

hyväksymispäivä arvosana

arvostelija

Hajautetun ohjelmistokehityksen laatu

Tuomo Jokimies

Helsinki 05.11.2013

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta – Fakultet – Faculty		Laitos – Institution – Department	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen laitos	
Tekijä – Författare – Author			
Tuomo Jokimies			
Työn nimi – Arbetets titel – Title			
Hajautetun ohjelmistokehityksen laatu			
Oppiaine – Läroämne – Subject			
Tietojenkäsittelytiede			
Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages	
Seminaariraportti	05.11.2013	10 sivua + 1 liitesivu	
Tiivistelmä – Referat – Abstract			
<p>Maantieteellisen hajautuksen on uskottu heikentävän ohjelmistojen laatua monien siihen liittyvien haasteiden vuoksi. Tutkimustulokset kuitenkin osoittavat, että yhden organisaation sisällä tapahtuvalla maantieteellisesti hajautetulla ohjelmistokehityksellä on korkeintaan pieni heikentävä vaikutus ohjelmistojen laatuun. Sen sijaan organisaationaalisella hajautuksella on havaittu olevan vaikutuksia ohjelmistojen laatuun ja virheiden lukumäärään. Avoimen lähdekoodin projektit ovat puolestaan pärjänneet hyvin hajautuksen haasteiden kanssa, sillä niiden yhteistyömekanismit ovat hajautettuja.</p> <p>ACM Computing Classification System (CCS): D.2.8 [Metrics] D.2.9 [Management] K.6.4 [System Management]</p>			
Avainsanat – Nyckelord – Keywords			
Hajautettu ohjelmistokehitys, ohjelmistojen laatu, maantieteellinen hajautus, organisaationaalinen hajautus, avoin lähdekoodi			
Säilytyspaikka – Förvaringställe – Where deposited			
Muita tietoja – Övriga uppgifter – Additional information			

Sisältö

1 Johdanto	1
2 Laatu maantieteellisesti hajautetussa ohjelmistokehityksessä	2
2.1 Hajautuksen vaikutus ohjelmatiedostojen laatuun.....	2
2.2 Hajautuksen vaikutus ohjelmakooditiedostojen laatuun.....	4
3 Organisaation rakenteen vaikutus laatuun	5
4 Laatu avoimen lähdekoodin projekteissa	7
4.1 FreeBSD.....	7
4.2 Mozilla Firefox ja Eclipse.....	8
5 Yhteenveto	10
Lähteet	11

1 Johdanto

Ohjelmistokehitys voi olla hajautettu monella eri tavalla. Maantieteellisesti hajautettu ohjelmistokehitys tapahtuu useassa eri maantieteellisessä sijainnissa [BND09]. Organisaationaalisesti hajautettu ohjelmistokehitys puolestaan tapahtuu usean eri tahon kesken. Tahot voivat olla saman organisaation sisällä tai kokonaan toisia organisaatioita. [NMB08]

Maantieteellisen hajautuksen on uskottu heikentävän ohjelmistojen laatua monien siihen liittyvien haasteiden vuoksi [KZB13]. Maantieteellisesti hajautettuun ohjelmistokehitykseen yhdistetään monesti myös ulkoistettu ohjelmistokehitys (*offshoring*) [BND09]. Yhden organisaation sisällä tapahtuva maantieteellisesti hajautettu ohjelmistokehitys poikkeaa kuitenkin merkittävästi ulkoistetusta ohjelmistokehityksestä, sillä ulkoistetussa ohjelmistokehityksessä esiintyy monia ongelmia, jotka eivät liity pelkästään maantieteelliseen hajauttamiseen [BND09]. Kokonaan yhden organisaation sisällä tapahtuvaa maantieteellisesti hajautettua ohjelmistokehitystä käsitellään luvussa 2.

Organisaationaalisella hajautuksella on havaittu olevan vaikutuksia ohjelmistojen laatuun ja virheiden lukumäärään [BiN12, NMB08]. Conwayn laiksi kutsutun ajattelman mukaan organisaatiot, jotka suunnittelevat järjestelmiä, ovat rajoitettuja tuottamaan järjestelmiä, jotka ovat kopioita organisaation kommunikaatorakenteesta [Con68]. Myös Brooks esittää tunnetussa kirjassaan *The Mythical Man Month: Essays on Software Engineering* organisaation rakenteen vaikuttavan suuresti tuotteen laatuun [Bro95]. Organisaation vaikutusta ohjelmistojen laatuun käsitellään luvussa 3.

