

Rolf Jacksen

PIENEN TAI KESKIKOKOISEN AVOIMEN LÄHDEKOODIN PROJEKTIN TUOTTEISTAMINEN

Opinnäytetyö

Tekniikan ammattikorkeakoulututkinto

Logistiikan koulutus

2023



**Kaakkois-Suomen
ammattikorkeakoulu**

Tutkintonimike	Insinööri (AMK)
Tekijä/Tekijät	Rolf Jacksen
Työn nimi	Pienen tai keskikokoisen avoimen lähdekoodin projektin tuotteistaminen
Toimeksiantaja	Open Memorylab-hanke ja Suomen avoimien tietojärjestelmien keskus - COSS ry
Vuosi	2023
Sivut	48 sivua
Työn ohjaaja(t)	Eeva Ala-Krekola, Timo Väliharju

TIIVISTELMÄ

Tämän opinnäytetyön tavoitteena oli tarkastella, mitkä tekijät johtavat onnistuneeseen avoimen lähdekoodin projektiin ja mikä edistäisi pienten ja keskikokoisten toimijoiden ja projektien kohdalla liiketoimintaa avoimen lähdekoodin parissa. Produktiivisena osana syntyi ohjeistus Suomen avoimien tietojärjestelmien keskus -COSS ry:lle pienten ja keskikokoisten avoimen lähdekoodin projekteja varten. Tämä ohjeistus on suunnattu potentiaalisille avoimen lähdekoodin käyttäjille, jotka eivät ole tietoisia tai varmoja avoimen koodin mahdollisuuksista ja kaupallisesta potentiaalista. Ohjeistus on 17-sivuinen Powerpoint-esitys, joka tukee ja rohkaisee avoimen lähdekoodin valintaan, kun hankinnassa vertaillaan avoimen- ja suljetun koodin vaihtoehtoja.

Tutkimuksen teoreettinen viitekehys käsittelee projektityötä, sen rakennetta ja etenemistä nelivaiheisen projektimallin kautta. Samalla tarkastellaan onnistuneen projektipolun tuotteistamista ja avoimeen koodiin liittyviä erityispiirteitä. Taustalla on systeemiajattelusta johdetun ekosysteemin tarkastelua, jonka olemassaolo ja kasvattaminen ovat elintärkeä osa koko avoimen koodin liiketoiminnan olemusta.

Tutkimus on laadullinen. Siinä perehdyttiin projektityötä koskevaan kirjallisuuteen, avointa lähdekoodia koskeviin tutkimuksiin ja korkeakoulutöihin sekä systeemiajatteluun, johon ekosysteemi perustuu. Tarkastelun kohteina olivat myös valtion- ja kuntahallinnon suositukset ja ohjeistukset avoimen koodin käytöstä. Lähteitä analysoitiin kuvailevalla kirjallisuuskatsauksella, jonka johtopäätösten mukaan rakennettiin projektipolkua koskevan ohjeisto sekä selvitettiin, mitkä yleisesti ovat onnistuneen projektin- ja projektityöskentelyn taustat.

Opinnäytetyössä pohdittiin myös, mikä edistäisi avoimeen lähdekoodiin liittyvää liiketoimintaa. Avoimen koodin ratkaisut ovat tällä hetkellä julkisella puolella EU-tasolta kuntatasolle asti voimakkaasti suositeltuja, mutta niitä toteutetaan silti vähän. Sama koskee kaupallisia ratkaisuja ja niiden kehittämistä. Osasyys on epävarmuus, jota opinnäytetyön produktiivisena osana olevalla oppaalla pyritään hälventämään. Tutkimuksellisessa osuudessa kerrotaan, miksi on päädytty oppaassa tehtyihin ratkaisuihin ja mihin niillä pyritään.

Asiasanat: avoin lähdekoodi, projektityö, tuotteistaminen, palvelumuotoilu, systeemiajattelu

Degree title	Bachelor of Engineering
Author (authors)	Rolf Jacksen
Thesis title	Productization of a small and medium-sized Open-source project
Commissioned by	Open MemoryLab & The Finnish Centre for Open Systems and Solutions – COSS ry
Time	2023
Pages	48 pages
Supervisor	Eeva Ala-Krekola & Timo Väliharju

ABSTRACT

This thesis examined which factors lead to a successful open source project and what would promote business with open source for small and medium-sized operators and projects. As a productive part, guidelines were created for The Finnish Centre for Open Systems and Solutions - COSS ry for small and medium-sized Open-source projects. The guidance is aimed at potential open source users who are not yet aware of, or confident about, the possibilities and commercial potential of open source. A 17-page PowerPoint presentation was created as a productive part, which supports and encourages the choice of Open-source code when comparing open and closed code options in procurement.

The theoretical framework of the study deals with project work, its structure, and progress through a four-phase project model. At the same time, we examine the productization of a successful project path and the related special features of the open code. Ecosystem in review is derived from systems thinking, which existence and growth are a vital part of the essence of the entire Open-source business.

The research method is qualitative research. In the work, we familiarized ourselves with the literature on project work, studies on Open-source code and university work, as well as systems thinking, on which the ecosystem is based. The review also included recommendations and guidelines of the state and municipal administrations on the use of Open-source code. The sources were analysed with a descriptive literature review, the conclusions of which were used to build guidelines for the project path, and the background of successful projects and project work, in general was clarified.

The thesis also considered what would promote business related to Open-source code. Open-source solutions are currently strongly recommended on the public side, from the EU level to the municipal level, but they are still rarely implemented. The same applies to commercial solutions and their development. Part of the reason is uncertainty, which the guide aims to dispel. The research part explains why the solutions made in the guide have been reached, and what they aim for.

Keywords: open source, project work, productization, system thinking

SISÄLLYS

1	JOHDANTO	5
2	AVOIN LÄHDEKOODI.....	10
2.1	Avoimen lähdekoodin lisenssityypit	11
2.2	Avoim tai salliva lisenssi.....	12
3	PROJEKTITYÖ	12
3.1	Projektin elinkaari.....	13
3.2	Projektinhallintamenetelmät	14
3.2.1	Ketterät menetelmät	14
3.2.2	Muut menetelmät.....	16
4	TUOTTEISTAMINEN	17
5	SYSTEEMIAJATTELU	20
6	TUTKIMUSOSIO.....	22
6.1	Kuvaileva kirjallisuuskatsaus.....	22
6.2	Aineiston haku.....	23
6.3	Kirjallisuuskatsausten aineistot.....	25
7	KIRJALLISUUSKATSAUKSEN TULOKSET	27
8	JOHTOPÄÄTÖKSET JA YHTEENVETO	41
9	POHDINTAA	42
9.1	Opinnäytetyöprosessi.....	42
9.2	Eettisyys ja luotettavuus.....	43
9.3	Merkitys ja jatkotutkimusaiheet	43
	LÄHTEET	45

1 JOHDANTO

Avoin lähdekoodi ja kaikille avoin julkinen data ovat uusia toimintamalleja, jotka haastavat potentiaalillaan perinteiset toimijat ja ohjelmistoprojektit sekä parhaimmillaan lisäävät yhteiskunnan avoimuutta, riippumattomuutta toimittajamonopoleista sekä kehittävät yleistä innovatiivisuutta.

Tämän tutkimuksellisen opinnäytetyön tarkoituksena on tarkastella avoimen lähdekoodin projekteja ja niiden erityispiirteitä sekä tutkia, mikä voisi edistää pienten ja keskikokoisten toimijoiden kohdalla avoimen lähdekoodin liiketoimintaa. Opinnäytetyö liittyy Kaakkois-Suomen ammattikorkeakoulu XAMK:in **Open Memorylab** -hankkeeseen, jossa on oppilaitoksen lisäksi mukana avoimen lähdekoodin asiantuntijana Suomen avoimien tietojärjestelmien keskus – COSS ry (jäljempänä pelkkä COSS ry).

Open Memorylab -hankkeen tarkoitus on nostaa alueellista organisaatioiden muutos- ja innovaatiokyvykkyyttä tarjoamalla neuvontaa, ohjausta, koulutusta, sekä vertaisoppimista avoimen lähdekoodin hyödyntämismahdollisuuksista yritysten digitalisaatiopyrkimysten ja liiketoiminnan kehittämisen tueksi (OpenMemoryLab 2023). COSS ry on osatoteuttajana hankkeessa ja toimii sen alaisuudessa tämän opinnäytetyön virallisena tilaajana ja ohjaajana. Opinnäytetyön tutkimuksellisen osan lisäksi työssä luodaan COSS ry:lle projektiohjeistus avoimen lähdekoodin pieniä ja keskisuuria projekteja varten.

COSS ry on voittoa tavoittelematon yhdistys, jonka tavoitteena on edistää avoimen lähdekoodin käyttöä niin yksityisellä kuin julkisella sektorilla. COSS ry haluaa kasvattaa tietoutta avoimen lähdekoodin mahdollisuuksista ja samalla toimia alustana ja katalyyttinä alueen liiketoiminnan kehittämiseksi. Yhdistys pyrkii jäsenistönsä kanssa olemaan tunnetuin ja tunnustettu avoimen lähdekoodin toimija Suomessa. Yhdistys pyrkii rakentamaan yhteistyötä, verkostoja ja ekosysteemejä jäsentensä liiketoiminnan tueksi. (COSS 2023.) Opinnäytetyö ja sen aihe liittyvät Open Memorylab -hankkeen pyrkimykseen tiiviiseen vuorovaikutukseen yritysten kanssa, jotta voidaan tutkia ja ymmärtää paremmin yritysten avoimen lähdekoodin teknologioiden hyödyntämismahdollisuudet sekä osaamistarpeet (OpenMemoryLab 2023).

Tutkimuksellisessa osassa olevan kirjallisuuskatsauksen synteessin pohjalta kehitetty projektiohjeistus on suunnattu yrityksille ja yhteisöille, jotka harkitsevat avoimen lähdekoodin ohjelmiston käyttöönottoa tai jatkokehittämistä omaan käyttöönsä tai ovat muuten kiinnostuneita avoimen lähdekoodin tarjoamista ratkaisuksista ja mahdollisuuksista.

Opinnäytetyön aihe on kiinnostava useasta syystä: olen aikaisemmin työskennellyt it-alalla ja osallistunut erilaisiin verkkopalvelujen kehittämissuunnitelmiin, työskennellen erilaisissa projektimalleissa. Olen käyttänyt työssä ja harrastuksissa myös jonkin verran avoimella lähdekoodilla yhteisökehitettyjä ohjelmia (muun muassa kuvankäsittelyssä käytettävä GIMP, vektorigrafiikkaohjelma Inkscape, äänenmuokkausohjelma Audacity, videoleikkausohjelma Shotcut). Toisella työharjoittelujaksollani tein silloiselle harjoitteluyritykselle Kymenlaakson jäte Oy:lle informatiivisen kartan taajamien kiinteistöjen biojätteiden keräysvelvollisuudesta käyttämällä avoimen lähdekoodin karttaohjelmaa QGIS:ia ja Suomen viranomaistahojen tarjoamaa avointa kartta- ja asukasdataa (Maanmittauslaitos, Suomen ympäristökeskus). Selvitin projektissa verkon yhteisötuen ja etäapuna toimineen osaajan avustuksella ohjelman asentamisen, toiminnot, käyttämisen ja rakensin sillä informatiivisen kartan, jota Kymenlaakson jäte Oy on myöhemmin käyttänyt tiedottamisessaan.

Avoin lähdekoodi ja siihen liittyvä systeemiajattelu on osa mielenkiintoista uuden liike- tai yhteisötoiminnan laajempaa ajattelumallia, joka on kiinnostava ja jota kansainvälinen verkkotietosanakirja Britannica kuvaa pikemminkin *aatteeksi* (Britannica 2005). Avoin lähdekoodi on toimintamalli, joka haastaa nykyiset suljetut ohjelmistot ja niiden kehittämisen sekä ansaintalogiikan.

Avoimen lähdekoodin rinnalla kulkee puheissa usein myös avoin data eli esimerkiksi viranomaispalveluiden tietosisällön avoin jakaminen uudelleen käytettäväksi ja hyödynnettäväksi jopa taloudellisesti. Tämäkin linkittyy omalla tavallaan avoimen kehittämisen ja tietoyhteiskunnan ideaaliin, mutta näitä kahta termiä ja ilmiötä ei tule kuitenkaan sekoittaa keskenään. Euroopan unioni on antanut avoimesta datasta oman erillisen direktiivinsä (Euroopan parlamentin ja neuvoston direktiivi (EU) 2019/1024).

Julkisen hallinnon tietohallinnon neuvottelukunta JUHTA on julkaissut vuonna 2009 edelleen voimassa olevan JHS-suosituksensa "Avoimen lähdekoodin ohjelmien käyttö julkisessa hallinnossa", jota oli valtiovarainministeriön lisäksi laatimassa oikeusministeriö, työ- ja elinkeinoministeriö, Hansel, Suomen Kuntaliitto sekä Lahden, Oulun ja Uudenkaupungin kaupungit. Suosituksen valmistelutyötä koordinoi COSS ry. Suosituksen valmistelussa huomioitiin Euroopan komission silloisen IDABC-ohjelman suositukset ja ohjeet, jotka suosittavat avoimen lähdekoodin käytön huomioimista hankintatoimen yhteydessä, EU:n poliittisten tavoitteiden, kuten yhteismarkkinoiden, kilpailukyvyyn ja tasa-arvoisen kohtelun hallinnossa, saavuttamiseksi. (Valtiovarainministeriö 2009.)

EU-tasolla ollaan pitkällä Suomea edellä. Komission ensimmäinen avoimen lähdekoodin strategia oli jo vuodelta 2000 (Euroopan parlamentti 2020, 11). 2020–2023 avoin lähdekoodi on uuden eurooppalaisen tietoyhteiskunnan visiossa yksi pääperiaatteista. "OPEN SOURCE SOFTWARE STRATEGY 2020–2023 - Think Open" on EU:n oma kehittämisstrategia, jossa avoin lähdekoodi on otettu keskeiseksi työkaluksi Euroopan *Digitaalisen autonomian*, osallistavan kehittämisen ja samalla myös osallistavan yhteiskunnan sekä sen omat valtiorajat ylittävän kehittämisyyhteistyön rakentamisessa. Avainsanoina Think Openissa ovat edellä mainitut innovatiivisuus ja yhteistyö. (Euroopan parlamentti 2020, 5.) Digitaalinen autonomia on tällä hetkellä noussut esille senkin takia, että levottomina maailmanaikoina avoimen lähdekoodin ohjelmat, jotka ovat lähtökohtaisesti toimittajalukottomia, ovat samalla huomattavasti vähemmän alttiita esimerkiksi poliittisille juonitteluille ja kauppakiistoille. Avoimuuden kaikilla tasoilla katsotaan ylipäättään lisäävän myös luottamusta yhteiskunnan ja EU:n toimielimiin.

Avoimen lähdekoodin katsotaan myös hyödyttävän kaikkia osapuolia sen ideaalien mukaisen tiedon, ratkaisujen ja ohjelmistojen jakamisen ja uudelleenkäytön kautta. Tämä mahdollistaisi esimerkiksi samojen toimivien it-ratkaisuiden käytön ja monistamisen EU-tasolla valtioiden ja kuntatasolla kuntien kesken. Käytäntö voisi osaltaan johtaa mittaviin säästöihin valtionhallinnon ja kuntien it-menoissa. (Euroopan parlamentti 2020, 7.) Suomessa avoimen lähdekoodin eduiksi julkisella puolella onkin katsottu, että parhaassa tapauksessa

se poistaisi ohjelmistojen hankinnassa ja kehityksessä yritysten monopoleja, toimittajalukkoja ja riippuvuuksia. Tämä johtaa osaltaan taas suurempaan yhteiskunnalliseen riippumattomuuteen ja haavoittuvuuden vähenemiseen sekä kustannussäästöihin. (JHS 2009.)

