

Saimaan ammattikorkeakoulu
Liiketalous Lappeenranta
Liiketalouden koulutusohjelma
Yrityshallinnon juridiikka

Juha Kohvakka

AVOIN OHJELMISTONKEHITYSPROJEKTI LIIKETOIMINNAN ASiantuntijan silmin

Opinnäytetyö 2012

TIIVISTELMÄ

Juha Kohvakka

Avoim ohjelmistonkehitysprojekti liiketoiminnan asiantuntijan silmin, 60 sivua

Saimaan ammattikorkeakoulu, Lappeenranta

Liiketalouden koulutusohjelma

Yrityshallinnon juridiikka

Ohjaaja: Lehtori Samuli Nikkanen, Saimaan ammattikorkeakoulu

Tässä opinnäytetyössä syvennyttään avoimen ohjelmistonkehitysprojektin toimintaan mm. liike-elämän yritysjohton, projektityön ja henkilöstöhallinnon näkökulmasta. Case-projekti OpenTTD on avoimen lähdekoodin peliprojekti, jossa kirjoittaja on ollut kirjoitushetkellä seitsemän vuotta. Tutkimus pohjautuu case-projektin ja muiden avoimien ohjelmistoprojektien tarkasteluun. Taustalla ovat liiketalouden alan erilaiset teoriat ja tutkimustulokset.

Työssä ehdotetaan, että avoimessa ohjelmistonkehityksessä toimivia koordinaattoreita pidettäisiin vakavasti otettavina johtajina siinä missä esimerkiksi liikealan projektikoordinaattoreitakin. Väite perustuu Fayolin teoriaan johtamisen perustehtävistä. Liiketoiminnalle ominaisia piirteitä tunnistetaan case-projektista, ja pohditaan myös liiketoiminnan käsitteen merkitystä. Työssä verrataan tyypillisen ohjelmistoalan yrityksen ja avoimen ohjelmistoprojektin organisaatorakenteita ja todetaan, että tarpeelliset roolit ovat yhtenevät. Eroavaisuus näyttää olevan pääasiassa palkkauksessa. Avoimen ohjelmistonkehityksen parissa tehdään töitä harrastuksena ilmaiseksi, kun taas yrityksissä työ on perinteisempää.

Tutkimuksen loppupuolella tarkastellaan ihmisten asemaa ja vuorovaikutusta avoimissa ohjelmistonkehitysprojekteissa. Todetaan, että projektit syntyvät liikeyrityksen tavoin yksilön tai pienen ryhmän ajatuksesta ja niiden edetessä esiintyvät Tuckmanin ryhmän muodostumisen vaiheet. Lisäksi huomautetaan, että Tuckmanin teorian vaiheet esiintyvät avoimessa ohjelmistonkehityksessä eri järjestyksessä. Lisäksi työssä tarkastellaan avoimien projektien henkilöstön motivaattoreita, rooliutumista ja normistoa.

Tutkimuksen tuloksena ovat jatkotutkimuskohteet ja kehityskohteet akatemisille toimijoille, liikkeen- ja projektijohton ammattilaisille sekä avoimissa ohjelmistoprojekteissa toimiville. Työn loppupuolella suositellaan mm. avoimen lähdekoodin ohjelmistoissa huomattavaa tarvetta laadukkaammalle tuotteistukselle ja viimeistelylle. Toisaalta suositellaan, että liiketoiminnassa ajettaisiin olemassaolevaan toimintaan avoimuutta esimerkiksi tuotekehitys- ja asiakaspalveluprosesseihin.

Asiasanat: IT-ala, pelit, avoin lähdekoodi, vapaaehtoistyö

ABSTRACT

Juha Kohvakka

Open Source Software Development, a Business Perspective, 60 pages

Saimaa University of Applied Sciences

Degree Programme of Business Administration, Lappeenranta

Business Law

Bachelor's Thesis 2012

Instructor: Senior Lecturer Samuli Nikkanen, Saimaa University of Applied Sciences

This Bachelor's Thesis takes an in depth look at open source software development from the perspective of traditional profit-based business operations. OpenTTD, an open source logistics company simulator, is used as a case project. The thesis is based on observing the case project and other open source software projects. Various business theories, such as Fayol's theory of basic management functions, are used as a foundation for the work.

It is suggested, based on Fayol's theory, that coordinators in open source software development can be considered proper managers. Similar features are noted to exist in both free, open source software development and traditional profit-based software businesses. Roles in both kinds of organizations are compared and it is noted that the needed roles are very similar. The difference appears to be mostly in salaries, as the majority of people appear to work in open source software development for free, as a hobby, whereas business operations form a more traditional job.

The latter chapters focus on people's position and interaction within open source software groups. It is noted that both open source software groups and businesses initially form as a single person's, or a small group's project. Tuckman's stages of group development are noted to make an appearance in both cases. It is suggested, however, that the order of the stages is different in open source software groups. Finally, the motivators, roles and norms of open source groups' workforce are observed.

The thesis concludes with a firm belief that open source software development and software development businesses feature a multitude of similarities and should, in some cases, be compared side-by-side. Avenues for further research and organizational improvement are suggested for academics, business professionals and open source enthusiasts alike. For example, it is noted that productization and product polishing requires attention in open source software development. On the other hand, it is proposed that businesses place more emphasis on openness in both their inner workings such as product development, and exterior processes like customer relations.

Keywords: Information Technology, Games, Open Source, Volunteer Work

SISÄLTÖ

1 JOHDANTO.....	6
2 OPINNÄYTETYÖN KONTEKSTI.....	8
2.1 Open source, avoimen lähdekoodin ohjelmistokehitys.....	8
2.2 OpenTTD-peli ja case: pelin grafiikkakehitysprojekti.....	11
2.3 OpenTTD-pelin lyhyt historia.....	12
2.4 Sivullisesta OpenTTD-yhteisön jäseneksi.....	14
3 AVOIN OHJELMISTOPROJEKTI BUSINESS-NÄKÖKULMASTA.....	17
3.1 Liiketoiminnan piirteitä avoimessa ohjelmistonkehityksessä.....	18
3.2 Organisaatio.....	21
3.2.1 OpenTTD-organisaatio.....	24
3.2.2 Grafiikkaprojektin organisaatio.....	27
3.2.3 Henkilökuntana yhteisö.....	29
3.2.4 Työnjako ja vastuutus.....	30
3.3 Tuote	32
3.4 Toiminta-ajatus, arvot ja visio ohjaavat strategjaa.....	36
3.5 juridiikka	41
3.6 Asiakaskunta.....	43
4 IHMISENÄ AVOIMESSA OHJELMISTOPROJEKTISSA.....	44
4.1 Motivaatio.....	45
4.2 Roolit.....	48
4.3 Normit.....	49
5 YHTEENVETO.....	51
KUVAT.....	58
LÄHTEET.....	59

TERMISTÖÄ

avoin lähdekoodi, open source

toimintatapa, jossa tietokoneohjelman lähdekoodi on kenen tahansa saatavilla ja jossa usein kannustetaan kehittämään ohjelmiston virallista julkaisua

vapaa ohjelmisto, free software

tietokoneohjelmisto, jonka kehityksessä noudatetaan ääriiberaalia filosofiaa, jonka peruskivet ovat avoin lähdekoodi, avoin kehitystyö, sekä alhainen tai olematon hankintahinta loppukäyttäjälle

lähdekoodi

tietokoneohjelman ”tekninen perusta”, ikään kuin rakennuksen piirustukset

Y-sukupolvi

1970–1990-luvuilla syntyneiden sukupolvi

tiketti

vika-, kehityskohde-, ongelma-, yms. seurantajärjestelmissä yhtä kohdetta vastaava lipuke

binäärimuotoinen komponentti

grafiikkaa tai ääntä

GNU

erisnimi: **GNU is Not Unix**, käyttöjärjestelmäprojekti, joka on tuottanut tuloksenaan useita vapaita ja ilmaisia ohjelmistoja, sekä GNU General Public License -ohjelmistolisenssin

wiki

yleisnimi: vapaasti muokattavissa oleva tekstijärjestelmä

1 JOHDANTO

Avoimen lähdekoodin ohjelmistokehitys on ilmiö, joka on ollut noin puolitoista vuosikymmentä tärkeä sektori tietokoneohjelmien kehityksen valtavirrassa. Perusajatus on ollut it-alalla käytössä jo puoli vuosisataa. Tiedon ja prosessien jakaminen ja avoimuus taas on ollut ihmiskunnan toimintaa tukeva filosofia satoja vuosia. Esimerkiksi ruuanlaitossa avoimuutta on harrastettu kulttuurillisesti kehittyvän ihmiskunnan alusta asti.

Nyt eletään aikakaudella, jossa avoimuus ja tiedon vapaa virtaus korostuvat. Avoimen lähdekoodin ohjelmistokehitys on aivan omassa elementissään tällaisella avoimuuden aikakaudella. Tämän vuoksi avoin lähdekoodi on nyt kuuma puheenaihe mediassa, tietokoneiden ja kuluttajaelektronikan käyttäjäkunnan keskuudessa, sekä yksityisen ja julkisen sektorin organisaatioissa.

Liikeryityksillä on kuitenkin usein taipumus karttaa avoimuutta – onhan niiden etujen mukaista säilyttää ns. liikesalaisuudet. Avoimien ohjelmistoprojektien tuotteet kuitenkin kilpailevat jo aivan vakavissaan kaupallisten ohjelmistojen kanssa. Liiketalouden opintojen opinnäytteeksi tutkin, miltä avoimen lähdekoodin ohjelmistoprojekti näyttää liiketalouden ammattilaisen silmin. Tavoitteenani on tunnistaa yhtäläisyyksiä, eroavaisuuksia ja miksei kehityskohteitakin avoimessa ohjelmistonkehityksessä sekä ohjelmistoalan yritysten toiminnassa. Käytän esimerkkiprojektina peliprojektia nimeltä OpenTTD, jossa olen ollut mukana eri rooleissa vuodesta 2005.

Stephanie Freeman Helsingin yliopiston käyttäytymistieteiden laitoksesta julkaisi vain kolme kuukautta ennen tämän opinnäytetyön aloitusta väitöskirjansa ”Constructing a Community”. Hän syventyy teoksessaan nimenomaan käyttäytymistieteiden ja ihmisaspektin tieteelliseen analysointiin avoimen ohjelmistoprojektin kontekstissa. Itse pyrin tässä opinnäytetyössä lähestymään avoimen lähdekoodin ohjelmistokehitystä liiketaloudellisen ajattelun näkökulmasta.

Tutkimusmetodologia

Avoin ohjelmistonkehitys ja ns. *avointen ohjelmistojen liike* ovat varsin uusia käsitteitä. Aiheista on tehty yllättävän vähän tutkimuksia. Lisäksi on hyvin vähän liiketalouden alan kirjallisuutta, joka ottaa nimenomaan avoimet ohjelmistoprojektit huomioon tarjotuissa teorioissa ja case-analyseissa.

Harrastetausta ja liiketalouden koulutus antavat minulle sopivat lähtökohdat tutkia aihetta omiin kokemuksiini pohjautuen. Tutkimuskysymys kuitenkin on mielestäni sen luonteinen, ettei empiria ole avain siihen vastaamiseen. Kysymys on enemmän aiheesta, jonka tutkimiseen täytyy soveltaa vanhoja eri ympäristöissä käytettyjä teorioita, ja ajoittain vastaan tulevat sivujuonteet ulottuvat jopa filosofisen ajattelun puolelle.

Ns. case-tutkimus on mielestäni oikea metodologia tällaiseen aiheeseen syventymiseen. Teorioihin ja filosofiseen ajatteluun ei voi uppoutua täysin, vaan on tutustuttava välillä konkreettisiin esimerkkeihin siitä, kuinka toimitaan avoimessa ohjelmistonkehityksessä tai liike-elämässä. Seitsenvuotinen taipaleeni case-projektissa on tässä avuksi. On kuitenkin tosiasia, että työn otsikko on laaja. Yhden case-projektin tarkastelu ei kykene sitä avaamaan. Otan siis työn substanssiksi myös muiden projektien parissa kerääntyneen kokemuksen sekä alan harrastajan yleistiedon avoimien ohjelmistoprojektien toiminnasta.

2 OPINNÄYTETYÖN KONTEKSTI

2.1 Open source, avoimen lähdekoodin ohjelmistokehitys

Opinnäytetyön lukijalle on oleellista ymmärtää avoimen lähdekoodin, ns. open sourcen konsepti. Kyseessä on tietokoneiden ja muiden elektronisten laitteiden ohjelmistojen kehitykseen liittyvä toimintamalli, joka ideologiansa puolesta suorastaan määrittää informaatioajan ja Y-sukupolven yleiset ajatusmallit. Kyse on perustavanlaatuisesti tiedon, teknologian ja osaamisen vapaasta liikehdinnästä.

Avoimen lähdekoodin ohjelmistokehityksessä lähdekoodi on vapaasti kaikkien saatavilla. Tämä avaa useita ovia:

- Kuka tahansa voi tehdä kokonaiseen, toimivaan ohjelmistoon mitä tahansa haluamiaan muutoksia.
- Kuka tahansa voi todentaa ohjelmiston teknisen taustan: ohjelmiston ulkoisen kuoren alle ei ole mahdollista piilottaa haitallisia komponentteja ilman riskiä siitä, että kuka tahansa voi ne lukemalla löytää.
- Kuka tahansa voi ladata itselleen tai jakaa ohjelmiston lähdekoodin, ja sen avulla kääntää itselleen kopion itse ohjelmistosta.

On myös tyypillistä, että avoimen lähdekoodin ohjelmistojen rakennus ja ylläpito tapahtuu harrastuksena. Lukemattomat laajalle levinneet ja rahallisestikin mittavat projektit ovat saaneet alkunsa yhden ihmisen tai pienen ihmisjoukon yhteisestä harrasteesta, johon ei ollut sitoutunut lainkaan rahapääomaa: Linux-käyttöjärjestelmäydin, phpBB-keskustelufoorumisovellus, GIMP-kuvankäsittely-ohjelma ja niin edelleen.

Monelle tietokoneen käyttäjälle ovat tuttuja käsitteitä niin kutsutut freeware- tai shareware-ohjelmistot. Samalla tavalla kuin suuri osa avoimen lähdekoodin ohjelmistoista nykyään, ne ovat ilmaisia hankkia. (Dictionary.com LLC 2012a, 2012b.)

Freewaren ja sharewaren kehityskonsepti ei kuitenkaan välttämättä ole lähelläkään tämän opinnäytetyön kontekstia, avointa ohjelmistonkehitystä; pikemminkin kyseiset ohjelmistot voivat toteutukseltaan olla lähempänä kaupallisia, perinteisen käyttölisenssin alaisia ohjelmistoja.

Freeware ja shareware ovat 1980-luvulla yleistynyt ilmiö. Näille ohjelmistoille on tyypillistä, että niillä houkutellaan käyttäjä ostamaan saman ohjelmiston maksullinen versio, jossa on enemmän ominaisuuksia tai olemassa oleviin ominaisuuksiin tai koko ohjelmiston käyttöaikaan kytketyt rajoitteet on poistettu. Tässä ovat kyseessä englannin kielen free-sanan erilaiset sävyt. Englantia puhuvassa maailmassa voidaan samalla free software -käsitteellä puhua sekä ilmaisesta, että vapaasta tai avoimesta ohjelmistosta. Meillä Suomessa kuvio on vähän helpompi hahmottaa – on voitu puhua vain avoimesta ohjelmistonkehityksestä, ja toisaalta ilmaisohjelmistoilla on aina ollut oma kaupallisuuteen viittaava sävynsä.

Tietokoneohjelmistojen lähdekoodin kulku oli 1950- ja 1960-luvulla vapaata yliopistojen laboratorioissa, samalla tavalla kuin tekstien ja tiedon kulku on lähes aina ollut vapaata akateemisissa ympäristöissä. Ohjelmistokehitys otti ensimmäisiä askeliaan. 1960-luvun loppupuolella jopa nykyään hyvin suljetusti toimivien suurien IT-yritysten ohjelmistot olivat vapaasti muokattavissa.

1960-luvun lopussa ja 1970-luvun alussa alkoi tapahtua muutosta, joka on muovannut ohjelmistonkehityksen maailman sellaiseksi, kuin se on nykyään. Ohjelmistoista tuli suurempia ja kalliimpia projekteja, ja siten myös niiden teknistä toteutusta ja räätälöintiä alettiin säännöstellä. Tietokoneohjelmistobisneksessä siirryttiin patenttien ja lisenssien aikakaudelle.