Avoimen lähdekoodin projektit ovat luonteeltaan maantieteellisesti ja organisaationaalisesti hajautettuja [BiN12]. Lisäksi niiden prosessimallit poikkeavat merkittävästi kaupallisista ohjelmistoista [BND09]. Joidenkin mukaan avoimen lähdekoodin projektit voittavatkin globaalin hajautetun ohjelmistokehityksen haasteet, sillä niiden yhteistyömekanismit perustuvat internetiin, joka on luonteeltaan maantieteellisesti ja organisaationaalisesti hajautettu [BiN12]. Laatua avoimen lähdekoodin projekteissa tarkastellaan luvussa 3.

2 Laatu maantieteellisesti hajautetussa ohjelmistokehityksessä

Tässä luvussa käsitellään ohjelmistojen laatua maantieteellisesti hajautetussa ohjelmistokehityksessä kokonaan yhden organisaation sisällä. Bird ja kumppanit [BND09] tutkivat hajautuksen vaikutusta ohjelmistojen laatuun ohjelmatiedostojen (*binary*) tasolla. Birdin ja kumppaneiden tutkimusta käsitellään luvussa 3.1. Kocaguneli ja kumppanit [KZB13] tutkivat tarkemmin, miten hajautus vaikuttaa ohjelmistojen laatuun ohjelmakooditiedostojen tasolla. Kocagunelin ja kumppaneiden tutkimusta käsitellään luvussa 3.2.

2.1 Hajautuksen vaikutus ohjelmatiedostojen laatuun

Bird ja kumppanit [BND09] tutkivat maantieteellisesti hajautetun ohjelmistokehityksen vaikutusta ohjelmistojen laatuun yhden organisaation ja projektin sisällä. Tutkimuksessaan he pyrkivät selvittämään, miten hajautettu ohjelmistokehitys vaikuttaa ohjelmiston julkaisun jälkeen ilmenneiden vikojen määrään tavanomaisempaan paikalliseen ohjelmistokehitykseen verrattuna.

Tutkimusaineistona Bird ja kumppanit käyttivät Windows Vista -käyttöjärjestelmän versionhallinnan muutostietoja, kehittäjien sijaintitietoja ja julkaisun jälkeen ilmenneiden vikojen tietoja. Ohjelmistokehitys tapahtui kokonaan yhden organisaation, Microsoftin, sisällä, ja siihen osallistui useampi tuhat kehittäjää. Kehittäjät koottiin Aasiasta, Euroopasta ja Pohjois-Amerikasta. Heitä oli 21 kampuksella 59 eri rakennuksessa.

Tutkimuksessa tarkasteltiin Windows Vista -käyttöjärjestelmän suoritettavia ohjelmatiedostoja (*binary*), joita oli tuhansia. Ohjelmatiedostot jaettiin kahteen kategoriaan: maantieteellisesti hajautetusti ja paikallisesti kehitettyihin ohjelmatiedostoihin. Ohjelmatiedosto luokiteltiin paikallisesti kehitetyksi, mikäli vähintään 75 % ohjelmatiedostoon tehdyistä muutoksista (*commit*) tuli samalta kampukselta.

Birdin ja kumppaneiden hypoteesina oli, että maantieteellisesti hajautetusti kehitetyt ohjelmatiedostot sisältävät enemmän julkaisun jälkeisiä vikoja kuin paikallisesti kehitetyt. Tutkimustulokset osoittivat, että hajautetusti kehitetyissä oli keskimäärin 9,2 % enemmän vikoja. Otettaessa ohjelmatiedoston parissa työskentelevien kehittäjien lukumäärä huomioon, vikoja esiintyi keskimäärin vain 4,6 % enemmän. Tällöin hajautuksella ei siis ollut suurta vaikutusta ohjelmistojen laatuun, ja täten Bird ja

kumppanit eivät voineet vahvistaa hypoteesiaan.