Avoimen lähdekoodin kaupallisten toimijoiden määrä ei ole kuitenkaan Suomessa toistaiseksi oleellisesti noussut ja mahdollistanut tätä. Open Memory-lab-hankkeen (2023) mukaan tällä hetkellä avoimen lähdekoodin hyödyntämistä ei vielä edistetä riittävän systemaattisesti. Tämän lisäksi elinkeinoelämän ja julkisten toimijoiden yhteistyössä ja elinkeinolähtöisen toiminnan tukemisessa on edelleen kehittämisen varaa. Opinnäytetyössä esitetään tähän ratkaisua avoimen lähdekoodin projektien suoritusmallin tuotteistamisen pohjalta.

Työn ja samalla tutkimuksen rajaus tehtiin COSS ry:n kanssa koskemaan pieniä ja keskikokoisia projekteja sekä pieniä ja keskikokoisia toimijoita. Rajaukselle on syynä, että produktiivisen osan tarkoitus on rohkaista yrityksiä ja yhdistyksiä kokeilemaan avoimen lähdekoodin käyttöä ja kehittämistä ja tämä on järkevää aloittaa helposti hahmotettavista kokonaisuuksista ja hallittavista projekteista, joiden lopputulos on myös nopeasti saavutettavissa. Tämä projektien kokoluokan rajaus vaikutti myös oppaassa esimerkkinä käytetyn projektimallin valintaan. Pohdinta ja selvitys tehdään opinnäytetyössä juuri edellä mainitun kokoluokan yritysten ja projektien näkökulmasta.

Tutkimusongelmat valittiin yhdessä COSS ry:n kanssa ja ne tukevat opinnäytetyön produktiivista osaa, mutta muodostavat erikseen opinnäytteen itsenäisen tutkimuksellisen osan. Tutkimusongelmat perustuvat avoimen lähdekoodin ja liiketoiminnan yhdistämiseen. Tarkasteltavaksi otettiin onnistunut projektityöskentely ja avoimen lähdekoodin projektien mahdolliset poikkeavat vaatimukset tai metodit tällaisessa työskentelyssä. Toisena tutkimusongelma on, miten voisi heijastaa esimerkiksi edellisen kaltainen onnistunut projekti ja sen parhaan käytännön malli avoimen lähdekoodin liiketoiminnan ja houkuttelevuuden kasvattamiseen. Tarkasteltavana on avoimen lähdekoodin projektien piirteet ja ongelmat sekä julkisen että yksityisen puolen toimijoiden näkökulmasta. Tavoitteena on etsiä vastauksia näihin kahteen tutkimusongelmaan, jotka linkittyvät keskenään.

Tutkimuskysymyksiä on kaksi:

- Minkälaiset huomiot ja tekijät johtavat onnistuneeseen avoimen lähdekoodin projektiin?
- Mikä edistäisi pienten ja keskikokoisten toimijoiden ja projektien kohdalla liiketoimintaa avoimen lähdekoodin parissa?

Oletuksena toimeksiantajan antamassa tehtävässä on saada kumpaankin kysymyksen vastauksia, joita käytetään työn produktiivisen osan eli projektiohjeistuksen sisällön tueksi.

Lopputuloksessa selvitetään nelivaiheisen projektin elinkaarimallin sisällä, missä ja minkälaisia ongelmia ja mitä huomioita kannattaa tehdä avoimen lähdekoodin projekteissa, jotta projekti etenee onnistuneesti. Samalla käydään projektin elinkaari vaihe vaiheelta läpi ja sijoitetaan huomiot siihen. Tämän jälkeen pohditaan yleisesti avoimen lähdekoodin ja systeemiajattelun periaatteita ja tarkastellaan mahdollisia keinoja, joilla onnistunut projektimalli saataisiin tukemaan liiketoiminnan kasvua ja liiketoiminnan kasvattamisen metodeja yleisesti.

Tutkimus on teoreettinen. Tutkimukselliseksi lähtökohdaksi valikoitui työn tarkoituksen takia kvalitatiivinen eli laadullinen tutkimus. Laadullisessa tutkimuksessa käydään usein keskustelua, onko se teoreettista, vai empiiristä ja laadulliseksi tutkimukseksi laitetaan usein miltei kaikki, joka ei mahdu kvantitatiivisen tutkimuksen mittaukseen tai mittareihin. (Tuomi & Sarajärvi 2009.) Laadullinen tutkimus soveltuu kuitenkin erityisen hyvin tilanteisiin, joissa halutaan tutkia luonnollisia tilanteita, joita ei voi järjestää kokeeksi tai joissa ei voida kontrolloida läheskään kaikkia vaikuttavia tekijöitä. Laadullista tutkimusta käytetään myös, jos halutaan saada tietoa tiettyihin tapahtumiin liittyvistä syy-seuraussuhteista, joita ei voida tutkia kokeen avulla (Metsämuuronen 2000, 14).

Käytän opinnäytetyössä laadullisen tutkimuksen yhtä muotoa, kuvailevaa kirjallisuuskatsausta. Kirjallisuuskatsaus on tutkimusta olemassa olevasta tutkimus- ja muusta tiedosta. Tutkimusaineistona käytän tiedonkeruulla löytyvää kirjallista ja verkossa olevaa lähdemateriaalia, joka käsittelee projektihallintaa,

prosesseja, tuotteistamisesta, palvelumuotoilua, systeemiajattelua, avoimen lähdekoodin viranomaissuosituksia sekä COSS ry:n itse keräämää tietoa.

Teoriana käsitellään projektityötä ja sen hallintaa. Opinnäytetyössä tarkastellaan myös systeemiajattelua sekä siitä johdettua ekosysteemiä. Opinnäyte sivuaa myös palvelumuotoilua ja sen periaatteita.

Muodostin tietolähteistä tehdyn kuvailevan kirjallisuuskatsauksen tuloksena synteessin, jota käytin produktiivisen osan koostamisessa. Samalla esitin vastaukset tutkimuskysymyksiin, jotka perustelin lähdemateriaalista tekemälläni deduktiivisella analyysillä. Teorialähtöisessä eli deduktiivisessä analyysissä aikaisemman tiedon perusteella tehdään analyysirunko, johon etsitään sisällöllisesti sopivia asioita aineistosta. Tuomi & Sarajärvi (2009, 97) toteavat kyseen olevan siitä, että aineiston analyysiä ohjaa valmis aikaisemman tiedon perusteella luotu kehys. Teorialähtöisessä analyysissä päättelyn logiikka yhdistetään usein deduktiiviseen päättelyyn (yleisestä yksittäiseen) (Tuomi & Sarajärvi 2009, 98).

2 AVOIN LÄHDEKOODI

COSS ry (2023) määrittelee avoimen lähdekoodin niin, että se on tapa kehittää ja jaella tietokoneohjelmistoja, joita asiakas saa vapaasti käyttää, kopioida, muunnella ja jaella ilman lisenssimaksuja ja lisenssien ylläpitoa, joka voi toisinaan olla työlästä.

Tarkkaa ja lopullista määritelmää avoimesta koodista ei ole, mutta termi juontaa juurensa 1980-luvun Yhdysvalloissa alun perin hakkeripiirien alustamasta *free software* (avoin ohjelmisto) nimityksestä. Vuonna 1998 perustettu Open Source Initiative eli OSI lanseerasi tästä jatkotermin *open source* (avoin lähdekoodi), joka on nykyisin käytössä.

OSI:n määritelmän mukaan avoimen lähdekoodin ohjelman tulee täyttää seuraavat vaatimukset:

1. Ohjelman täytyy olla vapaasti levitettävissä ja välitettävissä.
2. Lähdekoodin täytyy tulla ohjelman mukana tai olla vapaasti saatavissa.

3. Myös johdettujen teosten luominen ja levitys pitää sallia.
4. Lisenssi voi rajoittaa muokatun lähdekoodin levittämistä vain siinä tapauksessa, että lisenssi sallii korjaustiedostojen ja niiden lähdekoodin levittämisen. Voidaan myös vaatia, ettei johdettua teosta levitetä samalla nimellä tai versionumerolla kuin lähtöteosta.
5. Yksilöitä tai ihmisryhmiä ei saa asettaa eriarvoiseen asemaan.
6. Käyttötarkoituksia ei saa rajoittaa.
7. Kaikilla ohjelman käsiinsä saaneilla on samat oikeudet.
8. Lisenssi ei saa olla riippuvainen laajemmasta ohjelmistokokonaisuudesta, jonka osana ohjelmaa levitetään, vaan ohjelmaan liittyvät oikeudet säilyvät, vaikka se irrotettaisiin kokonaisuudesta.
9. Lisenssi ei voi asettaa ehtoja muille ohjelmille. Ohjelmaa saa levittää myös yhdessä sellaisten ohjelmien kanssa, joiden lähdekoodi ei ole avointa.
10. Lisenssin sisällön pitää olla riippumaton teknisestä toteutuksesta. Oikeuksiin ei saa liittää varaumia jakelutavan tai käyttöliittymän varjolla.” (COSS 2023.)

Huomattavaa avoimen lähdekoodin ohjelmistoissa on, että ohjelmistot perustuvat käytännössä kehittäjien vapaaehtoistyöhön. Ne ovat ”aineettomien tuotteiden vertaistuotantoa” (Moore 2006, 17).

2.1 Avoimen lähdekoodin lisenssityypit

Lisenssi tarkoittaa oikeutta käyttää, tehdä tai muokata jotain tuotetta tai ohjelmistoa. Lisenssi on käyttöluva, jonka oikeudenhaltijana voit myöntää teokselle, tietokannalle tai muulle tekijänoikeuden suojaa nauttivalle aineistolle. (Helsingin yliopisto 2023.) Niin sanotussa *suljetuissa ohjelmistoissa* tämä lisenssi on yleensä tarkkaan rajattu ja yleensä maksullinen, eikä oikeuta toista tai kolmatta osapuolta muokkaamaan tai jakamaan ohjelmistoa ilman erillistä lupaa.

Avoimen lähdekoodin kohdalla lisenssin päämäärä on käytännössä päinvastainen. OSI:n sivuilla luetellaan lukuisa joukko erilaisia lisenssivaihtoehtoja avoimen lähdekoodin käytössä. (OSI 2023.) Yhteistä näille kaikille on kuitenkin, että lisenssi on muusta sisällöstään huolimatta aina maksuton eli ilmainen. Tässä mielessä avoin lähdekoodi on enemmänkin aate ja ilmiö, joka

haastaa perinteiset ohjelmistoyritykset ja niiden toimintamallit. (Madley ym. 2002.)

OSI:n luettelo olemassa olevista avoimen lähdekoodin lisenssityypeistä on pitkä, mutta käytännössä käytettävien ohjelmistojen lisenssipaletti on huomattavasti suppeampi. Mikäli ohjelmistoa kehitetään jo olemassa olevasta ohjelmasta, pitää yleensä käyttää alkuperäisen kanssa samaa ohjelmistolisenssiä.

Mikäli luodaan jotain uutta, suositellaan yleisesti käyttämään lähinnä seuraavia avoimen lähdekoodin yleisiä lisenssejä: MIT (Massachusetts Institute of Technology), GNU GPL (GNU General Public License), AGPL (Affero General Public License), tai BSD (Berkeley Software Distribution). (Tampereen yliopiston kirjasto 2023.)

2.2 Avoin tai salliva lisenssi

Avoin eli tarttuva lisenssi tarkoittaa sitä, että kaikki ohjelmistoon kehitetyt muutokset on julkaistava samalla lisenssillä ja ehdoilla kuin alkuperäinen ohjelmisto. Salliva lisenssi mahdollistaa suoraan ohjelmiston muuttamisen ja kehittämisen kaupallisiin tarkoituksiin. Käytännössä kaupallisen toiminnan kannalta näillä lisenssimalleilla ei ole suurta eroa: kumpikaan ei kiellä esimerkiksi kaupallisten tukipalvelujen kehittämistä. Ohjelmiston lisäpalvelujen kehittämisestä voidaan myös maksaa, vaikka lähdekoodi jaettaisiinkin lisenssin mukaisesti ilmaiseksi yleiseen käyttöön. (Tampereen yliopiston kirjasto 2023.)

3 PROJEKTITYÖ

Projektityö tarkoittaa projektimaisesti organisoitua työtä (Mäntyneva 2016). Projekti käsitteenä on johdettu latinan sanasta *proicere*, joka tarkoittaa sananmukaisesti suomennettuna ”heittää eteenpäin”. Suomessa projektin synonyyminä käytetään usein sanaa ”hanke”, joka on kuvaava: projektilla on yleensä tarkoitus toteuttaa jotain ja sille määritellyssä ajassa. (Ruuska 2007.)

Löow (2002, 17) määrittää projektimuotoiselle työskentelylle viisi vaatimusta, jotka sen tulisi täyttää. Ensimmäiseksi; projektilla pitää olla tilaaja. Tämän jälkeen projektin tulee olla ajallisesti ja laajuudeltaan rajattu. Kolmantena vaati-

muksena on, että projektilla tulee olla selvä tavoite, sillä projekti katsotaan tavoiteohjauksen vahvimaksi muodoksi. Neljäntenä on projektin dokumentoinnin harjoittaminen koko projektin ajan. Viidentenä ja viimeisenä vaatimuksena on, että projektista pitää laatia kuvaus, ja sen aikana tulee noudattaa tulosten saavuttamisen takaavia työskentelymuotoja.

Viimeiseksi mainittuja ovat muun muassa projektikokoukset, projektiin resursoidut ihmiset ja heidän työpanoksensa, jatkuva tiedottaminen asianosaisille, välitavoitteet ja pisteet sekä työn- ja vastuunjako osallistujien kesken.

Projektimuotoinen työskentely eli projektityö on kasvattanut suosiotaan kaikilla aloilla, mutta on ollut hyvin kiinteä ja olennainen osa ohjelmistokehityksessä läpi sen historian, niin että muutama projektityössä käytettävä yleinen työskentelymalli eli projektinhallintamenetelmä on syntynyt suoraan alun perinkin ohjelmistokehityksen tarpeisiin, kuten Waterfall, eli vesiputousmalli ja ketterän kehityksen Scrum. (Van Casteren 2017.)

Projektityöskentelyn suosion keskeiseksi syyksi katsotaan, että työskentelytapa on hyvin tulossuuntautunut (Löow 2002, 18). Tämä ohjaa automaattisesti projektissa asioiden laittamiseen tärkeysjärjestykseen. Projektiin varatut henkilöt voivat täten keskittyä ja laittaa koko tarmonsensa projektissa määriteltyihin tehtäviinsä, jolloin projektimuotoinen työ on yleensä nopeammin etenevää kuin tavanomainen työskentely.

3.1 Projektin elinkaari

Projektin elinkaari on Mäntynevan (2016, 5) mukaan nelivaiheinen prosessi, jossa on alku- ja loppupiste. Alussa on projektin **valmistelu**, tämän jälkeen sen toteutuksen **suunnittelu**, jonka jälkeen siirrytään projektin **toteutukseen**. Projektin lopussa **päätetään** projekti ja se on valmis. Työmäärällisesti suurin osuus sijoittuu projektin toteutusvaiheeseen, vaikka nykyään on pyrkimyksenä, että valmistelu- ja suunnitteluvaiheeseen siirretty panostus vähentäisi tätä (Ruuska 2007).

