitä ei voida käyttää moraalisesti väärin tarkoituksiin?

Vapauden rajoittamattomuutta on kritisoitu siitä, että vapaiden ohjelmistojen yhteisö tuottaa ohjelmia, joiden käyttö on sallittua esimerkiksi aseellisuudessa tai ihmisoikeusloukkauksien toteuttamisessa. Voidaan kysyä, pitäisikö tällaisen eettisesti motivoituneen yhteisön pyrkiä suojelemaan muitakin arvoja kuin ohjelmistojen vapautta, esimerkiksi ihmisoikeuksia. Esimerkiksi "Common Good Public License" (CGPL)<sup>7</sup> on vaihtoehtoinen ohjelmistolisenssi, jonka idea perustuu GNU General Public Licensseen, mutta johon sisältyy vaatimus, jonka mukaan ohjelmaa ei saa käyttää toiminnassa, joka rikkoo YK:n ihmisoikeuksien julistuksen (1948) suojaamia ihmisoikeuksia. CGPL laajentaa siten lisenssin suojaamaa arvojen ja yksilöiden piiriä kaventaen samalla kuitenkin vapaus nollan alaa. Stallmanilainen näkemys olisi, että CGPL on ohjelmistojen vapauden idean vastainen.<sup>8</sup>

Colemanin (2005, xxi-xxii) mukaan vapaiden ohjelmiston liikkeen vapauskäsitys seuraa J. S. Millin versiota liberalismista. Sille on keskeinen ajatus henkilökohtaisen vapauden alueesta, jonka piirissä yksilö voi vapaasti toimia kuten haluaa, kunhan ei teoillaan vahingoita muita tai rajoita heidän vapauksiaan. Ajatus rajattomasta käyttövapaudesta on yhteensopiva tämän periaatteen kanssa: jokaisella on oikeus käyttää ohjelmistoa mihin tahansa, eikä edes ohjelmiston alkuperäinen tekijä voi rajoittaa tätä vapautta.<sup>9</sup> Käyttövapautta rajoittava ehto olisi vastoin ohjelmistojen vapauden periaatetta, joka on käyttäjän vapaus ja riippumattomuus ohjelmiston tekijästä.

7. Ks. <http://www.cgpl.org/>

8. Vrt. FSF (2005).

9. Myös avoimen lähdekoodin liikkeen Open Source Definition kieltää tällaiset rajoitukset eksplisiittisesti.

Ongelmaa voidaan tarkastella soveltaen John Rawlsin (1971/1988) ajatusta tietämättömyyden verhosta: voimme kuvitella verhon taakse ohjelmistojen käyttäjät ja kehittäjät. Millaiset ohjelmistojen käytön ja levittämisen periaatteet he päättäisivät silloin, jos kukaan heistä ei tietäisi, ovatko he verhon toisella puolella käyttäjiä vai kehittäjiä ja millaiset heidän arvonsa olisivat? Voidaan väittää, että tällaisessa tilanteessa henkilöt valitsisivat rajattoman käyttövapauden, koska eivät itse tietäisi, millaisiin tarkoituksiin he verhon toisella puolen haluaisivat ohjelmistoa käyttää. (Chopra ja Dexter 2007, 46.)

Ohjelmiston tekijän näkökulmasta käyttövapaus edellyttää, että tekijä luopuu vallastaan määrätä, mihin ja kuinka ohjelmistoa käytetään. Chopra ja Dexter väittävät, että käyttövapaus edellyttäisi myös tekijän luopumista niin sanotuista moraalista oikeuksista kuten oikeudesta tulla mainituksi teoksen tekijänä. (Chopra ja Dexter 2007, 40.) Tällainen väite ei kuitenkaan saa tukea vapaiden ohjelmistojen yhteisön käytännöstä tai lisenssitesteistä. Yhteisön käytäntöön kuuluu, että tekijöiden nimet ovat ohjelmistossa tai siihen liittyvässä dokumentaatiossa mukana. Samoin GNU General Public License suosittaa, että tekijä liittyy ohjelmistoon nimensä ja yhteystietonsa. Nimen esittäminen on siis käytäntönä eikä tekijä nähdäkseen luovu moraalista oikeuksistaan.