Tutkimuksessa selvitettiin myös, eroavatko hajautetut ohjelmatiedostot ominaisuuksiensa suhteen paikallisesti kehitetyistä, sillä muun muassa koodin monimutkaisuus ja riippuvuuksien lukumäärä vaikuttavat suuresti vikojen määrään. Ohjelmatiedostojen ominaisuuksien eroavaisuuksia tutkittiin koodirivien määrän ja kompleksisuuden, koodin muuttuvuuden (*code churn*), testien kattavuuden, riippuvuuksien tai koodin parissa työskennelleiden ihmisten pohjalta. Osa valituista mittareista näkyy taulukossa 1. Bird ja kumppanit eivät havainneet ohjelmatiedostojen ominaisuuksissa selvää eroa kaikkien mittareiden korrelaatiokertoimien ollessa alhaiset, alle 0,25.

Mittari	Keskiarvo	Korrelaatiokerroin	Merkittävyys
Funktioiden määrä	895,86	0,114	$p < .0005$
Kompleksisuus	4603,20	0,069	$p < .0005$
Muuttuvuus	53430,00	0,057	$p = .033$
Muutosten määrä	63,82	0,134	$p < .0005$
Ohjelmatiedoston riippuvuuksien määrä	13,04	-0,024	$p = .363$
Muiden riippuvuus ohjelmatiedostosta	9,67	0,100	$p < .0005$
Kehittäjien määrä	21,55	0,183	$p < .0005$

Taulukko 1: Hajautettujen ja paikallisesti kehitettyjen ohjelmatiedostojen ominaisuuksien eroavaisuus valituilla mittareilla [BND09].

Birdin ja kumppaneiden mukaan yksi tutkimustuloksia selittävä syy saattaa olla kehityspaikkojen välisissä suhteissa. Suurin osa aikaisemmasta tutkimuksesta on keskittynyt ulkoistuksiin liittyvään maantieteellisesti hajautettuun ohjelmistokehitykseen. Tällöin eri sijainneissa olevat ohjelmistokehittäjät eivät ole yhdenvertaisessa asemassa. Microsoftin käytäntönä oli myös kaventaa kulttuurieroista aiheutuvaa kiihkoa lennättämällä kehittäjiä eri sijaintien välillä. Kehittäjät pyrkivät myös kommunikoimaan synkronisesti joka päivä venyttämällä työpäiviään myöhempään tai saapumalla töihin aikaisin. Lisäksi projektin aikataulu oli sama eri puolella maapalloa työskentelevien kesken. Birdin ja kumppaneiden mukaan myös yhtenäiset työkalut ja vahva koodin omistajuus yhdessä alhaisen ja yhtenäisen organisaatorakenteen kanssa saattoivat vaikuttaa siihen, että laatu pysyi lähes samana maantieteellisestä hajautuksesta huolimatta.

2.2 Hajautuksen vaikutus ohjelmakooditiedostojen laatuun

Kocaguneli ja kumppanit [KZB13] tutkivat maantieteellisesti hajautetun ohjelmistokehityksen vaikutusta ohjelmistojen laatuun ohjelmakooditiedostojen tasolla – poiketen Birdin ja kumppaneiden [BND09] tutkimuksesta, jossa keskityttiin ohjelmakooditiedostoista käännettyihin ohjelmatiedostoihin. Tutkimusaineistona Kocaguneli ja kumppanit käyttivät Microsoftin Office 2010 -toimisto-ohjelmiston versionhallinnan muutostietoja, kehittäjien sijaintitietoja ja julkaisun jälkeen ilmenneiden vikojen tietoja.

Microsoft Office 2010 kehitettiin kokonaan Microsoftin sisällä, ja siihen osallistui yli 1500 kehittäjää. Ohjelmakooditiedostoja oli yli kymmenen tuhatta. Ohjelmakooditiedostot luokiteltiin pienimmän mahdollisen maantieteellisen alueen mukaan, jolta tuli vähintään 75 % tiedostoon kohdistuneista muutoksista. Luokkia oli viisi: rakennus, kaupunki, osavaltio, valtio ja maailma. Jokainen tiedosto luokiteltiin vain yhteen luokkaan. Rakennuksen tasolla maantieteellinen hajautus oli pienintä ja maailman tasolla suurinta. Luokkien pohjalta Kocaguneli ja kumppanit muodostivat neljä skenaariota, joissa tiedostot oli jaettu paikallisesti kehitettyihin ja maantieteellisesti hajautettuihin. Skenaariot on kuvattu taulukossa 2. Kocaguneli ja kumppanit tutkivat myös tiedostojen pääkehittäjien lukumäärän vaikutusta laatuun suhteessa hajautukseen. Kehittäjä luettiin tiedoston pääkehittäjäksi, mikäli hän oli tehnyt yli 40 % tiedostoon kohdistuneista muutoksista.