Oleellista projektin kannalta on, että se saadaan päätösvaiheeseen ja lopetettua, sillä projektilla pitää olla päättepiste.

3.2 Projektinhallintamenetelmät

Projektimalli on yhdessä sovittu tapa johtaa projektia läpi sen elinkaaren. Projektin johtamiseen ja rakenteeseen on kehitetty monia menetelmiä, joita käytetään yleisesti esimerkiksi ohjelmistokehityksessä. Tällaisia menetelmiä kutsutaan projektinhallintamenetelmiksi ja ne ovat jaettavissa karkeasti kahteen eri ryhmään: ketterät menetelmät (Agile) ja muut menetelmät (Huosianmaa 2020, 9). Jälkimmäisiä kutsutaan usein myös *perinteisiksi* projektinhallintamenetelmiksi tai perinteisiksi projektimalleiksi (Finni 2022).

3.2.1 Ketterät menetelmät

Lähtökohta kaikelle ketterälle kehittämiselle on vuonna 2001 julkaistu Agile Manifesto (Van Casteren 2017). Se syntyi ilmenneeseen tarpeeseen, kun ohjelmistojen ja järjestelmien kehitystyössä huomattiin, että yleisesti siihen asti käytetyssä vesiputousprojektimallissa oli paluu edellisiin projektivaiheisiin estetty tai jähkkyä ja tämä rajoitti työskentelyä.

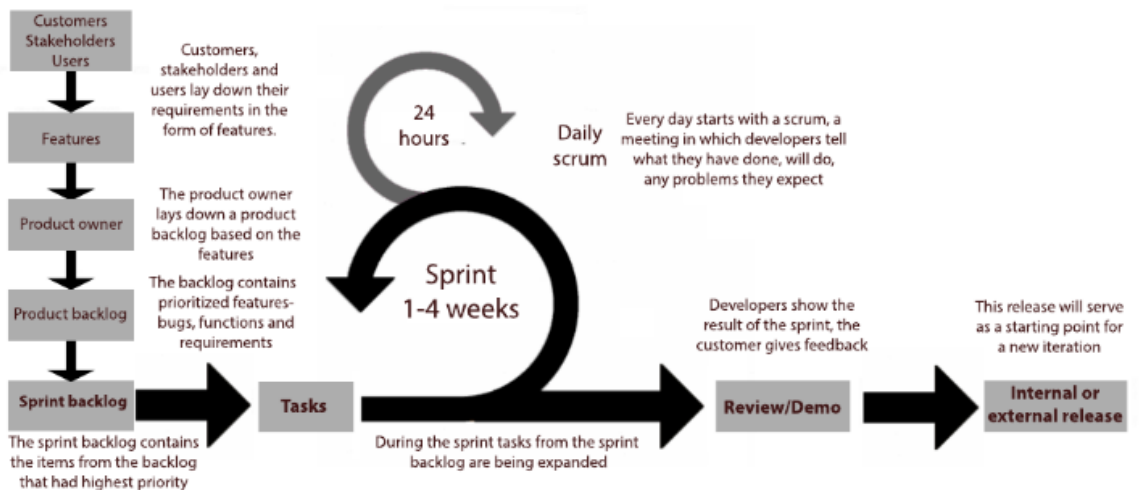
Rugbypelistä nimensä saanut **Scrum** on tunnetuin ketterien menetelmien alainen menetelmä ja se on erittäin suosittu ohjelmistokehityksessä. Ideana on, että koko projektia ei ole tarkoituskaan suorittaa alusta loppuun yhtenä katkeamattomana polkuna, vaan projekti koostuu pienemmistä osaprojekteista: projekteista projektien sisällä, joiden kehityssykli suunnittelusta toteutukseen voi olla hyvinkin nopea.

Ketteristä menetelmistä toinen yleisesti tunnettu ja käytetty on **Kanban** (se tarkoittaa sanatarkasti lippua japanin kielellä), joka on monelle tuttu seinätauluineen ja tunnistettavine värikkäine post-it lappuineen. Ideana Kanbanissa on ohjelmiston, tuotteen tai palvelun jatkuva kehittäminen lyhytprojekteina ja varsinaiseen lopulliseen maaliin tai lopullisesti valmiiseen tuotteeseen ei välttämättä edes pyritä. Kanbania käytetään tyypillisesti esimerkiksi jo olemassa olevan verkkopalvelun tai käyttöjärjestelmän pienkehityksessä.

Yhteistä ketterille menetelmille on nopea reagointi muutostarpeisiin ja uusien ominaisuuksien liittäminen mukaan kesken projektin. Ketterä kehittäminen so-

pii erityisen hyvin projekteihin, joissa voi käyttää luovuutta ja niiden suunnitelmat muuttuvat lennossa. Van Casterenin (2017) mukaan Agile-pohjaiset projektit perustuvatkin aina tietynlaiselle *kaaoksen kulttuurille*.

Agile-pohjaisen Scrumin sprintti eli projektiin liittyvän osatavoitteen toteutus määräajassa on pituudeltaan yleensä yhdestä neljään viikkoon. Tässä ajassa suoritetaan useita erillisiä taskeja eli tehtäviä, joista sprintti koostuu. Mukaan voidaan ottaa helposti uusiakin ominaisuuksia, tai ominaisuuksia ja vaatimuksia voidaan muuttaa. Yleinen tapa on pitää päivittäin Daily Scrum eli palaveri, jolloin sekä tekijät että projektin haltija tietävät, missä vaiheessa projektissa mennään (kuva 1).



Kuva 1. Scrumin toimintamalli (van Casteren 2017)

Agile-pohjaisten projektinhallintamenetelmien heikkoutena on, että salliessaan projektin nopeatkin muutokset, projektien kokonaiskeston arviointi on hankalaa (Forbes Advisor 2022). Toisaalta esimerkiksi avoimella lähdekoodilla tapahtuva valtavan käyttöjärjestelmän kehittämistyö voi kokonaisprojektina viedä vuosia tai jopa vuosikymmeniä. Kehittämistyön lähtötilanne voi olla usein tällaisissa projekteissa, että kohtuullisen valmiiseen runkoon liitetään modulaarisen, verkon kautta tapahtuvan ohjelmistokehityksen avulla jatkuvasti uusia tarvittavia ominaisuuksia, joita pienet kehitystiimit työstävät erillään (Koponen 2020, 17). Nämä ovat kuitenkin pääprojektin osaprojekteja, jotka toteutetaan Agile-pohjaisina, mahdollisesti Kanban -periaatteella. Tällaisessa kehitystyössä kokonaiskeston pituus tai sen tarkka arviointi ei ole enää oleellista.

Ketterät projektit ja niiden metodit ovat myös omalla tavallaan vahvasti tuotestettu tehtävänimikkeineen ja nimineen (Task, Sprint ja niin edelleen). Vaikka niiden käyttöönotto ja opetteleminen onkin helppoa, se vaatii silti asiaan kouluttautuneen tiimin, joka tuntee toimintatavan tai vähintäänkin projekti-päällikön, joka osaa toimia Scrum masterina eli on opetellut Agile-Scrumin metodit. Tällöin kynnys siirtyä käyttämään näitä menetelmiä saattaa olla olemassa.

3.2.2 Muut menetelmät

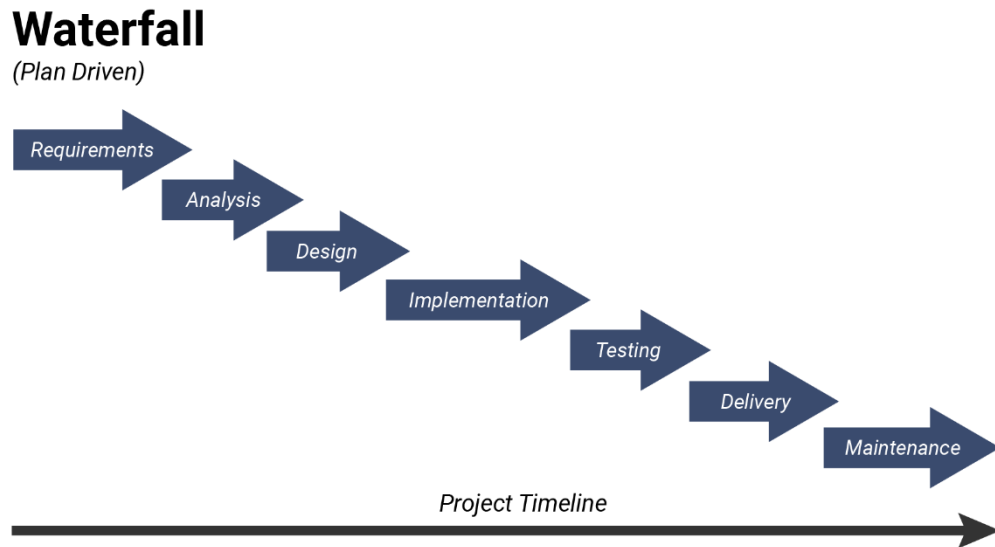
Muista projektinhallintamenetelmistä tunnetuin ja käytetyin on niin kutsuttu **Vesiputousmalli** (Waterfall). Mallin juuret ovat johdettavissa 60-luvun loppuun ja erääseen NATO-konferenssiin, jossa keskusteltiin ja suunniteltiin monimutkaisten kylmän sodan ajan tutkavalvontaan liittyvien tietokoneohjelmistojen käyttöönotoista. (Van Casteren 2017.)

Siihen saakka ohjelmistoja oli aina kehitetty ilman projektimaista työtapaa ja oli jo pitkän aikaa ilmennyt, että projektien koon ja monimutkaisuuden kasvaessa tämä alkoi olla työskentelytapana tavattoman hidas ja vaikeasti hallittava. Tarvittiin yhteinen malli kehitystyölle ja se saatiin tai tarkemmin otettiin valmistavasta teollisuudesta. Mallin tuotteisti Winston Royce 1970-luvun alun tieteellisessä artikkelissaan. Vesiputousmalli otettiin heti käyttöön ohjelmistokehityksessä, jonka yhtenä päämenetelmänä se toimi pitkälle 2000-luvulle. Vesiputousmallin suosiota on selittänyt sen nopea ja helppo käyttöönotto. (Huosianmaa 2020, 12.)

Malli on edelleen käytössä, vaikkei se ole enää yleisin ohjelmisto- tai muun kehityksen projektinhallintamenetelmä. Ideana on, että projekti kulkee virtaavan veden tavoin projektin vaiheesta seuraavaan. Seuraavaan vaiheeseen siirtyminen vaatii kuitenkin aina edellisen vaiheen valmistumisen ennen kuin tämä voidaan tehdä. Menetelmä on hyvin suoraviivainen, sen heikkoutena on projektimallin idea, että menneisiin vaiheisiin ei saisi palata ja vaikka käytännössä niihin palattaisiinkin, siitä tulee projektille aina ajallisia tai rahallisia kustannuksia. Vesiputousmalli perustuukin vahvasti suunnitelmallisuudelle. Jos tähän tulee projektin edetessä poikkeuksia tai korjauksia, malli ei välttämättä toimi koko potentiaalinsa vahvuudella. Kommunikaatio eri projektivaiheiden

välillä käydään tekemällä koko projektin ajalta dokumentaatiota. Tämä on oleellinen osa vesiputousmallia (Huosianmaa 2020, 12). Vesiputousmallia pidetään edelleen hyvin toimivana projekteissa, joissa on selkeä tavoite tai päämäärä (Forbes Advisor 2022).

Vesiputousmallissa vaiheet seuraavat lineaarisesti toisiaan (kuva 2).



Kuva 2. Vesiputousmallin toimintaperiaate (Actitime 2022)

Lisäksi on olemassa lukuisa joukko muitakin ei-ketteriä projektimenetelmiä, mutta niiden käsittely ei ole tässä opinnäytetyössä oleellista. Käytännössä projektimenetelmiä ei välttämättä myöskään käytetä niin puritaanisesti, kuin niiden ideaalina on: esimerkiksi vesiputousmallissa voidaan kyllä siirtyä takaisin aikaisemmalle vaiheelle ja sitä joudutaan usein projekteissa tekemäänkin.

Eri malleista voidaan yhdistelemällä tehdä myös niin sanottuja hybridejä, jolloin ne käyttävät ristiin toistensa elementtejä, jotka voivat olla ketteriä tai suunnitelmallisia. Vuonna 2016 tehdyn kansainvälisen kyselytutkimuksen mukaan tällaisten hybridimallien käytön osuus maailman ohjelmistoprojekteista oli miltei puolet: 45,3 %. (Vijayasathy & Butler 2016.)

4 TUOTTEISTAMINEN

Tuotteistaminen on käsite, jolle ei ole yhtään suoraa ja oikeaa määritelmää. Yleensä ihmiset kuitenkin tietävät tai hahmottavat mistä on kyse. Edellisessä

luvussa mainitut projektinhallintamenetelmätkin termeineen ja kiinteine metodeineen ovat tuotteistamisen tulosta.

Tuotteistamisesta voidaan puhua teollisuusprosessienkin yhteydessä, mutta yleensä siinä on kyse palvelun lisäarvon liittämisestä tuotteeseen. Luodaan jollekin palvelulle, prosessille, tai asialle yleinen ja yhteinen mielikuva ja toimintatapa. Tuote voi siis olla konkreettinen ja käsin kosketeltava, palvelu, tai näiden yhdistelmä. Tuotteistamisessa on kyse myös monistettavuudesta: oli tuote sitten fyysinen tai palvelu, se on nopeasti toistettavissa miltei samanlaisena.

Parantainen (2007, 13–14) esittelee kirjassaan kuusivaiheisen mallin siitä, miten palvelua tuotteistetaan. Mallissa ensimmäinen kohta on, että osa toimintamalleista alkaa erottumaan joukosta. Tuotteistamisen ensimmäisiä ituja on, kun asiantuntija alkaa monistaa joitakin projektissaan hyväksi havaitsemiaan toimintamalleja ja käytäntöjä, jotka toistuvat samankaltaisina hankkeista toiseen.

Toisessa vaiheessa palvelu alkaa hahmottua ja tässä vaiheessa liiketoiminnallisena mahdollisuutena voi miettiä jo esimerkiksi koulutuspalveluja. Kolmannessa vaiheessa palvelun toimitussisältö alkaa vakiintua. Seuraavaksi palvelun sisältö on vakiintunut jo niin paljon, että sitä voi alkaa monistamaan yrityksessä. Tämä on yksi ratkaisevista vaiheista.

Viidennessä vaiheessa palvelu alkaa monistua yrityksen ulkopuolella. Tämä edellyttää tietoista päätöstä, jossa palvelu dokumentoidaan niin, että ohjeistuksen avulla kuka tahansa pystyy suorittamaan saman. Kuudennessa ja viimeisessä vaiheessa palvelukonsepti on edennyt siihen vaiheeseen, että se muuttuu jälleenmyytäväksi. (Parantainen 2007, 13–14.)