1980-luvulta lähtien siirryttiin aikaan, jota voidaan kutsua avoimen ohjelmistonkehityksen varsinaiseksi renessanssiksi. Internet löi itseään läpi, ja avoimen ohjelmistokehityksen guru Richard Stallman (Massachusetts Institute of Technology) starttasi GNU-projektiaan. GNU:sta, joka alkoi käyttöjärjestelmäprojektina, tuli avoimen ohjelmistokehityksen ekosysteemi, jonka lisensseihin tulisivat nojaamaan tulevat avointen ohjelmistojen massat. 1991 Linus Torvalds (Helsingin yliopisto) julkaisi Linux-ytimen, johon pohjautuvat kaikki nykyiset Linux-jakelupaketit.

Nykyään avoimen lähdekoodin ohjelmistot ja kehitysprosessit ovat osa tietotekniikka-alan ammattilaistenkin arkea. Vuonna 2012 noin seitsemänsadan IT-alan ammattilaisen joukossa tehdyssä avoimen lähdekoodin ohjelmistojen käyttöä koskevassa tutkimuksessa selvisi, että yli puolet IT-alalla tapahtuvista ohjelmistohankinnoista tulee kohdistumaan avoimen lähdekoodin ohjelmistoihin (North Bridge Venture Partners 2012). Tutkimusorganisaatio Gartner puolestaan ennusti 2008, että vuonna 2012 kaupallisista ohjelmistoista 80 % tulisi pohjautumaan ainakin osaksi avoimen lähdekoodin komponentteihin (Ars Technica 2008).

Avoimen lähdekoodin lisenssejä käytetään lukemattomissa kaupallisissa ohjelmistoissa, esimerkiksi MySQL-tietokantaohjelmistossa, StarOffice-toimisto-ohjelmistopakettissa, sekä Red Hat Linux -käyttöjärjestelmässä.

Suomessa avoimen lähdekoodin ohjelmistot ovat laajalti käytössä. Esimerkiksi Saimaan ammattikorkeakoulussa opiskelijat pääsevät halutessaan käyttämään GIMP-kuvankäsittelyohjelmistoa, joka kuuluu myös opintosuunnitelmaan, LibreOffice-toimisto-ohjelmia ja Firefox-internet-selainta. Myös opettajien käyttämä Moodle-kurssialusta on avoimen ohjelmistoprojektin tuote.

Vaikka useista kaupallisista projekteista onkin tehty avointa lähdekoodia, tulee vapaista ohjelmistoista harvoin onnistuneita kaupallisia projekteja. Otaksun, että tähän on syynä perustavanlaatuisen ero kaupallisen ja vapaan projektin filosofioissa. Vapaiden ohjelmistojen kirjoittajien motivaattorina on harvoin raha, kun taas liiketoiminnan ainoa perustavanlaatuinen motivaattori on raha. Tämä ristiriita aiheuttaa kipinöintiä, jos avoimen lähdekoodin ohjelmiston tekijöiden harrastuksesta leivotaan kaupallinen tuote. Tämä on erityisesti ongelma, jos myytävästä tuotteesta rahat menevät tuntemattomille toimijoille.

Red Hat Linux ja MySQL Enterprise ovat silti menestyviä kaupallisia tuotteita. Miksi se ei aiheuta närää avoimen lähdekoodin yhteisössä? Syy löytyy liikeideasta: tuotteella ei rahasteta samalla tavalla kuin Microsoft Windows -käyttöjärjestelmällään tai omalla SQL-ohjelmistollaan. Yritykset tekevät tuottonsaa oheispalveluilla, kuten asennus-, ylläpito-, koulutus- ja konsultointipalveluilla, ja itse ohjelmisto pysyy avoimena ja ilmaisena tuotteena.

Valtaosa avoimen lähdekoodin ohjelmistoista kuitenkin on vieläkin vapaaehtoisvoimin ja joko pienellä tai suurella ihmisjoukolla rakennettavia harrasteprojekteja. Yleensä projektit lähtevät jostain tarpeesta, joka ohjelmistolla täytyy täyttää. Ohjelmistot voivat olla esimerkiksi työkaluja, leluja, pelejä, tieteellisiä kokeita tai prototyyppisiä.

2.2 OpenTTD-peli ja case: pelin grafiikkakehitysprojekti

OpenTTD-yhteisö valmistaa samannimistä tietokonepeliä, jossa pelaaja hallinnoi kuljetusyhtiötä. Pelissä on tavoitteena rakentaa ennen pelin alkamista sattumanvaraisesti generoituun pelimaailmaan tie-, rautatie-, meri- ja ilmakalustoon perustuva kuljetusverkosto. Kuljetuspalveluilla tehdään yritykselle tuloja, joilla laajennetaan toimintaa, millä tehdään lisää tuloja ja niin edelleen.

Yhteisö toimii vapaaehtoisvoimin rakentaen ilmaiseksi jaettavaa, avoimen lähdekoodin lisenssin alaista tuotetta. Peliprojekti ja sen alaprojektit ovat toisistaan erillään tarkasteltavia itsenäisiä prosesseja, joiden komponentteja erittelen myöhemmin opinnäytetyössä.

OpenTTD-peli on hyvin tyypillinen harrasteena kehitelty avoimen lähdekoodin ohjelmisto. Projektin alku oli yhden henkilön kehittämä prototyyppi, joka saavutti suuren suosion ja joka kehittyi nykyään isomman kehitystiimin voimin. Koko historiansa ajan varsinaista tuotetta on saanut netistä sekä lähdekoodina että valmiina ohjelmistona ilmaiseksi. Sen ympärille on kehittynyt itseään ruokkiva kehittäjien, pelaajien, komponenttien ja kustomoinnin ekosysteemi.

Pelin tekniikka mahdollistaa erilaisten visuaalisten ja pelitekniisten lisäosien asentamisen. Tällä hetkellä onkin käynnissä alaprojekti, jossa pelin jo 15 vuotta ikääntynyttä ulkonäköä päivitetään 2000-luvulle. Opinnäytetyöni käsittelee tämän alaprojektin parissa keräämääni tietoa ja kokemusta.

2.3 OpenTTD-pelin lyhyt historia

Vaikka OpenTTD-projektin historia alkaa vuodesta 2003, jolloin ruotsalainen Ludvig "ludde" Strigeus julkaisi pelistä ensimmäisen prototyyppinsä, täytyy itse asiassa palata ensin vuoteen 1994, jolloin Microprose toi markkinoille Transport Tycoonin, britannialaisen pelinkehittäjän uuden pelin. Pelin idea oli täsmälleen sama kuin OpenTTD:n vielä nykyäänkin. Transport Tycoon keräsi suuren yleisön ja erinomaiset arvostelut medialta.

Vaikka Chris Sawyer jatkoi varsin menestyksekkäälle pelinkehittäjäuralle, ja Transport Tycoonilla oli vankka kannattajakunta, peli ei saanut kuin yhden virallisen jatko-osan. Transport Tycoon Deluxe oli vain hieman hiottu ja laajennettu versio alkuperäisestä pelistä.

Muut pelialan yritykset ovatkin huomanneet markkinoilla olevan kysyntää juna- ja kuljetusyhtiösimulaattoreille. Markkinoille on tuotu muiden muassa suosiotakin paljon saanut Railroad Tycoon -sarja, joka ei liity Transport Tycooniin muuten kuin teemaltaan.

Vuonna 2004 tapahtui kehitystä, joka määritteli Transport Tycoonin jatkon 2000-luvun loppupuolelle ja siitä eteenpäin. Maaliskuussa Ludvig Strigeus julkisti jo edellisenä vuonna aloittamansa projektin. Hän päätti kopioida pelin alkuperäisen mekaniikan ja rakentaa uuden toteutuksen pelistä niin, että siinä saattoi käyttää alkuperäisen pelin grafiikoita. Pelin tekninen pohja päivitettiin nykyaikaan, lisäksi puutteita ja virheitä korjailtiin. Syyskuussa taas itse Chris Sawyer julkaisi uuden pelin, Locomotionin, jonka hän väitti olevan ”Transport Tycoonin henkinen seuraaja”.

Fanikunta kuitenkin äänesti lompakoillaan. Locomotion sai huonot arvostelut, pelin myynti jäi heikoksi eikä sen maine ole mairitteleva. OpenTTD sen sijaan keräsi mittavan yleisön ja hyvät arvostelut.

OpenTTD-projektin historia on moniin muihin avoimen lähdekoodin projekteihin verrattuna poikkeuksellisen rauhallinen. On tyypillistä, että projekteja ostetaan, pilkotaan, versioidaan (”forkkaus”) tai niiden kehittäjä jättää projektin heitteille ilman kehittäjää. Näistä mitään ei ole kuitenkaan tapahtunut OpenTTD:ssä. Projektilla on ollut tauotta selkeä henkilöstö ja infrastruktuuri. Pelaaja- ja faniyhteisö on lisäksi hyvin positiivinen pelaamista ja kehitystyötä kohtaan.

Strigeus jäi pois kehittäjäkunnasta vain muutamia kuukausia prototyypin julkaisun jälkeen. Tällä hetkellä merkittävä osa pelin muutoksista on tullut kahdeltatoista tunnetulta kehittäjältä. Teknisiä ratkaisuja, parannuksia, laajennuksia jne. on tullut lukemattomilta osallistujilta.

2.4 Sivullisesta OpenTTD-yhteisön jäseneksi

Olen ollut kiinnostunut Transport Tycoon -pelistä jo 90-luvulta saakka. Alussa kyse oli perinteisestä pelifaniudesta: aihealue oli kiinnostava, grafiikat veivät mennessään, ja mekaniikka toi haastetta arjen monotonisuuteen. Peli on kuljetusyhtiösimulaattori, ja olen ollut pienestä pitäen hurahtanut vanhoihin juniin ja laivoihin. Nuorena aikuisena löysin OpenTTD:n, ja sen vetonaulana oli ilmaisuus ja nostalgia. Etenin nopeasti passiivisesta pelaajasta osaksi yhteisöä.

Aluksi, vuosina 2005 - 2007, roolini OpenTTD-projektissa oli varsin värikäs. Se kuvasti osaamistani ja mielenkiinnon kohteitani kuluneen vuosikymmenen ajalta: kirjoittelin wikiä, pidin yllä listaa "peliskenaarioista", joita pelaajat pystyivät ratkomaan, ja rakensin myös joitakin omia pelikenttiä muiden pelailtavaksi.

Jo vuonna 2004 alkoi aliprojekti, jossa alkuperäisiä 1990-luvun grafiikoita korvattiin modernein tekniikoin kehitetyillä vastineilla. Nämä uudet 32-bittisen väripaletin grafiikat olivat tarkempia ja värikkäämpiä kuin Tycoonissa käytetyt 8-bittiset grafiikat. Ajankäyttöni siirtyi vuoden 2007 tienoilla täysin tämän grafiikkaprojektin saralle.

Hallinnollisia tehtäviä grafiikkaprojektissa

Alussa harrastelu painottui samantyyliisiin tehtäviin, kuin ne olivat muualla isommassa projektikuviossa olleet: olin näkyvä hahmo projektikeskusteluissa ja päivitin ja kehitin projektidokumentaatiota wikissä. Vuonna 2007 eräästä projektikehityskeskustelusta lähti idea uuteen aliprojektin aliprojektiin: kaikki kehitteillä olevat grafiikat olisi hyvä saada keskitettyä yhteen hallintaohjelmistoon. Prototyyppejä ja kokeiluja oli useita, mutta lopullisen teknisen toteutuksen kirjoitin ja julkaisin keväällä 2009.

Näiden vuosien aikana minusta kasvoi ikään kuin vahingossa näkyvämpi ja kokeneempi hahmo projektin keskustelualueella, ja keväällä 2010 sain julkisen Manager-tittelin, johon kuuluu keskustelun organisointi- ja moderointioikeus. Roolini nyt, vuonna 2012, on kiteytettävissä siihen, että olen nähnyt paljon, kysynyt paljon, tehnyt paljon, ja mikä tärkeintä, ohjeistanut paljon niitä, jotka eivät ole nähneet, kysyneet tai tehneet.

Olen ollut siis noin kolme vuotta projektin de facto -koordinaattori. Olen myös projektin kolmesta kokeneimmasta edelleen aktiivisesta jäsenestä yksi: olen nähnyt projektin tähänastisesta elinkaaresta yli puolet omin silmin. Tämä antaa aivan erityisen perspektiivin, josta kirjoittaa tällaista projektia analysoiva opinnäytetyö.

Harrastelija ilmaisprojektin ”johtaja” – naurettavaa. Vai onko?

Kun puhutaan ilmaiseksi jaettavasta tuotteesta, ilmaisesta työvoimasta, ja henkilöstöstä, jolle projekti on harraste, onko reilua käyttää keskeisistä toimijoista termejä kuten manager – päällikkö, johtaja tai koordinaattori? Perinteisestihän tuota termiä käytetään yritysmaailmassa niistä henkilöistä, joilla on ympärillä toimivien henkilöiden arvostus sekä suuri määrä tietoa organisaation päivittäisestä toiminnasta ja sen perusteista.

Robbins, Judge ja Campbell (2012, 4) linjaavat johtajan tehtäväkuvan: *he aikaansaavat asioita muiden ihmisten kautta, tekevät päätöksiä, kohdistavat resursseja ja ohjaavat muiden toimintaa asetettujen päämäärien saavuttamiseksi*. Kirjoittajat ottavat esille Fayolin viisi johtotehtävää, joiden he mainitsevat olevan tiivistettävissä neljään: **suunnittelu, organisointi, koordinaointi ja kontrolli**.

Suunnittelu on ollut kuluneiden vuosien aikana vahvasti läsnä toiminnassani grafiikkaprojektissa. Esimerkkejä tästä ovat grafiikkatiedostojen keskitetyn tallennusjärjestelmän, kehitystiimin ja lopputuotteen välisen suhteen suunnittelu,

uusien teknisten ratkaisujen migraatiosuunnittelu, ja tietysti itse projektin etenemisen suunnittelu ja suuntaa-antava aikataulutus.

Organisointi on oleellinen osa avoimen lähdekoodin projektin hallintoa. Vaihtuvuus on välillä hyvinkin suurta, joten on elintärkeää, että uudet harrastajat pääsevät mahdollisimman nopeasti oppimisvaiheesta tekemisvaiheeseen. Tässä auttavat havainnolliset ohjeet ja infrastruktuuri. On hallintoväen tehtävä järjestää asiat niin, että kaikenlaiset työohjeet, prosessit, säännöt, työkalut, materiaalit ja muut hivenaineet ovat helposti kaikkien saatavilla.

Koordinoinnilla tarkoitetaan henkilöstön ohjaamista ja motivointia työtehtäviin yhteisten tavoitteiden saavuttamiseksi. Nämä käsittävät myös kommunikoinnin, ongelmanratkaisun ja kitkan vähentämisen organisaatiossa. Niin hyvässä kuin pahassa, olen viime vuosina saanut useammin kuin kerran osallistua sanaharkkoihin ja väittelyihin erilaisista teknisistä ratkaisuista, yhteisön normeista, tulevaisuuden suunnitelmista, ja tietyissä tapauksissa jopa sosiaalisten ongelmien ilmentyessä.

Kontrolli on johtamisen osa-alue, joka on avoimen lähdekoodin projekteissa harvoin näkyvästi mukana. Organisaatiot eivät tee etenemisellä tuottoa, eivätkä hidastelulla tappiota, joten motivaatio käyttää aikaa ja nähdä vaivaa toimintatehon seurantaan on heikko. Siitä huolimatta kontrolli on prosessi, joka on mukana toiminnassani grafiikkaprosessissa. Projektin etenemisestä on graafinen kuvaaja, jota pidetään ajoittain silmällä. Lisäksi projektin koosta ja viimeisistä tapahtumista on olemassa kovaa dataa, jota seurataan.

Kaikki hallintotyön aspektit ovat läsnä. Minusta on siis reilua käyttää avoimen lähdekoodin projektikoordinaattoreista samoja nimikkeitä kuin liiketalouden johtajistakin. Jos projekti saa hallintoväen toimimaan tehtävissä ilmaiseksi, on se minusta projektille meriitti, ei häpeän aihe.

3 AVOIN OHJELMISTOPROJEKTI BUSINESS-NÄKÖKULMASTA

Business may be defined as human activity directed towards producing or acquiring wealth, through buying or selling goods.

- L. H. Haney

Liiketoiminta, *business*, on amerikkalaisen ekonomistin L. H. Haneyn määritelmän mukaan ihmisten toimintaa hyödykkeiden ostossa ja myynnissä oman varakkuuden kasvattamiseksi (Jain ym. 2009, 4). Avoin ohjelmistokehitys ei täytä tätä määritelmää. Eihän siinä myydä tuotetta. Vai myydäänkö?

