Jari Alahuhta

Virheraportoijien virhemäärien jakaumat virhetietokannassa

Informaatio- ja luonnontieteiden tiedekunta

Kandidaatintyö

Espoo 31.8.2010

Vastuuopettaja:

Prof. Harri Ehtamo

Työn ohjaaja:

TkT Mika Mäntylä

Aalto-yliopisto
Teknillinen korkeakoulu

AALTO-YLIOPISTO
TEKNILLINEN KORKEAKOULU

KANDIDAATINTYÖN
TIIVISTELMÄ

Tekijä: Jari Alahuhta

Työn nimi: Virheraportoijien virhemäärien jakaumat virhetietokannassa

Päivämäärä: 31.8.2010 Kieli: Suomi Sivumäärä:6+33

Tutkinto-ohjelma: Teknillinen fysiikka ja matematiikka

Vastuuopettaja: Prof. Harri Ehtamo

Ohjaaja: TkT Mika Mäntylä

Työssä tutkittiin virhetietokantaan raportoitujen virheiden jakautumista yksittäisten virheraportoijien välille. Kuutta empiiristä virhejakaumaa tarkastelemalla tutkittiin aiemmassa tutkimuksessa esitettyä hypoteesia, että virhejakauma noudattaisi tyypillisesti Pareto-jakaumaa. Edelleen empiiristä aineistoa ja olemassaolevaa kirjallisuutta tarkastelemalla tutkittiin mahdollisia syitä eri virheraportoijien raportointimäärien eroille. Erityisesti selvitettiin, voisiko virheraportoijan korkeaa virheraportointimäärää jotenkin selittää matalalla virheraportointialttiudella tai sillä, että virheraportoija olisi keskittynyt testaamaan jo valmiiksi runsasvirheisiä ohjelmiston komponentteja.

Empiirisenä tutkimusaineistona työssä käytettiin kolmen suomalaisen ohjelmistoyrityksen virhetietokantoja sekä kolmen avoimen lähdekoodin ohjelmiston – Apachen HTTP-palvelimen, Linux-käyttöjärjestelmän ja Mozillan Firefox-Internet-selaimen – virhetietokantoja. Virhejakaumien yhteensopivuutta Pareto-jakauman kanssa tutkittiin Matlab- ja R-ohjelmistoilla toteutetulla työkalulla.

Erityisesti avoimen lähdekoodin ohjelmistojen virhejakaumien häntien nähtiin noudattavan Pareto-jakaumaa melko hyvin. Yritysten virhejakaumien tarkastelun perusteella ei voitu tehdä kovin luotettavia johtopäätöksiä virheraportoijien melko pienen lukumäärän takia.

Virheraportoijan virheraportointialttiudella ei havaittu olevan juuri mitään yhteyttä virheraportoijan raportointimäärään. Osassa aineistoista havaittiin viitteitä siitä, että eniten virheitä raportoineet virheraportoijat olisivat keskittyneet testauksessaan eniten virheitä sisältäneisiin komponentteihin.

Avainsanat: virheraportoija, virhetietokanta, ohjelmistojen testaaminen, avoimen lähdekoodin ohjelmisto, ohjelmistotuotanto

Esipuhe

Minulla on ollut ilo ja kunnia kirjoittaa kandidaatintyöni Ohjelmistotuotannon laboratorion SPRG-tutkimusryhmän Evidence-Based Software Quality: Practices and Assessment (ESPA) –tutkimusprojektissa kesällä 2010.

Erityisesti haluan kiittää ohjaajaani Mika Mäntylää työni ohjaamisesta sekä Timo Lehtistä yleisestä tuesta ja avusta. Haluan lisäksi kiittää professori Harri Ehtamoa työni valvomisesta.

Otaniemi, 31.8.2010

Jari Alahuhta

Sisältö

Symbolit ja lyhenteet

Symbolit

$1_A(\cdot)$	joukon A indikaattorifunktio
α	Pareto-jakauman parametri
$\hat{\alpha}$	empiiriseen jakaumaan sovitetun Pareto-jakauman parametrin estimaatti
c	ristiintaulukon sarakkeiden lukumäärä
C_j	ristiintaulukon sarakkeessa j olevien solujen arvojen summa
D	Kolmogorov-Smirnovin testisuure
E_{ij}	ristiintaulukon rivillä i sarakkeessa j olevan solun odotettu frekvenssi
$\Gamma(\cdot)$	gammafunktio
$\Gamma(\cdot,\cdot)$	epätäydellinen gammafunktio
H_0	nollahypoteesi
H_1	vaihtoehtoinen hypoteesi
n	virheraportoijien lukumäärä virhetietokannassa
O_{ij}	ristiintaulukon rivillä i sarakkeessa j oleva solu
r	Pearsonin korrelaatiokerroin
S	satunnaismuuttujan X otosavaruus
$S(\cdot)$	empiiriseen aineistoon sovitetun Pareto-jakauman kertymäfunktio
s_i	yksittäisen virheraportoijan raportoimien korjaustoimenpiteisiin johtaneiden virheiden lukumäärä
T	ristiintaulukon kaikkien solujen arvojen summa
$\varsigma(\cdot,\cdot)$	Hurwitzin zeta-funktio
v_i	yksittäisen virheraportoijan raportoimien virheiden korjausprosentti
X	virhetietokannasta satunnaisesti valitun virheraportoijan raportoimien virheiden lukumäärää kuvaava satunnaismuuttuja
X^2	testisuure, noudattaa approksimatiivisesti χ^2-jakaumaa
x_i	yksittäisen virheraportoijan raportoimien virheiden lukumäärä
x_{min}	empiirisen jakauman hännän pienin piste
y	ristiintaulukon rivien lukumäärä
Y_i	ristiintaulukon rivillä i olevien solujen arvojen summa

Lyhenteet

ACM	Association for Computing Machinery
ESPA	Evidence-Based Software Quality: Practices and Assessment
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
lkm	lukumäärä
SPRG	Software Process Research Group
UO	normalisoitu log-uskottavuusosamäärä

1 Johdanto

1.1 Työn taustaa

Tämä työ pohjautuu ESPA-tutkimusprojektin tutkijoiden Mika Mäntylän, Joonas Iivosen ja Juha Itkosen tekemään tapaustutkimukseen, jossa tutkittiin kolmen suomalaisen ohjelmistoyrityksen virheraportointia. Tutkimuksessa tehtiin kiinnostava havainto, että yritysten virhetietokantoihin raportoitujen virheiden lukumäärät jakautuivat eri virheraportoijien välille hyvin epätasaisesti: kussakin yrityksessä noin 20% virheraportoijista oli raportoinut noin 60% virheistä. Tutkimuksessa tutkittiin, mitä teoreettista jakaumaa empiirinen virhejakauma voisi noudattaa, ja joiksikin mahdollisiksi teoreettisiksi jakaumiksi esitettiin lognormaali-, gamma- ja Pareto-jakaumia. Erityisesti virhejakauman mahdollinen yhteensopivuus Pareto-jakauman kanssa nähtiin kiinnostavaksi, sillä Pareto-jakauman on havaittu esiintyvän muissakin ohjelmistotestaukseen liittyvissä ilmiöissä. [27]

Vaikka ohjelmistotestaus ja virheraportointi ovat melko paljon tutkittuja aiheita, ei virheiden jakautumista virheraportoijien välille ole aikaisemmin perusteellisesti tutkittu. Mäntylä ym. [27] olivat tietojensa mukaan ensimmäiset, jotka tutkivat virhejakaumaa teollisuuskontekstissa. Teollisuuskontekstin ulkopuolella esimerkiksi Mockus ym. [24] tarkastelivat tutkimuksessaan lyhyesti tutkimiensa avoimen lähdekoodin ohjelmistojen virheraportoijien raportointimääriä ja taustoja. Tämä työ pyrkii täyttämään pientä osaa aikaisemmassa ohjelmistotuotantokirjallisuudessa olevasta aukosta ja tutkimaan virhejakaumaa aikaisempia lähinnä heuristisia tarkasteluja tarkemmin.

1.2 Käsitteitä

Virheelle on esitetty ohjelmistotuontantokirjallisuudessa useita erilaisia määritelmiä [26]. Tässä työssä ei rajoituta mihinkään tiettyyn yksittäiseen englanninkielisen termin määritelmään, vaan määritellään virhe laveasti millaiseksi tahansa syyksi, jonka takia virhetietokantaan lähetetään virheraportti. Hyvään ohjelmistotuotantotapaan kuuluu oleellisesti, että löydetyistä virheistä sekä niille tehdyistä korjaustoimenpiteistä pidetään kirjaa. Usein virhekirjanpito toteutetaan *virhetietokannan* avulla. Virhetietokantaan lähetetään virheraportti, kun uusi virhe löytyy, ja jo olemassa olevia virheraportteja muokataan, kun virheille tehdään korjaustoimenpiteitä. Yleensä yksittäiseen virheraporttiin kirjataan hieman virheestä ja virhetietokannasta riippuen ainakin esimerkiksi virheen sanallinen kuvaus, arvio virheen vakavuudesta sekä tieto virheraportoijasta. [5, s. 122–124]

Virheraportoija määritellään tässä työssä sellaiseksi henkilöksi, joka raportoi virheestä virhetietokantaan. Selkeä ero on tehtävä virheraportoija- ja *ohjelmistotestaaja*-termien välille. Ohjelmistotestaajalla tarkoitetaan yleensä työntekijää, jonka pääasiallisena työtehtävänä on ohjelmistotestaus [8, s. 235–240], ja termiä käytetään

paljon laajassa ohjelmistotestausta käsittelevässä kirjallisuudessa. Tässä yhteydessä ohjelmistotestaaja-termiä ei kuitenkaan tule käsitellä virheraportoija-termin synonyyminä, koska virhetietokantaan raportoivat virheitä tyypillisesti muutkin kuin vain ohjelmistotestaajat [19, 27]. Ohjelmistotestaajien voidaan siten katsoa olevan ainoastaan virheraportoijien osajoukko. *Testaaminen* määritellään tässä työssä sellaiseksi toiminnaksi, jonka yhteydessä virheraportoijalla on mahdollisuus löytää virhe ohjelmistosta.

Eräs tämän työn tärkeimmistä päämääristä on tutkia, kuinka virhetietokantaan raportoitujen virheiden lukumäärät jakautuvat eri virheraportoijien välille. Ellei erikseen toisin mainita, tarkoitetaan *virhejakauma*-termillä tämän työn kontekstissa juuri virheiden jakautumista eri virheraportoijien välille.

Komponentti määritellään tämän työn kontekstissa yksinkertaisesti sellaiseksi itsenäiseksi osakokonaisuudeksi, jollaisista yksittäinen ohjelmisto koostuu. Ohjelmistotuotannon standardisanaston IEEE 610.12 [17, s. 18] mukaan termejä ”moduuli” (engl. module), ”yksikkö” (engl. unit) ja ”komponentti” (engl. component) käytetään usein toistensa synonyymeinä.

1.3 Työn tavoitteet

Tämän työn tavoitteena on tutkia Mäntylän ym. [27] tekemää hypoteesia siitä, että yrityksen tai laajan ohjelmiston virhejakauma noudattaisi tyypillisesti Paretojakaumaa. Lisäksi tavoitteena on löytää syitä virheraportoijien välisten raportointimäärien eroille. Työn tavoitteita pyritään lähestymään olemassaolevaa kirjallisuutta tarkastelemalla ja empiiristä aineistoa tutkimalla. Empiirisenä tutkimusaineistona käytetään Mäntylän ym. [27] keräämää dataa kolmen suomalaisen ohjelmistoyrityksen virhetietokannoista sekä tämän lisäksi dataa kolmen avoimen lähdekoodin ohjelmiston – Apachen HTTP-palvelimen, Linux-käyttöjärjestelmän ja Mozillan Firefox-Internet-selaimen – virhetietokannoista.

1.4 Työn rakenne

Tämän työn rakenne on seuraavanlainen: luvussa 2 esitellään tässä työssä käytettäviä tutkimusmenetelmiä, aineiston keräämistä ja aineiston analysointia, ja luvussa 3 tarkastellaan lyhyesti tämän työn aihepiiriin liittyvää aikaisempaa tutkimusta. Luvussa 4 sovelletaan luvussa 2 esiteltyjä menetelmiä tutkimusaineistoon, minkä jälkeen luvussa 5 annetaan vastaukset tutkimuskysymyksiin, tarkastellaan saatujen tuloksien validiteettia sekä esitetään ajatuksia tulevasta tutkimuksesta.

2 Tutkimusmenetelmät

2.1 Tutkimuskysymykset

Luvussa 1.3 esitettyjä tavoitteita lähestytään tässä työssä erityisesti kahden tutkimuskysymyksen avulla.

Tutkimuskysymys 1: Noudattavatko virheraportoijien virhetietokantaan raportoimien virheiden jakaumat Pareto-jakaumaa tai jotain muuta tunnettua teoreettista jakaumaa tarkasteltavissa yrityksissä tai avoimen lähdekoodin ohjelmistoissa?
Tässä tutkimuskysymyksessä tavoitteena on selvittää, voidaanko tarkasteltavien empiiristen virhejakaumien katsoa noudattavan Pareto-jakaumaa. Empiiristen jakaumien ja Pareto-jakauman yhteensopivuuden lisäksi tutkitaan myös, voisiko jonkin muun teoreettisen jakauman katsoa sopivan empiiriseen jakaumaan Pareto-jakaumaa paremmin. Koska tunnettuja teoreettisia jakaumia on lukuisa määrä, ei kaikkia mahdollisia tunnettuja jakaumia ole käytännössä mahdollista tutkia – tässä työssä käytetyllä työkalulla Pareto-jakaumaa pystytään vertaamaan eksponenttijakaumaan, lognormaalijakaumaan, Poisson-jakaumaan, Weibull-jakaumaan, Yulen jakaumaan sekä erääseen gammajakaumaa muistuttavaan jakaumaan.

