

arvosana

päiväys

arvostelija

AVOINTEN PROJEKTtien RYHMÄT JA YHTEISÖT

Kalle Pyykkönen

21.10.2008

Seminaari: Avoin ohjelmistokehitys

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisällys

1	Johdanto	1
2	Avoimen ohjelmistokehityksen piirteitä	1
3	Avointen projektien yhteisöjen sosiaalinen rakenne	2
4	Avointen projektien yhteisöt organisaatioina.....	5
5	Yhteenveto	6
6	Lähteet.....	7

1 Johdanto

Avoimen ohjelmistokehityksen puolestapuhujien mukaan se on perinteistä ohjelmistokehitystä parempi tapa tehdä laadukkaita ohjelmistoja. Monet avoimen ohjelmistokehityksen vahvuudet perustuvat laajaan ja aktiiviseen kehittäjäyhteisöön.

Tässä työssä kuvataan avointen projektien organisaatioiden piirteitä, esitellään avointen projektien yhteisöjen sosiaalista rakennetta ja lopuksi käsitellään yhteisöjä virtuaalisina organisaatioina johtamisviestinnän näkökulmasta.

2 Avoimen ohjelmistokehityksen piirteitä

Avoimen ohjelmistokehityksen käytännöt ovat kirjavia. Ei ole olemassa mitään tiettyä ”open source -prosessimallia”, jota avoimen ohjelmistokehityksen projektit noudattaisivat. Takavuosien avoimen lähdekoodin ideologiaa muokanneet kirjoitukset, kuten Eric S. Raymondin *the Cathedral and the Bazaar* [RAY00] luettelevat joitakin yleisiä avointa ohjelmistokehitystä leimaavia piirteitä. Avoimen ohjelmistokehityksen vertaaminen baasarin hulinaan viittaa osallistujien määrään, toiminnan näennäiseen organisoimattomuuteen ja kehittäjien erilaisiin tavoitteisiin, sekä lähestymistapoihin.

Myös tarkastelemalla menestyksekkäitä avoimia projekteja voidaan löytää niille yhteisiä ominaisuuksia. Paljon tutkittuja avoimia projekteja ovat mm. Linux-käyttöjärjestelmäydin, Mozilla-selain ja Apache-webpalvelin.

Avoin ohjelmistokehitys on luonteeltaan enemmän rinnakkaista kuin lineaarista. Rinnakkaisen työskentelyn mahdollistaa monien avoimien tuotteiden modulaarinen rakenne. Avointen projektien kehittäjien joukko on lukumääräisesti suuri, jopa satoja tai tuhansia henkilöitä ja kehitystyö on maailmanlaajuisesti hajautettua. [FeF01]

Raymondin mukaan riittävän suuri beta-testaajien ja kehittäjien määrä takaa sen, että melkein kaikki ongelmat kyetään kuvaamaan nopeasti ja joku kykenee löytämään siihen ratkaisun. Tätä Raymond kutsuu Linuksen laiksi. [Ray00]

Avointen projektien organisointi poikkeaa merkittävästi perinteisestä ohjelmistokehityksestä. Työtehtäviä ei osoiteta työntekijöille projektin johdon toimesta, vaan työntekijät voivat itse valita haluamansa tehtävät. Projekteilla ei ole välttämättä minkäänlaista projektisuunnitelmaa tai aikataulua. Kaikissa menestyksekkäissäkin projekteissa ei ole edes kirjattua arkkitehtuurisuunnitelmaa. On kuitenkin hyvä huomata, että nykyään monet avoimet projektit ovat liikeyritysten hallinnassa ja osallistujat eivät ole vapaaehtoisia ja tilanne on toinen. [FeF01], [MFH02]

Tarkan organisoinnin ja suunnittelun sijaan avoin ohjelmistokehitys perustuu kehittäjäyhteisön suureen kokoon ja aktiiviseen osallistumiseen. Myös avointen tuotteiden käyttäjät osallistuvat aktiivisesti lähettämällä palautetta ja vikaraportteja ja saavat vastineita palautteeseensa nopeasti. Usein oletetaan, että avoimuus parantaa kehittäjien motivaatiota ja parempi motivaatio oletettavasti lisää tuottavuutta. [MFH02]