Skenaario	Paikallinen	Hajautettu
Rakennus	Rakennus	Kaupunki, osavaltio, valtio, maailma
Kaupunki	Rakennus, kaupunki	Osavaltio, valtio, maailma
Osavaltio	Rakennus, kaupunki, osavaltio	Valtio, maailma
Valtio	Rakennus, kaupunki, osavaltio, valtio	Maailma

Taulukko 2: Maantieteellisten alueiden jaottelu eri skenaarioiksi [KZB13].

Tutkimuksessa Kocaguneli ja kumppanit tutkivat viittä hypoteesia kaikkia neljää eri skenaariota vasten. Kahta hypoteesia he eivät onnistuneet kumoamaan. Toinen näistä hypoteeseista oli, että paikallisesti ja maantieteellisesti hajautetusti kehitetyillä ohjelmakooditiedostoilla, joilla oli vähintään yksi pääkehittäjä, olisi samanlainen julkaisun jälkeinen laatu. Toinen hypoteesi oli sama, mutta ilman yhtään pääkehittäjää. Näiden

hypoteesien tulokset eivät kuitenkaan olleet tilastollisesti merkittäviä p-arvojen vaihdellussa suuresti (0,24–0,83).

Viimeiset kolme hypoteesia Kocaguneli ja kumppanit kumosivat. Ensimmäinen hypoteesi oli, että paikallisesti ja maantieteellisesti hajautetusti kehitetyt ohjelmakooditiedostot ovat yhtä virheettiitä. Toinen hypoteesi oletti tiedostojen muutos- ja kokometriikoiden olevan samankaltaiset ja kolmas tiedostojen omistajuuden erityispiirteiden olevan samankaltaiset. Kocaguneli ja kumppanit tutkivat myös hypoteesien vaikutuksen kokoa ja havaitsivat, että kolmella viimeisellä hypoteesilla koko oli todella vähäinen. Kocaguneli ja kumppanit totesivatkin, että vaikka he onnistuivat kumoamaan kolme viimeistä hypoteesia, niiden vaikutus oli pieni ja merkityksetön. Täten Kocagunelin ja kumppaneiden tutkimustulokset olivat linjassa aikaisempien tutkimustulosten [BND09] kanssa.

3 Organisaation rakenteen vaikutus laatuun

Nagappan ja kumppanit [NMB08] tutkivat organisaation rakenteen vaikutusta ohjelmistojen laatuun. Tutkimuksessa Nagappan ja kumppanit rakensivat mallin, jolla pystyy ennakoimaan organisaation rakenteen vaikutusta julkaisun jälkeen ilmenevien virheiden määrään. Tutkimus perustui Microsoftin Windows Vista -käyttöjärjestelmästä kerättyihin versionhallinnan muutostietoihin, työntekijöiden tietoihin ja julkaisun jälkeen ilmenneiden virheiden tietoihin. Tutkitussa aineistossa oli 3 404 ohjelmatiedostoa, jotka sisälsivät yli 50 miljoonaa riviä koodia.

Nagappan ja kumppanit rakensivat mallin kahdeksan mittarin pohjalta. Ensimmäinen mittari oli ohjelmatiedostoa muokanneiden insinöörien lukumäärä. Nagappan ja kumppanit päättelivät, että mitä suurempi lukumäärä insinöörejä on muokannut tiedostoa, sitä todennäköisempää on, että siinä on virheitä. Toinen mittari oli ohjelmatiedostoa muokanneiden entisten työntekijöiden lukumäärä. Mittari kuvasi tietämyksen siirtoon liittyviä ongelmia. Kolmas mittari oli ohjelmatiedostoon tehtyjen muutoksien määrä. Mittarilla oli kaksi tarkoitusta: kuvata tiedoston vakautta sekä muutoksien jakautumista eri insinöörien kesken, eli sitä, tulivatko kaikki muutokset yhdeltä vai useammalta henkilöltä.