Tietokoneohjelmistoa valmistavan organisaation tärkein resurssi, siis henkilöstö, on mielestäni ”palkattomassakin” yhteisössä ostettu. Palkka on jotain paljon rahaa rakkaampaa: hauskanpitoa, mielenkiintoa, oman itsensä kehittämistä, ja harrastajamieltä. Eikö tuotetta siis myydä yhteisön vaivannäköä vastaan? Ilman yhteisöähän projekti kuolisi. Kun siirrytään toimintaympäristöstä, jossa valuuttana toimii raha, ilmaiseksi toimivaan projektiin, muuttuvat ostamisen ja myymisen, maksamisen ja palkan käsitteet abstraktimmiksi käyttäytymistieteen ja organisaatiokäyttäytymisen aloja hipoviksi pulmiksi.

Termi *business* on englantia, ja on sanaa lähemmin tarkasteltaessa adjektiivista *busy*, *kiireinen*, johdettu sana. Liiketoiminta on siis sananmukaisesti *kiireellisyttä*. Liiketoiminnasta käytetyksi sanaksi ei koskaan vakiintunut *earning*, *ansaitseminen*, tai *profiting*, *tuottaminen*. Itse asiassa jopa suomenkielinen käsite *liiketoiminta* käsittää vain *liikkumista*. Sanaa analysoimalla toimintaan ei liity raha. Voisi siis hyvin perustein asettaa kyseenalaiseksi, että liiketoiminnan tyyppisessä toiminnassa on pakko tehdä omistajalle rahaa. Asia ansaitsee mielestäni lähempää tarkastelua.

3.1 Liiketoiminnan piirteitä avoimessa ohjelmistonkehityksessä

Liiketoiminnalle tyypilliset piirteet vaihtelevat lähteestä toiseen. Esimerkiksi *Business Environment* -teoksen mukaan keskeiset piirteet ovat:

- tuotteiden ja palvelujen vaihdanta
- tavoitteena tuoton tekeminen
- jatkuvaa ja vakaata toimintaa
- riski-elementti
- arvon lisääminen
- asiakastyytyväisyyden tavoittelu
- innovointi ja tuotekehitys
- henkilöresurssien hyväksikäyttö
- vuorovaikutus ympäröivän yhteisön kanssa
- liiketoimintaa voidaan käsitellä sekä tieteenä, että taiteena
- toiminnan absoluuttinen monimuotoisuus

(Jain ym. 2009, 4.)

Muita ominaisuuksia tunnistaa R. K. Singla (2009) teoksessaan *Business Studies*:

- toiminta on taloudellista, siihen liittyy raha
- taustalla on yksi yrittäjä
- toiminnan muodostaa organisaatio
- toiminnan mahdollistaa rahoitus

On mielenkiintoista, kuinka monta liiketoiminnan laajan määritelmän osaa avoin ja ilmainen OpenTTD-projekti kattaa:

- OpenTTD on lopputuotteen jakelua pelaajakunnalle, siis asiakaskunnalle, ja ajoittain myös palvelujen ostamista.
- OpenTTD on jatkuvaa ja vakaata toimintaa. Se on ollut päivittäistä vuodesta 2003 lähtien.

- OpenTTD:hen aikaa ja vaivaa käytettäessä aivan selvästi toimitaan riskillä, sillä jos projekti kuolee ja lopputuotteen jakelu lopetetaan, ei käytetty aika ja vaiva johda enää uusiin versioihin.
- OpenTTD:n toiminnan perusajatuksena on asiakkaiden tyytyväisyyden tavoittelu. Siihen pyritään käyttäen hyväksi innovointia ja henkilöresurssien tuotekehitykseen käyttämää työpanosta.
- OpenTTD toimii selvästi vuorovaikutuksessa sitä ympäröivän yhteisön kanssa – yhteisön, joka muodostaa sekä sen pelaaja- että kehittäjäkunnan.
- OpenTTD:n toiminta on perustavanlaatuisesti monimuotoista. Itse asiassa projektin toiminta käsittää aivan yllättäviä piirteitä, joita olettaisi olevan vain pelialan yritystoiminnassa.
- OpenTTD:n taustalla on yksi henkilö, Ludvig Strigeus, joka julkaisi ensimmäisen prototyypin OpenTTD:sta.
- OpenTTD:n yhteisö taatusti muodostaa selvän organisaation, jolla on määriteltävissä oleva rakenne, ja tiettyjä reittejä pitkin kulkeva kommunikointi.

Minun nähdäkseni ainoa oleellinen liiketoiminnan osa-alue, joka ei suoraan päde OpenTTD-projektiin, on taloudellinen perspektiivi. Siitäkin itse asiassa puolet pätee, sillä vaikka tuotetta hinnoiteltaessa on päädytty noltaan ja palkkakustannukset ovat nolla rahayksikköä, on käytetyn infrastruktuurin välttämättömyys projektille todellinen rahallinen menoerä.

Sekä OpenTTD että sen alaisuudessa toimiva grafiikkaprojekti (case-projekti) ovat varsin tyypillisiä avoimen ohjelmistonkehityksen organisaatioita. Voisi sanoa, että ne ovat tyypillisiä organisaatioita. Niitä voisi verrata voittoa tavoittelemattomiin eli ns. ”non-profit”-organisaatioihin.

OpenTTD-projektissa, ja itseasiassa myös sen aliprojekteissa, on tunnistettavissa samat elementit kuin missä tahansa liiketoiminta- tai muussa organisaatiossa:

- Niillä on asiakaskunta.
- Niillä on henkilöstö, joka noudattaa sovittuja toimintamalleja, jotka on laatinut projektin johto.
- Niillä on kiinteätä omaisuutta, jonka ylläpito on vuosittainen menoerä.
- Tasaisin väliajoin ydinosoaminen johtaa uuteen versioon tuotteesta.
- Tuotetta markkinoidaan osin jopa varsin tarkasti määriteltävissä olevalle asiakaskunnalle.
- Asiakaskunta käyttää tuotetta, mikä generoi yleistä kiinnostusta, mikä osaksi toimii henkilökunnan palkkana, mikä varmistaa ydinosoamisen säilyvyyden ja projektin jatkuvuuden.

Yhteisön toiminta projekteissa on valtaosin virtuaalista. Silti niillä on reaali-omaisuutta, joka ei rajoitu pelkästään alati kehittyvään tuotteeseen:

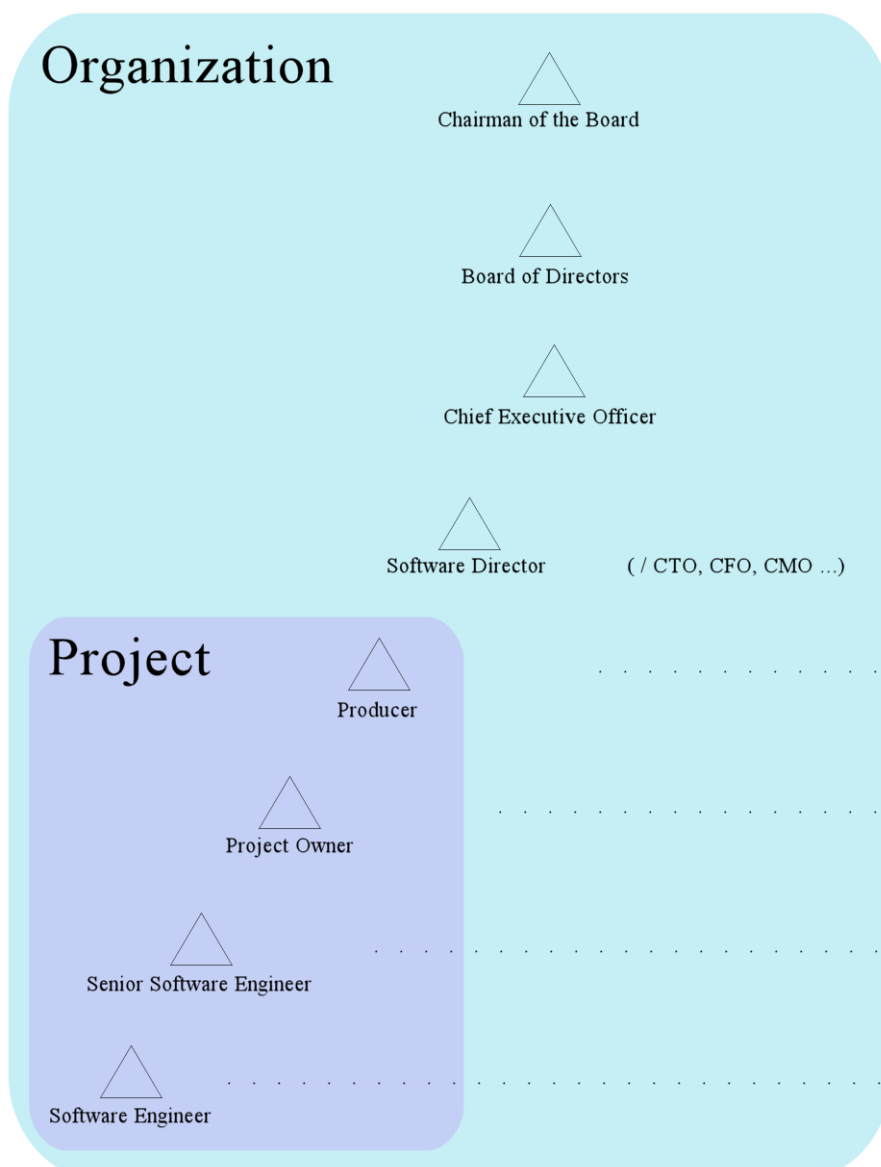
- kirjoitettu koodi
- erilaiset tekstimuotoiset tekniset komponentit ja ohjeistus
- binäärimuotoinen sisältö kuten grafiikka ja äänet
- nettisivujen, keskustelufoorumien ja muiden medioiden sisältö
- alihankintana käyttöön otettu palvelintila
- erilaiset kehitystyössä käytetyt laitteet
- muu merkittävydessään pienempi hajaomaisuus.

Tuotanto tapahtuu käyttäen työkaluja, joita ovat mm. koodinkirjoittajien käyttämät kehitysympäristöt tai ASCII-kirjoitusohjelmat, grafiikankehittäjien käyttämät kuvankäsittelyohjelmat ja metadatakirjoittimet, sekä tiketti- ja versionhallintaohjelmat internet-palvelimella.

3.2 Organisaatio

Liiketoimintaorganisaation, esimerkiksi pelialan yrityksen organisaatiokartta on monelle tutun näköinen eritoten ylä- ja alalaidoistaan. Tyypillisessä organisaatiossa pyramidin kärjessä on toimitusjohtaja. Hänen rinnallaan, tai teknisesti ottaen yläpuolella, toimii hallitus, jolle toimitusjohtaja on vastuussa. Toimitusjohtajan alla ovat C-tason johtajat, jotka tyypillisesti on jaoteltu toimintalan mukaan, esimerkiksi talousjohtaja, markkinointijohtaja, operatiivinen johtaja, ja niin edelleen.

Pelialan yrityksissä *Software Director* on tyypillinen tuotteen tai tuotteiden hyvinvoinnista ja kehityksestä vastaava nimike. Hänen alapuolellaan toimivat itse tuotteista vastuussa olevat johtajat. Jokaiselle näistä on vastuutettu yksi tuote. Niinpä osasto- tai organisaatioyksikkökohtaisten johtajien määrä



Kuva 3.1 Tyypillisen ohjelmistoalan yrityksen rakenne, löyhästi mukaillen CCP hf:n organisaatorakennetta. Tulkinta tekijän.

vaihtelee ja joskus niitä on useammassa tasossa yrityksen koon ja erityisesti maantieteellisen laajuuden mukaan. Hierarkiassa alhaalla ovat ns. linjatyöntekijät ja heidän välittömät esimiehensä.

Kuinka sitten avoimen ohjelmistonkehityksen projektit organisoituvat? Taustalla on koordinoinnin ja tehtävänjaon selvä poissaolo. Yhteisössä, jossa siteet organisaatioon ja sitä johtaviin henkilöihin ovat löyhät, rooleja ei täytäkään tarkasti mikään johtoryhmä. Pikemminkin kyse on henkilöstölähtöisestä vastuun ottamisesta, *kontribuutioiden* tekemisestä ja osallistumisesta ilman siteitä.

Vapaan organisoitumisen ilmiötä kuvailee Jean Laven ja Etienne Wengerin *käytäntöyhteisön* teoria. Sen mukaan ryhmä muodostuu sen jäsenien yhteisen mielenkiinnon seurauksena. Käytäntöyhteisön muodostuminen koskee teorian mukaan nimenomaan tiedon välitystä, joka tehostuu huomattavasti. Ryhmän jäsenet jakavat tietojaan ja kehittyvät yksilöinä. Tietoa voi siirtyä esimerkiksi ongelmatilanteessa. Sellaisia ovat henkilökohtaiset ongelmatilanteet, tuotteessa olevat ongelmat tai kehitystyön ongelmatilanteet. Lisäksi käytäntöyhteisö voi toimia hyväksi havaittujen toimintatapojen jakamisessa ja uuden kehitystyön suunnittelussa. (Wenger 2006.)

Käytäntöyhteisöt ovat luonteeltaan erilaisia organisaatioita kuin liike-elämän projektiryhmät. Käytäntöyhteisöä ei mikään aja hajoamaan, kun ongelma on ratkaistu tai jokin tavoite saavutettu. Lisäksi käytäntöyhteisöissä jäsenten roolit voivat olla hyvin häilyviä, kun taas projektiryhmissä jäsenten roolit on voitu määrittellä hyvinkin tarkasti, ja jäsen on voitu hankkia suorittamaan nimenomaan tiettyä tehtävää.

Crowston ja Howison (2005) totesivat 140:een avoimen lähdekoodin projektiin kohdistuneen tilastollisen analyysin perusteella, että projektien organisoitumisen välillä oli suuria eroja ja projekteissa toimijoiden välillä tapahtuvan kommunikoinnin muoto ei vastaa yleisiä ennakkoluuloja.

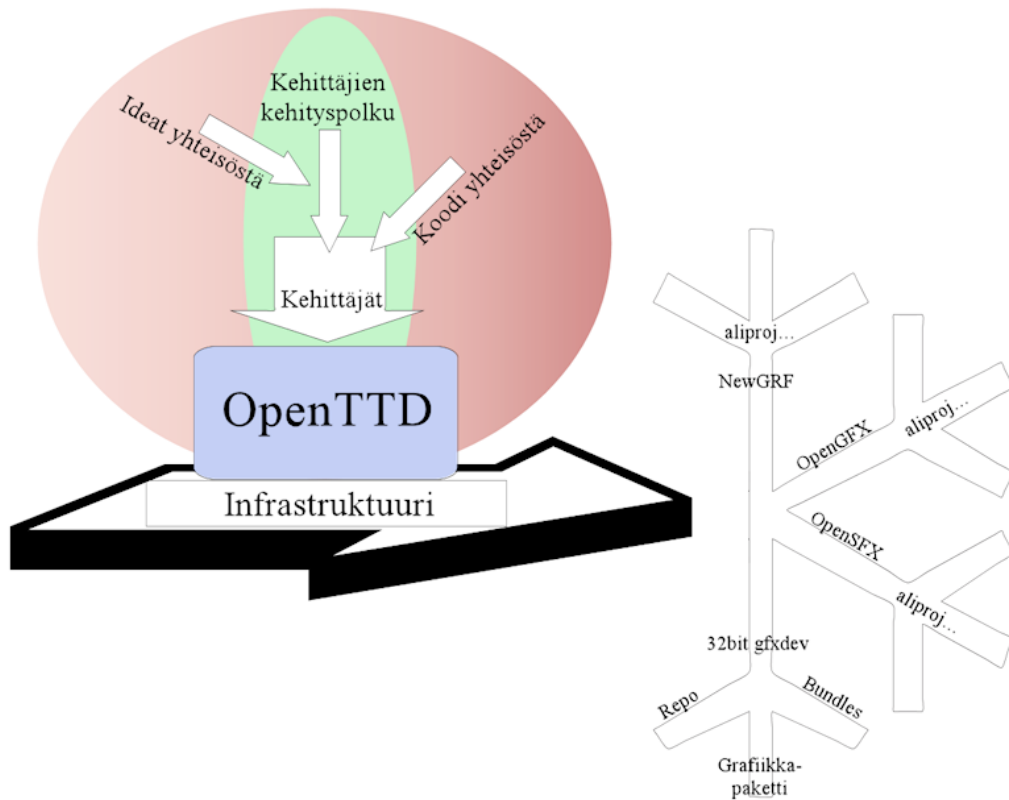
Crowston ja Howison (2005) löysivät esimerkkejä sekä hyvin keskitetyksi organisoiduista että hajautetuista projekteista. Esimerkiksi curl-komentotulkki valmistava yhteisö oli analyysin perusteella erittäin keskitetysti hallinnoitu organisaatio. Toisaalta squirrelmail-sähköpostinlukija oli selvästi toisesta ääripäästä: kommunikoinnin eteneminen hierarkiassa on hyvin epäsäännöllistä ja välitasoja on useita. Tekijät huomauttavatkin tutkimuksen päätelmissä, ettei yleisesti voi tehdä olettamuksia avoimen lähdekoodin ohjelmistoprojektien organisoinnista.

