

LAPPEENRANNAN-LAHDEN TEKNILLINEN YLIOPISTO

School of Engineering Science

Tietotekniikan koulutusohjelma

Kandidaatintyö

Ilja Kotirinta

Menestyksen avaintekijät avoimen lähdekoodin projekteissa

Tarkastaja: Tutkijaopettaja Jouni Ikonen

Ohjaaja: Tutkijaopettaja Jouni Ikonen

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT

School of Engineering Science

Tietotekniikan koulutusohjelma

Ilja Kotirinta

Menestyksen avaintekijät avoimen lähdekoodin projekteissa

Kandidaatintyö 2020

41 sivua, 5 kuvaa, 8 taulukkoa

Ohjaaja: Tutkijaopettaja Jouni Ikonen

Avainsanat: avoimen lähdekoodin ohjelmisto, projekti, menestyksen tekijät

Tässä tutkimuksessa pyritään löytämään menestyksen mittareita avoimen lähdekoodin projekteista sekä tavan, jolla menestykseen johtaneet syyt voidaan identifikoita. Tutkimusmetodina käytetään systemaattista kirjallisuuskatsausta, jossa muodostetaan yleiskäsitys siitä miten menestys on määritelty ja mitattu aikaisemmin avoimen lähdekoodin projekteissa. Kirjallisuuskatsauksen tavoitteena on luoda kuva millaisia mittareita voidaan käyttää menestyksen mittaamiseen. Tutkimuksen tuloksena sovelletaan 18 menestyksen mittaria, jotka ovat saatu systemaattisen kirjallisuuskatsauksen tuloksena, kahteen avoimen lähdekoodi projektiin. Yhteenvedossa kerrotaan tutkimuksen sovellettavan vaiheen havainnot ja haasteet sekä annetaan ohjeet tutkimuksen kehittämismahdollisuuksista.

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Degree Programme in Software Engineering

Ilja Kotirinta

Success factors in open source software projects

Bachelor's Thesis 2020

41 pages, 5 figures, 8 tables

Examiner: Associate professor Jouni Ikonen

Keywords: open source software, project, success factors

This research aims to advance understanding what are success factors in open source software projects and how they can be measured. The main research method is a literature study which is used to build a view how the success has been measured previously in different projects and what kind of metrics have been used. As the result of the study we apply 18 success factors to two different open source software projects and discuss our observations and challenges faced during that process. Lastly we discuss challenges and observations made during this research and give guidelines on how to improve and continue our research.

ALKUSANAT

Haluan kiittää itseäni kandidaatintyön kirjoittamisesta sekä Ikosen Jounia kaikesta avusta ja tuen määrästä.

Sisällys

| | |
|---|-----------|
| 1 Johdanto | 4 |
| 1.1 Tausta | 4 |
| 1.2 Avoin lähdekoodi | 4 |
| 1.3 Tutkimuskysymykset | 5 |
| 1.4 Tutkimusongelma, tavoitteet ja rajaus | 6 |
| 1.5 Tutkimusjärjestelyt | 6 |
| | |
| 2 Systemaattinen kirjallisuuskatsaus | 7 |
| 2.1 Tiedonhaku ja valintaperusteet | 7 |
| 2.2 Avoimen lähdekoodin projekti ja menestys | 9 |
| 2.3 Kirjallisuuskatsauksen tulosten kartoitus | 10 |
| 2.3.1 DeLone-McLeanin malli | 10 |
| 2.3.2 GitHub | 11 |
| 2.3.3 Käyttäjien tyytyväisyys | 13 |
| 2.3.4 Yhteisö | 13 |
| 2.3.5 Kehittäjät | 14 |
| 2.3.6 Lataukset | 15 |
| 2.3.7 Konkreettiset rajaukset | 18 |
| 2.3.8 Kansainvälinen standardi mittarina | 20 |
| | |
| 3 Menestyksen mittareiden soveltaminen avoimen lähde-koodin projekteihin | 21 |
| 3.1 Valitut projektit | 21 |
| 3.2 Menestyksen mittarit Apache projektissa | 23 |
| 3.3 Menestyksen mittarit VueJS projektissa | 26 |

| | |
|---------------------------------|-----------|
| 4 Pohdinta ja yhteenveto | 29 |
| Lähteet | 32 |

Symboli- ja lyhenneluettelo

| | |
|-------------|--|
| Open source | avoin lähdekoodi |
| OSS | Open Source Software, avoimen lähdekoodin ohjelmisto |
| SaaS | Software as a Service |
| MIT | Massachusetts Institute of Technology |
| OSI | Open Source Initiative |
| PD | Public Domain |

1 Johdanto

1.1 Tausta

Avoimen lähdekoodin ohjelmistoon liittyvät hyödyt ovat sen ympärillä oleva yhteisö, ohjelmiston käytön tuomat taloudelliset säästöt sekä lähdekoodin saatavuus [5]. Vuonna 2019 The Linux Foundationin tehdyn kyselyn mukaan 67 prosenttia kaikista kyselyyn vastanneista yrityksistä käyttivät avoimen lähdekoodin ohjelmia sisäisesti tai ei kaupallisesti ja 50 prosenttia käytti niitä kaupallisessa tarkoituksessa [16]. Tunnetut avoimen lähdekoodin projektit ovat GNU/Linux (käyttöjärjestelmä), Git (hajautettu versionhallintajärjestelmä) ja MySQL (realaatiotietokantaohjelmisto). Monet kaupalliset ohjelmat käyttävät avoimen lähdekoodin ohjelmistoja omissa tuotteissaan esimerkiksi Instagram [20] ja jotkut taas tekevät osasta omaa tuotetta avoimen, kuten Spotify [47]. Avoimen lähdekoodin yhteisön keulahahmo Eric Raymond toteaa esseessään *The Cathedral and the Bazaar* [43], että ”Avoimen lähdekoodin projektissa koodi on kirjoitettu huolellisesti ja innovatiivisesti, koska ohjelmistokehittäjät työskentelevät vain itselleen kiinnostavien asioiden parissa”. Raymondin toteamuksesta huolimatta, Karl Beecherin ja hänen ryhmänsä [28] tekemän tutkimuksen mukaan todellisudessa enemmistö avoimen lähdekoodin projekteista epäonnistuvat. Miten voidaan taata, että avoimen lähdekoodin projekti ei ole esimerkki epäonnistumisesta?

1.2 Avoin lähdekoodi

Open Source Initiative (OSI) määritelmän mukaan avoimen lähdekoodin ohjelmisto on kaikkien vapaasti saatavissa, käytettävissä, muutettavissa ja jaettavissa [37]. Yksi OSI:n perustajista ja avoimen lähdekoodin manifestin kirjoittaja Bruce Perens [41] toteaa: ”Jotta ohjelmistokehittäjät kontribuoisivat avoimen lähdekoodin projekteihin, koodin on oltava luvallista kopioida, levittää, sekä parantaa.”. Red Hat, joka on yksi ohjelmistoteollisuuden suurimmista yrityksistä, määrittelee avoimen lähdekoodin ohjelmiston seuraavasti: ”Ohjelmisto, joka on julkaistu erityisen lisenssin alla. Lisenssi antaa loppukäyttäjälle laillisen mahdollisuuden nähdä ohjelman lähdekoodin. Ohjelma on saatavilla ilman lisämaksuja, mikä tarkoittaa, että käyttäjä näkee lähdekoodin ja voi muokata sitä haluamallaan tavalla. Lähdekoodia voi käyttää muissa ohjelmissa, joten kuka tahansa voi käyttää ja levittää sitä. Tarkemmat rajoitukset ja käytänteet kuitenkin riippuvat lisenssistä, jonka alaisuuteen ohjelmisto kuuluu.”[44].

Suomessa valtion tasolla on kiinnostusta avoimen lähdekoodin ohjelmistostosta, sillä Suo-

men julkisen hallinnon tietohallinnon neuvottelukunta on antanut ohjeistuksen avoimen lähdekoodin ohjelmiston käytöstä julkisella sektorilla [25]. Ohjeistuksen tarkoituksena on kertoa avoimen lähdekoodin ohjelmistoihin liittyvistä hyödyistä ja haitoista sekä opastaa niiden käyttöönottoon ja ylläpitoon.

On olemassa useita kymmeniä avoimen lähdekoodin lisenssejä ja jotkut niistä antavat enemmän oikeuksia kuin toiset. Tärkeä huomio on, että avoimen lähdekoodin lisenssi on nimenomaan tekijäoikeuksia suojaava, joten se ei ole sama asia kuin PD (Public Domain), jossa teoksen tekijä on luopunut tekijänoikeuksistaan [48]. Berkeley Software Distribution (BSD) lisenssi on yksi avoimen lähdekoodin lisensseistä, joka antaa eniten vapauksia sitoamatta käyttäjää saman lisenssin jatkuvaan käyttöön tai muihin rajoituksiin. Tämän lisäksi BSD lisenssi on yksiselitteinen ja helposti ymmärrettävä. Lisenssi on seuraavanlainen: 1. Älä väitä, että olet kirjoittanut koodin itse 2. Älä syytä meitä (tekijöitä), jos koodi ei toimi 3. Älä käytä meitä (tekijöitä) tuotteesi markkinoinnissa [32]. Kun taas GNU GPL (GNU General Public License) lisenssi on esimerkki niin sanotusta käyttäjänoikeus lisenssistä, joka tarkoittaa, että sama lisenssi pysyy tuotteessa kokoajan, myös tuotteen muokkaamisen jälkeen. Tämän tarkoituksena on taata kaikille samat vapaudet ohjelman muokkaamiseen, kopiointiin ja levittämiseen [12]. Vuonna 2015 GitHub palvelussa oli lisensoitu noin 20 prosenttia kaikista projekteista. Suosituin lisenssi oli MIT (Massachusetts Institute of Technology) lisenssejä, jota käytti 44.69 prosenttia projekteista [4]. MIT lisenssi antaa käyttäjälle oikeuden kopioida, käyttää, myydä, muokata ja jälleen lisensoida ohjelmistoa [19]. Jälleen lisensointi liittyy aiemmin mainittuun käyttäjänoikeus lisenssiin, mutta tässä tapauksessa alkuperäinen lisenssi ei ole sidottu ohjelmistoon vaan lisenssin vaihtaminen on sallittua.

1.3 Tutkimuskysymykset

Tämän tutkimuksen aloittamiseen on kaksi tärkeää syytä. Ensimmäinen on kirjoittajan oma halu olla osallisena Suomen avoimen lähdekoodin yhteisön toimintaan ja tuottaa suomenkielistä tutkimusta avoimeen lähdekoodiin liittyen. Toinen syy, on löytää avaintekijät menestykseen avoimen lähdekoodin projekteista sekä menetelmät, joilla ne voidaan mitata.

Tutkimuskysymykset, joihin tämä työ pyrkii löytämään vastauksen ovat:

1. Millä tavalla voidaan mitata menestyksen avaintekijät avoimen lähdekoodin projekteissa?
2. Miten menestyksen indikaattorit soveltuvat käytännössä?

1.4 Tutkimusongelma, tavoitteet ja rajaus

Tässä luvussa esitellään tutkimusmenetelmät ja tutkimuksen ongelmat. Tämän jälkeen käydään läpi tutkimuksen tavoitteet ja rajauksen. Tutkimusmenetelmä käytämme järjestelmällistä kirjallisuuskatsausta, jossa etsitään ja tarkastellaan olemassa olevia tutkimuksia avoimen lähdekoodi projektien menestyksestä.

