

TAMPEREEN YLIOPISTO

”Tää on hyvä tapa saada ihmiset kutsumaan sua velhoksi”

Koodaustaitoisten toimittajien kokemukset ohjelmoinnista arjen toimitustyössä

Informaatioteknologian ja viestinnän tiedekunta

Pro gradu -tutkielma

Teemu Kakko

Huhtikuu 2019

Tampereen yliopisto

Informaatioteknologian ja viestinnän tiedekunta

Teemu Kakko: ”Tää on hyvä tapa saada ihmiset kutsumaan sua velhoksi” – Koodaustaitoisten toimittajien kokemukset ohjelmoinnista arjen toimitustyössä

Pro gradu -tutkielma, 63 sivua ja 1 liitesivu

Tutkimus tarkastelee koodaustaitoisten toimittajien kokemuksia ohjelmoinnista journalismissa. Sitä varten haastattelin seitsemää koodaustaitoista toimittajaa, jotka kertoivat omista taustoistaan, taidoistaan ja työssä kohtaamistaan haasteista. Tarkoitukseni oli selvittää, mitä koodaustaitoisten toimittajien työ on, miten koodaustaitoisten toimittajat kuvailevat ohjelmoinnin merkitystä arjen toimitustyössä ja miten ohjelmoinnin yleistyminen muuttaa journalismia. Tutkimuksen tavoitteena on antaa ääni koodaustaitoisille toimittajille. Heidän ammattiryhmäänsä ei ole vielä juuri tutkittu Suomessa, vaikka heidän tehtävänsä on uudistaa koko verkkojournalismia. Tutkimuksen tulokset avaavat koodaustaitoisten toimittajien ajatusmaailmaa muille toimittajille.

Nykyiset koodaustaitoiset toimittajat ovat jatkoa 1960-luvun Yhdysvalloissa syntyneelle CAR-suuntaukselle. Tutkimus asettaa koodaustaitoiset toimittajat osaksi tietokoneavusteisen journalismin historiaa. Olen verrannut haastateltavieni sanomisia aiemmassa kirjallisuudessa esille nostettuihin kokemuksiin. Aiempi kirjallisuus koostuu muun muassa ulkomaisten koodaustaitoisten toimittajien haastatteluiden perusteella tehdyistä tutkimuksista. Koodaustaitoisia toimittajia on tutkittu aiemmin muun muassa Yhdysvalloissa, Britanniassa ja Norjassa.

Keräsin aineistoni teemahaastatteluilla. Haastattelin viittä Ylen koodaavaa toimittajaa sekä kahta Helsingin Sanomien koodaustaitoista toimittajaa. Nauhoitin haastattelut ja litteroin ne analyysia helpottaakseni. Analysoin aineistoa laadullisesti luokittelemalla ja teemoittelemalla. Tulokset olen jaotellut teemoihin, jotka ovat nousseet aineistosta. Tulokset kertovat, että vaikka suurin osa koodaustaitoisista toimittajista on saanut koulutuksensa tekniseltä alalta, he kokevat silti olevansa ennen kaikkea toimittajia. Heitä ajaa halu muuttaa journalismia lukijaystävällisempään suuntaan. Journalismista tulee visuaalisempaa ja personoidumpaa. Haaste koodaustaitoisille toimittajille on etenkin se, etteivät muut toimittajat tiedä, mitä ohjelmoinnilla voidaan saavuttaa.

Asiasanat: journalismi, ohjelmointi, verkkojournalismi, datajournalismi, visuaalisuus

SISÄLLYS

1	JOHDANTO	5
1.1	Mitä on ohjelmointi?	6
1.2	Vaaliennustuksista kolmiulotteisiin karttoihin	8
1.3	Tutkimuksen tavoite	11
1.4	Tutkimuskysymykset	12
2	KOODAUSTAITOISET TOIMITTAJAT MEILLÄ JA MUUALLA	13
2.1	Koodaustaitoinen toimittaja vai datajournalisti?	13
2.2	Aiempi tutkimus	14
2.2.1	Koodaustaitoisten toimittajien taustat	16
2.2.2	Työnkuva ja taidot	18
2.2.3	Koodaustaitojen määritelmä	21
2.2.4	Haasteet	22
2.2.5	Merkitys journalismille	24
2.2.6	Journalistisen työn muutos	25
2.2.7	Journalismin muutos	27
3	AINEISTO JA MENETELMÄT	29
3.1	Aineiston kerääminen	29
3.2	Aineiston perustelut	30
3.3	Kysymysrunko	31
3.4	Teemahaastattelu	32
3.5	Laadullinen analyysi	34
4	KOODAUSTAITOISTEN TOIMITTAJIEN KOKEMUKSET	37
4.1	Ohjelmointi arjen toimitustyössä	37
4.1.1	Koodaustaitoisten toimittajien taustat	37
4.1.2	Työnkuva ja taidot	40
4.1.3	Koodaustaitojen määritelmä	43
4.1.4	Haasteet	44

4.2	Ohjelmointi osana journalismia	47
4.2.1	Merkitys journalismille	47
4.2.2	Journalistisen työn muutos	48
4.2.3	Journalismin muutos	50
5	POHDINTA	54
5.1	Mitä tuloksista voi päätellä?	54
5.3	Miten tutkimusmenetelmät sopivat aiheeseen?	56
5.4	Tulosten merkitys journalismille ja toimittajille	58
	LÄHTEET	60
	LIITTEET	64

1 JOHDANTO

Siitä on toimittajalle hyötyä. Niin minulle vastattiin, kun kerroin opettelevani ohjelmoimaan. Olin juuri aloittanut journalistiikan opinnot Tampereen yliopistossa, ja mieleni teki erottua muista. Sen sijaan, että olisin valinnut alan opiskelijoiden keskuudessa suositumpia kursseja, päätin kokeilla jotain aivan muuta. Uutta se ei ollut, koska olin opetellut Javan perusteita jo lukiossa. Ohjelmoinnin kurssien ottaminen tuki ajatusta siitä, että toimittajalla pitää olla erityisosaamista, jolla voi erottautua muista. Omien opiskeluvuosieni aikana ohjelmoinnin kursseilla kävi satunnaisia journalistiikan opiskelijoita, mutta pääosin kuuluin valinnaisine opintoineni vähemmistöön. Ohjelmoinnin kurssit loivat journalistiikan opinnoille vastapainon. Ohjelmointi perustuu logiikkaan, journalismi taas ehkä enemmän taitoon esittää tarinoita. Sama lehtijuttu voi olla ihmisten mielestä hyvä tai huono, mutta koodattu ohjelma joko toimii tai ei toimi.

Oman kokemukseni mukaan työnantajat arvostavat toimittajien ohjelmointitaitoja paljon. Mahdollinen ohjelmointiosaaminen pyydetään tuomaan ilmi jo hakuvaiheessa, ja työhaastattelussa ohjelmointitaitoja kohtaan osoitetaan syvää kiinnostusta. Ohjelmoinnista on hyötyä, koska verkon uutisoinnissa hyödynnetään yhä monipuolisempia esitystapoja. Tai niin minulle on ainakin sanottu. Todellisuudessa tilanteita, joissa jutun tekeminen on edellyttänyt ohjelmointitaitoja, on ollut hyvin vähän. Ohjelmointi-innostuksesta on puhuttu jo pitkään, ja ohjelmoinnin opetus on aloitettu jo peruskouluissakin. On mahdollista, että ohjelmointi on ollut vain yksi esimerkki tulevaisuuden epävarmuuden ilmaiseisesta. Moni kurssikavereistani ei välttämättä edes tiedä, mitä ohjelmointi on, vaan heistä saattaa olla vain nykyaikaista olla mukana teknologian kehityksessä. On myös mahdollista, etteivät toimittajat edes halua työympäristöönsä ohjelmointitaitoisia ihmisiä, koska kokevat heidät uhaksi journalismille ja työyhteisölleen. Ohjelmointitaidoista voi olla toimittajalle hyötyä, mutta epäselvää on, missä yhteydessä.

Tutkin pro gradu -tutkielmassani sitä, miten ohjelmointiin suhtaudutaan arjen toimitustyössä. Tarkastelun kohteena ovat toimittajat, joiden työnkuvaan ohjelmointi kuuluu. He osaavat kertoa omasta suhtautumisestaan ohjelmointiin, koska ohjelmoivat työkseen, mutta myös arvioida muun toimituksen suhtautumista, koska tekevät projekteja yhdessä muiden toimittajien kanssa. Ennen kaikkea minua kiinnostaa ohjelmointi taitona, josta on tulossa jatkuvasti suurempi osa journalismia. Toimittajat, joiden työnkuvaan ohjelmointi kuuluu, osaavat kertoa taidoista, joita he tarvitsevat päivittäisessä työssään. Lisäksi minua kiinnostaa, millaisia vaikutuksia ohjelmoinnin yleistymisellä

on esimerkiksi journalismin vakiintuneisiin työkäytänteisiin. Toimittajat haluavat varmasti tietää, tarvitseeko toimittajan tulevaisuudessa opetella koodaamaan.

Minulle ohjelmointitaitojen kerrottiin tekevän työllistymisestä helpompaa. Tässä vaiheessa allekirjoitan sen, että taidot avaavat journalismin parissa uusia ovia. Aloitin maaliskuussa 2019 koodaavana toimittajana Ylen Tampereella sijaitsevassa Mediadeskissä. Työ antoi mahdollisuuden tutustua tutkimaani ammattiryhmään entistä paremmin. Siitä huolimatta aineistoni koostuu pelkästään syksyllä 2018 keräämistäni haastatteluista. Tulokseni perustuvat vain syksyllä litteroimaani tutkimusaineistoon, jotta työni Mediadeskissä ei vaikuttaisi millään tavalla havaintoihini koodaustaitoisista toimittajista ja heidän työstään. Pysin siis joka tapauksessa käsittelemään kaikkien haastateltavieni mielipiteitä tasapuolisesti.

1.1 Mitä on ohjelmointi?

Jotta voimme tarkastella ohjelmointia journalismissa, on ensin syvennyttävä ohjelmoinnin määritelmään. Tässä tutkielmassa ohjelmointi tarkoittaa tietokoneelle tai vastaavalle laitteelle annettavia toimintaohjeita. Heikkisen ja Saarenpään (2016) artikkelissa *Koodi on kaikkialla – lyhyt johdatus ohjelmoinnin maailmaan* muistutetaan, että ohjelmakoodia on nykyisin kaikkialla, myös autoissa ja jääkaapeissa. Ilman ohjelmistoja kaikki laitteet olisivat vain ”hyödytön kasa elektroniikkaa”. Ohjelmoinnissa laitteelle annetaan tarkka toimintaohje, jolla se tekee halutun asian. Koska mikään laite ei suoraan ymmärrä ihmisen ohjeita, toimintaohje on kirjoitettava jollakin tietokoneen ymmärtämällä ohjelmointikielillä. Tällaisia yleisesti käytettyjä ohjelmointikieliä on kehitetty paljon vuosikymmenten varrella.

Ensimmäiset ohjelmointikieliset olivat niin sanottuja konekieliä. Niillä kirjoitetut ohjelmat olivat pelkkiä ykkösistä ja nolista koostuvia jonoja. Toimintaohjeet annettiin tietokoneelle kytkimiä kääntämällä. Symbolinen konekieli kehitettiin sen vuoksi, että konekielinen ohjelmointi oli hyvin vaikeaa ja vaati äärimmäistä tarkkuutta. Konekieliset käskyt korvattiin symboleilla, eli tekstillä. Symbolisella konekielellä tehdyt ohjelmat piti kääntää konekieliseksi koodiksi, jotta tietokone ymmärsi niitä. Heikkisen ja Saarenpään artikkelissa kerrotaan, kuinka vielä 1980-luvulle saakka ohjelmakoodia saatettiin syöttää tietokoneelle reikäkorteilla. Myös symbolinen konekieli koettiin liian vaivalloiseksi, joten sen vuoksi kehitettiin korkean tason ohjelmointikieliä. Ne toimivat rajapintana ihmisten ja tietokoneiden välillä. Korkean tason ohjelmointikieliset eivät ole aiempien

kielien tapaan konekohtaisia, vaan mahdollistavat myös saman ohjelman käytön eri konekieliä käyttävissä tietokoneissa.

Ensimmäinen korkean tason ohjelmointikieli oli vuonna 1957 julkaistu FORTRAN (engl. FORMula TRANslator). Sen tehtävänä oli kääntää matemaattisia lausekkeita konekielille. Sitä käytetään edelleen raskasta tieteellistä laskentaa vaativissa tehtävissä. Kun tietojenkäsittely alkoi siirtyä liike-elämään, kehitettiin suurten tietomäärien käsittelyyn tarkoitettu COBOL (engl. COmmon Business Oriented Language) vuonna 1959. COBOLin käskyt laadittiin niin, että englantia taitava oppisi ne helposti. 1960-luvulta eteenpäin suosittuja ohjelmointikieliä olivat ALGOL ja sen pohjalta kehitetyt Pascal ja C-kielen eri versiot. Opetuskäytössä suosittu kieli on ollut BASIC (engl. Beginner's All-purpose Symbolic Instruction Code). C-kielen pohjalta on syntynyt monia uudempia ohjelmointikieliä. Sellainen on esimerkiksi Java, joka luotiin alkujaan verkkosovelluksia varten. 1990-luvulla alkunsa sai Python, joka sopii myös verkkosovelluksiin, mutta jota on mahdollista käyttää myös vaativassa ja tieteellisessä laskennassa. PHP ja JavaScript kehitettiin helpottamaan dynaamisen sisällön lisäämistä verkkosivuille. Näistä ensimmäinen on kuitenkin laajentunut jo kieleksi, jota käytetään monissa suurissakin palveluissa. (Heikkinen ja Saarenpää 2016)

Lähes kaikissa ohjelmointikielissä on rakenteellisia samankaltaisuuksia. Ohjelmakoodi koostuu erilaisista ehtolausekkeista, vertailuista ja toistolausekkeista. Esimerkiksi alla olevassa Java-kielen pätkässä on sovellettu if-lausetta. Koodi palauttaa tiedon siitä, onko luku 3 pariton vai parillinen.

```
int num = 3; // Asetetaan muuttujan num arvoksi 3
if (num % 2 == 0) // Jos muuttujan num jakojäännös on 0
    System.out.println(num + " on parillinen"); // tulosta "3 on parillinen"
else // Muussa tapauksessa
    System.out.println(num + " on pariton"); // tulosta "3 on pariton"
```

Käytän tässä tutkielmassa rinnakkain termejä ohjelmointi ja koodaaminen. Myös alalla työskentelevät ihmiset käyttävät näitä termejä sekaisin. Todellisuudessa sanoilla on pieniä merkityseroja, sillä ohjelmointi on koodaamista laajempi käsite. Koodaaminen on ohjelmoinnin vaihe, jossa tuotetaan varsinainen ohjelmakoodi (EDU.fi 2016). Siinä ihmisen käskyt käännetään koneen ymmärtämäksi kieleksi. Tässä tutkielmassa ohjelmointi on käytännössä koodaamista. Sen

vuoksi katson voivani käyttää molempia termejä tarkoittaessani samaa asiaa. Opetushallituksen EDU.fi-sivuston mukaan koodaamista käytetään puhekielessä ohjelmoinnin synonyyminä. Samalla periaatteella kutsun ohjelmoivaa ihmistä ohjelmoijaksi ja koodaavaa ihmistä koodariksi.

1.2 Vaaliennustuksista kolmiulotteisiin karttoihin

Journalismissa on käytetty tietokoneita jo useiden vuosikymmenten ajan. Melisma Coxin (2000, 6) mukaan tietokonetta käytettiin ensimmäisen kerran journalistisen jutun tekemiseen vuonna 1952, jolloin yhdysvaltalainen joukkoviestijä CBS ennusti Remington Rand UNIVAC -tietokoneella maan presidentinvaalien tuloksen. Tulos osui oikeaan, sillä republikaanien Dwight D. Eisenhower voitti demokraattien Adlai Stevensonin. Reavy (1996 Coxin 2000, 6 mukaan) kertoo tietokoneiden yleistyneen yhdysvaltalaisissa toimituksissa kolmessa vaiheessa. Ensimmäisessä vaiheessa vuoden 1952 presidentinvaalien jälkeen tietokoneita käytettiin vain kirjanpidossa. Toisessa vaiheessa 1960-luvun puolivälissä tietokoneita käytettiin jo toimituksissa, mutta ainoastaan toimistotarvikkeiden hallintaan. Viimeistään 1970-luvulla tietokoneista tuli pysyvä osa yhdysvaltalaisista journalismia.

Cox (2000, 7) nostaa esille Philip Meyerin, joka analysoi Detroitin mielenosoituksia koskevan mielipidekyselyn tuloksia IBM 360 -suurtietokoneella vuonna 1967. Analyysi paljasti, että mielenosoitukseen osallistui yhtä todennäköisesti korkeakoulutettuja kuin lukion keskeyttäneitä. Meyer mielletään tietokoneavusteisen journalismin pioneeriksi. Tietokoneavusteisesta journalismista käytetään lyhennettä CAR, joka tulee englannin kielen sanoista *computer-assisted reporting*. Suuntaus syntyi Yhdysvalloissa 1960-luvulla ja kasvatti suosiotaan 1980- ja 1990-luvuilla, kun tietokoneavusteiset jutut voittivat useita Pulitzer-palkintoja (Lewis & Usher 2014, 2). CAR-journalismissa tietokoneita käytetään uutisten etsinnän apuna (Garrison 1998 Toikkasen 2014, 7 mukaan). Siinä voidaan käyttää apuna esimerkiksi tietokantoja. Suuntauksen edustajat uskoivat, että tilastot voisivat paljastaa asioita, joita kansalaiset ja toimittajat eivät pelkillä omilla havainnoillaan voisi löytää (Parasie & Dagiral 2012, 863).

Meyerin tarkoituksena oli soveltaa yhteiskuntatieteellistä tutkimusta journalismiin. Hän halusi, että yhteiskuntatieteellisen tutkimuksen menetelmät olisivat osa journalistisia käytänteitä (Cox 2000, 7). Detroitin mielenosoitusten jälkeen Meyer oli ainoa, joka käytti säännöllisesti tietokonetta työssään toimittajana (Reavy 1996 Coxin 2000, 7 mukaan). Ensimmäisen kerran vuonna 1973 julkaistussa kirjassa *Precision Journalism: A Reporter's Introduction to Social Science Methods* Meyer kertoo,

että 1970-luvulla journalismista tuli aiempaa tieteellisempää. Meyerin mielestä toimittajat olisivat harvemmin väärässä, jos soveltaisivat työssään useammin tieteellisiä menetelmiä (Meyer 1973, 3). Hänen mielestään toimittajan pitää osata käyttää tietokantoja ja analysoida käsittelemäänsä dataa (Meyer 1991, 1). *Precision Journalism* on katsottu olevan koko CAR-perinteen peruskivi. Toikkasen (2014, 16) mukaan tietokoneavusteisen journalismin työkalut ovat pysyneet pitkälti samoina teoksen ensimmäisestä julkaisusta aina nykypäivään asti. Toikkanen siteeraa McGregoria (2013), joka toteaa, että uutisia etsitään datasta edelleen taulukkolaskennan, tietokantaohjelmien ja verkon tarjoamien keinojen avulla.

1980-luvulla tietokoneavusteisen journalismin kynnystä madalsi mikrotietokoneiden yleistymisen (Cox 2000, 10). Samaan aikaan toimitukset alkoivat käyttää entistä enemmän tietokantoja päivittäisessä työssään. 1980-luvun puolivälissä tietokoneita käytettiin jo monin tavoin apuna tutkivassa journalismissa. 1990-luvulle tultaessa tietokoneiden avulla tehtyjen juttujen määrä nousi niin suureksi, ettei niitä pysty enää luettelemaan. Tietokoneavusteista journalismia alkoivat määrittää uudet työkalut, kuten taulukkolaskentaohjelmat, tietokantaohjelmat ja sähköiset lähteet (Houston 1996 Coxin 2000, 11 mukaan). Sähköisillä lähteillä Houston tarkoittaa muun muassa sähköpostia ja keskusteluryhmiä. Parasien ja Dagiralin (2012, 856) mukaan suosiota vauhditti etenkin internetin leviäminen. Lisäksi tietokantojen käyttämisestä tuli helpompaa, kun koneluettavaa dataa alkoi tulla enemmän saataville. 2000-luvulla datajournalististen juttujen määrä on vain jatkanut kasvuaan.

Nykyisten ohjelmointitaitoisten toimittajien voidaan katsoa olevan näiden CAR-toimittajien perillisiä. Royal (2010) oli ensimmäinen, joka erotti nykyisten koodaustaitoisten toimittajien ammattiryhmän aiemmasta CAR-suuntauksesta. 2000-luvun puolivälissä yhdysvaltalaisiin toimituksiin alkoi saapua uusia koodaustaitoisia tulokkaita. Useimmiten heihin on viitattu ulkomaisessa kirjallisuudessa nimellä *programmer journalist*, vaikkei vakiintunutta nimeämiskäytäntöä ole (Pilhofer 2010). Nämä uudet koodaustaitoiset toimittajat työskentelevät CAR-toimittajien tapaan osana toimitusta, mutta suhtautuvat tietoon, tilastoihin ja lukijoiden tarpeisiin edeltäjiään erilaisella tavalla. Parasic ja Dagiral (2012, 862–866) haastattelivat Chicago Tribunessa työskenteleviä koodaustaitoisia toimittajia ja huomasivat kolme tapaa, joilla uudet koodaustaitoiset toimittajat erottuvat CAR-toimittajista:

1. *Uutiset ovat rakenteista tietoa*

Uutiset itsessään pitäisi nähdä tietokoneen avulla käsiteltävänä datana, ei pelkästään tarinana, joka on piilossa datan sisällä.

2. Yleisön pitää voida tarkastaa ja käyttää dataa

Journalismin pitää olla niin avointa, että lukijoilla on mahdollisuus tarkastaa jutuissa käytettyä dataa. Lisäksi heidän pitää pystyä yhdistelemään dataa ja käyttämään sitä uusiin tarkoituksiin.

3. On vähennettävä riippuvuutta hallinnon agendoista

Journalismin puolueettomuutta parantaa koodaustaitoisten toimittajien pyrkimys suunnitella uutissovelluksensa mahdollisimman pitkälle itse. Mitä vähemmän he käyttävät muiden valmiita materiaaleja, sitä vähemmän he kokevat olevansa ulkopuolisten agendojen vaikutuksen alaisina.

Lisää uusien koodaustaitoisten toimittajien vaikutuksesta journalismiin kerron kappaleessa 2.2.7.

Vuoden 2005 jälkeen yhdysvaltalaismedia on palkannut useita koodaustaitoisia toimittajia luomaan sovelluksia, joilla on journalistinen tarkoitus. Kenties tunnetuimmat uuden suuntauksen edustajat ovat New York Timesin Aron Pilhofer ja Washington Postin Adrian Holovaty (Parasie & Dagiral 2012, 860). Esille on nostettava myös Chicago Tribunen koodaustaitoisen tiimin perustanut Brian Boyer (em. 861). Heidän näppäimistöillään on syntynyt suuri osa ohjelmoinnin keinoin toteutetun journalismin klassikoista, jotka toimivat innoituksena koodaustaitoisille toimittajille vielä nykyäänkin. Tunnetuin niistä on Holovaty'n vuonna 2005 rakentama ChicagoCrimes-sivusto, joka yhdisti Chicagon poliisilaitoksen verkkosivut ja Google Mapsin. Käyttäjät pystyi selaamaan Chicagossa tehtyjä rikoksia alueittain (em. 863). Suuren vaikutuksen on tehnyt myös New York Timesin vuonna 2012 julkaisema *Snow Fall*. Se on interaktiivinen feature-juttu, joka kertoo Yhdysvaltojen Kaskadeilla vuonna 2012 tapahtuneesta lumivyörystä. Juttu voitti Pulitzer-palkinnon, koska se edusti aivan uudenlaista tapaa kertoa tarinoita (Usher 2016, 1). Lumivyörystä kerrottiin innovatiivisesti videoilla ja äänellä. Täysin uutta oli Kaskadivuoristosta tehty kolmiulotteinen kartta, jolla lukija pystyi itse seikkailemaan. Suosio oli valtava, sillä juttu keräsi yli 3,5 miljoonaa sivulatausta.