Koska tutkimus on tätä opinnäytetyötä kirjoitettaessa kahdeksan vuotta vanha, on nyt mahdollista tutkia, kuinka tutkimuksessa käsiteltyjen projektien menestys korreloi niiden organisointityylin kanssa. Niistä seitsemästä nimetystä projektista, joilla *keskitettyysaste* (suomennettu termistä *degree of centrality*) oli noin 0,4 tai vähemmän, kolme on edelleen aktiivisia ja neljä on kuollut. Neljästä nimetystä projektista, joilla aste oli noin 0,6–1,0, kaksi on kuollut ja kaksi aktiivista. Kuinka keskitetysti projekti on organisoitu, ei siis näytä ehdottomasti vaikuttavan projektin elinedellytyksiin. On huomion arvoista, että mainitut aktiiviset projektit ovat vertailujoukkonsa suurimpia sekä keskitetysti että hajautetusti toimivien joukossa. Näyttäisi siis siltä, että projektin menestymisen kannalta on oleellista, että projektilla on taustallaan kookas väkijoukko.

Eric S. Raymond (1997) toi julkisuuteen basaari- ja katedraalimallin projektiorganisaation eroteltavissa olevista käyttäytymismalleista. Käyttäen 1990-luvun Linux-kehitystä esimerkkinä basaarimallista hän kuvailee sitä sanoin *julkaise aikaisin ja usein, delegoi kaikki mahdollinen, ja vaihtelevien agendojen ja lähestymismallien mahtava kohiseva basaari*.

Avoimen lähdekoodin ohjelmistoprojektien rakenne voi siis olla lähes mitä vain anarkian ja tarkan hallinnoinnin väliltä. Sen sisäinen kommunikointi voi käyttää mitä muotoa hyvänsä, ja sen jäsenet voivat suorittaa vapaasti valitsemiaan tehtäviä. Moni projekti voi ulkopuolisen silmin olla Raymondin kuvailema *basaari*. Silti projekti toimii ja sen tuotteena syntyy toimivia ohjelmistoja.

3.2.1 OpenTTD-organisaatio



Kuva 3.2 OpenTTD-peliä kehittävän organisaation rakenne

Grafiikkaprojektin organisaation tarkastelussa on lähdeittävä sitä ympäröivän OpenTTD-peliprojektin organisaatiosta. Peliprojekti noudattaa hyvin lineaarista kaavaa. Kuvaajan taustalla punainen muoto kuvaa OpenTTD-yhteisöä. Sen vaaleampi alue, vasemmalle painottunut puoli, kuvaa passiivista pelaajakuntaa. Tummempi alue, painottunut oikealle, kuvaa pelaajakunnan osaa, jonka voi laskea aktiiviseksi ja toimintakykyiseksi osaksi kehittäjäkuntaa. Aktiivisilla ja toimintakykyisillä pelaajilla on taito ja halu toteuttaa peliin muutoksia ja laajennuksia. He muodostavat epävirallisten kehittäjien ryhmän.

Kaiken ytimessä ovat aktiiviset ja viralliseksi tunnustetut pelinkehittäjät (kuvaajassa "Kehittäjät", vihreä muoto). Valtaosa peliin tehtävistä muutoksista on heidän tekemäänsä työtä. He kirjoittavat siis koodia ja ajavat sitä sisään tuotteeseen. He tekevät lisäksi suurimman osan projektin toistuvista ylläpito-

töistä. He lukevat ja hyväksyvät epävirallisten kehittäjien ja käyttäjien lähettämät muutokset, asettavat tavoitteet tuleville versioille, asettavat standardit kehitystyölle ja ylläpitävät infrastruktuuria.

Pelinkehittäjät ovat projektissa pisimpään mukana olleita, kokeneimpia jäseniä ja heidän vaikutusvaltaansa päätöksentekoon ja kiistatilanteisiin liittyvissä asioissa ei aseteta kyseenalaiseksi. Tarkasteltaessa huhtikuun 2010 ja huhtikuun 2012 välillä valmistuneita n. 1200 muutostikettiä OpenTTD-projektissa, kuusi kehittäjää vastasi yli 90 prosentista suljetuista tiketeistä. Merkittäviä aktiivisia virallisia kehittäjiä on siis kuusi - joukko on hyvin pieni. Projektin n. 10-vuotisen historian aikana peliin on ajanut muutoksia sisään 31 kehittäjää.

Epävirallisilta kehittäjiltä (kuvaajassa ”Koodi yhteisöstä”) tulee peliin enemmän tai vähemmän jatkuvana virtana muutos- tai laajennuspalikoita. Ne voivat olla koodia, grafiikoita tai äänitiedostoja. Epävirallinen kehittäjä ei ole virallinen termi. Kyseessä on pelin kehitykseen osallistuva yksilö, jota ei lueta mukaan varsinaiseen kehitystiimiin. Ero on abstrakti, ja nimitykset eivät ole missään nimessä seremoniallisia tai minkään projektinjohdollisen toimijan hyväksymiä. Usein viralliset ja epäviralliset kehittäjät kuitenkin erottaa toisistaan keskusteluissa näkyvillä titteleillä. Lisäksi virallisilla kehittäjillä on oikeuksia ja vastuuta, joita epävirallisilla kehittäjillä ei ole. Esimerkiksi muutoksien ja laajennusten ajaminen viralliseen pelin jakelupakettiin ei ole mahdollista muiden kuin virallisen kehitystiimin oikeuksin.

Pelaaja- ja kehittäjäyhteisössä toimii jatkuva yritysmaailman aloitelaatikko-ajatteluun rinnastettava aivomyrskyilyprosessi. Keskustelupalstalle on perustettu idea-alue, johon kuka tahansa voi kirjoittaa ominaisuuskonsepteja ja toivomuksia. Tämä vastaa osittain pelialan yritysten *Game Design* (suunnittelijat) -ryhmää. Aloitelaatikko-konsepti toimii tietysti osittain myös yritysmaailman puolella, ja avoimen ohjelmistokehityksen puolella taas on usein suunnittelutehtävä osittain vastuuntunut kehitystiimille.

Yhteisön ideoita toteuttavat sekä virallinen että epävirallinen kehitystiimi. Virallinen kehitystiimi toteuttaa ideoita huomattavasti harvemmin. Syyt tähän ovat pääasiassa seuraavat:

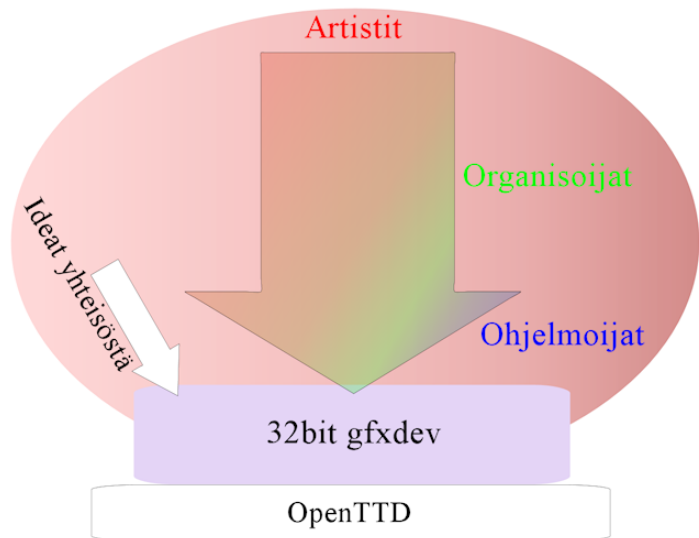
- Kehitystiimi on kiireinen ja priorisoi korjauksia olemassaoleviin ominaisuuksiin.
- Abstraktien konseptien muuttaminen käytännön peliominaisuuksiksi vaatii pitkää suunnittelua ja harkintaa. Suunnittelu puolestaan vaatii omistautumista asialle, joka ei ole käytännöllistä, kun puhutaan työntekijästä, jolle on vastuutettu paljon muita tehtäviä.
- Epävirallisten kehittäjien ryhmässä on enimmäkseen ohjelmoijia, jotka ovat täysin omistautuneet uusien ominaisuuksien toteuttamiselle, ja heidän projekteistaan valtaosa on valmiita tai peruttu. Silloin hyvälle idealle löytyy helposti toteuttaja.
- Kaikkien kehittäjien joukossa esiintyy tietty määrä ennakko-asenteita ”ehdottajia” kohtaan; heidät leimataan laiskoiksi, kun ehdottaja ei jaksa tai osaa, tai jaksa opetella toteuttamaan haluamaansa ominaisuutta, vaan kalastelee jotakuta tekemään sen hänen puolestaan.

Peliprojektin alla peruskivenä on infrastruktuuri, johon kuuluvat mm. ohjelmistot, palvelimet ja kehitykseen käytetty laitteisto.

Peliprojektista haarautuvat sivulle kaikki sen aliprojektit. Näihin ovat luettavissa kaikki projektit, joilla laajennetaan, mutta ei varsinaisesti muuteta peliä. Esimerkkejä ovat jokainen NewGRF-laajennus (jotka käyttävät pelin NewGRF-laajennusmekaniikkaa), OpenGFX-grafiikkapaketti, OpenSFX-äänipaketti ja 32bit gfxdev -nimellä kuvaamani grafiikkapaketti, jossa itse olen mukana.

3.2.2 Grafiikkaprojektin organisaatio

Grafiikkaprojektin organisaatiokartta on hyvin saman näköinen kuin isäprojektinsa. Osaksi tämä johtuu siitä, että rakenne on hyväksi todettu, mutta osaksi siitä, että rakenne on luonnollinen muoto, jonka avoin ohjelmistoprojekti ottaa.



Kuva 3.3 Grafiikkaprojektin organisaatio

Taustalla on yhteisö, jonka jäsenten halu ja kyky kehittää projektin lopputuotetta teknisesti vaihtelee. Teknisesti kyvykkäät rooliutuvat johonkin tehtävään, ja loput julkaisevat konsepteja, prototyyppejä tai ideoitaan. Tämä yhteisö on osa isompaa OpenTTD-yhteisöä.

Peruskivenä on tietysti itse peliprojekti. Ilman sen tarjoamaa ekosysteemiä, johon henkilöstö ja infrastruktuuri kuuluvat, grafiikkaprojekti ja muut OpenTTD:n aliprojektit eivät voisi elää.

Erot painottuvat pääasiassa kehitysmalliin. Siinä, kuten peliprojektissa, toimii kolmijaottelu ohjelmoijat, organisoiijat ja artistit. Roolien painotus ja kriittisyys on kuitenkin erilainen. Grafiikkaprojektissa kriittisinä kehitystiimin jäseninä ovat artistit, kun taas peliprojektissa sitä ovat ohjelmoijat. Artistit, organisoiijat ja ohjelmoijat muodostavat yhden kehitystiimin.

Artistit vievät läpi kehitysprosessin jokaisen paketin komponentin haluamassaan järjestyksessä. Organisoijat astuvat kuvioon loppupuolella prosessia, kun sisältöä täytyy seurata, visuaalista ilmettä valvoa ja seuranta- sekä muuta dokumentaatiota pitää ajan tasalla ja järjestyksessä. Aivan lopussa ohjelmoijat ajavat artistien kehittämän grafiikkakomponentin sisään grafiikkapakettiin.

Kehitysprosessin sisällöstä ovat projektin elinkaaren ajan vastanneet projektikoordinaattorit. Selkeys ja käyttökelpoisuus on ollut yhteisesti koordinaattorien ja organisoijien vastuulla. Prosessi elää isäprojektin muuttuessa ja sen täytyy vastata ympäröivää ekosysteemiä ja erityisesti muuttuvaa teknologiaa.

Kuten peliprojektikin, grafiikkaprojekti kehittyy osittain yhteisön ideoiden perusteella. Ei ole määrätty mitään paikkaa, missä niitä otetaan vastaan. Erityisesti tässä projektissa olen huomannut, että kehittäjien asenne ehdotuksia kohtaan on huono. Oletus on pääasiassa se, että jos projektia halutaan kehittää, se tehdään itse.

Merkittävä ero sekä peliprojektin että grafiikkaprojektin organisaatiossa verrattuna liike-elämän organisaatioihin on henkilöstön uran pituus. OpenTTD-kehittäjän aktiivisuus vaihtelee alle vuodesta pariin kolmeen vuoteen. Grafiikkaprojektissa tämä ilmiö korostuu. On tyypillistä, että grafiikkayksiköt valmistuvat kehittäjältä, joka ei palaa projektiin työnsä jälkeen. Yhden tai kahden vuoden uraa voi pitää pitkänä. Tämä asettaa erityisiä vaatimuksia projektiorganisoinnille uuden työvoiman vastaanotossa.

On syytä huomata, että OpenTTD-projektissa, kuten yritysmaailman organisaatiossakin, toimintamallit periytyvät pääasiassa hierarkian ylälaidasta alalaitaan päin. Näitä ovat esimerkiksi koodinkirjoitusohjeistus (*Coding Standards*), binäärimuotoisten komponenttien formaattiin liittyvät asiat, käytettävät työkalut, käyttäytymisstandardit, ym.

3.2.3 Henkilökuntana yhteisö

Pelialan liiketoiminnassa on puhuttava yrityksen henkilöstöstä tai henkilökunnasta. Heihin kuuluvat yrityksen ylimmät toimihenkilöt, kuten myös kehittäjät (*Developers*), suunnittelijat (*Game Designers*), laadunvalvojat (*Quality Assurance*), organisoijat/koordinaattorit (*Senior / Lead / Director*) ja asiakaspalvelijat (*Customer Support*).

Case-projektissa edellä mainitut roolit hoitaa joku – onhan jonkun pakko koordinoida projektia, kehittää peliä, tukea pelaajia ja ratkaista ongelmia. Rooliutumuksesta voi lukea lisää kappaleessa 4.1. Tehtävien hoitajista ei kuitenkaan puhuta mainituilla nimikkeillä, eikä heitä voi kutsua henkilöstöksi.

Avoimen ohjelmistokehityksen perusajatus on yhteisöllisyys. Kun ihmiset työskentelevät ilmaiseksi jonkin hyvän asian puolesta, esimerkiksi ilmaisen tuotteen tekemiseksi, yhteishenki luo yhteisön. Henkilöstö-sanassa on hierarkiaan, käskytykseen ja vastuutukseen viittaavia miellelyhtymiä. Yhteisöllisyys taas viittaa samanarvoisuuteen, vapaaehtoisuuteen ja tekemisestä nauttimiseen.

Suuren avoimen ohjelmistoprojektin loppuvaihe on usein työläs. Nautinto on silloin kaukana. Tämän olen saanut kuulla monilta pitkäaikaisilta avoimen ohjelmistokehityksen aktiiveilta, ja olen myös lukenut sivustaseuraajana samanlaisia kommentteja sekä tässä että monessa muussa projektissa.

Ilman hierarkista organisaatiota toimiminen ja samanarvoisuus toteutuu itse asiassa aniharvoin – se ei toteudu OpenTTD-projektissa, ei grafiikkaprojektissa, eikä missään muussakaan suuressa avoimessa ohjelmistokehitysprojektissa joka tulee mieleen. Alan Cox kuvaili tilannetta, jossa on pari aikaansaajaa ja parisataa sekalaista samanarvoista mielipiteen tarjoajaa, ”täydelliseksi tavaksi möhliä avoin ohjelmistoprojekti” (1998).

Mainituista parista selvästi negatiivisesta ominaisuudesta huolimatta jokin houkuttelee harrastajia toimimaan tällaisissa projekteissa. Minusta on mielenkiintoista, että näin on, ja olisi mielenkiintoista selvittää perinpohjaisemmin, kenties empiirisesti, miksi.

3.2.4 Työnjako ja vastuutus

Avoimessa ohjelmistoprojektissa työnjako ja vastuutus ovat merkittävästi erilaisia prosesseja verrattuna työelämän vastineisiinsa.

