

Vilma Toivanen

**KÄYTTÄJÄDATAN KOONTI JA HYÖDYNTÄMINEN
OHJELMISTO- JA PELINKEHITYKSESSÄ**



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2018

TIIVISTELMÄ

Toivanen, Vilma Kerttu

Käyttäjädatan koonti ja hyödyntäminen ohjelmisto- ja pelinkehityksessä

Jyväskylä: Jyväskylän yliopisto, 2018, 35 s.

Tietojärjestelmätiede, kandidaatin tutkielma

Ohjaaja: Palonen, Teija

Teknologisen kehityksen ansiosta käyttäjistä alkaa syntyä monipuolista dataa lähes kaikista heidän toiminnoistaan ohjelmistojen sekä digitaalisten pelien käytön myötä. Digitaaliset pelit muodostavat nykyään merkittävän kokonaisuuden ohjelmistokehityksessä ja pelien käyttötarkoitukset ovat alkaneet laajentua viihteen ulkopuolelle esimerkiksi opetukseen, mikä laajentaa käyttäjädatan monipuolisuutta. Tämä kandidaatin tutkielma, joka toteutetaan kirjallisuuskatsauksena, käsittelee käyttäjädatan merkitystä pelinkehityksessä käyttäen aineistona tieteellisiä julkaisuja. Tutkielma etenee käyttäjädatan käytön tarkastelulla yleisesti ohjelmistokehityksen piirissä, jonka jälkeen tarkastellaan, miten pelinkehitys ilmenee ohjelmistokehityksenä. Tutkielmassa siirrytään tämän jälkeen käyttäjädatan hyödyntämisen tarkasteluun pelinkehityksen asiayhteydessä. Käyttäjädatta on erittäin arvokas resurssi sekä ohjelmisto- että pelinkehitykselle, sillä sen ansiosta voidaan kartoittaa käyttäjän odotuksia ja vaatimuksia, jotka kohdistuvat ohjelmistoon. Pelinkehitys ohjelmistokehityksenä muistuttaa ketteriä ohjelmistokehitysmenetelmiä, mutta erityisesti pelinkehityksessä tulee huomioida käyttäjien vaatimuksia huolellisesti, jotta kehittäjät saisivat omaksuttua käyttäjien subjektiivisia näkökulmia pelinkehitykseen. Käyttäjädatan koonti ja hyödyntäminen edellyttää sekä ohjelmisto- että pelinkehityksessä kehittäjien ja käyttäjien välistä vuorovaikutusta kehitysprosessin aikana, esimerkiksi viestintää sosiaalisen median kautta sekä erilaisia testauksia. Näin varmistetaan, että käyttäjädatta voidaan hyödyntää jatkuvasti kehityksen aikana tukien samalla päätöksentekoa kehitysprosessissa.

Asiasanat: käyttäjädatta, ohjelmistokehitys, pelinkehitys, käyttäjävaatimukset

ABSTRACT

Toivanen, Vilma Kerttu

Collecting and utilizing user data in software and game development

Jyväskylä: University of Jyväskylä, 2018, 35 pp.

Information Systems, Bachelor's thesis

Supervisor: Palonen, Teija

Thanks to the advances in technological development, its users have started to generate rich data of themselves from virtually all of their activities with the use of software and digital games. These days digital games form a significant portion of software development and the use of games has begun to expand beyond entertainment to education, for example, which extends the versatility of user data. This Bachelor's thesis, which is carried out as a literature review, examines the importance of user data in game development, using scientific publications as source material. The thesis progresses by examining the utilization of user data in general within software development, and proceeds by reviewing how game development acts as software development. After this, the thesis moves on to examine how user data is utilized within the context of game development. User data is a very valuable resource for both software and game developments, since it can help clarify the user's expectations and demands towards software. Game development as software development resembles agile software development methods, but during game development especially the user demands must be considered carefully in order to ascertain that developers attain the subjective perspectives that are utilized during game development. Collecting and utilizing user data during both software and game development requires communication between developers and users through contact via social media and various tests, for example. This ensures that user data can be continuously utilized during development while supporting decision-making in the development process.

Keywords: user data, software development, game development, user requirements

SISÄLLYS

TIIVISTELMÄ
ABSTRACT

1	JOHDANTO	5
2	KÄYTTÄJÄDATA OHJELMISTOKEHITYKSESSÄ.....	8
2.1	Käyttäjädatan rooli ohjelmistokehityksessä.....	8
2.2	Käyttäjädatan koontimenetelmät.....	9
2.3	Käyttäjädatan käyttötavat ohjelmistokehityksessä	11
2.4	Käyttäjädatan käyttö ja käyttäjän tietosuoja	13
3	PELINKEHITYKSEN MENETELMÄT JA VAIHEET	16
3.1	Ohjelmistokehityksen määrittely	17
3.2	Pelinkehitys ohjelmistokehityksenä	19
3.3	Pelinkehityksen vaiheet	22
4	KÄYTTÄJÄDATA PELINKEHITYKSESSÄ.....	25
4.1	Käyttäjädatan koontimenetelmiä pelinkehitykseen.....	26
4.2	Käyttäjädatan käyttötavat pelinkehityksessä	27
5	JOHTOPÄÄTÖKSET	29
	LÄHTEET	32

1 JOHDANTO

Nyky-yhteiskunnassa teknologian käytöstä on muodostunut olennainen osa arkipäivää. Teknologian nopean kehityksen saatossa erilaisten ohjelmistojen merkitys on kasvanut ja vahvistanut asemaa vapaa-ajanvieton lisäksi esimerkiksi liiketoiminnassa, työskentelyssä sekä opiskelussa sitä mukaa, kun teknologian ja ohjelmistojen saatavuus on parantunut kehityksen myötä. Teknologian kehityksen myötä myös erilaisten laitteiden sirujen, sensoreiden ja käyttäjien itse syöttämien tietojen avulla syntyy valtavia määriä dataa ohjelmistojen käytöstä (Greengard, 2015, s. 44). Huomioimalla myös data, jota syntyy pilvipalveluiden käytössä, kerätyn datan määrä voi vaihdella teratavuista exatavuihin (Chen, Chiang & Storey, 2012, s. 1166). Tätä käyttäjädataa analysoimalla voidaan huomata erilaisia piirteitä ohjelmistoista sekä käyttäjistä, joita ei ole huomattu ohjelmiston kehityksessä. Käyttäjädatabaasista onkin muodostunut organisaatioille merkittävä liiketoiminnallinen resurssi (Greengard, 2015, s. 56).

Teknologisen kehityksen myötä myös pelit ovat yleistyneet arkielämässä. Käsitteenä peli voidaan määritellä ohjelmistoksi, jossa yksi tai useampi samanaikainen pelaaja suorittaa päätöksentekoa ohjaamalla pelin esittämiä kohteita erilaisilla laitteistoilla, kuten pelikonsolilla, tietokoneella, mobiililaitteella tai älylaitteella, jolle se on asennettu (Aleem, Capretz & Ahmed, 2016, s. 1). Pelien käyttäjä, pelaaja, ponnistelee voidakseen vaikuttaa pelin lopputuloksiin, muodostaen peliin tunnesiteen pelaamisen aikana (Mäyrä, 2008, s. 34). Digitaaliset pelit muodostavat merkittävän kokonaisuuden ohjelmistokehityksessä (Murphy-Hill, Zimmermann & Nagappan, 2014, s. 1), jossa pelit ovat alkaneet saavuttaa merkitystä siten, että niiden käyttötarkoitukset eivät enää rajaudu viih-teeseen, vaan pelejä voidaan yhdistää myös käytännöllisiin käyttötarkoituksiin, kuten koulutukseen ja terveydenhuoltoon (Aleem, Capretz & Ahmed, 2016, s. 4). Tarve saattaa ohjelmistokehittäjiä ja käyttäjiä lähemmäs toisiaan on muodostunut yhdeksi merkittävimmäksi ohjelmistokehityksen menestyksen tekijäksi, jotta kehittäjät pystyisivät vastaamaan tarkasti käyttäjien asettamiin vaatimuksiin (Alvertis, Koussouris, Papaspyros, Arvanitakis, Mouzakitidis, Franken, Kolvenbach & Prinz, 2016, s. 73), mikä pätee myös pelinkehitykseen.

Käyttäjätietoa kerätään pelin käyttäjän, pelaajan, toiminnoista digitaalisen pelin pelaamisen, toisin sanoen ohjelmiston käytön aikana (Kim, Zimmermann, DeLine & Begel, 2016, s. 104). Erityisesti verkkopeleistä voidaan kerätä dataa lähes kaikista pelaajan toiminnoista pelaamisen aikana (Koskenvoima & Mäntymäki, 2015, s. 328). Käyttäjätietoa voidaan jaotella sekä implisiittiseen, että eksplisiittiseen dataan. Implisiittinen käyttäjätietoa voi koostua esimerkiksi datasta, jota syntyy käyttäjien toimista, kuten ohjelmiston painalluksista ja virheilmoitusten syntymisestä ohjelmistossa (Maalej, Nayebi, Johann & Ruhe, 2016, s. 1), kun taas eksplisiittinen käyttäjätietoa koostuu käyttäjiltä suoraan saadusta datasta, kuten palautteista ja kommentteista (Maalej, Nayebi, Johann & Ruhe, 2016, s. 3). Kun käyttäjätietoa kertyy, käyttäjätietoa ja tietoturvan merkitys kasvavat. Käyttäjien tulisi Alpaydinin (2016, s. 157) mukaan saada tietää, mitä dataa hänestä kerätään ja mitä osaa siitä analysoidaan. Hartikaisen (2018) mukaan käyttäjät arvostavat tietosuojaa ja tietojen hallintaa enemmän kuin datamäärän kasvattamista. Esimerkiksi Facebook oli ennen vuotta 2014 mahdollistanut käyttäjätietojen keräämisen palvelun sovellusten, kuten kyselyiden, kautta, jota ohjelmistokehittäjät todennäköisesti myivät eteenpäin kolmansille osapuolille. Vuoden 2014 jälkeen Facebook määräsi, että kehittäjät eivät saaneet kerätä tietoa sosiaalisista verkostoista ilman lupaa. Suomessa astuu voimaan 25.5.2018 uusi tietosuojalaki, jonka avulla pyritään saattamaan tietosuojaa koskeva sääntely ajan tasalle samalla mukauttamalla EU:n jäsenvaltioiden tietosuojan liittyviä asetuksia yhteen (Talus, Autio, Hänninen, Pihamaa & Kantonen, 2017, s. 9). Asetuksen myötä rekisteröidyille käyttäjille taataan oikeudet muun muassa tietojen oikaisuun sekä tietojen poistamiseen, saamalla siten oikeuden tulla unohdetuksi (Talus, Autio, Hänninen, Pihamaa & Kantonen, 2017, s. 25).