Tiivistettynä tuotteistamista voisi kutsua idean tai palvelun paketoinniksi, jolloin se on helppo selittää, monistaa ja myydä eteenpäin. Myös asiakkaan on helpompi ostaa palvelu, kun hän tietää mitä on ostamassa tai saamassa. Sama pätee toisin päin: palvelua on yleensä helpompi myydä, kun tietää tarkalleen mitä on myymässä. Idea palvelee liiketoimintaa siten, että asiakas harvoin ostaa palvelua, jos siinä on hänen mielestään merkittävä *riskin tunne*, tai

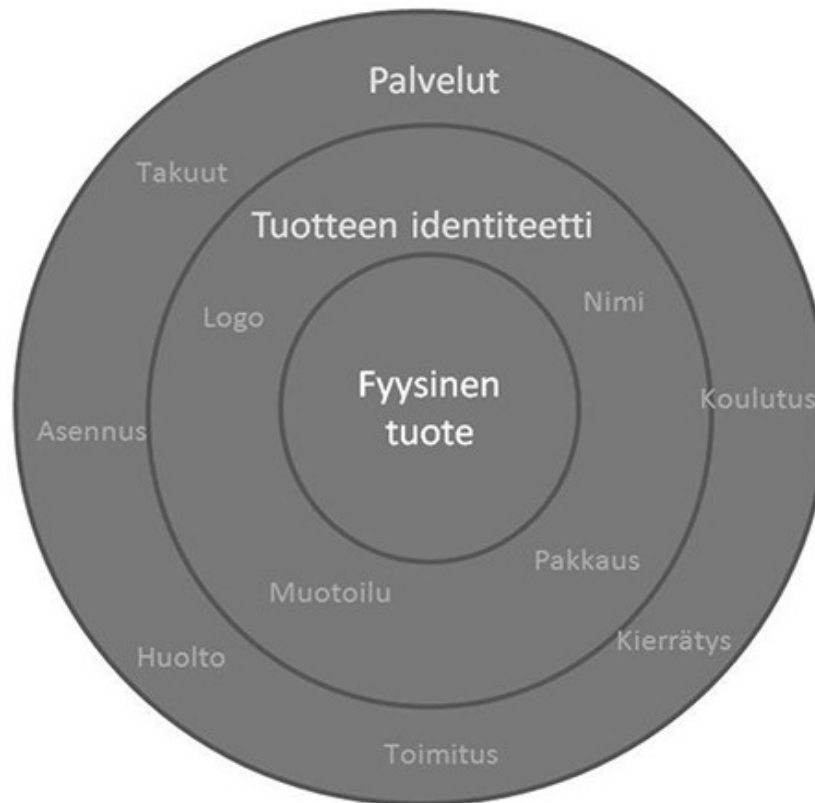
se on hänen silmissään liian monimutkainen ja monitulkintainen. Hyvin tuotteistettu palvelu vaikuttaa asiakkaan silmissä houkuttelevammalta ja luotettavammalta, kuin tuotteistamaton. (Parantainen 2007.) Halonen (2021) toteaa tuotteistamisen ytimeistä: ”Tuotteistamisprosessin kautta pyritään poistamaan epävarmuutta.”

Palvelumuotoilu

Palvelumuotoilu on yksi työkaluista, jolla tuotteistamista voidaan edistää. Yhteiskuntien siirtyessä yhä enemmän fyysisten tuotteiden teollisesta valmistuksesta palveluyhteiskuntaan, palvelumuotoilu on noussut liiketoiminnassa keskeiseksi erottautumis- ja kilpailutekijäksi.

Palvelut ovat länsimaissa jo kauan sitten ohittaneet teollisen tavarantuotannon (Grönroos 2009). Ne ovat myös laajentaneet fyysisen tuotteen määrettä antaen sille lisäarvoa tai kilpailuetua esimerkiksi verkkomyymälästä tilatun fyysisen tuotteen lähetyksenseurannan kautta. On arvioitu, että tuotteen ympärillä olevasta palvelutoiminnasta voidaan nykyisin saada jopa merkittävämpi osa liikevaihdosta, kuin itse fyysisestä tuotteesta (Hietikko 2021).

Laajennettu tuote koostuu fyysisen tuotteen lisäksi tuotteen identiteetistä (nimi, logo, muotoilu, pakkaus) ja sen ympärille tehdyistä palveluista (takuu, asennus, huolto, toimitus, kierrätys ja koulutus) (kuva 3).



Kuva 3. Laajennettu tuote. Tuotteen käsite 1.1. (Hietikko 2021)

Palvelumuotoiluprosessin tavoitteena on ymmärtää ja tunnistaa palvelun käyttäjien tarpeita. Tuulaniemi (2011, 24–25) toteaa kirjassaan palvelumuotoilusta: ”Palvelumuotoilu on osaamisala, joka tuo muotoilusta tutut toimintatavat palveluiden kehittämiseen ja yhdistää ne perinteisiin palvelun kehityksen menetelmiin. Palvelumuotoilu ei ole uusi innovaatio vaan pikemminkin tapa yhdistää vanhoja asioita uudella tavalla.”

Pääajatus on, että ihminen eli tässä tapauksessa käyttäjä tai tuotteen ostaja on tapahtuman keskiössä ja palvelumuotoilu tehdään juuri hänen tarpeistaan. Palvelumuotoilussa tärkeä asia on palvelun saavutettavuus, joka voi olla esimerkiksi palvelun löydettävyyttä verkossa (Karhapää ym. 2022) tai palvelun käytettävyyttä.

5 SYSTEEMIAJATTELU

Systeemiajattelun juuret ovat Antiikin Kreikassa, josta sana *systema* eli osista koostuva kokonaisuus on peräisin. Nykyaikaisen systeemiajattelun syntyhetki sijoittuu 1920-luvun biologian tutkimukseen, jolloin havaittiin, että luonnossa eliöiden ja kasvien elämä oli voimakkaasti sidoksissa kaikkeen elolliseen ja

elottomaan niiden ympäristössä. (Currie & Galliers 1999, 47.) Toinen systeemijatteluun voimakkaasti vaikuttanut tieteen suuntaus oli toisen maailmansodan jälkeen kehittynyt kybernetiikka. 1950-luvulle tultaessa teorian takana oleva ajatusmalli oli muokkautunut niin, että sen idea omaksuttiin muillekin tieteenaloille.

Systeemijattelussa on kysymyksessä teoria, jossa koko olemassa oleva maailma koostuu erillisistä systeemeistä, jotka mahdollisesti yhdistyvät tai linkittyvät toisiinsa (Kim 1999). Yksittäiselle systeemille on ominaista, että sen kaikki osat ovat suorassa tai välillisessä joskus monimutkaisessakin vuorovaikutuksessa keskenään, muodostaen itsestään elävän riippuvaisuuksien kokonaisuuden, jolle voidaan tarkastellessa kuitenkin määritellä rajat ja rajata se tietyn kokoiseksi.

Anttila (2014) toteaa systeemijattelusta tieteellisenä keinona, että siinä usein suoritettu systeemianalyysi on tutkimusmenetelmällisesti tehokas selitys- ja kuvausjärjestelmä pikemminkin, kuin syy-seurausmalli, jota muun muassa positivistinen tutkimusote edustaa. Siinä pienillä tekijöillä saattaa olla suuri vaikutus kokonaisuuteen. Niillä voidaan Anttilan mukaan saada aikaan synergiavaikeus ja ilmiötä voidaan tarkastella myös tietyissä ajanjaksoissa, kuten esimerkiksi sykleittäin, vaiheittain ja niin edelleen.

Systeemi on aina enemmän kuin osiensa summa. Systeemijattelussa tarkoituksena on niin sanottujen *piilovaikutusten* löytäminen ja ennen kaikkea niiden olemassaolon tiedostaminen. (Anttila 2014.)

Ekosysteemi

Systeemijattelusta on kehittynyt yleisesti käytetty termi ekosysteemi, jolla voidaan tarkoittaa sen alkuperäistä biologiantutkimuksen merkitystä eli rajattua biologista systeemiä sekä riippuvuuksien ja ristikkäisyyksien kautta toimivaa taloudellista systeemiä. Luonnon- ja talouden samankaltaisuus ei rajoitu pelkkään termiin sillä biologisen evoluution ja taloudellisen kehityksen välisistä yhtäläisyyksistä on keskusteltu jo yli vuosisata (Hytti & Ruusunen 2016). Sosiaali-tieteissä esimerkiksi kansantaloutta on aina lähestytty ekosysteeminä. Taloudellisessa ekosysteemissä yhteistyö ja vuorovaikutus mahdollistavat uusien

innovaatioiden synnyn ja tuottavat tuotteita ja palveluita asiakkaille. Toimijoiden välillä on sekä yhteistyötä että kilpailua. (Hytti & Ruusunen 2016, 28.)

6 TUTKIMUSOSIO

Opinnäytetyön tutkimuksellisenä lähtökohtana on kvalitatiivinen eli laadullinen tutkimus. Laadullinen tutkimus soveltuu tilanteisiin, joissa halutaan tutkia luonnollisia tilanteita, joita ei voi järjestää kokeeksi tai joissa ei voida kontrolloida läheskään kaikkia vaikuttavia tekijöitä. (Metsämuuronen 2000, 14.) Tutkimusongelmat tässä opinnäytetyössä ovat vahvasti ketjuuntuneita ja rinnakkaisia vaikuttavia tekijöitä voi olla useita, eikä niitä voida tarkasti määrittää tai kontrolloida, jolloin laadullinen tutkimus on perusteltu valinta lähtökohdaksi.

Kirjallisuuskatsaus kuuluu tutkimuksellisenä menetelmänä laadullisen tai laadullisen ja määrällisen tutkimuksen yhdistelmän alle. Kirjallisuuskatsausta käytetään tutkimuksessa erilaisista lähtökohdista. Kirjallisuuskatsaus voi kehittää olemassa olevaa teoriaa ja rakentaa uutta teoriaa. Sen avulla voidaan myös arvioida teoriaa. Kirjallisuuskatsaus voi **rakentaa kokonaiskuva**a tietystä asiakokonaisuudesta ja sillä voidaan pyrkiä tunnistamaan ongelmia. Se tarjoaa myös mahdollisuuden kuvata jonkun ilmiön historiaa. (Salminen 2011, 3.)

Tämän opinnäytetyön tutkimuskysymykset pyrkivät tunnistamaan juuri, mitä ongelmia tässä tarkastellussa ilmiössä ilmenee ja mitä ratkaisuja kokonais kuvan kannalta niihin voisi esittää. Tämä tukee kirjallisuuskatsauksen valintaa tutkimusmenetelmäksi.

6.1 Kuvaileva kirjallisuuskatsaus

Opinnäytetyön tutkimuksellinen osa on toteutettu kuvailevana kirjallisuuskatsauksena. Produktiivinen ja tutkimuksellinen osa perustuvat aiheesta löytyvään valmiiseen tutkimustietoon, kirjallisuuteen, materiaaleihin ja artikkeleihin.

Kirjallisuuskatsauksessa tehdään tutkimusta jo olemassa olevista tutkimuksista ja lähteistä. Kirjallisuuskatsauksia on erilaisia, mutta opinnäytetöissä yleisimmin on käytetty kuvailevaa kirjallisuuskatsausta, jota tämäkin opinnäytetyö edustaa.

Kuvaileva kirjallisuuskatsaus on yleiskatsaus ilman tiukkoja ja tarkkoja sääntöjä. Se sallii laajojen aineistopiirien käytön, eikä aseta tarkkoja metodistisia sääntöjä aineiston rajaukselle. (Salminen 2011, 5.) Tutkittava ilmiö kuitenkin pystytään tai se on pystyttävä kuvaamaan laaja-alaisesti. Narratiivista kuvailevan kirjallisuuskatsauksen muotoa on käytetty runsaasti erityisesti opetuksen alalla ja tässä mielessä metodin valintana se sopii hyvin opinnäytetyön produktiivisenkin osion pohjalle. Kuvaileva kirjallisuuskatsaus pystyy luomaan siltoja sinne, missä niitä ei ole ja se voi käyttää tähän keinona yleistämistä (Vilkkä 2023, luku 1.2.1: Narratiivinen kirjallisuuskatsaus).

Kaikille kirjallisuuskatsauksille on yhteistä, että ne sisältävät kirjallisuuden haun, kriittisen arvioinnin (kuvailevassa kirjallisuuskatsauksessa arviointi tehdään sen suhteen mitä valitset mukaan ja mitä rajata pois) sekä aineiston analysoinnin ja siitä tehdyn synteessin. (Jyväskylän ammattikorkeakoulu 2022.)

Kirjallisuuskatsauksella kartoitetaan siis, mitä valmista tietoa aiheesta tai aihepiiristä on jo olemassa. Tästä löydetyistä aineistosta voidaan hakea vastausta esimerkiksi opinnäytetyön tutkimusongelmaan.

6.2 Aineiston haku

Opinnäytteen tutkimusongelma ja tutkimuskysymykset ovat taustaltaan monia-
laisia, joten lähdekirjallisuuden etsiminen myös ulottuu useammalle kuin yhdelle osa-alueelle. Rajaan haettavan kirjallisuuden käsittämään opinnäytetyön kannalta sen keskeiset alueet, joita on neljä: avoimen lähdekoodin, projektityön, tuotteistamisen sekä systeemiajattelun.

Keskenään linkittyneet aihealueet ja niiden lähdemateriaali yhdistetään opinnäytetyön produktiivisessa osiossa synteessissä yhdeksi kokonaisuudeksi, josta muodostan produktiivisena osana olevan avoimen lähdekoodin projektiooppaan sisällön. Koska käsiteltävät osa-alueet poikkeavat toisistaan myös niistä saatavilla olevan lähdetiedon ja sen laadun kannalta, en tee aineistohakua yksiselitteisten ja tarkkojen rajausten kautta (esimerkiksi vain tieteellisiä artikkeleita, vain väitöskirjoja, vain vuoden 2021 jälkeisiä lähteitä ja niin edel-

leen). Tutkimusmenetelmänä käytetty kuvaileva kirjallisuuskatsaus on tässäkin suhteessa salliva (Vilkka 2023, luku 1.2.1: Narratiivinen kirjallisuuskatsaus).

Käytin kirjallisuuden ja lähteiden haussa apuna Kaakkois-Suomen ammattikorkeakoulu Kaakkurin hakutietokantaa, XAMKin omaa fyysistä kirjastoa, Google Scholaria ja Theseusta. Lisäksi sain COSS ry:ltä jonkin verran kirjallista ja suullista dataa, esimerkiksi käyttökelpoisen ja ajankohtaisen Kuntaliiton avoimen lähdekoodin oppaan. Theseuksesta hain AMK- ja YAMK-opinnäytetöitä, jotka sivusivat jotain neljästä aihealueesta. Näiden läpiluvun jälkeen kävin läpi vielä niiden omat lähdeluettelot ja poimin sieltä tutkimuslähteitä.

Avoimesta lähdekoodista on tehty jonkin verran opinnäytetöitä, mutta ne sivuavat aihetta enemmänkin tarkastelemalla sopisiko avoimen lähdekoodin ohjelmisto johonkin tiettyyn opinnäytetyössä tarkasteltuun projektiin. Tämä opinnäyte tarkastelee avoimen lähdekoodin toimintaa laajemmasta näkökulmasta ja tein päätöksen jättää lähteistä ammattikorkeakoulutasoiset aihetta koskevat opinnäytetyöt pois, säilyttäen kuitenkin YAMK-opinnäytetyöt. Tämä siksi, että opinnäytetyössäni niiden kanssa yhteneväiset osat koskivat lähinnä avoimen lähdekoodin peruskäsitettä, joka on avattu muulla tavalla.

Aineistohauulla löytyi kattava määrä ammattikirjallisuutta, tutkimuksellisia artikkeleita, opetusmateriaalia, tiedekorkeakoulujen opinnäytetöitä, uutisia, valtion- ja virastojen suosituksia mielestäni riittävän laaja-alaisen kokonaiskuvan luomiseen aihealueesta.