Grafiikkaprojektin työnjako on, yritysmaailmasta poiketen, henkilökunnan näkökulmasta proaktiivinen, ei reaktiivinen prosessi. Työnjako tapahtuu kehittäjän ottaessa tehtäväkseen jonkin julkaistun työtehtävän. Työtehtävät julkaistaan pieninä komponentteina seurantaohjelmistossa. Ne voivat olla ongelmia, jotka vaativat uuden ratkaisun kehityksen tai pelkästään ns. yksinkertaista työtä.

Työn määrääminen, ns. *assignaus* on negatiivissävytteinen menettelytapa ja herättää vastustusta. On epätodennäköistä, että määrätty työ tulisi koskaan tehtyä, eikä tätä toimintatapaa ole kokeiltu. Työn valinta perustuu tekijän mielenkiintoihin ja motivaatioon, esimerkiksi grafiikka-artisti voi olla kiinnostunut luonnon (kasvien tai maanmuotojen) mallinnuksesta, joten hän keskittyy niiden grafiikkaobjektien valmistamiseen.

Vastuu on moniulotteinen käsite, joka täytyy jakaa vastuuseen tuotteesta ja vastuuseen komponentista. Vastuu tuotteesta on juridinen ongelma, jonka ratkomista ei juuri näy mediassa. Lisää pohdintaa projektin juridiikasta on kappaleessa 3.4.

Vastuu komponenteista muotoutuu luonnollisella tavalla niiden valmistajille ja periytyy osittain hierarkiassa ylöspäin samalla tavalla kuin työelämässä.

Komponenttien toiminnasta, tekijänoikeudellisista asioista ja laadusta on vastuussa komponentin kirjoittaja. Ongelmatilanteissa yleensä edellytetään nopeahkoa ratkaisua ongelmaan, tai erikoistapauksissa poistetaan komponentti. Riitatapauksia esiintyy harvoin. Lead Developerin vastuu on luonnollisella tavalla rajallinen, jos komponentteja voi ajaa tuotteeseen alemmilta tasoilta.

Koordinaatiotasolla on syytä huomata, että vastuun ottaminen tai vastuuttaminen on turhaa, jos sidosryhmien välisessä kommunikaatiossa on ongelmia. Jos projektin onnistuminen riippuu teknisestä toteutuksesta, jonka määrittävä osapuoli ei kommunikoi päätöksiään riittävästi tai ei toimi käytyjen keskustelujen mukaan, ei projektista kannata kantaa vastuuta. Vastuuta määriteltäessä on ensisijaisen tärkeää, että tavoitteet ja toimintatavat ovat organisaation eri tasoilla ja rinnakkaisilla toimijoilla yhtenevät.

3.3 Tuote

OpenTTD-projektin ydintuote on samanniminen peli. Organisaatio on lähtöisin kyseisestä tuotteesta, ja muut ekosysteemissä olevat tuotteet ovat pääasiassa itse peliin kytkeytyviä liitännäisiä. Pelin eri versioita on ladattu sadasta kolmeen sataan tuhanteen kertaan. Tämä on huomattavasti mitään projektin muita tuotteita suurempi luku.

Valtaosa resursseista käytetään itse pelin kehitykseen ja ylläpitoon. On syytä huomata, että vaikka peli on tuote, ei palvelu, kuuluu siihen ylläpitoa, koska tuote sisältää palvelumuotoisia komponentteja, kuten moninpelipalvelimien seurantajärjestelmän.

Ohjelmistokehitysprojektien lopputuotteiden arvon määrittämiseen on kehitetty algoritmeja, jotka käyttävät lähdetietonaan enimmäkseen projektissa käytettyjä resursseja. Tällainen on esimerkiksi Boehmin (1981) COCOMO-malli (*Constructive Cost Model*). Projektiin nähdyn vaivan, käytetyn ajan ja henkilöstövoimavarojen perusteella COCOMO laskee hinta-arvion tietyllä oletusarvoisella henkilöstövoimavaran arvolla. Esimerkiksi:

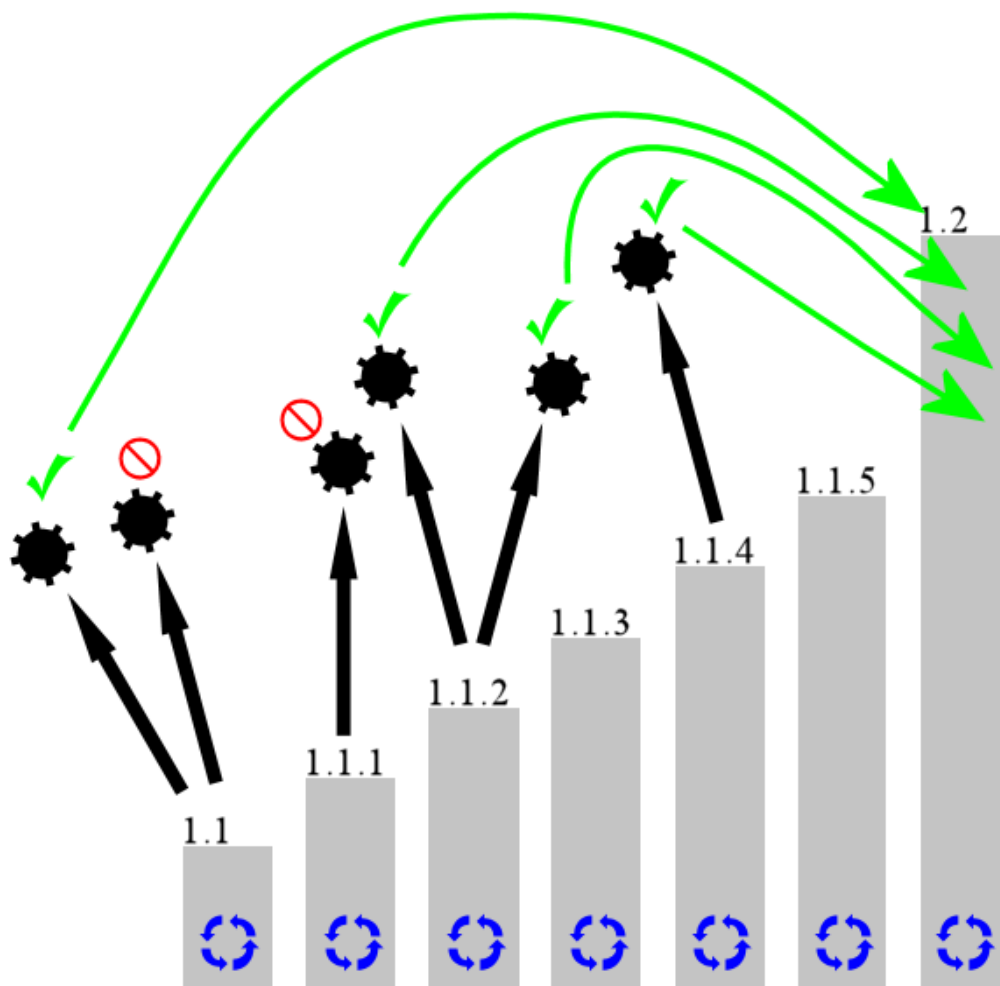
Projektin koko	$225\,371 \text{ loc}^1 = 58 \text{ htv}^2$
Ohjelmoijan palkka	$55\,000 \text{ USD} / v^3$
<hr/>	
Projektin arvo	3 202 250 USD

1 *loc = line of code, yksi rivi lähdekoodia – yleisesti käytetty tapa mitata tietokoneohjelman kokoa*

2 *htv = henkilötyövuotta*

3 *Ohjelmoijan keskipalkka Yhdysvalloissa Kenexa-yhtiön ylläpitämän Salary.com-palvelun tietojen perusteella*

Kuten kaikkien ohjelmistotuotteiden, OpenTTD:n kehitystä mitataan versioinnin etenemisellä. Ludvig Strigeuksen vuoden 2004 ensiprototyyppi oli versio 0.1.1. Avoimen ohjelmistonkehityksen graalin malja, niin sanotun valmiin ohjelmiston rajapyykki 1.0 ylitettiin aprillipäivänä 2010. Tälle rajapyykille asetettu tavoite oli pelin pelattavuus ns. itsenäisenä pelinä, ilman binäärimuotoista dataa alkuperäisestä Transport Tycoonista, joka oli ollut välttämätön komponentti versioon 1.0 asti. Kirjoitushetkellä uusin versio on 1.2.1.



Kuva 3.4 Versiomuotoisen kehitystyön pääkomponentit: kehitys, korjailu ja prototyyppien kehitys.

Kuvaajassa 3.4 on esitetty kolmiosainen projektin **tuotekehitysprosessi** versiosta 1.1 versioon 1.2. Prosessin komponentit ovat

- kehitys
- vikojen korjaus
- prototyyppien kehitys.

Harmaa portaikko kuvaa ohjelmiston kehitystä. Yksi porras on yksi ohjelmistoversio. Yhteen versioon on kerätty edellisestä korjatut ongelmat ja mahdolliset uudet ominaisuudet. Sininen kiertokuvio kuvaa vikojen eli ns. bugien korjausprosessia. Prosessi on jatkuva ja tapahtuu syvemmällä kuin versiotasolla. Ohjelmistoa on mahdollista kehittää ja testata päivittäisversioiden (*nightly build*) ja jopa ns. veitsenteräversioiden (*CVS build*) tasolla. Mustat hammaspyörät kuvaavat komponenttiprototyyppijä, jotka haarautetaan pääversioista jatkokehitykseen. Ne joko menestyvät tai lakkautetaan. Jos prototyyppi menestyy ja aikanaan valmistuu, se sulautetaan pääjulkaisuversioon. Tuotekehitysprosessiin osallistuvat sekä viralliset että epäviralliset kehittäjät.

Grafiikkaprojektin tuotekehitys on huomattavasti yksinkertaisempi kokonaisuus. Grafiikkaobjektit ovat irrallisia komponentteja, joten niiden irrotus korjausta tai kehitystä varten on helppoa. Tuotteesta julkaistaan yksi versio päivässä.

Laadunvalvonta on tuotteistuksen osa-alue, joka harvoin toteutuu riittävällä tasolla avoimessa ohjelmistonkehityksessä. Ohjelmistot ovat usein tiettyyn tarpeeseen tehtyjä ratkaisuja (*every good work of software starts by scratching a developer's personal itch*, totesi Eric S. Raymond 1997) – niinpä niiden toiminnan helppouteen, graafiseen ilmeeseen tai muuten vain yleiseen viimeistelyyn kiinnitetään vähän tai ei lainkaan huomiota, kunnes projekti alkaa selvästi kärsiä huonosta ilmeestä.

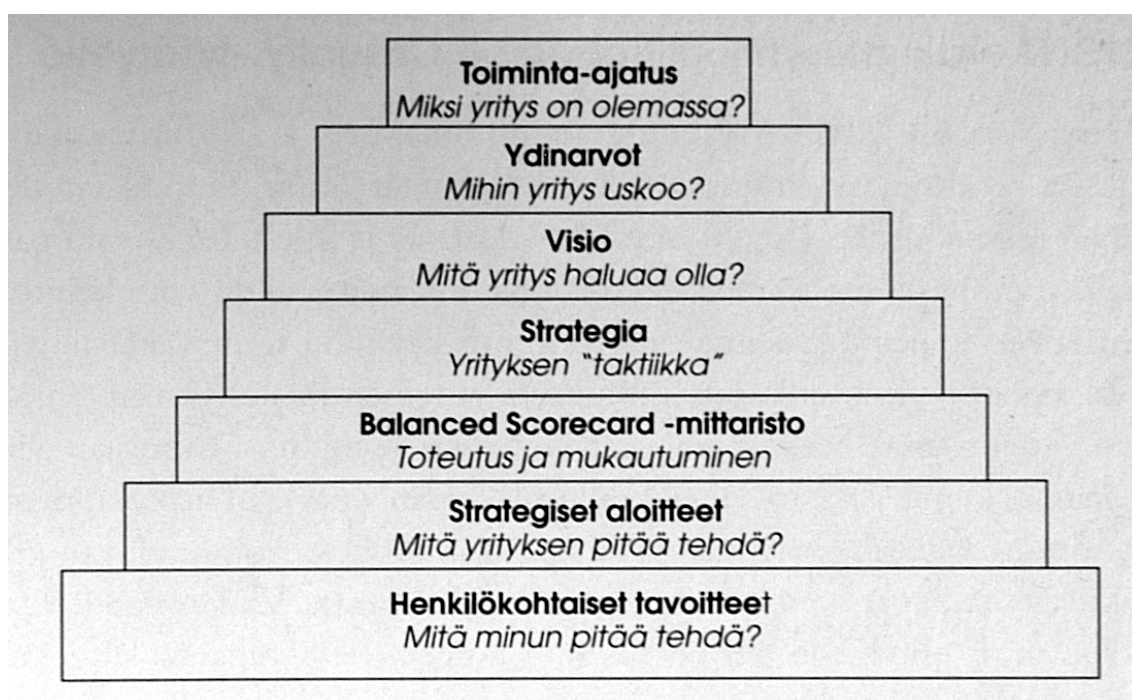
OpenTTD:ssä ei ole laadunvalvontaelintä. Yhteisö muodostaa filtterin, jonka läpi ei pääse massiivisen huono käytettävyys tai yleisilme. Toisaalta uusissa komponenteissa on usein tavoitteena saavuttaa yleisöä, eikä tavoite täyty, jos yleisö ei ota komponenttia käyttöön huonon käytettävyyden takia.

OpenTTD:n laatu käytettävyydessä, grafiikassa, teknisessä toteutuksessa ja luotettavuudessa on mielestäni keskimääräistä parempaa. Asiasta ei ole tehty tutkimuksia. On kuitenkin tosiasia, että OpenTTD:n tekninen pohja ei ole muuttunut vuosikymmeneen, joten hiomista on tapahtunut huomattavan pitkä aika suhteutettuna keskimääräiseen tietokoneohjelmistoon.

Grafiikkaprojektiin muodostui projektikoordinaattoreista laadunvalvontaelin projektin saavutettua noin viiden vuoden iän. Projektissa tunnistettiin tarve keskitetylle laadunvalvonnalle, joka kiinnittää huomiota grafiikkakomponenttien esteettiseen yhteensopivuuteen ja toteutuksen tekniseen laatuun. Pelin mukana toimitettavan grafiikkapaketin yleiselle laadulle asetettiin tietty taso.

3.4 Toiminta-ajatus, arvot ja visio ohjaavat strategiaa

Aivan kuin pelialan yrityksissäkin, tai yritysmaailmassa yleisesti, on avoimen ohjelmistonkehityksen projekteissa tehty vaihtelevasti analyysiä organisaatioiden luonteesta. Tähän analyysiin voi käyttää esimerkiksi seitsentoista pyramidia, jossa organisaation luonteenpiirteitä tunnistetaan:



Kuva 3.5: Organisaation luonneanalyysin portaat. (Kaplan & Norton 2002, 81)

Joomla (2012), avoin ohjelmistonkehitysprojekti, joka tuottaa samannimistä Internet-sisällönhallintasovellusta, on esimerkiksi julkaissut nettisivuillaan toiminta-ajatuksensa eli missionsa, arvonsa ja visionsa. Joomla'n missio on "tarjota joustava alusta digitaaliseen julkaisuun ja yhteistyöhön". Sen ydinarvot ovat vapaus, yhdenvertaisuus, luottamus, yhteisöllisyys, yhteistyö ja käytettävyys.

Joomlan (2012) visio on monikohtainen:

- Ihmiset käyttävät Joomlan ohjelmistoa julkaisuun ja yhteistyöhön ympäri maailman.
- Joomla on ilmainen, turvallinen ja korkealaatuinen.
- Joomla-yhteisö on nautinnollinen ja palkitseva osallistuville.
- Ohjelman käyttäjät ympäri maailman käyttävät sitä omalla kielellään.
- Projekti toimii itsenäisesti.
- Projekti on sosiaalisesti vastuullinen.
- Projekti on sitoutunut ylläpitämään sen käyttäjien luottamusta.

OpenTTD toisaalta ei ole julkaissut toiminta-ajatusta, arvoja tai visiota. Organisaatioidentiteetin kannalta voisi olla suotavaa, että ne mietittäisiin ja julkaistaisiin. Periaatteessa ne voisivat olla lähes samat kuin Joomlan – ovathan Joomla ja OpenTTD luonteeltaan samanlaisia organisaatioita, vaikka tuotteet eivät ole samanlaisia. Poikkeuksena ovat Joomlan luotettavuuteen ja luottamukseen liittyvät kohdat, jotka johtuvat web-sovellusten luontaisista tietoturvariskeistä.