Tämän tutkielman tarkoituksena on tuoda esille käyttäjätietoa merkitystä sekä ohjelmisto- että pelinkehitykselle, jota tarkastellaan yhtenä ohjelmistokehityksen muotona. Lisäksi tutkielmassa tarkastellaan, miten käyttäjätietoa hyödynnetään pelinkehityksen aikana. Käyttäjätietoa syntyy valtavia määriä päivittäisessä ohjelmistojen käytössä, joten tutkielma pyrkii kartoittamaan käyttäjätietoa resurssina etenkin ohjelmistokehityksessä, mutta sivuaa käyttäjätietoa roolia lyhyesti myös organisaation päätöksenteon avustajana. Käyttäjätietoa tarkastelussa hyödynnetään sekä ohjelmisto- että pelinkehityksen piirissä kahta tutkimuskysymystä:

- Ensimmäinen kysymys, ”Mikä merkitys käyttäjätietolla on ohjelmistokehityksessä?” tarkastelee käyttäjätietoa roolia ohjelmistokehityksessä yleisesti.
- Toinen tutkimuskysymys koostuu kahdesta osasta, ”Miten ohjelmistokehitys ilmenee pelinkehityksenä?” sekä ”Miten pelinkehitys hyötyy käyttäjätietosta?”, jossa ensimmäisen osan avulla kartoitetaan pelinkehitystä ohjelmistokehityksenä, ja toisen osan avulla tarkastellaan käyttäjätietoa hyödyntämistä pelinkehityksessä.

Tutkielma on toteutettu kirjallisuuskatsauksena, jossa tutkimusaineisto koostuu ohjelmistokehityksen, pelinkehityksen, pelisuunnittelun ja pelianalytiikan tie-

teellisistä julkaisuista ja kirjallisuudesta. Aineistoa on kerätty pääsääntöisesti Google Scholarin sekä ProQuestin avulla, joissa hakusanoina on käytetty muun muassa seuraavia termejä: "software development methodology", "agile software development", "user data in software development", "game development as software development", "game development" sekä "user data in game development". Koska ohjelmisto- ja pelinkehitysmenetelmät kehittyvät jatkuvasti, aineiston haussa asetettiin ehdoksi, että aineiston tulee olla julkaistu vuoden 2010 jälkeen. Poikkeuksena olivat yksittäiset aineistot, joiden Google Scholarin mukainen viittausmäärä oli hyvin korkea ja joiden sisältö tuki tutkielmaa.

Tutkielman sisältö koostuu viidestä luvusta. Ensimmäinen luku kattaa johdannon. Toisessa luvussa tarkastellaan käyttäjätietoa ohjelmistokehityksessä, pyrkien vastaamaan ensimmäiseen tutkimuskysymykseen "Mikä merkitys käyttäjätiedolla on ohjelmistokehityksessä?". Samassa luvussa tuodaan esille ajankohtainen tietosuojasetus ja tarkastellaan, mitä tietoturvan aiheita käyttäjätietoa hyödyntämiseen liittyy. Kolmannessa luvussa kartoitetaan pelinkehityksen menetelmiä ja vaiheita määrittelemällä ensin ohjelmistokehitystä, jotta seuraavien alalukujen avulla voidaan vastata toisen tutkimuskysymyksen ensimmäiseen osaan "Miten ohjelmistokehitys ilmenee pelinkehityksenä?", tuoden samalla esille keskeisimpiä eroja ja haasteita kehitysten välillä. Neljäs luku kartoittaa käyttäjätietoa koonti- ja käyttömenetelmiä pelinkehityksen asiayhteydessä, vastaten samalla toisen tutkimuskysymyksen toiseen osaan "Miten pelinkehitys hyötyy käyttäjätietosta?". Viidennessä kappaleessa esitetään johdopäätöksiä, joita on saatu vastaamalla tutkimuskysymyksiin ja tarkastelemalla tutkimusaineistoa. Tämä tutkielma on suunnattu käyttäjätietoa hyödyntämistä kiinnostuneille ohjelmisto- ja pelinkehitykseen suuntautuneille lukijoille.

2 KÄYTTÄJÄDATA OHJELMISTOKEHITYKSESSÄ

Digitaalisten palveluiden ja ohjelmistojen käyttäjien tuottamaa dataa voidaan kerätä ja tallentaa organisaatioiden käyttöön liiketoiminnassa, esimerkiksi ohjelmistokehityksessä (Chen, Chiang & Storey, 2012, s. 392). Ohjelmistokehityksestä on muodostunut liiketoiminnan kannalta kriittinen aktiviteetti, sillä ohjelmistot kattavat nykyään jokaisen liiketoiminnan osa-alueen, ja ohjelmistot ovat vahvasti läsnä myös arkielämässä (Fuggetta & Di Nitto, 2014, s. 1). Käytämme arkielämässä tietotekniikkaa melkein tiedostamatta, mikä tuo organisaatioille uusia liiketoiminnallisia mahdollisuuksia, joita voidaan tavoitella ohjelmistokehityksen avulla. Vaikka ohjelmistokehityksen kustannukset ovat laskeutuneet teknologioiden ja ohjelmistokehitysmenetelmien kehittymisen myötä, organisaatioiden välinen kilpailu on kasvanut, kun teknologiasta on tullut saatavampaa ja tieto pystytään välittämään tehokkaasti (Alvertis ym., 2016, s. 73).

Tässä luvussa tarkastellaan käyttäjätiedon hyödyntämistä ohjelmistokehityksessä. Aluksi määritellään roolia käyttäjätiedolle ohjelmistokehityksessä sekä sen merkitystä liiketoiminnalle, jonka jälkeen siirrytään tarkastelemaan käyttäjätiedon yleisimpiä koontimenetelmiä ja käyttötapoja ohjelmistokehityksessä. Samalla huomioidaan, miten käyttäjätieto voi tukea ohjelmistokehityksen päätöksentekoa. Näiden lukujen jälkeen tarkastellaan käyttäjätiedon käyttöä ja käyttäjän tietosuojaa sekä uuden tietosuojasetuksen merkitystä käyttäjätiedon hyödyntämiselle.

2.1 Käyttäjätiedon rooli ohjelmistokehityksessä

Kilpailukyvyyn ylläpito nopeasti muuttuvassa liiketoimintaympäristössä lukeutuu myös organisaatioiden haasteisiin, mikä on ominaista tietotekniselle toimialalle, jotta organisaatio kykenisi pitämään kiinni markkinaosuudestaan sekä kasvattamaan sitä (Papadopoulos, 2015, s. 455). Myös ohjelmistokehityksen tuominen lähemmäs käyttäjien saataville tuo oman haasteensa organisaatiolle (Franken, Kolvenbach, Prinz, Alvertis & Koussouris, 2015, s. 188). Tarve saattaa

ohjelmistokehittäjät lähemmäs käyttäjiä on muotoutunut yhdeksi merkittävimmäksi menestyksen tekijäksi, jotta kehittäjät pystyisivät vastaamaan tarkasti käyttäjien asettamiin vaatimuksiin. Erityisesti ohjelmistokehityksen alkuvaiheessa käyttäjien ja ohjelmistokehittäjien yhteen saattaminen on tärkeää, jotta kehityksessä voitaisiin havaita puutteita ja vähentää ohjelmistokehityksen kustannuksia puutteiden korjaamisesta. Näin saataisiin myös parannettua ohjelmiston merkityksellisyyden mielikuvaa käyttäjän näkökulmasta. (Alvertis ym., 2016, s. 73.)

Vaikka ohjelmistoja tarkastellaan usein kuluttajatuotteena, jonka kuluttamien resurssien tulisi vastata pelkästään markkinaodotuksiin, ohjelmistoista on muodostunut kriittinen osa organisaation tuotteita ja palveluita (Fuggetta & Di Nitto, 2014, s. 10). Käyttämällä sosiaalisen median palvelukanavia ohjelmistokehittäjät saavat reaaliaikaisen katsauksen käyttäjien toimintaan, suosittuihin suuntauksiin sekä asenteisiin muun muassa politiikkaan, viihteeseen ja kulutustottumuksiin liittyen. Sosiaalisen median palvelut ovat alkaneet suosia algoritmien käyttöä, joilla voidaan tarkastella ohjelmistojen käyttäjämääriä, niihin kohdistuneita hakuja, ainutkertaisia käyttäjiä, painettuja painikkeita, sisällön jakoa, käyttäjän ystävien ja seuraajien määrää sekä muutoksia näissä sosiaalisissa piireissä, lukuisten muiden tarkastelukohteiden lisäksi. (Greengard, 2015, s. 56.) Jotta ohjelmistokehittäjät voisivat ymmärtää käyttäjien tarpeita, kehittäjät voivat tarkkailla käyttäjädataa. Mikäli käyttäjädata jätetään huomioimatta ohjelmistokehityksessä, toiminnasta tulee riskialtista sekä ohjelmiston että organisaation liiketoiminnan kannalta (Maalej, Nayebi, Johann & Ruhe, 2016, s. 5).

Käyttäjädatabaasi on muodostunut tärkeä liiketoiminnallinen vara, jota analysoimalla voidaan huomata puutteita ohjelmiston toiminnassa, ja vähentää ohjelmistokehityksen kuluja (Greengard, 2015, s. 56). Käyttäjädatabaasiin liittyvien liiketoiminnallisten mahdollisuuksien esiintyminen on kasvattanut organisaatioiden kiinnostusta liiketoimintatiedon hallintaa (engl. Business Intelligence) kohtaan. Liiketoimintatiedon hallinnalla viitataan ohjelmistokehityksen toimialalla usein tärkeän liiketoiminnallisen datan analysoinnin tekniikoihin, teknologiaan, järjestelmiin ja sovelluksiin, jotka mahdollistavat kattavamman käsityksen organisaation markkina-alueesta tukien päätöksentekoa. (Chen, Chiang & Storey, 2012, s. 1166.)