Empiiristen virhejakaumien ja Pareto-jakauman yhteensopivuuden selvittäminen on kiinnostavaa monesta syystä. Puhtaan teoreettisesti ilmiön tutkiminen on kiinnostavaa esimerkiksi siksi, että viimeaikaisissa tutkimuksissa Pareto-jakauman on havaittu esiintyvän lukuisissa tietojenkäsittelytieteen ilmiöissä: muun muassa ohjelmistojen komponenttien koot koodirivien lukumäärällä mitattuna, eri Internet-sivujen latauskerrat sekä yksittäisten Internetin käyttäjien käyntikerrat eri Internet-sivuilla noudattavat Pareto-jakaumaa [6, 16]. Esimerkiksi Mitzenmacherin [23] mukaan "Pareto-jakaumia havaitaan parhaillaan kaikkialla tietojenkäsittelytieteessä". Onkin mielenkiintoista selvittää, onko tässä työssä tarkasteltava ilmiö jälleen eräs Pareto-jakaumaa noudattava tietojenkäsittelytieteen ilmiö. Puhtaan teoreettisen arvon lisäksi tällä tutkimuskysymyksellä on luonnollisesti myös käytännön arvoa muun tutkimuksen kannalta – tarkasteleehan tutkimuskysymys suoraan aiemmassa tutkimuksessa [27] esitettyä hypoteesia.

Tutkimuskysymys 2: Mistä erot raportointimäärissä yksittäisten virheraportoijien välillä johtuvat?
Tähän tutkimuskysymykseen ei niinkään pyritä hakemaan yksityiskohtaista selvitystä, vaan ennemminkin tavoitteena on identifioida empiiristä aineistoa ja olemassaolevaa kirjallisuutta tutkimalla mahdollisia syitä raportointimäärien eroille. Empiirisen aineiston avulla on tarkoitus erityisesti etsiä vastauksia seuraviin kahteen osakysymykseen.
Osakysymys 1: Onko virheraportoijien raportoimien virheiden lukumäärillä ja korjausprosenteilla jonkinlaista yhteyttä tarkasteltavissa yrityksissä tai avoimen lähdekoodin ohjelmistoissa?
Tämän osakysymyksen ajatuksena on tutkia virheraportoijien raportointialttiuden

vaikutusta raportointimäärään: voiko suurta virheraportointimäärä selittää sillä, että raportoija raportoi pienimmätkin havaitsemansa virheet? Virheen korjausprosenttia käytetään siis ilmaisemaan virheen eräänlaista tärkeyttä. Usein virheraporteissa ilmoitettavaa virheen kiireellisyyttä (engl. priority) tai vakavuutta (engl. severity) ei tässä yhteydessä hyödynnetä, sillä tarkasteltavissa virhetietokannoissa virheraportoija saa oman subjektiivisen mielipiteensä mukaan itse päättää raportoimansa virheen kiireellisyyden ja vakavuuden.

Osakysymys 2: Keskittyvätkö runsaasti virheitä raportoivat virheraportoijat tyypillisesti raportoimaan virheitä niistä komponenteista, joista ylipäätään on raportoitu paljon virheitä?

Tämän osakysymyksen ajatuksena on tutkia, voiko virheraportoijan suurta virheraportointimäärää perustella sillä, että virheraportoija olisi keskittynyt testaamaan sellaisia komponentteja, joissa on ylipäätään paljon virheitä. Tätä kysymystä ei pystytä tarkastelemaan käytetyllä yritysaineistolla, koska tarkasteltavien yritysten virhetietokannoissa monien virheiden sijainteja ei ole mitenkään spesifioitu. Avoimen lähdekoodin ohjelmistojen komponenttitason rakenteet jäsennetään samoin kuin kunkin ohjelmiston virhetietokannan yhteydessä.

2.2 Aineiston kerääminen

Tutkimusaineistona käytetään kolmen avoimen lähdekoodin ohjelmiston – Apachen HTTP-palvelimen, Linux-käyttöjärjestelmän ja Mozillan Firefox-Internet-selaimen – sekä kolmen suomalaisen ohjelmistoyrityksen virhetietokantojen dataa. Seuraavaksi selostetaan, kuinka tutkimusaineisto tässä työssä kerättiin.

2.2.1 Avoimen lähdekoodin ohjelmistojen aineiston kerääminen

Tarkasteltavien avoimen lähdekoodin ohjelmistojen virhetietokannat on toteutettu Mozillan Bugzilla-ohjelmistolla. Bugzillaa voi yleisesti käyttää esimerkiksi virheiden ja koodimuutosten kirjanpitoon, kehittäjien ja testaajien väliseen kommunikointiin, koodimuutosten palauttamiseen ja arviointiin sekä laadunvarmistuksen hallintaan [7]. Tässä työssä hyödynnetään erityisesti Bugzillalla toteutettuja avoimen lähdekoodin ohjelmistojen virhekirjanpitohistorioita.

Bugzillaa ylläpidetään yleensä palvelinkoneella, johon käyttäjä ottaa Internet-selaimella yhteyden [7]. Käyttäjän näkökulmasta Bugzilla on tavanomainen Internet-sivu. Halutessaan käyttäjä voi avata itselleen tilin, jolloin Bugzillaan pystyy kirjautumaan sisään. Ilman sisäänkirjautumistakin voi lukea olemassa olevia virheraportteja sekä näiden kommentteja; lisäksi virheraportteja, niin avoimia kuin suljettujakin, voi hakea monin eri parametrein selausikkunassa. Kirjautumalla sisään käyttäjä pääsee syöttämään itse virheraportteja sekä muokkaamaan ja kommentoimaan jo olemassa olevia virheraportteja. Tilin avanneet käyttäjät eli virheraportoijat ja virheraporttien kommentoijat identifioidaan sähköpostiosoitteen perusteella.

Bugzillalla toteutetun virhetietokannan virheraportilla on tyypillisesti useita parametreja: esimerkiksi virheen tila (engl. status), virheelle tehdyt toimenpiteet (engl. resolution), virheen kiireellisyys, virheen vakavuus ja virheen komponenttitason sijainti kirjataan yleensä yksittäisen virheen virheraporttiin. Kun virheelle tehdään esimerkiksi korjaustoimenpiteitä, niin virheraportin parametreja päivitetään vastaavasti.

Tässä työssä käytettävä avoimen lähdekoodin ohjelmistojen aineisto haettiin Apachen [3], Linuxin [22] ja Mozillan [25] Bugzillalla toteutetuista virhetietokannoista. Tutkimusaineistoksi valittiin kullakin ohjelmistolla ajanjaksolla 1.1.2000–1.1.2010 virhetietokantaan raportoidut virheraportit, jotka oli 1.7.2010 mennessä kirjattu suljetuiksi. Pelkästään suljettuihin virheraportteihin rajoituttiin siksi, että sulkemattoman virheraportin virheelle voidaan vielä esimerkiksi tehdä korjaustoimenpiteitä, kun taas suljetun virheraportin virheen mahdolliset korjaustoimenpiteet on jo tehty ja kirjattu virheraporttiin. Tietoa virheiden mahdollisista korjaustoimenpiteistä tarvittiin tutkimuskysymyksen 2 osakysymyksen 1 tarkastelussa.

2.2.2 Yritysaineiston kerääminen

Tarkasteltavat kaupalliset ohjelmistoyritykset ovat samoja, joita Mäntylä ym. [27] tutkivat tapaustutkimuksessaan, ja tässä työssä voitiinkin hyödyntää suoraan heidän keräämäänsä dataa. Tutkittavat kolme yritystä ovat keskikokoisia, hyvin tuottavia, kasvavia ja käytännössä velattomia suomalaisia ohjelmistoyrityksiä. Yritysten anonymiteetin säilyttämiseksi yritysten nimiä ei tässä yhteydessä kerrota, vaan niihin viitataan vain yrityksinä A, B ja C. Taulukossa (1) esitetään tiivistetysti perustietoja yrityksistä ja näiden tarkasteltavista tuotteista.

Tutkimusaineistoksi rajattiin tässä työssä kunkin yrityksen virhetietokantaan vuonna 2008 syötetyt virheet, jotka datan keräyshetkellä oli kirjattu suljetuiksi. Pelkästään suljettuihin virheraportteihin rajoituttiin samasta syystä kuin avoimen lähdekoodin ohjelmistoissa: sulkemattoman virheraportin virheelle voidaan vielä esimerkiksi tehdä korjaustoimenpiteitä, kun taas suljetun virheraportin virheen mahdolliset korjaustoimenpiteet on jo tehty ja kirjattu virheraporttiin. Yritys A uusi virhetietokantansa vuoden 2008 aikana, ja tämän takia yrityksen A virhetietokannasta saatiin dataa vain kuuden kuukauden ajalta. Yrityksien B ja C virhetietokannoista dataa saatiin sen sijaan käyttöön koko vuodelta 2008. Kussakin yrityksessä yksittäiseen virheraporttiin oli sisällytetty tyypillisesti ainakin esimerkiksi virheraportoijan nimi, raportointiaika ja mahdollinen virheenkorjausaika. Ohjelmistoyritysten virhetietokannoissa ei avoimen lähdekoodin ohjelmistojen virhetietokannoista poiketen monien virheiden esiintymispaikkaa ollut mitenkään spesifioitu.

Taulukko 1: Perustietoja yrityksistä A–C ja näiden tarkasteltavista tuotteista. [27]

	Yritys A	**Yritys B**	**Yritys C**
Työntekijöitä	>110	>60 tarkasteltavalla osastolla (>300 koko yrityksessä)	>70 tarkasteltavalla osastolla (>100 koko yrityksessä)
Asiakkaita	>200	>80 tarkasteltaville tuotteille	>300
Ikä	>10 vuotta	>20 vuotta	>20 vuotta
Tarkasteltu tuote	-yksittäinen tuote -erään teollisuuden-alan ammattiohjelmisto -integroidaan suoraan asiakkaan liiketoimin-tajärjestelmiin	-kaksi tuotetta eri tekniikan aloille -tuotteilla yhteinen perusrakenne -integroidaan suoraan asiakkaan liiketoimin-tajärjestelmiin	-yksittäinen tuote tek-niseen suunnitteluun -eräällä toisella yrityk-sen tuotteista sama ydinrakenne -ei vaadi juurikaan integ-rointia eikä kustomointia
Julkaisu-prosessi	-sisäinen suurempi jul-kaisu kerran kuukaudessa -suurin osa ohjelmisto-kehityksestä tehdään asiakkaan tiloissa	-ulkoinen suurempi julkaisu kaksi ker-taa vuodessa -ulkoinen pienempi julkaisu neljä ker-taa vuodessa -suurin osa ohjelmis-tokehityksestä tehdään omassa toimistossa	-ulkoinen suurempi julkaisu kerran vuodessa -ulkoinen pienempi julkaisu kerran vuo-dessa -suurin osa ohjelmis-tokehityksestä tehdään omassa toimistossa

2.3 Aineiston analysointi

2.3.1 Empiirisen jakauman yhteensopivuus Pareto-jakauman tai muun teoreettisen jakauman kanssa

Luvussa 2.1 tutkimuskysymykseksi 1 muotoiltiin, voidaanko empiiristen virhejakaumien katsoa noudattavan Pareto-jakaumaa tai jotain muuta teoreettista jakaumaa. Jakauman tutkimisessa käytettiin Clausetin ym. [9] kuvaamaa menetelmää, johon liittyy myös vapaasti Internetistä saatavilla oleva, tässäkin työssä hyödynnetty työkalu [18]. Työkalu on toteutettu Matlab- ja R-ohjelmistoilla, ja sillä voidaan käytännössä toteuttaa kaikki Clausetin ym. [9] kuvaaman menetelmän ominaisuudet. Seuraavaksi esitellään menetelmää, työkalua ja menetelmän taustalla olevaa teoriaa siinä laajudessa, kuin tämän työn ymmärtämiseksi on tarpeellista. Käytetyn menetelmän vaiheet esitetään tiivistetysti taulukossa (2).

Pareto-jakauma on eräs potenssijakaumiin (engl. power law distribution) kuuluva todennäköisyysjakauma. Jakauma esitellään kirjallisuudessa usein jatkuvana jakau-

Taulukko 2: Empiirisen jakauman analysoinnissa käytetyn menetelmän vaiheet. [9]

1.	Sovitetaan empiiriseen virhejakaumaan Pareto-jakauma suurimman uskottavuuden menetelmällä.
2.	Määritetään empiirisen virhejakauman ja sovitetun Pareto-jakauman yhteensopivuuden aste. Yhteensopivuustestin perusteella tulkitaan, voidaanko empiirisen virhejakauman katsoa noudattavan Pareto-jakaumaa.
3.	Tutkitaan, sopiiko jokin muista tarkasteltavista teoreettisista jakaumista empiiriseen virhejakaumaan Pareto-jakaumaa paremmin.

mana, mutta tässä työssä tarkasteltiin tutkitun ilmiön luonteen takia ainoastaan Pareto-jakauman diskreettiä versiota, josta käytetään joissain lähteissä Pareto-jakauman ohella myös nimeä Zipfin laki tai zeta-jakauma. [14, 15, 29] Diskreetti satunnaismuuttuja X noudattaa Pareto-jakaumaa parametrilla α, mikäli sen pistetodennäköisyysfunktio on muotoa

$$Pr(X = k) = \frac{k^{-\alpha}}{\sum_{i=1}^{\infty} i^{-\alpha}}, \qquad k \in \mathbb{Z}_+, \tag{1}$$

missä yleensä $2 < \alpha < 3$ [9, 29]. Esimerkiksi Arnoldin [4] teoksessa käsitellään paljon lisää erilaisia Pareto-jakauman ominaisuuksia.