Omasta näkökulmastani yksi tärkeimpiä kohtia vuoden 2012 alussa julkaisemassani grafiikkaprojektin projektikuvauksessa oli mission ja vision asetus. Missio kuului yksiselitteisesti ”Teemme OpenTTD:stä kauniimman”. Tämän taustalla oli kiteyttää lyhyeen lausahdukseen projektin toiminnan tarkoitus ja lopputuotteen luonne – tarkoitus on joskus tulevaisuudessa liittää grafiikkaprojektin tuote, grafiikkapaketti, OpenTTD-pelin julkaisuversioon.

Grafiikkaprojektin vision asetin seuraavasti:

- Grafiikkakehittäjiä eivät häiritse keskeneräiset, ristiriitaiset tai epäselvät ohjeet, ja objektien matka ideasta lopputuotteeksi on mahdollisimman suora.
- OpenTTD tukee kaikkia kolmea grafiikkaprojektin tukemaa zoom-tasoa (asetin tämän tavoitteen 2010 ja se saavutettiin jouluaattona 2011).

- Kaikki grafiikkakehitys tapahtuu tukemalla edellä mainittuja kolmea zoom-tasoa (tilanne tämän suhteen on parantunut vuoden 2012 aikana merkittävästi).
- OpenTTD:n nettisivujen kautta tuleva uusi pelaaja voi ladata peliin sisäänrakennetun latauspalvelun tai muun palvelun kautta ilmaiseksi tasokkaan 32-bittisen grafiikkapaketin, joka tukee kaikkia edellä mainittuja zoom-tasoja.
- Käyttäjät voivat ladata ja pelata 32-bittisiä grafiikkalaajennuksia käyttämällä pelin sisäistä latauspalvelua (tämä toteutui huhtikuussa 2012).
- Kuka tahansa käyttäjä voi ladata grafiikkojen lähdemateriaalit tehdäkseen niihin haluamansa muutokset (tätä kohtaa rikkovia komponentteja ei enää hyväksytä projektiin).

Tämän vision pohjalta on mahdollista lähteä rakentamaan projektin strategia tulevaisuuden toimintaan. Avoimessa ohjelmistonkehitysprojektissa on strategiaa laadittaessa syytä kiinnittää huomiota seuraaviin asioihin:

- Strategian voi laatia ja ”myydä” henkilöstölle kuka tahansa, toisin kuin yritysmaailmassa, missä strategian asettaa yrityksen ylin johto.
- Oli strategia kenen tahansa asettama, se joko hyväksytään yhteisössä, tai kehitys kohti mieluisaa tavoitetta tapahtuu muuta polkua, toisin kuin yritysmaailmassa, missä henkilöstön tehtävänä on toteuttaa johdon asettamaa strategiaa.
- Strategialla on onnistumisen edellytykset vain, jos valtaosa sidosryhmistä on samaa mieltä sen tarkoituksenmukaisuudesta ja toteuttamiskelpoisuudesta, aivan kuten yritysmaailmassakin.
- Strategian onnistuminen on todennäköisintä silloin, kun siinä hyödynnetään mahdollisimman paljon olemassa olevaa infrastruktuuria, toimintatapoja. Avoimissa ohjelmistonkehitysprojekteissa laajat muutokset (ns. revoluutiot) ovat vaarallisia ja voivat jopa tappaa projektin, sillä kukaan ei halua ilmaiseksi tekemänsä työn häviävän savuna ilmaan. Yritys-maailmassa päin vastoin muutosvastarinta on hyvin vaarallista

erityisesti ICT-alalla, jolle on ominaista nopea liikkuminen kuluttajatottumusten ja teknologisen kehityksen mukana. Matkapuhelimia valmistava yritys saattaa hylätä vuosikymmenen ajan kehitetyn ohjelmistoympäristön ja siirtyä toiseen, joka on vieras sekä käyttäjille että suurelle osalle kehittäjistäkin.

- Missään nimessä ei kannata vapaaehtoisesti rikkoa yhteensopivuutta olemassa olevan sisällön kanssa, vaikka niin tekeminen edistäisi jotain valittua strategiaa.

Kuten yritysmaailmassakin, tärkeitä on määritellä strategia riittävällä tarkkuudella ja varmistaa, että henkilökunta on ajan tasalla sen muodosta. Henkilökunnan on tiedettävä, kuinka strategiaa toteutetaan käytännössä. *Measures that Matter* -tutkimuksessa todetaan: sijoittajat arvostavat kykyä toteuttaa strategiaa enemmän kuin strategian varsinaista sisältöä (Ernst & Young 1997).

Strategialähtöinen yritystoiminnan analyysi on suhteellisen uusi tieteenala. Teollisella aikakaudella, vuoteen 1975 asti, yritysten toiminnan ja tehokkuuden mittaamiseen käytettiin taloudellisia indikaattoreita. Informaatioaikana, jota nyt elämme, korostuvat

- toimintojen ristikkäistoiminta
- arvoketjun vertikaaliset virrat
- markkinasegmentointi
- globalisaatio
- innovointi
- henkilöressurssien mahdollisimman tehokas hyödyntäminen

(Kaplan & Norton 1996, 4–5, suomenos tekijän)

Robert S. Kaplan ja David P. Norton kehittivät 90-luvun alussa informaatioajan strategialähtöiseen yritysanalyysiin niin sanotun *Balanced Scorecardin*. Siinä yrityksen toimintaa tarkastellaan neljästä näkökulmasta (taloudellinen, asiakaslähtöinen, sisäinen, ja oppimisen ja kasvun näkökulma), ja jokainen

osa-alue jaetaan neljään analyysikohteeseen (teemat, tavoitteet, toimenpiteet ja mittarit).

Kaplan ja Norton kommentoivat voittoa tavoittelemattomien organisaatioiden strategisesta johtamisesta:

Voittoa tavoittelemattomien organisaatioiden [...] on kiinnitettävä erityistä huomiota toiminta-ajatuksensa kommunikointiin ja määriteltävä tarkasti tavoitteensa ja toimintatapansa, joihin pohjaten niiden toimintaa tulisi mitata.

[...]

Samalla tavalla kuin valtion virastoissa on niiden toiminnassa taloudellinen näkökulma enemmän rasite kuin tavoite.

(Kaplan & Norton 1996, 185, suomennos tekijän)

Edellä mainitussa teoksessa esitellään USA:n Massachusettsin osavaltion erityisolympialaisten Balanced Scorecard -analyysia. Siitä mainitaan, että se oli koostumukseltaan lähes identtinen yritysmaailmassa tehtyjen BSC-analyysien kanssa.

Avoimen ohjelmistonkehitysprojektin strategisessa johtamisessa on vältettävä yritysmaailmassakin tuttuja sudenkuoppia. Kaplan ja Norton esittävät neljä tyypillistä estettä strategian onnistumiselle:

- Strategiaa ei muotoiltu liiketoimintaoperaatioissa hyödynnettäviksi toimintamalleiksi.
- Organisaation tai organisaatioyksikön strategiset tavoitteet eivät järkevästi siirry osasto-, tiimi- tai henkilökohtaiselle tasolle.
- Strategiassa asetetuille prioriteeteille ei anneta riittävästi resursseja.
- Strategian toteuttamisesta ja toteutumisesta ei kommunikoida riittävästi.

(Kaplan & Norton 1996, 193–196, suomennos tekijän)

OpenTTD-projektin analysointi Balanced Scorecardilla olisi palkitseva toimenpide, ja jo itsenäisenäkin tehtävänä yhden opinnäytetyön laajuinen suoritus.

3.5 Juridiikka

Grafiikkaprojektin juridiset ominaisuudet periytyvät osittain sen isäprojektilta OpenTTD:ltä. Peliä ympäröi *GNU General Public License* -ohjelmistolisenssi, joka asettaa pelin lähdekoodille, tuotteen käytölle, projektin hallinnolle ja muille aspekteille tietyt vaatimukset. Lisenssin vaatimukseen kuuluu lähdekoodin ja muiden pelin julkaisuun käytettyjen materiaalien saattaminen yleisön saataville, ja mahdollisuus pelin vapaaseen muokkaukseen.

Lisenssi edellyttää myös, että kaikki pelin komponentit ovat saman lisenssin tai yhteensopivan lisenssin alaisuudessa. Niinpä grafiikkaprojekti on myös GPL-lisensioitu tuote. Se ei ollut sitä aina – alussa OpenTTD:n grafiikkaprojektien juridiikkaan ei ollut kukaan perehtynyt, ja grafiikkaobjekteja julkaistiin osaksi pakettia sekalaisten lisenssien tai ei minkään lisenssin alla. Asia otettiin kuitenkin esille 2008 ja tehtiin selväksi, että GPL-lisensiointi on ainoa yksinkertainen tapa lisensioida viralliset pelin grafiikkapaketit. Tammikuun lopussa 2012 julkaistiin valmistelemäni projektikuvaus, jonka yksi komponentti oli pakollinen GPL-lisensiointi. Tämä tarkoittaa grafiikkaobjektien kehittäjille sitä, että objektin valmistukseen käytetty lähdemateriaali ja välivaiheet on julkaistava yleisölle. Tämä voi olla artistin kannalta mieluisa tai epämieluisa ominaisuus. Se on kuitenkin elinehto grafiikkapaketin lailliselle julkaisemiselle.

Grafiikkaprojektin yksittäisten objektien lisenssit määritellään julkaisuvaiheessa. Useimmiten se on projektin määräämä GPL, versio 2. GPL:n luonteen takia lisenssi periytyy myös johdetuille töille (*derivative works*). Näitä voivat olla esimerkiksi yksittäisestä puusta koostuvat metsät. Myös sellaisessa tapauksessa, jossa tekstuuri on tietyn lisenssin alainen, kyseinen lisenssi periytyy objektille, jossa tekstuuria on käytetty.

Grafiikkaprojektin räikein tekijänoikeusrikkomus tapahtui 2007, kun venäläinen nimimerkki Sergej_S latasi keskustelupalstalta eri artistien työn alla olevia grafiikkaobjekteja ja julkaisi niitä huonosti hiottuina pelikomponentteina oman nimimerkkinsä alla. Hän julkaisi pakettien yhteydessä alkuperäisten artistien nimet, mutta komponenttien huono laatu ja se, että ne oli tehty keskeneräisistä prototyyppitasen objekteista, aiheuttivat närää artistien keskuudessa. Sergej_S kieltäytyi muuttamasta toimintatapaansa ja sen seurauksena hänet suljettiin pois yhteisöstä. Tämä on avoimen ohjelmistonkehityksen sisäistä kärkeä-oikeutta: häiriötekijät ja ihmisten vaivannäön loukkaajat poistetaan kuin voikukat nurmikolta.

OpenTTD-projektin kaltaisissa Internet-yhteisöissä on tyypillistä, että oikeustapauksissa noudatetaan sen maan lainsäädäntöä, jossa tapahtumapalvelun fyysinen perusta eli palvelin sijaitsee. OpenTTD:n tapauksessa tämä maa on Iso-Britannia.

Euroopan Unionin sisäisen poliisiyhteistyön avulla olisi mahdollista saada rikosoikeudelliseen vastuuseen henkilöitä EU:n sisäisesti ja mahdollisuuksien mukaan maista, joiden kanssa Britannialla on asiaan liittyvät sopimukset. Rikostapauksissa nimimerkkien yhdistäminen oikeaan henkilöön on IP-osoitteen perusteella mahdollista, mutta ei sataprosenttisen luotettavaa.

Keskustelufoorumilla käytyjen keskustelun tulee noudattaa Britannian lakeja. Keskusteluja valvotaan ja siivotaan keskustelun alueen henkilökunnan voimin. Pienissä rikkeissä ja etikettiasioissa yhteisö valvoo itseään. Merkittävässä väärinkäyttötapauksissa on mahdollista aloittaa asian käsittely Britannian oikeuslaitoksessa.

Lähtökohtaisesti peliprojektin, grafiikkaprojektin tai muiden peliin liittyvien projektien avainhenkilöt tai henkilökunta eivät ole rikosoikeudellisessa vastuussa projektin kehityksessä tapahtuvasta tiedonsiirrosta, kommunikoinnista tai muista prosesseista, joissa voi tapahtua väärinkäytöksiä.

3.6 Asiakaskunta

Avoimien ohjelmistoprojektien asiakaskunnasta voi tehdä muutamia oletuksia:

- Avoimen lähdekoodin ohjelmiston käyttäjä arvostaa nimenomaan avointa lähdekoodia – suljetut ohjelmistot eivät edes kilpaile samasta roolista.
- GPL:ää ja muita avoimia lisenssejä pidetään välttämättöminä.
- Ohjelmiston valintaperusteena käytetään tekijän motiiveja – jos ohjelmistolla ei pyritä tekemään rahaa, on sen tekemiseen paneuduttu eri tavalla, kenties paremmin.
- Yhteisöllisyyden ja harrastelijamiljöön luonne on viehättävä, mikä vetää puoleensa samanhenkisiä ohjelmistojen käyttäjiä.
- Ohjelmisto halutaan käyttöön vain ja ainoastaan ilmaiseksi.

OpenTTD-projektin asiakaskunta ovat pelin pelaajat. Asiakaskunnan koko on mahdotonta määrittää tarkasti, aivan kuten yritysmaailmassakin. Perinteisellä markkinatutkimuksella olisi mahdollista selvittää, millaisista yksilöistä asiakaskunta koostuu. Asiakaskunnan ominaisuudet ovat kuitenkin selvästi hyvin samanlaiset kuin avoimessa ohjelmistonkehityksessä yleisesti: asiakkaiden oletamat ovat pääasiassa ohjelmiston ilmaisuus, avoin lähdekoodi ja avoin lisenssi.

OpenTTD:n pelaajista suuri osa on todennäköisesti alkuperäisen Transport Tycoonin pelaajia ja siten 25–35-vuotiaita. Virallisista kehittäjistä jokainen on tähän asti ollut miespuolinen.

Pelillä on vain vähän varsinaisia kilpailijoita – kaupallisia niistä ovat Railroad Tycoon -sarja ja Locomotion, jotka ovat molemmat luonteeltaan hieman erilaisia pelejä. OpenTTD:n varsinainen kilpailija on TTDPatch, joka oli samanlainen Transport Tycoonin laajentava paketti. Molemmat ovat ilmaisia tuotteita ja niiden kehitys tapahtui samassa yhteisössä. Mikään ei estä pelaajia käyttämästä molempia.

4 IHMISENÄ AVOIMESSA OHJELMISTOPROJEKTISSA

Avoimilla ohjelmistoprojekteilla on yhteistä liiketoimintaorganisaatioiden kanssa se, että niiden alkukipinä on usein yhden ihmisen tai hyvin pienen ihmisjoukon aikaansaamassa ideassa. Tietokoneohjelmistojen rakentaminen ideasta toimivaksi prototyypiksi on nykyään hyvin edullista, jopa ilmaista, joten kynnyksellä lähteä toimimaan asian edistämiseksi on matala.

Seuraava askel on prototyypin ja sen kehittäjien organisoituminen avoimeksi ohjelmistonkehitysprojektiksi tai kehitystä tekeväksi organisaatioksi. Se tapahtuu yleensä siinä vaiheessa, kun prototyyppi saavuttaa yleisön tietoisuuden ja siinä tunnustetaan reaali maailman arvoa – oli se sitten taloudellista tai muunlaista.

Projektien sisäinen normien asetus ja henkilöroolitus tapahtuu organisoitumisen aikana ja jatkuvana prosessina projektin toiminnan ajan. Normien asettuminen ja roolien täyttäminen tapahtuu huomattavasti nopeammin, jos projekti perustuu johonkin aikaisempaan projektiin tai sen kehitystiimissä on paljon ennalta tuttuja toimijoita.

Avoimille projekteille ovat tyypillisiä organisaation sisäiset myrskyt. Kehittäjien välillä syntyy erimielisyyttä ohjelmien ominaisuuksista, teknisistä asioista tai kehitysfilosofiasta. Avoimen lähdekoodin kehitysmalli on tähän teknisesti hyvin sopeutuvainen: konfliktitilanteissa käy usein niin, että projekti *forkataan* eli käynnistetään rinnakkaisprojekti, jolla on oma organisaatio ja normisto. Tällöin molemmat filosofiat voivat menestyä, ja kilpailevien ohjelmien vaikutus on yleishyödyllisestä näkökulmasta edelleen vain positiivinen.