Kirjallisuuskatsauksessa käytettävien tutkimuksen otos valitaan käsin. Ongelmana suuren otoksen saavuttamisessa on saadun datan filtriointi ja varmistus siitä, että se on tutkimuskelpoista. Tutkimuksen tavoitteena on vastata tutkimuskysymyksiin ja tuottaa laadukasta sekä tieteellistä materiaalia.

Tuloksena saatuja indikaattoreita voidaan käyttää projektin hallintaan ja ohjaamiseen oikeaan suuntaan sekä epäonnistumisen riskien ennaltaehkäisemiseen. Myös omassa projektissa avointa lähdekoodia käyttöön ottava voi tehdä valinnan eri projektien välillä perustuen niiden menestykseen, jonka tulisi takaa vähintään haavoittuvuuksien korjaamista ja ohjelmiston ylläpitoa.

Meidän tutkimuksen käytännön osaan valittujen projektien tulisi selvästi näyttää pyrkimystään menestykseen. Projektin täytyy olla saatavilla versiohallinnan lisäksi vähintään yhdestä paketinhallintaohjelmasta ja olla käytössä yrityksessä tai jonkun yliopiston kurssilla. Yrityskäyttö kertoo projektin kypsyudesta ja vakaudesta mikä taas tarkoittaa, että sitä voidaan käyttää tuotannossa. Käyttö yliopiston kurssilla kertoo, että tiedemaailma on kiinnostunut projektista ja haluaa panostaa omalta osalta sen edistämiseen. Projektilla täytyy olla dokumentaatio ja kontribuutio ohjeet. Valitut kriteerit kertovat siitä, että projekti halutaan antaa käytettäväksi ja sitä halutaan kehittää.

1.5 Tutkimusjärjestelyt

Tutkimuksen alussa määritellään mikä on avoin lähdekoodi projekti ja mikä on sen menestys. Luvussa 2 suoritetaan systemaattinen kirjallisuuskatsaus, jonka tulokset tarkastellaan luvussa 2.3. Tämän luvun jälkeen lukijalla on ymmärrys siitä, millaiset tekijät vaikuttavat menestykseen avoimen lähdekoodin projektissa ja millä tavalla niitä voidaan mitata. Luvussa 3 valitaan kaksi menestynyttä avoimen lähdekoodin projektia, esitellään ne ja sovelletaan luvun 2.3 menestyksen mittareita projekteihin. Lopuksi pohditaan miten luvun 2.3 mittarit vastasivat tai eivät vastanneet luvussa 3 käsiteltyjä projekteja.

2 Systemaattinen kirjallisuuskatsaus

Kirjallisuuskatsaus on tieteellisen tutkimuksen metodi, jossa tiivistetään aiempien tutkimusten sisältö, kuten tutkimuskysymys ja -tulos. Systemaattisessa kirjallisuuskatsauksessa etsitään ja tarkastella olemassa olevia alan tutkimuksia sekä niiden tuloksia. Metodin tarkoituksena on antaa yleiskuva tutkimusallalla olevasta kirjallisuuden määrästä ja laadusta.[42] Kirjallisuuskatsauksen on myös täytettävä tieteellisen yhteisön asettamat vaatimukset, jotka ovat sen julkisuus, objektiivisuus ja itsekorjautuvuus. Objektiivisuudella tarkoitetaan, että tutkimus ei ole kirjoittajan mielipidettä eikä päätökset ole tehty intuiitioon nojaten. Itsekorjautuvuus taas merkitsee mahdollisuuden korjata ja poistaa tutkimuksessa olevat puutteiden sekä virheet uusilla tutkimuksilla. [31]

2.1 Tiedonhaku ja valintaperusteet

Systemaattisessa kirjallisuuskatsauksessa on tarkoitus kerätä aineistoa aikaisemmin tehdyistä tutkimuksista ja löydetyistä menestyksen mittareista. Näiden avulla pyritään hahmoittamaan nykyinen tilanne avoimen lähdekoodin projektien tunnettavuudesta ja antaa lukijalle ymmärryksen millaisilla tavoilla avoimen lähdekoodin projektien menestystä on mitattu aiemmin. Tiedonhaku varten määritellään valinta- ja hylkämiskriteerit sekä käytetään seuraavia tietokantoja:

1. ACM (Association for Computing Machinery)
2. IEEE (Institute of Electrical and Electronics Engineers)
3. Google Scholar

Tarvittaessa lisämateriaalia etsitään LUT Finna palvelusta. Tiedonhaun aikana luetaan ja arvioidaan löydetyt julkaisut sekä ollaan lähdekriittisiä koko tutkimuksen ajan. Näin tutkimus täyttää akateemisen toiminnan edellytykset. Taulukko 1 esittää valintaperusteet, joita noudetaan aineiston valinnassa ja hylkäämisessä.

| Hyväksytään | Hylätään |
|---|-----------------------------|
| Tieteellinen julkaisu | Ei selkeitä tuloksia |
| Työssä viitataan tieteellisiin lähteisiin | Saatavilla vain tiivistelmä |
| Työhön on viitattu ainakin kahdesti | |
| Julkaisu tutkii avointa lähdekoodi projektia tai sen menestystä | |

Taulukko 1. Aineiston valintaperusteet.

Haku suoritettiin seuraavalla lauseella ("open source"AND "project"AND "success") ja se tuotti tulokset tietokannoittain ACM 46 089; IEEE 219; Google Scholar 247 000. Tuloksia oli liikaa, joten ACM ja IEEE hakua rajoitettiin ("open source"AND title:"project"AND "success"), jolloin sanan "project" on pakko löytyä otsikosta ja loput hakusanat muualta metadatatista. Tällöin tuloksia saatiin ACM 264 ; IEEE 55; Google Scholar 74 000. Taulukossa 2 on esitetty ensimmäisen haun tulokset tietokannoittain. Toisen haun tulokset ja lopullinen valinta ovat taulukossa 3, jossa esitetään tutkimukseen valitun aineiston määrä, läpikäytyjen tulosten määrä sekä tietokannan tulokset yhteensä.

| Tietokanta | Löydetyt julkaisut |
|----------------|--------------------|
| ACM | 46 089 |
| IEEE | 219 |
| Google Scholar | 247 000 |

Taulukko 2. Tietokanta ja hakutulokset ensimmäinen haku.

| Tietokanta | Tulokset (Valitut/tarkastetut/kaikki) |
|----------------|---------------------------------------|
| ACM | 11/264/264 |
| IEEE | 5/55/55 |
| Google Scholar | 12/100/74 000 |

Taulukko 3. Tietokanta ja hakutulokset toinen haku.

Valituista tutkimuksista 14 olivat empiirisiä eli niissä tutkittiin olemassa olevien projektien menestystä. Huomattava osa aineistosta, joka oli muuten sopivaa, jouduttiin hylkäämään sillä

niistä oli saatavilla ainoastaan tiivistelmä. Haun aikana löytyi yksi Suomessa tehty tutkimus avoimen lähdekoodi projektin menestyksestä. Muita huomion arvoisia havaintoja ei ilmennyt.

2.2 Avoimen lähdekoodin projekti ja menestys

Avoimen lähdekoodi projektin menestystä on ollut tapana mitata kehittäjien, heidän kotiribuutioiden ja projektin käyttäjämäärän avulla [28]. Toinen suosittu mittari on latausten määrä [10]. Edellä mainituissa mittareissa on kuitenkin hyvät sekä huonot puolet. Mittareihin on tehtävä tarkennuksia, kuten riittävän käyttäjä- ja latausmäärän asettaminen, mutta silti ne eivät aina ole vertailukelpoisia keskenään.

Cambridge Dictionaryn mukaan projekti on suunniteltu työ, jolla on kesto ja päämäärä [7]. On olemassa kaksi yleistä tapaa valita ihmisiä projektiin. Ensimmäinen on rationaalisen prosessin malli, jossa projektin jäsenet valitaan käytännönsaamisen perusteella, kuten ongelmanratkaisukyvyyn tai työkokemuksen. Toinen malleista suosii psykologisia ominaisuuksia ja yhteistyökykyä, jossa tärkeimpänä tekijänä on ihmisten välinen vuorovaikutus sekä sosiaaliset taidot. [17] On tieteellistä näyttöä, että sosiaalisella tuella on tärkeä rooli ihmisen käyttäytymisessä (Thoits, 1984). Tästä johtuen hyvä ja kannustava ilmapiiri projektin sisällä saa ihmiset ylittämään itsensä arkipäivän työtehtävissä.

Tavalla millä ihmiset valitaan projektiin on merkitystä avoimen lähdekoodi projektin tapauksessa, sillä ihmiset jotka ovat kiinnostuneita samoista asioista tuntevat usein muita samasta aiheesta kiinnostuneita. Näin muodostuvat samanhenkisten ajattelijoiden piirit[1]. Avoimen lähdekoodin projektiin ei valita osallistujia vaan projekti on kaikille avoin. Projektiin liittyvät osallistujat sopivat molempiin edellä mainituihin malleihin. Avoimen lähdekoodin projekteilla ei ole samanlaista koulutusjaksoa uusille työntekijöille kuin yrityksissä, joten kontribuojien on usein hankittava tarvittava osaaminen muulta. Lakhanin tekemässä tutkimuksessa [27], jossa haastateltiin 684 kokenutta ohjelmistokehittäjää, suurimmaksi syyksi osallistua avoimen lähdekoodin projektiin paljastui hyvän olon ja onnistumisen tunteen saaminen. Muita syitä osallistumiseen olivat euforian tunne koodin kirjoittamisesta ja ohjelmointitaitojen kehittäminen. Avoimen lähdekoodin projektiin kontribuivat tekevät siis työtä ilmaiseksi ja itse valitsevat miten pitkään he osallistuvat projektiin sekä miten paljon he tekevät töitä sen eteen.

Fairley määrittelee menestyneen ohjelmistoprojektin seuraavasti: ”Ohjelmisto, joka ratkaisee käyttäjän ongelman, se on julkaistu ajallaa ja projekti on pysynyt budjetissa”[9]. Tässä työssä tutkitaan menestyksen mittareiden olemassaoloa ja sopivuutta avoimen lähdekoodi projek-

tiin. Palvelu SourceForge, jossa voi jakaa avoimen lähdekoodin projekteja, on ollut usean avoimeen lähdekoodiin liittyvän tutkimuksen kohteena. ¹ Foushee ja hänen ryhmänsä tutkimuksessa [11] todettiin, että 68 prosenttia kaikista SourceForgen projekteista ovat refleksiivisiä ja määrä on 89 prosenttia projekteissa, joita on ladattu yli miljardi kertaa. Reflektiivisyys tarkoittaa Eric Raymondin mukaan ”*Ohjelmistokehittäjältä toiselle ohjelmistokehittäjälle tehty*”. Samassa tutkimuksessa ehdotettiin menestyksen määrittelylle hetki jolloin projektin tuloksena saatava ohjelma ratkaisee sen kohderyhmän ongelman. Vaikka määritelmä kuulosta oikealta, sen mittaaminen ei ole helppoa ja vaatii palautetta jokaiselta ohjelman käyttäjältä.