Hajautettu malli mahdollistaa eritasoisten kehittäjien osallistumisen. Kukin kehittäjä voi työskennellä omalle tasolleen sopivan tehtävän parissa. Tutkittaessa perinteistä ohjelmistokehitystä on havaittu, että jopa 20% kehittäjistä antaa hankkeelle itse asiassa negatiivisen työpanoksen sitomalla kokeneempien kehittäjien resursseja. [FeF01]

Brooksin laki on yksi siteeratuimpia ohjelmistotekniikan lainalaisuuksia. Sen mukaan uusien työntekijöiden lisääminen myöhässä olevaan projektiin saa projektin myöhästymään lisää. Avoimen lähdekoodin projekteissa tämä ei kuitenkaan näyttäisi pätevän. [Ray00]

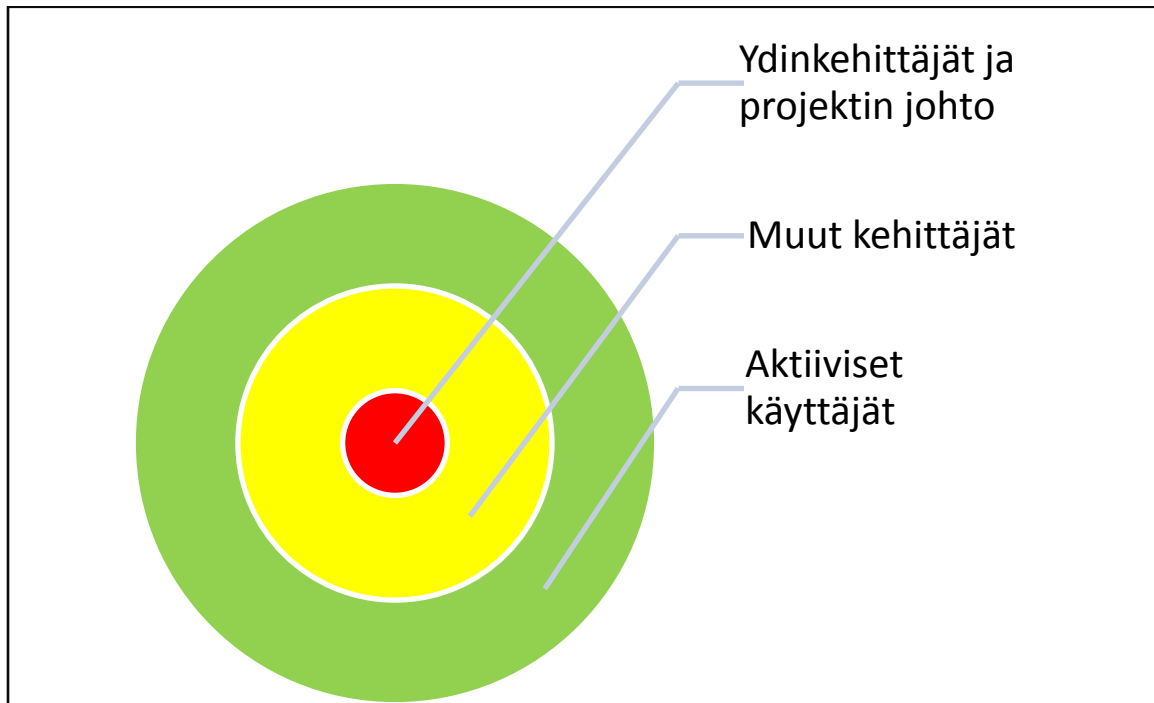
3 Avointen projektien yhteisöjen sosiaalinen rakenne

Monet avoimen ohjelmistokehityksen vahvuudet liittyvät kiinteästi avoimen lähdekoodin yhteisöjen sosiaaliseen rakenteeseen. Terve kehittäjäyhteisö onkin, itse tuotteen laatukriteerien ohella, yksi tärkeä kypsyyssmittari, jota voidaan käyttää esimerkiksi avoimen lähdekoodin tuotteen kaupallista potentiaalia arvioitaessa.