Neljäs mittari kuvasi ohjelmatiedoston omistajuuden syvyyttä koko organisaatiossa. Omistajuuden syvyys määräytyi sen mahdollisimman alhaalla olevan esimiehen tason mukaan, jonka alaiset tekivät yli 75 % ohjelmatiedostoon kohdistuneista muutoksista. Mitä syvempi oli omistajuus, sitä helpompaa olivat aktiviteetit, kommunikointi ja vas-

tuunjako ohjelmatiedoston parissa työskennelleiden insinöörien kesken. Pienempi omistajuuden syvyys saattoi aiheuttaa ongelmia muun muassa päätöksenteossa ja koodin ymmärtämisessä. Viides mittari suhteutti neljännen mittarin tasolla olevien alaisten määrän koko organisaation työntekijöiden määrään. Tällä mittarilla pyrittiin poistamaan epätasapainoisen organisaation vaikutus tuloksiin.

Kuudes mittari kuvasi sitä, kuinka suuri osuus ohjelmatiedostoon kohdistuneista muutoksista tuli samasta organisaatiosta, jossa ohjelmatiedoston omistaja oli. Nagappan ja kumppanit olettivat, että saman organisaation sisällä on parempi yhteinen kulttuuri, fokus ja sosiaalinen yhteenkuuluvuus kuin eri organisaatioiden välillä. Seitsemäs mittari oli neljännen mittarin tasolla olevien ohjelmatiedostoa muokanneiden alaisten määrä suhteessa kaikkiin ohjelmatiedostoa muokanneisiin insinööreihin. Mittarilla pyrittiin tasoittamaan isoissa tiimeissä olevien ”superinsinöörien” vaikutusta kahteen edelliseen mittariin. Näillä insinööreillä oli merkittävä tietämys ohjelmakoodikannasta, ja he kirjoittivat merkittävän määrän ohjelmakoodista. Viimeinen, kahdeksas mittari kertoi ohjelmatiedostoon yli 10 % muutoksista tehneiden organisaatioiden lukumäärän. Organisaatiotaso määräytyi samoin kuin neljännessä mittarissa.

Nagappan ja kumppanit vertasivat rakentamaansa mallia aikaisemmissa tutkimuksissa esitettyihin julkaisun jälkeen ilmenneitä vikoja ennakoiviin malleihin. Näitä olivat koodin muuttuvuuteen, koodin monimutkaisuuteen, riippuvuuksiin, testien kattavuuteen ja julkaisua ennen ilmenneisiin virheisiin pohjautuvat mallit. Eri mallit, niiden tarkkuudet ja saannit näkyvät taulukossa 3. Organisaation rakenteeseen pohjautuva malli ennakoi julkaisun jälkeisten virheiden määrän merkittävästi paremmalla tarkkuudella ja saannilla kuin muut mallit. Nagappan ja kumppanit totesivat tulosten pohjalta, että organisaationaaliset mittarit ennakoivat virhealttiuden paremmin kuin perinteiset mittarit.

Malli	Tarkkuus	Saanti
Organisaation rakenne	86,2 %	84,0 %
Koodin muuttuvuus	78,6 %	79,9 %
Koodin monimutkaisuus	79,3 %	66,0 %
Riippuvuudet	74,4 %	69,9 %
Testien kattavuus	83,8 %	54,4 %
Julkaisua ennen ilmenneet virheet	73,8 %	62,9 %

Taulukko 3: Mallin sopivuus julkaisun jälkeen ilmenneiden virheiden ennakointiin [NMB08].

4 Laatu avoimen lähdekoodin projekteissa

Tässä luvussa käsitellään ohjelmistojen laatua avoimen lähdekoodin projekteissa. Avoimen lähdekoodin projektit ovat luonteeltaan erilaisia kuin kaupalliset projektit, mutta ne poikkeavat myös keskenään toisistaan [BiN12]. FreeBSD-projekti perustuu enimmäkseen vapaaehtoisein kehittäjiin [Spi06], kun taas Firefox- ja Eclipse-projektien kehitys tapahtuu pääasiassa yrityksien kautta [BiN12]. Ohjelmistojen laatua FreeBSD-projektissa tarkastellaan luvussa 4.1. Firefox- ja Eclipse-projekteja tarkastellaan luvussa 4.2.