Tarkastellaan yksittäisen yrityksen tai avoimen lähdekoodin ohjelmiston virhetietokantaa. Määritellään satunnaismuuttuja X virhetietokannasta satunnaisesti valitun virheraportoijan raportoimien virheiden lukumääräksi. Olkoon lisäksi n virheraportoijien lukumäärä virhetietokannassa ja x_i, missä $i \in \{1, \ldots, n\}$, yksittäisen virhetietokannan virheraportoijan raportoimien virheiden lukumäärä. Satunnaismuuttujan X otosavaruus S on siten

$$S = \{x_1, \ldots, x_n\}. \tag{2}$$

Satunnaismuuttujan X otosavaruuden S määritelmästä (2) nähdään suoraan, että satunnaismuuttuja X voi saada arvokseen pelkästään positiivisia kokonaislukuja, sillä luonnollisesti yksittäisen virheraportoijan raportoimien virheiden lukumäärä x_i on positiivinen kokonaisluku. Satunnaismuuttujan X pistetodennäköisyysfunktio saadaan siten klassisena todennäköisyytenä, se on muotoa

$$Pr(X = k) = \frac{\sum_{i=1}^{n} 1_{\{k\}}(x_i)}{n}, \qquad k \in \mathbb{Z}_+, \tag{3}$$

missä $1_{\{k\}}$ on joukon $\{k\}$ indikaattorifunktio – termillä $\sum_{i=1}^{n} 1_{\{k\}}(x_i)$ tarkoitetaan niiden virheraportoijien lukumäärää, jotka raportoivat virhetietokantaan tasan k virhettä. Yhtälön (3) perusteella satunnaismuuttujan X kertymäfunktio on

$$Pr(X \leq k) = \frac{\sum_{i=1}^{n} 1_{\{1,\ldots,k\}}(x_i)}{n}, \qquad k \in \mathbb{Z}_+, \tag{4}$$

missä termi $\sum_{i=1}^{n} 1_{\{1,\ldots,k\}}(x_i)$ merkitsee niiden virheraportoijien lukumäärää, jotka raportoivat virhetietokantaan korkeintaan k virhettä.

Yhtälössä (3) esitettyyn satunnaismuuttujan X empiiriseen jakaumaan voidaan sovittaa yhtälössä (1) esitetty Pareto-jakauma usealla eri menetelmällä [9]. Tässä työssä käytetty sovitusmenetelmä on suurimman uskottavuuden menetelmä (engl. maximum likelihood method), jonka avulla voidaan laskea teoreettisten jakaumien parametreille yleensä varsin toimivia ja teoreettisilta ominaisuuksiltaankin hyviä estimaattoreita [21]. Yhtälön (1) Pareto-jakauman parametrin α suurimman uskottavuuden estimaatin $\hat{\alpha}$ lauseketta ei voida määritellä suljetussa muodossa, mutta se voidaan ratkaista numeerisesti yhtälöstä

$$\frac{\varsigma'(\hat{\alpha},1)}{\varsigma(\hat{\alpha},1)} = -\frac{1}{n}\sum_{i=1}^{n}\ln x_i, \tag{5}$$

missä $\varsigma(\hat{\alpha},k) = \sum_{i=0}^{\infty}(i+k)^{-\hat{\alpha}}$ on Hurwitzin zeta-funktio. Estimaatin $\hat{\alpha}$ keskivirhe $\hat{\sigma}$ noudattaa approksimatiivisesti yhtälöä

$$\hat{\sigma} = \frac{1}{\sqrt{n\left[\frac{\varsigma''(\hat{\alpha},1)}{\varsigma(\hat{\alpha},1)} - \left(\frac{\varsigma'(\hat{\alpha},1)}{\varsigma(\hat{\alpha},1)}\right)^2\right]}}. \tag{6}$$

Hurwitzin zeta-funktion derivointi yhtälöissä (5) ja (6) viittaa derivointiin funktion ensimmäisen argumentin suhteen. [9, 15, 33]

Kun Pareto-jakauman parametrille α on laskettu suurimman uskottavuuden estimaatti, voidaan siirtyä tarkastelemaan Pareto-jakauman ja empiirisen virhejakauman yhteensopivuuden astetta (engl. goodness of fit). Empiirinen virhejakauma ja tähän sovitettu Pareto-jakauma eivät ole tilastollisesti riippumattomia, koska Pareto-jakauman parametrin määrityksessä hyödynnettiin suoraan empiiristä dataa. Tämän vuoksi jakaumien yhteensopivuutta ei voida suoraan tutkia millään tavallisesti käytetyllä jakaumien yhteensopivuustestillä, kuten χ^2-testillä tai Kolmogorov-Smirnovin testillä, koska jakaumien välinen riippumattomuus kuuluu näiden testien oletuksiin [15].

Clauset ym. [9] esittelevät Monte Carlo –simulointia hyödyntävän tavan testata empiirisen jakauman ja tähän sovitetun Pareto-jakauman yhteensopivuuden astetta. Hypoteeseiksi asetetaan

$$\begin{aligned} H_0 &: \text{empiirinen jakauma ei eroa Pareto-jakaumasta} \\ H_1 &: \text{empiirinen jakauma eroaa Pareto-jakaumasta,} \end{aligned}$$

missä luonnollisesti H_0 on nollahypoteesi ja H_1 vaihtoehtoinen hypoteesi. Empiiriseen aineistoon sovitetusta Pareto-jakaumasta generoidaan aluksi suuri määrä n:n kappaleen satunnaislukujoukkoja. Yhteensopivuustestin p-arvoksi määritellään se osuus satunnaislukujoukoista, joiden etäisyys sovitetusta Pareto-jakaumasta on suurempi kuin empiirisen jakauman etäisyys sovitetusta Pareto-jakaumasta. Pienet p-arvot viittaavat siten luonnollisesti siihen, että H_0 tulisi hylätä ja empiirisen jakauman voitaisiin katsoa eroavan Pareto-jakaumasta; testin riskitasoksi valitaan 0.10.

Jakaumien välistä etäisyyttä mitataan Kolmogorov-Smirnovin testisuureella D, joka määritellään yksinkertaisesti Pareto-jakauman kertymäfunktion ja tähän vertailtavan jakauman kertymäfunktion välisenä maksimietäisyytenä

$$D = \max_{k \in \mathbb{Z}_+} |Pr(X \leq k) - S(k)|, \tag{7}$$

missä $S(k)$ merkitsee Pareto-jakauman kertymäfunktiota ja $Pr(X \leq k)$ tähän vertailtavan jakauman kertymäfunktiota.

Vaikka empiirinen jakauma ei testin mukaan eroaisikaan Pareto-jakaumasta, on täysin mahdollista, että jokin toinen teoreettinen jakauma sopisi empiiriseen jakaumaan yhtä hyvin tai jopa paremmin kuin Pareto-jakauma – tämä mahdollisuus pitää selvittää. Tässä työssä käytetyllä työkalulla [18] Pareto-jakaumaa pystytään vertaamaan eksponenttijakaumaan, lognormaalijakaumaan, Poisson-jakaumaan, Weibull-jakaumaan, Yulen jakaumaan sekä erääseen gammajakaumaa muistuttavaan jakaumaan, jonka Clauset ym. [9] nimeävät Pareto-eksponenttijakaumaksi (engl. power law plus exponential distribution tai power law distribution with cutoff). Koska satunnaismuuttuja X on diskreetti ja osa jakaumista on jatkuvia, pitää jatkuvat teoreettiset jakaumat saattaa diskreettiin muotoon ennen sovittamista empiiriseen jakaumaan. Huomattavaa on, että jatkuvat jakaumat voidaan tunnetusti yleensä diskretoida tai tulkita diskreetissä muodossa useammalla kuin yhdellä tavalla. Tässä työssä on valittu samat esitykset kuin Clausetin ym. [9] artikkelissa ja käytetyssä työkalussa [18].

Eksponenttijakauma: Satunnaismuuttujan X diskreettiyden takia tässä yhteydessä ei voida käyttää eksponenttijakauman tavallista jatkuvaa muotoa. Eksponenttijakauma määritelläänkin seuraavasti: satunnaismuuttuja X noudattaa eksponenttijakaumaa parametrilla $\lambda > 0$, mikäli sen pistetodennäköisyysfunktio on muotoa

$$Pr(X = k) = (e^{\lambda} - 1)e^{-\lambda k}, \qquad k \in \mathbb{Z}_+. \tag{8}$$

Lognormaalijakauma: Jatkuva satunnaismuuttuja Y noudattaa lognormaalijakaumaa, jos sen luonnollinen logaritmi $\ln Y$ noudattaa normaalijakaumaa. Lognormaalijakaumaa noudattavan satunnaismuuttujan Y tiheysfunktio f on muotoa

$$f(y) = \frac{1}{y\sqrt{2\pi\sigma^2}} e^{-\frac{(\ln y - \mu)^2}{2\sigma^2}}, \qquad y \in \mathbb{R}_+, \tag{9}$$

missä $\sigma^2 > 0$ ja μ ovat jakauman parametrit. Käytetty työkalu [18] diskretoi jatkuvan jakauman numeerisesti Clausetin ym. [9] esittämällä menetelmällä.

Poisson-jakauma: Satunnaismuuttuja X noudattaa Poisson-jakaumaa parametrilla $\lambda > 0$, mikäli sen pistetodennäköisyysfunktio on muotoa

$$Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \qquad k \in \mathbb{Z}_+. \tag{10}$$

Weibull-jakauma: Jatkuva satunnaismuuttuja Y noudattaa Weibull-jakaumaa, jos sen tiheysfunktio f on muotoa

$$f(y) = \frac{k}{\lambda}\left(\frac{y}{\lambda}\right)^{k-1} e^{-(y/\lambda)^k}, \qquad y \in \mathbb{R}_+, \tag{11}$$

missä $k > 0$ ja $\lambda > 0$ ovat jakauman parametrit. Työkalu [18] diskretoi Weibull-jakauman Nakagawan ja Osakin menetelmällä [28].

Yulen jakauma: Satunnaismuuttuja X noudattaa Yulen jakaumaa parametrilla $\alpha > 1$, jos sen pistetodennäköisyysfunktio on muotoa

$$Pr(X = k) = (\alpha - 1)\frac{\Gamma(k)\Gamma(\alpha)}{\Gamma(k + \alpha)}, \qquad k \in \mathbb{Z}_+, \tag{12}$$

missä gammafunktio $\Gamma(s) = \int_0^\infty t^{s-1}e^{-t}dt$ on määritelty kaikilla s:n arvoilla paitsi ei-positiivisilla kokonaisluvuilla.

Pareto-eksponenttijakauma: Jatkuva satunnaismuuttuja Y noudattaa Pareto-eksponenttijakaumaa, jos sen tiheysfunktio f on muotoa

$$f(y) = \frac{\lambda^{1-\alpha}}{\Gamma(1 - \alpha, \lambda)} y^{-\alpha} e^{-\lambda y}, \qquad y \in \mathbb{R}_+, \tag{13}$$

missä $\alpha > 0$ ja $\lambda > 0$ ovat jakauman parametrit ja epätäydellinen gammafunktio $\Gamma(s, x) = \int_x^\infty t^{s-1}e^{-t}dt$ on määritelty kaikilla x:n arvoilla ja kaikilla s:n arvoilla paitsi ei-positiivisilla kokonaisluvuilla. Käytetty työkalu [18] diskretoi jakauman Clausetin ym. [9] esittämällä menetelmällä ja laskee parametrien α ja λ arvot numeerisesti. Huomattavaa on, että Pareto-eksponenttijakauma on muodoltaan melko samantyyppinen gammajakauman kanssa [20]. Lisäksi huomattavaa on, että sopivasti parametrit valitsemalla Pareto-eksponenttijakaumasta pystytään muodostamaan niin eksponenttijakauma kuin Pareto-jakaumakin.

Clausetin ym. [9] menetelmässä Pareto-jakaumaa vertaillaan vuorotellen erikseen kuhunkin muuhun teoreettiseen jakaumaan: käytetty menetelmä on uskottavuusosamäärätesti, jonka avulla voidaan päätellä, sopiiko tarkasteltava teoreettinen jakauma valitulla riskitasolla empiiriseen aineistoon Pareto-jakaumaa paremmin tai huonommin.