Esimerkkejä myrskyjä kohdanneista ja forkatuista projekteista ovat mm.:

- Ubuntu-Linux, joka forkattiin Debian-Linuxista. Ubuntu halusi tehdä erityisesti helppokäyttöisen työpöytä-Linuxin.
- LibreOffice, joka forkattiin OpenOfficesta. Oracle Corporation, joka ei ole ollut kovin myötämielinen avoimia ohjelmistoprojekteja kohtaan, osti OpenOfficen Sun Microsystemsiltä. Sun päin vastoin oli aikanaan tukenut avointa lähdekoodia.
- Applen Safari-nettiselain, joka on forkattu WebKit-projektista. Seurauksena Safarin GPL-lisensioitujen komponenttien on aina säilyttävä avoimena lähdekoodina, vaikka itse Safari-selain on Applen suljettu tuote.

Elementit ovat monelle lukijalle tuttuja, mikä ei ole ihme, sillä ne ovat Bruce W. Tuckmanin jo v. 1965 julkaisemat neljä ryhmän muodostumisen perusvaihetta: *forming*, *storming*, *norming*, ja *performing*.

Eroavaisuus normaalin tiimityön ja ohjelmistonkehitysprojektien välillä näyttääkin olevan myrskyvaiheessa. Avoimissa ohjelmistonkehitysprojekteissa se näyttää alkavan vasta ikään kuin suoritusvaiheen sisäisesti. Järjestys on siis pikemminkin *forming*, *norming*, *performing*, *storming*. Myös McGrew, Bilotta ja Deeney (*Software Team Formation and Decay*, 1999) ovat sitä mieltä, että elementtien järjestys vaihtelee tapauskohtaisesti. (Robbins ym. 2010, 231)

4.1 Motivaatio

Kuten mainitsin aiemmin opinnäytetyössä, on erityisesti suurissa avoimissa ohjelmistoprojekteissa toimiminen käynyt työstä. Se on perustunut vastuuseen tietyssä tehtävässä, ja nautinto on ollut kaukana, kun tehtävänä on ollut käyttäjien neuvominen, ongelmanratkaisu tms. ja työt on tehtävä, vaikka ei olisi kiinnostustakaan.

Motivaattorina ei ole raha, vaan selvästi jokin muu. Mikä saa ihmisen tekemään ohjelmistonkehitys- tai helpdesk-työtä ilmaiseksi? Ohjaajieni kanssa käydyn keskustelun perusteella ulkopuolisen silmin mahdollisia motivaattoreita ovat:

- Tekemisen hauskuus harrastettaessa
- Itse pelin kehittyminen ja mahdollisuus vaikuttaa projektin lopputulokseen
- Kunnioituksen kerääntyminen yhteisön keskuudessa
- Tekemällä oppiminen, taitojensa parantaminen ja siten henkilökohtainen hyöty

Keskeinen asia on taloudellisen motivaattorin puuttuminen.

Omasta puolestani puhuen ratkaiseva motivaattori on nimenomaan harrastus. Harrastusmielisyys on moniulotteinen käsite, johon sisältyvät esimerkiksi:

- Työtehtävien vapaus
- Aikataulun joustavuus tai jopa puute
- Eskapismi – siis uppoutumisella tekemiseen pyritään pakenemaan arjen oravanpyörästä

Hauskuus on vaarallinen käsite tällaisen harrastuksen tapauksessa. OpenTTD-projektin parissa työskentely ei ole tarjonnut koskaan hauskuutta huvipuisto-merkityksessä. On palkitsevaa nähdä oman tuotteen prototyyppinä, ja vapaassa organisaatiossa toimiminen on rentoa, mutta työtehtävät eivät ole hauskoja tehdä, vaikka asiaa vääntäisi mihin suuntaan.

On selvää, että projektissa toimiminen on sisältänyt henkilökohtaisia hyötyjä, kuten oppimista. Esimerkkejä voisivat olla kielitaidon kehittyminen, tutustuminen C++-ohjelmointiin, php-ohjelmointitaitojen kehittyminen ja MySQL-tietokantojen nuanssien hiominen. Lisäksi toiminta on muokannut radikaalisti tapaa, jolla ajattelen yleisesti projekti- ja tiimityöskentelystä.

On syytä huomata, että motivaatiotaso kehittäjien keskuudessa nousee silmännähtävästi, kun projekti osoittaa eloa. Tämä on nähtävissä mm. uusina

komponentteina, uutena lähdekoodina tai uutena sisältönä. Kuolleet projektit eivät houkuttele kehittäjiä.

Strategisen johtamisen perspektiivistä voisi siis sanoa, että ilmaistyö jonkin tavoitteen saavuttamiseksi ei ole hukkaanheitettyä vaikka kyseinen tavoite ei olekaan lopullinen tai päätetään myöhemmin muuttaa tavoitetta. Edistys vääräänkin suuntaan osoittaa eloa, ja tämä on tärkeää vapaaehtoisten kehittäjien houkuttelemiseksi.

Avoimen ohjelmistonkehityksen motivaattoreita ovat tutkineet useat akateemikot. T. Mikkonen, T. Vadén ja N. Vainio (2007) ehdottavat tutkimuksessaan *The Protestant Ethic Strikes Back: Open Source Developers and the Ethic of Capitalism* motivaattorien jaottelua sen mukaan, kehen hyöty kohdistuu (suomennos tekijän):

Oma hyöty

- *Rahallinen hyöty*
- *Tulevaisuuden mahdollisuudet*

Ryhmädynaaminen hyöty

- *Seura*
- *Tiedon jakaminen*
- *Maine ja kunnia*
- *Nautinto*
- *Altruistisuus*

Yleishyödyllisyys

- *Ohjelmiston parantaminen*
- *Oman tietotason kohentaminen*

Oman hyödyn, ryhmädynaamisen hyödyn ja yleishyödyllisyyden jaottelu perustuu Martine Aalbersin tutkielmaan *Motivation for participation in an open source software community* vuodelta 2004.

Mikkonen, Vadén ja Vainio (2007) havaitsivat tutkimuksessaan, että neljän tutkitun projektin joukossa oma hyöty oli selvästi häviävässä asemassa. Jokaisessa projektissa ryhmädynaaminen hyöty ja yleishyödyllisyys olivat lähekkäin, ja jokaisessa yleishyödyllisyys oli tärkein motivaattori. Projektit olivat Debian-Linux-käyttöjärjestelmä, Eclipse-kehitysympäristö, Gnome-työpöytä-ohjelmisto, sekä MySQL-tietokantaohjelmisto.

Valitettavasti tutkimus avoimen ohjelmistonkehityksen motivaattoreista sekä tarkempi vertaileva pohdinta avoimen ohjelmistonkehityksen ja yritysmaailman henkilöstön motivaattoreista eivät mahdu tähän opinnäytetyöhön. Mainitut aiheet ovat silti jännittäviä ja ehdottomasti tutkimisen arvoisia.

4.2 Roolit

Aivan kuin sitä ympäröivässä peliprojektissa, OpenTTD:n grafiikkaprojektissa jokaisella kehittäjällä ja aktiivisella käyttäjällä on oma ydinosaamisensa, johon he rooliutuvat. Esimerkkejä näistä rooleista voivat olla ohjelmoija, grafiikka- tai ääniantisti ja organisoiija. Tuo jaottelu toimii kumpaakin projektia tarkasteltaessa.

Yhteisön jäsenen oltua tehtävässä ajan, joka on huomattavasti ympäröiviä toimijoita pidempi, tunnustetaan senioriteetti yleensä riidatta. Tällaisissa tilanteissa on hyvin luonnollinen väylä yletä projektikoordinaatioon esimerkiksi *Lead Developer* -nimikkeellä, jos yleneminen tapahtui virallisen kehitystiimin sisällä, tai *Developer*-nimikkeellä, jos yleneminen tapahtui epävirallisista kehittäjistä viralliseen kehitystiimiin.

Lead Developer on esimiesasemassa oleva ohjelmoija, joka ohjelmointitehtävänsä lisäksi hoitaa suunnittelu-, organisointi- ja koordinaatiotehtäviä. Tässä tehtävässä käytetään hyödyksi projektin parissa vietetyn ajan tarjoamaa asiantuntemusta ja kertyneitä kokemuksia hyvistä ja huonoista toimintamalleista. Lead Developer on vain rajoitetusti vastuussa tuotteessa esiintyvistä epäkohdista.

4.3 Normit

As a member of a group, you desire acceptance by the group. Because of your desire for acceptance, you are susceptible to conforming to the group's norms.

- Robbins, Judge & Campbell: Organizational Behavior (2010)

Normien käsitettä lähestytään Organizational Behavior -teoksessa siitä näkökulmasta, että yksilö on asetettu ryhmään, jolloin yksilö perii norminsa ryhmältä. Mielestäni avoimille projekteille on sitä vastoin tyypillistä, että ryhmä perii normistonsa yksilöiltä. Tämä johtuu siitä, että yksilöt ovat ryhmässä omasta tahdostaan. Projektin ydinarvot seuraavat siis sen normistoa, ja normisto noudattelee yksilöiden ydinarvoja. Niistä voi mainita:

- avoimuuden
- itsenäisyyden kaupallisista toimijoista
- vapauden käyttää mitä vain teknisiä ratkaisuja ja toimintatapoja
- taipumuksen seurata julkaistuissa ratkaisuisa omia mieltymyksiään
- halun miellyttää yleisöä.

Yksilöiden avoimuus ilmenee ryhmän taipumuksena hyväksyä estoitta uudet, asialliset jäsenet. Avoimuus on myös vapautta toimia oman aikataulunsa mukaan ja projektin ulkopuolisten mahdollisuus tutkia ja muokata tuotetta millä tahansa haluamallaan tavalla.

Yksilöiden halu irtaantua kaupallisten ohjelmistojen massasta näkyy pakonomaisena taipumuksena käyttää pelin yhteydessä ilmaisia avoimen lähdekoodin ohjelmistoja. Vain aniharvat kehittäjät, pääasiassa binäärisisällön kehitystiimeissä, käyttävät kalliita kaupallisia piirto- ja suunnitteluohjelmia grafiikan ja äänen tekemiseen. Kaikki OpenTTD:n komponentit ovat avointa lähdekoodia – edellytys sille tulee OpenTTD:lle asetetusta lisenssistä itsestään.

OpenTTD:hen ajettava lähdekoodi saa olla niin tyylikästä tai tehokasta kuin ohjelmoija on kykenevä kirjoittamaan. Kukaan ei määrittele tiettyä teknistä tapaa ratkaista käsillä oleva ongelma, eikä ajankäyttöä rajoita ns. *deadline*. Grafiikkaprojektissa kivet, penkit, rakennukset tai muut objektit voivat olla mallinnettuja oikeiden olemassaolevien objektien mukaan, tai ne voivat olla keksittyjä. Poliitiikka muodostuu yksilöiden vapaudenhakuisuudesta: tahdosta saada toteuttaa itseään, miten haluaa. Tämä normi on tärkeä projektin luonnetta määrittävä tekijä.

Vapaus toteuttaa itseään johtaa taipumukseen tehdä tekniset ratkaisut omien mieltymyksien mukaan. Avoimen ohjelmistonkehityksen tuotteita on kritisoitu tilkkutäkkimäisestä rakenteesta ja ulkonäöstä, mikä on suoraa seurausta mainitusta vapaudesta. Mieltymykset voivat konkretisoida niinkin yksinkertaisissa asioissa kuin lähdekoodin kirjoitusasussa. Toisaalta mieltymykset voivat olla vaarallisia, kuten hutilointi graafisessa julkaisussa, mikä johtaa brändin arvon laskuun.

Yleisön palveleminen, siis halu miellyttää yleisöä, toteutuu luonnostaan kehittäjien osallistuessa tuotetta koskeviin keskusteluihin. Tuotteen tuntumaa muokataan ja ominaisuuksia lisätään ajoittain yleisön ideoiden mukaisesti.

On syytä huomata, että sääntöjä on kaikesta vapaudesta huolimatta. Ne ovat ryhmän normeja, jotka velvoittavat tietynlaiseen käyttäytymiseen. Nämä ovat ohjelmointia ohjaavia (tyyliseikkoja ja yhteensopivuutta varmistavia säänneltyjä rakenteita, kirjastoja, ym.), grafiikan ulkoasua ja yleisilmettä ohjaavia (valoisuus, väritasapaino, arkkitehtuuri, ym.), ja vastaavia kokonaispaketin synergiaa ylläpitäviä määritteitä. Nämä yhdessä tekevät harvinaisen poikkeuksen projektin muuhun avoimuuteen. Lisäksi yksilöiden käyttäytymistä ohjailee silmännähtävästi tavallinen länsimaalainen siveellisyysnormisto. Kommunikointi näyttää hyvin asialliselta, yksilöitä ei hevin suljeta yhteisön ulkopuolelle ja toisia ryhmän jäseniä kunnioitetaan. Heidän mielipiteitään kysytään ja heitä kuunnellaan.

5 YHTEENVETO

Nyt, vuonna 2012, eletään avoimuuden ja tiedon vapaan kulun aikakaudella. Avoimen lähdekoodin ohjelmistokehitys on tällaisina aikoina luonteensa puolesta elementissään. Siksi avoin lähdekoodi on kuuma puheenaihe mediassa, tietokoneiden ja kuluttajaelektroniikan käyttäjäkunnan keskuudessa, sekä yksityisen ja julkisen sektorin organisaatioissa.

Avoimen lähdekoodin ohjelmistonkehityksessä lähdekoodi on vapaasti kaikkien saatavilla. Kuka tahansa voi muuttaa ohjelmistoa, kuka tahansa voi varmistaa ohjelmiston tekevän vain mitä sen kuuluukin ja kuka tahansa voi julkaista oman versionsa ohjelmistosta.

On tyypillistä, että avoimen lähdekoodin ohjelmistojen rakennus ja ylläpito tapahtuu harrastuksena. Lukemattomat laajalle levinneet ja rahallisestikin mittavat projektit ovat saaneet alkunsa yhden ihmisen tai pienen ihmisjoukon yhteisestä harrastuksesta, johon ei ollut sitoutunut lainkaan rahapääomaa. Vuonna 2008 kuitenkin ennustettiin, että vuonna 2012 kaupallisista ohjelmistoista 80 % pohjautuisi avoimen lähdekoodin komponentteihin. Suomessa sekä julkisen että yksityisen sektorin organisaatiot käyttävät avoimen lähdekoodin ohjelmistoja operatiivisissa toiminnoissaan.

Tässä opinnäytetyössä keskityin analysoimaan avoimen lähdekoodin OpenTTD-pelin kehitystä liiketalouden näkökulmasta. Pelissä on tavoitteena rakentaa ennen pelin alkamista sattumanvaraisesti generoituun pelimaailmaan tie-, rautatie-, meri- ja ilmakalustoon perustuva kuljetusverkosto. Kuljetuspalveluilla tehdään yritykselle tuloja, joilla laajennetaan toimintaa, jolla tehdään lisää tuloja ja niin edelleen. OpenTTD on hyvin tyypillinen harrasteena kehitelty avoimen lähdekoodin ohjelmisto. Projektin alku oli yhden henkilön kehittämä prototyyppi, joka saavutti suuren väkijoukon suosion ja kehittyi nykyään isomman kehitystiimin voimin.

Olen ollut vuodesta 2005 mukana OpenTTD-yhteisössä. Vuoden 2007 jälkeen olen keskittynyt projektiin, jossa pelin grafiikoita päivitetään modernille tasolle. Olen tehnyt koordinaatio- ja hallinnollisia tehtäviä, ja minulle myönnettiin 2010 Manager-arvonimi. Mikä on tuon tittelin arvo, kun kyseessä on harrasteprojekti, jossa liikkuu vähäinen määrä rahaa? Mielestäni manager- tai johtaja-arvonimen käyttäminen on oikeutettua, sillä tehtäväni ovat käsittäneet jossain määrin jokaista Robbinsin, Judgen ja Campbellin esittämää neljää keskeistä johtotehtävää: suunnittelua, organisointia, koordinoitua ja kontrollia.

Miksi avoimen lähdekoodin ilmaisia ohjelmistoprojekteja sitten tulisi analysoida liiketaloudellisesta näkökulmasta? Rahaahan niissä liikkuu vain absoluuttinen minimi. Syyn selviämiseksi on analysoitava *businessin* tai *liiketoiminnan* käsitettä. Termi on englantia, ja on sanaa lähemmin tarkasteltaessa adjektiivista *busy*, *kiireinen*, johdettu sana. Liiketoiminta on siis sananmukaisesti *kiireellisyttä*. Liiketoiminnasta käytetyksi sanaksi ei koskaan vakiintunut *earning*, *ansaitseminen*, tai *profiting*, *tuottaminen*. Itse asiassa jopa suomenkielinen käsite *liiketoiminta* käsittää vain *liikkumista*. Sanaa analysoimalla toimintaan ei liity raha. Voisi siis hyvin perusteiden kysyä, onko liiketoiminnassa pakko tehdä voittoa.