4.1 FreeBSD

Spinellis [Spi06] tutki kehittäjien välisen etäisyyden vaikutusta tuottavuuteen ja laatuun maantieteellisesti hajautetussa avoimen lähdekoodin FreeBSD-projektissa. Projektissa työskenteli yli 350 kehittäjää maailmanlaajuisesti. Suurin osa kehittäjistä oli vapaaehtoisia, mutta osa kehittäjistä työskenteli FreeBSD-projektia hyödyntävien yritysten kautta.

FreeBSD-projektia koordinoitiin versionhallinnan, ongelmaraporttitietokannan, sähköpostilistojen, jatkuvan integraation, julkisten manuaalien ja erilaisten kehityspalvelimien kautta. Uusille kehittäjille nimettiin ohjaaja (*mentor*), joka valvoi heidän työtään ja hyväksyi heidän tekemänsä muutokset koodiin. Spinellis sai kerättyä paikkatiedot 292 kehittäjältä, jotka asuivat 206 eri sijainnissa ympäri maailmaa. Suurin osa kehittäjistä asui Pohjois-Amerikassa tai Euroopassa, mutta kehittäjiä oli myös Aasiassa, Australiassa, Etelä-Amerikassa ja Etelä-Afrikassa.

Spinellis tutki kehittäjien välisen etäisyyden vaikutusta koodin tyyliin ja raportoitujen virheiden lukumäärän. Koodin tyylistä Spinellis tutki kuinka moni koodirivi noudattaa FreeBSD-projektin koodin tyylin ohjesääntöjä. Korrelaatiokerroin oli mittauksille hyvin alhainen, 0,05, ja täten kehittäjien välisellä maantieteellisellä etäisyydellä ei ollut vaikutuksia koodin laatuun tällä mittarilla. Spinellis tutki myös kehittäjien välisen etäisyyden vaikutusta tiedostoon liittyvien ongelmaraporttien määrään. Jälleen korrelaatiokerroin oli alhainen, 0,07, joten ei ole viitteitä siitä, että maantieteellisellä etäisyydellä olisi vaikutusta ohjelmakoodin laatuun FreeBSD-projektissa.

Spinellis tutki myös, tekevätkö kehittäjät enemmän yhteistyötä heitä lähellä kuin kaukana sijaitsevien kehittäjien kanssa. Tämän selvittääkseen Spinellis tutki saman päivän aikana tiedostoa muokanneiden kehittäjien välistä etäisyyttä. Etäisyyden kes-

kiarvo oli 6 489 kilometriä, joka oli hyvin lähellä kehittäjien keskimääräistä etäisyyttä. Täten FreeBSD-projektissa kehittäjien välisellä etäisyydellä on harvoin vaikutusta heidän väliseensä yhteistyöhön. Spinellis tutki myös ohjaajien ja ohjattavien välistä etäisyyttä ja huomasi etäisyyksien olevan alhaisempia. Spinellis päätteli, että osa ohjattavista saa ohjaajan samalta alueelta, mutta siitä huolimatta ohjaussuhteita muodostetaan myös maailmanlaajuisesti.

Tutkimustuloksia tarkasteltaessa tulisi kuitenkin Spinellisin mukaan ottaa huomioon FreeBSD-projektin erityispiirteet. FreeBSD-projektilla ei ollut perinteisiä toimistotiloja, jossa samalla paikkakunnalla olevat kehittäjät olisivat voineet tehdä yhteistyötä ja tavata kasvotusten. Lisäksi FreeBSD-projekti koostui pääasiassa vapaaehtoisista kehittäjistä, jotka olivat erittäin motivoituneita ja päteviä. Kehittäjät saattoivat käyttää projektiin juuri niin paljon aikaa kuin he halusivat. FreeBSD-projektin kehittäjät olivat myös tyypillisesti FreeBSD-käyttäjärjestelmän käyttäjiä.