2.2 Käyttäjädatan koontimenetelmät

Jotta ohjelmistokehitys pystyy hyödyntämään käyttäjädataa liiketoiminnan menestyksen takaamiseksi, käyttäjädataa tulee koota ohjelmistokehityksen käyttöön. Käyttäjistä kerätään dataa esimerkiksi heidän käyttäytymisen ja toiminnan kannalta ohjelmistosta sekä määritellysti muun muassa sosiaalisen median palveluiden kautta (Kim, Zimmermann, DeLine & Begel, 2016, s. 104). Kuten edeltävässä kappaleessa mainittiin, sosiaalisen median palveluiden kautta voidaan kerätä tarkkoja käyttäjätietoja, esimerkiksi käyttäjän sukupuolesta, iästä, maantieteellisestä sijainnista sekä käyttäjien suosimista toiminnoista. Käyttä-

jätietoja voidaan hyödyntää käyttäjien tarpeiden tarkempaan kartoittamiseen, jotta niihin voitaisiin kohdistaa ohjelmistokehityksen resursseja tarkoituksenmukaisesti. (Alvertis ym., 2016, s. 76.) Sosiaalisen median palveluita käytetään myös yhä enemmän viestintävälineinä käyttäjien ja ohjelmistokehittäjien kesken, jotta käyttäjät saataisiin otettua mukaan ohjelmistokehitykseen (Alvertis ym., 2016, s. 75).

Käyttäjistä kerätyn datan määrä on kasvanut merkittävästi pilvipalveluiden yleistymisen myötä (Kim, Zimmermann, DeLine & Begel, 2016, s. 104). Pilvipalveluiden käytössä syntyy niin sanottua Big Dataa, jolla tarkoitetaan datajoukkoja, joiden rakenne on hyvin monimutkainen ja jota syntyy määrällisesti hyvin paljon. Datan rakenteen monimuotoisuus voi vaihdella esimerkiksi fyysisten sensoreiden rekisteröimästä ja sosiaalisen median kautta kerätystä datasta. Datan määrä voi taas vaihdella teratavuista exatavuihin. Tämän takia Big Datan datajoukot vaativat kehittyneitä ja ainutlaatuisia tiedonhallinnan menetelmiä, analysointia sekä datan visualisoinnin tekniikoita, jotta kerätystä Big Datasta johdettaisiin hyödyllistä dataa. (Chen, Chiang & Storey, 2012, s. 1166.) Big Datan käsite kattaa myös datan keräämisen, tallentamisen ja sen käytön, jossa data voi olla joko jäsenneiltyä tietokantoihin tallennettua dataa, tai tietokantojen ulkopuolista jäsennelemätöntä dataa (Greengard, 2015, s. 44). Big Datan ominaisuuksiin kuuluu siis tiedon määrän jatkuva kasvu, tämän kasvun nopeus sekä datan kirjavuus, joten Big Dataan liittyvinä haasteina onkin tunnistaa hyödyllistä dataa ja hyödyntää sitä tehokkaasti ohjelmistokehityksessä (Greengard, 2015, s. 46).

Käyttäjät voivat suostua itse luovuttamaan tietojaan ohjelmiston käyttäjäprofiilista, mitä voidaan toteuttaa käyttäjän ja ohjelmistokehittäjän välillä sovel-lusohjelmointirajapintojen (engl. Application Programming Interface) kautta (Alvertis ym., 2016, s. 76). Ohjelmistojen järjestelmät käyttävät sovellusohjelmointirajapintoja tarvitun datan hakemiseen silloin, kun sitä tarvitaan, myös automaattisesti, sijoittaen oikean datan oikeaan paikkaan, mikä mahdollistaa esimerkiksi datan louhinnan, kasvojen tunnistamisen ja kielten kääntämisen (Greengard, 2015, s. 56). Sovellusohjelmointirajapintojen käytön hyötyihin kuuluvat datan keräämisen nopeus ja saatavilla olevan datan määrä, mutta heikkouksia ovat taas sovellusohjelmointirajapintojen läpinäkymättömyys, mikä voi aiheuttaa mahdollisia puutteiden ilmenemisiä datassa (McCormick, Lee, Cesare, Shojaie & Spiro, 2017, s. 399). Datan käsittelyyn voidaan käyttää myös tiedonlouhintaa, jotta pystytään tarkastelemaan datan asiayhteyksiä, jakautumisia, klusteroitumista sekä mahdollisia poikkeamia (Chen, Chiang & Storey, 2012, s. 1166). Datan valmistaminen tiedonlouhintaa varten on moniosainen prosessi, jossa tietokannasta valitaan tarkoituksen mukaisesti mielenkiintoisia osia, jotka kootaan omiksi tietokannoiksi, samalla mahdollisesti kokoamalla yhteen dataa useasta eri tietokannasta. Data tallennetaan tietovarastoon, jossa sitä voidaan tarkastella erilaisilla analyysin työkaluilla, kuten verkkoanalyysin käsittelyllä. (Alpaydin, 2016, s. 154.) Yleensä kun puhutaan datan analyysistä, tarkoitetaan tilastollisten analyysimenetelmien lisäksi tiedonlouhinnan prosesseja (Chen, Chiang & Storey, 2012, s. 1174).

Kun ohjelmiston ja käyttäjädatan ominaisuuksia tarkastellaan huolellisesti, voidaan löytää keinoja saavuttaa haluttuja vaikutuksia ohjelmistokehityksessä (Chen, Chiang & Storey, 2012, s. 1169). Verkkoyhteyksien kautta kerätty käyttäjädاتا kerätään usein verkkoalustalta, joka järjestetään tulkittavaan muotoon sovellusohjelmointirajapinnan avulla, jolla voidaan määrittää mitä käyttäjädatan ominaisuuksia halutaan hakea verkkoalustojen ohjelmistoista (McCormick, Lee, Cesare, Shojaie & Spiro, 2017, s. 398). Näiden verkko-ohjelmistojen kautta kerätty käyttäjädاتا on usein järjestelemätöntä ja omaa runsaasti laadullisia ominaisuuksia, kuten käyttäjien mielipiteitä ja käyttäjäpalautteita (Chen, Chiang & Storey, 2012, s. 1169). Verkkoalustoja käyttävien ohjelmien sekä pilvipalveluiden lisäksi ohjelmistotoimittajat keräävät suuria määriä epäsuoraa käyttäjädاتا käyttäjien tietojen, virhelokien ja sensoreiden keräämän datan kautta sekä käyttäjien palautteen ja ideoinnin avulla (Maalej, Nayebi, Johann & Ruhe, 2016, s. 1). Käyttäjädатаsta johdettu palaute voi koostua ohjelmiston virheen ilmoittamisesta, kuten ohjelmistoon kohdistuvien odotusten vastaisesta toiminnasta sekä ohjelmiston ominaisuuksien pyynnöistä, joissa käyttäjät pyytävät ohjelmistokehittäjiä oikaisemaan ohjelmiston puutteita tai toiminnallisuuksien muokkaamista. Käyttäjädатаsta saadun tekstipohjaisen palautteen luokitteluun ohjelmistokehittäjille oleelliseksi tiedoksi ei ole yhtä oikeaa menettelytapaa. Palautteen luokittelu ohjelmistokehityksen hyödyksi edellyttää sen tehokasta käsittelyä, kuten yksittäisten hyödyllisten palautteiden esittämistä ohjelmistokehittäjille sekä sidosryhmille. (Maalej, Nayebi, Johann & Ruhe, 2016, s. 2.) Palautetta voidaan kerätä myös sosiaalisen median palveluiden kautta, kuten keskusteluareenoilta ja käyttäjäpalautteista, joissa käyttäjät voivat antaa palautetta ohjelmistosta, arvioida ohjelmiston kehitystä, ilmoittaa ohjelmistokehittäjille virheistä, sekä pyytää ohjelmistoon uusia ominaisuuksia (Maalej, Nayebi, Johann & Ruhe, 2016, s. 1). Palautetta voidaan hyödyntää ohjelmistokehityksen lisäksi myös ohjelmistoa testattaessa (Franken, Kolvenbach, Prinz, Alvertis & Koussouris, 2015, s. 191).

Teknologian kehityksen myötä käyttäjädatan rooli on muuttunut passiivisesta roolista, jossa data syntyi ohjelmistojen prosessoinnissa ja tulosteina, keskeiseen resurssiasemaan ohjelmistokehityksessä (Alpaydin, 2016, s. 11). Jotta käyttäjädата osoittautuu hyödylliseksi ohjelmistokehitykselle, se tulee analysoida, suodattaa ja tiivistää ohjelmistokehittäjille ja päätöksenteon tueksi (Maalej, Nayebi, Johann & Ruhe, 2016, s. 3).

2.3 Käyttäjädatan käyttötavat ohjelmistokehityksessä

Käyttäjädатаa voidaan kerätä eri menetelmin ja eri muodoissa ohjelmistokehityksen tueksi sensoreiden keräämän datan ja virhelokien lisäksi myös käyttäjiltä saatuna eksplisiittisenä palautteena. Tämä käyttäjiltä saatu palaute toimii viestintävälineenä ohjelmiston käyttäjien ja kehittäjien välillä, minkä avulla kehittäjät kykenevät käyttämään kerättyä käyttäjäpalautetta käyttäjien tyytyväisyyden parantamiseksi ohjelmistokehityksessä, vaikka tekstimuotoinen palaute

on usein jäsentelemätöntä ja palautteiden väliset laatuerot ovat suuria (Panichella, Sorbo, Guzman, Visaggio, Canfora & Gall, 2015, s. 281). Kuitenkin jatkuva palaute, mitä käyttäjädatabasta kerätään, mahdollistaa käyttäjien ja ohjelmistokehittäjien yhteistyön (Franken, Kolvenbach, Prinz, Alvertis & Koussouris, 2015, s. 188). Käyttäjiltä saatuihin palautteisiin lukeutuvat ohjelmiston virheilmoitukset voidaan koota ohjelmistokehittäjille, jotta nämä virheet voidaan huomioida ohjelmistokehityksessä. Ominaisuuksien pyynnöt voidaan esittää ohjelmistokehityksen esitutkimusvaiheesta vastaavalle ohjelmistokehittäjälle sekä kehitysprojektin hallinnoitsijalle tulevien ohjelmistovaatimusten kartoittamiseksi. (Maalej, Nayebi, Johann & Ruhe, 2016, s. 2.) Käyttäjiltä kerätty palaute ohjelmistokehityksen varhaisessa vaiheessa voi parantaa ohjelmiston laatua merkittävästi (Peischl, Ferk & Holzinger, 2015, s. 517). Käyttäjäpalautteiden hyödyntäminen voi silti osoittautua haasteelliseksi, sillä tekstipohjaisen palautteen asiayhteys ei usein tule ilmi palautteesta, joten sitä on vaikea suhteuttaa oikeaan asiayhteyteen saati ongelmaan (Maalej, Nayebi, Johann & Ruhe, 2016, s. 3). Jotta käyttäjiltä saatu palaute osoittautuisi hyödylliseksi ohjelmistokehittäjien näkökulmasta, ohjelmistokehityksen tulisi tukea palautteen luomista ja jakamista tukemalla yhteistyötä ja sosiaalisia verkostoja ohjelmistojen käyttäjien, kehittäjien ja sidosryhmien välillä (Franken, Kolvenbach, Prinz, Alvertis & Koussouris, 2015, s. 190).