Projektia analysoidessa olikin mielenkiintoista huomata, kuinka monta liiketoiminnan laajan määritelmän osaa avoin ja ilmainen OpenTTD-projekti kattaa. Nähdäkseni ainoa oleellinen määritelmän osa, jota projekti ei kata, on taloudellinen perspektiivi. Liiketoimintaorganisaatioiden elementeistä on tässä avoimen lähdekoodin projektissa tunnistettavissa suurin osa: asiakaskunta, henkilöstö, kiinteätä omaisuutta, ydintuote, prosessit, ja niin edelleen. Mielestäni avoimen lähdekoodin ohjelmistoja kehittäviä yhteisöjä voisi verrata luonteeltaan voittoa tavoittelemattomiin ns. *non-profit*-organisaatioihin.

Avoimen lähdekoodin projektit organisoituvat eri tavoin, mutta yleensä niitä koskevat samat ominaisuudet kuin OpenTTD-projektia. Keskeisessä asemassa ovat aktiiviset ja viralliseksi tunnustetut kehittäjät. Nämä ovat projektissa pisimpään mukana olleita, kokeneimpia jäseniä ja heidän vaikutusvaltaansa päätöksentekoon ja kiistatilanteisiin liittyvissä asioissa ei aseteta kyseenalaiseksi. Yhteisöstä, epävirallisilta kehittäjiltä, tulee tuotteeseen enemmän tai vähemmän jatkuvana virtana muutos- tai laajennusosia. OpenTTD-projektissa hyödynnetään yritysmaailmasta tuttua aloitelaatikko-periaatteeseen rinnastettavaa aivomyrskyprosessia.

Avoimen lähdekoodin projekteissa esiintyvät samat roolit kuin ohjelmistoalan yrityksissäkin. Jonkun on hoidettava operatiivisia toimintoja, asiakaspalvelua, tuotesuunnittelua ym., eikä projektin kaikki toiminta voi tapahtua pelkästään itse ohjelmistonkehityksessä. Niin sanotusti epäoleellisten tehtävien hoitaminen voi tuntua välillä palkattomalta työltä. Silti yhteisön jäsenet ottavat tehtäviä tehtäväkseen. Tässä opinnäytetyössä ei ollut mahdollista tutkia empiirisesti, mitkä asiat toimivat vastaavan yhteisön motivaattoreina. Tämä ansaitsee mielestäni syvempää tutkimista.

On syytä huomata, että avoimen lähdekoodin ohjelmistonkehitysprojekteissa esiintyy hierarkisia organisaatorakenteita, joissa toimintamallit periytyvät ylhäältä alas, kuten liiketoimintaorganisaatioissakin. Vaikka OpenTTD-yhteisön arvoja ja normeja ei ole julkaistu missään, pätevät kehityksessä tietyt perusasiat, kuten vapaus ja avoimuus. Avoimen ohjelmistonkehityksen strateginen johtaminen on aivan omanlaisensa haaste. Kuitenkin se pohjautuu samalla tavalla arvoihin, normistoon sekä projektin tavoitteisiin kuin yritysmaailmassakin. Lisäksi se voi prosessina kaatua samoihin ongelmiin. Oleellista on määritellä strategia riittävällä tarkkuudella ja varmistaa, että henkilökunta on ajan tasalla sen muodosta. Henkilökunnan on oltava oleellinen osa strategiaa toteuttavaa voimaa. Olisi mielenkiintoista käyttää Balanced Scorecard -analyysiä avoimen ohjelmistonkehityksen projektien tutkimiseen. Valitettavasti siihen ei ole tässä opinnäytteessä tilaa.

Avoimia ohjelmistoja syytetään usein huonosta tuotteistuksesta. Erityisesti laadunvalvonta on tuotteistuksen osa-alue, joka harvemmin toteutuu riittävällä tasolla avoimessa ohjelmistonkehityksessä. Ohjelmistot ovat usein tiettyyn tarpeeseen tehtyjä ratkaisuja. Niinpä niiden toiminnan helppouteen, graafiseen ilmeeseen tai muuten vain yleiseen viimeistelyyn kiinnitetään vähän tai ei lainkaan huomiota. OpenTTD:ssä ei ole eriteltyä laadunvalvontaelintä.

Avoimen lähdekoodin ohjelmistoilla, kuten OpenTTD:lläkin, on tietyn tavoin rajattavissa oleva asiakaskunta. Käyttäjät arvostavat nimenomaan avoimuutta, mistä johtuen kaupalliset ja suljetut ohjelmistot eivät ole edes vaihtoehtoja. Myös ohjelmiston tekijöiden motiivit ovat tärkeitä asiakaskunnalle. Asiakaskunta pitää myös ohjelmistonkehityksen yhteisöllisyyttä ja kenties harrastelijamaista miljöötä viehättävänä. Lopuksi tuotteen ilmaisuus on varmasti vetoava ominaisuus. Olisi mielenkiintoista tehdä empiiristä tutkimusta siitä, millä tarkkuudella nämä oletamat pitävät paikkansa.

Avoimien ohjelmistoprojektien henkilöstön toiminnassa on paljon samaa, mutta myös jonkin verran erilaista kuin liiketoiminnassa. Vastaavasti kuin liiketoiminnassa, tuotteen ydinajatuksen ja projektin alkukipinän takana on usein yksi ihminen tai enintään pieni ryhmä. Projektin aikana tapahtuu jatkuvaa normien asetusta ja henkilöroolitusta tai -rooliutumista. Avoimen lähdekoodin luonteen vuoksi tyypillisiä ovat myös myrskyt, jotka johtavat projektien jakautumiseen, ns. *forkkaukseen*. Olen huomannut, että avoimessa ohjelmistonkehityksessä Tuckmanin neljä tiimityön perusvaihetta *forming*, *storming*, *norming* ja *performing* asettuvat eri järjestykseen. Tyypillinen järjestys on *forming*, *norming*, *performing* ja *storming*. Kaksi viimeistä vaihetta muodostavat kierteen, jonka aikana myrskyjä häviää projekteista ja saattaa syntyä rinnakkaisia versioita tuotteesta. Myös McGrew, Bilotta ja Deeney ovat *Organizational Behavior* -teoksen mukaan sitä mieltä, että elementtien järjestys vaihtelee tapauskohtaisesti (Robbins ym. 2010, 231).

Muille avoimen lähdekoodin ohjelmistonkehityksessä toimiville suosittelen lämpimästi seuraavia kantapään kautta oppimiani tehtäviä:

- Kommunikoinnin selkeyttäminen ja informaation organisointi on suureksi avuksi uusien henkilöresurssien tehokkaassa hyödyntämisessä henkilöstön suuren vaihtuvuuden vuoksi.
- Toiminta-ajatuksen, ydinarvojen, normiston ja strategian pohtiminen ja selkeä kommunikointi sidosryhmille auttaa projektia "löytämään itsensä" ja kehittymään nopeammin kohti sen lopullista tavoitetta. Eihän ole oleellista, mitä polkua tavoitteeseen päästään, kunhan sinne lopulta päästään.
- Roolien ja tehtävien selvä, mutta vapaamuotoinen asettaminen parantaa merkittävästi työtehoa kautta projektin, mikä lopulta vain parantaa tuotteen laatua.
- Kaupallisen ohjelmistonkehityksen roolit kannattaa sisällyttää yhteisön rooleihin. On muistettava, että ominaisuuksien kehittämisen lisäksi tehtävät, kuten laadunvalvonta ja asiakaspalvelu ovat oleellinen osa yleisön palvelemista.

Ohjelmistokehityksessä toimiville yrityksille suosittelen avointa suhtautumista konsepteihin, jotka ensisilmäyksellä voivat vaikuttaa utopistisilta tai avoimuuden aikakauden lyhytaikaisilta muoti-ilmiöiltä:

- Avoimuus tuotekehityksessä, operatiivisessa toiminnassa ja ongelmatilanteissa. Suosittelen lämpimästi kuuntelemaan oudoltakin kuulostavia ideoita ja erityisesti valmiiksi mietittyjä konsepteja. Molempien toteutusta voidaan harkita tapauskohtaisesti, ja yleensä niiden kohdeyleisö olisi suurempi kuin pelkkä idean esittäjä.
- Liian jäykän organisaatorakenteen välttäminen on oleellista työmotivaation kannalta, ja ohjelmistonkehityksessä nopeus ja joustavuus on suuri valtti.
- Lähdekoodin avaaminen on suureksi hyödyksi varsinkin niissä tilanteissa, kun ohjelmisto palvelee tietokannalla tehtävää liikeideaa, tai jos toimintaa tehdään rajallisella henkilöstöllä.

Minusta ei ole lainkaan utopistista ajatella, että avoimille ohjelmistoprojekteille ominaista liberaalisuutta ja myös käytännön avoimuutta saataisiin ajettua liiketoimintaan erityisesti tuotekehitys- ja asiakasrajapinnan prosesseihin. Tarkasteltavaa riittää projektijohtamisessa ja henkilöstöjohtamisessa. Kuinka avoimien ohjelmistoprojektien henkilöstön motivaattoreita saataisiin ajettua yritysmaailmaan ja siten parannettua työhyvinvointia?

Avoimilla ohjelmistonkehitysyhteisöillä on yhteisesti paljon opittavaa yritysmaailmassa toimimisesta. Se on tärkeää huomata erityisesti nyt, kun Ubuntu-Linux pyrkii Windowsin dominoimille markkinoille, Android pyrkii iOS:n markkinoille ja lukuisat muut avoimen lähdekoodin ohjelmistot laajentavat reviiriään. Tietokoneohjelmistot kehittyvät vuosi vuodelta ja niitä saa yhä halvemmalla. Mainosrahoitteiset työkalut ja pelit mobiilialustoille ovat usein peräti ilmaisia, mutta viimeistelyltään valovuosia avoimen lähdekoodin vastineita edellä. Olisikin syytä kiinnittää erityistä huomiota tuotteistukseen ja viimeistelyyn, kuten käyttöliittymäsuunnitteluun ja graafiseen toteutukseen.

Akateemisessa ympäristössä ja mediassa toimiville haluan painottaa avoimen lähdekoodin ohjelmistojen merkitystä nykypäivän elektronisessa elämässä. Kyse ei ole pelkästä muoti-ilmiöstä tai harrastelijoiden puuhastelusta, sen on avoimuuden ja vapauden liike ohjelmistonkehityksessä osoittanut 1990- ja 2000-luvun aikana. Sekä käyttöjärjestelmien, toimisto-ohjelmien että vapaa-ajan ohjelmien kentällä on vakavasti otettavia pelureita, jotka toimivat avoimen ohjelmistonkehityksen periaatteita noudattaen.

Tämän opinnäytetyön tekeminen oli luonnollinen yhteinen lopputulos harrastukselleni ja koulutukselleni. Kirjoitusprosessin aikana vahvistin näkemystäni siitä, että avoimet ohjelmistonkehitysprojektit sisältävät hyvin paljon samoja ominaisuuksia kuin liike-elämän ohjelmistonkehitys ja projektityö.

Avoim ohjelmistonkehitys tapahtuu minun silmissäni selvästi organisaatioissa, jotka ovat täysin verrattavissa liike-elämän vastineisiinsa. Avoimen lähdekoodin projektit ovat avoimempia, organisoitu eri tavalla, vapaampia ja niiden tuotteet ovat pääasiassa ilmaisia. Liiketalouden asiantuntijan silmät näkevät parannuskohteita, mutta toisaalta myös piirteitä, joita projektipäällikön tai henkilöstöjohtajan on syytä kadehtia. Tämän opinnäytetyön arvo on mielestäni kaksijakoinen. Se kohdistuu sekä avoimessa ohjelmistonkehityksessä toimiville, erityisesti projektikoordinaattoreille, että toisaalta liikejohdon parissa työskenteleville, erityisesti ohjelmistoalalla.

Henkilökohtaisella tasolla opinnäytetyöprosessilla oli selvä vaikutus. Huomaan kehittyneeni lähdekriittisemmäksi opiskelijaksi. Lisäksi haluan painottaa opinnäytetyöprosessin minulle vahvistamaa käsitystä ns. akateemisesta kirjoittamisesta. Vaikka olen nuoruudesta asti nauttinut kirjoittamisesta sen eri muodoissa, on tieteellisen tekstin kirjoittaminen omanlaisensa haaste. Asiallinen ja neutraali sävy ovat tietenkin oleellinen osa opinnäytetyön kirjoittamista, ja tässä on ohjaajani ollut suureksi avuksi. Haluan myös kiittää Jukka Hilvosta, Kirsi Viskaria ja Pasi Juvosta, jotka osallistuivat työhöni hyödyllisine kommentteineen.

KUVAT

Kuva 3.1 Tyypillisen ohjelmistoalan yrityksen rakenne, löyhästi mukailleen CCP hf:n organisaatorakennetta, s. 21

Kuva 3.2 OpenTTD-peliä kehittävän organisaation rakenne, s. 24

Kuva 3.3 Grafiikkaprojektin organisaatio, s. 27

Kuva 3.4 Versiomuotoisen kehitystyön pääkomponentit: kehitys, korjailu ja prototyyppien kehitys, s. 33

Kuva 3.5: Organisaation luonneanalyysin portaat, s. 36

LÄHTEET

Ars Technica. 2008. Gartner: 80% of commercial apps to use open source by 2012

<http://arstechnica.com/information-technology/2008/02/gartner-80-percent-of-commercial-software-programs-will-include-open-source-by-2012/> (Luettu 18.6.2012)

Black Duck Software, Inc. 2012. Ohloh: Estimated Cost, OpenTTD

https://www.ohloh.net/p/openttd/estimated_cost (Luettu 12.6.2012)

Cox, A. 1998. The Cathedrals, Bazaars and the Town Council

<http://slashdot.org/story/98/10/13/1423253/featurecathedrals-bazaars-and-the-town-council> (Luettu 12.6.2012)

Crowston, K. & Howison, J. 2005. The Social Structure of Free and Open Source Software Development”

<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1207/1127> (Luettu 12.6.2012)

Dictionary.com LLC. 2012a. Define Freeware at Dictionary.com

<http://dictionary.reference.com/browse/freeware> (Luettu 6.9.2012)

Dictionary.com LLC. 2012b. Define Shareware at Dictionary.com

<http://dictionary.reference.com/browse/shareware> (Luettu 6.9.2012)

Ernst & Young LLP. 1997. Measures that Matter

<http://www.valuementors.com/pdf/Measures%20that%20Matter.pdf> (Luettu 27.6.2012)

Jain, T. R., Trehan, M. & Trehan, R. 2009. Business Environment

<http://books.google.fi/books?id=nfADHVmJkTUC> (Luettu 27.6.2012)

Joomla. 2012. Mission, Vision & Values <http://www.joomla.org/about-joomla/the-project/mission-vision-and-values.html> (Luettu 27.6.2012)

Kaplan, R. S. & Norton, D. P. 1996. The Balanced Scorecard. Boston: Harvard Business School Press.

Kaplan, R. S. & Norton, D. P. 2002. Strategialähtöinen organisaatio. Jyväskylä: Gummerus.

McGrew, J. F., Bilotta, J. G. & Deeney, J. M. 1999. Software Team Formation and Decay <http://sgr.sagepub.com/content/30/2/209.abstract> (Luettu 2.7.2012)

Mikkonen, T., Vadén, T. & Vainio, N. 2007. The Protestant Ethic Strikes Back: Open Source Developers and the Ethic of Capitalism
<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1623/1538>
(Luettu 2.7.2012)

North Bridge Venture Partners. 2012. Sixth Annual "Future of Open Source" Survey Results Point to Era of "Open Innovation on Demand"
<http://northbridge.com/sixth-annual-future-open-source-survey-results-point-era-open-innovation-demand> (Luettu 27.6.2012)

Raymond, E. S. 1997. The Cathedral and The Bazaar
<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1472/1387>
(Luettu 12.6.2012)

Robbins, S. P., Judge, T.A. & Campbell, T. T. 2010. Organizational Behavior. Barcelona: Grafos

Singla, R. K. 2009. Business Studies
<http://books.google.fi/books?id=55eeqYklbuAC> (Luettu 17.9.2012)

Tuckman, B. W. 1965. Developmental Sequence in Small Groups. Psychological Bulletin 63, 396.

Wenger, E. 2006. Communities of Practice, a Brief Introduction
<http://www.ewenger.com/theory/index.htm> (Luettu 6.9.2012)