Kaupalliset ohjelmistot taas mittaavat menestystä tuoton avulla. Koska tutkimuksessa tutkittiin SourceForgen projekteja, mittareiden piti löytyä myös sieltä. Potentiaalisiksi mittareiksi osoittautuivat julkaisujen tiheys, ohjelmistokehittäjien aktiivisuus, ohjelmiston laatu ja latausten määrä. Refleksiivisyyden mittamiseen käytettiin käyttäjäryhmä, ohjelman kategoria ja tuetut käyttöjärjestelmät. Latausmäärien huonoista puolista huolimatta ne valittiin mittaamaan menestystä. [11]

2.3 Kirjallisuuskatsauksen tulosten kartoitus

Avoimen lähdekoodin projektien menestystä on tutkittu ja mitattu aiemmin [24, 11, 30]. Avoimen lähdekoodin ohjelmisto on aiheena hyvin laaja ja monipuolinen, joten jokainen tutkimus on ollut arvokas ja tuonut uutta tietoa tiedeyhteisölle. Kuten johdannossa todettiin, avoimelle ohjelmistolle on kysyntää niin julkisella kuin yksityisellä sektorilla. Tulosten kartoituksessa käydään läpi kirjallisuuskatsauksen tulokset. Olemme ryhmitelleet tulokset niiden aihepiirien mukaan, jotta tarkastelu olisi selkeämpää, yhtenäisempää ja ryhmän nimen perusteella tiedetään aihe, jota se käsittelee.

2.3.1 DeLone-McLeanin malli

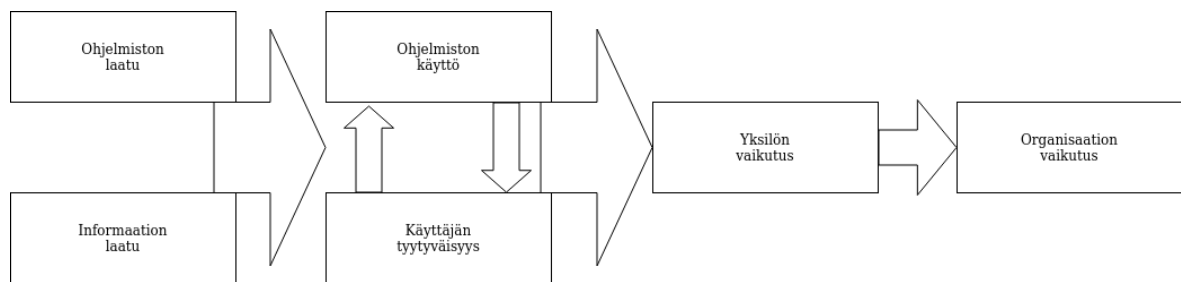
Tietojärjestelmän menestykseen on käytetty monesti DeLonen ja McLeanin kehittämää mallia ² Joka on esitetty kuvassa 1. Mallin mukaan ohjelmistoprojektin menestykseen vaikuttavat seuraavat tekijät: ohjelmiston laatu, dokumentaation laatu, ohjelmiston käyttö, käyttäjän

¹Association for Computing Machinery tietokanta antaa 991 tulosta hakusanalla SourceForge

²Tutkimukseen, jossa malli määritellään on viitattu yli 11 823 kertaa Google Scholar palvelun mukaan.

tyytyväisyys, yksilön vaikutus ja organisaation vaikutus [53]. Mallia on sovellettu avointa lähdekoodia käsittelevissä tutkimuksissa. Muun muassa Lee tutkimusryhmänsä kanssa [45] ovat tehneet mallista avoimen lähdekoodin ympäristöön sopivan version. Heidän versionsa oli samanlainen kuin alkuperäinen, mutta ilman informaation laatua. Tutkimuksessaan he haastattelivat GNU/Linux käyttöjärjestelmän käyttäjiä. He perustelevat informaation laadun poistamisen seuraavalla tavalla: ”Informaation laatu saattaa olla tärkeä aspekti avoimeen lähdekoodiin perustuvassa ohjelmassa, meidän tapauksessa (GNU/Linux käyttöjärjestelmä) ei ole suunniteltu tuottamaan mitään informaatiota, verrattuna tavalliseen tietojärjestelmään. Tämän takia poistamme informaation laadun DeLone-McLean mallista mittaaksen avoimen lähdekoodi ohjelmiston menestystä.”[45]

DeLone-McLean mallin soveltaminen ja muuttaminen tapauskohtaisesti on yksi mahdollisista tavoista mitata menestystä. Siinä tapauksessa kun alkuperäinen malli on laajasti käytetty, päivitetty ja todettu toimivaksi useissa tuhansissa tutkimuksissa.

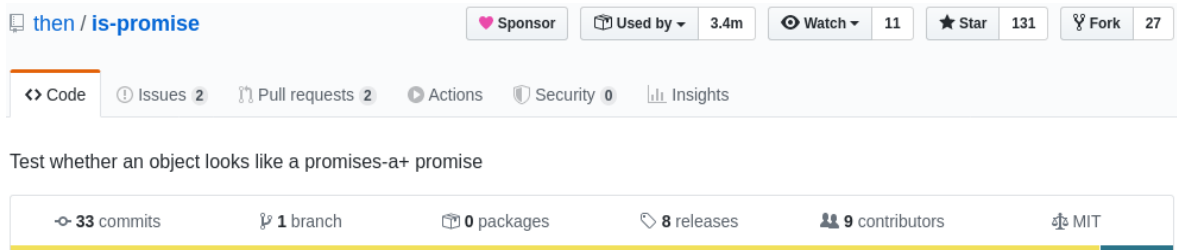


Kuva 1. DeLone-McLean IS(Information system, tietojärjestelmä) menestyksen malli.

2.3.2 GitHub

GitHub palvelussa käyttäjien tyytyväisyyttä indikoi projektin tähtien määrä. GitHub palvelun mukaan tähden antaminen on tapa kiittää ohjelman kehittäjiä [14]. Se ei kuitenkaan ole välttämätön toimenpide, joten tyytyväisten käyttäjien määrä voi todellisuudessa vaihdella suuresti. Ongelmaksi muodostuu myös tyytymättömien käyttäjien löytäminen, sillä heillä ei ole yhtä näkyvää tapaa ilmaista mieltään. Tyytymättömyyden ilmaiseminen sekä ongelmasta kertominen on usein mahdollista sähköpostilla tai suoraan GitHub palvelussa, luomalla uuden virheilmoituksen (issue ticket). Tämä ei kuitenkaan ole yhtä näkyvää kuin tähtien määrä, koska se ei ole yhtä yksiselitteinen ja projektin ylläpitäjä sulke virheilmoituksen ilman ratkaisua tai ratkaisun kanssa. Toinen ongelma tyytymättömien käyttäjien mittaamisessa on heidän olemassaolon epäileminen ylipäänsä, sillä projektien käyttö on täysin vapaaehtoista. Projektista pystyy haarukoimaan (fork) oman version ja muokkaamaan sitä halutulla tavalla, joten epämiellyttävien ominaisuuksien poistaminen on kiinni vain omasta osaamisestaan.

Kuva 2 esittää miten GitHubista saatavilla olevat indikaattorit *used by*, *watch*, *star* ja *fork*, eivät aina ole keskenään samassa suhteessa.



Kuva 2. is-promise projektia käytetään yli 3.4m muussa projektissa. Projektin tähtien määrä on kuitenkin vain murto-osa siitä.

Coelhon tutkimuksessa [6] analysoitiin GitHubissa olevia projekteja ja haastateltiin niiden kehittäjiä. Tutkimuksessa olivat mukana niin menestyneet kuin epämenestyneet projektit. Kaikki projektit liittyivät ohjelmistokehitykseen ja 81 prosenttia niistä oli kirjastoja tai kehyksiä. Tutkimuksen tuloksena oli menestykseen vaikuttavat tekijät sekä niiden vaikutuksen suuruus menestykseen. Pieni vaikutus oli lisenssillä, projektin kotisivulla ja virheilmoituksen pohjalla (issue template). Keskiverto vaikutus oli jatkuvalla integraatiolla ja suuri vaikutus oli kontribuutioilla. Jatkuva integraatio on uusien lähdekoodimuutoksien jatkuva yhdistäminen kehitettävään ohjelmistoon. Kontribuutio on toisen ihmisen tekemä työ projektin eteen, joka voi olla esimerkiksi logon tekeminen, dokumentaation kääntäminen tai ohjelmointi. Jokainen kontribuutio näkyy projektissa yksittäisenä tekona (commit). Projektin epäonnistuminen oli laajasti määritelty ja esimerkiksi yritysosto laskettiin epäonnistumiseksi, vaikka yritysosto voi olla merkittävä saavutus projektin menestyksen ja jatkon kannalta.

Projektien epäonnistumisten syyt: [6]

1. Kehittäjien ajan puute
2. Kehittäjien kiinnostuksen puute
3. Kilpailija ostaa projektin
4. Projekti on vanhentunut
5. Vanhanaikaiset teknologiat
6. Lailliset ongelmat
7. Kehittäjien väliset konfliktit

8. Vähäinen ylläpito

9. Yritys ostaa projektin

Eräässä toisessa tutkimuksessa Maqsood ja hänen ryhmänsä [33] tutkivat 5 000 avoimen lähdekoodin projektia. Heidän tutkimuksensa tuloksena oli, että kriteerit joilla on suuri vaikutus projektin menestykseen ovat *haarukoinnit* (forks) ja *katsojat* (watchers). Yhdenkin suurivaikutteisen kriteerin puuttuminen vaikuttaa vahvasti projektin lopputulokseen ja menestykseen. Kriteerit, joiden merkitys oli taas tutkimuksen muvaan vähäinen, ovat *työpanokset* (commits), *kehittäjät* (contributors), *oksat* (branches), *virheilmoitukset* (issues), *muutos ehdotukset* (pull requests) ja *julkaisut* (releases).

2.3.3 Käyttäjien tyytyväisyys

Crowston tutkimuksessaan [30] ehdotti yhdeksi tavaksi mitata käyttäjien tyytyväisyyttä integroimalla käyttäjätyytyväisyyskyselyn ohjelmistoon. Myös Garcia-Gathright tutkimusryhmänsä kanssa [13] ehdottivat käyttäjien tyytyväisyyden mittaamista. He esittivät tyytyväisyyden jakamista kahteen eri tapaan, määrällinen ja laadullinen. Määrällisessä tavassa on tarkoituksena kerätä mahdollisimman paljon dataa kuten klikkauksia ja ohjelman ominaisuuksien käyttökertoja. Tämä antaa yleiskuvan vallitsevista trendeistä ja käyttäjien käyttäytymistavoista. Laadullisessa tavassa taas tyytyväisyyden mittaaminen toteutetaan kyselyiden ja haastattelujen avulla. Sen tarkoituksena on selittää löydetty trendit ja käyttäytymistavat.

Muun muassa Netcraft palvelu tekee käyttäjäkyselyjä ja vuonna 2019 elokuussa nettipalvelimiin liittyvään kyselyyn vastasi 1 271 920 923 nettisivua. Siitä on mahdollista nähdä tarkkaan, että avoimen lähdekoodin nettipalvelinta Nginx käyttää 34.59 prosenttia kaikista nettisivuista [35]. Avoimen lähdekoodin käyttöjärjestelmässä Ubuntuissa on mahdollista jakaa automaattisesti omaa käyttödataa kehittäjille. Tämä taas herättää kysymyksen minkä datan kerääminen on välttämätöntä? Esimerkkinä käyttöjärjestelmä, tekstieditori ja kehys, joista jokainen voi olla avoimen lähdekoodin projekti, ovat keskenään erilaisia käyttötarkoitukseltaan. Edellä mainituille projekteille eri data on eriarvoista eikä sama data ole aina saatavilla kaikista projekteista.