Vaikka käyttäjien sosiaalisia aktiviteetteja voidaan mitata sosiaalisen median palvelujen tilien kautta, on olemassa tapoja luokitella ja seurata käyttäjien toimintaa myös verkkoyhteyden ulkopuolella implisiittisesti fyysisten sensoreiden sekä sovellusten avulla, joiden avulla ohjelmistokehityksessä voidaan huomata käyttäjän tottumuksia, joita käyttäjä itse ei välttämättä tiedosta (Alvertis ym., 2016, s. 76). Joachimsin, Grankan, Panin, Hembrooken ja Gayn (2017, s. 8) mukaan ohjelmistossa tehtyinä painalluksina kerättyyn käyttäjätietoon vaikuttaa käyttäjien luottamus ohjelmiston odotettuun toimintaan ja laatuun, joita on haastavaa tarkastella. Implisiittisen käyttäjätieton avulla kyetään myös kartuttamaan käyttäjän riippumattomia ja objektiivisia näkemyksiä ohjelmistosta, esimerkiksi huomioimalla käytön ajankohtaa (Maalej, Nayebi, Johann & Ruhe, 2016, s. 3). Tarkastelemalla käyttäjätietoa voidaan huomata esimerkiksi käyttäjän tottumuksia verkkokaupassa asioidessa, kuten tuotteiden selaamista ja ostomenetelmien hyödyntämistä (Chen, Chiang & Storey, 2012, s. 1167). Käyttäjätieton avulla voidaan huomata kaavoja ohjelmiston käytössä sekä tehdä johtopäätöksiä uusista markkinamahdollisuuksista, jotka olisivat jääneet huomaamatta ilman käyttäjätieton käyttöä. Myös mobiililaitteista koottu käyttäjätieto voi tuoda esille tietoa esimerkiksi ohjelmistojen aikaleimoista, käyttäjien kirjautumisista ja käyttäjän maantieteellisestä sijainnista. (Greengard, 2015, s. 41.) Erilaisten laitteiden sirujen ja sensoreiden sekä käyttäjien syöttämien tietojen kautta syntyy valtavia datamääriä (Greengard, 2015, s. 44), ja vaikka sensorit keräävät kirjavaa dataa muun muassa lämpötilasta, sensorin tärähtelystä, kameroista, äänestä sekä videosta, vaaditaan aina tietokoneita, tallennusjärjestelmiä ja ohjelmistoja, jotta datasta voidaan johtaa informaatiota (Greengard, 2015, s. 58). Fyysisillä sensoreilla voidaan kuitenkin tunnistaa käyttötilanteita sekä

asiayhteyksiä ohjelmistokehityksen ja liiketoiminnan tueksi (Maalej, Nayebi, Johann & Ruhe, 2016, s. 4).

Liiketoiminnan tietostandardien kehittyminen on auttanut suuresti liiketoiminnallisen datan luonnissa ja sen hyödyntämisessä, samalla nopeuttaen maailmanlaajuisia tietoverkkoyhteyksiä (Chen, Chiang & Storey, 2012, s. 1168). Teknologian kehittyessä ohjelmistojen kohdeympäristöjen määrä on kasvanut, joten ohjelmistokehittäjien tulee ottaa kohdeympäristöjen kirjavuus huomioon ohjelmistokehityksessä. Kohdeympäristöjen määrän kasvaessa myös ohjelmistojen laadunvalvonnan ja tietoturvan merkitys on kasvanut suuresti. (Fuggetta & Di Nitto, 2014, s. 8.) Ohjelmistokehityksen laadunvalvontaa suoritetaan ohjelmiston rinnakkaistestauksella sekä hyväksyntätestauksella, joilla varmistetaan ohjelmiston kyvykkyys vastata käyttäjien vaatimuksiin (Papadopoulos, 2015, s. 457). Epäselvät käyttäjävaatimukset ohjelmistoa kohtaan lisäävät haasteita käyttäjakeskeisen ja hyödyllisen ohjelmiston kehitykseen (Peischl, Ferk & Holzinger, 2015, s. 512), mutta käytettävyyden toteuttaminen ohjelmistokehityksessä saattaa osoittautua hankalaksi käytettävyyden parantamiseen tarkoitettujen menetelmien kustannusten ja niihin kuuluvien aikaresurssien takia (Peischl, Ferk & Holzinger, 2015, s. 517). Ohjelmistojen tulee täyttää niille asetetut liiketoiminnalliset vaatimukset ja ohjelmistokehityksen tulee varmistaa, että vaadittu palvelun taso toteutetaan, huomioiden myös, että vaatimukset käyttäjän yksityisyyden kunnioittamisesta täytetään (Peischl, Ferk & Holzinger, 2015, s. 512).

Käyttäjädatta voi siis tukea ohjelmistokehittäjien päätöksentekoa ja ohjelmistoon kohdistuvien vaatimusmäärittelyjen kartoittamista myös ilman tekstipohjaista käyttäjäpalautetta esimerkiksi tutkimalla käyttäjädatta ohjelmiston ominaisuuksien käyttämisen järjestyksestä, käytön kestoa, kellonaikaa, käyttäjän sormen liikkeen tarkkailua laitteen näytöllä sekä muita asiayhteyksiä huomioimalla, minkä avulla käyttäjästä muotoutuu tarkempi käsitys ohjelmistokehittäjille (Maalej, Nayebi, Johann & Ruhe, 2016, s. 4). Käyttäjädatan avulla ohjelmistokehittäjät kykenevät ymmärtämään käyttäjän toimintoja ja käyttäjäpalautetta paremmin, ja siten tekemään perustellumpia päätöksiä ohjelmistokehityksessä (Maalej, Nayebi, Johann & Ruhe, 2016, s. 3). Käyttäjädatta voidaan hyödyntää ohjelmistokehityksen eri vaiheissa sekä asettaa käyttäjädattasta koottu vaatimukset etusijalle seuraavassa ohjelmistokehityksen vaiheessa, iteraatiossa (Maalej, Nayebi, Johann & Ruhe, 2016, s. 5). Alan kirjallisuus ja tutkimukset tukevat väitettä, että ohjelmiston käytettävyyden jatkuva huomioiminen ohjelmistokehityksen koko elinkaaren ajan tuottaa suuria liiketoiminnallisia hyötyjä, sillä käytettävyys koetaan tärkeänä ohjelmiston laatua parantavana tekijänä (Peischl, Ferk & Holzinger, 2015, s. 509).

2.4 Käyttäjädatan käyttö ja käyttäjän tietosuoja

Kun käyttäjädatta kertyy ja siitä alkaa tarkentua yksityiskohtia sen käsittelyn aikana, käyttäjädatan yksityisyyden ja tietoturvan merkitys kasvaa. Perusvaa-

timuksena käyttäjätietojen yksityisyydelle voisikin pitää sitä, että käyttäjän, joka tuottaa dataa, tulisi saada tietää mitä dataa ja mitä määriä hänestä kerätään, mitä osaa tästä käyttäjätiedoista tallennetaan ja tullaanko tätä dataa koskaan analysoidaan. (Alpaydin, 2016, s. 157.) Organisaatioiden kohtaamiin haasteisiin kuuluu siis myös tietomurron riski, jossa käyttäjistä kerättyä dataa päätyy joko vahingossa tai tarkoituksenmukaisesti organisaation ulkopuolisille osapuolille, mikä voi vahingoittaa käyttäjän mielikuvaa organisaatiosta. Kun käyttäjä on kerran joutunut tietomurron kohteeksi, käyttäjä usein lopettaa asiakassuhteensa organisaation kanssa. (Goode, Hoehle, Venkatesh & Brown, 2017, s. 703.)

Tietoturvan kannalta erityisen suuren riskin muodostavat mobiililaitteet ja niiden ominaisuudet, sillä mobiililaitteita on helppo kuljettaa paikasta toiseen, samalla yhdistäen laitteita useampaan langattomaan verkkoyhteyteen, jotka saattavat omata puutteellisen tietosuojan, ja laite voi myös unohtua tai tulla varastetuksi. Näin käyttäjän mobiililaitteet tulevat alttiiksi tietoturvahalle, jossa on mahdollista menettää käyttäjän henkilökohtaista dataa mobiililaitteesta. (Peischl, Ferk & Holzinger, 2015, s. 511.) Tietoturvan muodostaa myös tietovuoto, jolla tarkoitetaan elektronisen palvelun, kuten ohjelmiston, toiminnan häiriötä, jossa arkaluontoista taloudellista, työntekijöille henkilökohtaista tai käyttäjätietojen dataa vapautetaan organisaation ulkopuolisille osapuolille joko vahingossa tai tarkoituksenmukaisesti esimerkiksi hakkeroinnin tai työntekijän sabotaasin kautta (Goode, Hoehle, Venkatesh & Brown, 2017, s. 705).