Avoimen lähdekoodin projekteja on valtava määrä. Niistä suurin osa on kuitenkin yhden kehittäjän projekteja. Tässä kuvattava sosiaalinen rakenne sopiikin kuvaamaan vain mur-

to-osaa yli sadasta tuhannesta avoimen lähdekoodin projekteista. On tutkittu, että n. 1% avoimista projekteista ylittää koskaan kymmenen kehittäjän rajan [CRH05].

Avointen projektien yhteisöistä voidaan löytää muutamia perusrooleja. Rooleja kuvataan usein halkileikattua sipulia muistuttavalla kehämallilla.



Kuva 1: Kehämalli

Kuvassa 1 olevan kehämallin keskiössä on *kehittäjien ydinryhmä*, johon kuuluvilla kehittäjillä on lähdekoodin versionhallintajärjestelmään lisäys- ja muokkausoikeudet. Ydinryhmän koko on tavallisesti kolmesta kymmeneen henkilöä. Jos ydinryhmä on tätä suurempi, ryhmän sisäinen viestintä vaikeutuu huomattavasti ja yhteydenpito vie aikaa päätehtävältä, eli järjestelmän kehittämiseltä. [CRH06]

Myös *projektin johto* on kehämallin keskiössä. Vahva johtaja tai johtajien ryhmä auttaa projektia selviämään ja kehittymään vakauteen ja kypsyyteen. Avoimen lähdekoodin projektien oletetaan usein olevan demokraattisesti johdettuja. Oletuksen vastaisesti näyttää kuitenkin siltä, että projektien johtajat tekevät usein päätöksiä itsenäisesti, yhteisöä kuulematta. Autokraattinen johtamistyyli ei välttämättä ole huono asia, jos päätöksillä on

yhteisön tuki. Toisaalta omavaltainen johtaminen voi vähentää osallistumista ja estää projektin kasvun. [CRH05]

Projektin vetovastuun siirtyminen perustajalta eteenpäin voi olla hankalaa, koska vaikka perustaja on siirtynyt muodollisesti syrjään, hänen sanomisillaan on edelleen merkittävä painoarvo. Avointen projektien vetovastuun vaihtuminen onkin varsin harvinaista. [CRH06]

Yleensä projektin perustajalla ajatelleen olevan subjektiivinen oikeus päättää viime kädessä uusien versioiden julkaisemisesta. Tämä oikeus on ns. kirjoittamaton laki, sillä monet avoimet lisenssit antavat saman oikeuden kaikille.

Ydinryhmään kuuluu myös julkaisukoordinaattori, joka kokoaa lähdekoodista julkaistavan versio päivityksen. Julkaisukoordinaattori päättää milloin kaikki kriittiset viat on kunnossa ja uusi versio on valmis julkaistavaksi. [YeK05]

Kehämallin keskiötä ympäröi *muiden kehittäjien* joukko. Muut kehittäjät kirjoittavat koodia, mutta heillä ei ole suoraa lisäys-oikeutta lähdekoodiin. Tavallisesti he lähettävät korjauksia kehittäjien ydinjoukon arvioitavaksi. Tässä ryhmässä kunnostautuminen voi johtaa kehittäjän siirtymiseen ydinkehittäjien joukkoon. [CRH06]

Kehämallin seuraavalla kehällä ovat *aktiiviset käyttäjät*. Aktiiviset käyttäjät testaavat uusia julkaisuja, lähettävät vikaraportteja ja kirjoittavat projektin dokumentaatiota. Vikaraporttoijat ovat yksi aktiivisten käyttäjien osajoukko [YeK05].