2.3.4 Yhteisö

Avoimen lähdekoodin yhteisöt koostuvat yksilöistä ja organisaatioista, jotka erillisillä tavoilla kontribuovat avointa lähdekoodia tukeviin hankkerisiin, esimerkiksi ylläpitämällä pro-

jekteja tai etsimällä virheitä olemassa olevista projekteista. Näiden yhteisöjen hyvinvoinnilla on kriittinen rooli avoimen lähdekoodin projektin menestykseen, sillä avoimen lähdekoodin projekteissa tapahtuu todella paljon yhteistyötä ja kommunikaatiota, etenkin Internetin välityksellä [40]. Koska etäisyydet kehittäjien välillä ovat pitkät ja kommunikointi tapahtuu Internetin välityksellä, fyysinen tunteiden ilmaisu ei ole mahdollista.

Avoimen lähdekoodin projekti on usein yhden ihmisen aloittama. Vuonna 2010 kaikista SourceForgen projekteista yhden ihmisen ylläpitämiä oli 82.64 prosenttia [8]. Projektin julkaisun jälkeen on yleistä, että sen ylläpitäjä jakaa tietoa projektista alan foorumeilla, Redditissa tai kirjoittaa siitä artikkelin esimerkiksi dev.to palveluun. Avoimen lähdekoodin projektin julkaisemiseen, hallintaan ja ylläpitoon käytetään monesti verkossa olevaa version- ja projektinhallintaa tarjoavaa palvelua kuten GitLab tai GitHub.

Avoimen lähdekoodin projektissa, niin kuin missä tahansa projektissa, siihen osallistuvien ihmisten hyvä mieli edistää heidän oppimista, luovuutta ja tehokkuutta [22], jotka puolestaan vaikuttavat projektin edistymiseen. Tutkimuksessa [39] todettiin, että käyttäjien ja kehittäjien välillä sähköpostiketjuissa on niin positiivista kuin negatiivista henkeä. Lisätutkimuksia kuitenkin tarvitaan, jotta analyysi pystyttäisiin tekemään systemaattisesti. Toinen tutkimus [24] osoittaa, että sähköpostiketju oli suurimmalta osin neutraali, vaikka sisälsi kritiikkiä. Toinen tutkimuksessa esille noussut havainto oli toistuvat akronyymit ”LGTM”(näyttää hyvältä), ”nacked”(ei hyväksytty) ja ”nits”(pieniä ongelmia), joiden täsmentämiseen tarvittiin yhteisön apua. Molemmat edellä mainitut tutkimukset totesivat, että lisää tutkimusaineistoa ja tutkimusmetodeja tarvitaan, jotta voidaan varmistua tulosten luotettavuudesta.

Yhteisöä käsittelevässä luvussa tehtyjen havaintojen perusteella voimme todeta, että yhteisön roolilla on suuri vaikutus projektin elämänkaareissa ja sen menestyksessä. Kuitenkaan avoimen lähdekoodin projektit, joissa kehittäjät ovat sosiaalisesti aktiivisia eivät ole kaikista aktiivisimpia eikä menestyneimpiä [50].

2.3.5 Kehittäjät

Katsamakasen tekemä tutkimus [29], jossa simuloitiin malli empiirisesti kerätyn datan avulla avoimen lähdekoodin projektien elämästä, päättyi seuraaviin tuloksiin. Suurin osa projekteista olivat luonnostaan epävakaita ja kilpailivat ohjelmistokehittäjistä sekä käyttäjistä, joita on rajallinen määrä. Vapaaehtoisuuden takia monet kehittäjät vaihtoivat projekteja usein, tai tekivät töitä yhden projektin parissa niin kauan kunnes heille mieluisin työ oli tehty. Tämän takia projekteihin osallistujien määrä ja tulokset vaihtelevat paljon. Sisäisesti motivoituneet ohjelmoijat tekivät heitä kiinnostavan työn ja vaihtoivat projekteja kaikista eniten. Kun taas

ulkoisesti motivoituneet ohjelmoijat jättivät usein projektin kesken, saavutettuaan tarpeeksi työkokemusta.

Tästä herää kysymys, vaikuttaako kehittäjien korkea vaihtuvuus avoimen lähdekoodin projektin menestykseen? Kokeneiden ohjelmoijien mukana siirtyy valtaosa hiljaista tietoa ja osaamista, jolloin yhä useampi uusi ohjelmoija joutuu käyttämään aikaa perusasioihin tutustumiseen ja käytäntöjen opetteluun. Tässä korostuu kattavan dokumentaation luomisen ja ylläpitämisen merkitys avoimen lähdekoodin projektissa.

Rekka kerroin (truck factor) on ohjelmoijien vähimmäismäärä, joiden on jäätävä rekan alle ennen kuin projektin kehitystä ei pystytä jatkamaan. Toisin sanoen suurin osa projektin osallistujista tietää sen verran paljon, että projekti pystyy selviämään vähäisellä osallistujamäärällä[52]. GitHub palvelussa 133 suosituimman projektin rekka kerrointa tutkiva Avelino ja hänen ryhmänsä [3]. saivat tulokseksi, että rekka kerroin yksi oli 34 prosentissa projekteista ja rekka kerroin kaksi 30 prosentissa. Tästä voimme päätellä, että yli puolet GitHub palvelun 133 suosituimmista projekteista ovat riippuvaisia yhdestä tai kahdesta ihmisestä. Tämä ei tietenkään ole projektin kannalta hyvä asia, sillä kehittäjästä tulee yksittäinen epäonnistumisen kohta (SPOF). Emanuelin tutkimuksessa [8], jossa analysoitiin 133 892 avoimen lähdekoodin projektia, toteaa, että 94.11 prosenttia projekteista ovat yhden tai kahden ihmisen ylläpitämiä.

Goeminne ja Mens tutkivat Pareto periaateen esiintymistä avoimen lähdekoodin projektiin osallistuvien kehittäjien keskuudessa ja he päätyivät tulokseen, että Patero periaate on havaittavissa. Projekteissa, joita he tutkivat oli pieni, mutta todella aktiivinen joukko kehittäjiä ja paljon suurempi joukko ei niin aktiivisia kehittäjiä [15]. Tämä on Pareto periaatteen mukainen tulos, jossa suurin osa vaikutuksesta syntyy pienestä osasta vaikuttajia.

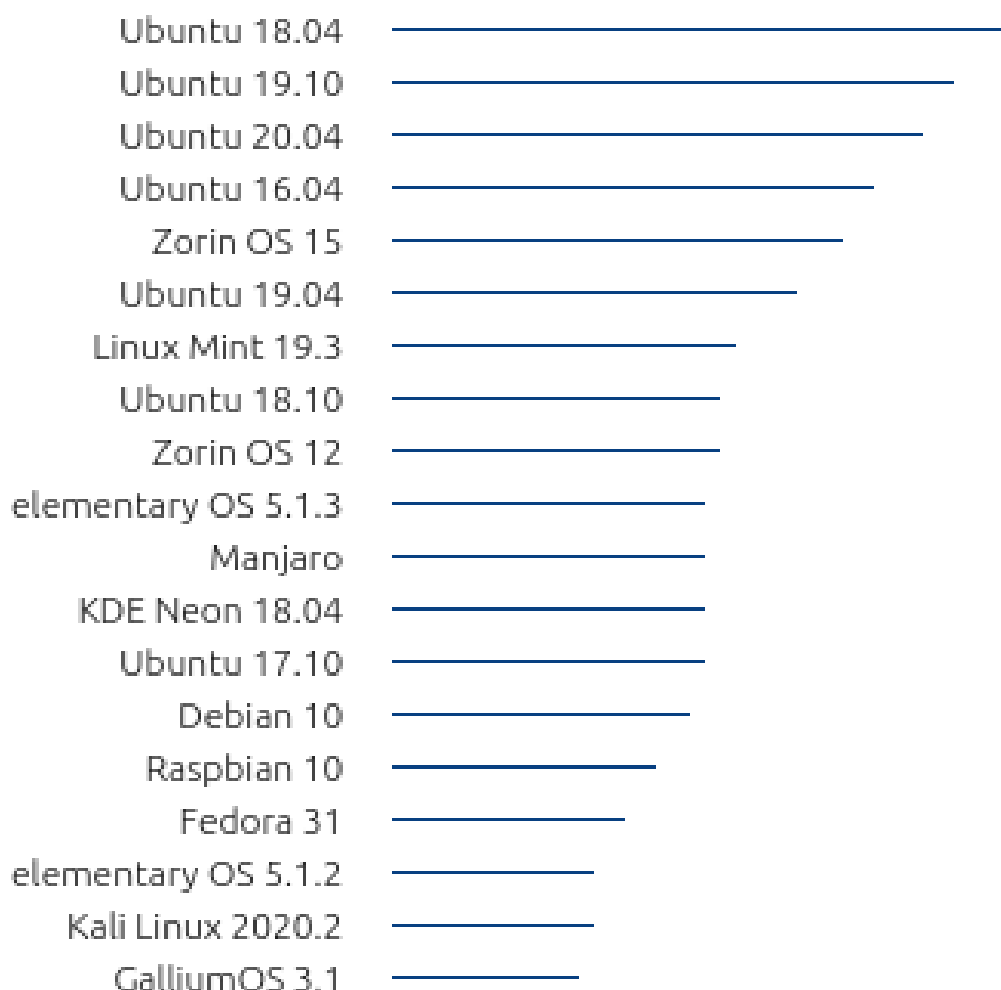
2.3.6 Lataukset

Latausmäärä oli suosion ja menestyksen mittari useammassa tutkimuksessa [18, 10, 8, 11]. Voimme todeta, että projekti, jota on ladattu miljoona kertaa on tietysti mittakaavassa suosittu. Entäs projekti, jolla on vain 100 latausta? Luku 100 on minun mielestä tarpeeksi iso, jotta voidaan sanoa projektin olevan suosittu. Tässä on kuitenkin iso ongelma, sillä 100 on minun oma mielipide eikä sille ole mitään perustelua, joten muut tutkijat päätyvät usein omiin lukuihin suosion rajan määrittämisessä.

Latausmäärän käyttö mittarina hankaloittaa varsinkin se, että ohjelmisto voidaan ladata usean kerran saman projektin toimesta, lataamisen tapoja on monia erilaisia ja koska avoimen

lähdekoodin lisenssi sallii ohjelmiston levittämisen niin kaikista latauksista ei saada tarkkaa lukua. Jos projekti on saatavilla GitHubista niin se voidaan ladata itselle jättämättä latauksesta mitään jälkeä. Kun taas SourceForge palvelussa, projektin latausten määrä kasvaa yhdellä, jokaisen latauksen myötä. Kuva 4 esittää SourceForge palvelun projektinäkömän, josta nähdään kuluvan viikon lataukset. Tyypillinen tapa ladata ohjelmia ohjelmointimaailmassa on pakettihallintaohjelman kautta ja riippuen pakettihallintaohjelmasta, latausten määrä on tai ei ole saatavilla. Kuva 3 esittää pakettihallintaohjelman Snapcraft antamat ohjelmakohtaiset lataukset.

Users by distribution (log)



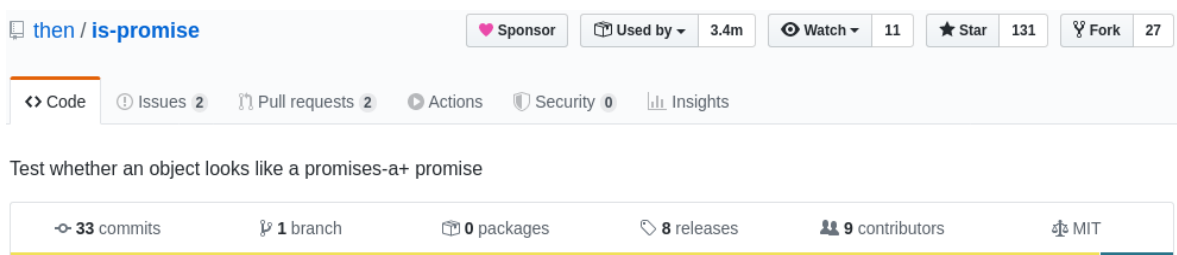
Kuva 3. Latausten määrän Snapcraft palvelussa on ilmaistu latausten suhteessa käyttöjärjestelmiittäin.