Vuonna 1996 Yle julkaisi maailman ensimmäisen vaalikoneen (Varho 2016). Sen jälkeen suomalainen journalismi hurautti vaalidataa käyttäviin sovelluksiin (Toikkanen 2014, 17). Helsingin Sanomat julkaisi erilaisia testejä ja Flash-animaatioita jo 2000-luvun alussa (em. 18). 2010-luvun puolella Helsingin Sanomissa alettiin tehdä uutissovelluksia ja erilaisia dataan pohjautuvia juttuja, ja kesäkuussa 2012 lehti perusti nykyisen datadeskinsä. Vuonna 2010 Yle avasi beta.yle.fi-sivuston, jolla se esitteli uusia tapoja rakentaa verkkokerrontaa. Säännöllisempi datajournalismi sai alkunsa

Svenska Yleltä vuonna 2012, jolloin sen uutisissa alettiin painottaa datajournalismia. Aloite nykymuotoisen Plus-deskin perustamiseen tehtiin marraskuussa 2012, ja keväällä 2013 deski perustettiin. Ylen aluetoimituksia palveleva Mediadeski perustettiin vuoden 2017 loppupuolella, ja se aloitti täyden toimintansa vuoden 2018 alussa (Nordell 2017). Tiimeissä työskentelee ihmisiä, jotka tuottavat datavetoista ja visuaalista journalismia. Tällä hetkellä ainoastaan Yle ja Helsingin Sanomat tekevät sitä säännöllisesti. Muu media tekee samanlaista työtä silloin tällöin satunnaisten ohjelmointitaitoisten toimittajien voimin. Muista maista ainakin Britanniassa (Hannaford 2015) ja Norjassa (Karlsen & Stavelin 2014) toimituksiin on palkattu koodaustaitoisia toimittajia.

1.3 Tutkimuksen tavoite

Tutkimuksen tavoitteena on saada tietoa siitä, keitä koodaustaitoiset toimittajat ovat ja mikä heidän merkityksensä on journalismille ja työyhteisöille. On tärkeää päästää alan ammattilaiset ääneen. Koodaustaitoiset toimittajat tekevät joka päivä töitä sekä ohjelmoinnin että journalismin parissa, joten heillä on pienestä ammattiryhmästään paljon sanottavaa. Työnantajat ovat jo mielestäni saaneet ilmaista tarpeensa, koska he ovat kertoneet niistä työpaikkailmoituksissa. Esimiehet ovat asettaneet ne vaatimukset, joiden perusteella toimittajia palkataan töihin. Tutkimukseni tehtävä on kertoa kaunistelematon versio koodaustaitoisten toimittajien arjesta. Koodaustaitoiset toimittajat osaavat lisäksi arvioida, miten muut toimittajat ovat suhtautuneet ohjelmointiin toimitustyössä. Tutkimus tuottaa uutta tietoa uutistuotannon uudistumisesta. Journalismi muuttuu ja niin muuttuu myös toimittajien tapa tuottaa journalismia.

Tutkimus antaa arvokasta tietoa toimittajille ja heidän työnantajilleen. Toimittajat saavat tietoa koodaustaitoisten toimittajien ammattiryhmästä ja sen tavasta toimia journalismin kentällä. He pystyvät arvioimaan, miten koodaustaitoisten toimittajien yleistyminen vaikuttaa heidän työhönsä ja tarvitseeko heidän kenties opetella tulevaisuudessa ohjelmoimaan. Työnantajat puolestaan saavat tietoa uusista ominaisuuksista, joita toimittajat tarvitsevat pärjätäkseen journalismin muuttuvalla kentällä. Työnantajat pystyvät arvioimaan, olisiko ohjelmointitaitoisten toimittajien lisääminen omassa toimituksessa kannattavaa. He saavat tietää, miten ohjelmoinnin avulla voidaan parantaa journalistista sisältöä. Tutkimuksesta on hyötyä koodaustaitoisille toimittajille, koska sen kautta he saavat tietää, millaisia taustoja ja taitoja heidän kollegoillaan on. Koodaustaitoisten toimittajien joukko on pieni, ja yhteistyötä eri mediatalojen välillä tehdään yllättävän vähän, joten heistä on varmasti kiinnostavaa tietää, mitä muut samanlaista työtä tekevät osaavat. Niiden tietojen pohjalta

he pystyvät kehittämään omaa osaamistaan. Tutkimuksen tulokset kiinnostavat varmasti myös niitä päättäjiä, jotka valinnoillaan ohjaavat media-alaa. Tietoja voisi käyttää esimerkiksi toimittajien työnkuvan valvomiseen. Tuloksilla voidaan ottaa kantaa toimittajien työn laatuun ja määrään ja sitä kautta jopa toimittajien työhyvinvointiin.

1.4 Tutkimuskysymykset

Tutkimus vastaa kolmeen pääkysymykseen. Ne pureutuvat ohjelmoinnin merkitykseen yleisellä tasolla, liittävät ohjelmoinnin osaksi arjen toimitustyötä ja kartoittavat koodaavien toimittajien tarvitsemia taitoja. Kysymykset ovat järjestyksessä yksityiskohtaisemmasta yleisempään.

1. Mitä koodaavien toimittajien työ on?
2. Miten koodaustaitoiset toimittajat kuvailevat ohjelmoinnin merkitystä arjen toimitustyössä?
3. Miten ohjelmoinnin yleistyminen toimitustyössä muuttaa journalismia?

Nämä kysymykset kysyn keräämältäni haastatteluaineistolta. Analysoin koodaustaitoisten toimittajien vastauksia ja pohdin heidän työnsä luonnetta ja journalismin muutosta. Ensimmäisen kysymyksen kohdalla käsitelen lisäksi koodaustaitoisten toimittajien tarvitsemia taitoja. Toinen kysymys pitää sisällään koodaustaitoisten toimittajien näkemykset omasta ja journalismin tulevaisuudesta.

2 KOODAUSTAITOISET TOIMITTAJAT MEILLÄ JA MUUALLA

2.1 Koodaustaitoinen toimittaja vai datajournalisti?

Kutsun tutkimiani toimittajia joko ohjelmointitaitoisiksi toimittajiksi tai koodaustaitoisiksi toimittajiksi. Koodaustaitoinen toimittaja on toimittaja, jolla on koodaamiseksi määriteltyjä erityistaitoja. Näitä taitoja hän voi käyttää itsenäisesti omissa jutuissaan tai tarjota muiden toimituksen jäsenten avuksi, jos hän työskentelee erityisessä visuaalista, interaktiivista tai datavetoista sisältöä tuottavassa tiimissä. Termin koodaustaitoinen toimittaja olen keksinyt itse, jotta voin käsitellä näitä erityistaitoja hyödyntävää ryhmää kokonaisuutena. Koodaustaitoisten toimittajien ammattinimikkeet ja työnkuvat vaihtelevat työpaikasta ja kokemuksesta riippuen. Esimerkiksi Ylellä koodaustaitoisia toimittajia kutsutaan koodaaviksi toimittajaksi ja Helsingin Sanomissa datajournalisteiksi. Erilaiset nimikkeet johtuvat siitä, ettei vakiintunutta nimeämiskäytäntöä ole. Päätös siitä, miksi heitä kutsutaan, on tehty jokaisella työpaikalla erikseen.

Sen lisäksi, että koodaustaitoisia toimittajia on kutsuttu ulkomailla nimellä *programmer journalist* (Pilhofer 2010), heistä on käytetty myös nimityksiä *hacker journalist* (Usher 2016, 73), *interactive journalist* (em. 3) ja *data journalist* (em. 90). Usherin mukaan nimike riippuu siitä, millainen toimittajan tausta on. *Programmer journalist* on koulutukseltaan toimittaja, joka on aloittanut myöhemmin ohjelmoimaan, kun taas *hacker journalist* edustaa teknisempää ”hakkerikulttuuria” ja on innostunut myöhemmin journalismista. Ensimmäinen luottaa töissään kokemuksiinsa toimittajan töistä, ja jälkimmäinen tuo toimitukseen enemmän teknistä osaamista. *Interactive journalist* puolestaan tuottaa interaktiivista sisältöä. *Programmer journalist* ei voi olla samaan aikaan *hacker journalist*, mutta molempia voi kutsua nimellä *interactive journalist*, jos toimenkuvaan kuuluu interaktiivisen sisällön tekeminen. Omalla termilläni koodaustaitoinen toimittaja en ota kantaa siihen, millainen tausta henkilöllä on. Esittelen suomalaisten koodaustaitoisten toimittajien koulutustaustoja ja aiempia työtehtäviä myöhemmin kappaleessa 4.1.1.

Ero koodaustaitoisen toimittajan ja datajournalistin välillä on samaan aikaan selkeä ja hämärä. Yleensä datajournalisti tarkoittaa ihmistä, joka työstää avoimen lähdekoodin tietoa (Gynnild 2014, 719). Tämä tieto on verkossa vapaasti saatavilla, ja sitä voi käsitellä vapaasti avoimen lähdekoodin

työkaluilla. Gynnildin mukaan datajournalismissa toimittaja käyttää tietokantoihin tallennettua numeerista tietoa pääasiallisena uutislähteenään. Usherin (2016, 90) mukaan datajournalisti etsii, järjestee, analysoi ja julkaisee numeroihin ja tilastoihin perustuvia juttuja. Datajournalisti eroaa koodaustaitoisesta toimittajasta siten, että datajournalisti ei yleensä osaa ohjelmoida. Datajournalisti ei siis luo sovelluksia, joilla olisi interaktiivisia ominaisuuksia. Datajournalisti voi kuitenkin kokea olevansa koodaustaitoinen toimittaja, jos hän osaa koodata (em. 97). Samoin koodaustaitoinen toimittaja voi kokea olevansa datajournalisti, jos hän käyttää työssään taulukkolaskentaohjelmia ja tietokantoja. Tässä tutkimuksessa keskityn toimittajiin, jotka kokevat hyödyntävänsä työssään ohjelmointia.

2.2 Aiempi tutkimus

Tietokoneavusteisen journalismin vaikutuksia on tutkittu siitä lähtien, kun toimitukset ovat alkaneet tietokoneistua. Tällä vuosikymmenellä niitä ovat pohtineet muun muassa Sarah Cohen, James T. Hamilton ja Fred Turner artikkelissaan *Computational Journalism* (2011) sekä Terry Flew, Christina Spurgeon, Anna Daniel ja Adam Swift artikkelissaan *The Promise of Computational Journalism* (2012). Empiirisiä tutkimuksia nykyajan koodaustaitoisista toimittajista on olemassa hyvin vähän. Ensimmäinen sellainen julkaistiin vuonna 2010, kun Cindy Royal haastatteli New York Timesin interaktiivista sisältöä tuottavan osaston jäseniä ja julkaisi tutkimuksensa *The Journalist as Programmer: A Case Study of The New York Times Interactive News Technology Department*. Royal vietti New York Timesin toimituksessa viikon ja tarkkaili osaston käytäntöjä, osallistui heidän palavereihinsa ja haastatteli henkilökuntaa. Etnografiselle tutkimukselle tyypilliseen tapaan Royal seurasi sivusta, kun toimittajat tekivät töitään. Toinen merkittävä empiirinen tutkimus on Sylvain Parasien ja Eric Dagiralin *Data-driven journalism and the public good: “Computer-assisted-reporters” and “programmer-journalists” in Chicago* vuodelta 2012. Parasien ja Dagiralin tutkimus koostuu laadullisten tutkimushaastatteluiden ja määrällisen sisällönanalyysin yhdistelmästä. Tutkijat haastattelivat sanomalehti Chicago Tribunesta 15:tä ihmistä, joista kaksi työskentelivät koodaustaitoisina toimittajina uutissovelluksia tekevässä tiimissä. Tutkimuksesta käy ilmi, millaisia vaikutuksia sillä on, kun koodarit ja toimittajat työskentelevät yhdessä journalistisen sisällön tuottamiseksi.

Nikki Usher on julkaissut kokonaisen teoksen interaktiivisesta journalismista. Usherin *Interactive journalism: hackers, data, and code* ilmestyi vuonna 2016. Teoksessa käydään läpi interaktiivisen

journalismin kehittymiseen johtaneita tapahtumia ja kuvaillaan interaktiivisen journalismin ammattilaisia eri toimituksissa. Usher vieraili yhteensä seitsemässä toimituksessa vuosina 2011–2015, englanninkielisessä Al Jazeeraassa, NPR:llä, New York Timesissa, The Guardianissa, Wall Street Journalissa sekä uutistoimisto AP:n New Yorkin ja Washington D.C:n toimituksissa. Usherin haastattelemat koodaustaitoiset toimittajat kertoivat ennen kaikkea taustoistaan, taidoistaan, arvoistaan ja kohtaamistaan haasteista. Sen lisäksi Usherin tutkimuksesta käy ilmi, miten interaktiivista sisältöä tuottavat tiimit työskentelevät suhteessa toimituksen muihin jäseniin ja miten toimitukset eroavat kulttuureiltaan toisistaan.

Yhdysvaltojen ulkopuolella koodaustaitoisista toimittajista on tehty muutamia empiirisiä tutkimuksia. Yksi niistä on norjalaisten Joakim Karlsenin ja Eirik Stavelinin *Computational Journalism in Norwegian Newsrooms* vuodelta 2014. Karlsenin ja Stavelinin aineisto koostuu yhdeksän toimittajan haastattelusta. Haastatelluilla on joko ohjelmointitaitoja tai he käyttävät tietokantoja ja määrällisiä keinoja tuottaakseen tietokoneavusteista journalismia. Tutkimuksen tarkoituksena on kertoa tietokoneavusteisen journalismin käytännöistä ja vaikutuksista perinteiseen journalismiin norjalaisissa toimituksissa. Toinen eurooppalainen empiirinen tutkimus on Liz Hannafordin vuonna 2015 julkaisema *Computational Journalism in the UK newsroom: hybrids or specialists?* Hannaford tutki Britannian yleisradioyhtiön BBC:n visuaalisen journalismin yksikön ja taloussanomalehti Financial Timesin interaktiivista journalismia tuottavan tiimin käytäntöjä ja merkitystä. Kummassakin tiimissä tehdään datavetoista ja interaktiivista sisältöä. Hannafordin aineisto koostuu BBC:n yksikön ryhmäkeskustelusta sekä 10 puhelinhaastattelusta BBC:n ja Financial Timesin toimittajien, koodareiden ja tiiminvetäjien kanssa. Haastatellut ovat kertoneet taidoistaan ja ohjelmoinnin merkityksestä journalismille tulevaisuudessa. Suomessa koodaustaitoisia toimittajia on tutkinut Ninni Lehtniemi, jonka artikkeli *Nerdistan–Designia–Journaland: Best practices in multiskilled digital journalism teamwork* julkaistiin vuonna 2016. Lehtniemi tutki Helsingin Sanomien stipendiaattina sitä, miten toimittajat ja koodarit voisivat työskennellä keskenään paremmin. Hän haastatteli datajournalisteja, graafisia suunnittelijoita, koodareita ja tiiminvetäjiä BBC:ltä, The Guardianilta, Helsingin Sanomista ja Yle Kioskilta. Tutkimus kertoo, millaisia tietoja ohjelmoinnista tulevaisuuden toimittajalla kannattaa olla ja millaisia haasteita haastateltavat näkevät toimittajien ja koodareiden yhteistyössä.

Empiiristen tutkimusten lisäksi ulkomaisessa kirjallisuudessa on käsitelty toimittajien ja koodareiden yhteistyötä. Nikki Usher on tutkinut yhdessä Seth C. Lewisin kanssa avointen rajapintojen merkitystä toimittajien ja koodareiden törmäyttämiseksi artikkelissa *Open source and journalism: Toward new*

frameworks for imagining news innovation (2013) ja yhtä varta vasten toimittajien ja koodareiden törmäyttämiseksi perustettua organisaatiota artikkelissa *Code, collaboration, and the future of journalism: A case study of the hacks/hackers global network* (2014). Näiden lisäksi työpari on tutustunut yhdysvaltalaisten Knight- ja Mozilla-säätiöiden yhteiseen projektiin artikkelissa *Trading zones, boundary objects, and the pursuit of news innovation: A case study of journalists and programmers* (2016). Yhteistä Usherin ja Lewisin tutkimuksille on se, ettei niissä juuri puhuta koodaustaitoisista toimittajista, vaan toimittajista ja koodareista erillisinä ammattiryhminä. Matthew Powers on puolestaan tutkinut laajemmalla tasolla uusien teknisten työtapojen rantautumista yhdysvaltalaisiin mediataloihin artikkelissaan ”*In forms that are familiar and yet-to-be invented*”: *American journalism and the discourse of technologically specific work* vuodelta 2012. Powers määrittelee kolme tapaa, joilla mediataloissa yleensä suhtaudutaan uusiin työtapoihin.

Seuraavissa kappaleissa esittelen tutkijoiden havaintoja koodaustaitoisten toimittajien taustoista, taidoista, haasteista ja kokemuksista. Olen muodostanut teemoja, joista jokaisessa käsittelen koodaustaitoisten toimittajien työtä eri näkökulmasta. Nämä teemat ovat nousseet esille aiemmasta kirjallisuudesta ja omien haastateltavieni puheista. Teemat eivät noudata omiin haastateltaviini käyttämäni kysymysrunkoa, vaan ne ovat syntyneet valmiin aineiston ja aiemman kirjallisuuden vuoropuheluna. Kappaleessa 2.2.1 kerron havainnoista, joita tutkijat ovat tehneet koodaustaitoisten toimittajien taustoista ja ammatti-identiteetistä. Seuraavassa kappaleessa 2.2.2 tutustun tutkijoiden haastateltavien toimittajien työnkuvaan ja taitoihin. Kappaleeseen 2.2.3 olen listannut tutkijoiden ja haastateltujen mielipiteitä siitä, miten toimittajat suhtautuvat ohjelmointiin. Koodaustaitoisten toimittajien työn haasteista kerron kappaleessa 2.2.4. Kappaleeseen 2.2.5 poimin esimerkkejä siitä, miten koodaustaitoiset toimittajat ovat nähneet merkityksensä työyhteisölle. Lopuksi kerron, miten aiemmassa tutkimuksessa on arvioitu ohjelmoinnin vaikuttavan toimittajien työnkuvaan (kappaleessa 2.2.6) ja yleisesti journalismiin (2.2.7). Aiemman kirjallisuuden havaintoihin palaan kappaleessa neljä, kun peilaan niitä omiin tutkimustuloksiini.

2.2.1 Koodaustaitoisten toimittajien taustat

Aikaisemmassa tutkimuksessa ei ole juurikaan otettu kantaa siihen, millaisissa oppilaitoksissa koodaustaitoiset toimittajat ovat saaneet koulutuksensa. Sen sijaan on keskitytty heidän aiempiin työtehtäviinsä ja aiemmin hankittuihin taitoihinsa. Usher (2016, 84) toteaa useimpien koodaustaitoisten toimittajien olevan taustoiltaan toimittajia. Suurin osa on saanut

toimittajakokemusta joko journalismin oppilaitoksesta tai työskentelemällä toimittajana perinteisessä uutistoimituksessa. Suurin osa koodaustaitoisista toimittajista ei ole alun perin tekniseltä alalta, vaan he ovat opetelleet itse ohjelmoimaan. Heille on tärkeää, että he kokevat itse olevansa ennen kaikkea toimittajia. Hannaford (2015, 16) on Usherin kanssa samoilla linjoilla. Hän haastatteli koodaustaitoisia ihmisiä, jotka työskentelevät journalismin parissa. Näistä vähemmistöllä oli ylempi korkeakoulututkinto tietojenkäsittelytieteistä ja suurin osa oli opiskellut humanistisia tieteitä tai taidealaa. Ohjelmointi oli alkanut kiinnostaa heitä vasta myöhemmin. Usher (2016, 84–85) on haastatellut toimittaja Michelle Minkoffia, joka opiskeli toimittajaksi, mutta päätti alkaa opetella ohjelmointia osallistuttuaan New York Timesin koodaustaitoisen toimittajan Derek Willisin kurssille. Ennen valmistumistaan Minkoff ei tiennyt koodaamisesta mitään, mutta töissään Los Angeles Timesissa, PBS:llä ja uutistoimisto AP:lla hän on onnistunut yhdistämään ohjelmoinnin ja journalismin. Vaikka kaksi hänen viimeisintä ammattinimikettään eivät olekaan sisältäneet viitteitä journalismista, pitää hän silti itseään toimittajana, koska hän työskentelee päivittäin tuottaakseen journalistista sisältöä.

Parasie ja Dagiral (2012, 861) ovat haastatelleet Chicago Tribunen koodaustaitoisia toimittajia, jotka olivat alun perin koodareita, mutta innostuivat myöhemmin journalismista. Tällaisia ihmisiä kiinnostaa tutkijoiden mukaan usein yhteiskunnallinen vaikuttaminen. Koodarit ovat tulleet töihin toimituksiin, koska he ovat kokeneet taidoistaan olevan hyötyä demokratian kannalta. Parasien ja Dagiralin haastatteleman koodaustaitoisen toimittajan Brian Boyerin mielestä journalismi ja demokratia mahdollistavat asioiden muuttumisen paremmiksi. Demokratian tavoittelemisen tuntuu ajavan taloudellisten etujen edelle, kun koodarit päättävät kokeilla siipiään journalismin parissa. Parasien ja Dagiralin mukaan journalismi on koodareille palkitseva työympäristö, vaikka ohjelmistopuolelta saisi mahdollisesti parempaa palkkaa. Usherin (2016, 76) mukaan koodareita on lähtenyt ohjelmistoalalta sen vuoksi, ettei siellä saa käyttää luovuuttaan kuten journalismissa. Toimitustyö tarjoaa uusia haasteita joka päivä, ja koodaustaitoiset toimittajat saavat mahdollisuuden rakentaa asioita, joiden rakentaminen ei olisi ollut ohjelmistopuolella mahdollista. Usherin haastattelema koodaustaitoinen toimittaja Ryan Mark kiteyttää ohjelmistojen ja journalismin parissa työskentelevien ihmisten erot näin: Ohjelmistopuolella ihmisiä ei juurikaan kiinnosta, minkä parissa he työskentelevät. Media-alalla tilanne on täysin päinvastoin. Ihmiset välittävät työstään ja haluavat tehdä oikeita asioita.

New York Timesin interaktiivisen journalismin osaston toimittajista osa oli työskennellyt aiemmin teknisen alan yrityksissä ja osa journalismin parissa (Royal 2010, 12). Suurin osa oli työskennellyt

aiemmin media-alalla. Jotkut olivat olleet töissä Las Vegas Sunissa, Washington Postissa ja Kanadan yleisradioyhtiössä ja jotkut olivat olleet aiemmin toisessa roolissa New York Timesissa. Useat Royalin haastateltavista mainitsivat vaikuttaneensa opiskeluaikoinaan koulunsa sanomalehdessä, vaikka he eivät välttämättä opiskelleet journalismia. Osastolla työskentelee ihmisiä, jotka ovat opiskelleet aiemmin muun muassa taidealaa, antropologiaa, historiaa, englantia ja journalismia. Vain muutamalla oli koulutus tietotekniseltä alalta. Suurin osa oli alkanut opetella ohjelmoimaan hyvin nuorena tai suuntautunut ohjelmointiin myöhemmällä iällä. Itseoppineisuutta pidettiin hyvänä asiana, koska se kertoo samalla ihmisen luontaisesta uteliaisuudesta.