## Vapaus I: muuttamisvapaus

Vapaus I pitää sisällään vapauden sekä tutkia ohjelmistoa että muuttaa sitä. Tutkimisvapauden tausta on alkuperäisen hakkeriyhteisön kiinnostuksessa oppia ja tutkia, mihin tekniset ratkaisut perustuvat. Tämä on mahdollista vain, jos on pääsy tekniikan sisälle – tietokoneohjelmien lähdekoodiin. (Chopra ja Dexter 2007, 40.) Kun käyttäjä pystyy tutkimaan käyttämänsä ohjelman toimintaperiaatteita ja voi vakuuttua, ettei ohjelma tee jotain muuta kuin sen väitetään tekevän, hänellä on suurempi kontrolli omaan elämäänsä ja hän on tässä mielessä vapaampi ulkoisista pakoista.

Muuttamisvapauden kautta ohjelmistojen vapauden käsite saa positiivisen vapauden latauksen: pystyessään muuttamaan ohjelmistoa itse tarpeidensa mukaiseksi käyttäjä on riippumattomampi ohjelmiston alkuperäisestä tekijästä. Muuttaminen voi tarkoittaa sellaisen ominaisuuden lisäämistä ohjelmistoon, joka on käyttäjälle tärkeä, tai se voi tarkoittaa sellaisen virheen korjaamista, jota valmistaja ei ole ehtinyt tai ei aio korjata.

Vapaudella muuttaa ohjelmistoa voi olla myös kulttuurisiin merkityksiin liittyviä vaikutuksia. Tietokoneohjelmistot, niihin liittyvät käsitteet ja esitystavat, oikeastaan koko tietokonekulttuuri, ovat syntyneet angloamerikkalaisessa maailmassa. Tietokoneissa käytetyt metaforat, kieli ja värien symboliset merkitykset ovat sidoksissa amerikkalaiseen ja eurooppalaiseen kulttuuriin ja elämisen tapaan. (Tedre ym. 2006.) Muuttamisvapaus antaa käyttäjien yhteisöille vapauden kehittää ohjelmistoja ja paikallista ohjelmistokulttuuria lähemmäksi paikallisen kulttuurin käsitteistöä ja elämäntapaa. Vapaita tietokoneohjelmistoja on pystytty tämän vuoksi vapaaehtoisvoimin kääntämään pienille kielille tilanteessa, jossa kaupalliset toimijat eivät ole nähneet kääntämistä taloudellisesti kannattavaksi.

Muuttamisvapaus liudentaa rajaa ohjelmiston tekijän ja käyttäjän välillä (Chopra ja Dexter 2007, 40). Ohjelmistoihin liittyvää valtaa kuvaava vastakohtaispari tekijä-käyttäjä poistuu ja tilalle tulee henkilön roolia kuvaava akseli, jonka vastakkaisissa päissä (alkuperäinen) tekijä ja käyttäjä ovat. Tällä akselilla henkilö voi liikkua tarpeidensa ja kykyjensä mukaan: ottaa osittaistekijän roolin tai toimia pelkkänä käyttäjänä. Muuttamisvapaus vaikuttaa kahteen suuntaan: toisaalta käyttäjä itse muuttuu ohjelmiston muuttamiseen liittyvän oppimisprosessin myötä, toisaalta ohjelmisto muuttuu paremmin tarpeita vastaavaksi tai ainakin toisenlaiseksi (vrt. Chopra ja Dexter 2007, 40).

Muuttamisvapauden liittyviä taustaolettamia siitä, keitä ohjelmistojen vapaudesta hyötyvät käyttäjät, hakkerit tai muut yhteisön jäsenet ovat. Jotta muuttamisvapaudella olisi vapauttava merkitys, käyttäjän on kyettävä hyödyntämään hänelle tarjoutuvia mahdollisuuksia tutkia ja muuttaa ohjelmistoa. Tämä edellyttää ohjelmointitaitoa ja

muita teknisiä edellytyksiä. Toisaalta se edellyttää myös taloudellisia edellytyksiä käyttää aikaa tietokoneohjelman tutkimiseen ja muuttamiseen. Siten muuttamisvapauden vaikuttavuus on riippuvaista sosioekonomisista seikoista kuten toimeentulosta ja työnjaosta.