Kuva 4. Latausmäärä SourceForge palvelussa on tavallisesti viikko näkyssä. Myös tarkempi data latauksista on saatavilla.

Feitelson ja hänen ryhmänsä [10] analysoivat 122 205 projektia SourceForge palvelusta ja huomasivat, että Zipfin lain ja Pareto periaatteen mukaisesti 24.9 prosenttia projekteista vastasivat 98.5 prosentista kaikista latauksista. 75.1 prosenttia kaikista projekteista oli ladattu alle 1 680 kertaa per projekti, joten he päätyivät tulokseen, että menestystä edistävä latausten määrä on 1 680 ja yli. Emanuelin tutkimuksen mukaan [8] yli 1 000 latauksen saavuttaminen edistää projektin menestystä.

Vaihtoehtoisesti voidaan mitata ohjelmiston käyttöä sen perusteella monessa eri projektissa sitä käytetään. GitHub palvelussa on mahdollista seurata projektin käyttöä. Tällä hetkellä se on saatavilla vain tietyille kehyksille ja siihen liittyy ongelmia sekä rajoitteita. Käyttöjen määrällä GitHubissa tarkoitetaan projekteja, joissa kyseistä kehystä käytetään. Tämän lisäksi käytön suuruutta ei voida mitata, jotta se olisi vertailukelpoista. GitHub palvelun mukaan kehysten käyttö on yhtä arvokasta sitä käytettävän projektin koosta riippumatta. Eri kokoisissa projekteissa on eri määrä kehittäjiä ja loppukäyttäjiä, mikä tekee vertailusta hankalaa. Kuten voidaan huomata, käyttömäärän tulkitseminen ei ole niin yksinkertaista miltä se ensialkuun saattoi näyttää. Käyttökertoja voidaan kuitenkin soveltaa menestyksen mittaamisessa tapauskohtaisesti ja mahdollisesti muita indikaattoreita apuna käyttäen. Kuva 5 esittää kehystä, jota käytetään yli 3.4 miljoonaa projektissa ja samalla tähtiä on 131.



Kuva 5. Used by kertoo GitHub palvelussa monessa eri projektissa kyseistä projektia käytetään.

2.3.7 Konkreettiset rajaukset

Jokaisella läpi käydylle tutkimuksella oli oma näkemys menestyksestä ja miten menestykseen johtaneita syitä lähdetään tutkimaan. Selkeitä tuloksia ei kuitenkaan saatu aina. Seuraavaksi käydään läpi kirjallisuuskatsauksen ne tutkimukset, joiden tuloksena saatiin lista ohjeista, mittareista tai indikaattoreista, jotka vaikuttavat avoimen lähdekoodi projektin menestykseen..

Emanuel [8] ja hänen tutkimusryhmänsä kävivät läpi 84 prosenttia(134 549) kaikista Source-Forgen avoimen lähdekoodin projekteista. Tutkimuksessaan he analysoivat yli 27 parametria jokaisesta projektista. He päätyivät tutkimuksen tuloksena seuraaviin menestyksen kriteereihin:

1. Projektin käyttäjinä tavalliset ihmiset
2. Lähdekoodi on valmis käyttöä varten
3. Projekti toimii Linuxilla tai Windowsilla
4. Projektilla on ainakin yksi arvostelu
5. Projekti jaetaan zip tiedostona ja sen kokoluokka on megatavuissa
6. Projekti on saatavilla englanniksi tai muulla eurooppalaisella kielellä

Senyardin ja Michlmayrin tekemässä tutkimuksessa [46] kriteerit jaettiin välttämättömiin, tärkeisiin ja toivottuihin. Näiden täytyminen takaisi projektin menestyksen. Heidän tutkimuksen mukaan kriittinen piste avoimen lähdekoodin projektin kehityksessä on se, jolloin yhteisö aloittaa projektin aktiivisen kehittämisen. Tässä pisteessä erottuvat projektit, joiden metodin toimivat ja joiden metodin eivät toimi. Alla ovat tutkimuksen tuloksena saadut kriteerit:

Välttämättömät

1. Projektista on saatavilla lupaava prototyyppi
2. Projekti on suunniteltu modulaarisesti
3. Lähdekoodi on saatavilla ja se on käyttökelpoinen (kääntyy ja käynnistyy)
4. Yhteisö ja kehittäjät ovat kiinnostuneita projektista

5. Projektin luoja on motivoitunut hallinnoimaan sitä tai löytämään itselleen vastikkeen
6. Kommunikaatio ja kontribuutio ovat mahdollisia

Tärkeät

1. Projektin laajuus on määritelty
2. Projektilla on ohjelmointikäytännöt
3. Kehitysversioiden julkaisu on lyhytsyklistä ja käyttäjäversio on vakaa
4. Kehittäjiä houkutteleva lisenssi
5. Toimiva hallinto projektin sisällä

Toivotut

1. Projektista on saatavilla riittävä dokumentaatio

Crowstonin tekemä tutkimus [26] ehdottaa seuraavia mittareita menestyksen seuraamiseen:

1. Kehitysryhmän koko
2. Aika virheiden korjaamiseen
3. Latausten määrä
4. Aktiivisuuden taso SourceForgessa

Istiyanto ja Emanuelin tekemän tutkimuksen [23] datana käytettiin 5 000 eniten ladattua avoimen lähdekoodin projektia SourceForge palvelusta. Menestyksen määrittämiseen he käyttivät yli 10 indikaattoria, jotka olivat saatavilla projekteista. Indikaattoreina olivat muun muassa latausten määrä, ohjelmointikieli, projektin kategoria ja ohjelmistokehittäjien määrä.

Heidän tutkimuksen mukaan eniten projektin menestykseen vaikuttavat:

1. Projektin kasvava kehitys
2. Isompi suosio muihin projekteihin verrattuna

3. Projektiin osallistuvien jatkuva rekrytointi
4. Projektin ikä

Vähiten projektin menestykseen vaikuttavat:

1. Vain yhden lisenssin käyttö
2. Vain yhden käyttöjärjestelmän tuki
3. Vain yhden käyttöliittymän tuki
4. Vain yhden ohjelmointikielen käyttö

2.3.8 Kansainvälinen standardi mittarina

Tässä vaiheessa meidän tutkimusta voimme todeta, että monet mittareista ovat tulkinnanvaraisia ja ne voidaan ymmärtää eri tavoin eri ihmisten keskuudessa. Kyseinen ongelma ei rajoitu ainoastan meidän tutkimuksen piiriin. International Organization for Standardization, jonka standardeja ovat muun muassa parerikoot, kuten A4, ja päivämäärien muoto, on määritellyt standardin ohjelmiston laadun evaluointia varten. Standardi on nimeltään ISO/IEC 25010 [21] ja se on mahdollista soveltaa yhdeksi menestyksen mittariksi. On huomattavaa, että standardi ei ole tehty ainoastaan avoimen lähdekoodin ohjelmistoa varten vaan ohjelmistoja yleisesti. Laadukas ohjelmisto on kehittäjän ja käyttäjät yhteinen etu. ISO 25010 on jaettu kahdeksaan osaan, joista jokaiseen liittyy vielä tarkentavat kriteerit. ISO 25010 ohjelmiston laadun standardin osat ovat:

1. Toiminnallinen soveltuvuus
2. Suorituskyky
3. Yhteensopivuus
4. Käytettävyys
5. Luotettavuus
6. Turvallisuus
7. Ylläpidettävyys
8. Siirrettävyys

3 Menestyksen mittareiden soveltaminen avoimen lähdekoodin projekteihin

Tässä luvussa tutkitaan kahden avoimen lähdekoodin projektin menestystä ja sovelletaan kirjallisuuskatsauksen tuloksena saatuja menestyksen mittareita. Aluksi esitellään projektit ja vahvistetaan niiden menestys kolmella menestyksen indikaattorilla. Projektien esittelyssä käytetään apuna taulukkoa, jossa on projektien perustiedot vertailukelpoisessa muodossa. Tämän jälkeen esitellään kaksi tutkimusta, joiden menestyksen indikaattoreita tullaan soveltamaan. Kappaleen tuloksena saamme kaksi taulukkoa kummastakin projektista, joissa on menestyksen mittari ja sen löydettävyys projektista. Ennen jokaista taulukkoa on sen esittelykappale, jossa muun muassa keskustelemme sen käyttämisestä menestyksen mittareista ja projektista itsestään. Tämän luvun tarkoitus on soveltaa tieteellisen tutkimusmenetelmän tuloksia käytännön projekteihin.

Suurin osa kirjallisuuskatsauksen tutkimuksista päätyivät tuloksessaan indikaattoreihin, jotka eivät olleet löydettävissä projekteista, joita käsittelemme tässä luvussa. Osa indikaattoreista kuten yhteisön hyvinvointi ja projektin taloudellinen tilanne eivät olleet helposti mitattavissa, jonka takia emme valinneet kyseisiä indikaattoreita sovellettavaksi tässä luvussa. Toinen havainto, jonka teimme on, että osa tutkimuksista antoi tulokseksi menestyksen mittareita, mutta ei niiden indikaattoreita. Indikaattoreiden löytäminen mittareille olisi taas ollut laaja tutkimuksen kohde, joka on työmme rajojen ulkopuolella. Tässä luvussa päädyimme tarkastelemaan Emanuelin ja hänen tutkimusryhmänsä [8] sekä Senyard ja Michlmayer [46] tutkimuksia ja tuloksena saatuja mittareita.

3.1 Valitut projektit

Tarkasteltavaksi valitsimme Apache ja VueJS projektit. Valitsemamme projektit ovat menestyneitä ja suosittuja muun muassa vastaavien projektien keskuudessa sekä yhteisöjensä kanssa ja aktiivisuudessa. Esittelemme projektit ja perustelut niiden valintaan myöhemmin tässä luvussa. Muut syyt, jotka vaikuttivat valintaan ovat seuraavat: projektien ensimmäisen julkaisun välissä on 19 vuotta, joka antaa mahdollisuuden eri aikakausien tarkasteluun, projektit ovat ekvivalenttiansa keskuudessa laajasti käytettyjä ja projektit ovat suosion eri vaiheissa. Tiedon etsimiseen käytimme projektien nettisivuja sekä sieltä löytyviä lähteitä.

Apache on yksi suosituimmista palvelimista [34] ja tällä hetkellä sitä kehittää Apache Software Foundation. Huhtikuussa 2020 Apache palveli 308 143 708 nettisivua [34]. Sadat ihmiset

ovat osallistuneet tämän projektin kehitykseen [2]. Esimerkkinä merkittävistä kontribuoisista ovat Henry Spencer, joka on regex kirjaston kehittäjä ja Rob Hartill, joka loi suosittun elokuvatietokannan IMDb hän on myös yksi Apache projektin perustajista.