2.2.2 Työnkuva ja taidot

New York Timesin koodaustaitoiset toimittajat tekevät muun muassa interaktiivisia karttoja, visualisaatioita, aikajanoja ja grafiikoita (Royal 2010, 7). He työskentelevät paljon tietokantojen parissa, ja heidän käyttämänsä data voi olla peräisin käyttäjiltä tai jostakin ulkoisesta lähteestä. New York Timesin interaktiivisen osaston päätavoite on tuottaa journalismia, johon lukijat voivat osallistua. Myös Usher (2016, 122–125) tutki samaa New York Timesin osastoa. Hänen tutkimuksessaan kerrotaan, kuinka osaston toimittajat kokevat palvelevansa uutisosaston tarpeita sen sijaan, että he kokisivat tuottavansa uutissisältöä omana erillisenä yksikkönään. Kirjassaan *Making News at The New York Times* Usher (2014, 184) sanoo, että tämä yhteistyö muun toimituksen kanssa tarkoittaa kuitenkin sitä, että osaston tekemät interaktiiviset elementit häviävät netin uutisvirran uumeniin hyvin nopeasti. Interaktiivinen elementti, jonka tekemiseen meni kolme kuukautta, pysyi lehden etusivulla vain muutaman tunnin. Parasien ja Dagiralin (2012, 854) mukaan Chicago Tribunen koodaustaitoiset toimittajat puolestaan suunnittelevat niin sanottuja uutissovelluksia. Nämä sovellukset voivat olla esimerkiksi interaktiivisia karttoja ja visualisointeja. Yhteistä sovelluksille on se, että ne perustuvat erilaisiin tapoihin kerätä, käsitellä, analysoida ja esittää dataa tietokoneella.

Usher (2016, 104–138) on tutustunut erilaisiin interaktiivista sisältöä tuottaviin tiimeihin ja vertaillut muun muassa niiden kokoonpanoja keskenään (taulukko 1). Usherin vierailemissa toimituksissa tiimien työnkuvaan vaikutti esimerkiksi sen koko. Al Jazeeraan englanninkielisessä toimituksessa työskenteli kesällä 2012 vain yksi toimittaja, joka osasi tehdä interaktiivisia elementtejä (em. 102). Usherin haastattelema Mohammed el-Haddad teki muun muassa interaktiivisia karttoja ja infografiikoita yhteistyössä toimittajien ja kuvaajien kanssa. Ideoita grafiikoihin hän keräsi usein netistä. Tuotannon lisäksi hän opasti muita toimittajia tuottamaan interaktiivista sisältöä. Taustaltaan

Haddad on tietojenkäsittelytieteilijä, mutta pystyäkseen työskentelemään päivittäisen uutistuotannon parissa, on hänen ymmärrettävä jotakin myös journalismista (em. 105–106). Julkisen palvelun NPR:llä koodaustaitoiset toimittajat työskentelivät hyvin erilaisella tavalla Al Jazeeraan verrattuna (em. 116). Tuolloin NPR:n tiimi koostui koodaustaitoisista toimittajista ja datajournalisteista, ja heitä luotsasi henkilö, jolla on taustaa tekniseltä alalta. Tiimin tarkoituksena oli palvella uutistuotannon tarpeita tuottamalla toimivia sovelluksia. Tiimin jäsenet keskittyivät enemmän tuotekehitykseen kuin tarinankerrontaan. He puhuivat teknisillä termeillä, ja ryhmä työskenteli kuten ohjelmistoalalla. NPR:lle oli tyypillistä kokeilunhaluisuus ja se, että tiimi oli valmis julkaisemaan tuotoksensa usein jo ennen kuin se oli hiottu loppuun asti.

News Organization	People	Process	Integration
Al Jazeera English	Hacker journalist	Slow	Service/cooperative
National Public Radio (NPR)	Hacker journalist head, programmer journalists, data journalists	Slow	Service
The Wall Street Journal	Programmer journalists	Fast	Independent/Service
The Guardian	Data journalists	Slow	Independent/Service
The New York Times	Programmer journalist head, data journalists, hacker journalists	Slow/Fast	Service
Associated Press (New York)	Hacker journalists head, hacker journalists/ programmer journalists	Fast	Service
Associated Press (D.C.)	Programmer journalist head, programmer journalists	Slow/Fast	Service

Taulukko 1: Interaktiivista sisältöä tuottavien tiimien erot ja samankaltaisuudet (Usher 2016, 134).

Wall Street Journalissa koodaustaitoisten toimittajien luomat sovellukset saattoivat toimia tiedon välittäjinä ilman, että niiden ympärille luotiin minkäänlaista tarinaa (Usher 2016, 120). Wall Street Journalissa koodaustaitoisten toimittajien tiimi ei myöskään ollut pelkkä palveludeski, vaan toimittajat pystyivät tekemään myös omia projektejaan. The Guardianin tiimi koostui pääasiassa datajournalisteista (em. 130). Se ei siis tehnyt juurikaan interaktiivisia elementtejä, vaan keskittyi datan puhdistamiseen ja lajitteluun. The Guardianin datajournalistit käyttivät koodattuja ohjelmistoja

ilman, että osaisivat itse välttämättä koodata. Heidän työtapansa oli hyvin samankaltainen kuin Wall Street Journalissa, sillä he palvelivat muuta uutistuotantoa, mutta työstivät myös omia projektejaan. Uutistoimisto AP:lla koodaustaitoisten toimittajien tapa toimia riippui täysin toimituksesta (em. 133). AP:lla on toimitukset sekä New Yorkissa että Washington D.C:ssä. New Yorkissa tiimi koostui koodaustaitoisista toimittajista, joista osalla on tekninen koulutustausta ja osalla toimittajatausta, ja heitä luotsaavasta tiiminvetäjästä, jolla on tekninen tausta. New Yorkissa työskentelytapa oli nopea, ja siellä osallistuttiin päivittäiseen uutistuotantoon. Washington D.C:n tiimi puolestaan koostui toimittajataustaisista koodaustaitoisista toimittajista, joita johti toimittajataustainen henkilö. He keskittyivät pidempiin, tutkivan journalismin juttuihin.

Sanomalehti Las Vegas Sun esitti vuonna 2008 listan 22 teknologiasta, joiden avulla verkkosivuille saa luotua interaktiivista sisältöä (Royal 2010, 4). Listalle pääsivät muun muassa HTML, CSS ja JavaScript, jotka ovat perinteisiä front-end-kieliä. Niiden lisäksi listalla mainittiin muun muassa ohjelmistokehys Django sekä Adoben kehitysympäristö Flash, jonka omalla ohjelmointikielellä oli erityisen suosittua tehdä interaktiivisia elementtejä 2000-luvun puolivälissä (Usher 2016, 49–50). PostgreSQL mainittiin listassa esimerkkinä back-end-puolen järjestelmästä. Front-end tarkoittaa koodia, jota ajetaan verkkoselaimessa, kun taas back-end tarkoittaa koodia, jota ajetaan sivuston palvelimella. New York Timesissa, jota Royal (2010, 16–17) tutki, tyypillinen projekti edellytti aina sekä front-end-ohjelmoijaa että back-end-ohjelmoijaa. Siellä toimittajat pystyivät käyttämään Djangon sijaan myös Ruby on Railsia. Chicago Tribunessa suurella osalla koodaustaitoisista toimittajista oli jo ennestään kokemusta Pythonista, Rubysta tai PHP:stä, koska heidän juurensa ovat CAR-journalismissa (Parasie & Dagiral 2012, 861). Britanniassa Hannaford (2015, 16) pyysi haastattelemaan koodareita ja toimittajia listaamaan omia teknisiä avaintaitojaan (taulukko 2). Koodarit mainitsivat perinteisten front-end-kielten lisäksi Pythonin, Rubyn, SQL:n ja Unix-ohjelmoinnin. Sen sijaan toimittajat kieltäytyivät mainitsemasta ohjelmointitaitoja ollenkaan, mutta kertoivat osaavansa käyttää Exceliä ja Tableautta, jolla voi tehdä interaktiivisia datavisualisoiteja. Sekä koodarit että toimittajat kertoivat hallitsevansa erilaisia paikkatieto-ohjelmistoja sekä perusteet tilastolliseen laskentaan tarkoitettusta R-ohjelmointikielestä.

Lehtniemi (2016a, 18) on listannut vaatimuksia hyvälle verkkojournalismia tekeväälle tiimille. Hänen mukaansa tiimissä tulisi olla vähintään ohjelmointi-, journalismi- ja grafiikkaosaamista. Jos tiimi haluaa käsitellä suuria tietomääriä, jollakin tiimin jäsenellä pitää olla lisäksi taito analysoida dataa. Lehtniemen mukaan on harvinaista, että yhdellä ihmisillä olisi kaikki nämä taidot. Sen vuoksi tiimin vähimmäiskoko on yleensä kaksi tai kolme henkilöä. Tutkimusta varten haastatellut suomalaiset

korostivat, että tiimin jokaisella jäsenellä olisi hyvä olla ainakin kaksi taitoa. Pelkät tekniset taidot eivät kuitenkaan riitä. Halu oppia on ominaisuus, jota Lehtniemen mukaan arvostetaan enemmän kuin mitään muuta (em. 19). Jokaisen tiimin jäsenen pitäisi olla kiinnostunut tuottamaan mullistavaa journalismia ja olla halukas oppimaan uusia asioita. Jokaisen tulisi lisäksi pitää muutoksista ja olla sosiaalisesti lahjakas (em. 37).

	Developers	Journalists
Front End (User Interface)	HTML/CSS JavaScript	
Back End (support systems eg Database management)	Python PHP Ruby SQL UNIX	
Data-analysis/presentation	MPS SPSS GIS mapping software R	Excel GIS mapping software R Tableau

Taulukko 2: Hannaford (2015, 16) pyysi haastattelemiaan brittiläisiä koodareita ja toimittajia listaamaan omia teknisiä avaintaitojaan.

2.2.3 Koodaustaitojen määritelmä

Aikaisemmassa kirjallisuudessa ei oteta kantaa siihen, miten koodaustaitoiset toimittajat itse määrittelevät toimittajien koodaustaidot. Sen sijaan koodaustaitoiset toimittajat ovat kertoneet, miten he kokevat tavallisten toimittajien suhtautuvan ohjelmointiin. Usherin (2016, 33) tutkimissa toimituksissa muut pitivät koodaustaitoisia toimittajia arvokkaana työvoimana. Koodaustaitoisten toimittajien katsottiin uudistavan journalismia, ja heidän taitojensa vuoksi heitä pidettiin sankareina tai pelastajina. Usherin haastatteleman Chicago Tribunen koodaustaitoisen toimittajan Ryan Markin mielestä muut pitivät heitä melkein taikureina, koska eivät oikeastaan tieneet, mitä koodaustaitoiset toimittajat tekevät (em. 139). Mark antaa esimerkin siitä, millaisia ennakkokäsityksiä hänen tiiminsä

kohtaa: Hänen työpisteensä ohi kävelee päivittäin ihmisiä, joiden mielestä Mark vain istuu päivän tietokoneensa edessä ja pitää ruudullaan ikkunoita, joissa on värikkäitä sanoja.

Koodaustaitoisten toimittajien alaa on kutsuttu erityisalaksi ja ohjelmointia erityistaidoksi (Usher 2016, 18). Powers (2012, 30–36) on määritellyt kolme tapaa, joilla mediataloissa voidaan suhtautua uusiin teknisiin erityistaitoihin. Ensinnäkin, uutta työtapaa voidaan pitää esimerkkinä jatkuvuudesta (Exemplar of Continuity). Powers siteeraa Woltonia (1979, 37), jonka mukaan uusi teknologia nähdään vain uutena tapana saavuttaa perinteiset journalistiset tavoitteet, ei voimana, joka rikkoisi ne. Toiseksi, uusi työtapa voidaan Powersin mielestä nähdä alistettavana uhkana (Threat to Be Subordinated). Jos uuden teknologian ei katsota asettuvan linjaan aiempien ammatillisten normien kanssa, toimittajat voivat pitää sitä uhkana, joka vaarantaa ammatillisia käytäntöjä. Kolmanneksi uusi teknologia voi olla perusta journalismin uudistumiselle (Basis for Journalistic Reinvention). Uudet erityistaidot nähdään mahdollisuutena vaikuttaa olemassa oleviin käytäntöihin ja arvoihin, mutta myös mahdollisuutena luoda joitakin ennen näkemättömiä journalismin muotoja.

Hannaford (2015, 19) on kiinnittänyt huomiota toimittajien koodaustaitojen määritelmän puuttumiseen. Hänen mielestään koodariyhteisöt ja useat taitavat toimittajat ovat vaarassa syrjäytyä journalismin kentältä, jos ei määritellä tarkasti, mitä koodaamisella tarkoitetaan. Hannaford on huolissaan määritelmän puuttumisesta journalismin koulutukseen. Toimittajien koodaustaitoja on haastavaa opettaa uudelle toimittajasukupolvelle, jos ei ole olemassa ajantasaista tietoa siitä, miten toimituksessa nykyään työskennellään. Joillekin ohjelmointitaidot tarkoittavat ymmärrystä ohjelmointikielistä, kuten JavaScriptista tai Pythonista (em. 8). Toisille se saattaa tarkoittaa vähän parempia tietoteknisiä taitoja. Hannafordin mukaan olemassa on hyvin vähän tutkimuskirjallisuutta siitä, käyttävätkö toimittajat ohjelmointitaitoja työssään ollenkaan.

2.2.4 Haasteet

Haasteita saattaa aiheuttaa etenkin koodaustaitoisten ja koodaustaidottomien toimittajien yhteistyö, jos toimituksessa työskentelevillä koodaustaitoisilla ihmisillä ei ole käsitystä journalismin perinteisistä tavoista toimia. Boczkowski (2005, 135) kertoo haasteiden ilmenneen jo 1990-luvun loppupuolella, jolloin toimituksissa alettiin tuottaa entistä enemmän interaktiivista sisältöä. Sanomalehti Houston Chronicle päätti vuosituhannen vaihteessa tehdä kunnianhimoisen juttuprojektin, joka sisälsi kuvia, ääntä, videota ja päivittyvän kartan. Se toteutettiin yhteistyössä

toimittajien ja koodareiden kanssa. Kun projektin tekeminen oli saatu aloitettua, ilmeni Boczkowskin mukaan ongelmia toimittajien ja koodareiden välisessä viestinnässä. Juuri kukaan toimittajista ei osannut vielä ohjelmoida, joten yhteistyö teknisten osaajien kanssa koettiin vaikeaksi. Jotta projekti saatiin vietyä maaliin, osapuolten piti ensin kehittää yhteinen kieli, jolla tehtävän vaatimuksista pystyttiin keskustelemaan. Projektin vetäjän mielestä ongelmilta olisi välttytty, jos hän olisi tiennyt enemmän ohjelmoinnista. Tällöin hän olisi pystynyt toimimaan välikätenä ja ohjeistamaan osapuolia paremmin.

Koodaustaitoisten ja koodaustaidottomien välisen viestinnän haastavuuteen on kiinnitetty huomiota myöhemminkin. Lewis ja Usher (2016, 556) tutkivat yhdysvaltalaisen Knight- ja Mozilla-säätiöiden yhteistä projektia, jossa koodarit ja toimittajat työskentelivät yhdessä, ja tekivät johtopäätöksiä koodareiden ja toimittajien suhteesta heidän keskusteluidensa, kysymystensä ja kommenttiansa perusteella. Heidän mukaansa koodareilla ja toimittajilla on erilaisia käsityksiä journalismista, mikä saattaa vielä osoittautua ongelmaksi koodareiden ja toimittajien välisessä suhteessa. Lewisin ja Usherin tutkimuksesta puhuttaessa pitää ottaa huomioon se, että siinä on tutkittu nimenomaan koodareiden ja toimittajien välistä yhteistyötä, ei koodaustaitoisten toimittajien käsityksiä journalismista.

Toisessa yhteydessä Lewis ja Usher (2014, 7) nostavat esille koodareiden ja toimittajien erilaiset suhteet aikaan. Lewis ja Usher tutkivat organisaatiota, joka pyrkii saattamaan koodarit ja toimittajat lähemmäksi toisiaan. He kertovat havainnoistaan organisaation järjestämässä tapahtumassa Rochesterissa vuonna 2012. Tutkijoiden mukaan toimittajat olivat enemmän kiinnostuneita lyhyen aikavälin jutuista, kun taas koodareita kiinnosti enemmän pidemmän aikavälin projektit. Rochesterissa monet toimittajat keskittyivät etsimään ratkaisua jatkuvasti kasvavaan työtaakkaansa, kun taas koodarit osallistuivat tapahtumaan ratkaistakseen suurempia ongelmia ja rakentaakseen jotakin, mistä he voisivat olla ylpeitä. Nämä erilaiset tavoitteet johtivat väistämättä väärinkäsityksiin. Tutkijoiden haastatteleman Matt Berniuksen mukaan monella teknisen alan osaajalla on perinteinen, sankarillinen käsitys toimittajuudesta, eivätkä nämä juuri anna arvoa niille paineille, joita esimerkiksi paikallislehtien toimittajat kohtaavat päivittäisessä työssään.

Powers (2012) tutki artikkeleita uutistuotannon tietokoneistumisesta journalismin ammattilehdissä vuosina 1975–2011. Hän huomasi haasteita siinä, miten mediataloissa suhtaudutaan, kun niille esitetään uusia työtapoja. Powersin (2012, 31–33) mukaan erityistaidot on koettu toimituksissa usein uhkana. Hän mainitsee kuvajournalismin yleistymisen 1900-luvun alussa, jolloin toimittajat

kyseenalaistivat valokuvaajien kyvyn toimia journalistisessa roolissa. Joitakin vuosikymmeniä myöhemmin, kun journalistisia juttuja alettiin julkaista tietokoneella, huolta aiheutti graafisten suunnittelijoiden tulo toimituksiin. Visuaalisten päätösten katsottiin vaikuttavan liiaksi perinteiseen journalistiseen päätöksentekoon. 1990-luvun lopussa verkkotoimittajat saivat osakseen samanlaisia epäluuloja. Powersin mukaan koodarit ovat tunteneet itsensä samalla tapaa vähemmän tervetulleiksi, koska toimittajat ovat pitäneet heidän työtänsä pelkästään teknisenä. Vuosituhannen vaihteessa monet koodarit vaihtoivatkin journalismin parista ohjelmistoalalle, koska he pettyivät siihen, että heitä pidettiin toimittajia alempiarvoisina.

Lehtniemi (2016a, 8–13) on listannut edellä mainittuja haasteita tutkimuksessaan moniammatillisten tiimien yhteistyöstä. Hän mainitsee Lewisin ja Usherin havainnot koodareiden ja toimittajien erilaisista aikakäsityksistä ja Powersin havainnon siitä, että erityistaitoja pidetään toimituksissa usein uhkana, mutta myös Charles Warnerin (2014) huomion siitä, ettei journalismissa hyväksytä virheitä. Warnerin mukaan uutisorganisaatiot ja journalismin säännöt on suunniteltu minimoimaan kaikki virheet. Tämä on täysin päinvastoin kuin ohjelmistoalalla, jossa työskennellään yrityksen ja erehdyksen kautta. Ohjelmistoalalla työskennelleet koodarit ovat tottuneet siihen, että he saavat rauhassa kokeilla uusia asioita. Heidän on tehtävä virheitä, jotta he voivat luoda uutta. Powersin havaintoon erityistaitojen pitämisestä uhkana liittyy myös Lehtniemen toteamus toimittajien eristäytyneisyydestä (Lehtniemi 2016a, 9). Toimittajilla on taipumus eristää itsensä henkisesti ja fyysisesti muista ammattiryhmistä. Tämän on katsottu johtuvan toimittajan ammatin avoimesta luonteesta. Koska alalle voi edelleen päästä ilman asiaankuuluvaa koulutusta, ryhdytään ”oikean journalismin” ja ”oikean toimittajan” käsitteitä pohtimaan usein vasta työelämässä.

2.2.5 Merkitys journalismille

Koodaustaitoiset toimittajat eivät ole juurikaan kertoneet omasta merkityksestään työyhteisölle tai journalismille. Sen sijaan muutama tutkijoiden tekemä havainto kertoo siitä, että toimitusten koodaustaitoista työvoimaa arvostetaan paljon. Ensinnäkin koodaustaitoisista toimittajista koostuvat tiimit saavat muilta toimituksen jäseniltä enemmän yhteistyöpyyntöjä, kuin mihin ehtivät tarttumaan. Esimerkiksi New York Timesissa jokaisen tiimille esitetyn ehdotuksen kohdalla mietitään, voiko projektin toteuttaa, onko dataa tarpeeksi saatavilla ja onko projekti kaiken vaivan arvoinen (Royal 2010, 16). Suuri kysyntä kertoo siitä, että tiimin työtä tarvitaan. New York Timesissa jokainen koodaustaitoinen toimittaja lisäksi tiesi, etteivät työt olleet hetkeen loppumassa. Kaikki tiesivät, että

ala on kasvava ja että heidän taitojaan arvostettaisiin myös tulevaisuudessa (em. 20). Toiseksi arvostuksesta kertovat palkinnot, joita koodaustaitoisten toimittajien jutut ovat voittaneet jo tietokoneavusteisen journalismin alkuajoista lähtien (Lewis & Usher 2014, 2). Vaikka koodarit ovat kokeneet itsensä toimittajia alempiarvoisiksi, ovat lukuisat koodaustaitoisille toimittajille jaetut palkinnot merkki siitä, etteivät heidän journalistiset saavutuksensa kalpene perinteisten toimittajien saavutusten rinnalla.

Journalismissa on sovellettu ohjelmointia jo pitkään, mutta koodaustaitoisten toimittajien todellista merkitystä voidaan vasta vain arvailla. Se, miten koodaustaitoiset toimittajat kertovat omasta työstään, paljastaa sen, mitä he haluavat sillä saavuttaa. Esimerkiksi Norjassa koodaustaitoiset toimittajat etsivät uusia tapoja löytää tarinoita ja kertoa tarinoita (Karlsen & Stavelin 2014, 42). Suurta tietomäärää analysoimalla he pyrkivät tukemaan olemassa olevia teorioita ja löytämään uusia totuuksia. Flewn, Spurgeonin, Danielin ja Swiftin (2012, 167) mielestä taulukoiden pyörittely muuttaa toimittajien panosta. Toimittajien ei tarvitse käyttää aikaansa faktojen etsimiseen, vaan he voivat keskittyä tarkistamaan ja selittämään ilmiöitä. Voidaan siis olettaa, että koodaustaitoisten toimittajien merkitys on suuri, sillä he pyrkivät päivittämään sekä journalistista sisältöä että toimitusten vakiintuneita työtapoja. Se on hyvä asia, koska journalismi on yleensä laahannut muihin aloihin nähden perässä, mitä tulee uusien teknologioiden omaksumiseen (Usher 2016, 11).

2.2.6 Journalistisen työn muutos

New York Timesin interaktiivinen osasto perustettiin vähentämään byrokratiaa koodaustaitoja tarvitsevien projektien aloittamisesta (Royal 2010, 15). Osaston perustamisen jälkeen projektien aloittamisesta tuli joustavampaa, koska osasto pystyi reagoimaan muuttuviin uutistilanteisiin muiden uutistiimien tapaan. Osaston toiminnan juuret ovat syvällä ”hakkerikulttuurissa” (em. 18). New York Timesin koodaustaitoisten toimittajien työtä ohjaavat luovuus ja avoimuus. ”Hakkerikulttuurin” edustajat haluavat tiedon keräämisestä ja tiedon esittämisestä läpinäkyvämpää ja juttuprojekteista kokeilevampia (Lewis & Usher 2013, 608, 613). Tässä tapauksessa hakkerilla ei tarkoiteta varasta, vaan ennemminkin henkilöä, joka osaa rakentaa muille työkaluja, joilla voi ratkaista teknisiä ongelmia (Usher 2016, 75). New York Timesissa interaktiivisen osaston kulttuuri on vaikuttanut muun toimituksen arkeen. Toimittajien on mahdollista saada juttuihinsa elementtejä, joilla asioiden esittäminen käy innovatiivisemmin ja lukijaystävällisemmin. Koodaustaitoisten toimittajien työnkuva riippuu toimituksesta, mutta New York Timesissa he toimivat myös toimittajien apukäsinä.