Uskottavuusosamäärätestissä testisuureena on normalisoitu log-uskottavuusosamäärä (engl. normalized log-likelihood ratio). Testisuureen positiiviset arvot viittaavat siihen, että Pareto-jakauma sopii empiiriseen aineistoon vertailujakaumaa paremmin, ja negatiiviset arvot vastaavasti siihen, että vertailujakaumaa sopii empiiriseen aineistoon Pareto-jakaumaa paremmin. Päätös siitä, onko testisuureen arvo tilastollisesti merkitsevästi positiivinen tai negatiivinen, perustuu Vuongin [34] esittelemään menetelmään: uskottavuusosamäärän ja tämän hajonnan avulla lasketaan p-arvo, jonka perusteella tehdään päätös, voidaanko toisen jakaumista katsoa sopivan empiiriseen jakaumaan toista paremmin. Pienet p-arvot viittaavat tilastollisesti merkitsevään eroavuuteen, ja tässä yhteydessä käytetään riskitasoa 0.10. Koska

Pareto-eksponenttijakaumasta pystytään sopivasti parametrit valitsemalla muodostamaan niin eksponenttijakauma kuin Pareto-jakaumakin, sopii Pareto-eksponenttijakauma empiiriseen aineistoon aina vähintään yhtä hyvin kuin Pareto-jakauma; normalisoitu log-uskottavuusosamäärä ei voi tällöin olla positiivinen. Tämän vuoksi Pareto-eksponenttijakaumaa tarkasteltaessa p-arvo lasketaan hieman muihin jakaumiin nähden eri tavalla. [9, 34]

Toisinaan empiiristä jakaumaa tutkittaessa ollaan kiinnostuneita erityisesti jakauman hännän käyttäytymisestä: kuinka raskas jakauman häntä on, eli kuinka harvinaisia todella suuret tai – tarkasteltavasta ilmiöstä riippuen – todella pienet havainnot ovat? Tässä työssä käytetyllä työkalulla [18] pystytäänkin tutkimaan myös pelkästään empiirisen jakauman häntää. Suurimman uskottavuuden menetelmällä pystytään laskemaan, mistä pisteestä x_{min} empiirinen jakauma tulisi katkaista, jotta katkaistun empiirisen jakauman häntä sopisi parhaiten Pareto-jakaumaan. Piste x_{min} siis viittaa hännän pienimpään pisteeseen, joka voidaan tässä yhteydessä tulkita pienimmäksi yksittäisen virheraportoijan raportointimääräksi virhejakauman hännässä. Hännän yhteensopivuutta Pareto-jakauman tai muiden teoreettisten jakaumien kanssa voidaan tutkia tässä luvussa esitellyllä menetelmällä. Huomioitava on, että mikäli katkaisupiste on suurempi kuin 1, joudutaan katkaisupiste x_{min} ottamaan huomioon joidenkin teoreettisten jakaumien lausekkeissa [9].

Tässä työssä Pareto-jakauma halutaan ensisijaisesti sovittaa koko empiiriseen virhejakaumaan, mutta toki kiinnostavaa on myös tarkastella, noudattaisiko pelkästään jakauman häntä Pareto-jakaumaa – varsinkin siinä tapauksessa, että jakauman ei voi kokonaisuudessaan katsoa noudattavan Pareto-jakaumaa. Toisaalta pelkkää empiirisen virhejakauman häntää ei tässä työssä rajoituta tutkimaan, sillä mikäli katkaisupiste x_{min} on suuri, saattaa havaintojen lukumäärä hännässä jäädä hyvin pieneksi, mikä taas ei ole tämän tutkimuskysymyksen tavoitteen kannalta tarkoituksenmukaista.

2.3.2 Virheraportoijan raportoimien virheiden lukumäärän ja korjausprosentin yhteys

Luvussa 2.1 tutkimuskysymyksen 2 ensimmäiseksi osakysymykseksi asetettiin tutkia, onko virheraportoijan raportoimien virheiden lukumäärällä ja korjausprosentilla jonkinlaista yhteyttä käytetyssä empiirisessä tutkimusaineistossa. Yksittäisen virheraportoijan raportoimien virheiden korjausprosentti v_i, missä $i \in \{1, \ldots, n\}$, voidaan määritellä yhtälöllä

$$v_i = \frac{s_i}{x_i}, \qquad i \in \{1, \ldots, n\}, \tag{14}$$

missä s_i on virheraportoijan raportoimien korjaustoimenpiteisiin johtaneiden virheiden lukumäärä. Yksittäisen virheraportoijan raportoimien virheiden lukumäärän ja korjausprosentin yhteyttä tutkitaan tässä työssä Pearsonin korrelaatiokertoimella r,

joka määritellään yhtälöllä

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(v_i - \overline{v})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(v_i - \overline{v})^2}}, \tag{15}$$

missä termi $\overline{v}$ tarkoittaa virheiden korjausprosenttien aritmeettista keskiarvoa ja termi $\overline{x}$ virheraportoijien löytämien virheiden aritmeettista keskiarvoa. Yhtälöllä (15) määritelty korrelaatiokerroin r kuvaa virheraportoijien löytämien virheiden lukumäärän ja korjausprosentin välisen lineaarisen tilastollisen riippuvuuden voimakkuutta ja saa arvonsa väliltä [-1,1]. Korrelaatiokertoimen arvo -1 tarkoittaa, että muuttujien välillä on havaintoaineistossa täydellinen negatiivinen lineaarinen riippuvuus, arvo 1 tarkoittaa, että muuttujien välillä on havaintoaineistossa täydellinen positiivinen lineaarinen riippuvuus, ja arvo 0 tarkoittaa, että muuttujien välillä ei ole havaintoaineistossa lainkaan lineaarista riippuvuutta. [32]

Virheraportoijien raportointimäärien ja virheiden korjausprosenttien lineaarisen riippuvuuden olemassaoloa voidaan testata tilastollisesti. Muodostetaan hypoteesit

$$\begin{aligned} H_0 &: r = 0 \\ H_1 &: r \neq 0, \end{aligned}$$

missä luonnollisesti H_0 on nollahypoteesi ja H_1 vaihtoehtoinen hypoteesi. Testisuure

$$t = \sqrt{n-2}\frac{r}{\sqrt{1-r^2}} \tag{16}$$

noudattaa nollahypoteesin pätiessä Studentin t-jakaumaa vapausastein (n–2), ja itseisarvoltaan suuret testisuureen t arvot viittaavat siihen, että nollahypoteesi H_0 ei päde [10, s. 360–363]. Testin p-arvo voidaan laskea helposti esimerkiksi SPSS-ohjelmistolla. Mikäli testin p-arvo on valittua riskitasoa 0.05 pienempi, niin nollahypoteesi H_0 hylätään, ja vaihtoehtoinen hypoteesi H_1 astuu voimaan.

Virheraportoijan raportoimien virheiden lukumäärän ja korjausprosentin yhteyden tutkimisen taustalla oli hypoteesi, että yksittäisen virheraportoijan suuri raportointimäärä johtuisi tyypillisesti siitä, että raportoija raportoisi virhetietokantaan pienimmätkin havaitsemansa virheet. Tällaisessa tapauksessa raportointimäärän ja korjausprosentin välinen korrelaatio olisi odotettavasti negatiivinen. On kuitenkin huomattava, että raportointimäärän ja korjausprosentin välinen negatiivinen korrelaatio ei välttämättä tarkoita, että taustalla oleva hypoteesi olisi tosi, koska negatiivisen korrelaation voi aiheuttaa jokin muukin syy. Kontekstin pohjalta tehty hypoteesi saisi toki negatiivisen korrelaation myötä vahvistusta, mutta lopullista päätöstä hypoteesin pätevyydestä ei pelkän negatiivisen korrelaation perusteella voida tehdä.

2.3.3 Virheraportoijan erikoistuminen ohjelmiston komponenttiin

Luvussa 2.1 tutkimuskysymyksen 2 osakysymykseksi 2 asetettiin tutkia, keskittyvätkö avoimen lähdekoodin ohjelmistoissa runsaasti virheitä raportoivat virheraportoijat raportoimaan virheitä tyypillisesti niistä komponenteista, joissa ylipäätään

on paljon virheitä. Kunkin avoimen lähdekoodin ohjelmiston virhetietokannasta oli mahdollista kerätä ristiintaulukko, jossa kunkin virheraportoijan raportoimat virheet oli lajiteltu komponenteittain. Osalla virhetietokantojen virheistä komponenttitason sijainti saattoi olla spesifioimaton – tällaiset virheet sivuutettiin tässä tarkastelussa.

Jos eri komponentit sisältävät keskenään erisuuren määrän virheitä, löytää tasaisesti eri komponentteja testaava virheraportoija oletettavasti eri määrät virheitä eri komponenteista. Kysymys siitä, raportoivatko virheraportoijat tyypillisesti virheitä tasaisesti eri komponenteista vai ennemminkin keskittyvät tiettyihin komponentteihin, voidaankin tiivistää kysymykseen, jakautuvatko kunkin virheraportoijan raportoimat virheet eri komponenttien välille samalla tavalla. Mikäli vastaus tähän kysymykseen on kieltävä eli keskittymistä raportoijilla tyypillisesti tapahtuu, on kiinnostavaa selvittää, keskittyvätkö runsaasti virheitä raportoivat virheraportoijat tyypillisesti niihin komponentteihin, joissa ylipäätään on paljon virheitä. Tässä yhteydessä tehdäänkin tutkimus kaksivaiheisesti. Ensiksi tutkitaan, onko keskittymistä ylipäätään havaittavissa; hypoteesit ovat

H_0 : Raportoidut virheet jakautuvat komponenttien välille samalla tavalla virheraportoijasta riippumatta

H_1 : Raportoitujen virheiden jakautumisessa komponenttien välille on eroja eri raportoijien välillä,

missä H_0 on nollahypoteesi ja H_1 vaihtoehtoinen hypoteesi. Mikäli nollahypoteesi H_0 hylätään ja vaihtoehtoinen hypoteesi H_1 astuu voimaan, siirrytään tutkimaan sitä, ovatko paljon virheitä raportoineet virheraportoijat raportoineet odotettua enemmän virheitä juuri niistä komponenteista, joista ylipäätään on raportoitu eniten virheitä.

Nollahypoteesia H_0 voidaan testata tilastollisesti esimerkiksi Pearsonin χ^2-testillä. Olkoon c ristiintaulukon sarakkeiden lukumäärä, y rivien lukumäärä ja O_{ij} solu, joka on ristiintaulukossa rivillä i sarakkeessa j. Kullekin ristiintaulukon solulle O_{ij} lasketaan aluksi odotettu frekvenssi E_{ij} yhtälöllä

$$E_{ij} = \frac{Y_i \cdot C_j}{T}, \tag{17}$$

C_j on sarakkeessa j olevien solujen arvojen summa missä Y_i on rivillä i olevien solujen arvojen summa, C_j sarakkeessa j olevien solujen arvojen summa ja T kaikkien ristiintaulukon solujen arvojen summa. Jos nollahypoteesi H_0 pätee, niin testisuure X^2, joka määritellään yhtälöllä

$$X^2 = \sum_{i=1}^{y} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \tag{18}$$

noudattaa suurissa otoksissa approksimatiivisesti χ^2-jakaumaa vapausastein $(y-1)(c-1)$. Suuret testisuureen X^2 arvot viittaavat siihen, ettei nollahypoteesi H_0 päde.

[10, s. 302–310] Testin p-arvo voidaan laskea helposti esimerkiksi SPSS-ohjelmistolla; tässä yhteydessä riskitasoksi valitaan 0.05.

Huomattavaa χ^2-testin käytössä on se, ettei testi välttämättä anna luotettavia tuloksia, mikäli odotetut frekvenssit E_{ij} ovat kovin matalia. Eräs nyrkkisääntö on, että odotetuista frekvensseistä E_{ij} vähintään 80% tulisi olla suurempia kuin 5. [10, s. 310–312] Tämän ehdon täyttyminen varmistettiin tarkasteltavissa avoimen lähdekoodin ohjelmistoissa siten, että eniten virheitä raportoinut prosentti virheraportoijista yhdistettiin yhdeksi raportoijaluokaksi, ja loput virheraportoijat toiseksi raportoijaluokaksi. Lisäksi komponenttien lukumäärästä riippuen joko 90% tai 80% komponenteista, joista oli raportoitu vähiten virheitä, yhdistettiin yhdeksi komponenttiluokaksi. Tällä tavoin kategorioita yhdistelemällä varmistuttiin nyrkkisäännön toteutumisesta, ja ilmiötä pystyttiin silti edelleen hyvin tarkastelemaan.

Mikäli nollahypoteesi H_0 hylättiin käytetyllä riskitasolla 0.05, siirryttiin vertailemaan raportoijista eniten virheitä raportoineen prosentin raportointia muiden virheraportoijien raportoinnin kanssa. Odotettuja frekvenssejä E_{ij} ja toteutuneita frekvenssejä O_{ij} vertailemalla voitiin päätellä, mihin komponentteihin eniten virheitä raportoineet virheraportoijat olivat keskittyneet.

3 Aikaisempi tutkimus

Tässä luvussa esitellään lyhyesti ja tiivistetysti tämän työn kannalta oleellisin aikaisempi tutkimus. Aihepiiriin liittyviä tutkimuksia etsittiin seuraavista alan tietokannoista: ACM Digital Library, Google Scholar ja IEEE Xplore. Käytettyjä hakusanoja olivat ”software test”, ”software defect” ja ”defect detection”. Tutkimusten yhteensopivuutta tämän työn aihepiiriin arvioitiin lähinnä otsikon ja tiivistelmän perusteella. Mikäli yhteensopivuutta näiden perusteella oli, tarkasteltiin tutkimusta lähemmin. Suoran tietokantahaun lisäksi tutkimuksia etsittiin myös tietokantahaulla löytyneiden, hyvin tämän työn aihepiiriin sopivien tutkimusten lähteistä.