Henry Spencer, Rob Hartill ja Taylor Otwell ovat suosittujen ja menestyneiden avoimen lähdekoodin projektien luojia. Se, että he ovat kontribuoineet valitsemiimme projekteihin osoittaa heidän kiinnostuksesta niitä kohtaan. Meille se kertoo, että arvostetut ohjelmistokehittäjät ovat osallistuneet näiden projektien kehitykseen. Me voidaan päätellä, että he uskovat näiden projektien ideaan ja niiden menestykseen. Tälle päätelmälle ei kuitenkaan anneta liikaa painoarvoa, sillä yksi menestys ei johda vääjäämättä toiseen. On kuitenkin selvää, että kun kuuluisa henkilö kontribuoi projektiin tai mainitsee siitä, se tuo projektille näkyvyyttä.

VueJS on JavaScript kehys, jota ladataan viikoittain yli 1.5 miljoonaa kertaa. Näin iso latausten määrä kertoo projektin suosiosta [26, 8, 18, 10]. VueJS projektilla on oma kehitysryhmä, jossa on myös projektin perustaja Evan You. Projektilla on 293 aktiivista kontribuojaa, joista osa on isojen ohjelmistoyhtiöiden, kuten Netlify ja Netguru, työntekijöitä. Laravel kehysten perustaja Taylor Otwell on luonnehtinut VueJS kehystä aloittelijaystävälliseksi ja sen nettisivun olevan kaunis [38].

Taulukossa 4 esittelemme perustiedot molemmista projekteista kuten nimi, tyyppi ja käytössä oleva lisenssi. Taulukon tietojen perusteella voimme nähdä, että projektit ovat hyvin erilaiset toisiinsa nähden. Ne ovat julkaistu eri aikaan, kirjoitettu eri kielillä ja niillä on eri tarkoitus.

| | Apache | VueJS |
|---------------------------------|------------------------|-------------------|
| Ensijulkaisu | 1995 | 2013 |
| Ohjelmointikieli | C | JavaScript |
| Suosio | Laskemassa [34] | Nousemassa [36] |
| Versionhallinta | SVN | Git |
| Tyyppi | Palvelinohjelmisto | Kehys |
| Lisenssi | Apache 2.0 | MIT |
| Pääasiallinen kommunikointitapa | Sähköpostilista ja IRC | GitHub ja Discord |

Taulukko 4. Apache ja VueJS projektien perustiedot.

3.2 Menestyksen mittarit Apache projektissa

Taulukossa 5 tarkastelemme Emanuelin ja hänen tutkimusryhmänsä [8] löytämiä menestyksen mittareita. Heidän tutkimuksessaan tuloksena oli kuusi menestyksen mittaria, jotka tulisi löytyä avoimen lähdekoodin projektista, jotta sen menestyksen todennäköisyys kasvaisi. Apache projektista löysimme viisi mittaria. Mittari, jota emme löytäneet oli *”Projektin käyttäjinä tavalliset ihmiset”*. Apache projekti määrittelee sen käyttäjää Apache ohjelmiston käyttäjänä, jotka ovat ohjelmistokehittäjiä. Voidaan kuitenkin ajatella, että Apache projekti toimii pohjana projekteissa, joiden loppukäyttäjinä ovat tavalliset ihmiset, jolloin tämä mittari voidaan myös hyväksyä. Esimerkiksi verkkokauppa projektissa voidaan käyttää Apache ohjelmistoa, jolloin verkkokaupan loppukäyttäjistä tuleekin tietämättä Apache projektin käyttäjiä.

Apache projekti on ollut käytössä jo yli 25 vuotta. Linuxin ja Windowsin tuki Apache projektissa antaa sille mahdollisimman suuren käyttäjäryhmän. Projektista on kirjoitettu useita oppaita ja arvosteluja sekä sillä on paljon tähtiä GitHub palvelussa. Vaikka arvostelut ja tähdet eivät vaikuta projektin laatuun suoraan, ne antavat projektille palautetta ja lisäävät sen näkyvyyttä, jolloin potentialiset käyttäjät ja kontribuoijat löytävät projektin helpommin ja todennäköisemmin kokeilevat sen käyttöä. Apache on saatavilla zip tiedostoja ja sen voi asentaa käyttämällä useita paketin hallintaohjelmia. Apache projektin pääkieli on englanti ja yhtenä maailman puhutuimmista kielistä se antaa projektille suuren potentiaalisten käyttäjien määrän.

| Menestyksen mittari | Löytyy | Perustelut |
|---|--------|---|
| Projektin käyttäjinä tavalliset ihmiset | Ei | <i>Käyttäjä</i> on joku, joka käyttää meidän ohjelmaa |
| Lähdekoodi on valmis käyttöä varten | Kyllä | Projekti on ollut käytössä yli 25 vuotta |
| Projekti toimii Linuxilla tai Windowsilla | Kyllä | Toimii Linuxilla ja Windowsilla |
| Projektilla on ainakin yksi arvostelu | Kyllä | Projektista on kirjoitettu lukuisia arvosteluja, kirjoja sekä oppaita. Projektilla on GitHubissa 2 412 tähteä |
| Projekti jaetaan zip tiedostona ja sen kokoluokka on megatavuissa | Kyllä | Saatavilla zip ja tar.bz2 tiedostoina. Koko on 7MB |
| Projekti on saatavilla englanniksi tai muulla eurooppalaisella kielellä | Kyllä | Saatavilla kymmenellä eri kielellä, mukaan lukien englanti |

Taulukko 5. Emanuelin tutkimuksen menestyksenmittarit ja niiden olemassaolo Apache projektissa.

Taulukossa 6 käymme läpi Senyardin ja Michlmayerin [46] tutkimuksen löytämiä mittareita. Heidän tutkimuksen mukaan projektin on suoritettava tiettyjä toimenpiteitä, jotta se pystyy onnistuneesti toimimaan avoimen kehityksen ympäristössä. Apache projektin rakenne suunniteltu alusta asti modulaariseksi. Projekti koostuu toimivasta ytimestä ja lukuisista moduleista, joilla ytimen toimintaa voidaan laajentaa. Esimerkiksi Python serverin toimintaa varten on `mod_wsgi` ja käyttäjän autentikointiin käytetään `mod_auth` modulia. Apache projekti on avoimen lähdekoodin projekti, jota kuka tahansa voi käyttää, muokata ja jakaa. Projektin sivulla on näkyvillä lista kaikista aktiivisimmista ja ahkerimmista kontribuoista. Nykyään projektia hallinnoi ja kehittää Apache Foundation. Se kuvaa omaa toimintaa seuraavalla tavalla ”*Apache projektissa kaikki ovat tasa-arvoisia. Ihmiset voivat ilmaista mielipiteensä ja heidän näkemykset otetaan huomioon*”. Apache Foundationin yksi päätöksenteon kulmakivistä on ”Laiska Yhteisymmärrys”. Sen mukaan, mikäli kehittäjä on vakuuttunut siitä mitä yhteisö haluaa niin hän voi olettaa, että yhteisö on hänen kanssaan samaa mieltä ja ryhtyä töihin. Lopuksi Apache projektista on saatavilla valtavasti tietoa kuten ohjeita, päivityksessä tulleita muutoksia ja ominaisuuksien dokumentointi.

| Välttämättömät | Löytyy | Perustelut |
|---|---------------|---|
| Projektista on saatavilla lupaava prototyyppi | Kyllä | On ollut käytössä tuotantoympäristössä |
| Projekti on suunniteltu modulaarisesti | Kyllä | Projektin ydintä pystyy laajentamaan erilaisilla moduleilla |
| Lähdekoodi on saatavilla ja se on käyttökelpoinen (kääntyy ja käynnistyy) | Kyllä | Lähdekoodi on saatavilla ja on käyttökelpoinen |
| Yhteisö ja kehittäjät ovat kiinnostuneita projektista | Kyllä | Projektia käytetään ja kehitetään jatkuvasti |
| Projektin luoja on motivoitunut hallinnoimaan sitä tai löytämään itselleen vastikkeen | Kyllä | Projektia hallinnoi The Apache Software Foundation |
| Kommunikaatio ja kontribuutio ovat mahdollisia | Kyllä | Kommunikaatio tapahtuu IRC kanavassa tai sähköpostitse. Kontribuutiot ovat mahdollisia |
| Tärkeät | Löytyy | Perustelut |
| Projektin laajuus on määritelty | Kyllä | Määritelty projektin kotisivulla |
| Projektilla on ohjelmointikäytännöt | Kyllä | Kontribuutio ja ohjelmointi ohjeet löytyvät dokumentaatiosta |
| Kehitysversioiden julkaisu on lyhytsyklistä ja käyttäjäversio on vakaata | Kyllä | Ohjelmiston versiot ovat taaksepäin yhteensopivia, julkaistut versiot vakaita |
| Kehittäjiä houkutteleva lisenssi | Kyllä | Apache Lisenssi 2.0 |
| Toimiva hallinto projektin sisällä | Kyllä | Projektia hallinnoi The Apache Software Foundation |
| Toivotut | Löytyy | Perustelut |
| Projektista on saatavilla riittävä dokumentaatio | Kyllä | Projektista on saatavilla dokumentaatio usealla kielellä, ohjeita ja tieteellistä materiaalia |

Taulukko 6. Senyardin ja Michlmayerin tutkimustuloksen aktiviteetit, jotka tulisi suorittaa, jotta projekti menestyisi avoimessa yhteisössä.

3.3 Menestyksen mittarit VueJS projektissa

VueJS on progressiivinen JavaScript kehys, joka on viime vuosien aikana noussut suosituimpien kehysten joukkoon [36, 49]. Kasvava suosio olikin yksi menestyksen merkeistä [23, 46]. VueJS kuvailee itseään helposti lähestyttävänä, monipuolisena ja suorituskykyisenä kehysenä. Tässä kappaleessa esitellään VueJS projektia ja tutkitaan miten kahden valitun tutkimuksen tulokset sopivat projektiin.

Taulukossa 7 esitämme Emanuelin ja hänen tutkimusryhmänsä [8] menestyksen mittareiden olemassaoloa VueJS projektissa. VueJS on julkaistu vuonna 2013 ja se on MIT lissenssin alainen. VueJS kehysten avulla tehdään nettisivuja, sen käyttäjinä ovat tavalliset ihmiset. Tämä menestyksen mittari ei täyty VueJS projektissa. Samalla tavalla kuin Apache projektissa, VueJS on tarkoitettu käyttäjille, mutta lopputuotetta käyttävät tavalliset ihmiset. VueJS lähdekoodi on käyttövalmis tuotantoon [46] ja se toimii Linuxin ja Windows käyttöjärjestelmällä. Käyttöjärjestelmä ei kuitenkaan ole relevantti VueJS projektissa. VueJS on saatavilla zip tiedostona, mutta sekään ei ole tavallinen tapa ottaa käyttöön JavaScript kehysiä. Projektin voi asentaa JavaScript kielelle tarkoitettuna npm paketinhallintaohjelman kautta. Projektin kielenä toimii englanti ja sen lisäksi dokumentaatio on käännetty yhdeksälle kielelle. VueJS projektilla on selvästi menestyksen merkkejä Emanuelin saamien tutkimustulosten mukaan.

| Menestyksen mittari | Löytyy | Perustelut |
|---|--------|--|
| Projektin käyttäjinä tavalliset ihmiset | Ei | Käyttäjinä ovat ohjelmoijat |
| Lähdekoodi on valmis käyttöä varten | Kyllä | Projektista on saatavilla versio 3. Projekti on käytössä muun muassa verohallinnon nettisivuilla |
| Projekti toimii Linuxilla tai Windowsilla | Kyllä | Projekti toimii Linuxilla tai Windowsilla |
| Projektilla on ainakin yksi arvostelu | Kyllä | Projektista on kirjoitettu useita arvosteluja eri kielillä. GitHub palvelussa 163 419 tähteä |
| Projekti jaetaan zip tiedostona ja sen kokoluokka on megatavuissa | Kyllä | On saatavilla zip tiedostona. Projektin koko on 33.30KB |
| Projekti on saatavilla englanniksi tai muulla eurooppalaisella kielellä | Kyllä | Saatavilla yhdeksällä eri kielellä, mukaan lukien englanti |

Taulukko 7. Emanuelin tutkimuksen menestyksenmittarit ja niiden olemassaolo VueJS projektissa.