Toimittaja oli vastuussa projektista ja sai ehdottaa koodareille, mitä juttuunsa tarvitsi (Royal 2010, 15). Osasto on otettu toimituksessa hyvin osaksi päivittäistä tekemistä, koska se saa ehdotuksia enemmän kuin mihin aika riittää (em. 16).

Lehtniemen (2016, 20) kaikki haastateltavat olivat sitä mieltä, ettei tavallisen toimittajan tarvitse opetella tulevaisuudessa koodaamaan. Datajournalistien pitäisi osata ohjelmoida sujuvasti, mutta perinteisten toimittajien ei tarvitse hankkia itselleen tällaisia taitoja, vaikka he työskentelisivätkin koodareiden kanssa. Tärkeintä on ymmärtää, mitä koodaaminen on ja mitä sillä voidaan tehdä. Esimerkiksi Helsingin Sanomien Jarmo Lundgrenin mielestä on hyvä tietää, mikä on välimuisti, miten verkkosivu on rakennettu ja mitä CSS-kielellä tehdään. BBC:n Ransome Mpinin mielestä toimittajien olisi hyvä tietää, mitä ohjelmointi edellyttää, jotta toimittajat voisivat antaa projektille parhaan mahdollisen työpanoksensa (em. 21). Myös Lehtniemi itse on ottanut aiheeseen kantaa Suomen Lehdistön artikkelissaan. Hän esittää, että sen sijaan, että kaikki toimittajat opettelisivat koodaamaan, toimittajien kannattaisi opetella esimerkiksi johtamaan projekteja. Hän rohkaisee myös tutustumaan koodareiden maailmaan ja luomaan ymmärryksen siitä, miten koodi toimii (Lehtniemi 2016b).

Hannaford (2015, 18) on Lehtniemen kanssa samaa mieltä. Hän ei löytänyt mitään, mikä tukisi ajatusta siitä, että kaikkien toimituksen jäsenten pitäisi osata koodata. Hannaford teki päätelmiä keräämästään haastatteluaineistosta. Sen lisäksi hän keräsi journalismin työpaikkailmoituksia brittiläiseltä avoimista työpaikoista listaavalta sivustolta. Tämän aineiston 1 166 työpaikasta vain 24:ssä edellytettiin jonkinlaisia ohjelmointitaitoja. Vaikka mikään ei viittaa siihen, että Britanniassa etsittäisiin erityisesti koodaustaitoisia toimittajia, työnantajat näyttävät Hannafordin mukaan arvostavan toimittajia, joilla on pientä ymmärrystä HTML:stä. Tätä tukee Hannafordin huomio siitä, että monet mediatilat tarjoavat valmistuneille kursseja, joilta työnantajat toivovat löytävänsä koodaamisesta tai datan analysoimisesta kiinnostuneita ihmisiä (em. 19). Britanniassa myös jotkut yliopistot tarjoavat kursseja tietokoneavusteisesta, datavetoisesta ja interaktiivisesta journalismista. Näin on mahdollista, että suurempi osa journalismin opiskelijoista tutustuu ohjelmointia hyödyntävään journalismiin jo opiskeluaikoinaan. Hannafordin tutkimus osoittaa tarpeen ongelmia ratkoviille, uteliaille ja nopeasti uusia asioita oppiville toimittajille.

2.2.7 Journalismin muutos

”Hakkerikulttuuri” muuttaa myös itse journalismia. Se tarjoaa toimittajille ja koodareille mahdollisuuden miettiä journalismin uusia arvoja yhdessä (Lewis & Usher 2013, 615). ”Hakkerikulttuurin” tarkoituksena ei ole kumota journalismia pitkään ohjanneita hyveitä, vaan tarjota toimittajille työkaluja, joiden avulla journalismi pysyy digitaalisen kehityksen menossa mukana. Hakkeri tekee mielellään toistettavia palasia ja kokeilee asioita ilman pelkoa epäonnistumisesta. Tutkijoiden mukaan nämä arvot voidaan tuoda osaksi toimitusta, jolloin journalismissa käytettävät teknologiat muuttavat muotoaan. Uudet toimintatavat muuttavat journalismin merkitystä verkottuneella mediakentällä.

Lewisin ja Usherin (2016, 554) tutkimuksessa kävi ilmi, että toimittajat ja koodarit alkoivat puhua uutisista samalla tavalla, kun he olivat viettäneet keskenään tarpeeksi aikaa. He keskittyivät yhteisessä kielessään ennen kaikkea tulevaisuuden journalismin innovaatioihin. Tutkijat tekivät puheista kolme havaintoa. Ensinnäkin suuri osa toimittajista ja koodareista purki turhautumistaan siitä, että journalistiset tarinat on esitetty aina samalla tavalla. ”Tarinat on vangittu artikkelin muotoon”, sanoi toimittaja Jordan Wirfs-Brock. Toiseksi toimittajat ja koodarit toivoivat, että ihmiset kutsuttaisiin mukaan luomaan, muokkaamaan ja kommentoimaan uutisia (em. 555). Moni näki tavalliset ihmiset mahdollisina tiedon kerääjinä. Cohenin, Hamiltonin ja Turnerin (2011b) mukaan tämä kasvattaisi ihmisten halukkuutta osallistua poliittiseen päätöksentekoon. Lisäksi toivottiin työkalua, joka tuottaisi yhteenvetoja ihmisten kirjoittamista kommenteista uutisille. Niiden avulla ihmiset voisivat helposti jäsentää uutissisältöä, jonka kokevat tärkeäksi. (Lewis & Usher 2016, 555)

Chicago Tribunessa suurin osa nykyisistä koodaustaitoisista toimittajista ei jaa aiempien CAR-toimittajan näkemyksiä journalismista, vaikka ryhmillä on yhteiset juuret (Parasie & Dagiral 2012, 862). Uusien koodaustaitoisten toimittajien mielestä uutinen itsessään pitäisi nähdä tietokoneen avulla käsiteltävänä datana, ei pelkästään tarinana, joka on piilossa datan sisällä. Jokaisen vaiheen, jonka toimittaja juttunsa eteen käy läpi, voisi tehdä tehokkaammin tietokoneella, koska kaikki tieto on jäsenneltyä. Tietokoneavusteisen journalismin pioneeri Adrian Holovaty (2006) kertoo esimerkin: ”Suurta osaa tiedosta, jota toimittajat keräävät päivittäin, voi automatisoidusti leikellä ja paloittaa. Jos lehdessä on uutinen paikallisesta tulipalosta, haluan lukijana tietää palon raa’at faktat, eli milloin ja missä palo syttyi, keitä uhrit ovat, mikä on pelastuslaitoksen numero, paljonko pelastuslaitokselta

on matkaa palopaikalle, keitä palomiehet ovat, kuinka paljon kokemusta heillä on ja kuinka kauan palomiehillä kesti päästä palopaikalle.”

Datavetoinen journalismi edistää objektiivisuutta (Parasie & Dagiral 2012, 854). Sen lisäksi journalismin puolueettomuutta parantaa koodaustaitoisten toimittajien pyrkimys suunnitella uutissovelluksensa mahdollisimman pitkälle itse (em. 865–866). Mitä vähemmän he käyttävät muiden valmiita materiaaleja, sitä vähemmän he kokevat olevansa ulkopuolisten agendojen vaikutuksen alaisina. Siten he pitävät projektin langat omissa käsissään ja tietävät tarkkaan, miten sovellus toimii. Tietokoneavusteinen journalismi vähentää kustannuksia (Cohen et al. 2011a, 68). Cohenin, Hamiltonin ja Turnerin mukaan se voi jopa korvata taloudellisista syistä irtisanottuja toimittajia. Koodarit voivat löytää, varmistaa ja julkaista yhteiskunnallisesti kiinnostavia tarinoita halvemmalla. Flew, Spurgeon, Daniel ja Swift ovat samaa mieltä tietokoneavusteisen journalismin kustannustehokkuudesta (Flew et al. 2012, 168). Sen lisäksi he ovat sitä mieltä, että tietokoneavusteinen journalismi vapauttaa toimittajat haastavampien journalististen ongelmien ääreen. Siten tietokoneavusteisuus tekee journalismista myös laadullisesti parempaa.

3 AINEISTO JA MENETELMÄT

3.1 Aineiston kerääminen

Keräsin aineiston haastattelemalla seitsemää koodaustaitoista toimittajaa loka-marraskuussa 2018. Haastateltavistani kolme työskentelevät Ylen Plus-deskissä Helsingissä, kaksi Ylen Mediadeskissä Tampereella ja kaksi Helsingin Sanomien datadeskissä.

Lähetin ensimmäiset haastattelupyynnöt syyskuun lopussa ja aloitin haastatteluiden tekemisen lokakuun alussa. Lähestyin Ylen Plus-deskin ja Mediadeskin sekä Helsingin Sanomien datadeskin lisäksi Aamulehteä ja Alma Talentia, mutta marraskuun puoleen väliin mennessä en ollut saanut sovittua uusia haastatteluja. Sovin tapaamisesta jokaisen haastateltavan kanssa sähköpostitse, useimmiten niin, että lähestyin ensin tiimin esimiestä. Esimies, joka toimii tiimissä usein myös tuottajana, vastaa yleensä alaistensa työpäivien aikataulutuksesta. Esimies sitten kertoi minulle, olisiko hänen tiiminsä jäsenillä aikaa haastatteluille. Sovin tarkat haastatteluajankohdat suoraan haastateltavien kanssa. En lähettänyt heille etukäteen tarkkaa kysymysrunkoa, vaan mainitsin haastateltavani heitä muutamista teemoista. Haastattelut toteutettiin haastateltavien työpaikoilla: Plus-deskin haastattelut Ylen tiloissa Pasilassa, Mediadeskin haastattelut Mediapoliksessa Tohlopissa ja Helsingin Sanomien haastattelut Sanomatalossa Helsingissä.

Haastattelin koodaustaitoisia toimittajia temahaastattelun avulla. Mukanani oli nauhuri ja laatimani kysymyslista, jonka pyrin käymään kokonaan läpi. Kysymykset oli jäsennetty eri teemojen alle. Teemahaastattelulle on tyypillistä, ettei kaikilta haastateltavilta tarvitse kysyä samoja kysymyksiä, vaan riittää, että jokaisen haastateltavan kanssa käydään läpi samat teemat. Kysymysrungosta poiketen esitin siis myös lisäkysymyksiä eri aiheisiin liittyen. Haastateltavien vastaukset olivat eri pituisia, ja jokainen haastateltava kiinnitti vastauksissaan huomiota erilaisiin asioihin. Haastattelut kestivät keskimäärin 69 minuuttia, lyhyimmän ollessa 54 minuuttia ja pisimmän 101 minuuttia. Haastatteluiden jälkeen kävin ääninauhat läpi ja kirjoitin sekä kysymykset että vastaukset puhtaaksi paperille. Pyrin litteroimaan haastateltavien vastaukset mahdollisimman tarkasti niin, että ne sisältäisivät myös puhujien itsekorjaukset ja katkokset puheen sisällä. Keskityin kuitenkin enemmän asiasisältöihin kuin haastateltavien vuorovaikutukseen, joten kovin yksityiskohtainen litterointi ei ollut tarpeen (Ruusuvoori 2010, 356). Litteroinnin jälkeen yhden haastattelun keskimääräiseksi

kooksi tuli noin 14 sivua. Tulostin sivut, jotta pystyin tekemään aineistoon analyysia helpottavia merkintöjä.

3.2 Aineiston perustelut

Kävin keväällä 2018 alustavan keskustelun yhden koodaavan toimittajan ja yhden koodaustaitoisen tiimin esimiehen kanssa siitä, keitä minun kannattaisi haastatella ja mihin asioihin minun kannattaisi kysymyksissä keskittyä. Olin alun perin ajatellut haastattelevani koodaustaidottomia toimittajia ohjelmoinnin tulosta osaksi journalismia. Näiden keskustelujen kautta päädyin kuitenkin haastattelemaan koodaustaitoisia toimittajia, koska ajattelin, että näillä ihmisillä olisi ilmiöstä enemmän sanottavaa. Keskustelujen kautta minulle selvisi, kuinka pienen ammattikunnan koodaustaitoiset toimittajat muodostavat. Suomessa on heitä karkean arvion mukaan alle parikymmentä. Seitsemän haastateltavaani edustavat siis melko suurta joukkoa koko maan koodaustaitoisista toimittajista.

Haastatteluin keräämäni aineisto on aiheen kannalta parempi, kuin jos olisin etsinyt vastauksia kysymyslomakkeen avulla. Kun pyrin selvittämään ohjelmoinnin merkitystä journalismille ja toimitustyölle, halusin päästä syvemmälle haastateltavien ajatuksiin kuin mihin pelkällä kyselylomakkeella olisi päässyt. Haastatteluissa kyse on myös merkityksistä, joihin kuuluvat haastateltavien taustat ja asenteet. En analysoi litteroimaani aineistoa sanatarkasti, vaan pyrin selvittämään, mitä haastateltava on halunnut sanoa. Haastateltavien aitoa suhtautumista asioihin pystyy tarkastelemaan paremmin kasvotusten tapahtuvan vuorovaikutuksen kautta. Haastateltujen suhtautuminen toisiinsa ei ole tässä oleellista, joten sen vuoksi haastattelut on toteutettu vain haastatellun ja haastattelijan välisenä. Olisin voinut ottaa mukaan myös määrällisen aineiston koodaustaitoisten toimittajien tekemistä jutuista, mutta se olisi kasvattanut työmäärääni kohtuuttomasti. Tutkimukseni tarkoituksena on kuitenkin selvittää pääasiassa koodaustaitoisten toimittajien työkäytäntöjä, eikä niinkään luokitella niitä artikkeleita, joissa on tarvittu koodaustaitoja. Minulla ei ole myöskään syytä epäillä, etteivätkö koodaustaitoisten toimittajien tekemät jutut olisi sellaisia, kuin he ovat sanoneet niiden olevan. Lopullisesta työstä ei käy ilmi, kuka haastateltavista on minkäkin vastauksen antanut, joten haastateltavien ei ole tarvinnut liioitella omaa tai tiiminsä osaamista.

3.3 Kysymysrunko

Jäsentelin esittämäni kysymykset seitsemän teeman alle. Näiden teemojen rajoissa kysyin useita kysymyksiä sen perusteella, miten kattavasti haastateltava niihin vastasi ja mitä hän oli aiemmin kertonut. Teemat käytiin läpi suunnilleen aina tässä järjestyksessä, vaikka toisinaan teemojen välillä liikuttiin sen mukaan, mistä haastateltava halusi kertoa. Joissakin tapauksissa esitin itsekin toiseen teemaan kuuluvan kysymyksen, jos katsoin sen sopivan loogisuutensa puolesta paremmin johonkin toiseen kohtaan. Haastattelua sopiessani mainitsin muutamia teemoja, joista voisimme puhua. Jokaisen haastattelun alussa kertosin teemat vielä haastateltavalle.

Teemat:

1. Haastateltavan tausta
2. Ohjelmointi käsitteenä
3. Koodaustaitoisten toimittajien taidot
4. Koodaustaitoisten tiimien merkitys
5. Koodaustaidottomien toimittajien taidot
6. Journalismin muutos
7. Tulevaisuudennäkymät

Jos katsoin tarpeelliseksi poiketa ennalta laatimastani listasta ja kysyä jotain ylimääräistä, se johtui yleensä siitä, että haastateltava jätti jotakin oleellista sanomatta. Esimerkiksi ensimmäisen teeman kohdalla haastateltava unohti usein kertoa, mistä oppilaitoksesta on aikanaan valmistunut. Joskus esitin kysymyksen uudelleen eri tavalla, jos haastateltava ei osannut vastata kysymykseen, hän ei ymmärtänyt kysymystä tai hän kiersi vastauksen vastaamalla asian vierestä. Tätä tapahtui haastavampien kysymysten kohdalla ainakin silloin, kun puhuimme ohjelmoinnin merkityksestä journalismille. Muutamissa tapauksissa huomasin vasta parin ensimmäisen haastattelun jälkeen, kuinka kiinnostavaa olisi ollut kysyä jostakin tietystä yksityiskohdasta. Nämä eivät olleet välttämättömiä kysymyksiä, mutta ne olisivat auttaneet minua täydentämään kokonais kuvaa haastateltavieni toimenkuvasta. Haastattelun lopuksi kysyin aina, jäikö haastateltavien mielestä jotakin sanomatta, mikä tarjosi haastateltaville vapaan sanan sanoa mitä tahansa aiheeseen liittyen. Tämä nosti esiin usein mielenkiintoisia huomioita, joiden kautta saatoin vielä lopuksi esittää muutamia kysymyksiä. Haastattelun jälkeen monet haastateltavista kysyivät, menikö haastattelu

hyvin. Korostin heille sitä, ettei oikeita tai vääriä vastauksia ole olemassa. Kaikki kiinnittävät huomiota erilaisiin asioihin, joten haastattelut eivät oikeastaan olisi voineet epäonnistua.

3.4 Teemahaastattelu

Aineistonkeruumenetelmäni on teemahaastattelu. Sille ominaista on se, ettei haastattelu etene tarkkojen, valmiiksi muotoiltujen kysymysten kautta, vaan väljemmin ennalta suunniteltujen teemojen mukaisesti (Saaranen-Kauppinen & Puusniekka 2006). Teemat ovat kaikille haastateltaville samat, koska ne on laadittu etukäteen aiemman tutkimuskirjallisuuden ja alustavien keskustelujen pohjalta. Kävin itse keskustelua teemoista sekä keväällä haastattelemieni asiantuntijoiden että Parasien ja Dagiralin artikkelin kanssa. Teemahaastattelussa ei ole kyse kysymysrunгон orjallisesta noudattamisesta, vaan teemojen sisällä voidaan liikkua joustavasti. Esitin haastattelutilanteessa useasti tarkentavia kysymyksiä ja keksin uusia, jos huomasin jo itse tilanteessa, että jostain asiasta olisi hyvä kysyä. Tämä johti moniin mielenkiintoisiin vastauksiin. Samalla huomasin, että haastateltava rentoutui, jos esitin silloin tällöin lyhyitä lisäkysymyksiä jähmeiden, ennalta määriteltyjen siirtymien sijaan. Tarkentavat kysymykset korostivat haastatteluiden loogisuutta ja ilmaisivat haastateltavalle, että analysoin hänen sanomaansa jatkuvasti. Saaranen-Kauppinen ja Puusniekan mukaan teemahaastattelu on keskustelunomainen tilanne. Siinä pyritään huomioimaan ihmisten tulkinnat ja heidän merkityksenantonsa. Siksi en analysoi tutkimusaineistoani sanasta sanaan, vaan pyrin selvittämään sanoman merkityksen. Teemahaastattelussa on tärkeää antaa tilaa haastateltavalle. Sanoin ennen haastatteluja jokaiselle haastateltavalle, että vastauksiin saa käyttää niin paljon aikaa, kuin vain tarvitsee. En myöskään kokenut missään vaiheessa tarpeelliseksi keskeyttää haastateltavaa, jos hänen vastaamisensa oli kesken. Haastateltavan oli tärkeää saada sanottua juuri ne asiat, jotka hänen mielessään olivat, jotta pystyin myöhemmin tekemään johtopäätöksiä siitä, millaisiin aiheisiin haastateltavani tarttuivat. Tämä johti siihen, että lyhyimmillään haastattelu kesti alle tunnin verran ja pisimmillään yli puolitoista tuntia.

Teemahaastattelu edellyttää perehtymistä aihepiiriin (Saaranen-Kauppinen & Puusniekka 2006). En kuitenkaan tarvinnut itse enempää perehtymistä, jotta pystyin käymään keskustelua teknisistä termeistä, järjestelmistä tai alan käytännöistä. Olen opiskellut tietojenkäsittelytieteitä yliopistossa ja kesätoimittajana olen toteuttanut juttuja koodaavien toimittajien kanssa. Tämän myötä tiesin myös, keitä minun kannattaisi tutkimukseen haastatella. Haastateltavilta sain kuitenkin lisää vinkkejä ihmisistä, joita minun kannatti tavoitella. Koodaustaitoisia toimittajia ei kuitenkaan ole jonoksi asti,

vaan haastateltavat työskentelevät kaikki toimituksissa, jollaisia on Suomessa vain muutamia. Perehtymisen myötä osasin esittää jatkokysymyksiä ja analysoida kuulemaani jo itse tilanteessa. Teemahaastattelulle onkin tyypillistä, että aineistoa tarkastelee jo aineistonkeruuvaiheessa (Hirsjärvi & Hurme 2008, 136). Hirsjärven ja Hurmeen mukaan analysointi alkaa jo haastatteluvaiheessa, sillä haastattelija tekee jo silloin havaintoja ilmiöistä niiden useuden, toistuvuuden ja jakautumisen perusteella. Tein siis jo haastatteluvaiheessa kiinnostavia huomioita siitä, miten samankaltaisia tai erilaisia asioita haastateltavat tuovat puheissaan esiin. Tämä helpotti myöhemmin aineiston analyysia, koska olin käynyt aineiston läpi kertaalleen jo aikaisemmin. Saaranen-Kauppinen ja Puusniekan mukaan teemahaastattelun suosio perustuu sen vapauteen. Haastateltava saa puhua haluamallaan tavalla, eikä haastattelijan tarvitse kuin myöhemmin jäsenellä samaansa tietoa. Koska haastattelu on toteutettu teemoittain, sen läpikäyminen on suhteellisen helppoa.

Haastattelu on joustava tutkimusmenetelmä (Hirsjärvi & Hurme 2008, 34). Se sopi omaan käyttötarkoitukseeni hyvin, koska minulla ei ollut selkeää käsitystä siitä, millaisia tuloksia voisin tutkimuksellani saavuttaa. Hirsjärven ja Hurmeen mukaan haastattelun etuja ovat muun muassa se, että haastateltava nähdään siinä subjektina (em. 35). Tämä tarkoittaa sitä, että tutkimuskohteena on oikea ihminen tilastojen ja todennäköisyyksien sijaan. Haastateltavan sanomalla on merkitys, ja on tärkeää, että ihminen saa ilmaista itseään haastattelutilanteessa vapaasti. Näin saatu tutkimusaineisto sisältää inhimillisempää, konkreettisempää ja käytännöllisempää tietoa, kuin mitä esimerkiksi tilastoja tutkimalla olisi saatu aikaan. Kuten Hirsjärvi ja Hurme toteavat, haastattelun avulla on mahdollista saada selville vastausten taustalla olevia motiiveja. Aineisto ei siis koostu pelkästään sanoista, vaan myös merkityksistä niiden takana. Joitakin ongelmiakin haastattelulla on, kuten se, että haastattelemisen vaatii haastattelijalta ammattitaitoa. Haastattelijan pitää esimerkiksi osata esittää kysymyksiä, joihin ei vahingossa itse samalla vastaa. Haastattelut vievät myös paljon aikaa, joten jonkin muun aineistonkeruumenetelmän avulla olisin päässyt aloittamaan aineiston analyysin nopeammin. Koen, että haastattelun suurin ongelma on sen vapaus. Kuten Hirsjärvi ja Hurmekin sanovat, vapaamuotoisen haastatteluaineiston työstäminen on haasteellista, koska valmiita malleja ei ole tarjolla. Jokainen tutkija toimii omalla tavallaan, joten joku muu tutkija olisi saattanut toteuttaa haastattelut eri tavalla. Tiedostan kuitenkin ongelman, ja olen sen vuoksi kertonut tutkimuksen vaiheista mahdollisimman läpinäkyvästi.