Hakusanoina käytin seuraavia: avoin lähdekoodi, open source, projektityö, project model, systeemiajattelu, system thinking, ekosysteemi, ecosystem ja tuotteistaminen.

Aineiston ajallisena rajauksena pyrin mahdollisuuksien mukaan käyttämään tuoretta tietoa, kun sitä on saatavilla. Käsiteltävän aihealueen ja ilmiön pääalueen kannalta tämä ei ole ongelma: avoin lähdekoodi on suhteellisen tuore asia ja ilmiö ja siitä kirjoitettu tieto on vanhimmillaankin 2000-luvun alusta. Ohjelmistoteknisen osa-alueen kohdalla pidin ajankohtaisuuden rajana lähdekirjallisuutta valittaessa 1990-luvun loppua.

6.3 Kirjallisuuskatsausten aineistot

Kirjallisuuskatsausten tieteellisesti painavimmat aineistot voidaan Vilkan (2023, luku 1.3: Kirjallisuuskatsauksen aineistot) mukaan ryhmitellä tieteellisesti vertaisarvioituun- ja vertaisarvioimattomaan tutkimukseen.

Vertaisarvioituun tutkimukseen perustuvia aineistoja ovat tieteelliset artikkelit, jotka ovat tieteellisissä aikakauslehdissä tai kokoomateoksissa julkaistuja alkuperäiseen tutkimukseen perustuvia tekstejä. Tällaisia ovat myös tieteelliset monografiat, jotka ovat yhden tekijän nimissä olevia yhtenäisiä tieteellisiä esityksiä ja perustuvat tutkimustyöhön. Monografia on yleensä myös aina aikaisemmin julkaisematon. Lisäksi vertaisarvioituun tutkimukseen perustuvina lähteinä pidetään esimerkiksi konferenssijulkaisuja, tiivistelmiä ja näiden tieteellisiä postereita. (Vilka 2023, luku 1.3: Kirjallisuuskatsauksen aineistot.)

Vertaisarvioimattomaan tutkimukseen perustuvia aineistoja ovat tutkimusorganisaatioiden tutkimukset, selvitykset ja raportit. Tällaisia tekevät esimerkiksi yliopistojen eri alojen tutkimuskeskukset, Suomen itsenäisyyden juhlarahasto Sitra sekä erilaiset ajatushautomot. Vertaisarvioimattomaan tutkimukseen perustuvina lähteinä pidetään myös julkisyhteisön tutkimuksia ja raportteja, joita tekevät esimerkiksi Terveystieteiden ja hyvinvoinnin laitos THL, Tilastokeskus, eri ministeriöt, Kansaneläkelaitos ja niin edelleen. Myös eri asiantuntijaorganisaatioitten tutkimukset, joita ovat esimerkiksi eri liittojen selvitykset ja raportit ovat vertaisarvioimattomaan tutkimukseen perustuvia aineistoja. Samaten vertaisarvioimattomat konferenssijulkaisut, niiden tiivistelmät ja tieteelliset posterit. (Vilka 2023, luku 1.3: Kirjallisuuskatsauksen aineistot.)

Näiden lisäksi kuvailevassa kirjallisuuskatsauksessa voi olla mukana myös **tieteelliseltä painoarvoltaan vähäisempää materiaalia**, kuten ammattikirjallisuutta, lehtiartikkeleita, uutisia, verkkosivustoja, oppimateriaalia ja niin edelleen.

Taulukossa opinnäytetyön kirjallisuuskatsauksessa käytettyjen lähteiden aineistotyyppit, lukumäärä sekä niiden vertaisarviointi. Tieteelliset artikkelit ja väitöskirja täyttävät vertaisarvioidun tutkimuksen määreet. YAMK-opinnäytetyö,

kandidaatin tutkielma, valtion ja virastojen raportit ovat vertaisarvioimatonta tutkimusta, joka on havainnollistettu taulukossa vaaleammalla punaisella. Ammattikirjallisuus, lehtiartikkelit, yhteisöjen verkkosivustot ja niin edelleen ovat myös vertaisarvioimattomia sekä tieteelliseltä painoarvoltaan vähäisempiä (Taulukko 1).

Taulukko 1. Opinnäytetyön kuvailevan kirjallisuuskatsauksen lähteiden tyypit

	Lukumäärä	Vertaisarvioitu
Tieteelliset artikkelit	8	Kyllä
Väitöskirjat	1	Kyllä
Opinnäytetyöt (kandidaatin tutkielma, YAMK)	4	Ei
Valtion ja virastojen, raportit, suositukset ja ohjeet	5	Ei
Ammattikirjallisuus	7	Ei
Lehtiartikkelit (uutiset, artikkelit)	2	Ei
Opetusmateriaali	2	Ei
Verkkosivustot (yhteisöt jne.)	5	Ei

Kirjallisuuskatsauksen lähdemateriaalin jakautuminen lähdetyyppien kesken eri aihealueittain (avoin lähdekoodi, projektityö, tuotteistaminen ja systeemijattelu) sekä aihealuetta käsittelevien lähteiden lukumäärä (Taulukko 2).

Taulukko 2. Opinnäytetyön kuvailevan kirjallisuuskatsauksen lähteiden aihealueet

	Avoin lähdekoodi	Projektityö	Tuotteistaminen	Systeemijattelu
Tieteelliset artikkelit	1	2		5
Väitöskirjat	1			
Opinnäytetyöt (kandidaatin tutkielma, YAMK)	1	2		1
Valtion ja virastojen, raportit, suositukset ja ohjeet	5			
Ammattikirjallisuus		3	4	
Lehtiartikkelit (uutiset, artikkelit)	1		1	

Opetusmateriaali			2	
Verkkosivustot (yhteisöt jne.)	3	2		
Yhteensä:	12	9	7	6

Kirjallisuuskatsauksen lähteet jakautuvat neljän aihealueen kesken ja lähteiden tyypit vertaisarvioituun- ja arvioimattomaan tutkimukseen sekä tieteellisesti painoarvoltaan vähäisempään materiaaliin, jota edustaa muun muassa ammattikirjallisuus, lehtiartikkelit, uutiset, opetusmateriaali, yhteisöjen verkkosivustot ja niin edelleen.

7 KIRJALLISUUSKATSAUKSEN TULOKSET

Onnistunut avoimen lähdekoodin projekti

Tämän opinnäytetyön ensimmäinen tutkimuskysymys oli: Minkälaiset huomioid ja tekijät johtavat onnistuneeseen avoimen lähdekoodin projektiin? Tarkastelussa on projektimuotoinen työskentely yleisesti, sen lähtökohdat, vaiheet ja mitä huomioita avoimen lähdekoodin kanssa työskentely mahdollisesti vaatii eri vaiheissa.

Jokainen projekti perustuu aina pohjimmiltaan johonkin **tarpeeseen**, joka projektilla pyritään täyttämään. Tämä tarve osaltaan rajaa ja määrittää projektin kohdistumisen ja laajuuden. (Mäntyneva 2016, 16.) Projektityön tarkastelussa yleinen tapa on jakaa projektin toteutus niin sanotussa elinkaarimallissa neljään eri vaiheeseen (Mäntyneva 2016, 5). Projekti on aina ajallisesti ja laajuudellisesti rajattu, sisältäen aloitus ja lopetuspisteen, jonka väliin elinkaarimallin neljä vaihetta sijoittuu (Löow 2002, 17).

Tarpeen tunnistamisen jälkeen projekti alkaa **valmisteluvaiheella**, jossa projektin suuntaviivat määritellään. Yrityksellä tai yhteisöllä saattaa olla sen toimintatapoja ohjaava ennalta määrätty toimintastrategia, johon avoin lähdekoodi saattaa kuulua tai olla kuulumatta. Esimerkiksi valtion tietojärjestelmistä on Valtiovarainministeriön suositus käyttää avoimen lähdekoodin ohjelmistoja (Valtiovarainministeriö 2009). Erityisen olennaista valmisteluvaiheessa on kuitenkin saada yrityksen tai yhteisön johdon tuki projektille. Mikäli tämä puuttuu

tai on puutteellista, projektin myöhempi eteneminen ja aikataulutus voi muuttua mahdottomaksi. (Ruuska 2007, 44.) Tässä vaiheessa projektille asetetaan sekä tavoite, että vastuullinen tilaaja.

Avoimen lähdekoodin projektin näkökulmasta valmisteluvaiheessa kannattaa selvittää, olisiko tarpeeseen kehitettävä yhteisötuote jo valmiina (esimerkiksi internetjulkaisuissa Drupal, ERP-järjestelmissä Odoo) (COSS 2023). Näin vältetään työläältä saman asian uudelleen tekemiseltä. Julkisen hallinnon neuvottelukunta JUHTA (2009) suosittelee myös oppaassaan pitäytymistä sellaisissa avoimen lähdekoodin ohjelmistoissa, jotka ovat jo kehityskaarellaan vakiintuneita, toiminnaltaan stabiileja ja joille on syntynyt laaja kehittäjäyhteisö.

Ohjelmiston käyttöönotto tai kehittäminen katsotaan monimutkaisuutensa takia sellaiseksi, että se vaatii käytännössä projektimuotoisen tai luontoisen työtavan eli projektinhallintamenetelmän. Tässä suhteessa avoimen tai suljetun lähdekoodin ohjelmistoprojektit eivät poikkea toisistaan. Tällaiseksi menetelmäksi kehitettiin esimerkiksi ohjelmistokehityksessä pitkälle 2000-luvulle asti pääasiallisena projektinhallintamenetelmänä käytetty ”vesiputousmalli”. (Van Casteren 2017.)

Produktiivisessa osassa oppaan projektinhallintamenetelmäpohjaksi valikoitui edellä mainittu vesiputousmalli. Perustin valintani siihen, että projektien kokoluokkien ollessa joko pieniä tai keskisuuria, kuten myös niiden tilaajien, kyseisessä projektinhallintamenetelmässä on useita hyödyllisiä ominaisuuksia. Huosianmaa (2020, 12) toteaa yleisesti projektinhallintamenetelmistä: ”Vaikka suurimmassa osassa ohjelmistoprojekteista käytetään jo ketteriä menetelmiä, perinteiset menetelmät ovat yhä olemassa ja niillä voi tietyissä tilanteissa saavuttaa etuja. Nämä edut voivat toteutua esimerkiksi silloin, kun muutoksia ei voi tapahtua.”

Vesiputousmalli perustuu täysin suunnitelmallisuudelle. Oleellista menetelmälle on myös, että projektin tavoitteet ja aikataulu ovat suurin piirtein selvillä ennen projektin alkua. Projektin suunnittelu alusta loppuun suoritetaan ennen sen aloittamista. (Finni 2022, 11.)

Vesiputousmallin etuna on, että se on nopea ja helppo omaksua sekä hallita, eikä sen käyttöönotossa tarvitse samanlaista projektikoulutusta, kuin esimerkiksi Agile-pohjaisessa Scrum:issa (Huosianmaa 2020, 12). Vesiputousmalli toimii pienissä projekteissa, joiden vaatimukset on helppo ymmärtää ja hanke on samalla nopea toimittaa maaliin (Actitime 2022).

Produktiivisessa osassa tähdätään, että siinä tehty ohjeistus rohkaisisi yrityksiä tai yhteisöjä pienien tai keskisuurien avoimen lähdekoodin projektien kokeylemiseen mahdollisesti pienenkin toimittajan kanssa. Jos oletetaan, että kyseessä voisi olla tilaajayrityksen ensimmäinen it-projekti ja toimittaja voi olla kokoluokaltaan pienyritys tai jopa yksinyrittäjä, vesiputousmallin suoraviivainen lineaarisuus ja selkeät askeleittain etenevät työvaiheet toimivat hyvänä tukikehikkona projektille, jonka kaikki osallistujat saattavat olla mahdollisesti ensikertalaisia projektityöskentelyssä. (Alshamrani & Bahattab 2015.)

Huosianmaan (2020, 12) esittämä vesiputousmallin heikkous eli sen joustamattomuus ja kankeus nopeisiin muutoksiin kesken projektin, eli sopeutumattomuus muuttuviin ympäristöihin voi olla myös etuna: tämä saattaa estää esimerkiksi yläjohdon tai ulkopuolisten tahojen haluamat, usein mahdollisesti huonostikin perustellut muutokset projektiin, jolloin projektin läpivienti ei kärsi viivästyksistä tai sitä ei ruveta paisuttamaan kesken projektia uusilla ominaisuuksilla. Mallin ideana on, että projekti suunnitellaan etukäteen ja toteutetaan sellaisenaan ilman suuria muutoksia, koska lopputulos on selkeästi määritelty, samoin budjetti ja vaiheet. (Forbes Advisor 2022.)

Avoimen lähdekoodin ideaaleihin kuuluu projektien dokumentointi (COSS 2023). Vesiputousmallin puolesta puhuu tässäkin yhteydessä portaittaiseen projektinhallintamenetelmään oleellisesti kuuluva jatkuva dokumentointi. Avoimen lähdekoodin dokumentoinnissa on kyse myös siitä, että ohjelman toiminnallisuudet ovat edelleen selvitetävissä, mikäli kehittäjä tai kehittäjäyritys vaihtuu.

Valmisteluvaiheen jälkeen projekti siirtyy **suunnitteluvaiheeseen**. Tässä vaiheessa määritellään projektin lopullinen laajuus ja kattavuus (Mäntyneva 2016, 25). Projektille määritellään tarkemmat tavoitteet ja sille osoitetaan ta-

loudelliset- ja henkilöresurssit. Tässä vaiheessa tarkastellaan kaikki vaihtoehtoiset ratkaisumallit, esimerkiksi kuuluuko avoin lähdekoodi niihin. Projektille tehdään projektisuunnitelma, jota on tarkoitus noudattaa koko projektin toteutuksen ajan. Projektin liittyvät lähtökohtaisesti myös aina riskit, jotka kartoitetaan projektin tässä vaiheessa (Ruuska 2007, 25). Ruuska toteaa (2007, 50), että onnistumisedellytykset lyödään pitkälle lukkoon jo projektin suunnitteluvaiheessa.

Projektisuunnitelman yksityiskohtaisuudelle ja yleisluontoisuudelle pitäisi löytää sopiva tasapaino. On tosiasia, että suunnitelmaa pitää yleensä aina päivittää projektin aikana. Tällöin liian tarkka ja yksityiskohtainen suunnitelma siihen laitettuihin työpanoksiin on hukkaa (Mäntyneva 2016, 41). Toisaalta liian karkeasti ja suuripiirteisesti tehty suunnitelma voi vaikeuttaa esimerkiksi projektin työmäärien arviointia. Projektista laaditaan kuvaus, määritellään välitavoitteet, miten sidosryhmiä tiedotetaan ja niin edelleen. Löw toteaa kirjassaan luvussa ”Projektin menestystekijät ja ansat” (2002, 39), projektin menestyksen kannalta vaaralliseksi sen, että projekti on lähtökohdiltaan sekava.

Opinnäytetyön produktiivisen osion projektiohjeistuksessa pyrittiin projektikoon rajauksella pois sulkemaan tämä sekavuuden uhka. Toisaalta Löw (2002, 40) toteaa myös, että yhtenä vaarana voi olla projektin suunnittelu virheellisesti liian pieneksi niin, että ilmenevät tarpeet ja muutokset paisuttavat sitä myöhemmin, jolloin se ei pysy enää sille varatussa aikataulussa ja työmäärissä. Tässä suhteessa projektien laajuuden ja tavoitteiden tarkahko ja realistinen rajaaminen auttaa ennakoimaan ja resursoimaan oikein. Suunnitelmallisuuden perustuvana projektinhallintamenetelmänä vesiputousmalli on tähänkin käytännöllinen.