3.1 Erilaisten virhejakaumien tutkimus

Virhetietokantaan raportoitujen virheiden jakautumista ohjelmiston komponenttien välille on tutkittu melko runsaasti. Esimerkiksi Denaro ja Pezzè [11] tutkivat Apachen HTTP-palvelimen virhetietokantaa ja havaitsivat, että virhetietokantaan raportoidut virheet olivat jakautuneet ohjelmiston komponenttien välille hyvin epätasaisesti: pieni osa komponenteista sisälsi suuren osan virheistä. Erityisesti kaupallisilla ohjelmistoilla virheiden jakautumista komponenttien välille on tutkittu paljon, ja havainnot ovat yleisesti ottaen olleet hyvin samantyyppisiä Denaron ja Pezzèn [11] tutkimuksen havainnon kanssa [1, 13, 31].

Jalote ym. [19] tutkivat erään Microsoftin Windows-käyttöjärjestelmän version virhetietokantaa kolmen dimension – ajan, virheraportoijan ja testaustekniikan – kautta. Virhetietokantaa tarkasteltiin tutkimuksessa niin kokonaisuuden kuin yksittäisten komponenttien tasolla. Eräs tutkimuksen havainnoista oli, että tyypillisesti yksittäisen komponentin testauksesta vastuussa ollut testausryhmä raportoi vain osan kaikista kyseisestä komponentista löytyneistä virheistä; tämän oletettiin johtuvan siitä, että komponenttien välisten kytkösten takia tyypillisesti yhden komponentin testauksesta vastuussa ollut ryhmä raportoi virheitä myös muista komponenteista. Andersson ja Runeson [2] tekivät samantyyppisiä havaintoja. He tutkivat tapaustutkimuksessaan erään suuren telekommunikaatioalan yrityksen virhetietokantaa komponenttitasolla ja havaitsivat, että yksittäisen komponentin testauksesta vastuussa ollut testausryhmä raportoi 2% – 95% kaikista kyseisestä komponentista raportoiduista virheistä.

3.2 Virheraportoijien tutkimus

Mockus ym. [24] tutkivat tapaustutkimuksessaan Apachen HTTP-palvelinta ja Mozillan Firefox-Internet-selainta useasta eri näkökulmasta: tutkimuksessa tarkasteltiin muun muassa ohjelmistojen virhetiheyttä, vakavuudeltaan erilaisten virheiden korjaukseen kuluvia aikoja, ohjelmistojen kehitykseen osallistuvien henkilöiden taustoja sekä ohjelmistojen kehitysprosesseja yleisesti. Aineistoina tutkimuksessa käytettiin

molempien ohjelmistojen virhetietokantoja sekä lisäksi ohjelmistokehittäjien sähköpostilistoja.

Mockuksen ym. [24] tutkimuksen aiheista erityisesti virheraportoinnin ja virheraportoijien taustojen käsittely on kiinnostavaa tämän työn kannalta. Apachen HTTP-palvelimen virheraportoijista kukaan yksittäinen virheraportoija ei ollut raportoinut virhetietokantaan poikkeuksellisen suurta määrää virheitä: eniten virheitä raportoinut virheraportoija oli raportoinut kaikkiaan 32 virhettä. Tutkimuksessa oletettiinkin tyypillisen Apachen HTTP-palvelimen virheraportoijan olevan esimerkiksi palvelimen ylläpitäjä tai Internet-kauppa vähän virheitä raportoineiden virheraportoijien melko suuren määrän vuoksi. Mozillan Firefox-Internet-selaimen virheet olivat jakautuneet raportoijien välille Apachea epätasaisemmin: yhteensä 113 virheraporotoijaa oli raportoinut virhetietokantaan yli sata virhettä. Tutkimuksessa epäiltiinkin, että Mozillalla – toisin kuin Apachella – olisi tietty joukko testaukselle omistautuneita vapaaehtoisia.

3.3 Virheiden ennustamisen tutkimus

Virhetietokantojen sisältämää dataa on hyödynnetty myös lukuisissa virheiden ennustamista käsittelevissä tutkimuksissa. Perusideana näissä tutkimuksissa on yleensä löytää jonkinlainen vahva suhde jonkin ohjelmiston komponentin kehityksen alkuvaiheessa mitattavan suureen ja komponentista myöhemmin löytyvien virheiden välille. Tällaisen suureen tunteminen olisi siitä hyödyllistä, että sen avulla pystyttäisiin aikaisin tunnistamaan ne ohjelmiston komponentit, joissa todennäköisesti esiintyy myöhemmin runsaasti virheitä. D'Ambros ym. [12] esittelevät ja vertailevat tutkimuksessaan kattavasti aikaisemmissa tutkimuksissa esiteltyjä virheiden ennustamismenetelmiä.

3.4 Johtopäätökset aikaisemmasta tutkimuksesta

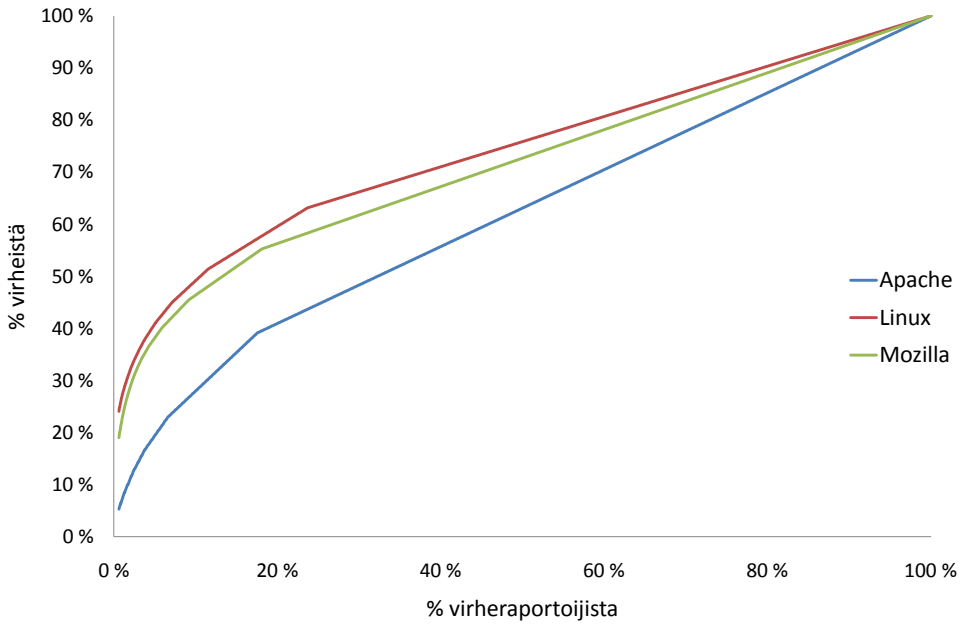
Yleisenä johtopäätöksenä aikaisemman tutkimuksen tarkastelun perusteella voitaneen todeta, että melko runsaasta virheraportoinnin, virheiden ja yleisestikin ohjelmistotestauksen tutkimuksesta huolimatta ei tämän työn aihepiiriä – virhejakaumia ja virheraportoijien välisten raportointimäärien erojen syitä – ole aiemmassa tutkimuksessa juuri tarkasteltu. Poikkeuksen tähän tekee Mockuksen ym. [24] tutkimus, jossa osin sivuttiin tämän työn aihepiiriä.

4 Tulokset

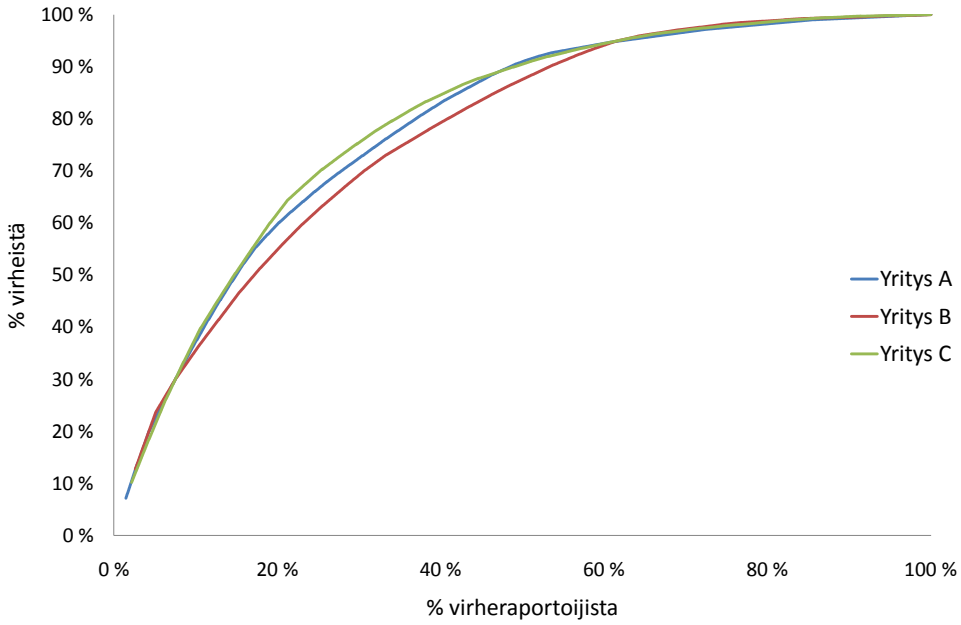
Tarkasteltujen avoimen lähdekoodin ohjelmistojen ja ohjelmistoyritysten virhekirjanpitohistoriat kerättiin luvussa 2.2 kuvatulla tavalla. Taulukossa (3) esitetään tiivistetysti joitain perustietoja kerätystä datasta. Kuvissa (1) ja (2) esitetään tarkasteltujen avoimen lähdekoodin ohjelmistojen ja ohjelmistoyritysten Albergin diagrammit [30], jotka havainnollistavat, kuinka raportoitujen virheiden lukumäärät jakautuvat virheraportoijien välille kussakin aineistossa. Liitteen A kuvassa (A1) esitetään Pareto-jakauman Albergin diagrammeja eri parametrien α arvoilla.

Taulukko 3: Perustietoja kerätystä datasta.

	Apache	**Linux**	**Mozilla**	**Yritys A**	**Yritys B**	**Yritys C**
Virheraporttien lkm	2389	6713	13610	981	790	1368
Korjaukseen johtaneiden virheraporttien lkm	842	4100	3246	825	719	920
Virheraportoijien lkm	1763	3241	7435	69	39	45
Suurin yksittäinen raportointimäärä	30	884	287	70	99	138
Keskimääräinen raportointimäärä	1.36	2.07	1.83	14.22	20.26	29.74



Kuva 1: Tarkasteltujen avoimen lähdekoodin ohjelmistojen Albergin diagrammit.



Kuva 2: Tarkasteltujen ohjelmistoyritysten Albergin diagrammit.

4.1 Empiirisen jakauman yhteensopivuus Pareto-jakauman tai muun teoreettisen jakauman kanssa

Empiirisen jakauman yhteensopivuutta Pareto-jakauman tai muun teoreettisen jakauman kanssa tutkittiin työkalulla [18], johon oli implementoitu luvussa 2.3.1 esitelty menetelmä. Yhteensopivuutta tutkittiin kokonaisen empiirisen jakauman lisäksi myös pelkällä jakauman hännällä, mikäli katkaisupiste x_{min} erosi empiirisen jakauman pienimmästä havaintoarvosta, joka oli itse asiassa jokaisella aineistolla 1.

Liitteen A taulukossa (A1) esitetään kokonaisiin empiirisiin virhejakaumiin sovitettujen Pareto-jakaumien parametrien suurimman uskottavuuden estimaatit $\hat{\alpha}$ sekä empiiristen virhejakaumien ja Pareto-jakaumien yhteensopivuustestien p-arvot. Liitteen A taulukossa (A2) esitetään suurimman uskottavuuden menetelmällä lasketut empiiristen virhejakaumien katkaisupisteet x_{min}, katkaistujen virhejakaumien häntiin sovitettujen Pareto-jakaumien parametrien suurimman uskottavuuden estimaatit $\hat{\alpha}$ sekä empiiristen virhejakaumien ja Pareto-jakaumien yhteensopivuustestien p-arvot. Taulukoissa (A1) ja (A2) esitetään myös tiivistetysti päätulokset Pareto-jakauman ja muiden teoreettisten jakaumien vertailusta: kunkin empiirisen jakauman osalta kerrotaan uskottavuusosamäärätestin testisuureena käytetyn normalisoidun log-uskottavuusosamäärän arvo (UO) sekä tämän perusteella laskettu p-arvo. Taulukoiden (A1) ja (A2) viimeisissä sarakkeissa esitetään tiivistetysti tarkastelun pohjalta tehty johtopäätös; johtopäätösvaihtoehdot esitellään taulukon (A1) selosteessa.