Tietoturvaan sidoksissa olevan käyttäjätietojen käsittely ja analysointi on hyvin haasteellista monen datalähteen, datan rakenteiden kirjavuuksien sekä suurten datamäärien takia (Chen, Chiang & Storey, 2012, s. 1172). Suomessa 25.05.2018 astuu voimaan EU:n tietosuoja-asetus. Asetuksen tarkoituksena on saattaa tietosuoja koskeva sääntely ajantasaiseksi ja tukea digitaalisen talouden kehitystä yhdenmukaistamalla EU:n jäsenvaltioiden tietosuoja koskevat säädökset. Tietosuoja-asetus koskee kaikkia soveltamisalaan kuuluvia henkilötietoja käsitteleviä organisaatioita. Asetuksessa henkilötieto määritellään vastaavasti kuin henkilötietolaissa. (Talus, Autio, Hänninen, Pihamaa & Kantonen, 2017, s. 9.) Tässä tietosuoja-asetuksessa säädetään henkilötietojen käsittelyä koskevista periaatteista, jotka ohjaavat rekisterinpitäjiä käsittelemään henkilötietoja rekisteröidyn oikeuksia ja vapauksia kunnioittavalla tavalla, huolehtien myös siitä, että tietosuojaperiaatteita noudatetaan kaikissa henkilötietojen käsittelyvaiheissa. Rekisterinpitäjän tulee myös kyetä osoittamaan, että periaatteita noudatetaan. (Talus, Autio, Hänninen, Pihamaa & Kantonen, 2017, s. 12.) Rekisterinpitäjä määrää pääsääntöisesti itse asianmukaisista suojatoimista, huomioiden käytettävissä olevan tekniikan, toteuttamiskustannukset, käsittelyn luonteen, laajuuden, asiayhteyden, tarkoitukset sekä henkilöiden oikeuksiin ja vapauksiin kohdistuvat riskit. (Talus, Autio, Hänninen, Pihamaa & Kantonen, 2017, s. 14.)

Tämä tietosuoja-asetus takaa rekisteröidylle oikeuden tietojen oikaisemiseen ja niiden poistamiseen. Rekisteröidyllä on siis oikeus tulla unohdetuksi. (Talus, Autio, Hänninen, Pihamaa & Kantonen, 2017, s. 25.) Organisaatioiden tulisi toimia läpinäkyvästi käyttäjätietojen keräämisessä ja sen käyttötarkoituksen

selventämisessä. Käyttäjää tulisi tiedottaa datan keruusta sekä sen käytöstä.
(Alpaydin, 2016, s. 157.)

3 PELINKEHITYKSEN MENETELMÄT JA VAIHEET

Jotta ohjelmistokehityksen alussa hyödyttäisiin parhaalla mahdollisella tavalla käyttäjien ja ohjelmistokehittäjien vuorovaikutuksesta, vuorovaikuttamisen tulisi edetä ohjelmistokehitysmenetelmien ja teknisten toteutusten sanelemana. Samalla vähennettäisiin käyttäjistä kerätyn tiedon ylikuormittamista sekä ohjelmistokehityksen kannalta turhanpäiväisen käyttäjäpalautteen käsittelyä. (Alvertis ym., 2016, s. 74.) Uusien ohjelmistojen kehittyessä, esimerkiksi mobiilialustoilla, saadaan uusia mahdollisuuksia uusien käyttäjäsuhteiden muodostumiseen (Lesser & Ban, 2016). Kehityksen lisäksi ohjelmistot muuttuvat jatkuvasti käyttäjien vaatimusten mukaan, mikä lisää Fuggettan ja Di Nitton (2014, s. 6) mukaan haasteita myös ohjelmiston käyttöönotolle ja toiminnalle. Kuitenkin uudet kehitykset ohjelmistokehityksessä tuovat esille uusia kanavia kehitykselle ja mahdollisuuksia parantaa ohjelmistokehityksen tuottavuutta (Lesser & Ban, 2016, s. 42). Pelinkehityksestä on muovautunut yksi huomattavimmista alueista ohjelmistokehityksessä, jossa yhdistyvät ohjelmiston kehittäminen, erilaisten ammattilaisten välinen yhteistyö, tasapainoilu ohjelmistoon kohdistuvien resurssivaatimusten välillä, käyttäjän ja teknologian välinen viestintä sekä ohjelmointi. Monissa muissa ohjelmistokehityksen piireissä ei vaadita yhtä runsaasti sekä teknisiä, että hallinnollisia taitoja, kuin pelinkehityksessä, jossa ohjelmiston tulee myös vastata usein subjektiivisiin käyttäjävaatimuksiin ja olla viihdyttävää pelata. (Scacchi, 2017, s. 111.) Pelinkehityksen haasteena on löytää tasapaino pelin erityispiirteiden ja joustavuuden välillä, sillä peli tarvitsee tyypillisesti tunnistettavan identiteetin, jotta se erottuisi muista. Peliohjelmiston tulee olla myös joustava, jotta se soveltuisi uusien pelien ja kehitystoimintojen kehittämiseen. (Salen & Zimmerman, 2004, s. 457.)

Tässä luvussa tarkastellaan pelinkehitystä yhtenä ohjelmistokehityksen muotona. Ohjelmistokehityksen määrittelyssä tarkastellaan ohjelmistokehityksen yleisimpiä menetelmiä, menestymiseen liittyviä tekijöitä sekä tyypillisiä vaihteita. Ohjelmistokehityksen määrittelyssä keskitytään kahteen tyypilliseen ohjelmistokehityksen menetelmään, perinteisiin ja ketteriin malleihin, joista nimetään vesiputous- ja Scrum-mallit tyypillisinä menetelmien edustajina. Tämän jälkeen siirrytään tarkastelemaan pelinkehitystä ohjelmistokehityksenä,

tuoden esille kehitysprosessien välisiä eroavaisuuksia ja samankaltaisuuksia. Tästä siirrytään tarkastelemaan pelinkehityksen vaiheita huomioiden ohjelmistokehityksen ominaisuuksia, mikä tuo edelleen esille kehitysprosessien eroja ja samankaltaisuuksia.

3.1 Ohjelmistokehityksen määrittely

Ohjelmistokehitykselle on ominaista teknologioiden ja käytettyjen standardien sekä ohjelmointikielien nopea muuttuminen samalla, kun uudet viitekehukset korvaavat vanhat ja kun uusia laitteita kehitetään markkinoille ja tuodaan lopulta myös käyttäjille saataviksi (Despa, 2014, s. 38). Perinteiset ohjelmistokehitysmenetelmät sekä jakelukanavat ovat muuttuneet uusien kehitysten mukana saapuneiden konseptien, kuten avoimen lähdekoodin kehityksen, ketterien menetelmien muovautumisen ja mobiililaitteiden yleistymisen myötä (Fuggetta & Di Nitto, 2014, s. 3). Myös hallinnolliset ja inhimilliset tekijät omaavat avainaseman menestymisen tavoittelussa ohjelmistokehityksessä (Fuggetta & Di Nitto, 2014, s. 3). Maantieteellisesti hajautettujen kehitystiimien avulla voidaan saavuttaa yhteys käyttäjiin eri maissa, samalla vähentäen kuluja kehityksen toteuttamisessa ulkoistamisen avulla (Fuggetta & Di Nitto, 2014, s. 4). Nykyään ohjelmistotuotanto toteutetaan usein maantieteellisesti hajautetuissa tiimeissä (Alvertis ym., 2016, s. 75), sillä ohjelmistokehitys ei välttämättä vaadi kasvokkain toteutettua viestintää ja kehityksen toteuttamisessa voidaan hyödyntää verkkoyhteyden palveluita, kuten Dropbox-tiedostopalvelua sekä Skype-viestintäpalvelua (Despa, 2014, s. 38). Kehittäjien tulee silti kommunikoida keskenään erityisesti ohjelmistokehityksen varhaisessa vaiheessa ohjelmiston julkaisua ja ylläpitoa ajatellen. Kommunikoinnilla voidaan jakaa kokemuksia ja tietämyksiä, niin sanottua hiljaista tietoa, ja ylläpitää jatkuvaa keskustelua mahdollisista ratkaisuksista, sopia muutoksista sekä pysyä ajan tasalla jatkuvista ohjelmistokehityksen prosesseista. (Alvertis ym., 2016, s. 75.)

Despan (2014, s. 39) mukaan ohjelmistokehityksen toteuttaminen vaatii kuitenkin useita työvaiheita, joissa on määritelty omat ajalliset resurssit jokaiseen vaiheeseen. On myös huomattavaa, että jotkin työvaiheet saattavat painottua enemmän kuin toiset vaiheet ohjelmistokehitysprojektista riippuen. Näitä kehitysvaiheita Despan (2014, s. 39-40) mukaan ovat:

- Esitutkimus, jossa kerätään oleellista tietoa ja määritellään vaatimuksia ja päämääriä.
- Järjestelmäsuunnittelu, jossa määritellään ohjelmiston toiminta, jakaen se suunnittelukokonaisuuksiin, sekä tietokannan rakenne.
- Ohjelmistosuunnittelu, jossa suunnitellaan ohjelmiston ulkoasu joka voi vaihdella suuresti sen käyttötarkoituksesta riippuen.
- Kehittäminen, jossa toteutetaan ohjelmiston koodi ja jossa laaditaan kehittämis- ja testausympäristö.

- Testaaminen, jossa pyritään tunnistamaan suunnittelu- ja ohjelmointivirheet.
- Käyttöönotto, jossa ohjelmisto asennetaan käyttöympäristöön, sekä ylläpito, joka kattaa ohjelmiston lopun elinkaaren samalla varmistuen, että ohjelmisto toimii suunnitellusti.