Yksi aktiivisten käyttäjien tärkeimmistä tehtävistä on muodostaa vertaisryhmä, joka auttaa passiivisia käyttäjiä peruskysymyksiin, kuten asennukseen, käyttöönottoon ja asetuksiin, liittyvissä ongelmissa. Tällä tavalla itse kehittäjät säästävät tältä usein turhauttavaltakin työltä ja voivat keskittyä koodin kirjoittamiseen. [CRH06]

Varsinaisen kehämallin ulkopuolelle jää epämääräinen joukko *passiivisia käyttäjiä*, jotka käyttävät ohjelmistoa, mutta eivät osallistu yhteisön toimintaan millään tavalla.

Avoimen lähdekoodin ohjelmistokehityksessä ei useinkaan ole kiinteitä resursseja, toisin kuin perinteisessä suljetun lähdekoodin sovelluskehityksessä. Yhteisö elää, siihen on helppo liittyä ja siitä on helppoa lähteä. Samoin osallistujien käytettävissä oleva aika ja

osaamistaso vaihtelevat. Noissa olosuhteissa projektin aikatauluttaminen, resurssien allokoiminen ja yleensäkin projektin organisoiminen tehokkaaksi voi olla vaikea tehtävä. Usein kehittäjiä kiinnostaa enemmän luova työskenteleminen ja uuden tekeminen. Enemmän puurtamista muistuttavat koodin kasaaminen julkaisuiksi ja testaaminen julkaistavaksi versioksi kiinnostaa vähemmän. [CRH05]

Kehämalli on näkökulmaltaan lähdekoodikeskeinen. Se johtuu oletettavasti siitä, että useissa tutkimuksissa lähdeaineistona on käytetty lähdekoodin versionhallintajärjestelmästä saatua metatietoa. Projektissa voi olla sen menestyksen kannalta keskeisiä henkilöitä, jotka lähdekoodikeskeinen tarkastelutapa sijoittaa kehämallin ulkokehälle.

4 Avointen projektien yhteisöt organisaatioina

Hajautetun ja globaalin luonteensa vuoksi avoimissa projekteissa käytetään Internet-pohjaisia tieto- ja viestintätekniikan välineitä [MFH02]. Lähdekoodien hallinnassa käytetään versionhallintajärjestelmiä (CVS) ja vikailmoitusten hallintajärjestelmiä (Bug tracker). Dokumentoinnissa käytetään wikejä ja viestinnässä sähköpostilistoja, IRC:ia ja blogeja. Avointen projektien organisaatiot ovat siis hyvin pitkälti virtuaalisia organisaatioita.

Virtuaalisen viestinnän ominaisuuksia ovat mm. tehtäväorientoituneisuus, aikaasäästävyys, mutta toisaalta huono konfliktinratkaisukyky ja johtajakeskeisyyden heikentyminen. [Åbe2005]

Pekka Aulan kehittämän johtamisviestinnän areenamallin mukaan erilaisille viestintätarpeille organisaatiossa tulisi olla erilaiset areenat. Esimerkiksi joihinkin tilanteisiin sopii kirjallinen tiedottaminen, joihinkin puhelu ja jotkut toiset tilanteet vaativat puolestaan kasvokkaisviestintää. Virtuaalisesta organisaatiosta monet areenoista puuttuvat kokonaan tai osittain. [Åbe2005]

Virtuaalisen viestinnän vihjeettömyys, siitä puuttuvat ilmeet, eleet ja äänenpainot, lisäävät viestinnän konfliktiherkkyttä. Vajavainen tunneviestintä voi johtaa myös osaltaan äärimmäisiin tunnereaktioihin, ”fleimaamiseen”, koska sanattoman viestinnän sosioemo-

tionaaliset vihjeet puuttuvat. Konfliktien ratkaisussa kasvokkain tapahtuva viestintä olisi tehokkain keino. [Mon03]

Virtuaalisessa organisaatiossa konfliktien välttämässä ja viestinnän sujumisessa auttaa projektiin osallistujien vapaaehtoisuus ja heidän jakamansa yhteiset arvot. Yhteisen arvomaailman muodostumisessa oleellisessa osassa ovat aiemmin mainitut avoimen lähdekoodin pioneerien kirjoitukset.