Taulukossa 8 esitämme Senyardin ja Michlmayerin [46] tutkimuksen löytämiä mittareita ja niiden yhteyden VueJS projektiin. VueJS ei ole ollut pelkkä prototyyppi jo pitkään aikaan ja siitä on suunnitella julkaistavaksi versio 3 vuoden 2020 puolella välissä. VueJS tärkeimmät ominaisuudet ovat *modulaarisuus ja joustavuus*, VueJS ei myöskään pakota käyttäjiään rakentamaan omia projekteja tietyllä tavalla. Projektiin on suunniteltu uusia ominaisuuksia ja toimintasuunnitelma löytyy pitkäaikaisille tavoitteille, jotka on jaettu vuosineljänneksittäin. Suunnitelmaa päivitetään usein, joten se antaa ajankohtaisen kuvan projektin tilasta ja tulevaisuudesta. Projektia hallinnoi ydinryhmä, johon kuuluu projektin luoja Evan You. Evan työskentelee projektin parissa täysipäiväisesti. VueJS käyttämää MIT lisenssiä käytti vuonna 2015 44 prosenttia GitHub:in projekteista ja täten se oli GitHub:in suosituin lisenssi [4]. Kaikki projektiin kontribuoineet ovat esillä sen GitHub sivulla.

| Välttämättömät | Löytyy | Perustelut |
|---|---------------|--|
| Projektista on saatavilla lupaava prototyyppi | Kyllä | Käytössä tuotannossa |
| Projekti on suunniteltu modulaarisesti | Kyllä | Joustavuus ja modulaarisuus ovat projektin päätavoitteen |
| Lähdekoodi on saatavilla ja se on käyttökelpoinen (kääntyy ja käynnistyy) | Kyllä | Lähdekoodi on avoin ja projekti käyttökelpoinen |
| Yhteisö ja kehittäjät ovat kiinnostuneita projektista | Kyllä | Kyllä, yli kuusi tuhatta katsojaa ja 382 kontribuojaa GitHubissa |
| Projektin luoja on motivoitunut hallinnoimaan sitä tai löytämään itselleen vastikkeen | Kyllä | Projektin on mukana kehityksessä täysipäiväisesti |
| Kommunikaatio ja kontribuutio ovat mahdollisia | Kyllä | Tapahtuu Discordissa ja GitHubissa |
| Tärkeät | Löytyy | Perustelut |
| Projektin laajuus on määritelty | Kyllä | Projektilla on etenemissuunnitelma |
| Projektilla on ohjelmointikäytännöt | Kyllä | Kontribuutio ja ohjelmointi ohjeet ovat saatavilla |
| Kehitysversioiden julkaisu on lyhytsyklistä ja käyttäjäversio on vaka | Kyllä | Kehitysversion voi rakentaa milloin tahansa. Käyttäjäversiot ovat käytössä tuotannossa |
| Kehittäjiä houkutteleva lisenssi | Kyllä | MIT |
| Toimiva hallinto projektin sisällä | Kyllä | Projektilla on ydinryhmä sekä laaja joukko osajia ja vastuuhenkilöitä |
| Toivotut | Löytyy | Perustelut |
| Projektista on saatavilla riittävä dokumentaatio | Kyllä | Projektin dokumentaatio on kymmenellä eri kielellä |

Taulukko 8. Senyardin ja Michlmayerin tutkimustuloksen aktiviteetit, jotka tulisi suorittaa, jotta projekti menestyisi avoimessa yhteisössä.

4 Pohdinta ja yhteenveto

Tutkimuksessamme löysimme menestyksen tekijöitä ja indikaattoreita, joiden avulla on mahdollista tutkia menestynyttä avoimen lähdekoodin projektia. Kokonaisvaltaisen kuvan muodostamiseksi käytimme systemaattista kirjallisuuskatsausta, jossa tutkimme 28 tieteellistä artikkelia, jotka käsittelivät avoimen lähdekoodi projektien menestystä.

Tutkimuksemme alussa meillä oli kaksi tutkimuskysymystä. Jotta pystyimme vastaamaan niihin, käytimme systemaattista kirjallisuuskatsausta, jossa keräsimme ja analysoimme aikaisempia tutkimuksia menestyksen tekijöistä avoimen lähdekoodin projekteissa. Tämän lisäksi tutkimme kahta menestynyttä projektia ja analysoimme niiden menestystä löytämiemme mittareiden avulla.

Kirjallisuuskatsauksessa löysimme joukon menestyksen tekijöitä, joista osa oli helppo mitata, kuten latausten määrä ja osa oli todella monimutkaisia ja monipuolisia kuten yhteisön hyvinvointi. Huomasimme kuitenkin, että myös latauksen määrästä voi tehdä monimutkaisen ja monipuolisen mittarin. Latauksien määrää mitatessa täytyy ottaa huomioon muun muassa seuraavat asiat ”*Monen latauksen jälkeen projektista tulee suosittu?*”, ”*Tarkoittaako lataaminen myös projektin käyttöä?*”, ”*Onko yhden henkilön lataus yhtä arvokas kuin ison yrityksen?*”. Menestystä mitatessa, nämä kysymykset voivat tuoda epätarkkuutta ja vaikeuksia, mikäli niihin ei ole vastausta.

Yksi tutkimuskysymyksistämme oli ”*Millä tavalla voidaan mitata menestyksen avaintekijät avoimen lähdekoodin projektissa?*”. Tähän kysymykseen etsimme vastausta kirjallisuuskatsauksessa, jossa tarkastelimme 419 tieteellistä artikkelia ja analysoimme niistä 28. Yhteenveto on seuraavanlainen:

Ensimmäisenä menestys täytyy määritellä. Määrittelyn apuna tulisi käyttää projektin omia arvoja ja tavoitteita. Tässä vaiheessa täytyy päättää monta menestyksen mittaria me halutaan tarkastella ja monta niistä ovat saatavilla sekä mitattavissa. Menestyksen mittareita määriteltessä täytyy olla täsmällinen ja mahdolliset arvelut on poistettava niin, että mittarin määritelmä on mahdollisimman yksiselitteinen. Esimerkiksi jos päätetään käyttää latausten määrää yhtenä mittarina meidän täytyy päättää, että kaikki lataukset ovat yhdenvertaisia eikä lataus tarkoita käyttöönottoa. Tämän vaiheen jälkeen määritellyt asiat olisivat selkeitä niin tutkijoille kuin myöhemmin lukijalle. Esimerkki latauksien määrästä on vain yksittäinen tapaus ja mittauksen yhteydessä siihen voi tulla yllättäviä käännteitä, niiden yllätyksellisyys täytyy tiedostaa ja hyvällä valmistautumisella osa yllätyksistä voidaan ennaltaehkäistä. Mittareita soveltaessa täytyy tutustua projektiin ja tutkia siitä saatavilla oleva aineisto, jotta voimme muodostaa kokonaisvaltaisen käsityksen projektista, mikä auttaa mittareiden so-

veltamisessa. Kun mittareita sovelletaan ja informaatiota etsitään on tärkeää pitää mielessä päämäärä ja etsittävät asiat, jotta pysytään tavoitteessa.

Esimerkit menestyksen mittareista ovat saatavilla kirjallisuuskatsauksen yhteydessä. Jos tavoitteesi on tehdä omasta projektista menestynyt niin sinun täytyy määritellä projektille tavoite ja tutkia millä tavalla muut ovat mitanneet menestystä. Sen jälkeen etsi asiat, jotka eivät ole onnistuneet projektissasi. Kiinnitä erityistä huomiota indikaattoreihin, jotka voisivat löytyä projektistasi, mutta eivät ole vielä havaittavissa sillä tämä saattaa tarkoittaa, että jotkut asiat täytyy tehdä eri tavalla.

Meidän toinen tutkimuskysymys oli ”*Miten menestyksen indikaattorit soveltuvat käytännössä?*”. Indikaattorit voidaan soveltaa, mikäli kyseessä on jo menestynyt projekti ja todeta sen menestys. Tutkimuksessamme törmäsimme siihen, että monet löydetyistä indikaattoreista ja mittareista eivät ole helposti käytettävissä sellaisinaan. Ne ovat kuitenkin arvokkaita ohjeita kun tähtäimessä on menestynyt projekti.

Tapaustutkimusta tehdessä totesimme, että kirjallisuuskatsauksen tuloksena saadut menestyksen mittarit olivat löydettävissä menestyneistä projekteista. Tämä näyttää meille sen, että on mahdollista arvioida onko projekti menestynyt vai ei. Eri projektien menestystä voidaan myös vertailla keskenään, mikäli molemmista projekteista on saatavilla samat indikaattorit ja projektien menestys on määritelty samalla tavalla. Vertailussa voidaan kuitenkin kohdata ongelmia kuten projektin eri iät ja niistä saatavilla olevan informaation arvo nykyajan ohjelmistokehityksessä.

Avoimen lähdekoodin yhteisössä voidaan kuitenkin kysyä missä määrin kahden avoimen lähdekoodi projektin menestyksen vertailu on tarpeellista? Avoimen lähdekoodin yhteisöt eivät ole yhtä kilpailuhaluisia kuin suljetun ohjelmiston markkinoilla olevat yritykset. Usein monessa avoimen lähdekoodin projekteissa on samoja kehittäjiä ja joskus jopa samassa kategoriassa olevat projektit onnittelevat toisiaan menestyksen johdosta [51]. Avoimen lähdekoodin projektien vertailu ei kuitenkaan ole turhaa. Se on tärkeässä roolissa varsinkin silloin kun käyttäjä tekee valinnan projektin käyttöönotosta toisessa projektissa. Jos vaihtoehtoina on kaksi projektia joista yhden menestys on hiipuva ja toisen kasvava, niin kannattaa valita jälkimmäinen. Kahden menestyneen projektin vertailu oli meidän tutkimuksemme ulkopuolella. Se on kuitenkin yksi osa-alueista, jolla tutkimustamme voidaan jatkaa ja täydentää.

Lukijan täytyy ottaa huomioon, että tarkastelimme menestyksen mittareita kahdessa projektissa, jotka olivat jo ennestään menestyneitä. Tutkimustamme voidaan täydentää tarkastelemalla menestyksen mittareita projekteissa, jotka eivät ole menestyneitä tai ovat jo epäonnistuneet. Enemmän tapatutkimuksia toisi uusia näkökulmia projektin menestyksestä

sekä muuttaisi tai vahvistaisi olemassa olevia havaintoja. Mielenkiintoinen osa-alue tutkitavaksi olisi miten menestyksen mittarit toimivat projektien alkuvaiheessa ja miten hyvin ne ennustavat menestystä tulevaisuudessa. Menestyksen mittareita voidaan myös etsiä tietokannoista, jotka eivät olleet saatavilla meidän tutkimuksessa. Voi olla mahdollista luokitella mittarit mittaamaan projekteja, jotka kuuluvat tiettyyn kategoriaan. Tämä helpottaisi niin projektin menestyksen mittaamista kuin projektien vertailua.