Ohjelmistokehitys voidaan mukauttaa moneen erilaiseen ohjelmistokehitysmenetelmään, vaihdellen perinteisestä vesiputousmenetelmästä ketteriin menetelmiin (Alvertis ym., 2016, s. 75). Ohjelmistokehityksen menetelmät jakaantuvat siis perinteisiin malleihin, kuten vesiputousmalliin, joka on suhteellisen helppo toteuttaa ja ketteriin menetelmiin, kuten Scrum:iin, joka sopeutuu ympäristöön, jossa tapahtuu runsaasti muutoksia. Ketterät menetelmät perustuvat usein porrastettuun kehitykseen, mikä tarjoaa kehitykselle joustavuutta ja itseohjautuvuutta. (Despa, 2014, s.54.) Viimeisen vuosikymmenen aikana etenkin ketterät menetelmät ovat muodostuneet yhdeksi keskeisimmäksi ohjelmistokehityksen kehitysmalliksi, jossa pyritään lisäämään ohjelmistokehitysprosessin nopeutta sekä joustavuutta (Brhel, Meth, Maedche & Werder, 2015, s. 163). Jo mainittu, perinteinen ohjelmistokehitysmalli, vesiputousmalli, korostaa kattavaa järjestelmäsuunnittelua, jossa jokainen kehitysvaihe omaa omat suoritteet, siirtyen seuraavaan vaiheeseen vasta, kun edeltävä vaihe on päättynyt (Despa, 2014, s.41). Ketterässä Scrum-mallissa ohjelmistokehityksen vaiheet toteutetaan nopeina, toistuvina sprintteinä tietyn ajan sisällä, joiden lopputuloksina on toimiva ohjelmisto. Scrum-mallissa kehittäjätiimit ovat itseohjautuvia, joita ohjaa kehityksestä vastaavat Scrum-Masterit, jotka varmistavat, että jokaisen sprintin tavoitteet täyttyvät. (Despa, 2014, s. 45.) Usein vain yksi kehitysmalli lukuisista vaihtoehdoista ei sovellu täydellisesti tiettyyn ohjelmistokehitysprojektiin (Despa, 2014, s. 54).

Kaikissa ohjelmistokehitysmenetelmissä on omat etunsa ja haittansa. Perinteiset menetelmät siirtyvät tyypillisesti vaiheesta vaiheeseen, joten uusia, kehityksen aikana syntyneitä vaatimuksia, ei voida ottaa mukaan kehitykseen. Ketterät menetelmät eivät tyypillisesti huomioi kehityksen ulkopuolisten tahojen, kuten käyttäjien osallistumista, sillä ketterien menetelmien viestintä rajoittuu tavallisesti kehittäjien, hallinnoitsijoiden sekä sidosryhmien välille. (Franken, Kolvenbach, Prinz, Alvertis & Koussouris, 2015, s. 189.) Toisaalta, ketterillä menetelmillä saadaan hyviä tuloksia, kun yhden kehittäjätiimin jäsenet tekevät yhteistyötä toisten kehittäjätiimien kanssa (Papadopoulos, 2015, s. 458). Ketterät menetelmät käsittävät ohjelmistokehityksen toimenpiteet dynaamisina prosesseina, joissa sidosryhmät ovat aktiivisesti mukana ja kehitysvaiheet etenevät toistuvissa sykleissä. Ketterät menetelmät pystyvät myös vastaamaan ennakoimattomiin muutoksiin paremmin, sillä menetelmät ovat riippuvaisia tekijöistä ja heidän luovuudestaan jäykkien prosessien sijaan. (Brhel, Meth, Maedche & Werder, 2015, s. 164.) Perinteiset menetelmät ovat joustamattomia, eivätkä pysty vastaamaan nopeisiin käyttäjävaatimuksiin. Ketterissä menetelmissä taas voidaan sopeutua kehitysympäristöön, jossa muutoksia tulee myös kehityksen loppuvaiheissa, soveltuen mainiosti nykyiseen ohjelmistokehitysympäristöön,

antaen mahdollisuuden huomioida käyttäjiltä kerättyä dataa kehitysvaiheissa. (Papadopoulos, 2015, s. 455-456.) Ketterien menetelmien ohjelmistokehityksessä kehitysvaiheiden ja onnistumisen arviointi tapahtuu nopeissa sykleissä kehityksen mukana, jotta ohjelmistoa voitaisiin jatkuvasti parantaa, varmistaen, että ohjelmisto täyttää niin monen käyttäjän odotukset kuin mahdollista (Alvertis ym., 2016, s. 74).

Ohjelmistokehitys vaatii suuren määrän sekä teknisiä resursseja, että sidosryhmien tukea, jonka aikana käyttäjät sijoittuvat markkina-alueelle kehityksen näkökulmasta. Ohjelmistokehityksen menestyminen riippuu siitä, miten hyvin ohjelmisto kykenee vastaamaan käyttäjien odotuksiin (Alvertis ym., 2016, s. 73), joten kehityksen hallinnoijan tulee olla ajan tasalla uusimmista kehityksistä toimialalla, jotta ohjelmisto kykenee vastaamaan käyttäjien ajankohtaisiin vaatimuksiin (Despa, 2014, s. 38). Ohjelmiston mukauttaminen sidosryhmien ja käyttäjien odotusten mukaisiksi on tärkeää, jotta kehitysvaiheissa säästyttäisiin turhilta kuluilta ja kehityksen toteuttaminen nopeutuisi (Alvertis ym., 2016, s. 74). Koska monet ohjelmistokehitysmenetelmät harjoittavat ketteriä kehitysmenetelmiä ja kartoittavat jatkuvasti muuttuvia käyttäjävaatimuksia, jatkuvat muutokset ja mahdolliset ennalta arvaamattomat haasteet saattavat tulla esille milloin tahansa kehityksen aikana. Tämän takia kehitystiimien on oltava jatkuvasti valmiina vastaamaan muutoksiin ja uusiin käyttäjävaatimuksiin. (Alvertis ym., 2016, s. 75.) Vikojen ja puutteiden korjaaminen ohjelmistosta tulee kalliimmaksi mitä myöhemmin ne korjataan (Alvertis ym., 2016, s. 73). Organisaation antamaa jatkuvaa tukea ei siis tule väheksyä, sillä strateginen tukeminen ja yhteistyöhakuinen organisatorinen kulttuuri tukevat ketterien menetelmien toimivuutta, jotta kehittäjien luovuus, oma-aloitteisuus sekä nopeus kehittyisivät (Brhel, Meth, Maedche & Werder, 2015, s. 177). Ohjelmistokehitys voi yhdistää asiakkaita, sidosryhmiä sekä kehittäjiä, mikä mahdollistaa innovatiivisen toiminnan ohjelmistokehityksessä (Lesser & Ban, 2016, s. 42).

3.2 Pelinkehitys ohjelmistokehityksenä

Teknologian nopean kehityksen saatossa ohjelmistojen merkitys arjessa, esimerkiksi työnteon, vapaa-ajan sekä opiskelemisen kannalta, on kasvanut. Teknologian kehittyessä myös pelit ovat yleistyneet arkielämässä; niiden saatavuus on parantunut ja pelien kohdeikäikäyhtäjäien käsite on laajentunut kattamaan runsaasti erilaisia käyttäjiä. (Aleem, Capretz & Ahmed, 2016, s. 1.) Digitaaliset pelit muodostavatkin merkittävän kokonaisuuden ohjelmistokehityksessä (Murphy-Hill, Zimmermann & Nagappan, 2014, s. 1), jonka toimialalla pelit ovat alkaneet saavuttaa merkityksellisyyttä myös siten, että niiden käyttötarkoitukset eivät rajaudu pelkästään viihteeseen, vaan niitä voidaan yhdistää myös muihinkin käyttötarkoituksiin (Aleem, Capretz & Ahmed, 2016, s. 4). Ohjelmistona pelejä voidaan soveltaa eri alueille, kuten koulutukseen, liiketoimintaan sekä terveydenhuoltoon (Aleem, Capretz & Ahmed, 2016, s. 1). Yhtenä keinona pelien ominaisuuksien soveltamisessa eri aloille on hyödyntää ohjelmiston pelillistä-

mistä. Ohjelmiston pelillistäminen merkitsee pelisuunnittelun valikoitujen osien hyödyntämistä ei-pelillisessä ohjelmistossa, jolla pyritään motivoimaan käyttäjiä sekä parantamaan käyttökokemusta, hyödyntämällä esimerkiksi pisteitä ja muita palkitsevia toimintoja. (Liu, Santhanam & Webster, 2017, s. 1012-1013.) Pelit synnyttävät tyypillisesti vahvoja tunnereaktioita käyttäjissä, joista monet käyttäjät ovat valmiina maksamaan uusien tunne-elämysten ja kokemusten etsinnässä (Kuikkaniemi, Laitinen, Turpeinen, Saari, Kosunen, Ravaja, 2010, s. 859).

Pelit ovat monimuotoisia, sisältörikkaita sekä usein laajoja ohjelmistoja (Scacchi, 2017, s. 110). Peli on ohjelmisto, jonka käyttötarkoituksena voi olla sen käyttäjien viihdyttäminen. Pelien kehittäminen kohtaisi kuitenkin lukuisia haasteita, mikäli kehitysprosessit seuraisivat pelkästään muiden ohjelmistokehitysten käyttämiä käytänteitä. (Aleem, Capretz & Ahmed, 2016, s. 2.) Ohjelmistokehityksenä pelinkehitys on hyvin monialaista toimintaa, jossa ohjelmoijat, suunnittelijat, taiteilijat sekä hallinnolliset tahot tekevät yhteistyötä pelin kehityksen aikana. Jokaisella näistä toimijoista on oman alansa termistö, joten kehittäjiimien välinen viestintä saattaa muodostua haasteelliseksi ilman yhteisiä standardeja, itse pelejä. (Whitson, 2017, s. 4.) Ohjelmistona peli voidaan Peuronin (2017, s. 10) mukaan erotella kolmeen osaan:

- Pelikohtaisiin ohjelmiston osiin, jotka ovat jokaiselle pelille ominaisia piirteitä.
- Pelimoottoriin, ohjelmistoon, jonka avulla peli kootaan.
- Kolmannen osapuolen kirjastoihin ja ohjelmistoihin, jotka ovat resursseja joita voidaan hyödyntää pelimoottorissa.