## Vapaus 2: levittämisvapaus

Vapaus 2 merkitsee vapautta levittää ohjelmiston kopioita. Stallmanin mukaan käyttäjällä tulisi olla oikeus levittää kopioita sellaisenaan tai muutettuna, kenelle tahansa ja missä tahansa. Oikeus ei ole riippuvainen siitä, vaatiiko käyttäjä levittämisestä rahaa vai ei, eikä hänen tarvitse pyytää levittämiseen lupaa tai maksaa tästä oikeudesta. Levittämisvapaus oikeutetaan sosiaalisilla arvoilla, naapurin tai lähimmäisen (neighbor) auttamisella. (Stallman 1996a/2002.)

Levittämisvapaus on keskeinen vapaiden ohjelmistojen liikkeen yhteisöllisten tavoitteiden toteuttamisessa. Ohjelmistojen vapaudessa ei ole kyse vain yksittäisten hakkerien tai käyttäjien vapaudesta, vaan pyrkimyksenä on mahdollistaa vapaus koko yhteisölle. Ohjelmistojen jakaminen on konkreettinen toimi, jolla tietokoneenkäyttäjien yhteisö ylläpitää keskinäistä auttamista ja yhteistyötä. Levittämisvapaus tekee siten koko yhteisön riippumattomaksi yksittäisistä ohjelmiston tekijöistä.

Rajaton levittämisvapaus on mahdollinen ohjelmiston digitaalisen luonteen vuoksi. Koska ohjelmistosta on mahdollista tehdä rajaton määrä kopioita käytännössä ilman kustannuksia<sup>10</sup>, kaikki kopioinnille asetetut rajoitukset ovat keinotekoisia ja perustuvat muihin kuin teknisiin tai aineellisiin syihin. Siitä, että digitaalista tietoa on mahdollista kopioida rajattomasti, ei tietenkään voida suoraan päätellä, etteikö kopioiminen voisi olla kiellettyä moraalisisilla tai oikeudellisilla perusteilla.

---

10. Kustannuksia tulee esimerkiksi sähköstä ja tallennustilasta, mutta käytännössä kustannukset ovat marginaaliset.

### Vapaus 3: kehittämisvapaus

Kehittämisvapaus pitää sisällään vapauden parannella ohjelmistoa ja levittää sitä edelleen. Siten kehittämisvapaus täydentää levittämisvapautta mutta laajentaa sen käsittämään myös ohjelmiston muuttamisen. Tämä on selvä laadullinen ero: kun levittämisvapaus sallii vain ohjelmiston levittämisen sellaisenaan, kehittämisvapaus pitää sisällään yhteisön hyödyn näkökulman: ohjelmistoa on lupa parantaa yhteisön tarpeiden mukaiseksi ja jakaa tästä koitua hyöty muille. Kehittämisvapaus laajentaa mahdollisten kehittäjien, parantelijoiden ja keksijöiden piiriä: mitä useampi henkilö voi parannella ohjelmistoa, sen parempi siitä tulee ja sitä suuremman hyödyn yhteisö saa.

Vapaiden ohjelmistojen liikkeen ja avoimen lähdekoodin liikkeen tärkeä keskinäinen ero liittyy siihen, mikä merkitys kehittämisvapaudelle annetaan. Vapaiden ohjelmistojen liike korostaa kehittämisvapaudteen liittyvää moraalista itseisarvoa. Avoimen lähdekoodin liikkeen argumentti on utilitaristinen: koska ohjelmisto paranee sitä enemmän, mitä useampi sitä voi kehittää, ohjelmistojen vapaudesta on yhteiskunnalle enemmän hyötyä kuin siitä, että kukin ohjelmisto on vain yhden omistajan kontrollissa (Raymond 1999a). (Vrt. Stallman 1998/2002.)

Stallman torjuu kopiointia rajoittavat argumentit viittaamalla vahinkoon, joka rajoittamisesta aiheutuu yhteistyön ilmapiirille ja yhteisölle. Hakkerietiikan perusnormi ”tieto tahtoo vapaaksi”, *information wants to be free*, perustuu ajatukselle siitä, että tiedolle on luontaista, että se on kaikkien saatavilla. Edelleen voidaan ajatella, että tieto tahtoo vapaaksi, koska vapaana sitä voidaan muokata ja jalostaa paremmaksi (Vadén 2002, 16). Hakkerietiikan päättely on luonnonoikeudellinen: koska tieto paranee vapaudessa, sen tulisi olla vapaata.