3.5 Laadullinen analyysi

Tarkastelin keräämäni aineistoa laadullisesti. Laadullisen analyysin tekemiseksi on olemassa monenlaisia ohjeita. Usein se alkaa määrittelemällä, mikä aineistossa on kiinnostavaa (Tuomi & Sarajärvi 2018, 78). Tuomen ja Sarajärven mukaan laadullisen tutkimuksen analyysimenetelmät perustuvat aina tavalla tai toisella sisällönanalyysiin, ja myös omassa tutkimuksessani tarkastelen haastateltavien sanomisia juuri sisällönanalyysin kautta. Joka kerta, kun löysin aineistosta jotakin kiinnostavaa, merkkasin ne kohdat, jotka kertovat tästä kiinnostuksesta. Tuomen ja Sarajärven ohjeiden mukaisesti luokittelin ja teemoittelin aineistoa (liite 1). Luokitellessani aineistoa vertailin haastateltavien vastauksia keskenään ja katsoin, kuinka usein haastateltavat viittaavat niissä eri asioihin. Analyysivaiheessa jäsentelin haastateltavien vastauksia heidän mielipiteidensä mukaan. Laadulliseen analyysiin kuuluu vahvasti tutkimusaineiston teemoittelu, jolla aineistosta pyritään nostamaan esille sen olennaiset osat (Tuomi & Sarajärvi 2018, 78; Hirsjärvi & Hurme 2008, 137). Teemoittelussa on kyse laadullisen aineiston pilkkomisesta ja ryhmittelystä aihepiirien mukaan. Se on siis vähän kuin luokittelua, mutta siinä otetaan huomioon suurempi määrä tietoa ja etsitään syitä vastauksissa esitetyille mielipiteille. Lopuksi analyysissä voitaisiin ryhmitellä aineisto tyypeiksi. Siinä teemojen sisältä etsitään yhteisiä ominaisuuksia, joiden perusteella muodostetaan eräänlainen yleistys eli tyyppiesimerkki. Kaikki edellä mainittu tähtää mielestäni aineiston jäsentämiseen, jonka tarkoitus on tehdä suuresta aineistomäärästä tutkijalle selkeämpi.

Hirsjärvi ja Hurme (2008, 137) ovat esittäneet erilaisia tapoja lähestyä aineiston analyysia, riippuen siitä, kuka analyysiä tekee. Omassa tutkimuksessani tulkitseen litteroitua aineistoa yksin. Ensimmäisessä vaiheessa pyritään selvittämään aineiston rakenne. Tämä on välttämätöntä, jos litteroitua aineistoa on paljon. Sen jälkeen aineistoa aletaan selventää, eli siitä poistetaan sinne kuulumattomia osia, kuten toistoja ja tarpeettomia puhekielisiä äännähdyksiä. Kolmannessa vaiheessa, eli varsinaisessa analyysissa, aineistoa tiivistetään, luokitellaan ja tulkitaan ja sieltä etsitään narratiiveja. Hirsjärven ja Hurmeen ehdottama kolmas vaihe on siis melkein kuin Tuomen ja Sarajärven analyysivaihe, jossa aineistoa luokitellaan, teemoitellaan ja tyyppitellään. Hirsjärven ja Hurmeen esittämä tiivistäminen merkitsee sitä, että haastateltavan esiin tuomat merkitykset puetaan lyhyempään sanalliseen muotoon. Sitä tein osaksi jo litterointivaiheessa, koska en sisällyttänyt aineistoon aivan kaikkia täytesanoja. Tiivistetysti analyysitapani on hyvin suoraviivainen (em. 144). Kun litteroin aineiston, luin sen läpi, luokittelin sitä ja etsin yhteyksiä asioiden väliltä. Lopuksi

raportoin löydöksistäni. Laadullisessa tutkimuksessa ei ainoastaan referoida aineistoa, vaan juuri yhteyksien löytyminen asioiden väliltä tuottaa uutta tutkimustietoa.

Menetelmäni on luonteeltaan induktiivinen (Tuomi & Sarajärvi 2018, 80; Saaranen-Kauppinen & Puusniekka 2006). Induktiivisella tarkoitetaan aineistolähtöisyyttä. En testaa ennalta päättämäni teoriaa haastateltavilla, vaan tutkimuksessa käytettävän teorian määrää aineisto, jonka muodostavat tekemäni teemahaastattelut. Tutustuin siis aiempaan kirjallisuuteen vasta sitten, kun olin käynyt läpi keräämäni aineiston. Tuomen ja Sarajärven määritelmän mukaan induktiivinen sisällönanalyysi tarkoittaisi myös analyysia yksittäisestä yleiseen. Aineistosta nousi esille yksittäistapauksia, haastateltavien mielipiteitä, joita sitten myöhemmin yhdistin osaksi suurempaa kuvaa, kun vertasin löytämiäni luokkia ja teemoja aiempaan tutkimuskirjallisuuteen. Tuomen ja Sarajärven luonnehdinnan (em. 81) mukaan aineistolähtöisessä analyysissa tutkimusaineistosta pyritään luomaan teoreettinen kokonaisuus. Ajatus siinä on se, ettei tutkija vielä aineistonkeruuvaiheessa tiedä, mihin kohtiin hän tulee analyysivaiheessa kiinnittämään huomiota. Tutkittavasta ilmiöstä tehtyjen aiempien havaintojen, tietojen tai teorioiden ei pitäisi vaikuttaa analyysin toteuttamiseen tai lopputulokseen, koska analyysin oletetaan olevan aineistolähtöistä. Tämä lähestymistapa sopii tutkimukseeni hyvin, koska en ennen haastatteluja tiennyt, kuinka paljon haastateltavat tulevat sivuamaan aiemmista tutkimuksista tuttuja aiheita. Aineistolähtöinen lähestymistapa vähentää myös sitä oletusta, että tutkimuksen lopputulokset olisivat riippuvaisia tutkijasta. Siinä tutkija ei määritä sitä, mikä on tärkeää, vaan sen tekevät haastateltavat vastaustensa kautta.

Aineistolähtöistä analyysia on arvosteltu muun muassa sen objektiivisuuden puutteesta (Tuomi & Sarajärvi 2018, 81). Aineistolähtöinen analyysi olettaa, ettei objektiivisia havaintoja ole olemassakaan, vaan muun muassa jo käytetyt käsitteet, tutkimusasetelma ja menetelmät ovat tutkijan asettamia ja vaikuttavat aina tuloksiin. Analyysissa pitää kiinnittää huomiota siihen, tapahtuuko se todella aineiston tiedonantajien ehdoilla eikä tutkijan ennakkoluulojen saattamana. Pohdin tätä ongelmaa myös haastatteluita tehdessäni, sillä väistämättä jokaiselle esittämälleni kysymykselle on olemassa peruste. Kuten Tuomi ja Sarajärvi sanovat, ongelma on laadullisessa tutkimuksessa yleisesti myönnetty. Pyrin ongelmista huolimatta esittämään kysymykseni mahdollisimman neutraalilla tavalla. Kun esimerkiksi kysyin, onko journalismin muutos haastateltavan mielestä hyvä vai huono asia, en jättänyt esittämättä jompaakumpaa puolta asiasta. En myöskään esittänyt kysymyksiä niin, kuin olettaisin tietäväni vastauksen ennalta. Ongelmista huolimatta kysyin kysymykset haastateltavilta aina samalla tapaa, joten haastateltavien vastausten erot eivät ole riippuvaisia haastattelijasta. Haastateltavat myös ilmaisivat aina selkeästi, jos jokin kysymyksen tiedoista ei

pitänyt heidän mielestään paikkansa. Tuomi ja Sarajärvi ehdottavat ratkaisuksi teoriaohjaavaa analyysia (em. 82), jossa aineistonkeruussa on mukana teoreettisia kytkeitä. Tätä sovelsin osittain, sillä osa kysymyksistäni perustuu yleiseen keskusteluun mediassa.

Ruusuvuori (2010, 13) kehottaa varmistamaan, vastaako aineisto tutkimuskysymyksiin riittävällä tavalla ja onko perusteltua olettaa, että aineiston pohjalta tehty tutkimus tuottaa luotettavaa ja yleistettävissä olevaa tietoa. Aineistolta on esimerkiksi turha kysyä sellaisia kysymyksiä, jotka eivät riitä luomaan tarpeeksi kattavaa kuvaa tutkimuksen kohteina olevista ihmisistä. Ruusuvuoresta kysymykset haastateltavien enemmistön tai vähemmistön näkemyksistä sopivat paremmin määrälliseen kuin laadulliseen tutkimukseen. Pienemmältä joukolta voi kysyä esimerkiksi heidän kokemuksistaan tai näkemyksistään. Pystyn itse hyödyntämään tutkimuksessani monenlaisia kysymyksiä, koska koodaustaitoisten toimittajien ammattiryhmä on niin pieni, että voin tehdä oletuksia koko joukosta. Haastateltavani muodostavat koko Suomessa toimivista koodaustaitoisista toimittajista suuren osan. Samalla haastateltavani ovat kuitenkin yksilöitä, ja jokainen kokee edustamansa ammattiryhmän eri tavalla. Tutkimuksessani annoin heille mahdollisuuden vastata kysymyksiin niin laajasti kuin he halusivat. Laadulliselle analyysille tyypillisemmällä kysymyksillä sain tietoa vastausten taustalla olevista ilmiöistä.

4 KOODAUSTAITOISTEN TOIMITTAJIEN KOKEMUKSET

4.1 Ohjelmointi arjen toimitustyössä

4.1.1 Koodaustaitoisten toimittajien taustat

Haastattelemillani koodaustaitoisilla toimittajilla on hyvin erilaisia koulutustaustoja. Seitsemästä haastateltavasta kaksi on kouluttautunut toimittajaksi, toinen ammattikorkeakoulussa ja toinen yliopistossa. Neljällä haastateltavallani on koulutus selkeästi teknisemmältä alalta. Heistä yksi on opiskellut yliopistossa teoreettista fysiikkaa, yksi tietojenkäsittelytieteitä ja vuorovaikutteista teknologiaa ja kaksi ovat koulutukseltaan diplomi-insinöörejä. Yksi haastateltavista on opiskellut yliopistossa sosiaalipoliittikkaa. Näin ollen suurimmalla osalla haastateltavistani on koulutus tekniseltä alalta ja vähemmistössä ovat ne, jotka ovat opiskelleet korkeakoulussa toimittajiksi. Usherin (2016) ja Hannafordin (2015) tutkimuksissa tilanne oli päinvastoin. Haastateltavistani ne, joilla on teknisen alan koulutus, kiinnostuivat journalismista vasta myöhemmin ja ne, jotka kouluttautuivat ensin toimittajiksi, kiinnostuivat ohjelmoinnista vasta myöhemmin.

On kiinnostavaa, ettei juuri kukaan haastateltavista ole opetellut ohjelmoimaan kouluttautuessaan, vaikka niin moni heistä on opiskellut teknistä alaa. Teknisellä alalla opiskelleiden mukaan koulutus tarjoaa perusteet ohjelmoinnista, mutta todellinen oppiminen tapahtuu silti vapaa-ajalla. Koodaustaitoisten toimittajien mukaan heidän ohjelmointitaitonsa ovat tämänhetkisellä tasolla sen ansiosta, että he ovat harjoitelleet ohjelmointia omissa vapaa-ajan projekteissaan. Kuten Royalin (2010) tutkimuksessa, jokainen haastateltavistani korosti, että ohjelmoinnin opettelu on vaatinut omistautumista vapaa-ajalla. He ovat rakentaneet ja päivittäneet omia henkilökohtaisia nettisivujaan ja sovelluksiaan. Siitä, kuinka paljon aikaa asioiden opettelu on vienyt aikaa, kertoo muun muassa tämä haastateltavan sitaatti:

”Oon vaan tehny tosi paljon omalla ajalla, että niinku jos jollain ihmisillä on ruuhkavuodet sen takia, että ne perustaa perheen, niin mulla oli ruuhkavuodet sen takia, kun mä opiskelin niitä asioita. Niinku vähän tavallaan, että mä käytin kaiken ajan tähän, tämmöseen opiskeluun. Mut se oli niinku intohimo, en mä pitäny sitä opiskeluna.”

Haastateltavistani monet ovat luoneet omille nettisivuilleen työkaluja, jotka liittyvät heidän harrastuksiinsa tai kiinnostuksenkohteisiinsa. Esimerkiksi yksi heistä on luonut omaan käyttöönsä verkkosovelluksen, josta hän voi helposti tarkistaa omistamansa levyt, kun hän käy levykaupassa. Toinen haastateltava on puolestaan perustanut sivuston, jonne hän julkaisee tekemäänsä infografiikkaa. Vapaa-ajan juttuprojektien lisäksi ohjelmoinnin opetteleminen on joillakin edellyttänyt oma-aloitteista kouluttautumista. Haastateltavat kehuvat verkon ilmaiskursseja ohjelmoinnista. Yksi on opetellut tekemään sovelluksia Codeacademyn kursseilla, ja moni on harjoitellut ohjelmointia netistä löytyvien esimerkkien avulla. Verkossa on useita sivustoja, joilla ihmiset auttavat toisiaan käytännön ohjelmointipulmien ratkomisessa. Osaltaan haastateltavieni ohjelmointitaitoihin vaikuttaa myös ympäristö. Yksi on ollut kiinnostunut tietokoneista lapsuudesta asti ja opetellut ohjelmoimaan jo nuorena. Toisella on puolestaan monia ohjelmointitaitoisia kavereita, joiden esimerkkiä on ollut helppo seurata.

Haastattelemi koodaustaitoiset toimittajat ovat hyvin nuoria. Sen vuoksi monella ei ole takanaan kovinkaan pitkää työhistoriaa. Suuri osa on työskennellyt samassa työpaikassa valmistumisestaan saakka, osa useamman vuoden ja osa haastatteluhetkellä vasta muutamia kuukausia. Haastateltavista kuudella on aiempaa kokemusta journalismista, joko aiemmista töistään tai opinnoistaan, vaikkei kaikilla olekaan toimittajan koulutusta. Näistä kuudesta yksi on ollut aiemmin töissä toisessa journalistisessa julkaisussa. Yksi on koulutukseltaan toimittaja ja tullut töihin suoraan koulun penkiltä työharjoittelun kautta. Kolme on työskennellyt ensin tavallisena toimittajana ja siirtynyt sitten sen jälkeen hyödyntämään työssään ohjelmointia. Yksi on tehnyt toimittajan töitä aikaisemmin opintojensa ohessa. Yhdellä haastateltavista ei ole aiempaa kokemusta journalismista, vaan hän on innostunut siitä työskenneltyään ensin ohjelmistoalalla. Näin ollen suurimmalla osalla haastattelemistani koodaustaitoisista toimittajista on kokemusta journalismista. He ovat joko itse työskennelleet toimittajina tai he ovat työskennelleet toimituksessa toimittajien kanssa. Hakeutumisessa koodaavan toimittajan tai datajournalistin tehtävään on erotettavissa kaksi trendiä. Ne, jotka ovat kouluttautuneet teknisellä alalla tai työskennelleet aiemmin jonkin muun kuin journalismin parissa, ovat kiinnostuneet journalismista vasta myöhemmin. Samoin ne, jotka ovat joko kouluttautuneet toimittajiksi tai työskennelleet ensin toimittajina, ovat kiinnostuneet ohjelmoinnista tai muista teknisistä asioista vasta myöhemmin. Aikaisemmin toisessa journalistisessa julkaisussa työskennellyt päätyi tehtävään tekniseltä alalta sattuman kautta. Häntä ja kahta muuta haastateltavaa journalismin alalle ajoi alun perin kiinnostus grafiikkaan ja tiedon esittämiseen. Yksi haastateltavista päätyi tämänhetkiseen tehtäväänsä, kun hän kouluttautui ensin teknisellä alalla, mutta halusi siitä huolimatta työskennellä ennemmin toimittajana. Yksi halusi journalismista ensisijaisesti uusia

kokemuksia. Toimittajataustaisista haastateltavista toinen kiinnostui ohjelmoinnista datajournalismin kurssilla ja toinen palkattiin datajournalistiksi sen jälkeen, kun hän alkoi hyödyntää yhä useammin ohjelmointia omissa jutuissaan.

Koodaustaitoisten toimittajien ammatillinen osaaminen muodostuu ohjelmointiosaamisesta ja toimittajaosaamisesta. Heidän työnsä on puoliksi ohjelmointia ja puoliksi journalismia. Koodaustaitoisista toimittajista tekee erityisiä heidän ohjelmointiosaamisensa, jota ei ole toimituksen muilla jäsenillä juuri yhtään. He tuottavat journalistista sisältöä, mutta ei perinteisellä tavalla, vaan ohjelmointia hyödyntämällä. Siitä huolimatta haastattelemani koodaustaitoiset toimittajat pitävät itseään ensisijaisesti toimittajina, kuten Usherin (2016) haastattelema Michelle Minkoff. He perustelevat ammatti-identiteettiään sillä, että heidän työnsä lopputuote on journalismia. He työskentelevät uutistoimituksessa ja osallistuvat journalististen juttujen tekemiseen. Koodaustaitoiset toimittajat pitävät itseään toimittajina, vaikka he eivät olisi koulutukseltaan toimittajia. Toimittajaidentiteetti ei silti synny hetkessä, vaan se edellyttää tarpeeksi monen journalistisen päätöksen tekemistä. Tekniseltä alalta tulleiden toimittajaidentiteetti rakentuu sitä mukaa, kun he saavat ja haluavat ottaa osaa journalistiseen päätöksentekoon. Kahden teknisellä alalla kouluttautuneen haastateltavani mukaan työ on aluksi enemmän ohjelmointia, jos kokemusta journalismista ei ennestään ole. Tällöin journalistista vastuuta ottaa yleensä samassa projektissa mukana oleva toimittaja ja koodaustaitoinen toimittaja keskittyy jutun teknisten osien rakentamiseen. ”Mulla on vielä silmä jotenkin huono sellaiseen (journalismiin), joten olen ollut aika puhtaasti sellainen tekninen toteuttaja”, arvioi toinen haastateltavista. Toinen puolestaan sanoo ammatti-identiteettinsä muuttuneen vuosien aikana: ”Kyllä mä varmaan nykyään esittelen itseni toimittajana, mutta sitten jos keskustelu kääntyy monipolvisemmaksi niin ehkä sitten määrittelin sen myös sinne koodaamisen puolelle.” Koodaavien toimittajien ja datajournalistien työ on hyvin visuaalista. Siitä huolimatta haastateltavat kokevat heidän työnsä olevan enemmän ohjelmointia kuin visualisointia. Lopputulos on aina visuaalinen, mutta koska samassa tiimissä työskentelee usein ainakin yksi graafikko, eivät koodaustaitoiset toimittajat pidä visuaalista silmää niin tärkeänä.

Koodaustaitoisia toimittajia ajavat eteenpäin erilaiset asiat. Osa haluaa olla kehittämässä journalistisia juttuja lukijaystävällisempään suuntaan ja osa kehittää omia teknisiä taitojaan. Mielenkiinnon kohteissa on eroja riippuen siitä, millaisista taustoista koodaustaitoinen toimittaja tulee. Teknisellä alalla kouluttautuneita ajaa etenkin halu kehittää journalismia. Heitä kiinnostaa tehdä journalismia, joka on merkityksellistä. Yksi heistä haluaa tehdä omilla taidoillaan sisältöä, joka tulee lähelle ihmistä. Hän haluaa muuttaa kerronnan tapoja ja saada jotakin ”päryttävää” aikaiseksi.

Toinen haluaa havainnollistaa asioita entistä paremmin lukijoille: ”Hienoilla visualisaatioilla voidaan tarjoilla lukijoille vau-efektejä, sellaista silmäkarkkia ja mielihyvää, joka tekee juttujen lukemisesta miellyttävää ja koukuttaa ihmisiä jutun äärelle.” Alalle voivat ajaa kuitenkin myös suuremmat asiat. Ohjelmistoalalta journalismin pariin vaihtanut haastateltavani kertoi haluavansa vaikuttaa yhteiskunnallisiin asioihin ohjelmoijana. Hänellä on siis samanlainen motiivi kuin Parasien ja Dagiralin haastattelemissa koodaritaustaisilla toimittajilla (2012). Teknologian kehitys pakottaa uudistumaan. Haastattelemani koodaustaitoisia toimittajia ajaa eteenpäin voimakas itsensä kehittämisen tarve. Etenkin toimittajakoulutuksen saaneet haluavat oppia uusia työvälineitä ja kehittää teknisiä taitojaan. Mielenkiintoa ammattiryhmän sisällä herättää etenkin syväoppiminen, koneoppiminen, tekoäly, AR-teknologia ja VR-teknologia. Uusia taitojaan koodaustaitoiset toimittajat pyrkivät tuomaan esille journalistisissa juttuprojekteissaan.

4.1.2 Työnkuva ja taidot

Haastateltavistani viisi ovat koodaavia toimittajia, yksi on datajournalisti ja yksi datatuottaja. Työtä tehdään vahvuuksillaan, joten koodaustaitoisten toimittajien työnkuvat eroavat paljon toisistaan. Pääsääntöisesti he vastaavat journalistisen sisällön teknisestä toteutuksesta. Työtehtävät ovat tuttuja aiemmasta kirjallisuudesta. Koodaustaitoiset toimittajat tekevät muun muassa erikoistaittoja ja erilaisia interaktiivisia sovelluksia. He tekevät karttoja, kuvaajia, laskureita ja infografiikoita. Näkyvän työn lisäksi he tekevät taustaselvityksiä, joita toimittajat voivat hyödyntää jutuissaan. Esimerkiksi Ylen Mediadeskissä koodaava toimittaja voi lähettää kunnille tietopyynnön, koota ja analysoida tiedot ja toimittaa sitten valmiin, jäsennellyn paketin toimittajille. Sen sijaan Helsingin Sanomien datajournalistin työ muistuttaa perinteisen toimittajan työtä. Haastattelemani datajournalisti kertoo tekevänsä datavetoisia uutisia. Tietoa näitä juttuja varten hän hankkii joko valmiiden rajapintojen kautta tai haravoimalla verkkosivuja (engl. web scraping). Hänen tyypilliseen projektiinsa kuuluu suurten csv-tiedostojen käsitteleminen selkeämpään muotoon Excelillä tai Rubylla. Toisin kuin Ylen koodaavien toimittajien, Helsingin Sanomien datajournalistien ei tarvitse tuottaa muille toimittajille elementtejä uutisjuttuihin. Helsingin Sanomilla on oma Arkku-työkalu, jolla jokainen toimittaja voi lisätä juttuunsa karttoja, koneita ja erikoistaittoja, vaikkei osaisikaan ohjelmoida. Helsingin Sanomissa työskentelevä datatuottaja puolestaan kehittää Helsingin Sanomien Latoja-työkalua. Se on uutisautomaatiotyökalu, jolla datasta voidaan laatia automaattisesti uutisjuttuja, grafiikoita tai hälytyksiä, esimerkiksi metroliikenteen häiriöistä. Haastattelemani datatuottaja vastaa työkalun teknisestä parantamisesta. Kuten useimmat Usherin (2016) tutkimista

yhdysvaltalaisista toimituksista, molemmat tarkasteluni kohteena olevat toimitukset tarjoavat lisäksi erityisosaamistaan muille toimituksen jäsenille.