4.2 Mozilla Firefox ja Eclipse

Bird ja Nagappan [BiN12] tutkivat, kuinka organisaationaalisesti ja maantieteellisesti hajautettuja avoimen lähdekoodin Mozilla Firefox- ja Eclipse-projektit ovat. He tutkivat myös, kuinka organisaationaalinen ja maantieteellinen hajautus vaikuttavat projektien ohjelmistojen laatuun tutkimalla julkaisua edeltävien sekä julkaisun jälkeisten virheiden määrää projektien komponenteissa.

Tutkimusaineistot Bird ja Nagappan keräsivät Firefox- ja Eclipse-projektien versionhallinnoista ja virhetietokannoista. He määrittivät merkittävimpien kehittäjien organisaation ja maantieteellisen sijainnin muun muassa sähköpostiosoitteiden, sosiaalisen median sivustojen, blogien, yrityksen tai suoran kommunikaation kautta. He saivat selvitettyä 77 Firefox-projektin ja 100 Eclipse-projektin merkittävimmän kehittäjän tiedot. Nämä kehittäjät tekivät kummassakin projektissa yli 95 % projektiin kohdistuneista muutoksista.

Maantieteellisen hajautuksen tasoa Bird ja Nagappan tutkivat luokittelemalla projektien komponentit neljään luokkaan sen mukaan, miltä pienimmältä mahdolliselta maantieteelliseltä alueelta tuli 75 % komponenttiin kohdistuneista muutoksista. Luokat olivat kaupunki, maa, mantere ja maailma. Organisaationaalisen hajautuksen tasoa tutkittiin laskemalla, kuinka monta eri organisaatiota osallistui komponentin kehitykseen sekä mittaamalla komponenttien organisaationaalista omistajuutta, eli kuinka suuri osa komponenttiin kohdistuneista muutoksista tuli eniten muutoksia

tehneeltä organisaatiolta.

Bird ja Nagappan tutkivat Firefox-projektin julkaisuja 1.5 ja 2.0. Firefox-projekti oli selvästi organisaationaalisesti hajautettu – projektiin osallistui useita eri organisaatioita, ja niiden panos oli merkittävä siitä huolimatta, että Mozilla Corporation oli niistä suurin. Projektiin osallistuneet vähintään yhden prosentin muutoksista tehneet organisaatiot näkyvät taulukossa 4.

Organisaatio	Muutoksien lukumäärä	Osuus muutoksista
Mozilla Corporation	9551	47,6 %
Google	2102	10,4 %
MIT	1716	8,5 %
Nokia	758	3,8 %
Intel	705	3,5 %
Netscape	631	3,1 %
IBM	334	1,7 %
XForms	277	1,4 %
Sun	209	1,0 %
Yksittäiset kehittäjät	710	3,5 %
Tuntemattomat	1525	7,6 %

Taulukko 4: Firefox-projektiin vähintään yhden prosentin osuuden muutoksista tehneet organisaatiot [BiN12].

Firefox-projekti oli myös selvästi maantieteellisesti hajautettu. Melkein puolet komponenteista oli kehitetty vähintään kahdella eri toimipaikalla ja kolmasosa useammalla mantereella. Aikaisemmista tutkimuksista [BND09] poiketen maantieteellisesti hajautetusti kehitetyt komponentit poikkesivat ominaisuuksiltaan paikallisesti kehitetyistä – ne olivat suurempia, niihin tehtiin enemmän muutoksia, ja niihin osallistui enemmän kehittäjiä.

Firefox-projektissa maantieteellisellä hajautuksella oli vain pieni vaikutus virheiden lukumäärän kasvuun. Julkaisussa 2.0 maantieteellisellä hajautuksella ei kuitenkaan ollut lainkaan vaikutusta julkaisun jälkeisten virheiden lukumäärään. Organisaationaalisesta hajautuksesta vaikutukset eivät puolestaan olleet tilastollisesti merkittäviä.

Bird ja Nagappan tutkivat Eclipse-projektin alustan ydintä ja liitännäisiä (*plugins*). Projektiin kohdistuneista muutoksista suurin osa, 91 %, tuli IBM:ltä muiden organisaatioiden osuuksien ollessa alle kaksi prosenttiyksikköä. Täten projektia ei voida pitää organisaationaalisesti hajautettuna. Myös lähes puolet muutoksista tuli koko-

naan yhdestä IBM:n toimipaikasta Ottawasta, Kanadasta ja 75 % muutoksista tuli vain kolmesta eri toimipaikasta. Vaikka projekti oli maantieteellisesti hajautettu, komponenttien kehitys oli pääasiassa paikallista.