Avoimen lähdekoodin projektien yhteisöt ovat muodostaneet oman uuden kulttuurinsa, jolla on omat, ei välttämättä julkilausutut, mutta jaetut käytännöt ja ilmauksensa. Myös muodostunut kulttuuri auttaa korjaamaan virtuaalisen viestinnän vihjeettömyydestä johtuvia ongelmia. Virtuaalisten viestintäkanavien taidokas käyttö voi myös paikata puuttuvia areenoita. [Åbe05]

Suurissa avoimen lähdekoodin projekteissa tietotulva vaikeuttaa osallistumista. Esimerkiksi projektin postituslistalle voi tulla satoja viestejä päivässä. Täydellinen osallistuminen on käytännössä mahdotonta, osallistuminen onkin usein luonteeltaan ajoittaista ja lyhytaikaista.

5 Yhteenveto

Parhaimmillaan avoimet projektit tarjoavat kehittäjille matalan kynnyksen osallistua suurtenkin järjestelmien ohjelmistokehitykseen. Osallistua voi monella tavalla, kukin oman kiinnostuksensa ja osaamistasonsa mukaan. Menestyksekkäät projektit ovat osoittaneet, että avoin ohjelmistokehitys voi tuottaa yhtä hyviä tai parempia tuloksia kuin perinteinen suljettu ohjelmistokehitys.

Toimivia yhteisöjä on muodostunut kuitenkin ainoastaan murto-osaan avoimista projekteista. Valtaosa avoimista projekteista on yhden tai muutaman kehittäjän varassa. Lisäksi osa avoimen lähdekoodin tuotteista on liikeyritysten hallussa, eikä niitä kehitetä välttämättä avoimesti. Samaan aikaan monet menestyneiden avoimen ohjelmistokehityksen projektien käytännöistä ja menetelmistä on otettu käyttöön myös suljetun lähdekoodin projekteissa. Herääkin kysymys onko avoin lähdekoodi vs. suljettu lähdekoodi oikea kon-

teksti tarkastella näitä ohjelmistotekniikan ilmiöitä vai pitäisikö niitä tarkastella esimerkiksi hajautetun tai ketterän ohjelmistokehityksen kontekstissa?

6 Lähteet

CRH06 K. Crowston, J. Howison: Assessing the Health of Open Source Communities. (IEEE) Computer 39, 5, 2006, 89-91.

CWL06 K. Crowston, K. Wei, Q. Li, J. Howison: Core and Periphery in Free / Libre and Open Source Team Communications. In: Proc. 39th Annual Hawaii International Conference on System Sciences (HICSS'06). IEEE Computer Society, 2006, 118-124. (IEEE Xplore: 118a)

CRH05 K. Crowston, J. Howison: The social structure of free and open source software development, First Monday, 2005

FeF01 J. Feller, B. Fitzgerald: Understanding Open Source Software Development, Addison-Wesley Professional 2001

MFH02 A. Mockus, R.T. Fielding ja J.D. Herbsle: Two Case Studies of Open Source Software Development: Apache and MozillaACM, Transactions on Software Engineering and Methodology 2002.

Mon03 M. Montola: Organisaatiot ja johtaminen kyberavaruudessa, Pro gradu -tutkielma, Helsingin yliopiston viestinnän laitos 2003

Ray00 E.S. Raymond: The Cathedral and the Bazaar, <http://www.catb.org/~esr/writings/cathedral-bazaar/>

YeK05 Y. Ye, K. Kishida: Toward an Understanding of the Motivation of Open Source Software Developers. In: Proc. 25th International Conference on Software Engineering, Portland, Oregon, 2005. IEEE Computer Society, 2005, 419-429.

Åbe05 L. Åberg: esimiesviestinnän erikoiskurssin luentomateriaali 2002, <http://www.valt.helsinki.fi/staff/aberg/esv05.htm>