Ylen Plus-deski Helsingissä koostuu kolmesta koodaavasta toimittajasta, kolmesta graafikosta ja kahdesta tuottajasta, joista toinen on esimies. Ylen Mediadeskissä Tampereella työskentelee kaksi koodaavaa toimittajaa, kaksi graafikkoa, yksi tutkiva toimittaja, yksi editoiva toimittaja ja kaksi tuottajaa, joista toinen on esimies. Helsingin Sanomien datadeskissä on puolestaan toimittaja, kaksi koodari-graafikkoa, Arkku-työkalua kehittävä koodari, Latoja-työkalua kehittävä koodari, useampi infograafikko, kaksi animaattoria ja esimies. Jokaisella tiimillä on oma erilainen tapansa tarttua projekteihin. Ylen molemmissa tiimeissä projekti saa alkunsa ideasta, joka voi tulla omasta tiimistä, jostakin muusta tiimistä, Mediadeskin tapauksessa kokonaan toisesta toimituksesta tai koodaavalta toimittajalta itseltään. Jos idea tulee tiimin ulkopuolelta, se esitellään yleensä ensin tiimin tuottajalle, joka esittelee sen muulle tiimille. Tässä vaiheessa puntaroidaan, onko tiimillä projektille jotakin annettavaa. Jos projektiin päätetään tarttua, sen ympärille luodaan tiimi, jonka jokaiselle jäsenelle annetaan jokin vastuualue. Projektiin voi osallistua koko tiimi tai yksittäisiä tiimin jäseniä, mutta lähes aina siinä on mukana ainakin koodaava toimittaja, graafikko, tuottaja ja toimittaja, joka pitää projektin lankoja käsissään. Tiimien tuottajat pallotelevat ideoita omissa kokouksissaan, mutta yleensä ehdotus projektista tulee toimittajalta itseltään. Keskimäärin Ylen koodaavat toimittajat kertovat tekevänsä viikossa yhdestä kahteen juttua, vaikka projektien pituus saattaa vaihdella muutamasta tunnista useampaan viikkoon, jopa kuukauteen. ”Joskus on sellasia että hei meil ois tällöinen juttu me tarvittais tähän vaan järjestettävä taulukko. Ja sit se tehään ja se on joku kahden tunnin homma.” Helsingin Sanomien datadeskissä juttuaiheita etsitään itsenäisesti, ja aiheista kirjoitetaan juttuja joko itse tai kirjoittajaksi etsitään joku toinen toimittaja. Haastattelemani datajournalisti tekee myös erikoistaittoja Kuukausiliitteelle ja sunnuntaitoimitukselle, ja silloin datajournalistille tulevat työtehtävät päätetään esimiesten kesken kokouksissa. Datajournalisti miettii aiheita etsiessään, mikä voisi olla kiinnostavaa ja mistä tiedon voisi löytää. Kuten Usherin (2016) tutkimissa toimituksissa, Yle ja Helsingin Sanomat eroavat toisistaan siinä, tekevätkö ne nopeaa vai hidasta journalismia. Plus-deski ja Mediadeski eivät yleensä osallistu päivänpäällisten uutisten tekemiseen, kun taas Helsingin Sanomien Arkku- ja Latoja-työkalut mahdollistavat nopeammankin reagoimisen.

Suomalaiset koodaustaitoiset toimittajat eroavat hieman taidoiltaan ulkomaisista. Haastateltavani osaavat kuitenkin kaikki perinteisiä front-end-kieliä, HTML:ää, CSS:ää ja JavaScriptia. Eroja haastateltavien välille syntyi siinä, mitä JavaScriptin kirjastoja he kertoivat hallitsevansa. Kaksi

mainitsi käyttävänsä Reactia, yksi JQuerya, yksi D3:a ja yksi Vanilla JavaScriptia. On kuitenkin mahdollista, että he osaavat monia muitakin, sillä haastatteluissa osa puhui yleisesti JavaScriptin kirjastoista tai kehyksistä (engl. framework). Royalin (2010) mukaan koodaustaitoiset toimittajat käyttävät usein joko Djangoa tai Ruby on Railsia. Django on Pythonin yksi suosituimmista ohjelmistokehyksistä, mutta silti kukaan haastateltavistani ei maininnut erikseen Djangoa. Haastateltavistani viisi sanoi kuitenkin osaavansa edes jossakin määrin Pythonia. Yksi sanoi käyttävänsä Pythonin avoimen lähdekoodin jakelua nimeltä Anaconda, jolla voidaan tehdä esimerkiksi koneoppivia sovelluksia. Yksi haastateltavista sanoi käyttävänsä Pythonin sijaan R-ohjelmointikieltä ja toinen kertoi osaavansa paremmin Rubya. Tavallisesti koodaustaitoinen toimittaja osaa front-end-kielten lisäksi ainakin yhtä analyysikieltä eli Pythonia, R:ää tai Rubya. Yleisin ohjelma, mitä haastateltavani sanoivat osaavansa käyttää, on Excel. Haastateltavista neljä mainitsi Excelin erikseen, vaikka useampi heistä käyttää sitä silti työssään. Useat laskivat taidokseen rakentaa verkon harvointityökaluja eli ”screippereitä”. Hannafordin (2015) tutkimuksessa mainitut paikkatieto-ohjelmistot laski omiin taitoihinsa yksi haastateltavista. Ohjelmistoalalta journalismin pariin siirtynyt haastateltava kertoi lisäksi hallitsevansa useita back-end-kieliä ja -ohjelmistoja. Tällaisia olivat muun muassa Java, Apache Tomcat, CentOS ja JPA. Vähemmän tiettyihin ohjelmistoihin sidotuista taidoista haastateltavat mainitsivat osaamiseen julkaisujärjestelmät, pilvipalvelut, selaimen kehitystyökalut, matemaattiset taidot, koneoppimisen ja versionhallinnan. Erikoista kyllä, kukaan haastateltavista ei maininnut erikseen tietokantoja, vaikka niitä tarvitaan monissa koodaustaitoisten toimittajien tekemissä sovelluksissa. Lisäksi vain yksi mainitsi PHP:n, joka on Parasien ja Dagiralin (2012, 861) mukaan ollut etenkin aiempien CAR-toimittajien keskuudessa suosittu työväline.

Haastateltavani kertoivat työnsä olevan luonteeltaan uusien asioiden oppimista. Uteliaisuus ja avoimuus ovat koodaustaitoisen toimittajan tärkeimpiä ominaisuuksia. Koodaustaitoisen toimittajan pitää osata sisäistää uusia asioita nopeasti, mutta samalla olla valmis päivittämään vanhoja tietojaan. Uusia vinkkejä etsitään esimerkiksi yhdysvaltalaisista julkaisuista. Yksi haastateltavistani mainitsi tilastotieteilijä Nate Silverin perustaman FiveThirtyEight-sivuston, jolla julkaistaan muun muassa politiikkaan, talouteen ja urheiluun keskittyviä artikkeleita. Yhteistä niille on se, että ne perustuvat suuriin tilastoaineistoihin. Toinen puolestaan kertoi New York Timesin vuonna 2012 julkaiseman *Snow Fallin* olleen yksi jutuista, joka sai hänet alun perin kiinnostumaan nykyisestä työstään. Koodaustaitoiset toimittajat saattavat ottaa mallia muista julkaisuista hyvinkin suoraan: ”No joo siis tota varast– eikun kopioidaan tota, varsinkin jos on joku semmonen niinkun uus tapa visualisoida ja tämmönen.” Haastateltavieni mukaan työ on kiinnostavaa, jos pitää asioiden penkomisesta.

Koodaustaitoisella toimittajalla pitää olla ongelmanratkaisukykyä, koska ohjelmointi on harvoin suoraviivaista työtä. Työssä tarvitaan kärsivällisyyttä, koska ohjelmointivirheitä ei yleensä ratkota nopeasti. Haastateltavien mukaan sosiaalinen kyvykkyys on eduksi. Sekä Ylellä että Helsingin Sanomissa koodaustaitoinen toimittaja tekee yhteistyötä erilaisten ihmisten kanssa, myös sellaisten, joilla ei ole välttämättä lainkaan tietoa ohjelmoinnista. Lisäksi koodaustaitoisella toimittajalla olisi hyvä olla innostusta yhteiskunnallisia asioita kohtaan. Kuten Lehtniemi (2016a), haastateltavani olivat sitä mieltä, että halulla oppia pääsee jo pitkälle.

4.1.3 Koodaustaitojen määritelmä

Ei ole yksiselitteistä tapaa määritellä sitä, mikä lasketaan koodaamiseksi. Haastateltavistani suurin osa on sitä mieltä, että Excelin tai muun vastaavan taulukkolaskentaohjelman käytön voisi laskea koodaamiseksi, jos osaa käyttää esimerkiksi sen funktioita. Plus-deskin koodaavan toimittajan mielestä Excelin käyttö on koodaamista, jos sillä tekee jotakin monimutkaisempaa kuin tallentaa tietoa ja laskee summan. Excel on helppo tapa tutustua koodaamiseen. Helsingin Sanomien datatuottaja kuvailee Exceliä ”kevyeksi huumeeksi”, jos koodaaminen on ”kunnon huume”. Monet haastateltavistani määrittelevät ohjelmointitaidot sen perusteella, mitä he itse tekevät työssään. Ohjelmointi on heidän mielestään SQL-kyselyiden, laskureiden ja interaktiivisten verkkosovellusten tekemistä. Jos osaa rakentaa verkkosivujen haravointityökalun eli ”screipperin” tai JavaScript-ohjelman, voi sanoa osaavansa koodata. Osalle riittää pienempikin osaaminen. Heidän mielestään koodaustaidoiksi voidaan laskea myös se, jos ymmärtää algoritmien perusteet tai osaa ajatella tarpeeksi loogisella tavalla. Kaikki eivät ole sitä mieltä ollenkaan, että tavalliset toimittajat osaisivat koodata. ”Jos koodaaminen määritellään siksi, mitä mä teen työssäni, niin kukaan toimittajista ei osaa koodata.”

Haastateltavat kuvailevat ohjelmointia hyödylliseksi taidoksi, joka palkitsee nopeasti. Suurin osa heistä ei kuitenkaan kutsuisi ohjelmointia kansalaistaidoksi. ”Puhutaan paljon, että alakoulussa ruvetaan koodaamaan tai on ruvettukin, niin eihän se oikeesti mitään koodaamista ole vaan logiikan opettelua.” Yhden haastateltavan mielestä kansalaistaito on sellainen, jonka voi oppia järkevin ponnisteluin hyötyyn ja välttämättömyyteen verrattuna. Sillä perusteella hän voisi laskea Excelin käytön kansalaistaidoksi. Toinen haastateltava kertoo kansalaistaidon käsitteen olevan sidoksissa aikaan. Työelämä muuttuu teknologisemmaksi, joten tulevaisuudessa yhä useamman ihmisen pitää tietää, miten teknologia toimii. ”Jos vaikka otettais jostain kahdeksankytluvulta työntekijöitä ja

laitettais ne tähän nykyajan työympäristöön niin ne olis varmasti ihan sekasin.” Työelämän muuttumisesta kertoo myös tämä haastateltavan esimerkki: ”Kun puhuu jonkun oman äidin kanssa siitä että mitä tekee töissä ja yrittää selittää sille että tällasia juttuja niin ei sillä oo siihen mitään kosketuspintaa.”

Haastateltavat jakavat käsityksen siitä, ettei tavallinen toimittaja oikeastaan tiedä, mitä koodaaminen on. Koodaustaitoiset toimittajat uskovat, että koodaaminen nähdään mustana laatikkona tai salamyhkäisenä verhona. Heidän mielestään tavalliset toimittajat eivät tiedä, mitä koodaamisella voidaan saavuttaa. Mediadeskin koodaava toimittaja on ollut tilanteissa, joissa toimittaja on ajatellut hänen osaavan kaikkea. ”Se on ehkä semmoinen käsitys, mikä monella on. En koe, että olisin mikään ässäkoodari kuitenkaan.” Toimittajien voisi kuitenkin nähdä suhtautuvan ohjelmointiin hyvillä mielin. Haastateltavien puheista voisi päätellä, että ohjelmointiin suhtaudutaan Powersin (2012) kuvaamien ensimmäisen ja kolmannen tavan mukaisesti: Ohjelmointi nähdään uutena keinona saavuttaa journalismin perinteiset tavoitteet (Exemplar of Continuity) ja mahdollisuutena luoda uudenlaisia journalismin muotoja (Basis for Journalistic Reinvention). Haastateltavat kertovat koodaustaitoisten toimittajien puhuvan muita toimittajia enemmän journalismin uudistamisesta ohjelmoinnin keinoin, mutta tiimien ammattitaidon kysyntä on merkki siitä, että myös muut toimittajat näkevät ohjelmoinnin mahdollisuudet. Haastateltavien puheista ei paljastunut, että muut toimittajat kokisivat ohjelmoinnin yleistymisen uhkana ammatillisille käytännöille (Threat to Be Subordinated).

4.1.4 Haasteet

Suurena haasteena koodaustaitoiset toimittajat pitävät viestintää tavallisten toimittajien kanssa. Suurin ongelma ei kuitenkaan ole ohjelmointiin liittyvä käsitteistö, vaan hankalinta on kertoa koodaustaidottomille toimittajille, mikä on mahdollista toteuttaa ja mikä ei. Toimittaja haluaisi monesti, että hänen juttunsa olisi jotakin ennennäkemätöntä. Se ei ole aina mahdollista, sillä koodaustaitoisten toimittajien pitää pohtia, miten esitystapa tukee jutun sisältöä. Juttu ei saa olla niin erikoinen, että se säikäyttää ihmiset pois.

”Hankalinta on tehdä toimittajalle selväksi, miksi sinne ei kannata laittaa niitä vilkkuvia glitter-pääkalloja pyörimään jonkun 90-luvun nettisivun tyyliin.”

Epäselvyyksiä syntyy sen vuoksi, ettei toimittaja tiedä tavallisia tilastollisen matematiikan käsitteitä. Koodaustaitoisten ja tavallisten toimittajien yhteisissä projekteissa menee aikaa siihen, että toimittajalle selitetään, mikä on mediaani ja miten se lasketaan Excelillä. ”Toimittajilla pitäisi olla datankäsittelytaitoja, koska niin monet uutisjutut perustuvat johonkin dataan.” Suurinta osaa haastateltavista ei kuitenkaan haittaa, että toimittajille joudutaan selittämään asioita. He pitävät sitä kiinteänä osana projektia.

Kuten Lewisin ja Usherin tutkimuksessa (2014), haastateltavani nostivat esiin koodaustaitoisten ja koodaustaidottomien toimittajien erilaiset aikakäsitykset. Haastateltavieni mukaan monelle toimittajalle on tullut yllätyksenä, kuinka kauan projektit koodaustaitoisten toimittajien kanssa ovat kestäneet. Joissain tapauksissa toimittajat ovat ihmetelleet, kuinka nopeasti jotkut asiat ovat onnistuneet. Erilaiset käsitykset johtuvat siitä, ettei toimittaja välttämättä tiedä, millaista työtä koodaustaitoiset toimittajat tekevät. Toimittajilla voi olla myös vahvoja ennako-oletuksia siitä, miten jotkut asiat toimivat. Väärinkäsityksiä on ilmennyt sitä vähemmän, mitä enemmän kokemusta koodaustaitoisella toimittajalla on toimittajan töistä. Ilmiö on siis sama kuin Boczkowskin (2005) tutkimuksessa. Tehtävän vaatimuksista keskusteleminen on helpompaa, jos ammattiryhmillä on yhteinen kieli. ”Toimittajat, joiden kanssa mä tein yhteistyötä, oli mun vanhoja tuttuja talon sisältä. Kommunikaatiota helpotti tosi paljon se, että mä tiesin, miten toimittajat ajattelee, mikä niitä kiinnostaa ja mikä niiden aikataulu on.”

Selvä haaste koodaustaitoisille tiimeille on niiden lähestyttävyyys, vaikka tiimit on perustettu sillä oletuksella, että toimittajien olisi helppo lähestyä niitä. Esimerkiksi Ylen Plus-deski vieraillee muiden tiimien aamupalaverissa, jotta koodaustaitoiset toimittajat kävisivät muille tutuiksi, mutta siitä huolimatta yhteistyötä heidän kanssaan tekevät eniten ne, jotka ovat tehneet yhteistyötä aiemminkin. Koodaustaitoiset toimittajat ovat korostaneet, että toimittajat saavat esittää heille keskeneräisiäkin ideoita. Ideoita saa myös esittää mitä kautta tahansa, itsenäisesti tai oman tuottajansa välityksellä. Helsingissä Plus-deski on fyysisesti osa muuta toimitusta. Sen sijaan Ylen Mediadeskissä haaste on etäisyys toimittajiin. Mediadeski tekee projekteja Ylen aluetoimitusten kanssa, ja nämä aluetoimitukset sijaitsevat ympäri Suomea. Ideoita joudutaan pallottelemaan yleensä puhelimen tai pikaviestisovellusten välityksellä, mikä voi olla toisinaan aiheuttaa väärinkäsityksiä: ”Siinä voi tulla sellaisia rikkinäisen puhelimen efektejä sitten.”

Koodaustaitoiset toimittajat ovat tällä hetkellä keskittyneet tiimeihin, joita on vain harvoissa suomalaisissa toimituksissa. Tiimi on oma kokonaisuutensa, joka toimii itsenäisesti ja tekee kaikkien

muiden tiimien kanssa yhteistyötä. Jotta koodaustaitoisten apua osattaisiin pyytää tulevaisuudessa matalammalla kynnyksellä, haastateltavat ehdottavat koodaustaitoisten toimittajien jalkauttamista toimituksiin. Sen sijaan, että koodaustaitoiset toimittajat muodostavat oman tiiminsä, jokaisessa toimituksen tiimissä voisi olla yksi koodaustaitoinen toimittaja, joka keskittyisi vain sen tiimin juttuihin. Toimittaja olisi osa ryhmää, ja muut toimittajat uskaltaisivat lähestyä häntä helpommin. Kaikkia haastateltavia ajatus ei miellytä. Olemassa olevat tiimit, Ylen Plus-deski ja Mediadeski sekä Helsingin Sanomien datadeski, keräävät jutuillaan huomattavia määriä lukijoita. ”Joka vuosi luetuimmat jutut ovat tulleet meiltä, joten vaikea semmosta on tietenkin hajottaa.” Lisäksi tiimit keräävät jutuillaan palkintoja, kuten Koura-palkintoja.

Haastateltavani näkevät työssään aiempaa kirjallisuutta enemmän sisällöllisiä haasteita. Yksi sellainen on niin sanottujen erikoistaittojen lisääntyminen. Erikoistaitot ovat julkaisun muuhun sisältöön verrattuna suurieleisempiä ja viestivät lukijalle, että jutun tekemiseen on käytetty enemmän aikaa kuin tavallisiin uutisjuttuihin. Erikoistaitot korvaavat yleensä koko verkkosivuston tavallisen taiton, eli ne ovat yleensä käyttäjän koko ruudun kokoisia. Kuvat ovat suuria, ja harvinaisempia elementtejä, kuten videota ja ääntä, on käytetty tavallista enemmän. Koodaustaitoiset toimittajat osaavat tehdä erikoistaittoja vaivattomasti. Lisäksi Helsingin Sanomien Arkku-työkalulla jokainen toimittaja voi lisätä itse juttuunsa erikoistaiton. Haasteen erikoistaitoista tekee se, että niitä aletaan käyttää liiaksi. Jokainen toimittaja uskoo oman juttunsa olevan erityislaatuinen ja ansaitsevan muista uutisjutuista poikkeavan tyylin. Vaarana on, että erikoistaitoista tulee arkipäivää. Erikoistaitto ei kerro jutun erityislaatuudesta, jos kaikissa jutuissa on erikoistaitto. Sen vuoksi esimerkiksi Helsingin Sanomissa on alettu kiinnittää huomiota siihen, etteivät erikoistaitot koe inflaatiota. Erikoisen tilanteesta tekee se, että vaikka kaikista jutuista ei haluta ”päryttäviä”, haluavat koodaustaitoiset toimittajat tehdä nimenomaan ”päryttävää” sisältöä. Useampi haastatteleman koodaustaitoinen toimittaja sanoi tämän tavoitteekseen. Tämä ei kuitenkaan tarkoita sitä, että koodaustaitoiset ja koodaustaidottomat toimittajat haluaisivat samoja asioita. Koodaustaitoiset toimittajat haluavat tehdä vaikuttavia sisältöjä, mutta sen lisäksi tarvitaan sisällöllisiä perusteluita. Pelkkä erikoistaitto ei myöskään yleensä riitä, vaan erikoistaitto pitää tehdä tavalla, jolla sitä ei ole vielä aikaisemmin tehty. Koodaustaitoiset toimittajat etsivät jatkuvasti uusia tapoja tehdä journalismia.

Jatkuva haaste on teknologian kehityksen mukana pysyminen. Uusien innovaatioiden hyödyntäminen journalismissa edellyttää tietämystä teknologian uusista trendeistä. Koodaustaitoisten toimittajien mainitsemat kehityskohteet liittyvät heidän omiin taitoihinsa. Tavallisille toimittajille he haluaisivat

järjestettävään koulutukseen, jotta toimittajat oppisivat, mitä ohjelmointi mahdollistaa ja miten sitä voi omassa työssään hyödyntää. Koulutuksen jälkeen toimittajien tulisi kuitenkin pystyä soveltamaan taitoja työssään, jotta taidot eivät unohdu heti koulutuksen jälkeen. Suuri harppaus olisi jo se, jos kaikki toimittajat suhtautuisivat ohjelmointiin myönteisesti ja innokkaasti. Yksi haastateltavista ehdottaa hieman samantapaista törmäyttämistä kuin Lewisin ja Usherin (2013) tutkimuksessa, jotta ammattiryhmät löytäisivät helpommin yhteisen kielen. Sen sijaan, että toimituksessa olisi ”musta huone, jossa istuu jotain graafikoita”, ihmiset voitaisiin tuoda lähemmäksi toimintaa ja toisiaan. Törmäyttäminen saisi haastateltavan mukaan alkaa jo alojen oppilaitoksissa.

4.2 Ohjelmointi osana journalismia

4.2.1 Merkitys journalismille

Koodaustaitoisten toimittajien merkitys konkretisoituu journalistisissa sisällöissä. Haastateltavieni mielestä he ovat onnistuneet nostamaan verkkojournalismin tekemisen kokonaan uudelle tasolle. Journalismissa on ryhdytty käyttämään huomattavasti enemmän toiminnallisuutta ja asioita havainnollistavia elementtejä. Haastateltavani uskovat, että heidän työnsä on synnyttänyt paljon hyvää ja luettua journalismia. Helsingin Sanomien datatuottajan mielestä koodaustaitoiset toimittajat ovat tehneet journalismia, joka on lukijalle relevanttia. Jutun perustana olevaa dataa voidaan käsitellä siten, että lukija löytää sieltä itselleen tärkeitä asioita. Helsingin Sanomien datajournalisti kuvailee omaa tiimiään prototyypiksi, jonka tehtävänä on kokeilla uusia asioita. Hän uskoo, että uudenlaista tarinankerrontaa tullaan kokeilemaan tulevaisuudessa muillakin osastoilla. Plus-deskin koodaava toimittaja kertoo, että mahdollisuus kokeilla asioita on vaikuttanut journalismiin myönteisesti. Hyväksi todetuista asioista on tehty toimitukselle arkea ja ”tarpeettomasta ja typerästä” on opittu myöhempiä projekteja varten. Muut toimittajat ovat hyötäneet koodaustaitoisista toimittajista muun muassa koulutusten kautta. Mediadeskin koodaava toimittaja kertoo tiimin merkityksen olevan valtava niille aluetoimituksille, joissa ei ole vastaavanlaista osaamista. Mediadeski on voinut tarjota aluetoimitusten toimittajille esimerkiksi kuvitusapua tai apua jutun rakenteen kanssa.