Bird ja Nagappan havaitsivat maantieteellisen ja organisaationaalisen hajautuksen lisäävän virheiden määrää komponenteissa, mutta tämä ei kuitenkaan ollut yhdenmukaista kaikkien julkaisujen välillä. Lisäksi hajautettuja komponentteja oli vain pieni määrä, joten tulosten tilastollinen voima ei ole merkittävä.

5 Yhteenveto

Maantieteellisen hajautuksen on uskottu heikentävän ohjelmistojen laatua [KZB13]. Tutkimustulokset kuitenkin osoittavat, että kokonaan yhden organisaation sisällä tapahtuvalla maantieteellisesti hajautetulla ohjelmistokehityksellä on korkeintaan pieni heikentävä vaikutus ohjelmistojen laatuun [BND09, KZB13]. Vaikutuksen koko on niin pieni, että Kocagunelin ja kumppaneiden mukaan sitä ei kannata edes ottaa huomioon [KZB13]. Birdin ja kumppaneiden mukaan onnistuneella hallinnolla pystytäänkin vähentämään hajautuksen haitallisia vaikutuksia [BND09].

Organisaatiota on pidetty rajoittavana tekijänä ohjelmistokehityksessä [Con68] ja syyinä ohjelmistojen huonoon laatuun [Bro95]. Nagappan ja kumppanit rakensivat organisaation rakenteeseen perustuneen mallin, jolla pystyttiin merkittävästi perinteisiä mittareita paremmin ennakoimaan ohjelmatiedostojen virhealttius. Conway esittääkin [Con68], että hyvän organisaatorakenteen tulisi olla ohut ja ketterä sekä järjestetty kommunikaation tarpeen mukaan. Tämä pätee erityisesti maantieteellisesti hajautettuihin projekteihin, joissa on paljon kommunikaatioon liittyviä ongelmia [BND09].

Avoimen lähdekoodin projektien on väitetty voittavan normaalit hajautetun ohjelmistokehityksen haasteet niiden hajautettujen yhteistyömekanismien johdosta [BiN12]. Hajautuksella ei olekaan havaittu olevan tilastollisesti merkittävää vaikutusta ohjelmistojen laatuun avoimen lähdekoodin projekteissa [BiN12, Spi06].

Lähteet

- BiN12 C. Bird, N. Nagappan. Who? Where? What? Examining distributed development in two large open source projects. *MSR 2012, 9th IEEE Working Conf. on Mining Software Repositories*, 2012, sivut 237–246. [Myös: <http://dx.doi.org/10.1109/MSR.2012.6224286>].
- BND09 C. Bird, N. Nagappan, P. Devanbu, H. Gall, B. Murphy. Does distributed development affect software quality? An empirical case study of Windows Vista. *ICSE 2009, Proc. of the 31st Int. Conf. on Software Engineering*, 2009, sivut 518–528. [Myös: <http://dx.doi.org/10.1109/ICSE.2009.5070550>].
- Bro95 F.P. Brooks. *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, 1995.
- Con68 M.E. Conway. How do committees invent? *Datamation*, 14,5 (1968), sivut 28–31. [Myös: <http://www.melconway.com/research/committees.html>].
- KZB13 E. Kocaguneli, T. Zimmermann, C. Bird, N. Nagappan, T. Menzies. Distributed development considered harmful? *ICSE 2013, Proc. of the 35th Int. Conf. on Software Engineering*, 2013, sivut 882–890. [Myös: <http://dx.doi.org/10.1109/ICSE.2013.6606637>].
- NMB08 N. Nagappan, B. Murphy, V.R. Basili. The influence of organizational structure on software quality: An empirical case study. *ICSE 2008, Proc. of the 13th Int. Conf. On Software Engineering*, 2008, sivut 521–530. [Myös: <http://dx.doi.org/10.1145/1368088.1368160>].
- Spi06 D. Spinellis. Global software development in the FreeBSD project. *GSD 2006, Proc. of the 2006 Int. Workshop on Global Software Development for the Practitioner*, 2006, sivut 73–79. [Myös: <http://dx.doi.org/10.1145/1138506.1138524>].