Tutkimuksemme yhteenvetona voimme sanoa, että avoimen lähdekoodin projektin menestystä voidaan mitata, mutta mittaaminen ja siihen liittyvät määritelmät on tehtävä projektiokohtaisesti. Projektin menestyksen täytyy olla määriteltynä ja sen pitää olla realistinen ja saavuttavissa oleva. Tutkimuksemme ja tuloksemme voidaan käyttää käsikirjana ja inspiraationa kun tähdätään menestyneeseen avoimen lähdekoodin projektiin.

Lähteet

- [1] Howard E. Aldrich ja Phillip H. Kim. ”SMALL WORLDS, INFINITE POSSIBILITIES? HOW SOCIAL NETWORKS AFFECT ENTREPRENEURIAL TEAM FORMATION AND SEARCH” (2007).
- [2] Apache. *Apache contributors*. 2020. URL: <http://httpd.apache.org/contributors/> (viitattu 08.06.2020).
- [3] Guilherme Avelino, Marco Valente ja Andre Hora. ”What is the Truck Factor of popular GitHub applications? A first assessment” (tammikuu 2015). DOI: 10.7287/PEERJ.PREPRINTS.1233.
- [4] Ben Balter. *What is open source software*. 2015. URL: <https://github.blog/2015-03-09-open-source-license-usage-on-github-com/> (viitattu 26.05.2020).
- [5] Ben Bromhead. *10 advantages of open source for the enterprise*. 2017. URL: opensource.com/article/17/8/enterprise-open-source-advantages (viitattu 28.05.2020).
- [6] Jailton Coelho ja Marco Tulio Valente. ”Why Modern Open Source Projects Fail”. Teoksessa: *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. ESEC/FSE 2017. Paderborn, Germany: Association for Computing Machinery, 2017, s. 186–196. ISBN: 9781450351058. DOI: 10.1145/3106237.3106246. URL: <https://doi.org/10.1145/3106237.3106246>.
- [7] Cambridge Dictionary. *Definition of 'project'*. 2020. URL: dictionary.cambridge.org/dictionary/english/project (viitattu 15.05.2020).
- [8] A. W. R. Emanuel, R. Wardoyo, J. E. Istiyanto ja K. Mustofa. ”Success factors of OSS projects from sourceforge using Datamining Association Rule”. Teoksessa: *2010 International Conference on Distributed Frameworks for Multimedia Applications*. 2010, s. 1–8.
- [9] Richard E. Fairley. ”Software Project Management”. Teoksessa: *Encyclopedia of Computer Science*. GBR: John Wiley ja Sons Ltd., 2003, s. 1634–1636. ISBN: 0470864125.

- [10] D. G. Feitelson, G. Z. Heller ja S. R. Schach. "An empirically-based criterion for determining the success of an open-source project". Teoksessa: *Australian Software Engineering Conference (ASWEC'06)*. 2006, 6 pp.–368.
- [11] Brandon Foushee, Jonathan L. Krein, Justin Wu, Randy Buck, Charles D. Knutson, Landon J. Pratt ja Alexander C. MacLean. "Reflexivity, Raymond, and the Success of Open Source Software Development: A SourceForge Empirical Study". Teoksessa: *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*. EASE '13. Porto de Galinhas, Brazil: Association for Computing Machinery, 2013, s. 246–251. ISBN: 9781450318488. DOI: 10.1145/2460999.2461036. URL: <https://doi.org/10.1145/2460999.2461036>.
- [12] Inc. Free Software Foundation. *GNU General Public License*. 2007. URL: gnu.org/licenses/gpl-3.0.en.html (viitattu 15.05.2020).
- [13] Jean Garcia-Gathright, Christine Hosey, Brian St. Thomas, Ben Carterette ja Fernando Diaz. "Mixed Methods for Evaluating User Satisfaction". Teoksessa: *Proceedings of the 12th ACM Conference on Recommender Systems*. RecSys '18. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2018, s. 541–542. ISBN: 9781450359016. DOI: 10.1145/3240323.3241622. URL: <https://doi.org/10.1145/3240323.3241622>.
- [14] GitHub. *Getting started with github*. 2020. URL: help.github.com/en/github/getting-started-with-github/saving-repositories-with-stars#about-stars (viitattu 02.06.2020).
- [15] Mathieu Goeminne ja Tom Mens. "Evidence for the Pareto principle in Open Source Software Activity" (2011).
- [16] TODO Group. *2019 Open Source Program Survey Results*. 2019. URL: <https://github.com/todogroup/survey/tree/master/2019> (viitattu 28.05.2020).
- [17] Phillip H. Kim Howard E. Aldrich. *Small worlds, infinite possibilities? How social networks affect entrepreneurial team formation and search*. 2007. URL: dictionary.cambridge.org/dictionary/english/project (viitattu 19.05.2020).

- [18] Francis Hunt ja Paul Johnson. ”On the Pareto distribution of Sourceforge projects” (tammikuu 2002).
- [19] Open Source Initiative. *The MIT License*. 2020. URL: opensource.org/licenses/mit-license.html (viitattu 10.05.2020).
- [20] Instagram. *Libraries We Use*. 2020. URL: <https://www.instagram.com/about/legal/libraries/> (viitattu 29.05.2020).
- [21] ISO/IEC. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. 2011. URL: <https://www.iso.org/standard/35733.html> (viitattu 02.06.2020).
- [22] Cary Cooper Ivan Robertson. *Wellbeing: Productivity and Happiness at Work*. 2011.
- [23] Andi Wahyu Rahadjo Emanuel Jazi Eko Istiyanto. ”Success Factors of Open Source Software Projects using Datamining Technique” (2009).
- [24] David Hunt Jennifer Ferreira Michael Glynn, Jaganath Babu, Denis Dennehy ja Kieran Conboy. *Sentiment analysis of open source communities: An Exploratory Study*. 2019. URL: <https://doi.org/10.1145/3306446.3340816> (viitattu 07.06.2020).
- [25] JUHTA. *JHS 169 Avoimen lähdekoodin ohjelmien käyttö julkisessa hallinnossa*. 2012. URL: docs.jhs-suositukset.fi/jhs-suositukset/JHS169/JHS169.html (viitattu 28.05.2020).
- [26] J. Howison K. Crowston H. Annabi ja C. Masango. ”Collaboration, Conflict and Control: The 4th Workshop on Open Source Software Engineering” (toukokuu 2004), s. 29–33.
- [27] Robert G. Wolf Karim R. Lakhani. ”Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects” (2003). URL: <https://ssrn.com/abstract=443040>.
- [28] Andrea Capiluppi Karl Beecher Cornelia Boldyreff ja Stephen Rank. ”Evolutionary Success of Open Source Software: an Investigation into Exogenous Drivers” (2007), s. 29–33.

- [29] Evangelos Katsamakos ja Nicholas Georgantzas. ”Why Most Open Source Development Projects Do Not Succeed?” Teoksessa: *Proceedings of the First International Workshop on Emerging Trends in FLOSS Research and Development*. FLOSS ’07. USA: IEEE Computer Society, 2007, s. 3. ISBN: 0769529615. DOI: 10.1109/FLOSS.2007.15. URL: <https://doi.org/10.1109/FLOSS.2007.15>.
- [30] Hala Annabi Kevin Crowston ja James Howison. *Defining Open Source Software Project Success*. 2003. URL: aisel.aisnet.org/icis2003/28 (viitattu 18.05.2020).
- [31] Ilkka Niiniluoto Leila Haaparanta. *Johdatus tieteiliseen ajatteluun*. 2016.
- [32] Michael W. Lucas. *Absolute openBSD*. No startch press, 2003. ISBN: 0-596-00027-8.
- [33] Junaid Maqsood, Iman Eshraghi ja Syed Sarmad Ali. ”Success or Failure Identification for GitHub’s Open Source Projects”. Teoksessa: *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*. ICMSS ’17. Wuhan, China: Association for Computing Machinery, 2017, s. 145–150. ISBN: 9781450348348. DOI: 10.1145/3034950.3034957. URL: <https://doi.org/10.1145/3034950.3034957>.
- [34] Netcraft. *April 2020 Web Server Survey*. 2020. URL: <https://news.netcraft.com/archives/category/web-server-survey/> (viitattu 08.06.2020).
- [35] Netcraft. *August 2019 Web Server Survey*. 2019. URL: <https://news.netcraft.com/archives/2019/08/15/august-2019-web-server-survey.html> (viitattu 02.06.2020).
- [36] Netlify. *Vue.js Experience Over Time*. 2020. URL: <https://2019.stateofjs.com/front-end-frameworks/vuejs/> (viitattu 15.06.2020).
- [37] OSI. *The Open Source Definition*. 2007. URL: <https://opensource.org/osd> (viitattu 10.05.2020).
- [38] Taylor Otwell. *Taylor’s Tweet*. 2020. URL: <https://twitter.com/taylorotwell/status/590281695581982720> (viitattu 09.06.2020).
- [39] Bram Adams Parastou Tourani Yujuan Jiang. *Monitoring Sentiment in Open Source Mailing Lists: Exploratory Study on the Apache Ecosystem*. 2014, s. 34–44.

- [40] Yujuan Jiang Parastou Tourani ja Bram Adams. "Monitoring sentiment in open source mailing lists: exploratory study on the apache ecosystem." Teoksessa: CASCON '14. Riverton, NJ, USA, 2014, s. 34–44.
- [41] Bruce Perens. *Open Sources: Voices from the Open Source Revolution*. 1999. ISBN: 1-56592-582-3.
- [42] Kai Petersen, Robert Feldt, Shahid Mujtaba ja Michael Mattsson. "Systematic Mapping Studies in Software Engineering". *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering* 17 (kesäkuu 2008).
- [43] E. S. Raymond. *cathedralandbazaar*. 1999. URL: <http://tuxedo.org/~esr/writings/cathedral-bazaar/> (viitattu 26.05.2020).
- [44] RedHat. *What is open source software*. 2020. URL: <https://www.redhat.com/en/topics/open-source/what-is-open-source-software> (viitattu 29.05.2020).
- [45] H.-W. Kim S.-Y.T. Lee ja S. Gupta. "Measuring open source software success". 37 (2009), s. 426–438. URL: <http://dx.doi.org/10.1016/j.omega.2007.05.005>.
- [46] A. Senyard ja M. Michlmayr. "How to have a successful free software project". Teoksessa: *11th Asia-Pacific Software Engineering Conference*. 2004, s. 84–91.
- [47] Spotify. *Spotify*. 2020. URL: <https://github.com/spotify/android-sdk> (viitattu 29.05.2020).
- [48] Rich Stim. *Welcome to the Public Domain*. 2020. URL: <https://fairuse.stanford.edu/overview/public-domain/welcome/> (viitattu 15.05.2020).
- [49] Google Trends. *vuejs search trend*. 2020. URL: <https://trends.google.com/trends/explore?date=today%5C%205-y&q=vuejs> (viitattu 10.06.2020).
- [50] Jason T. Tsay, Laura Dabbish ja James Herbsleb. "Social Media and Success in Open Source Projects". Teoksessa: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion*. CSCW '12. Seattle, Washington, USA: Association for Computing Machinery, 2012, s. 223–226. ISBN: 9781450310512. DOI:

10.1145/2141512.2141583. URL: <https://doi.org/10.1145/2141512.2141583>.

- [51] Ubuntu. *Ubuntu congratulates Fedora*. 2020. URL: <https://twitter.com/ubuntu/status/1255143290006990851> (viitattu 05.06.2020).
- [52] Wikipedia. *Bus factor*. 2020. URL: https://en.wikipedia.org/wiki/Bus_factor (viitattu 08.06.2020).
- [53] Ephraim R. McLean William H. DeLone. *Information systems success: the quest for the dependent variable*. 1992.