Haastateltavat kokevat, että heidän työnsä on vähentänyt toistettavan työn määrää, kuten Flewn, Spurgeonin, Danielin ja Swiftin (2012) tutkimuksessa. Yksi haastateltavista koki tärkeäksi, että toimittajat pääsevät koodaustaitoisten toimittajien avulla teknologian eturintamaan. Usherin (2016)

mielestä journalismi on omaksunut teknologioita muita aloja hitaammin. Yksi haastateltavista perusteli tiiminsä merkitystä palkinnoilla, joita koodaustaitoisten toimittajien jutut ovat voittaneet. Kuten ulkomailla (Lewis & Usher 2014), suomalaisten koodaustaitoisten toimittajien jutut ovat keränneet laadukkaista ja innovatiivisista jutuistaan palkintoja ja kunniamainintoja. Palkintoja on myöntänyt ainakin koulutusrahasto KOURA.

4.2.2 Journalistisen työn muutos

Koodaustaitoiset toimittajat on otettu toimituksissa hyvin vastaan. Haastateltavieni mukaan heiltä kysytään paljon apua ja heille ehdotetaan paljon ideoita. Ideoita saattaa tulla spontaanisti vaikka kahvipöytäkeskusteluissa. Haastateltavien mukaan se, että paikalla on ihmisiä, jotka osaavat tehdä innovatiivista sisältöä, ruokkii toimittajien mielikuvitusta. Joskus mielikuvitus laukkaa turhankin nopeasti, sillä joskus toimittajat luulevat, että jos joku osaa koodata niin silloin hän osaa mitä tahansa. Näin ei kuitenkaan välttämättä ole: ”En koe, että olisin mikään ässäkoodari kuitenkaan.” Koodaustaitoiset toimittajat eivät ole huomanneet, että ohjelmoinnista puhuttaisiin toimittajien keskuudessa. Sen sijaan ohjelmointiosaamista tulee vastaan yllättävissä tilanteissa. Sekä Ylellä että Helsingin Sanomissa on toimittajia, jotka ovat koodaustaitoisten toimittajien yllätykseksi sanoneet osaavansa koodata. Se on koodaustaitoisten toimittajien mielestä myönteinen asia. Se kertoo haastateltavieni mukaan siitä, että tietoisuus ohjelmoinnista on kasvanut toimituksissa. Mitä enemmän toimittajat tietävät koodaamisesta, sitä enemmän koodaustaitoisille toimittajille ehdotetaan yhteistyötä. ”Massahysteriasta” ei kuitenkaan voida puhua. Ohjelmointi herättää innostusta koodaustaitoisten toimittajien omassa kuplassa, mutta sen ulkopuolella sitä ei ole huomattu juuri olevan.

Koodaustaitoiset toimittajat tietävät, että heitä on pidetty joskus toimittajia alempiarvoisina (Powers 2012). Tällä hetkellä näin ei kuitenkaan ole. ”Kukaan ei ole sanonut, että olen koodarina toimittajan alapuolella”, sanoo Plus-deskin koodaava toimittaja. Mediadeskin koodaavan toimittajan mukaan koodaustaitoinen toimittaja olisi aikaisemmin saattanut kohdata vähättelyä, mutta nyt ihmiset jo ymmärtävät, kuinka tärkeää verkon tekeminen on. Muutos voi johtua myös eri maiden toimitusten kulttuurieroista. Mielestäni pitää muistaa, että Powersin tutkimus sijoittuu Yhdysvaltoihin, jossa työpaikkojen hierarkia on usein korkeampi kuin Suomessa. Haastateltavien mukaan työelämä tulee olemaan tulevaisuudessa teknologisempi. Koodaustaitoisten toimittajien painoarvo on suuri etenkin ideointivaiheessa. Helsingin Sanomien datatuottajan mukaan ohjelmoinnilla on hyödyllisenä taitona

hyvä maine. Osa toimittajista ihailee ohjelmointitaitoja, koska ne ovat mystisiä ja outoja. ”Tää on hyvä tapa saada ihmiset kutsumaan sua velhoksi.”

Sekä Ylen että Helsingin Sanomien työorganisaatiot ovat muuttuneet, kun koodaustaitoiset tiimit perustettiin joitakin vuosia sitten. Plus-deskissä työskentelevän koodaavan toimittajan mielestä Ylen työtavoissa on tapahtunut kehitystä. Hänestä on kuitenkin vaikea arvioida, millä tavalla Plus-deski on vaikuttanut kehitykseen. Plus-deskin olemassaolon aikana Ylessä on viety läpi useita organisaatiomuutoksia. Plus-deski on kuitenkin pysynyt rakenteeltaan ja tehtäviltään samanlaisena. ”Me ollaan oltu aika solidi kivi, kun muu organisaatio on muuttunut ympärillä useampaan kertaan.” Syynä koodaava toimittaja pitää tiimin hyvää menestystä. Organisaation muutosten vuoksi perinteisen toimittajan työ on muuttunut paljon enemmän kuin koodaavan toimittajan työ, joka on pysynyt vuosien ajan hyvin samanlaisena.

Haastateltavani ovat Lehtniemen (2016) kanssa samaa mieltä siitä, ettei tavallisen toimittajan tarvitse opetella tulevaisuudessa koodaamaan. Enemmän hyötyä toimittajalle on siitä, jos hän tietää, mitä ohjelmoinnilla voidaan tehdä. Haittaa ohjelmointitaidoista ei kuitenkaan ole. Plus-deskin koodaava toimittaja kuvailee ohjelmointitaitoja lihaksiksi, joilla on helpompi löytää töitä. Samanlaisia toimittajan työtä tukevia lihaksia ovat esimerkiksi valokuvaustaidot ja videoeditointitaidot. Helsingin Sanomien datajournalistin mielestä ohjelmointitaidoista tulee itsestäänselvempi osa toimittajan työtä. Hänen mielestään kaikkien ei tarvitse käyttää ohjelmointia työssään, mutta ohjelmoinnista tulee taito, jota ei pidetä samalla tavalla erityistaitona kuin nykyään. Helsingin Sanomien datatuottaja sanoo ohjelmointitaitojen olevan harvinaisempia, kuin mitä julkisessa keskustelussa on annettu ymmärtää. Hän ei ymmärrä sitä, että ohjelmointitaitoja on yritetty lyödä läpi kaikille myös muualla yritysmaailmassa. Hänen mielestään ohjelmointitaidot ovat erityistaitoja, joita jotkut opettavat erikoistuakseen. Ohjelmointi on vähän kuin matematiikka: jotkut ovat siinä parempia kuin toiset, mutta silti kaikkien pitää tietää, mitä sillä voidaan tehdä. Kaikista ei myöskään tule maantieteen opettajia, vaikka lukevat maantiedettä kouluissa. Sen vuoksi koodaustaitoiset toimittajat eivät koe sitä uhaksi, että ohjelmointia on alettu opetella peruskouluissa. Päinvastoin, ohjelmointiopetus nähdään hyvänä asiana: ”Ei siitä ole mun mielestä haittaakaan. En mä koe sitä silleen, että te viette mun työt, jos kaikki opettelee koodaamaan.”

Toimittajille, jotka osaavat koodata, on haastateltavien mukaan töitä jatkossakin. Koodaustaitoiset tiimit ovat niin työllistettyjä, ettei kysyntä lakkaa ainakaan lähiaikoina. Haastateltavat uskovat, että heidän tiimensä koko tulee kasvamaan, ellei katsota, että tiimeillä olisi hyvä olla jokin raja, kuinka

paljon ihmisiä yhdessä tiimissä kannattaa olla. Suurin osa koodaustaitoisista toimittajista on kiinnostunut tekemään työtänsä jatkossakin. He kokevat työn olevan merkityksellistä ja palkitsevaa. Teknisen alan koulutuksen saaneiden mielestä koodaustaitoisen toimittajan työ on sopivan lyhytjänteistä ohjelmistoalan töihin verrattuna. Useamman vuoden ajan samaa työtä tehneen mielestä pitää kuitenkin muistaa, että työn pitää olla myös kehittävä. Hänen mielestään myös koodaustaitoisen toimittajan työhön voi leipiintyä. ”Vaikee sanoa, onko tää semmonen eläkehomma kuitenkin.” Koodaustaitoista toimittajaa saattaa alkaa houkutella ura ohjelmistopuolella. Ohjelmointiosaaminen avaa ovia, joiden takana on kokonaan uudenlaisia haasteita ja kenties parempaa palkkaa. Ohjelmistoalalta tulleelle koodaustaitoisen toimittajan työt saattavat myös vaikuttaa liian yksinkertaisilta:

”Ite tykkään tosta yhteiskunnallisesta puolesta ja noista, että pääsee monenlaisiin monenlaisiin projekteihin, mutta ne eivät ole vielä teknisesti ollu tarpeeksi haastavia itelle. Mä oon tosta ton meidän tuottajan kanssa puhunutkin, että vähän ehkä niinkun haluaisin semmosta niinkun monimutkasempaa tai semmosta, että kun sen erikoistaiton on tehny kerran niin se on, kun en ole mitenkään visuaalisesti vahva.”

4.2.3 Journalismin muutos

Ohjelmointi on jo joissakin toimituksissa osa journalismia. Ainakin Ylessä ja Helsingin Sanomissa ohjelmointi on jo olennainen osa toimitustyötä. Helsingin Sanomien datajournalistin mukaan monet jutut olisivat jääneet syntymättä, jos niissä ei olisi käytetty apuna ohjelmointia. Ilman koodaustaitoja toimittajat eivät olisi esimerkiksi saaneet niitä tietoja, joihin jutut ovat perustuneet. Haastateltavien mukaan etenkin laskurit ja järjestettävät taulukot ovat arkipäivää. Kaikissa toimituksissa näin ei silti ole, sillä esimerkiksi paikallislehdissä ohjelmointiosaaminen on hyvin pientä. Onko ohjelmointi osa journalismia, riippuu haastateltavien mukaan toimituksesta ja käytössä olevista resursseista. Osaamista on enemmän, jos sitä on varaa hankkia. Ylen koodaavan toimittajan mukaan taitojen puuttuminen näkyy jo verkkosivujen käytävyydessä. Se ei välttämättä vaikuta paikallislehtien juttujen sisältöön vaan siihen, mitä sivuston taustalla tapahtuu.

Koodaustaitoiset toimittajat tekevät journalistisista jutuista visuaalisempia. Heidän työnsä kautta visuaalisten elementtien määrä jutuissa tulee tulevaisuudessa kasvamaan. Koodaustaitoisten toimittajien tehtävänä on tuottaa ”vau-efektejä”, joiden vuoksi lukija jäisi sivulle pidemmäksi aikaa

ihmettelemään ja ihastelemaan. Visuaalisuudella tavoitellaan myös sitä, että monimutkaiset asiat aukeaisivat lukijoille helpommin. Koodaustaitoiset toimittajat voivat esimerkiksi havainnollistaa jotakin ilmiötä selkeämmin grafiikalla tai interaktiivisella elementillä. Haastateltavat näkevät kehitystä siinä, mikä aiheuttaa ”vau-efektejä”. Plus-deskin koodaavan toimittajan mielestä ”yksinkertaiset testit ja laskurit olivat ehkä vuoteen 2013 asti vau”. Nykyään häntä säväyttävät tekoälyt ja New York Timesissa näkemänsä virtauskaaviot. Visuaalisuus tai ohjelmointi ei ole haastateltavien mukaan silti itseisarvo. Jutuista tehdään visuaalisempia, koska se tukee journalistisia tavoitteita. Visuaalisia elementtejä tehdään, koska niiden kautta lukija voi sisäistää tietoja helpommin. Lukija voi myös itse löytää niistä kiinnostavia tietoja. Toisen Plus-deskissä työskentelevän koodaavan toimittajan mielestä elementeillä etsitään uusia tarinankerronnan tapoja. Paperisen sanomalehden sanat siirrettiin pitkään sellaisinaan verkkoon. Nyt etsitään verkolle parasta tapaa kertoa tarinoita ja yritetään hyödyntää olemassa olevia tekniikoita mahdollisimman hyvin. Uusien tapojen etsimisessä on otettava huomioon päätelaitteiden erilaisuus. Nykyään journalismia voi kuluttaa kaikenkokoisilla näytöillä. Utisia voi lukea puhelimella, tabletilla, läppärillä tai pöytätietokoneella. Osa näytöistä on kosketusnäyttöjä. Lisäksi koodaustaitoisten toimittajien tekemät elementit voivat näyttää erilaisilta riippuen siitä, onko lukijalla selaimenaan Googlen Chrome, Mozillan Firefox vai Microsoftin Edge. Tulevaisuudessa journalismissa pitää siis ottaa huomioon se, että juttu toimii millä laitteella tahansa.

Ohjelmointi poistaa toimittajilta haastateltavien mukaan ”bulkkityötä”. Toimittajien ei siis tarvitse tehdä tulevaisuudessa yhtä paljon toistettavaa työtä. Ohjelmoinnilla voidaan automatisoida tiettyjä yksinkertaisia ja tylsiä asioita, jolloin toimittajille jää aikaa kehittää parempia juttuja. Esimerkiksi juuri Helsingin Sanomien Latoja-työkalulla on tehty juttuja muun muassa jääkiekon SM-liigan runkosarjan tuloksista, kuntavaalituloksista, metroliikenteen häiriöistä ja inflaatiosta. Ohjelmointia hyödyntämällä tuhansia dokumentteja ei tarvitse käydä itse läpi. Ohjelmointi myös tekee journalistisista jutuista personoituja. Jos jutun perustana oleva tietoaaineisto sallii, koodaustaitoiset toimittajat voivat lisätä juttuun ominaisuuden, jolla käyttäjä voi itse valita, mitä tietoja hän haluaa nähdä. Käyttäjä voi esimerkiksi tarkastella jutussa tietoja, jotka koskevat hänen omaa kotikuntaansa tai ikäluokkaansa. Toimittajan ei tarvitse itse nostaa aineistosta esimerkkejä, joiden kautta hän kertoisi ilmiöstä, vaan käyttäjät voivat valita jutusta ne osat, jotka kiinnostavat häntä. Käyttäjät voivat myös osallistua journalismiin entistä paremmin. Tietoa voidaan kerätä lukijoilta suoraan ja tehdä juttuja näiden tietojen perusteella. Kommenttien moderoinnissa voidaan käyttää tekoälyä, jolloin toimittajien ei tarvitse päivystää kommenttilaatikon ääressä karsiakseen asiattomuuksia.

Koodaustaitoiset toimittajat pohtivat jatkuvasti, minkä vuoksi jokin juttu tehdään. Tavalliset toimittajat haluvat usein koko sivun erikoistaittoja, koska ne ovat heidän mielestään hienoja. Koodaustaitoiset toimittajat pyrkivät tekemään jutusta sellaisen, että se palvelee mahdollisimman hyvin journalismia. Toistuva pohdinnan aihe on se, tehdäänkö juttu data vai tarina edellä. Jos juttu tehdään data edellä, toimittajien hallussa oleva tietoaaineisto on niin poikkeuksellinen, että juttu kannattaa tehdä jo pelkästään sen vuoksi. Tarinavetoisessa jutussa tietoaaineisto tukee tarinaa, jonka pääosassa on ihminen. Haastateltavat näkevät kehitystä siinä, mikä edellä journalistisia juttuja on tehty viime vuosina. Aiemmin puhtaasti dataan perustuvia juttuja tehtiin enemmän. Nyt esimerkiksi Mediadeskissä työskentelevän koodaavan toimittajan mielestä pelkästään dataan perustuva uutinen vaikuttaa sielultaan kylmältä. Juttuihin haetaan nykyään koskettavaa ihmisläheisyyttä. Helsingin Sanomien datajournalistin mukaan paras tapa on yhdistää hyvä data-aineisto ja hieno ihmistarina. ”Numerot edellä tehdyt jutut eivät ole lukijalle täydellinen elämys. Sen näkee kylmästi analytiikasta, etteivät ne ole kovin luettuja.”

On kiinnostavaa, miten ohjelmoinnin yleistymisen vaikuttaa journalismin vakiintuneisiin periaatteisiin. Haastateltavieni mukaan ohjelmointi ei vaikuta journalismiin ainakaan haitallisesti. Plus-deskin koodaavan toimittajan mielestä ohjelmointi on yhtä haavoittuvainen kuin journalismi itsessään. Jos journalismissa pitää esimerkiksi pyrkiä objektiivisuuteen, se on mahdollista myös ohjelmoimalla. Suurin osa haastateltavistani on sitä mieltä, että ohjelmointi parantaa objektiivisuutta ja tekee jutuista läpinäkyvämpiä ja monipuolisempia. Lukija voi keskittyä tietoihin, jotka hän kokee tärkeiksi tai jotka koskettavat häntä itseään ilman, että toimittaja on tehnyt valintoja omasta näkökulmastaan. Jos jutussa on kone, josta voi katsoa oman kotikuntansa tilanteen, jokainen lukija saa omasta kotikunnastaan saman verran tietoa kuin kaikki muutkin sen sijaan, että toimittaja olisi nostanut jalustalle muutaman esimerkin. Lukija voi tarkastella jutun perustana olevaa aineistoa ja tehdä itse päätelmiä siitä, miten juttu on tehty ja mitä tiedot merkitsevät. Koska jutuissa on mahdollista esittää valtavia määriä tietoa, asioita voidaan esittää helpommin useista erilaisista näkökulmista.

Se, että ohjelmointiosaamista on nimenomaan toimituksissa, edistää myös puolueettomuutta. Ulkopuolelta tulevien vaikutusyritysten määrä minimoidaan, kun kaikki toimituksen käyttämät työkalut tehdään lähtökohtaisesti itse. Helsingin Sanomien datatuottaja kertoo, että omilla työkaluilla pystytään tuottamaan omaa tietoa ja analysoimaan sitä ilman, että esimerkiksi valtiot tai yritykset vaikuttaisivat siihen. Itse kehitetyistä työkaluista saadaan myös juuri sellaisia, kuin niiden halutaan

olevan. Esimerkiksi Helsingin Sanomien käyttämät Arkku- ja Latoja-työkalut on kehitetty nimenomaan toimituksen omia tarpeita ajatellen.

Ohjelmoimalla voi päästä käsiksi aineistoihin, joihin ei olisi välttämättä aikaisemmin olisi päässyt. Sen vuoksi koodaustaitoisten toimittajien pitää pohtia julkaisemisen etiikkaa. Toimittajat ovat aina miettineet, mitä saa ja mitä ei saa julkaista, mutta nyt pohdinnan paikkoja tulee vastaan useammin. Kaikki tieto, johon toimittaja pääsee kiinni, ei välttämättä ole julkista. Koodaustaitoisia toimittajia pohdituttaa myös se, miten ohjelmointivirheisiin suhtaudutaan journalismissa. Selkeää käytäntöä ei ole, miten korjataan esimerkiksi jutussa oleva ohjelmointivirhe. ”Jos feature-jutussa on bugi, niin pitäisikö sinne korjata, että ’poistettu ohjelmointivirhe’?” Haastateltavia mietityttää myös, miten suhtaudutaan tekoälyn tekemiin virheisiin. Mediadeskin koodaavaa toimittajaa huolettää viihdepainotteisten uutisten lisääntyminen. Ohjelmoinnin yleistyminen saattaa johtaa siihen, että yhä suurempi osa uutisista on viihdepainotteista. Koodaavan toimittajan mielestä journalismin ja viihteen raja tulee tulevaisuudessa hämärtymään.

5 POHDINTA

5.1 Mitä tuloksista voi päätellä?

Kaikki journalismi on nykyään tietokoneavusteista. Aiempien CAR-toimittajien työkaluista, kuten tietokannoista ja taulukkolaskentaohjelmista, on tullut tämän päivän koodaustaitoisille toimittajille peruskauraa. Teknologia ja ihmisten vaatimukset journalismille ovat kehittyneet niin paljon, että koodaustaitoiset toimittajat tarvitsevat nykyisin huomattavasti laajemman työkalupakin kuin tietokoneavusteisen journalismin ensimmäisen sukupolven tekijät. Tulokset kertovat, ettei haastattelemillani koodaustaitoisilla toimittajilla ole juurikaan kosketuspintaa vanhoihin CAR-perinteisiin. Yksi heistä osasi sijoittaa itsensä osaksi tätä kehitystä, mutta suurin osa haastateltavistani on niin nuoria tai ollut asemassaan niin vähän aikaa, ettei osannut mainita sanaakaan edellisten vuosikymmenten tapahtumista. Se on ymmärrettävää, sillä ala muuttuu hurjaa vauhtia. Siinä ajassa, kun koodaustaitoinen toimittaja ehtii opetella uuden järjestelmän tai ohjelmointikielen, on saatettu keksiä jo monta parempaa. Teknologialla on suuri rooli siinä, mitä journalismilta halutaan nykyään. Ennen uutta oli se, että massiivisia data-aineistoja pystyi käsittelemään ja julkaisemaan helposti. Nyt se ei enää riitä, vaan journalismista halutaan ”päräyttävämpää” ja juttujen halutaan herättävän ”vau-efektejä”. Koodaustaitoinen toimittaja tuottaa luontevasti karttoja, laskureita ja kuvaajia, kerää, käsittelee ja analysoi valtavia tietoaaineistoja, ohjelmoi, kuvittaa ja kirjoittaa journalistisia juttuja sekä neuvoo toimituksen muita jäseniä tietoteknisissä asioissa.

Tuloksista ilmenee, että suurimmalla osalla haastattelemistani koodaustaitoisista toimittajista on teknisen alan koulutus. Tämä saattaa johtua siitä, ettei Suomen kokoisessa maassa ole montaa toimittajaa, jolla olisi ohjelmointitaitoja. Esimerkiksi Yhdysvalloissa suurin osa koodaustaitoisista toimittajista oli alun perin toimittajia, mutta siellä valtavasta väestömäärästä on varmasti helpompaa löytää osaajia. Suomalaisissa mediataloissa työskentelee koodaustaitoisia toimittajia, joille itsensä kehittäminen on tärkeää. Suurin osa haastateltavistani on opetellut töissä tarvittavia taitoja omissa vapaa-ajan projekteissaan. Toimittajat ovat aloittaneet rakentamalla verkkosivuja ja sovelluksia, ja koodarit ovat kiinnostuneet yhteiskunnallisista asioista ja demokratian toteutumisesta. Heitä kaikkia yhdistävät uteliaisuus, ongelmanratkaisukyky ja kärsivällisyys. Haastateltavani kuvailevat työtään loogisten ongelmien kanssa painimiseksi ja jatkuvaksi uudistautumiseksi.

Pyysin haastateltaviani listaamaan teknisiä taitoja, joita heillä on ja joita he tarvitsevat työssään. Kaikki sanoivat osaavansa perinteisiä front-end-kieliä eli HTML:ää, CSS:ää ja JavaScriptia. Muilla taidoilla koodaustaitoiset toimittajat pyrkivät erottautumaan muista. Jotkut sanoivat käyttävänsä datan analyysiin Pythonia, jotkut Rubyä. Tiimeissä tehdään töitä omilla vahvuuksillaan, joten eri taitoja arvostetaan paljon. Tutkimukseni tuloksista käy ilmi, että suomalaisten koodaustaitoisten toimittajien taidot eroavat monilta osin ulkomaisten kollegoiden taidoista. Tämä johtuu ainakin erilaisista toimituskulttuureista. Kuten Ylessä ja Helsingin Sanomissa, toimitukset ovat itse päättäneet, millainen työnkuva koodaustaitoisilla toimittajilla on. Tekijät taas ovat päättäneet, mitä ohjelmia he käyttävät ja mistä taidoista on heille hyötyä. Eroja selittää myös teknologian kehitys. Ulkomaisten koodaustaitoisten toimittajien taitoja listaavasta tutkimuksesta on saattanut vierähtää useampi vuosi. Siinä ajassa heidän käyttämänsä työkalut ovat jo vanhentuneet ja tilalle on tuotu parempia vaihtoehtoja.