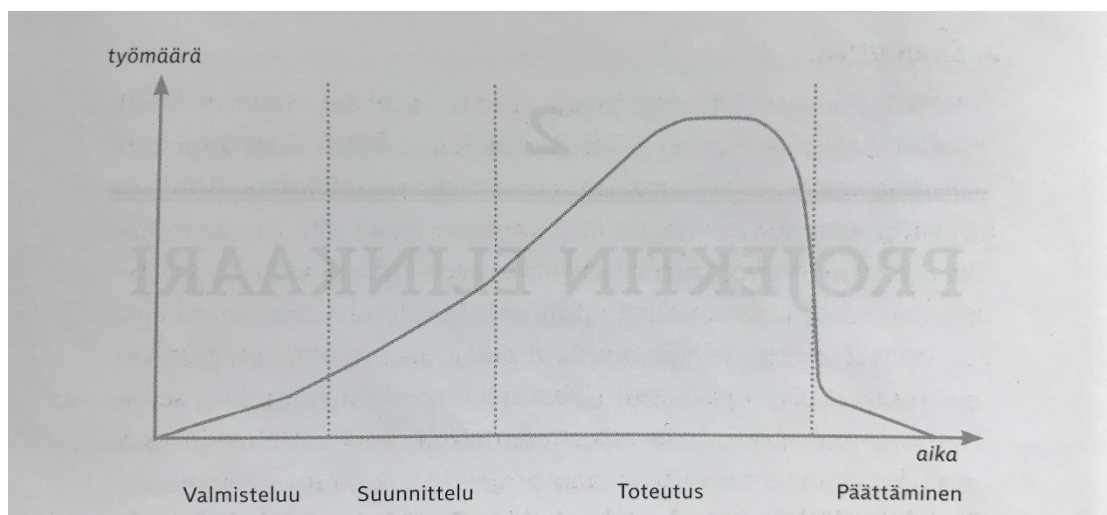
Avoimen lähdekoodin kannalta suunnitteluvaiheessa kannattaa tarkastella vielä valmisteluvaiheen jälkeenkin uudestaan, kehitetäänkö nyt jotain uutta, vai onko olemassa ennalta valmista yhteisötuotetta. Käytännössä tässä kohdassa on myös pakollista selvittää ja kartoittaa yrityksen tai organisaation oma osaaminen ja voimavarat palvelun kehittämistä, muokkaamista ja käyttöönottoa varten. Tämä on oleellista, jos avoimen lähdekoodin ohjelmiston käyttöönotolla haetaan niiden mahdollistamia kustannussäästöjä (tässä tapauksessa

esimerkiksi ulkoisesta tuesta säästämistä). (JUHTA 2009, 9–14.) Tässä suhteessa on myös hyvä huomioida valmiiden yhteisöpalveluiden tarjoamat yhteisölliset kehitysresurssit, jotka ovat vakiintuneiden ohjelmistojen kohdalla usein täysin riittävät (Kuntaliitto 2022, 18). Toisaalta edellä mainittujen yhteisöllisten kehitysresurssien riittävyyden arvioiminen voi olla tilaajayrityksen kannalta vaativaa ja hankalaa. Samoin myös avoimen koodin ratkaisumallin vertaaminen suljetun koodin ohjelmistoihin, joissa tuki saadaan yleensä maksullisella lisenssisopimuksella, jos kokemusta ei ole kummastakin.

Mikäli tässä vaiheessa päädytään avoimen lähdekoodin ohjelmistoon, oli se valmistuote tai täysin räätälöity asiakasyritykselle, tulee varmistaa yrityksessä tai yhteisössä itsessään riittävä koulutus sen käyttöönotolle. Samalla tulee arvioida, miten projektin päättymisen jälkeinen ylläpito huolehditaan ja riittääkö esimerkiksi pelkkä julkaisuyhteisön tuki tai oma osaaminen myöhemmin ilmenevissä ongelmatilanteissa. (JUHTA 2009, 17.) Avoimen lähdekoodin projektin osalta on myös oleellista kirjata suunnitteluvaiheessa ylös vaatimus projektin ja koodin dokumentoinnista, mistä on hyötyä ohjelmiston jatkokehityksen tai takuukorjausten kanssa, etenkin jos niitä suorittaa jokin muu kuin alkuperäinen toimittaja.

Suunnitteluvaiheen päätyttyä projekti siirtyy **toteutusvaiheeseen**, jossa noudatetaan projektisuunnitelman projektikuvausta ja sen työskentelymuotoja. Suunnitelmassa on kuvattu ylös projektiin nimetyt henkilöt, heidän vastuunsa, kokoukset, välitavoitteet, tiedottaminen ja niin edelleen. Mikäli toteutusvaiheessa ilmenee, että projektisuunnitelmaa on muutettava, tehdään tarvittavat toimenpiteet. Muutoksia tehdessä tulisi aina tarkastella, onko muutos projektille välttämätön, vai ylimääräinen lisäys. Tehdyt muutokset tulisi myös aina dokumentoida projektiohjeistoon. (Mäntyneva 2016, 105.) Mikäli projekti on kooltaan vähänkään laajempi, välitavoitteet ja niiden seuranta on tärkeää, sillä ne auttavat pitämään projektin kasassa ja antavat tilaisuuden tarkastella kokonaisuutta. Välitavoitteet parantavat myös työmotivaatiota (Ruuska 2007, 53).

Toteutusvaihe on projektin elinkaaren työkeskeisin vaihe. Alla oleva kuvaaja havainnollistaa, kuinka suurin osa projektin kokonaistyömäärästä sijoittuu toteutusvaiheeseen (kuva 4).



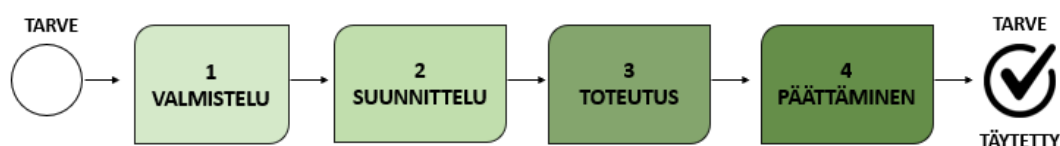
Kuva 4. Esimerkki projektin elinkaaren vaiheista ja työmäärästä (Mäntyneva 2016, 16)

Toteutusvaiheessa ilmeneviä yleisiä ongelmia ovat sekä avoimissa että suljetusta ohjelmistoissa projektin epäselvä rajausta ja siitä johtuva paisuminen. Jatkuvat muutostarpeet ja rajauksen tarkistukset voivat olla merkki huonosta suunnittelusta (Ruuska 2007, 42–43). Toteutusvaiheessa on hyvä huomioida, että ongelmakohtaisissa vakiintuneiden ohjelmistojen kohdalla avoimen lähdekoodin ohjelmilla on olemassa vahva yhteisötuki foorumeineen, joiden kautta voi hyödyntää sekä paikallista, että globaalia osaamista. (Kuntaliitto 2022, 18.) Tämä kaikki, yhdessä käyttöönoton kanssa vaatii kuitenkin tilaajaryitykseltä tavanomaista enemmän omaa osaamista ja asiantuntemusta (JUHTA 2009, 12).

Viimeisenä on projektin **päätösvaihe**. Se nivoo projektin projektiksi eli hankkeeksi, jossa tietty tarve on tyydytetty tietyssä rajatussa aikataulussa. Päätösvaiheessa viimeistellään kaikki projektiin liittyvät tehtävät. Tämä koskee niin toteutusta, kuin hallinnointiakin. Projektin päättämisessä keskeinen onnistumisen kriteeri on, että toimeksiantaja hyväksyy projektin tuotoksen ja lopputuloksen (Mäntyneva 2016, 143). Loppuraportti on projektin viimeinen työvaihe. Se ei ole historiikki tai tuoteseloste, vaan yhteenveto siitä mitä projektissa tapahtui, mikä meni oikein ja mitä kehitettävää ilmeni sekä arvio hankkeen onnistumisesta kokonaisuutena. Usein tähän liittyy vielä tieto, kenelle mahdollinen jatko-työ kuuluu. Ruuska toteaa kirjassaan (2007, 271), että projekti on aina oppimisprosessi. Projektin virallinen päättäminen on siinäkin mielessä tärkeää, että projektille saattaa, resursoinnin, aikataulutusten epäonnistumi-

sen, virheellisen työmääräarvion tai toteutusvaiheessa huomattujen puutteiden takia venyä niin kutsuttua *häntää* eli kesken jääneitä tehtäviä, jotka sitovat tekijänsä edelleen projektiin ja kuluttaa jatkossakin resursseja, kuten henkilötyöaika. (Mäntyneva 2016, 18.)

Projektin elinkaaren neljä vaihetta: valmistelu, suunnittelu, toteutus ja päättäminen (kuva 5).



Kuva 5. Projektin elinkaari

Projektin päätösvaiheessa avoimen lähdekoodin projektit eivät poikkea muista. Ideaali on tässäkin tapauksessa oppia jotain projektista, toistaa sen onnistumiset ja välttää epäonnistumiset. Huomioitavaa avoimen koodin tapauksessa on, että koodin jakamisen lisäksi kannattaa myös jakaa yleistä tietoutta projektin ja alustan onnistumisista tai mahdollisesta epäonnistumisista yhteisölle sekä yhteisön ulkopuolelle. Avoimen lähdekoodin idea perustuu alusta loppuun koko kehitystyön ja yhteisön avoimuudelle. (COSS 2023.)

Tutkimuskysymyksessä pohdittiin onnistunutta projektia. Käsitteenä **projektin onnistuminen** tai sen arviointi on loppujen lopuksi aina pitkälti mielipidekysymys (Ruuska 2007, 275).

Projektilla on yleensä useita saman aikaisia tavoitteita, jotka voivat olla ristiriidassakin keskenään tai niiden tärkeysjärjestys voi projektista toiseen vaihdella. Mutta yleisesti näitä voidaan määritellä neljä kappaletta:

- sisällölliset ja laadulliset tavoitteet
- toteutukselliset tavoitteet
- taloudelliset tavoitteet
- ajalliset tavoitteet

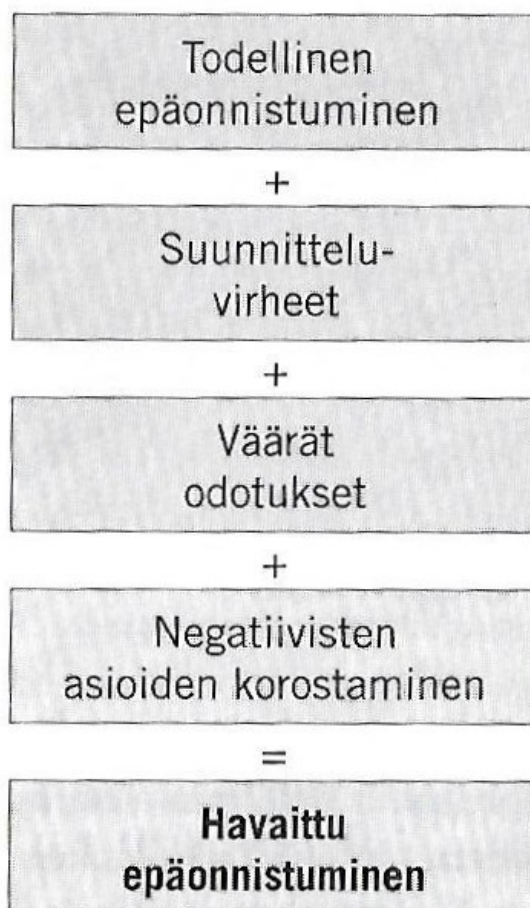
Näiden toteutumisen arviointiin vaikuttaa puhtaan mitattavuuden (numeeriset taloudelliset ja ajalliset tavoitteet) lisäksi vahvasti mielipidepohjaisia seikkoja.

Esimerkiksi laadullisten tavoitteiden mittaaminen on usein mahdotonta, ellei sille etukäteen sovita tiukkoja kriteereitä. (Ruuska 2007, 277.)

Tavoitteiden lisäksi tilaajaa ohjaavat myös tiedostamattomat odotukset, jotka voivat vääristää tai värittää projektista saatavaa mielikuvaa. Tätä lähdettiin torjumaan opinnäytetyön produktiivisen osan ohjeistuksen projektikoon rajauksella, sillä selkeän kokoinen ja rajattu projekti jättää vähemmän tilaa tulkinnoille ja väärille odotuksille tai niitä ei synny lisää projektin kasvaessa ja edessä. Tämä minimoi eron sen välillä, mistä asiakas luulee puhuvansa ja mitä hän saa (Ruuska 2007, 278).

Projektin epäonnistumisen tunteeksi voi riittää, että tavoitteista vain pelkääntään yksi epäonnistuu muiden samaan aikaan onnistuessa. Epäonnistuminenkin tai sen tunne on yleensä monen tekijän summa.

Todellinen epäonnistuminen voi tapahtua mahdollisesti ainoastaan projektin yhdellä osa-alueella, mutta suunnitteluvirheet, tilaajan virheelliset odotukset ja näihin huomion kiinnittäminen sekä niiden korostaminen vahventaa epäonnistumisen tunnetta, jolloin koko projektin koetaan epäonnistuneen (kuva 6).



Kuva 6. Mitä projektin epäonnistuminen tarkoittaa? (Ruuska 2007, 283)

Yhteenvedona voi todeta, että mikäli kaikki tai suurin osa aikaisemmin mainituista neljästä tavoitteesta täyttyy projektissa tyydyttävästi, projektia voi pitää onnistuneena. Jos aiempaan esitetyt eri projektivaiheiden huomioidut sekä avoimen lähdekoodin erityispiirteiden huomioiminen näissä edesauttavat tätä päämäärää, ne myös todennäköisesti edesauttavat projektin onnistumista.

Avoimen lähdekoodin liiketoiminnan kasvattaminen

Toisena tutkimuskysymyksenä on: Mikä edistäisi pienten ja keskikoisten toimijoiden ja projektien kohdalla liiketoimintaa avoimen lähdekoodin parissa? Tarkastelun alla oli yleinen liiketoiminnan edistäminen avoimen lähdekoodin yhteydessä. Tämä pohdinta rajataan opinnäytteessä pieniin ja keskikokoisiin toimijoihin.

Avoin lähdekoodi ei ole ilmiönä enää aivan uusi ja sen tuleamista yleiseen käyttöön on odotettu Suomessa pitkään. Kauppalehden verkkoartikkelissa (Avoin lähdekoodi on lyömässä läpi uuden lain ansiosta 2010) yli kymmenen vuoden takaa odotettiin avoimen lähdekoodin käytön viimeinkin yleistyvän, eritoten uuden hankintalain ansiosta. Tuolloin Microsoftilla yrityksenä oli ohjelmistoineen vankkumaton asema Suomen valtionhallinnon tietojärjestelmissä. Artikkelissa syitä open source -ohjelmistojen hitaalle leviämiselle oli useita: muutoksen pelko, puutteelliset tiedot, hankintalainsäädäntö ja ohjauksen puute. Mahdollisina hyötyinä avoimen lähdekoodin ohjelmistojen ja järjestelmien käytössä nähtiin tuolloin valtion it-kustannusten aleneminen. Tässä vaiheessa valtion it-menot olivat lähes tuplaantuneet vuosien 2000–2008 välillä.