## Vapaus ja vastavuoroisuus

Vapaiden ohjelmistojen filosofiassa korostuu yhteisön merkitys: yhteisöllistä jakamisen ilmapiiriä tulee varjella ja yhteisön toimintaa tulee suojata. Toisaalta avoimen lähdekoodin liikkeelle yhteisöllä on välineellinen merkitys: mitä laajempi ja monipuolisempi ohjelmistoa käyttävä ja kehittävä joukko on, sen laadukkaampi on lopputulos. Yhteisön käsitettä käytetään eri tasoilla, tarkoittamaan toisaalta hakkerien tai tietokoneenkäyttäjien yhteisöä, toisaalta koko yhteiskuntaa. Stallman perustelee vapauden tärkeyttä viittaamalla myös käyttäjän velvollisuuteen jakaa ohjelmistoja ”lähimmäisen” (neighbor) kanssa. Debian-projektin peruskirja puolestaan on nimeltään ”Yhteisösopimus’ vapaiden ohjelmistojen yhteisön kanssa (”Social Contract” with the Free Software Community; Debian 2004).

Yhteisökäsitteen laajentuessa muuttuu myös abstraktiotaso: hakkerille ohjelmistojen vapaudesta koitua hyöty ja arvo on välitön ja kiistaton, kun taas yhteiskunta kokonaisuutena ja ihmiskunta hyötyvät siitä vain välillisesti. Stallmanin mukaan ohjelmistojen vapaus mahdollistaa sen, että käyttäjät itse voivat muokata ohjelmista juuri tarvitsemansa, turhaa työtä poistuu ja lopputuloksena on parempi yhteiskunta. Avoimen lähdekoodin liikkeelle tämä on peruslähtökohta. Stallmanin mukaan jos esimerkiksi oppikirjat olisivat vapaita, niistä olisi valtava hyöty kehitysmaiden asukkaille. Siten tekijänoikeuden poistaminen auttaisi Stallmanin mukaan ihmiskuntaakin. Abstraktiotason kasvaessa kasvaa tosin myös ajallinen perspektiivi: hakkerille hyöty on välitön, mutta jotta yhteiskunta kokonaisuutena hyötyisi tekijänoikeuden lakkaamisesta, voidaan tarvita vuosikymmeniä, koko ihmiskuntaa ajatellen kenties vielä pidempi aika. Stallmanin mukaan tekijänoikeuden poistaminen johtaisi lopulta suureen rakenteelliseen työn muutokseen:

In the long run, making programs free is a step toward the post-scarcity world, where nobody will have to work very hard just to make a living. [...] We have already greatly reduced the amount of work that the whole



society must do for its actual productivity, but only a little of this has translated itself into leisure for workers because much nonproductive activity is required to accompany productive activity. [...] Free software will greatly reduce these drains in the area of software production. We must do this, in order for technical gains in productivity to translate into less work for us. (Stallman 1984/2002, 39.)

Tätä vuonna 1984 hahmoteltua tulevaisuuden kuvaa voi pitää ylioptimistisena tai materiaalisesti mahdottomana, mutta siinä tulee esille vapaiden ohjelmistojen liikkeen tekijänoikeuskritiikin keskeinen piirre: koska ohjelmistoja voi kopioida rajattomasti nolaa lähestyvin kustannuksin, ne eivät ole luonteeltaan niukkoja niin kuin materiaaliset hyödykkeet. Sen sijaan, Stallman korostaa, digitaalisista hyödykkeistä on tarkoituksellisesti, lain keinoin, tehty niukkoja. Tästä seuraa hänen huolensa siitä, että ohjelmistojen omistajien lupa rajoittaa ohjelmien käyttöä johtaa sekä tehottomaan että epäsosiaaliseen yhteiskuntaan, jossa ihmisten halu ja mahdollisuus auttaa toisiaan ja toimia yhdessä rapautuu. Stallman ei ole sinänsä antikapitalistinen tai yrityksiä vastaan. Kirjoituksissaan hän luonnostelee tapoja, joilla ohjelmistoyritykset voisivat toimia siitä huolimatta, että niiden tekemät ohjelmistot olisivat vapaita (esim. Stallman 1984/2002, 38–39; Stallman 1996b/2002). Hän on kuitenkin huolestunut siitä, kuinka ohjelmistojen myyjät haluavat “hajottaa ja hallita” käyttäjiä asettamalla heidän välilleen yhteistyön kieltäviä muureja. Avointen ohjelmistojen liikkeen myötä niukkuusargumentti on alkanut vaikuttaa siten: suuret IT-yritykset, jotka 1990-luvulla keskittyivät ohjelmistojen myyntiin, ovat alkaneet siirtää painopistettään palvelujen myyntiin.