Ohjelmistokehityksen näkökulmasta pelinkehityksellä on monta eroavaisuutta muuhun ohjelmistokehitykseen verrattuna. Pelinkehityksen kohdekäyttäjät huomioiden peliohjelmistot kehitetään perinteisesti viihdetarkoitukseen sekä ennalta määrätyn käyttäjäkokemuksen luomiseen. Pelinkehitysprojekteissa vaaditaan usein myös luovia menetelmiä, sillä käyttäjien vaatimuksia ei ole tyypillisesti kartoitettu kovin kattavasti ennen kehitysprosessin alkamista, mikä voi johtaa muutoksiin jopa myöhäisissä kehityksen vaiheissa sitä mukaan, kun vaatimuksia ilmenee. (Kasurinen, Palacin-Silva & Vanhala, 2017, s. 15.) Kun muutoksia täytyy tehdä nopeasti lähes koko kehitysprojektin ajan, pelinkehitykset hyötyvät suuresti ketteristä kehitysmenetelmistä (Murphy-Hill, Zimmermann & Nagappan, 2014 s. 6). Kasurinen, Palacin-Silva ja Vanhala (2017, s. 15) korostavat, että pelinkehityksessä hyödynnetään enemmän ketteriä kehitysmenetelmiä kuin muussa ohjelmistokehityksessä. Muita eroja ovat myös pelinkehitykseen kohdistuneiden vaatimusten subjektiivisuus sekä ohjelmiston toimintatehokkuuden korostaminen ylläpidon kustannuksella. Käyttäjät lopettavat pelaamisen usein jo yhden epätäydellisen kokemuksen jälkeen, mutta pelien puutteelliset toiminnot ja viat eivät tyypillisesti johda yhtä vakaviin seuraamuksiin muihin ohjelmistojen viallisiin toimintoihin verrattuna. Esimerkiksi, mikäli sähköpostipalvelussa tapahtuu toimintahäiriö, eikä työntekijän sähköposti saavu koskaan esimiehelle, työntekijän työsopimus saattaa katketa. (Mur-

phy-Hill, Zimmermann & Nagappan, 2014, s. 4.) Pelinkehityksessä on myös tavanomaista, että ohjelmiston testauksessa ja laadunvarmistuksessa hyödynnetään lähestymistapaa, jossa käytetään automatisoitujen testausmenetelmien sijaan kehitystiimin ulkopuolista testaajahenkilöstöä. (Washburn Jr, Sathiyarayanan, Nagappan, Zimmermann & Bird, 2016, s. 1.) Automatisoitujen testausten soveltaminen pelinkehityksessä on haastavaa esimerkiksi oikean toiminnan määrittämisessä pelikohtaisesti sekä käyttäjän, että itse ohjelmiston kannalta (Murphy-Hill, Zimmermann & Nagappan, 2014, s. 5).

Merkittäviä eroja muuhun ohjelmistokehitykseen verrattuna ovat myös lukuisten kehitysvälineiden luominen kutakin kehitysprojektia kohti sekä kehitysprojektin hyvin joustamattomat takarajat (Washburn Jr, Sathiyarayanan, Nagappan, Zimmermann & Bird, 2016, s. 1). Myös kokoonpanon hallinnan merkitys korostuu pelinkehityksessä, sillä muihin ohjelmistokehityksiin verrattuna pelinkehitysprojekteissa saatetaan hyödyntää huomattavasti enemmän ohjelmiston sisällöllisiä resursseja, peliassetteja, mikä lisää kokoonpanon hallinnan haastavuutta sekä testaamisen merkityksen kasvamista (Murphy-Hill, Zimmermann & Nagappan, 2014, s. 6). Ennalta määrättyjen vaatimusten puute, niiden usein tunteisiin pohjautuva subjektiivisuus, sisällöllisten resurssien hallinta ja testausten laatimisen haasteet saattavat tehdä pelinkehityksestä ennalta-arvaamattoman prosessin (Kasurinen, Palacin-Silva & Vanhala, 2017, s. 16). Vaikka pelinkehityksessä on paljon ominaisuuksia, joita ei tavallisesti kohdattaisi muussa ohjelmistokehityksessä, pelinkehitys hyötyy ohjelmistokehityksen tekniikoiden ja menetelmien soveltamisesta muun muassa ylläpidon, kehityksen joustavuuden, kustannusten hallinnoinnin sekä paremman suunnittelun tavoittelussa (Aleem, Capretz & Ahmed, 2016, s. 1). Esimerkiksi pelien ylläpitäminen alkaa muistuttaa muiden ohjelmistojen ylläpitoa hyödyntämällä pilvipalveluita, kuten Steam:ia (Murphy-Hill, Zimmermann & Nagappan, 2014, s. 6). Toisaalta hyväksi todetuista ohjelmistokehityksen käytänteistä on hyötyä pelinkehitykselle vain, jos pelinkehittäjät ymmärtävät nämä ohjelmistokehityksen käytänteet tarpeeksi kattavasti (Aleem, Capretz & Ahmed, 2016, s. 2).

Yksi ratkaisevista tekijöistä pelin toiminnassa on sen pelattavuus, minkä arvioiminen ei ole suoraviivainen prosessi. Tarkastelemalla käyttäjien saavuttamaan flow-tilaa, optimaallista kokemusta, jossa pelin vaikeus ja helppokäyttöisyys ovat täydellisessä tasapainossa pelin käytön aikana, kehittäjät voivat sisäistää käyttäjän näkökulmia pelisuunnitteluun sekä arvioida käyttäjien saamaa nautintoa pelin käyttämisestä. Flow-tila on kuitenkin käsitteenä hyvin subjektiivinen, jota on vaikea arvioida, ja sitä on haastavaa soveltaa kaikkiin mahdollisiin käyttäjiin heidän keskinäisten eroavaisuuksien takia. (McAllister & White, 2015, s. 13.) Pelaajia voidaan luokitella heidän toimintojensa avulla esimerkiksi saavuttajiksi, jotka hakevat haasteita, tutkijoiksi, jotka keräävät tietoa ja kokeilevat runsaasti erilaisia toimintoja, seurustelijoiksi, jotka etsivät vuorovaikutussuhteita muihin pelaajiin sekä tappajiksi, jotka haluavat päihittää muita pelaajia (Liu, Santhanam & Webster, 2017, s. 1020). Muodostamalla tasapainoisen yhdistelmän pelin ominaisuuksista ja sen tarjoamasta haasteesta käyttäjä tavanomaisesti kokee pelin hyväksi (Kuikkaniemi ym., 2010, s. 867).

Käyttäjien kokemukset voivat erota suuresti toisistaan pelin käytön aikana verrattuna muiden ohjelmistojen käyttökokemuksiin (Murphy-Hill, Zimmermann & Nagappan, 2014, s. 4), joten pelinkehittäjät pyrkivät luomaan vahvoja yhteyksiä käyttäjiin varmistaakseen pelin viihdyttävyyden ja kehittääkseen sitoututtavan pelin, jonka käyttökokemus säilyttäisi pelaajan sitoutumisen pelin käyttöön jopa useiden kuukausien ajan pelikohtaisesti (Murphy-Hill, Zimmermann & Nagappan, 2014, s. 7). Pelinkehittäjien tulee ymmärtää käyttäjien eitoiminnallisia vaatimuksia pelattavuuteen kohdistuvien vaatimusten lisäksi esimerkiksi viihteellisyyteen liittyen, mikä on käsitteenä subjektiivinen ja täysin riippuvainen käyttäjästä (Aleem, Capretz & Ahmed, 2016, s. 18). Pelien monimuotoisuudet tuovat haasteita ohjelmistokehitykseen, sillä ne yhdistävät luovia toimialoja, kuten suunnittelua, taidetta sekä äänituotantoa, teknologiseen ja toiminnalliseen toteutukseen (Aleem, Capretz & Ahmed, 2016, s. 2). Peliohjelmiston kehityksen aikana tulee myös muodostua käyttäjien ja kehittäjien välistä yhteistyötä, jotta kehittäjät kykenevät vastaamaan käyttäjien odotuksiin käyttäjien palautteen avulla (Whitson, 2017, s. 8).

3.3 Pelinkehityksen vaiheet

Pelinkehityksen toimialalle on ominaista, että tiettyjä kehitysvaiheita saatetaan painottaa muita vaiheita enemmän kehitysprosessista riippuen (McAllister & White, 2015, s. 14). Pelinkehittäjien tulee myös pohtia kehittymiseen liittyviä tekijöitä, kuten aikatauluja, resurssien jakamista sekä tiimien kouluttamista. Kehitysprojektin aikatauluttamista koskien pelinkehittäjien tulisi keskittää huomio kokonaisvaltaisen työtaakan arviointiin, jotta mikään kehityksen kannalta kriittinen tehtävä ei jäisi huomioimatta, kun arvioidaan realistisia ajallisia resursseja (Washburn Jr, Sathiyarayanan, Nagappan, Zimmermann & Bird, 2016, s. 8). Ajallisia ja liiketoiminnallisia kuluja voitaisiin pienentää käyttämällä uudestaan aiemmissa kehitysprojekteissa hyödynnettyjä ohjelmistoja, kuten pelimoottoreita. Kuitenkin pelinkehitysprojektin tarpeiden tulisi tällöin vastata täydellisesti siihen, mitä aiemmin käytetyt ohjelmistot kykenevät tarjoamaan. (Aleem, Capretz & Ahmed, 2016, s. 19.) Ohjelmistojen uudelleenkäyttöä ei juurikaan harjoiteta pelinkehityksessä, sillä pelit korostavat peliohjelmiston toimintatehokkuutta sekä muokattavuutta, jota aiemmin käytetyt ohjelmistot eivät välttämättä pysty tarjoamaan (Murphy-Hill, Zimmermann & Nagappan, 2014, s. 4). Etenkin pelimoottorit omaavat usein myös tunnistettavia piirteitä, kuten ohjattavuuksien eroja, joten käyttäjät voivat kokea pelin johdannaiseksi vanhemmasta pelistä omaperäisen pelin sijaan (Scacchi, 2017, s. 115).

Aikaa tulisi käyttää myös pelinkehitystiimiläisten kouluttamiseen, mikäli kehittäjien osaamisissa on puutteita, sekä huolehtia motivaation ylläpidosta (Washburn Jr, Sathiyarayanan, Nagappan, Zimmermann & Bird, 2016, s. 7). Kehitystiimien koolla on myös merkitystä pelinkehitykselle ja sen resurssien käytölle, sillä erikokoiset kehitystiimit voivat sijoittaa eri resursseja, kuten aikaa, eri osiin kehitystä pelinkehityksen aikana (Washburn Jr, Sathiyarayanan, Na-

gappan, Zimmermann & Bird, 2016, s. 9). Pelinkehityksen kehittäjätiimien koko vaihtelee tyypillisesti hyvin paljon yhden henkilön kehitysprosessista kansainvälisiin suuryrityksiin. Pelinkehityksen toiminnot kuitenkin pysyvät samankaltaisina tiimien koosta huolimatta. Kehitystiimiin osallistuu tyypillisesti ohjelmoijia, taiteilijoita, pelisuunnittelijoita, tuottajia sekä muuta henkilökuntaa. Ohjelmoijat toteuttavat peliohjelmiston ja vastaavat kehitystyökaluista hyödyntämisestä sekä mahdollisesti niiden kehittamisestä tai muokkaamisesta kehitykselle sopiviksi. Taiteilijat vastaavat pelin audiovisuaalisista sisällöistä, kuten musiikista, ääniefekteistä sekä graafisesta ilmeestä. Pelisuunnittelijat käsittelevät käyttökokemuksen edistämistä, pelin mahdollisen narratiivin ilmaisua sekä kenttäsuunnittelua. Tuottajat hallinnoivat kehitysprojektia ja laadunvarmistusta. Muuhun henkilökuntaan kuuluvat muun muassa testaajat, markkinoijat sekä julkaisijat. (Peuron, 2017, s. 19.)