Niin sanottu hankintalaki eli laki julkisista hankinnoista ja käyttöoikeussopimuksista muuttui vuonna 2016 ja on osaltaan ilmeisesti helpottanut ja selkeyttänyt hankintaa koskevaa lainsäädäntöä. Julkisissa hankinnoissa on valittava *kokonaistaloudellisesti* edullisin tarjous. Kokonaistaloudellisesti edullisin tarjous on se, joka on hankintayksikön kannalta hinnaltaan halvin, kustannuksiltaan edullisin tai hinta-laatusuhteeltaan paras. Hankintayksikkö voi itse asettaa hinta-laatusuhteeseen vertailuperusteita, jotka liittyvät laadullisiin, yhteiskunnallisiin, ympäristö- tai sosiaalisiin näkökohtiin tai innovatiivisiin ominaisuuksiin. Tämä antaa hankintayksikölle huomattavasti vapaammat kädet aikaisemman pelkän rahallisen hankintahinnan vertailemisen sijaan. (Laki julkisista hankinnoista ja käyttöoikeussopimuksista 29.12.2016/1397, 10. luku 93. §. mom. 1–2.)

Julkiselle puolelle on annettu valtio- ja EU-tasoiset suositukset avoimen lähdekoodin käytöstä ja suosimisesta (Valtiovarainministeriö 2009) (Euroopan parlamentti 2020, 7). Tilastokeskuksesta ei kyselyistä huolimatta löytynyt avoimen lähdekoodin osuuksista it-hankkeissa lähivuosilta tuoretta tutkimustietoa, mutta yleinen näkemys on, että näitä ohjelmistoja ja järjestelmiä ei käytetä niin paljon, kuin suositusten mukaan on tarkoitettu (Open Memorylab 2023).

Julkisen puolen odotukset avoimen lähdekoodin ohjelmistoille ovat kovia. Vahvuuksina ja mahdollisuuksina avoimen lähdekoodin ohjelmilla julkisella puolella katsotaan olevan (JUHTA 2009, 9–10):

- Ohjelmistokehityksen nopeutuminen
- Paikallisten toimijoiden kilpailuedellytysten parantaminen
- Toimittajariippuvuuden pieneneminen ja vapaa kilpailutus
- Yhteiskunnallinen riippumattomuus
- Kustannussäästöt
- Kansalaisten yhdenvertaisuuden paraneminen
- Lähdekoodin tarkistettavuus

Julkinen puoli on myös mahdollistanut jo pitkään keräämänsä avoimen datan käytön myös kaupallisella puolella jakamalla sitä, kuten esimerkiksi johdannossa mainitut Maanmittauslaitos ja Suomen ympäristökeskus SYKE ovat tehneet. Valtion omistajaohjauksessa toimiva Fintraffic on esimerkiksi perustanut vuonna 2021 Liikenteen dataekosysteemihankkeen, jossa se tavoittelee 160 yhteistyöyrityksen ja organisaation kanssa uusien toimintamallien, liikenteen yritysten kasvattamisen, tiedon jakamisen ja yhdessä kehittämisen kulttuuria. (Fintraffic 2022.)

Yllä olevat mahdollisuudet ja vahvuudet koskevat julkisen puolen lisäksi myös yksityisen puolen toimijoita. Lista avoimen lähdekoodin käytön eduista on houkutteleva, joten jo tämän pelkästään pitäisi puhua avoimen lähdekoodin käytön puolesta, mutta sen kasvu ei ole ollut vauhdikasta.

Nissilä (2016) on tutkinut ilmiötä väitöskirjassaan, joka käsitteli kehitysmaiden ICT-hankkeita (Tieto- ja viestintäteknikka hankkeet). Tavanomaiset ICT-hankkeet ovat tähän saakka pääsääntöisesti epäonnistuneet kehitysyhteistyöprojekteissa. Avoimen lähdekoodin ohjelmistoja on pitkään pidetty ratkaisuna tähän ja Nissilä otti tarkasteluun erään YK:n alaisuudessa Keniassa toteutetun avoimen lähdekoodin projektin.

Ilmeni, että ohjelmistot otettiin pilottihankkeissa hyvin käyttöön, mutta niiden skaalautuvuutta ei hyödynnetty, eikä niiden käyttö tämän jälkeen juurikaan levinnyt kokeiluosuuskuntia eteenpäin. Ongelmaksi muodostui, että loppukäyttäjät tarvitsivat apua ohjelmistojen asennukseen, käytön tukeen ja uusien ominaisuuksien kehittämiseen. Ohjelmistojen pelkkä hyödyllisyys ja ilmaisuus ei siis yksin riitä, vaan niille pitäisi syntyä paikallinen tai vaikkapa verkon välityk-

sellä taloudellisesti toimiva aktiivinen, kiinnostunut kehittäjäyhteisö, joka muodostaisi taas oman liiketoimintaekosysteemin, joka painollaan myös levittäisi avoimen lähdekoodin ratkaisuja eteenpäin. (Nissilä 2016, 88.)

Tästä kaksisuuntaisesta vaikutuksesta (Two-Sided Network Effects) on esitetty, että kasvu markkinoiden kummalla tahansa puolella (tilaajien tai toimittajien) johtaa yleensä koko alan liiketoiminnan kasvuun. Teorian mukaan liiketoiminta toisella puolella ruokkii siis liiketoimintaa myös toisella puolella. (Parker & Alstyne 2005, 1497.)

Sama pätee myös toisinpäin: avoimen lähdekoodin ratkaisujen huono tai vähäinen tunnettavuus ”kutistaa” sekä jarruttaa projektien syntyä. Tarkisteltaessa projektin elinkaaren vaiheita huomasiin, että valmisteluvaiheessa avoimen lähdekoodin käytön osalta on tunnistettavana ongelmana sen tarjoamien ratkaisuiden vähäinen tunnettavuus; yritykset sekä niiden johto turvautuvat mieluummin vanhoihin tuttuihin suljettuihin ohjelmistoihin ja niiden kehitysprojekteihin. (Kuntaliitto 2022, 13.) Kuntahankinnoissa myös avoimen lähdekoodin toimittajien ja osaamisen saatavuus sekä ohjelmistotuotteen myöhemmin tarvitseva tuki koetaan puutteellisiksi tai tunnetaan puutteellisesti. Tämä johtaa siihen, että hankintatilanne koetaan usein epätyypilliseksi ja hankintapäätös jää tästä syystä tekemättä. (Kuntaliitto 2022, 14.)

Jos vertaa yksityisten yritysten vastaavaan valintatilanteeseen, missä virheinvestointi voi olla taloudellisesti jopa kohtalokasta, on selvää, että hankintapäätöksiä tehdään vielä varovaisemmin. Yrityksillä liittyy myös epävarmuustekijöitä avoimen lähdekoodin lisenssien ja kaupallisen toiminnan yhdistämiseen. Parantainen (2017) nostaa tuotteistamista käsittelevässä kirjassaan **riskintunteen** termin, joka voi olla usein riittävä syy sivuuttaa toimittajavalinnoissa avoimen lähdekoodin vaihtoehto. Toisaalta avoimen lähdekoodin ohjelmat ovat lisensseiltään ilmaisia, joten pilotointi hoituu tässä mielessä kuitenkin varsin pienellä riskillä ja ratkaisusta pääsee suhteellisen kivuttomasti eroon, mikäli ilmenee, ettei se olekaan toimiva (JUHTA 2009, 9).

Koska ongelmakohta on edellä selkeästi tunnistettu, lähdin lähestymään toista tutkimuskysymystä tätä kautta. Systemiset ongelmat vaativat systemisiä

ratkaisuja (Koskinen 2018, 26). Koska kotimaankin avoimen lähdekoodin tilanne on pohjimmiltaan samankaltainen, kuin Nissilän (2016) väitöskirjassa esityttämä tilanne, kasvulle suurimman esteen asettaa tällä hetkellä avoimeen lähdekoodiin liittyvän ekosysteemin pienuus.

Ekosysteemin suora kasvattaminen esimerkiksi ulkopuolisen rahoituksen avulla on koettu aina haasteelliseksi, sillä ekosysteemissä on useita eri yrityksiä, joilla kaikilla on erilaiset omat, sisäiset taloudet ja tavoitteet (Hytti & Ruusunen 2016, 9). Kun ekosysteemi viimein syntyy, siinä yhdistyvät omalla painollaan teknologiaa, suunnitelmia ja sopimuksia sekä tuotteita ja palveluita. Nämä yhdessä muodostavat infrastruktuurin eli alustan, johon uudet asiakkaat ja hyväntekijät voidaan liittää. (Moore 2016, 31–58.)

Yksi ratkaisu on yrittää kasvattaa pientoimittajien ja tilaajien määrää, joka taas omalta osaltaan ruokkisi koko alan kasvua ja luonnollisesti samalla myös liiketoimintaa. Teorian kautta tarkasteltuna avoimen lähdekoodin tunnettavuuden lisääminen lisää sillä suoritettavien toteutusten todennäköisyyttä. Samalla kysynnän pitäisi lisätä pitkällä aikavälillä tarjontaa. (Parker & Alstyne 2005, 1497.) Oman haasteensa tai erikoisuutensa avoimen lähdekoodien osalta tuo kuitenkin se, että sen ekosysteemi rakentuu tavalla tai toisella pitkälti vapaaehtoisten kehittäjien harteille. Kehittäjiä voi olla jopa miljoonia, mutta he ovat jakautuneena verkkoon. (Moore 2006, 17.) Tästä voi seurata tilanne, jossa tarvittava kehittämisapu tai tuki olisi kyllä saatavissa verkosta, mutta yritys voi silti haluta niin sanottua lähitukea ja tämä koetaan ongelmalliseksi, samoin kun tilaajan ja toimittajan kohtaaminen. (Koponen 2020, 19.)

Liiketoimintamalleissa avoimen ja suljetun ohjelmiston välillä on ero siinä, että suljetuilla ohjelmistoilla se perustuu pitkälti lisenssimaksuihin ja näiden jakelutapaan. Avoimen lähdekoodin ohjelmissa ansainta perustuu pääasiassa palveluihin. (JUHTA 2009, 8.) Palveluliiketoimintaa voi olla muun muassa koulutus, maksettu kehittämistyö taikka ohjelmistojen tukitoiminnot. Huolimatta yhteisöltä saatavasta tuesta, moni yritys tai yhteisö saattaa olla valmis maksamaan niin sanotusta ”virallisemmasta tuesta” ja tämän tuen puuttuminen on yksi syy suljettujen ohjelmistojen suosimiseen. (Koponen 2020, 19.) Virallinen ja kaupallisesti saatava tuki ei puutu pääsääntöisesti kuitenkaan kaikista avoi-

men lähdekoodin ohjelmistoista. Suurista avoimen lähdekoodin yrityksistä esimerkiksi Red Hat Linux perustaa liiketoimintansa täysin sinänsä ilmaiseksi saatavan ja asennettavan Linux-käyttöjärjestelmän ja sen seuraajien tukeen ja konsultointiin (Koponen 2020, 20).

Pienemmissä kehitysprojekteissa, joita opinnäytteen produktiivinen osakin koski, kaupalliseen tukeen ja kumppanuuteen saattaisi hyvin riittää pienikokoinen toimija. Pientoimittajien ja pienprojektien synty palvelisi myös aikaisemmin mainittua avoimen lähdekoodin ideaalia paikallisten toimijoiden kilpailuedellytysten parantamisesta. (JUHTA 2009, 9–10.) Tämä voisi parhaimmillaan johdattaa pitkäaikaiseen molemminpuoliseen, kasvavaan kumppanuuteen, joka toisi mukaan osapuolten yhteistä sparrausta ja kehittämisvoimaa. Tämä taas hyödyttäisi osapuolten lisäksi koko yhteisön innovointia ja kasvua.

Tässä kohtaa päädyin ratkaisuun, jossa pyritään tuotteistamaan avoimen lähdekoodin projekti- tai hanketyöskentely, joka on tässä tapauksessa kokoluokaltaan pieni tai keskikokoinen. Tuotteistettu toimintatapa on helpompi esitellä, myydä ja ottaa käyttöön (Parantainen 2007).

Tuotteistamisessa palvelujen kohdalla ensimmäinen askel on, että osa toimintamalleista alkaa erottumaan joukosta. Tuotteistamisen ensimmäisiä ituja on, kun asiantuntija alkaa monistaa joitakin projektissaan hyväksi havaitsemiaan toimintamalleja ja käytäntöjä, jotka toistuvat samankaltaisina hankkeista toiseen. (Parantainen 2007, 13.)

Opinnäytetyön produktiivisessa osassa lähdettiin liikkeelle palvelumuotoilun perusideasta, että asiakas on sen keskiössä ja palvelu rakennetaan hänen näkökulmastaan (Alhonen & Iloranta 2021). Tuotteistamisessa lähdettiin jalostamaan tätä ideaa niin, että siinä pyrittiin luomaan projektien niin sanotuista ”parhaista käytännöistä” koottu ohjeisto, jota voidaan jakaa potentiaalisille asiakkaille, tässä tapauksessa asiakasyritysten tai yhteisöjen projektien tilaajille. Ohjeiston sisältö perustuu pääosin opinnäytetyön päätutkimuskysymyksestä: ”Minkälaiset huomiot ja tekijät johtavat onnistuneeseen avoimen lähdekoodin projektiin?” tehtyyn selvitykseen.

Paketoitu projektimalli huomioineen pystyy antamaan vastauksia avoimen lähdekoodin projekteissa yleisesti askarruttaviin kysymyksiin tai rohkaisee selvittämään tarkemmin avoimen lähdekoodin mahdollisuuksia. Samalla se lisää omalta osaltaan tietoutta avoimesta lähdekoodista, sen tunnettavuutta sekä toimii jo pelkän olemassaolonsa kautta yhtenä ekosysteemin osana.

8 JOHTOPÄÄTÖKSET JA YHTEENVETO

Tutkimusongelmina olivat; mitkä huomiot ja tekijät johtavat onnistuneeseen avoimen lähdekoodin projektiin sekä mikä edistäisi liiketoimintaa avoimen lähdekoodin ympärillä. Tutkimusosion rinnalla lopputyössä syntyi produktiona projektiohjeisto pienille ja keskisuurille yrityksille avoimen lähdekoodin projekteista ja niiden parhaista käytännöistä.

Projektityöstä sekä avoimen lähdekoodin projektien erityispiirteistä ja haasteista saatiin kirjallisuuskatsauksessa kattava kuva. Tämän kokonaiskuvan perusteella tehtiin produktiona COSS ry:lle projektiohjeistus.

Systeemiajattelusta ja ekosysteemistä löytyi tieteellisesti vertaisarvioitua lähdemateriaalia ja sen käsittelyssä olemassa olevan tieteellisen tutkimuksen käytöllä lähteissä oli suurin painoarvo. Ekosysteemin ja samalla liiketoiminnan kasvattamiselle löytyi kirjallisuuskatsauksen materiaalissa keinoja, mutta myös haasteita. Kirjallisuuskatsauksesta pystyi tekemään johtopäätöksen, missä avoin lähdekoodi kohtaa ongelmia projektipolussa ja miten tätä ongelmaa voisi lähestyä. Deduktiivinen lähdemateriaalin analysointi lähtee ajatuksesta, että tarkasteltava teoria pätee yleisestä yksityiseen (ja samalla myös toisinpäin). Analyysin mukaan yleispätevä ilmiö pätee siis myös yksilötasolla.