GNU-manifestissa Stallman viittasi “kultaiseen sääntöön” perustelunaan ohjelmistojen vapaudelle: “I consider that the golden rule requires that if I like a program I must share it with other people who like it” (Stallman 1984/2002). Kultaisella säännöllä hän tarkoittaa vastavuoroisuuden periaatetta, jonka tunnettuja muotoiluja ovat esimerkiksi “tee toisille niin kuin toivot itsellesi tehtävän” (Jeesus) ja “kohtele toisia

ihmisiä aina päämääränä itsessään” (Kant<sup>11</sup>). Vaatimus ohjelmistojen vapaudesta on siten vahvasti moraalinen vaatimus, jonka perusta on yhteisön merkityksen tunnistavassa vastavuoroisuuden etiikassa. Stallmanille läheinen yhteisö on hakkeriyhteisö ja tässä yhteisössä tärkeä ystävyyden muoto on ohjelmien jakaminen: “The fundamental act of friendship among programmers is the sharing of programs” (Stallman 1984/2002). Ideaalisen hakkeriyhteisön elämäntapaan kuuluu luoda ohjelmia ja jakaa ne toisten kanssa ja tämän kieltäminen on isku yhteisön perustavanlaatuisia lähtökohtia vastaan. Stallmanille yhteisön perustavia arvoja ovat yhteistyön henki, jakaminen ja solidaarisuus. Nämä toteutuvat ohjelmistojen jakamisessa, kun taas ohjelmistoyritykset “haluavat hajoittaa ja hallita käyttäjiä ja pakottaa jokaisen käyttäjän suostumaan olemaan jakamatta” (Stallman 1984/2002).

Avointen ohjelmistojen liikkeen motiivi puolestaan on käytännöllinen eli ohjelmistojen vapaudesta katsotaan seuraavan suurempi yleinen ja yhteinen hyöty: ohjelmistot ovat sekä parempia että halvempia ja niiden katsotaan vastaavan paremmin käyttäjäkunnan tarpeita. Ohjelmistojen neljä vapautta ovat tärkeitä oikeuksia näiden periaatteiden toteuttamiseksi. Ne mahdollistavat yhteisöllisen, avoimen kehitysprosessin. Avointen ohjelmistojen liike kuitenkin sallii myös suljetun kehitysmallin olemassaolon.

---

11. Stallman itseasiassa samaistaa kantilaisen etiikan ja kultaisen säännön (Stallman 1984/2002, 36).

## Lähteet

- Asay, Matthew N. (2005) "Open Source and the Commodity Urge: Disruptive Models for a Disruptive Development Process". Teoksessa Chris DiBona, Danese Cooper ja Mark Stone (toim.) *Open Sources 2.0. The Continuing Evolution*. O'Reilly, Sebastopol.
- Chopra, Samir ja Dexter, Scott D. (2007) *Decoding Liberation: The Promise of Free and Open Source Software*. Routledge, New York.
- Coleman, E. Gabriella (2005) *The Social Construction of Freedom in Free and Open Source Software: Hackers, Ethics, and the Liberal Tradition*. Väitöskirja, University of Chicago.
- Debian (2004) "Debian Social Contract". [http://www.debian.org/social\\_contract](http://www.debian.org/social_contract)
- FSF (2005) "The HESSLA's Problems". <http://www.fsf.org/licensing/essays/hessla.html>
- Gates, Bill (1976) "An Open Letter to Hobbyists". <http://www.blinkenlights.com/classiccmp/gateswhine.html>
- Hafner, Katie ja Lyon, Matthew (1996) *Where Wizards Stay Up Late: The Origins of the Internet*. Touchstone, New York.
- Himananen, Pekka (2001) *Hakkerietiikka ja informaatioajan henki*. WSOY, Helsinki.
- Kitsz, Dennis B. (1980) "Have the Courts Smashed Software Copyright". 80 *Microcomputing*, Vol. 10, October 1980. <http://slashdot.org/features/00/01/20/1316236.shtml>
- Levy, Steven (1984/1996) *Hackers. Heroes of the Computer Revolution*. Penguin, London.
- MacIntyre, Alasdair (1981/2004) *Hyveiden jäljillä. Moraaliteoreettinen tutkimus*. Suom. Niko Nojonen. Gaudeamus, Helsinki.
- Mockus, Audris; Fielding, Roy ja Herbsleb, James (2000) "A Case Study of Open Source Software Development: The Apache Server". *Proceedings of the 22nd International Conference on Software Engineering*. IEEE Computer Society.
- Open Source Initiative (2004) "The Open Source Definition". <http://www.opensource.org/osd>
- Rajani, Niranjani (toim., 2003) *Free as in Education. Significance of the Free/Libre and Open Source Software for Developing Countries*. Ministry for Foreign Affairs of Finland, Helsinki.
- Rawls, John (1971/1988) *Oikeudenmukaisuusteoria*. Suom. Terho Pursiainen. WSOY, Helsinki.
- Raymond, Eric (1998) "Goodbye, 'free software'; hello, 'open source'". <http://www.catb.org/~esr/open-source.html>
- Raymond, Eric (1999a) *The Bazaar and the Cathedral*. Sebastopol: O'Reilly.