Taulukon (A1) tulosten mukaan Apachen ja Linuxin kokonaisten virhejakaumien

voidaan katsoa noudattavan Pareto-jakaumaa kohtuullisesti – Pareto-jakauman ja empiiristen virhejakaumien yhteensopivuudet eivät eroa tilastollisesti merkitsevästi lognormaalijakauman vastaavista yhteensopivuuksista. Mozillan ja ohjelmistoyritysten virhejakaumat sen sijaan eivät selkeästi noudata Pareto-jakaumaa. Mozillan virhejakauma ei noudata mitään muutakaan tarkastelluista teoreettisista jakaumista Pareto-jakaumaa paremmin. Yritysten A–C virhejakaumat noudattavat Pareto-eksponenttijakaumaa puhdasta Pareto-jakaumaa paremmin, mutta muitakin virhejakaumiin Pareto-jakaumaa paremmin sopivia teoreettisia jakaumia on tarkastelun perusteella olemassa. Pareto-jakaumien parametrien suurimman uskottavuuden estimaatit $\hat{\alpha}$ saavat kullakin avoimen lähdekoodin ohjelmistolla arvonsa väliltä $[2.55, 2.97]$. Tämä sopii hyvin Clausetin ym. [9] väitteeseen siitä, että Pareto-jakauman parametri α kuuluu tyypillisesti välille $(2, 3)$. Yritysten virhejakaumiin sovitettujen Pareto-jakaumien parametrien suurimman uskottavuuden estimaatit $\hat{\alpha}$ ovat kukin selkeästi avoimen lähdekoodin vastaavia estimaatteja pienempiä, sillä Pareto-jakaumien parametrien suurimman uskottavuuden estimaatit $\hat{\alpha}$ saavat kussakin yritysaineistossa arvonsa väliltä $[1.31, 1.41]$. Tulkinnallisesti pienempi parametri α merkitsee raskaampaa jakauman häntää.

Taulukon (A2) tulosten mukaan yritystä C lukuunottamatta jokaisen tarkastellun yrityksen ja avoimen lähdekoodin ohjelmiston virhejakauman hännän voidaan katsoa noudattavan Pareto-jakaumaa. Avoimen lähdekoodin ohjelmistojen virhejakaumien katkaisupisteet ovat melko matalia: Linuxille $x_{min} = 2$, Mozillalle $x_{min} = 3$ ja Apachelle jopa $x_{min} = 1$, mikä tarkoittaa, ettei Apachen virhejakauman häntää tarvitse erikseen tarkastella – häntä muodostuu koko virhejakaumasta. Arvio Linuxin virhejakauman hännän yhteensopivuudesta Pareto-jakauman kanssa on kohtuullinen, koska Pareto-jakauma ei sovi häntään tilastollisesti merkitsevästi lognormaalijakaumaa, Poisson-jakaumaa eikä Yulen jakaumaa paremmin. Myös Mozillalla arvio jää hyvän sijasta kohtuulliseksi, sillä Pareto-jakauma ei sovi Mozillan virhejakauman häntään tilastollisesti merkitsevästi lognormaalijakaumaa eikä Yulen jakaumaa paremmin. Sekä Linuxin että Mozillan virhejakaumien häntiin sovitettujen Pareto-jakaumien parametrien suurimman uskottavuuden estimaattien $\hat{\alpha}$ arvot ovat kokonaisiin virhejakaumiin sovitettuja vastaavia estimaatteja pienempiä, mutta kuuluvat molemmat edelleen välille $(2, 3)$.

Taulukon (A2) tulosten mukaan yritysten virhejakaumien katkaisupisteet ovat avoimen lähdekoodin ohjelmistojen virhejakaumien katkaisupisteitä korkeampia: yrityksellä A $x_{min} = 12$, yrityksellä B $x_{min} = 26$ ja yrityksellä C $x_{min} = 15$. Yritysten virhejakaumien häntien tarkastelussa huomattavaksi tekijäksi muodostuukin havaintojen pieni lukumäärä hännässä: yrityksellä A virhejakauman hännän muodostaa 32, yrityksellä B 12 ja yrityksellä C 24 havaintoa. Näin pienellä havaintojen lukumäärällä yhteensopivuustarkastelu jää melko epäluotettavaksi, ja myös tarkastelun motiivi kenties hieman kyseenalaistuu. Joka tapauksessa yrityksen B virhejakauman hännän voidaan katsoa noudattavan Pareto-jakaumaa kohtuullisesti – Pareto-jakauma sopii häntään tilastollisesti merkitsevästi tarkastelluista jakaumista ainoastaan Poisson-jakaumaa paremmin. Pareto-eksponenttijakauma sopii yritysten A ja C virhejakaumien häntiin tilastollisesti merkitsevästi puhdasta Pareto-jakaumaa parem-

min, mutta muitakin hyvin sopivia teoreettisia jakaumia on tarkastelun perusteella olemassa. Yritysaineistolla virhejakauman katkaisu muuttaa voimakkaasti Pareto-jakauman parametrien suurimman uskottavuuden estimaattien $\hat{\alpha}$ arvoja: kullakin yrityksellä estimaatin arvo kasvaa, ja estimaatit saavat katkaistuilla aineistoilla arvonsa väliltä $[1.96, 3.01]$.

4.2 Virheraportoijan raportoimien virheiden lukumäärän ja korjausprosentin yhteys

Kustakin aineistosta laskettiin kaavalla (15) virheraportoijien raportoimien virheiden lukumäärien ja korjausprosenttien väliset korrelaatiot. Päätös siitä, eroaako yksittäinen korrelaatio tilastollisesti merkitsevästi nollasta, tehtiin yhtälöllä (16) lasketun t-testisuureen avulla määritetyn p-arvon perusteella. Taulukossa (4) esitetään lasketut korrelaatiot ja p-arvot; valitulla riskitasolla 0.05 tilastollisesti merkitsevää korrelaatiota ilmaisevat p-arvot on lihavoitu.

Taulukko 4: Virheraportoijien raportoimien virheiden lukumäärien ja korjausprosenttien väliset korrelaatiot sekä merkitsevyystestin p-arvo. Tilastollisesti merkitsevää korrelaatiota ilmaisevat p-arvot on lihavoitu.

	Apache	**Linux**	**Mozilla**	**Yritys A**	**Yritys B**	**Yritys C**
Korrelaatio	0.07	0.02	0.18	0.02	0.20	0.07
Merkitsevyystestin p-arvo	**0.00**	0.29	**0.00**	0.89	0.22	0.65

Taulukosta (4) nähdään, että Apachen ja Mozillan aineistoissa virheraportoijien raportoimien virheiden lukumäärien ja korjausprosenttien korrelaatiot eroavat tilastollisesti merkitsevästi nollasta, kun taas Linuxin ja yritysten A–C aineistoissa korrelaatiot eivät eroa tilastollisesti merkitsevästi nollasta käytetyllä riskitasolla 0.05. Tilastollisesti merkitsevästi nollasta eroavat korrelaatiot ovat molemmat positiivisia. Mikäli suuri virheraportointimäärä johtuisi tyypillisesti siitä, että raportoija raportoi virhetietokantaan pienimmätkin havaitsemansa virheet, olisivat virheraportoijien raportoimien virheiden lukumäärien ja korjausprosenttien korrelaatiot oletettavasti negatiivisia. Tällainen hypoteesi negatiivisesta korrelaatiosta ei kuitenkaan saa ainakaan tässä työssä tarkastellun empiirisen aineiston perusteella tukea. Virheraportoijan raportoimien virheiden lukumäärällä ja korjausprosentilla ei missään aineistoista ole vastaavien pistediagrammien tarkastelun perusteella havaittavissa myöskään mitään selkeää epälineaarista riippuvuutta.

4.3 Virheraportoijan erikoistuminen ohjelmiston komponenttiin

Virheraportoijan erikoistumista ohjelmiston komponenttiin tutkittiin avoimen lähdekoodin ohjelmistojen datalla luvussa 2.3.3 kuvatulla menetelmällä. Seuraavaksi esitellään erikseen kullakin ohjelmistolla saatuja tuloksia.

4.3.1 Apache

Apachella yksittäinen virhe kuuluu virhetietokannassa johonkin 63 mahdollisesta komponentista. Komponenttiluokkaan "All" on luokiteltu komponenttitason sijainniltaan spesifioimattomat virheet, jotka tässä tarkastelussa sivuutetaan. Keskimäärin yksittäisestä komponentista oli Apachella raportoitu noin 33 virhettä, ja suurin yksittäinen raportointimäärä oli komponentin "Core" raportointimäärä 392 virhettä.

Apachen aineistossa noin prosentti eniten virheitä raportoineista virheraportoijista koostui niistä raportoijista, jotka olivat raportoineet virhetietokantaan yhteensä 6 virhettä tai enemmän. Tämä prosentti virheraportoijista oli raportoinut noin 7% kaikista virhetietokantaan raportoiduista virheistä. Ne 10% komponenteista, joista oli raportoitu eniten virheitä, sisälsivät noin 58% kaikista virhetietokantaan raportoiduista virheistä.

Tehdyn Pearsonin χ^2-testin nollahypoteesi hylättiin käytetyllä riskitasolla; testin p-arvoksi saatiin likimain 1.4 tuhannesosaa. Tämän perusteella voidaan katsoa, että paljon virheitä virhetietokantaan raportoineiden virheraportoijien virheet jakautuvat komponenttien välille eri tavoin kuin muiden virheraportoijien raportoimat virheet. Odotettuja ja toteutuneita frekvenssejä vertailemalla voidaan huomata, että eniten virheitä raportoinut prosentti virheraportoijista raportoi odotettua frekvenssiä enemmän virheitä muun muassa niistä komponenteista, joista oli raportoitu ylipäätään eniten ja kolmanneksi eniten virheitä. Sen sijaan eniten virheitä raportoinut prosentti virheraportoijista raportoi odotetutta frekvenssiä vähemmän virheitä muun muassa niistä komponenteista, joista oli raportoitu ylipäätään toiseksi ja neljänneksi eniten virheitä.

4.3.2 Linux

Linuxin virhetietokannassa yksittäinen virhe kuuluu johonkin 138 mahdollisesta komponentista. Komponenttiluokkaan "Other" on luokiteltu komponenttitason sijainniltaan spesifioimattomat virheet, jotka tässä tarkastelussa sivuutetaan. Keskimääräinen raportointimäärä yksittäisessä komponentissa oli Linuxilla noin 40 virhettä, ja suurin yksittäinen raportointimäärä oli komponentin "Network" raportointimäärä 340 virhettä.

Linuxin aineistossa noin prosentti eniten virheitä raportoineista virheraportoijista koostui niistä raportoijista, jotka olivat raportoineet virhetietokantaan yhteensä 13

virhettä tai enemmän. Nämä virheraportoijat olivat raportoineet noin 25% kaikista virhetietokantaan raportoiduista virheistä. Ne 10% komponenteista, joista oli raportoitu eniten virheitä, sisälsivät noin 45% kaikista virhetietokantaan raportoiduista virheistä.

Tehdyn Pearsonin χ^2-testin nollahypoteesi hylättiin kaikilla järkevillä riskitasoilla; testin p-arvo oli yhtä miljoonasosaa pienempi. Tämän perusteella voidaan katsoa, että paljon virheitä virhetietokantaan raportoineiden virheraportoijien virheet jakautuvat komponenttien välille eri tavoin kuin muiden virheraportoijien raportoimat virheet. Odotettuja ja toteutuneita frekvenssejä vertailemalla voidaan huomata, että eniten virheitä raportoinut prosentti virheraportoijista raportoi odotettua frekvenssiä vähemmän virheitä kustakin niistä neljästä komponentista, joista oli raportoitu ylipäätään eniten virheitä.

4.3.3 Mozilla

Mozillan virhetietokannassa yksittäinen virhe voi kuulua johonkin 26 mahdollisesta komponentista. Komponenttiluokkaan "General" on luokiteltu komponenttitason sijainniltaan spesifioimattomat virheet, jotka tässä tarkastelussa sivuutetaan. Keskimäärin Mozillalla oli raportoitu yksittäisessä komponentista noin 346 virhettä, ja suurin yksittäinen raportointimäärä oli komponentista "Bookmarks & History" raportoidut 2172 virhettä.

Noin prosentti eniten virheitä raportoineista virheraportoijista koostui Mozillalla niistä virheraportoijista, jotka olivat raportoineet virhetietokantaan yhteensä 16 virhettä tai enemmän. Näiden virheraportoijien raportoimat virheet muodostivat noin 25% kaikista virhetietokantaan raportoiduista virheistä. Ne 20% komponenteista, joista oli raportoitu eniten virheitä, sisälsivät noin 59% kaikista virhetietokantaan raportoiduista virheistä.

Tehdyn Pearsonin χ^2-testin nollahypoteesi hylättiin kaikilla järkevillä riskitasoilla; testin p-arvo oli yhtä miljoonasosaa pienempi. Näin ollen voidaan katsoa, että paljon virheitä virhetietokantaan raportoineiden virheraportoijien virheet jakautuvat komponenttien välille eri tavoin kuin muiden virheraportoijien raportoimat virheet. Odotettuja ja toteutuneita frekvenssejä vertailemalla voidaan huomata, että eniten virheitä raportoinut prosentti virheraportoijista raportoi odotettua frekvenssiä enemmän virheitä siitä komponentista, josta oli raportoitu ylipäätään eniten virheitä. Kaikista muista yksittäisistä komponenteista sekä yhdistetystä komponenttiluokasta eniten virheitä raportoinut prosentti virheraportoijista oli raportoinut odotettua frekvenssiä vähemmän virheitä.

5 Pohdinta

Tässä luvussa vastataan aluksi luvussa 2.1 esitettyihin tutkimuskysymyksiin, minkä jälkeen käsitellään tämän työn tulosten ja tutkimusmenetelmien rajoituksia. Luvun lopuksi esitellään ajatuksia aihepiirin tulevasta tutkimuksesta.