Haastatteleman koodaustaitoiset toimittajat pitävät ohjelmoinnin yleistymistä journalismissa hyvänä asiana. He näkevät työssään kuitenkin muutamia haasteita. Suurimmaksi sellaiseksi nousi viestintä koodaustaidottomien toimittajien kanssa. Suuri osa koodaustaitoisten toimittajien työstä tehdään yhteistyössä muiden toimittajien kanssa, eivätkä he välttämättä tiedä, mitä ohjelmoinnilla voidaan saavuttaa ja mitä ei. Toimittajat eivät myöskään aina ymmärrä, kuinka kauan jonkun ohjelmointia vaativan asian tekeminen kestää. Tässä pitää kuitenkin muistaa, etteivät haastateltavat pohtineet lainkaan sitä, onko heidän puhettaan heidän omasta mielestään helppo ymmärtää. On yhtä lailla mahdollista, että viestinnälliset haasteet johtuvat koodaustaidottomien toimittajien kyvyttömyydestä avata asioita kansantajuisesti. Toimittajilta pitäisi kysyä, käyttävätkö koodaustaitoiset toimittajat heidän mielestään liikaa teknistä ammattislangia. Haastateltavat kokivat haasteeksi myös tiimien lähestyttävyyden, vaikka tiimit on perustettu sillä oletuksella, että toimittajien olisi helppo lähestyä niitä. Ratkaisuksi ehdotettiin koulutusten lisäämistä ja koodaustaitoisten toimittajien jalkauttamista eri toimituksiin. Sisällöllisistä haasteista havaitsin ainakin yhden ristiriidan: Koodaustaitoiset toimittajat haluavat tehdä ”päryttäviä” juttuja, mutta pelkäävät, että niitä tehdään liikaa. Toimittajat vaativat usein erikoistaitoja, jotka näyttävät hyviltä, mutta joiden tekeminen on koodaustaitoisesta toimittajasta vanhojen asioiden toistamista. ”Päryttävyyden” lisäksi koodaustaitoisen toimittajan on mietittävä, miten ulkoasu palvelee jutun aihetta.

Koodaustaitoiset toimittajat kokevat olevansa edelläkävijöitä. Heidän tehtävänä on keksiä uusia verkkokerronnan muotoja ja tuottaa miellyttävän näköistä journalismia. Ohjelmointi poistaa toimittajilta ”bulkkityötä” eli toistettavaa työtä ja vapauttaa heidät kehittämään parempia juttuja.

Haastateltavieni mielestä ohjelmointi ei vaikuta haitallisesti journalismin vakiintuneisiin periaatteisiin, koska esimerkiksi puolueetonta journalismia voi tehdä myös ohjelmoimalla. Haastateltavani kertovat ohjelmoinnin nostaneen verkon tekemisen kokonaan uudelle tasolle. Juttuihin on tullut lisää toiminnallisuutta ja asioita havainnollistavia elementtejä. Koodaustaitoiset toimittajat tietävät, että heidän osaamiselleen on kysyntää jatkossakin. Ennakko-odotusteni mukaisesti kukaan koodaustaitoisista toimittajista ei ollut sitä mieltä, että tulevaisuuden toimittajan pitäisi opetella koodaamaan. Heistä toimittajien olisi tärkeämpää oppia, mitä ohjelmointi on ja mihin sitä voidaan käyttää. Haittaa ohjelmointitaidoista ei kuitenkaan ole, sillä taidot ovat haastateltavieni mielestä kuin lihaksia, joilla on mahdollista saada entistä helpommin töitä.

Tuloksista voi päätellä, miten koodaustaitoiset toimittajat suhtautuvat ohjelmoinnin yleistymiseen arjen toimitustyössä. Olen pyrkinyt käsittelemään koodaustaitoisten toimittajien työtä, merkitystä ja mielipiteitä mahdollisimman monipuolisesti, mutta joitakin asioita jäi mainitsematta. Minusta olisi ollut kiinnostavaa tietää lisää siitä, miten koodaustaitoiset toimittajat kokevat muiden toimituksen jäsenten suhtautuvan heihin. Kokevatko koodaustaitoiset toimittajat, että työnantajat edellyttävät heiltä erilaisia asioita kuin toimittajat, joiden kanssa he tekevät päivittäin yhteistyötä? Saavatko he työssään kuulla, että heitä arvostetaan? Lisäksi haastateltavieni kuvaukset omasta merkityksestä journalismille ja työyhteisölle jäivät mielestäni pinnallisiksi. Olisin halunnut konkreettisempia esimerkkejä tilanteista, joista merkitys käy ilmi. Tulevaisuudessa aihetta kannattaisi tutkia myös muiden toimittajien näkökulmasta. Tämän tutkimuksen tulokset perustuvat koodaustaitoisten toimittajien kokemuksiin ohjelmoinnista, mutta koodaustaidottomien toimittajien näkemykset voisivat olla aivan yhtä kiinnostavia. Onko todella niin, että ohjelmointia pidetään salamyhkäisenä verhona, vai tietävätkö toimittajat ohjelmoinnista enemmän kuin mitä koodaustaitoiset toimittajat luulevat? Mielenkiintoista olisi myös toistaa tämä tutkimus muutaman vuoden kuluttua ja katsoa, onko esimerkiksi koodaustaitoisten toimittajien tarvitsemissa taidoissa tapahtunut muutosta.

5.3 Miten tutkimusmenetelmät sopivat aiheeseen?

Keräsin aineistoni teemahaastatteluilla ja analysoin sitä laadullisesti. Haastattelut olivat mielestäni oikea valinta aineistonkeruumenetelmäksi, koska siten sain monisanaista tietoa koodaustaitoisten toimittajien kokemuksista. Haastattelut etenivät väljästi ennalta pohdittujen teemojen mukaisesti, joten haastateltavat saivat nostaa esille haluamiaan asioita ja itse pystyin esittämään tarvittaessa tarkentavia kysymyksiä. En olisi voinut kysyä kaikilta haastateltavilta samoja kysymyksiä, koska

jokaisen haastateltavan työnkuva on erilainen. Esimerkiksi Helsingin Sanomien datajournalistin työ eroaa monilla tavoilla Ylen koodaavan toimittajan työstä ja Helsingin Sanomien datatuottajan työ on kokonaan oma kokonaisuutensa. Haasteellista on esittää haastattelutilanteissa kysymyksiä, joihin ei itse vahingossa vastaa. Menetelmän ongelmakohdat liittyvätkin haastattelijan ammattitaitoon esittää tasapuolisia kysymyksiä. Mielestäni onnistuin tehtävässä hyvin, joskin aina on myös parantamisen varaa. Jotkut kysymyksistä osoittautuivat vasta ääneen sanottuna niin monimutkaisiksi, että niiden tarkoitusta piti pohtia yhdessä haastateltavan kanssa. Vaikka pyrin esittämään kysymykset aina mahdollisimman neutraalilla tavalla, joihinkin eksyi aina ylimääräisiä täytesanoja, joita minun ei välttämättä olisi tarvinnut sanoa. Muuten haastattelut sujuivat hyvin: Haastateltavat saivat sanottua kaiken haluamansa ja minä sain haastettua heitä sopivan kiperillä kysymyksillä.

Aiemmassa kirjallisuudessa käytetyt aineistonkeruumenetelmät tukevat omaa valintaani. Mielestäni parasta tietoa ulkomaisista koodaustaitoisista toimittajista sain tutkimuksista, joissa koodaustaitoisia toimittajia oli haastateltu. Kaikissa tutkimuksissa ei kerrottu, miten tutkijat olivat haastattelunsa toteuttaneet, mutta monissa tutkimuksissa oli mukana myös määrällisiä tai etnografisia piirteitä. Parhaiten tutkimuksensa toteutti mielestäni Cindy Royal (2010), joka haastatteluiden lisäksi seurasi sivusta, kun koodaustaitoiset toimittajat tekivät töitään. Näin Royal sai varmistuksen siitä, että haastateltavien puheet olivat linjassa heidän tekojensa kanssa. Jos tutkimustani varten tarvittavaa työmäärää olisi pitänyt tai ollut mahdollista kasvattaa, olisin lisännyt myös omaan tutkimukseeni etnografisia osia. En ottaisi mallia Hannafordin (2015) ratkaisusta tehdä haastattelut puhelimitse. Vaikka Hannafordin tulokset ovat kiinnostavia ja varmasti tosia, ei hänen aineistostaan käy ilmi haastateltavien sanaton suhtautuminen kysymyksiin. Vaikka en itsekään analysoinut valtavasti haastateltavien pitämiä taukoja tai virheellisesti aloitettuja lauseita, sain kuitenkin mielestäni huomattavasti arvokasta ja ylimääräistä tietoa haastateltavieni aidosta suhtautumisesta aiheeseen siten, että toteutin haastattelut kasvotusten.

Laadullinen sisällönanalyysi valikoitui analyysimenetelmäksi luonnollisista syistä. Laadullisesti kerättyä aineistoa on luontaista tarkastella laadullisin tutkimusmenetelmin. Suurta aineistoa tuntui järkevältä ensin luokitella ja sitten teemoitella. Haastateltavani puhuivat monesti samoista asioista, mutta eri asiayhteyksissä, joten aineisto muuttui huomattavasti selkeämmäksi, kun kommentteja ryhmitteli omien alaotsikoidensa alle. Kun olin jaotellut haastateltavien sanomisia eri kategorioihin, aloin tarkastella sitä, mitkä asiat nousivat esille usein ja mistä aiheista oltiin eniten eri mieltä. Laadullista sisällönanalyysia ei ole kutsuttu kaikkein puolueettomimmaksi tutkimusmenetelmäksi, koska on totta, että tutkija tekee siinä myös omia valintojaan. Niin on kuitenkin lähes aina jossain

mielessä. Tutkimukseni puolueettomuutta kasvattaa se, että lähestyn aihetta aineistolähtöisesti. Annoin haastateltavilleni valtaa päättää, mistä he haluavat puhua, enkä tutustunut aiempaan kirjallisuuteen ennen kuin olin saanut aineiston kerättyä. Korostan myös, että tutkimukseni välittää haastateltavieni omia kokemuksia ohjelmoinnista arjen toimitustyössä. Menetelmä oli mielestäni oikea valinta analyysimenetelmäksi, vaikka se ei saa suoraan tukea aiemmasta kirjallisuudesta. Ulkomaisissa tutkimuksissa ei ole juuri kerrottu, millä tavoin aineistoa on tulkittu. Uskoisin kuitenkin, että niissä on sovellettu vastaavaa menetelmää.

5.4 Tulosten merkitys journalismille ja toimittajille

Tulokset osoittavat, mitä mieltä koodaustaitoiset toimittajat ovat ohjelmoinnin yleistymisestä arjen toimitustyössä. Tulosten avulla toimittajat voivat pohtia omaa suhtautumistaan ohjelmointiin. Jos koodaustaitoisia toimittajia ei ole uskaltanut lähestyä työympäristössä, avaavat tämän tutkimuksen tulokset heidän ajatusmaailmaansa muille toimittajille. Kuten tuloksista voi päätellä, tavalliset toimittajat eivät koodaustaitoisten toimittajien mielestä tiedä, mitä ohjelmoinnilla voidaan tehdä. Koodaustaitoisilla toimittajilla puolestaan ei ole tarkkaa käsitystä siitä, mitä tavalliset toimittajat ajattelevat ohjelmoinnista. Ajatustenvaihtoa ei tapahdu tarpeeksi, joten tämän tutkimuksen tulosten avulla voidaan rikkoa muureja. Työnantajat voivat hyödyntää tuloksia toimittajien palkkaamisessa. Koodaustaitoisten toimittajien visiot tulevaisuuden toimittajan tarvitsemista taidoista saattavat pitää hyvinkin paikkansa. Työnantajat saavat tietoa siitä, mitä ominaisuuksia tulevaisuuden työntekijältä kannattaa odottaa. Tulokset saattavat myös rohkaista niitä mediataloja, joissa ohjelmoinnin keinoin tuotettavaa journalismia ei ole vielä kokeiltu.

Koodaustaitoisten toimittajien innovatiivinen työskentelytapa saattaa hyvinkin kuvastaa sitä, mihin journalismi on tulevaisuudessa menemässä. Sen vuoksi on mahdollista, että tulokset peilaavat koko journalismin muutosta. On vaikeaa keksiä toista ammattiryhmää, joka työskentelisi yhtä utterasti verkkokerronnan uudistamiseksi. Verkkojournalismin painoarvo kasvaa varmasti tulevaisuudessa, joten on tärkeää ymmärtää, mihin sitä ollaan kehittämässä. Tulokset ovat tärkeitä senkin vuoksi, että yleisöä halutaan palvella mahdollisimman hyvin. Ihmisten tavat kuluttaa journalismia ovat kovassa muutoksessa, ja koodaustaitoiset toimittajat pyrkivät pitämään nämä tavat mielessään, kun he kehittävät uusia esitystapoja. Koodaustaitoisten toimittajien kyvyistä hyödyntää uusinta teknologiaa voi oppia koko media-ala. Tulokset kertovat, miten toimituksissa työskennellään tulevaisuudessa ja millaisia journalististen juttujen halutaan olevan jatkossa.

Samalla tuloksista on hyötyä itselleni ja muille toimittajiksi opiskeleville. Toimittajaopiskelijoille tarkoitettu ohjelmointivalistus on vielä lapsen kengissä, joten on ensiarvoisen tärkeää saada tietoa siitä, miten ohjelmoinnin merkitys tulee kasvamaan. Jotta kaikkien ei tarvitsisi uhrata kaikkea vapaa-aikaansa itsensä kehittämiseen, on ohjelmointikoulutusta tarjottava jo opiskeluvaiheessa. Itse sain tulosten myötä tietää, onko ohjelmointitaidoista toimittajalle hyötyä. Tuloksista on hyötyä myös tulevaisuuden tutkijoille. Koodaustaitoiset toimittajat edustavat muutosta, joka ei ole saavuttanut vielä päätöstään. Tätä uutta ja innovatiivista ryhmää on siis tutkittava tulevaisuudessakin. Muutoksia tapahtuu nopeasti, ja monia seurauksia voidaan vasta arvailla.

LÄHTEET

Boczkowski, Pablo J. 2005. *Digitizing the News: Innovation in Online Newspapers*. Cambridge (Mass.): MIT Press.

Cohen, Sarah, James Hamilton, ja Fred Turner. 2011a. Computational journalism. *Communications of the ACM* 54, (10), 10.1145/2001269.2001288.

Cohen, Sarah, Chengkai Li, Jun Yang ja Cong Yu. 2011b. Computational journalism: A call to arms to database researchers. In: 5th Biennial conference on innovative data systems research, Asilomar, Pacific Grove, CA, 9–12 January.

http://cidrdb.org/cidr2011/Papers/CIDR11_Paper17.pdf.

Cox, Melisma. 2000. The development of computer-assisted reporting. Association for Education in Journalism and Mass Communication, Southeast Colloquium, March 17–18, 2000, University of North Carolina, Chapel Hill.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.631.6220&rep=rep1&type=pdf>.

Flew, Terry, Christina Spurgeon, Anna Daniel, ja Adam Swift. 2012. The promise of computational journalism. *Journalism Practice* 6 (2): 157–71.

Garrison, Bruce. 1998. Newspaper size as a factor in use of computer-assisted reporting. Paper presented to the communication technology and policy division of the Association for Education in Journalism and Mass Communication, Baltimore, MD, 5–8 August.

Gynnild, Astrid. 2014. Journalism innovation leads to innovation journalism: The impact of computational exploration on changing mindsets. *Journalism* 15 (6): 713–30.

Hannaford, Liz. 2015. Computational journalism in the UK newsroom: hybrids or specialists? *Journalism Education* 4. <http://www.lizhannaford.com/wp-content/uploads/2015/07/273700290-Computational-Journalism-by-Liz-Hannaford-Manchester-Metropolitan-University-2.pdf>.

Heikkinen, Seppo, ja Saarenpää, Antti. 2016. Koodi on kaikkialla – lyhyt johdatus ohjelmoinnin maailmaan. Julkaistu 17.9. Viitattu 24.3.2019. <https://yle.fi/aihe/artikkeli/2016/09/17/koodi-kaikkialla-lyhyt-johdatus-ohjelmoinnin-maailmaan>.

Hirsjärvi, Sirkka, ja Helena Hurme. 2008. Tutkimushaastattelu: Teemahaastattelun teoria ja käytäntö. Helsinki: Gaudeamus Helsinki University Press.

Holovaty, Adrian. 2006. A fundamental way newspaper sites need to change. Julkaistu 6.9. Viitattu 14.3.2019. <http://www.holovaty.com/writing/fundamental-change/>.

Houston, Brant. 1996. Computer-assisted reporting. New York: St. Martin's Press.

Karlsen, Joakim, ja Eirik Stavelin. 2014. Computational journalism in norwegian newsrooms. *Journalism Practice* 8 (1): 34–48.

Lehtniemi, Ninni. 2016a. Nerdistan–Designia–Journoland: Best practices in multi-skilled digital journalism teamwork. Reuters Institute Fellowship Paper. <https://reutersinstitute.politics.ox.ac.uk/sites/default/files/research/files/Nerdistan-Designia-Journoland%2520Best%2520practices%2520inmultiskilled%2520digital%2520journalism%2520teamwork.pdf>.

Lehtniemi, Ninni. 2016b. Älä opettele koodaamaan. Julkaistu 17.11. Viitattu 25.3.2019. <https://suomenlehdisto.fi/ala-opettele-koodaamaan/>.

Lewis, Seth C., ja Nikki Usher. 2013. Open source and journalism: Toward new frameworks for imagining news innovation. *Media, Culture & Society* 35 (5): 602–19.

Lewis, Seth C., ja Nikki Usher. 2014. Code, collaboration, and the future of journalism: A case study of the hacks/hackers global network. *Digital Journalism*, 2(3), 383–393.

Lewis, Seth C., ja Nikki Usher. 2016. Trading zones, boundary objects, and the pursuit of news innovation: A case study of journalists and programmers. *Convergence*, 22(5), 543–560.

McGregor, Susan E. 2013. A Brief History of Computer-Assisted Reporting. Columbia Journalism School, Tow Center For Digital Journalism.

Meyer, Philip. 1973. Precision journalism: A reporter's introduction to social science methods. Bloomington: Indiana University Press.

Meyer, Philip. 1991. The new precision journalism. Bloomington: Indiana University Press.

Nordell, Marja-Liisa. 2017. Ylen aluetoimitusten Mediadeski käynnistyy Tampereella. Julkaistu 20.10. Viitattu 25.3.2019. <https://yle.fi/aihe/artikkeli/2017/10/20/ylen-aluetoimitusten-mediadeski-kaynnistyy-tampereella>.

EDU.fi. 2016. Ohjelmoinnin opetukseen liittyvien käsitteiden määritelmät peruskoulun opettajille. Viimeksi muokattu 15.9.2016. Viitattu 12.3.2019. https://www.edu.fi/materiaaleja_ja_tyotapoja/tvt_opetuksessa/ohjelmointi/kasitteet.

Parasie, Sylvain, ja Eric Dagiral. 2012. Data-driven journalism and the public good: "computer-assisted-reporters" and "programmer-journalists" in Chicago. New Media and Society Novembre.

Pilhofer, Aron. 2010. Programmer-journalist? Hacker-journalist? Our identity crisis. Julkaistu 22.4. Viitattu 14.3.2019. <http://mediashift.org/2010/04/programmer-journalist-hacker-journalist-our-identity-crisis107/>.

Powers, Matthew. 2012. In forms that are familiar and yet-to-be invented: American journalism and the discourse of technologically specific work. Journal of Communication Inquiry 36 (1): 24–43.

Reavy, Matthew. 1996. How the media learned computer-assisted reporting. Paper presented to the Southeast Colloquium, Newspaper Division, Association for Education in Journalism and Mass Communication. Roanoke, VA.

Royal, Cindy. 2010. The journalist as programmer: A case study of the New York Times interactive news technology department. In: Paper presented at the international symposium in online journalism, University of Texas, Austin, TX, 23 April. Viitattu 24.3.2019. http://www.cindyroyal.com/present/royal_isoj10.pdf.

Ruusuvuori, Johanna, Pirjo Nikander, Matti K. Hyvärinen, ja Ellibs. 2010. Haastattelun analyysi. Tampere: Vastapaino.

Saaranen-Kauppinen, Anita, ja Anna Puusniekka. 2006. KvaliMOTV – Menetelmäopetuksen tietovaranto. Tampere: Yhteiskuntatieteellinen tietoarkisto. Viitattu 20.11.2018. <http://www.fsd.uta.fi/menetelmaopetus>.

Toikkanen, Ilkka. 2014. Pinnallisista klikkikartoista yhteiskunnalliseen merkittävyyteen: Datajournalismin työprosessi Helsingin Sanomissa ja Ylessä. Pro gradu -tutkielma, Jyväskylän yliopisto. Viitattu 25.3.2019. <https://jyx.jyu.fi/handle/123456789/43189>.

Tuomi, Jouni, ja Anneli Sarajärvi. 2018. Laadullinen tutkimus ja sisällönanalyysi. Uudistettu laitos. ed. Helsinki: Tammi.

Usher, Nikki. 2014. Making news at the New York Times. Ann Arbor, Michigan: University of Michigan Press.

Usher, Nikki. 2016. Interactive journalism: Hackers, data, and code. Urbana: University of Illinois Press.

Varho, Esko. 2016. Yle julkaisi maailman ensimmäisen vaalikoneen 20 vuotta sitten – kadonneeksi luultu koodi löytyi muuttolaatikosta. Julkaistu 23.8. Viitattu 25.3.2019. <https://yle.fi/uutiset/3-9115786>.

Warner, Charles. 2014. Innovation in legacy media companies -- not! Julkaistu 8.1. Viitattu 25.3.2019. <https://www.forbes.com/sites/charleswarner/2014/01/08/innovation-in-legacy-media-companies-not/>.

Wolton, Dominique. 1979. Do you love your VDT? Columbia Journalism Review 18, (2).

LIITTEET

Liite 1. Esimerkki aineistosta. Haastateltavan työnkuvasta kertovat kohdat on merkitty keltaisella ja taidoista kertovat kohdat harmaalla.

K: Kuinka paljon sä saat vaikuttaa niinkun journalistiseen sisältöön siinä?

V: No se vähän riippuu mä oon yleensä mukana silleen niinkun, niinku ideoimassa mukana, siinä mä yritän antaa eniten inputtia ja sitten niinku ediointivaiheessa mä saatan heittää jotain kommenttia mutta ja kyl mä koen että mä saan, koen että mua kuunnellaan ja mutta kyl mä kans silleen et jos siinä on erikseen vaikka just tuottaja joka editoi tekstiä niin en mä sit meen niinku sen tontille, niinku heilumaan vaan keskityn siihen niinku koodipuoleen. Mä luotan että ne on sen oman niinkun hommansa ammattilaisia ja ne tekee niinkun maksimaalisen inputin siihen et mä koen mä koen että mä pystyn vaikuttamaan siihen sisältöön aina. Et ehkä just jossain datajutuissa ehkä sitten enemmän pystyn antaa semmost inputtia, vielä, koska sitä tapaa käsitellä yhdessä sen toimittajan kaa.

K: Millasia noi teidän vaatimukset tässä työssä on? Mitä sun pitää osata jotta sä voit olla koodaava toimittaja?

V: No pitää osata ohjelmoida, ja sitten pitää osata varsinkin verkko-ohjelmoida. Öhm. Pitää olla jonkunlaiset, ei nyt ehkä tarvii olla tilastotieteiden eksperti, mutta jotain niinku semmosta pitää esim osata niinku perus matikkaa tai esim laskee prosentteja tai jotain perusjuttuja, laskujuttuja, pitää osata käsitellä dataa ja just se että pitää olla jonkinlainen käsitys journalismista tai niinku et mitä se on. Ne olis ehkä ne semmoset ehkä suuret suuntaviivat.