- Raymond, Eric (1999b) "Why I Am An Anarchist". <http://catb.org/~esr/writings/anarchist.html>
- Raymond, Eric (2006) "Ethics from the Barrel of a Gun: What Bearing Weapons Teaches About the Good Life". <http://catb.org/~esr/guns/gun-ethics.html>
- Reasoning (2003) "How Open Source and Commercial Software Compare: Linux TCP/IP white paper". <http://www.reasoning.com/downloads.html>
- Rosen, Lawrence (2005) *Open Source Licensing. Software Freedom and Intellectual Property Law*. Prentice Hall, New Jersey.
- Suominen, Jaakko (1997) "Uhka bittiavaruudesta. Tietokonepelkojen kulttuurihistoriaa". *Kulttuurintutkimus* 4/1997.
- Stallman, Richard (2001) "The GNU GPL and the American Way". <http://www.gnu.org/philosophy/gpl-american-way.html>
- Stallman, Richard (1984/2002) "GNU Manifesto". Teoksessa Stallman (2002).
- Stallman, Richard (1996a/2002) "Free Software Definition". Teoksessa Stallman (2002).
- Stallman, Richard (1996b/2002) "Selling Free Software". Teoksessa Stallman (2002).
- Stallman, Richard (1998/2002) "Why 'Free Software' is Better than 'Open Source'". Teoksessa Stallman (2002).
- Stallman, Richard (1999/2002) "The GNU Project". Teoksessa Stallman (2002).
- Stallman, Richard (2000/2002) "What's in a Name?". Teoksessa Stallman (2002).
- Stallman, Richard (2002) *Free Software, Free Society. Selected Essays of Richard M. Stallman*. GNU Press, Boston. <http://www.gnu.org/philosophy/fsfs/rms-essays.pdf>
- Tedre, Matti; Sutinen, Erkki; Kähkönen, Esko ja Kommers, Piet (2006) "Ethnocomputing: ICT in cultural and social context". *Communications of the ACM* 49(1).
- Torvalds, Linus (1991) "Free minix-like kernel sources for 386-AT". <http://groups.google.com/groups?selm=1991Oct5.054106.4647%40klaava.Helsinki.FI>
- Turkle, Sherry (1982) "The subjective computer: A study in the psychology of personal computation". *Social Studies of Science*, 12(2).
- Vadén, Tere (2002) "Hakkerien vapaudenkaipuu". Teoksessa Vadén, Tere ja Stallman, Richard (2002) *Koodi vapaaksi – hakkerietiikan vaatuvuus*. Tampere University Press, Tampere.

- Vadén, Tere ja Vainio, Niklas (2005) "Free and Open Source Software Strategies for Sustainable Information Society". Teoksessa Hietanen, Olli (toim.) University Partnerships for International Development. Finnish Development Knowledge. Turku School of Economics and Business Administration, Turku.
- Vainio, Niklas ja Vadén, Tere (2006) "Valoa basaarista. Internetin vapaan ja avoimen koodin kollektiivinen kehitystyö". Teoksessa Parviainen, Jaana (toim.) Kollektiivinen asiantuntijuus. Tampere University Press, Tampere.
- Williams, Sam (2002) Free as in Freedom. Richard Stallman's Crusade for Free Software. O'Reilly, Sebastopol.