5.1 Vastaukset tutkimuskysymyksiin

Tutkimuskysymys 1: Noudattavatko virheraportoijien virhetietokantaan raportoimien virheiden jakaumat Pareto-jakaumaa tai jotain muuta tunnettua teoreettista jakaumaa tarkasteltavissa yrityksissä tai avoimen lähdekoodin ohjelmistoissa?
Luvun 4.1 tulosten perusteella voitaneen todeta, että erityisesti tutkittujen avoimen lähdekoodin ohjelmistojen – Apachen, Linuxin ja Mozillan – virhejakaumien häntien voidaan katsoa noudattavan melko hyvin Pareto-jakaumaa. Apachella ja Linuxilla myös kokonaisten virhejakaumien voidaan katsoa noudattavan Pareto-jakaumaa melko hyvin – Apachellahan katkaisua ei itse asiassa tarvinnut tehdä ollenkaan, vaan häntä muodostui koko virhejakaumasta. Pareto-jakauman lisäksi erityisesti lognormaalijakauman havaittiin sopivan useampaan empiiriseen jakaumaan tai jakauman häntään hyvin; tyypillisesti valitulla riskitasolla ei pystytty tekemään päätöstä, kumpi teoreettisista jakaumista sopi empiiriseen ainestoon paremmin. Tämäntyyppinen Pareto-jakauman ja lognormaalijakauman kytkös on havaittu muuallakin: keskenään hyvin samanlaiset generoivat mekanismit voivat tuottaa niin Pareto- kuin lognormaalijakauman, ja usein on hankalaa päätellä, kumpaa jakaumista empiirinen jakauma tarkasti noudattaa [23].

Ohjelmistoyritysten virhejakaumien ja Pareto-jakauman yhteensopivuudesta ei pystytä tekemään erityisen luotettavia johtopäätöksiä, koska havaintoja eli yksittäisiä virheraportoijia ei yritysten virhetietokannoissa ollut kovin paljoa. Clauset ym. [9] esittävät artikkelissaan nyrkkisäännön, jonka mukaan Pareto-jakauman parametri α voidaan estimoida luotettavasti silloin, kun havaintojen määrä on likimain 50 tai suurempi. Taulukosta (3) nähdään, että ainoastaan yrityksen A virhetietokannassa havaintomäärä eli yksittäisten virheraportoijien lukumäärä on suurempi kuin 50; yritysten B ja C havaintomäärät ovat 39 ja 45 vastaavasti. Yritysten virhejakaumien hännissä taas havaintojen lukumäärä on suurimmillaankin vain 32. Kohtuullisen luotettavasti voitaneen kuitenkin todeta luvussa 4.1 esitettyjen tulosten perusteella, ettei yrityksen A virhejakauman eikä mahdollisesti yrityksen B virhejakauman voida katsoa noudattavan Pareto-jakaumaa ja että Pareto-eksponenttijakauma on puhdasta Pareto-jakaumaa selkeästi parempi malli ainakin yritysten A ja B kokonaisille virhejakaumille. Taulukon (A2) yritysten virhejakaumien häntiä koskeviin tuloksiin tulee pienen havaintojen lukumäärän takia suhtautua melko suurella varauksella.

Jonkinlaisena yleisenä johtopäätöksenä tarkastelun perusteella voinee todeta, että tässä työssä saatiin jonkin verran empiiristä näyttöä sille, että virhejakaumat – tai

ainakin virhejakaumien hännät – todella noudattaisivat Pareto-jakaumaa. Pareto-jakauma ei tosin sopinut kaikkiin empiirisiin jakaumiin hyvin, ja lisäksi jotkin muutkin tarkastelluista teoreettisista jakaumista vaikuttivat lupaavilta malleilta osassa aineistoista.

Tutkimuskysymys 2: Mistä erot raportointimäärissä yksittäisten virheraportoijien välillä johtuvat?
Kirjallisuuden ja puhtaasti intuitionkin perusteella voidaan tehdä hypoteeseja seikoista, jotka saattaisivat vaikuttaa yksittäisen virheraportoijan raportoimien virheiden lukumäärään. Intuitiivisesti esimerkiksi lienee selvää, että virheraportoijan raportoimien virheiden lukumäärä on sitä suurempi, mitä enemmän aikaa virheraportoija käyttää ohjelmiston testaamiseen. Lisäksi esimerkiksi Jaloten ym. [19] mukaan ohjelmiston kehityksen aikaisemmissa vaiheissa löydetään tyypillisesti enemmän virheitä kuin myöhäisemmissä vaiheissa – virheraportoijan raportoimien virheiden lukumäärä saattaisi siten riippua myös siitä, missä ohjelmiston kehitysvaiheessa virheraportoija ohjelmistoa testaa. Tämän työn empiirisellä tutkimusaineistolla ei kuitenkaan voinut tutkia esimerkiksi tällaisia asioita lainkaan, sillä tarkasteluun vaadittavia tietoja ei ollut virhetietokannoissa ilmoitettu.

Seuraavaksi esitetään vastaukset kahteen tämän tutkimuskysymyksen osakysymykseen.

Osakysymys 1: Onko virheraportoijien raportoimien virheiden lukumäärillä ja korjausprosenteilla jonkinlaista yhteyttä tarkasteltavissa yrityksissä tai avoimen lähdekoodin ohjelmistoissa?
Tämän osakysymyksen ajatuksena oli tutkia virheraportoijien raportointialttiuden vaikutusta raportointimäärään: voiko yksittäisen virheraportoijan suurta virheraportointimäärää perustella sillä, että raportoija raportoi virhetietokantaan pienimmätkin havaitsemansa virheet? Luvun 4.2 tulosten perusteella voitaneen todeta, ettei raportoijien raportoimien virheiden lukumäärillä ja virheiden korjausprosenteilla näyttäisi olevan ainakaan minkäänlaista selkeää yhteyttä tarkastellussa empiirisessä aineistossa. Aineisto ei siis näyttäisi tukevan osakysymyksen taustalla olevaa hypoteesia.

Osakysymys 2: Keskittyvätkö runsaasti virheitä raportoivat virheraportoijat tyypillisesti raportoimaan virheitä niistä komponenteista, joista ylipäätään on raportoitu paljon virheitä?
Tämän osakysymyksen ajatuksena oli selvittää, voiko virheraportoijan suurta raportointimäärää perustella sillä, että virheraportoija keskittyy testaamaan sellaisia komponentteja, joissa ylipäätään on paljon virheitä. Huomattavaa on, että myös aikaisempi tutkimus omalla tavallaan motivoi tätä osakysymystä: luvussa 3.4 esitettyjen useiden tutkimusten [1, 11, 13, 31] havainnot siitä, että virheet jakautuvat tyypillisesti ohjelmiston komponenttien välille hyvin epätasaisesti, johtavat melko luontevasti kysymyksen taustalla olevaan hypoteesiin.

Luvun 4.3 tulosten perusteella nähdään, että erityisesti Mozillalla eniten virheitä raportoinut prosentti virheraportoijista oli voimakkaasti keskittynyt raportoinnis-

saan juuri siihen yksittäiseen komponenttiin, josta oli raportoitu ylipäätään eniten virheitä. Apachella voidaan nähdä samantyyppistä, joskaan ei yhtä voimakasta keskittymistä kuin Mozillalla. Linuxilla keskittymistä voidaan havaita, mutta se on juuri hypoteesiin nähden päinvastaista: eniten virheitä raportoinut prosentti virheraportoijista raportoi odotettua vähemmän virheitä niistä neljästä komponentista, joista oli raportoitu eniten virheitä. Johtopäätöksenä tarkastelusta voitaneen todeta, että Apachen ja erityisesti Mozillan aineistot tukevat kysymyksen taustalla olevaa hypoteesia, kun taas Linuxin aineisto todistaa hypoteesia vastaan.

5.2 Rajoitukset

Tässä luvussa esitellään saatujen tulosten sekä käytettyjen tutkimusmenetelmien mahdollisia rajoituksia. Yhden suuren uhan tämän työn tulosten validiteetille muodostavat mahdolliset virheet virhetietokantojen datassa. Periaatteessa täysin mahdollista on, ettei kaikkia ohjelmistosta havaittavia ongelmia raportoida virhetietokantaan, vaan osa ongelmista saatetaan esimerkiksi kertoa suoraan ohjelmiston kehittäjälle. Edelleen mahdollista on, että yksittäinen virheraportoija raportoi virhetietokantaan jonkun muun löytämiä virheitä.

Tutkimuskysymyksen 2 ensimmäisen osakysymyksen ajatuksena oli tutkia virheraportoijien raportointialttiuden vaikutusta raportointimäärään. Virheen korjausprosenttia käytettiin tässä yhteydessä eräänlaisena virheen tärkeyden mittarina, ja mittarin hyvyys on periaatteessa mahdollista kyseenalaistaa.

Esimerkiksi yrityskontekstissa virheraportoijan rooli tai asema organisaatiossa saattaa hyvinkin vaikuttaa korjausprosenttiin voimakkaasti – ehkä esimerkiksi toimitusjohtajan virheet korjataan, vaikka ne olisivatkin muiden virheraportoijien raportoimia virheitä merkityksettömämpiä. Avoimen lähdekoodin ohjelmistojen kontekstissa Linuxilla sama saattaisi päteä esimerkiksi Linus Torvaldsin raportoimiin virheisiin. Edelleen on huomattavaa, että tarkastellussa yritysaineistossa kullakin yrityksellä juuri ohjelmistokehittäjien virheiden korjausprosentit olivat systemaattisesti korkeimpia [27]. Näistäkin syistä huolimatta arvioisin kuitenkin virheiden korjausprosentin esimerkiksi virheraportissa ilmoitettavaa virheen kiireellisyyttä tai vakavuutta paremmaksi virheen tärkeyden mittariksi, sillä virheen korjausprosentti ei kuitenkaan ole täysin virheraportoijan subjektiivisesti päätettävissä. Mahdollista toki on myös se, että juuri esimerkiksi edellä esitetyt syyt aiheuttivat sen, ettei löydettyjen virheiden lukumäärien ja korjausprosenttien välillä nähty yhteyttä tarkastellussa empiirisessä aineistossa.

Tutkimuskysymyksen 2 toisen osakysymyksen ajatuksena oli selvittää, voitaisiinko virheraportoijan suurta raportointimäärää perustella sillä, että virheraportoija olisi keskittynyt testaamaan sellaisia komponentteja, joissa ylipäätään on paljon virheitä. Tämänkin osakysymyksen tarkastelussa on muutamia kyseenalaistettavia piirteitä.

Periaatteessa se, että komponentista on raportoitu paljon virheitä, ei välttämättä tarkoita, että komponentissa todella olisi paljon virheitä – onhan esimerkiksi mah-

dollista, että komponenttia, josta on raportoitu paljon virheitä, on vain testattu muita komponentteja perusteellisemmin. Tämän ongelman arvioisin kuitenkin merkitykseltään melko pieneksi. Tarkastellut avoimen lähdekoodin ohjelmistot ovat erittäin paljon käytettyjä ohjelmistoja, joita on oletettavasti testattu poikkeuksellisen kattavasti.

Eräs toinen mahdollisesti kyseenalaistettava piirre perustuu testausasetelman monimutkaisuuteen: kuinka voidaan olla varmoja, että yksittäistä komponenttia testaava virheraportoija löytää virheen juuri testaamastaan komponentista eikä jostain muusta komponentista? Kaupallisten ohjelmistojen tutkimuksissa [2, 19] on tehty havaintoja, että komponenttien välisten kytkösten vuoksi yksittäisen komponentin testauksesta vastuussa oleva ryhmä löytää tyypillisesti runsaasti virheitä myös muista kuin varsinaisesti testaamastaan komponentista. Mikäli yksittäistä komponenttia testaava virheraportoija todella voi löytää virheen muusta kuin testaamastaan komponentista, ei virheraportoijan raportoimien virheiden komponenttitason sijainnin perusteella voi siten luotettavasti päätellä, mihin ohjelmiston osa-alueeseen virheraportoija on testaamisessaan keskittynyt. Tämän ongelman merkitysen suuruutta on vaikea arvioida, mutta joka tapauksessa se on oletettava pieneksi, jotta tehty tarkastelu olisi järkevä.

5.3 Ajatuksia tulevasta tutkimuksesta

Tämän työn aihepiiristä ei juuri ollut aikaisempaa tutkimusta, joten tämän työn havaintojakin tulisi kenties pitää perustavanlaatuisten tulosten sijaan ehkä ennemminkin vain tulevien tutkimuksien hypoteeseina. Tässä työssä esitettyä tarkastelua olisikin mielenkiintoista laajentaa uusiin aineistoihin. Erityisen kiinnostavia saattaisivat olla aineistot, joissa kävisi jotenkin ilmi virheraportoijien testaamiseen käyttämät ajat – tällöin pystyttäisiin tutkimaan pelkkien virheraportoijien raportoimien virhemäärien lisäksi myös virheraportoijien tuottavuuksia raportoituina virheinä aikayksikköä kohti. Virheraportoijien tuottavuuksien tutkiminen voisi joiltain osin olla jopa pelkkien virhemäärien tutkimista kiinnostavampaa. Tuottavuuksia tutkimalla saatettaisiin parhaimmillaan pystyä esimerkiksi selvittämään, millaisia ominaisuuksia menestyksekkäillä virheraportoijilla on tai mitkä testaustekniikat ovat käytännössä kaikkein tehokkaimpia.

Viitteet

[1] Andersson, C. ja Runeson, P. A Replicated Quantitative Analysis of Fault Distributions in Complex Software Systems. IEEE Transactions on Software Engineering, 2007. Vol.33:5. S. 273–286. ISSN 0098-5589.