Liiketoiminnan kasvattaminen vaatii ekosysteemin kasvattamista. Tämän kasvattaminen esimerkiksi suoralla ulkopuolisella rahoituksella tai tuella ei välttämättä tuota haluttuja lopputuloksia, sillä terve ekosysteemi rakentuu monimutkaisesti omien syy- seuraussuhteidensa verkoston kautta ja omalla painollaan. Kirjallisuuskatsauksessa löytyi tähän mahdollinen ratkaisu avoimen lähdekoodin ratkaisujen tunnettavuuden lisäämisellä, joka omalta osaltaan taas ruokkisi sekä tilaajien että samalla toimittajien lukumäärää molempaan suuntaan. Tätä päämäärää pyrkii myös tukemaan produktiona syntynyt projektiohjeistus.

9 POHDINTAA

Pohdinnassa käydään läpi opinnäytetyön prosessia, työn eettisyyttä sekä luotettavuutta ja työn merkitystä sekä jatkotutkimus mahdollisuuksia.

9.1 Opinnäytetyöprosessi

Ajatus opinnäytetyön tekemisestä syntyi helmikuussa 2023, kun vuosikurssimme sähköpostilistalle tuli ilmoitus Open MemoryLab-hankkeen hakemasta opinnäytetyön tekijästä. Rohkaisevana tekijänä hankkeeseen osallistumiseen koin aikaisemmassa työharjoittelussa käyttämäni avoimen lähdekoodin kokemuksen sekä aikaisemman it-alan työkokemukseni. Koin aluksi, että aihe sivuaa hyvin vähän logistiikan koulutusta, mutta Open MemoryLabin edustajan kanssa käydyssä Teams-palaverissa ilmeni, että aiheessa sivutaan paljon projekti- ja prosessitoimintatapoja, jotka kuuluvat vahvasti kaikkien insinöörien koulutukseen.

Helmikuun puolessa välissä pidimme COSS ry:n kanssa yhteisverkkopalaverin ja opinnäytetyön tutkimusongelmat alkoivat hahmottua, jonka jälkeen kirjoitettiin koulun hyväksymä opinnäytesopimus.

Opinnäytteen työstö alkoi jo helmikuussa ja maaliskuun puolessa välissä produktiivinen osuus eli projektiohjeistus oli valmiina. Se perustuu pitkälti niihin johtopäätöksiin ja lähteisiin, joita käytän tässä tutkimuksellisessa osiossa.

Aloitin kirjallisuuskatsauksen lähteiden etsinnän jo helmikuussa. Kouvolan kampuskirjastosta löytyi hyvin projektityöstä kertovaa kirjallisuutta, hauilla Kaakkurista, Theseuksesta ja Google Scholarista löytyi tarvittava määrä artikkeleita ja e-kirjallisuutta. Neljästä aihealueesta löytyi jokaisesta tyydyttävä määrä lähdemateriaalia ja ilmiötä tutkittaessa lähteiden laatu ei ole tiukasti ja tarkasti rajattu. Menetelmänä kuvaileva kirjallisuuskatsaus on hivenen työläs, sillä se perustuu pitkälti toisten tekstien lukemiseen, asioiden poimimiseen ja viitekehykseen sijoittamiseen. Määrällinen tutkimus olisi tässä mielessä helpompi ja tulokset helpommin perusteltavissa.

9.2 Eettisyys ja luotettavuus

Kirjallisuuskatsauksen lähteiden laadussa ja määrässä on lopulta kysymys siitä, antavatko ne tarpeeksi laajan kokonaiskuvan tutkittavasta aihealueesta. Kuvailevassa kirjallisuuskatsauksessa on myös riskinä se, että katsauksen lähteiksi valikoituu tutkijan toimesta lähinnä vain sellaista informaatiota, joka tukee omia ennako-oletuksia (Vilkkä 2023, luku 1.2.1: Narratiivinen kirjallisuuskatsaus). Toisaalta deduktiivinen analyysitapa sallii tämän.

Tämä kirjallisuuskatsaus noudattaa eettistä lähestymistapaa, sillä se noudattelee kuvailevan kirjallisuuskatsauksen mallia. Lähteet ovat jäljennettävissä relevantteja ja osittain myös tieteellisesti vertaisarvioituja. Hain lähdemateriaalista vastauksia tutkimuskysymyksiini ja pyrin saamaan aiheesta tarpeeksi laajan kokonaiskuvan tekemilleni johtopäätöksille ja perusteluille.

9.3 Merkitys ja jatkotutkimusaiheet

Avoimesta lähdekoodista on tehty jonkin verran opinnäytetöitä, mutta ne sivuavat aihetta enemmänkin tarkastelemalla sopisiko avoimen lähdekoodin ohjelmisto johonkin tiettyyn opinnäytetyössä tarkasteltuun projektiin. Tämä opinnäytetyö lähestyy aihetta eri näkökulmasta.

Opinnäytetyötä ja etenkin sen tutkimuksellista osiota tehdessä tein huomion, että projektin varsinaisten toteuttamishaasteiden sijaan avoimen lähdekoodin ratkaisujen ongelma oli, että ne eivät tahdo läpäistä toteutustapana projektien niin sanottua ”alkuseulaa” päästäkseen suunnitteluvaiheeseen. Syitä tähän löytyi kirjallisuuskatsauksen lähdemateriaalista ja sille esitettiin opinnäytetyössä ratkaisu.

Mahdollisia jatkokehitysaiheita avoimen lähdekoodin osalta voisi olla kvantitatiivinen kyselytutkimus siitä, kuinka suuri osa ohjelmistoprojekteista tällä hetkellä käyttää tai perustuu avoimelle lähdekoodille. Opinnäytetyön tietojenkeräysvaiheessa olin yhteydessä Tilastokeskukseen ja pyysin heiltä tuoreita tilastotietoja asiaan liittyen, mutta heillä ei ollut tarjota dataa. Oma opinnäytetyö on myös puhtaasti teoriapohjainen ja siinä syntyneitä produktiivista osaa voisi

koetella myöhemmin asiakkailta kyselytutkimuksella, onko se koettu hyödylliseksi. Mikäli ohjeisto koetaan hyödylliseksi, myös sen jatkokehittäminen erilaisille media-alustoille voisi olla hyödyllistä.

LÄHTEET

- Actitime. 2022. Waterfall Model: What Is It, When and How to Use It? WWW-dokumentti. Saatavissa: <https://www.actitime.com/project-management/what-is-waterfall-model> [Viitattu: 15.5.2023].
- Alhonen, M. & Iloranta, R. 2021. Palvelumuotoilun menetelmiä ja työkaluja arkeen. Uudenmaan liitto: Helsinki.
- Anttila, P. 2014. Tutkimisen taito ja tiedon hankinta. WWW-dokumentti. Saatavissa: <https://metodix.fi/2014/05/17/anttila-pirkko-tutkimisen-taito-ja-tiedon-hankinta/> [Viitattu: 12.4.2023].
- Alshamrani, A. & Bahattab, A. 2015. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues* 1, 106–110.
- Britannica. 2005. WWW-dokumentti. Saatavissa: <https://www.britannica.com/topic/open-source> [Viitattu 21.3.2023].
- van Casteren, W. 2017. The Waterfall Model and the Agile Methodologies: A comparison by project characteristics. PDF-dokumentti. Saatavissa: <http://dx.doi.org/10.13140/RG.2.2.10021.50403> [Viitattu 21.8.2023].
- COSS. 2023. WWW-dokumentti. Saatavissa: <https://coss.fi/> [Viitattu: 1.4.2023].
- Currie, W. & Galliers, B. 1999. Rethinking Management Information Systems: An Interdisciplinary Perspective. Oxford University Press: Oxford.
- Euroopan parlamentin ja neuvoston direktiivi avoimesta datasta ja julkisen sektorin hallussa olevien tietojen uudelleenkäytöstä. (EU) 2019/1024.
- Euroopan parlamentti. 2020. Open source software strategy 2020–2023. PDF-dokumentti. Saatavissa: https://commission.europa.eu/system/files/2023-02/en_ec_open_source_strategy_2020-2023.pdf [Viitattu: 31.5.2023].
- Finni, J. 2022. Projektin hallintamenetelmien muutostarpeet ja mahdolliset vaihtoehdot. Kaakkois-Suomen Ammattikorkeakoulu. Tekniikan koulutusohjelma. Ylempi amk -opinnäyte. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi:amk-2022060816650> [Viitattu: 18.4.2023].
- Fintraffic. 2022. Liikenteen dataekosysteemi. WWW-dokumentti. Saatavissa: <https://www.fintraffic.fi/fi/liikenteenekosysteemi> [Viitattu: 8.3.2023].
- Forbes Advisor. 2022. Agile Vs. Waterfall: Which Project Management Methodology Is Best For You? WWW-dokumentti. Saatavissa: <https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/> [Viitattu: 22.8.2023].
- Grönroos, C. 2009. Palvelujen johtaminen ja markkinointi. 6. painos. Alma Talent: Helsinki. E-kirja. Saatavissa: <https://kaakkuri.finna.fi/Record/kaakuri.226255?sid=3083132637> [Viitattu 8.3.2023].

Laki julkisista hankinnoista ja käyttöoikeussopimuksista 29.12.1397/2016.

Halonen, K. 2021. Vaikuttava projekti. Metropolia ammattikorkeakoulu. WWW-dokumentti. Saatavissa: <https://vaikuttavaprojekti.metropolia.fi/> [Viitattu: 11.4.2023].

Hietikko, E. 2021. Tuotekehitystoiminta. BoD - Books on Demand GmbH. 4. painos. E-kirja. Saatavissa: <https://www.elibrary.com/xamk/9789528078968> [Viitattu: 11.4.2023].

Huosianmaa, T. 2020. Projektinhallinnan menetelmät ohjelmistokehityksessä. Tampereen yliopisto. Tuotantotalouden kandidaatin tutkinto-ohjelma. Kandidaatintutkielma. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi:tuni-202012068540> [Viitattu: 3.4.2023].

Hytti, S. Ruusunen, S. 2016. Ekosysteemit yritysmaailmassa. Lappeenrannan teknillinen yliopisto. Tuotantotalouden kandidaatin tutkinto-ohjelma. Kandidaatintutkielma. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:NBN:fi-fe2016052612799> [Viitattu: 12.4.2023].

JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta. 2009. JHS 169 Avoimen lähdekoodin ohjelmien käyttö julkisessa hallinnossa. Word-dokumentti. Saatavissa: <https://www.suomidigi.fi/sites/default/files/2020-06/JHS169.doc> [Viitattu: 8.3.2023].

Jyväskylän ammattikorkeakoulu. 2022. Kirjallisuuskatsaukset. WWW-dokumentti. Saatavissa: <https://oppimateriaalit.jamk.fi/yamk-kasikirja/kirjallisuuskatsaukset/> [Viitattu: 11.4.2023].

Karhapää, A., Nurmi, N. & Satama, M. 2022. Palvelumuotoilulla parempaa datatukea. *Signum* 2, 32–34.

Avoim lähdekoodi on lyömässä läpi uuden lain ansiosta. 2010. *Kauppalehti* 19.7.2010.

Kim, D. 1999. Introduction to Systems Thinking. *Innovations in management series* 16, 1–20.

Koponen, J. 2020. Avoimen lähdekoodin hyödyt ohjelmistoyrityksissä. Jyväskylän yliopisto. Tietojärjestelmätiede. Kandidaatin tutkielma. PDF-dokumentti. Saatavissa: <http://urn.fi/URN:NBN:fi:ju-202006023635> [Viitattu: 8.3.2023].

Koskinen, J. 2018. How to Build Competencies for a Data-Driven Business: Keys for Success and Seeds for Failure. *Technology Innovation Management Review* 8, 1–12.

Kuntaliitto. 2021. Avoimen lähdekoodin hankintaopas kunnille. WWW-dokumentti. Saatavissa: <https://www.kuntaliitto.fi/julkaisut/2022/2163-avoimen-lahdekoodin-hankintaopas-kunnille> [Viitattu 8.3.2023].

Löow, M. 2002. Onnistunut projekti – Projektijohtamisen ja suunnittelun käsikirja. Tietosanoma: Helsinki.

Madey, G., Freeh, V. & Tynan, R. 2002. The open source software development phenomenon: An analysis based on social network theory. *AMCIS 2002 Proceedings*, 1806–1807.

Metsämuuronen, J. 2000. Laadullisen tutkimuksen perusteet. Methelp International: Helsinki.

Moore, J. 2006. Business ecosystems and the view from the firm. *Antitrust Bulletin* 1.

Mäntyneva, M. 2016. Hallittu projekti – Jäntevästä suunnittelusta menestykselliseen toteutukseen. Helsingin kauppakamari: Helsinki.

Nissilä, J. 2016. Promoting Scalability and Sustainability of ICT4D Projects Using Open Source Software. Turun yliopisto. Turun kauppakorkeakoulu. Väitöskirja. Turun yliopiston julkaisuja – Annales E 8 Nissilä DISS. PDF-dokumentti. Saatavissa: <https://urn.fi/URN:ISBN:978-951-29-6619-6> [Viitattu: 8.3.2023].

OpenMemoryLab. 2023. WWW-dokumentti. Saatavissa: <https://www.xamk.fi/tutkimus-ja-kehitys/openmemorylab/> [Viitattu 10.5.2023].

OSI. 2023. WWW-dokumentti. Saatavissa: <https://opensource.org/licenses/> [Viitattu 2.4.2023].

Parantainen, J. 2007. Tuotteistaminen – Rakenna palvelusta tuote 10 päivässä. Talentum: Helsinki.

Parker, G. & Alstynne, M. 2005. Two-Sided Network Effects: A Theory of Information Product Design. *Management Science* 51, 1496–1497.

Ruuska, K. 2007. Pidä projekti hallinnassa – Suunnittelu, menetelmät, vuorovaikutus. 6. uudistettu painos. Talentum: Helsinki.

Salminen, A. 2011. Mikä kirjallisuuskatsaus? Johdatus kirjallisuuskatsauksen tyypeihin ja hallintotieteellisiin sovelluksiin. Vaasan yliopiston julkaisu. PDF-dokumentti. Saatavissa: https://www.uwasa.fi/materiaali/pdf/isbn_978-952-476-349-3.pdf [Viitattu: 12.4.2023].

Tampereen yliopiston kirjasto. 2023. Tutkimusaineistojen hallinta: Avoin lähdekoodi ja ohjelmat. WWW-dokumentti. Saatavissa: <https://libguides.tuni.fi/tutkimusaineistojen-hallinta/lahdekoodi> [Viitattu: 21.3.2023].

Tuomi, J. & Sarajärvi, A. 2009. Laadullinen tutkimus ja sisältöanalyysi. 6. uudistettu laitos. Helsinki: Tammi.

Tuulaniemi, J. 2011. Palvelumuotoilu. Talentum Media Oy. E-kirja. Saatavissa: <https://www.ellibslibrary.com/xamk/9789521416880> [Viitattu: 11.4.2023].

Valtiovarainministeriö. 2009. Suositus avoimen lähdekoodin ohjelmien käytöstä julkisessa hallinnossa. Tiedote. Saatavissa: <https://vm.fi/-/suositus-avoimen-lahdekoodin-ohjelmien-kaytosta-julkisessa-hallinnossa> [Viitattu: 31.5.2023].

Vijayarathy, L. & Butler, C. 2016. Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? *IEEE Software* 5, 86–94.

Vilka, H. 2023. Kirjallisuuskatsaus metodina, opinnäytetyön osana ja tekstilajina. Art House. Helsinki. E-kirja. Saatavissa: <https://www.elibrary.com/xamk/9789518849448> [Viitattu: 14.4.2023].