[2] Andersson, C. ja Runeson, P. Investigating Test Teams' Defect Detection in Function test. Teoksessa: First International Symposium on Empirical Software Engineering and Measurement. Madrid, Espanja. 20-21.9.2007. ESEM, 2007. S. 458–460. ISBN 978-0-7695-2886-1.

[3] Apache-ohjelmiston virhetietokanta. Viitattu 17.8.2010. Saatavissa: https://issues.apache.org/bugzilla/

[4] Arnold, B. C. Pareto Distributions. Fairland, Maryland, USA: International Co-operative Publishing House, 1983.

[5] Black, R. Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing. 2nd ed. New York, USA: John Wiley & Sons, 2002. 500 s. ISBN 0-471-22398-0.

[6] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. ja Wiener, J. Graph Structure in the Web. Computer Networks: The International Journal of Computer and Telecommunications Networking, 2000. Vol.33:1-6. S. 309–320. ISSN 1389-1286.

[7] Bugzilla-ohjelmiston dokumentaatio. Viitattu 17.8.2010. Saatavissa: http://www.bugzilla.org/docs/

[8] Burnstein, I. Practical Software Testing: a Process-oriented Approach. New York, USA: Springer Verlag, 2003. 709 s. ISBN 0-387-95131-8.

[9] Clauset, A., Shalizi, C. R. ja Newman, M. E. J. Power-law Distributions in Empirical Data. SIAM Review, 2009. Vol.51:4. S. 661–703. ISSN 0036-1445.

[10] Coolican, H. Research Methods and Statistics in Psychology. 3rd ed. London, UK: Hodder & Stoughton Educational, 1999. 591 s. ISBN 0-340-74760-9.

[11] Denaro, G. ja Pezzè, M. An Empirical Evaluation of Fault-proneness Models. Teoksessa: Proceedings of the 24th International Conference on Software Engineering. Orlando, Florida, USA: IEEE Computer Society, 2002. S. 241–251. ISBN 1-58113-472-X.

[12] D'Ambros, M., Lanza, M. ja Robbes, R. An Extensive Comparison of Bug Prediction Approaches. Teoksessa: Proceedings of the 7th International Working Conference on Mining Software Repositories. Cape Town, Etelä-Afrikka: MSR, 2010. S. 31–41. ISBN 978-1-4244-6802-7.

[13] Fenton, N. E. ja Ohlsson, N. Quantitative Analysis of Faults and Failures in a Complex Software System. IEEE Transactions on Software Engineering, 2000. Vol.26:8. S. 797–814. ISSN 0098-5589.

[14] Gabaix, X. Zipf's Law For Cities: An Explanation. Quarterly Journal of Economics, 1999. Vol.114:3. S. 739– 767. ISSN 0033-5533.

[15] Goldstein, M. L., Morris, S. A. ja Yen, G. G. Problems With Fitting to the Power-law Distribution. The European Physical Journal B, 2004. Vol.41:2. S. 255–258. ISSN 1434-6028.

[16] Hatton, L. Power-Law Distributions of Component Size in General Software Systems. IEEE Transactions on Software Engineering, 2009. Vol.35:4. S. 566–572. ISSN 0098-5589.

[17] IEEE Std 610.12. IEEE Standard Glossary of Software Engineering Terminology. New York: IEEE. 1990. 82 s.

[18] Jakauman sovituksessa käytetty työkalu. Viitattu 17.8.2010. Saatavissa: http://tuvalu.santafe.edu/~aaronc/powerlaws/

[19] Jalote, P., Munshi, R. ja Probsting, T. The When-Who-How Analysis of Defects for Improving the Quality Control Process. Journal of Systems and Software, 2007. Vol.80:4. S. 584–589. ISSN 0164-1212.

[20] Jambunathan, M. V. Some Properties of Beta and Gamma Distributions. The Annals of Mathematical Statistics, 1954. Vol.25:2. S. 401–405. ISSN 0003-4851.

[21] Le Cam, L. Maximum Likelihood: an Introduction. International Statistical Review, 1990. Vol.58:2. S. 153– 171. ISSN 0306-7734.

[22] Linux-ohjelmiston virhetietokanta. Viitattu 17.8.2010. Saatavissa: https://bugzilla.kernel.org/

[23] Mitzenmacher, M. A Brief History of Generative Models for Power Law and Lognormal Distributions. Internet mathematics, 2004. Vol.1:2. S. 226–251. ISSN 1542-7951.

[24] Mockus, A., Fielding, R. T. ja Herbsleb, J. D. Two Case Studies of Open Source Software Development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology, 2002. Vol.11:3. S. 309–346. ISSN 1049-331X.

[25] Mozilla-ohjelmiston virhetietokanta. Viitattu 17.8.2010. Saatavissa: https://bugzilla.mozilla.org/

[26] Mäntylä, M. V. ja Lassenius, C. What Types of Defects Are Really Discovered in Code Reviews? IEEE Transactions on Software Engineering, 2009. Vol.35:3. S. 430–448. ISSN 0098-5589.

[27] Mäntylä, M. V., Iivonen, J. ja Itkonen, J. Who Tested My Software – an Industrial Case Study of Organization, Values, and Distribution of Testing. Tarkastuksessa, ESEM 2010.

[28] Nakagawa, T. ja Osaki, S. The Discrete Weibull Distribution. IEEE Transactions on Reliability, 1975. Vol. 24:5. S. 300–301. ISSN 0018-9529.

[29] Newman, M. E. J. Power Laws, Pareto Distributions and Zipf's Law. Contemporary physics, 2005. Vol.46:5. S. 323–351. ISSN 1366-5812.

[30] Ohlsson, M. ja Alberg, J. Predicting Fault-Prone Software Modules in Telephone Switches. IEEE Transactions on Software Engineering, 1996. Vol.22:12. S. 886–894. ISSN 0098-5589.

[31] Ostrand, T. J. ja Weyuker, E. J. The Distribution of Faults in a Large Industrial Software System. Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis, 2002. Vol.27:4. S. 55–65. ISSN 0163-5948.

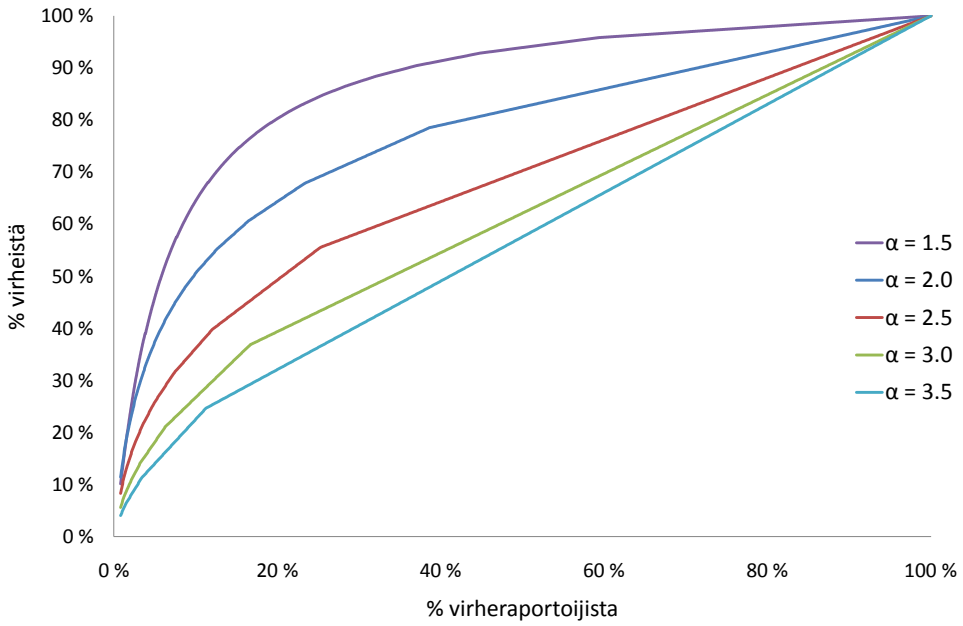
[32] Rodgers, J. L. ja Nicewander, W. A. Thirteen Ways to Look at the Correlation Coefficient. American Statistician, 1988. Vol. 42:1. S. 59–66. ISSN 0003-1305.

[33] Seal, H. L. The Maximum Likelihood Fitting of the Discrete Pareto Law. Journal of the Institute of Actuaries, 1952. Vol.78. S. 115–121. ISSN 0020-2681.

[34] Vuong, Q. H. Likelihood Ratio Tests for Model Selection and Non-nested Hypotheses, 1989. Vol.57:2. S. 307–333. ISSN 0012-9682.

A Pareto-jakauman Albergin diagrammeja sekä numeeriset tulokset jakaumien sovituksesta

Liitteessä esitetään Pareto-jakauman Albergin diagrammeja eri parametrien α arvoilla (kuva (A1)). Lisäksi esitetään Pareto-jakauman sovituksesta saatuja numeerisia tuloksia (taulukot (A1) ja (A2)).



Kuva A1: Pareto-jakauman Albergin diagrammeja eri parametrien α arvoilla.

Taulukko A1: Empiiristen jakaumien vertailu Pareto-jakauman ja muiden teoreettisten jakaumien kanssa. Positiiviset normalisoidun log-uskottavuusosamäärän arvot viittaavat siihen, että Pareto-jakauma sopii empiiriseen jakaumaan vertailtujakaumaa paremmin. Tilastollisesti merkitsevät p-arvot on lihavoitu. Viimeisessä sarakkeessa esitetään tehty johtopäätös. Mahdollisia vaihtoehtoja johtopäätöksiksi on neljä: ”huono” tarkoittaa, ettei aineisto todennäköisesti ole Pareto-jakautunut; ”kohtuullinen” tarkoittaa, että Pareto-jakauma sopii empiiriseen jakaumaan hyvin, mutta on myös muita uskottavia vaihtoehtoja; ”hyvä” viittaa siihen, että Pareto-jakauma sopii empiiriseen jakaumaan hyvin eikä mikään muu jakauma ole uskottava vaihtoehto; ”katkaistu” viittaa siihen, että Pareto-eksponenttijakauma on selkeästi puhdasta Pareto-jakaumaa parempi malli, mutta jotkin muutkin jakaumat saattavat olla uskottavia vaihtoehtoja.

	Pareto		eksponentti		lognormaali		Poisson		Weibull		Yule		Pareto-eksponentti		Tuki Pareto-
Aineisto	$\hat{\alpha}$	p	UO	p	UO	p	UO	p	UO	p	UO	p	UO	p	jakaumalle
Apache	2.97	**0.81**	3.83	**0.00**	-0.74	0.46	3.70	**0.00**	11.34	**0.00**	1.77	**0.08**	-0.67	0.25	kohtuullinen
Linux	2.55	**0.12**	2.60	**0.01**	-0.46	0.64	1.65	**0.10**	12.38	**0.00**	4.90	**0.00**	0.00	1.00	kohtuullinen
Mozilla	2.73	0.00	9.19	**0.00**	18.20	**0.00**	6.09	**0.00**	18.61	**0.00**	14.54	**0.00**	0.00	1.00	huono
Yritys A	1.41	0.00	-1.79	**0.07**	-3.78	**0.00**	5.02	**0.00**	-3.80	**0.00**	-8.27	**0.00**	-20.79	**0.00**	katkaistu
Yritys B	1.36	0.00	-1.86	**0.06**	-2.78	**0.01**	3.28	**0.00**	-2.86	**0.07**	-5.83	**0.00**	-15.85	**0.00**	katkaistu
Yritys C	1.31	0.00	-3.42	**0.00**	-4.78	**0.00**	4.98	**0.00**	-5.06	**0.00**	-10.80	**0.00**	-23.35	**0.00**	katkaistu

Taulukko A2: Empiiristen jakaumien häntien vertailu Pareto-jakauman ja muiden teoreettisten jakaumien kanssa. Positiiviset normalisoidun log-uskottavuusosamäärän arvot viittaavat siihen, että Pareto-jakauma sopii empiiriseen jakaumaan vertailujakaumaa paremmin. Tilastollisesti merkitsevät p-arvot on lihavoitu. Viimeisessä sarakkeessa esitetään tehty johtopäätös.

	Pareto			eksponentti		lognormaali		Poisson		Weibull		Yule		Pareto-eksponentti		Tuki Pareto-
Aineisto	x_{min}	$\hat{\alpha}$	p	UO	p	UO	p	UO	p	UO	p	UO	p	UO	p	jakaumalle
Apache	1	2.97	**0.81**	3.83	**0.00**	-0.74	0.46	3.70	**0.00**	11.34	**0.00**	1.77	**0.08**	-0.67	0.25	kohtuullinen
Linux	2	2.47	**0.70**	2.18	**0.03**	-0.11	0.91	1.42	0.16	3.20	**0.00**	1.46	0.14	0.00	1.00	kohtuullinen
Mozilla	3	2.27	**0.44**	5.55	**0.00**	-1.18	0.24	4.25	**0.00**	1.91	**0.06**	1.09	0.28	-0.54	0.30	kohtuullinen
Yritys A	12	2.42	**0.11**	-0.67	0.50	-1.09	0.28	3.50	**0.00**	0.46	0.65	-2.06	**0.04**	-1.86	**0.05**	katkaistu
Yritys B	26	3.01	**0.65**	-0.17	0.87	-0.49	0.62	1.94	**0.05**	-0.52	0.60	-0.87	0.38	-0.32	0,42	kohtuullinen
Yritys C	15	1.96	0.03	-1.30	0.19	-1.31	0.19	4.15	**0.00**	-1.40	0.16	-2.56	**0.01**	-3.48	**0.01**	katkaistu