Pelinkehityksen eri vaiheisiin kuuluu McAllisterin ja Whiten (2015) mukaan luonnosteluvaihe, alkumuotoilu, esituotanto, tuotanto sekä erilaiset testausvaiheet ennen pelin jakelua käyttäjille. Luonnosteluvaiheessa, eli konseptoinnissa liikkeelle laittava tekijä voi olla joko julkaisija, joka tarjoaa taloudellisia resursseja pelinkehitykselle, tai itse kehittäjätiimi, joka vastaa pelinkehityksen päivittäisestä kehittämisestä. Kun konseptista on sovittu kehittäjien sekä eri sidosryhmien kanssa, kehittäjät laativat alustavan pelisuunnitteludokumentin sekä visuaalisia esityksiä mahdollisesta pelin sisällöstä pelin näkemyksen viestimiseksi. Konseptoinnin aikana kehitykselle laaditaan myös budjetointia sekä erilaisia virstanpylväitä, mutta kaikki tässä vaiheessa muodostetut ehdot pelinkehitystä koskien ovat hyvin joustavia tuleville muutoksille. (McAllister & White, 2015, s. 14.) Pelinkehittäjien tulisi myös tiedostaa pelin toteuttamisen menetelmät konseptoinnin aikana, jotta pelisuunnittelun kannalta keskeiset ominaisuudet voidaan varmuudella toteuttaa pelinkehityksessä, tai muuttaa ominaisuuksia toteutettaviksi (Washburn Jr, Sathiyarayanan, Nagappan, Zimmermann & Bird, 2016, s. 9).

Alkumuotoilussa, pelinkehityksen ollessa vielä hyvin varhaisessa vaiheessa, pelin eri osista voidaan muodostaa prototyyppisiä, joilla voidaan tuoda esille esimerkkejä pelin eri ominaisuuksista, kuten pelattavuudesta ja pelin visuaalisesta ilmeestä. Kehityksen seuraava vaiheessa, esituotannossa, pelinkehittäjät kehittävät keskeisiä toimintatapoja, mekaniikkoja, sekä pyrkivät tunnistamaan ongelmallisia kohteita pelinkehitykselle. Esituotannon tarkoituksena on ideoida ja kehittää peliohjelmistoa nopeasti, jotta mahdollisimman monet riskit tunnistettaisiin ajoissa ja pelin keskeisimmät ominaisuudet tulisivat kehittäjille tunnetuiksi. (McAllister & White, 2015, s. 15.) Riskien hallinta on projektinhallinnan näkökulmasta hyvin tärkeä tekijä kehityksen aikana, jossa kehittämisstrategian, kehittäjien omaperäisyyden, aikataulutusten sekä budjetoinnin asemat korostuvat (Aleem, Capretz & Ahmed, 2016, s. 20). Seuraavassa vaiheessa eli tuotannossa, jonka aikana varsinainen pelinkehitys pääsee käynnistymään, kehittäjät tuottavat sekä sisällöllisiä että toiminnallisia ominaisuuksia. Tuotantovaiheessa liitetään myös laadunvalvonta kiinteäksi kehityksen ominaisuudeksi, joka voi koostua erilaisista testaajista sekä markkinoinnin asiantuntijoista. Testaajat voi-

vat havaita ohjelmistovirheiden lisäksi pelattavuutta koskevia toiminnallisia sekä sisällöllisiä haittoja, jotka vaikuttavat pelattavuuteen tai käyttäjäkokemukseen. Mitä pidemmälle tuotantovaihe jatkuu, sitä enemmän henkilöstöä käytetään testaamiseen ja laadunvalvontaan. (McAllister & White, 2015, s. 15.) Pelinkehityksen aikana projektinhallinnan merkitys kasvaa, jotta esimerkiksi ristiriitoja saataisiin vähennettyä (Washburn Jr, Sathiyarayanan, Nagappan, Zimmermann & Bird, 2016, s. 8).

Kehityksen loppua kohti peli läpäisee erilaisia vaihteita, jotka ovat alpha, beta ja kulta, joiden avulla määritellään, kuinka valmis peli on julkaistavaksi. Jotta peli saavuttaa alpha-vaiheen, kaiken pelisisällön tulee olla peliohjelmistossa, muttei välttämättä viimeisteltynä, ja pelin tulee toimia kokonaisuutena. Beta-vaiheeseen mennessä sisältöjen sekä ominaisuuksien tulee olla viimeistelyjä, muttei loppuun hiottuja, eikä peliohjelmistoon enää lisätä ylimääräisiä sisältöjä. Beta-vaiheessa voidaan tavanomaisesti myös keskittyä pelkästään ohjelmistovirheiden havaitsemiseen. (McAllister & White, 2015, s. 16.) Kuten Murphy-Hill, Zimmermann ja Nagappan (2014, s. 5) jo aiemmin mainitsivat ohjelmisto- ja pelinkehityksen erojen määrittelyssä, pelinkehittäjät ovat haluttomia käyttämään automaattisia testausmenetelmiä, koska pelinkehittäjien nopeat toiminnot tekevät automatisoiduista menetelmistä nopeasti vanhenevia, eivätkä pysty vastaamaan ajantasaiseen testaamisen tarpeeseen (Murphy-Hill, Zimmermann & Nagappan, 2014, s. 1). Beta -vaiheen testauksissa hyödynnetäänkin kehittäjatiimin ulkoisia testaajia, joiden avulla pyritään havaitsemaan ohjelmistovirheiden lisäksi käytettävyyden vikoja, jotka ovat jääneet kehittäjiltä huomaamatta (Aleem, Capretz & Ahmed, 2016, s. 24). Tämän vaiheen jälkeen kehittäjät ja julkaisijat kokevat pelin tavallisesti julkaisukelpoiseksi, ja antavat pelin lopulliselle julkaisijalle hyväksyttäväksi. Kun peli saa hyväksynnän, sen tuottaminen ja jakelu alkavat ja sille valitaan kiinteä julkaisupäivämäärä. (McAllister & White, 2015, s. 16.) Pelinkehittäjät voivat vaikuttaa toiminnoillaan lähes kaikkiin pelin kehityksen osiin, paitsi pelattavuuteen, johon vaikuttaa kaikkien toimintojen yhteinen toimivuus käyttäjän käyttäessä peliä (Peuron, 2017, s. 15). Pelinkehityksellä on käytössä runsaasti eri alustoja, joille peli voidaan kehittää, kuten lukuisat pelikonsolit, tietokoneet, mobiililaitteet sekä älylaitteet, mikä tarjoaa pelinkehitykselle mahdollisuuden kehittää peli ensin yhdelle alustalle, ja julkaista se myöhemmin toiselle, mikäli sille nähdään riittäviä liiketoiminnallisia mahdollisuuksia perusteeksi (Kasurinen, Palacin-Silva & Vanhala, 2017, s. 16).

4 KÄYTTÄJÄDATA PELINKEHITYKSESSÄ

Kilpailu yritysten kesken on merkittävä tekijä yritysten toiminnassa, joten tehokas pelinkehitys vaatii erilaisia työkaluja ja tekniikoita, joita on joko kehitetty pelinkehityksessä tai omaksuttu muilta IT-toimialoilta, esimerkiksi liiketoimintatavoista, projektin hallinnasta ja käyttäjien testaamisesta. Yksi näistä kilpailukykyä parantavista menetelmistä on analytiikka. (El-Nasr, Drachen & Canossa, 2016, s. 16.) Pelianalytiikka tarkoittaa analytiikan ja Big Datan sijoittamista pelaamisen kontekstiin, jota voidaan käyttää pelaajan pelikokemuksen parantamiseen tai pelin sisällä tehtyjen ostosten maksimoimiseen muokkaamalla tiettyjä pelin osia (Koskenvoima & Mäntymäki, 2015, s. 326). Analytiikan avulla pelinkehittäjät kykenevät ymmärtämään pelaajan näkökulmaa pelikokemukseen liittyen (Koskenvoima & Mäntymäki, 2015, s. 331). Termi pelianalytiikka tarkoittaa analytiikan soveltamista pelinkehitykseen ja tutkimukseen, jonka tarkoituksena on tukea päätöksentekoa operatiivisella, taktisella ja strategisella tasolla muodostaen keskeisen liiketoimintatiedon lähteen pelinkehitykselle. Pelianalytiikassa käytetyt mittarit toimivat tulkinnan välineinä dataa tarkastellessa. Yhtenä yleisimmistä datan lähteistä on käyttäjän käyttäytymisen data, joka voidaan tulkita esimerkiksi pelaamiseen käytettynä aikana sekä päivittäisten käyttäjien määränä. (El-Nasr, Drachen & Canossa, 2016, s. 6.) Pelien käyttäjille voidaan hahmottaa kaksi merkitystä sekä pelien pelaajina, että asiakkaina (El-Nasr, Drachen & Canossa, 2016, s. 15).

Tässä luvussa tarkastellaan lukujen 2 ja 3 pohjalta, miten käyttäjät dataa hyödynnetään pelinkehityksessä. Ensin tarkastellaan käyttäjätiedon koontimenetelmiä pelinkehitykseen sekä pelianalytiikkaa pelinkehityksen työvälineenä. Tämän jälkeen tarkastellaan käyttäjätiedon käyttötapoja pelinkehityksessä, jotta pelinkehittäjät kykenevät kehittämään pelejä, jotka eivät turhauta pelaajia liikaa ja kannustavat flow-tilan syntymiseen.

4.1 Käyttäjätiedon koontimenetelmiä pelinkehitykseen

Pelianalytiikka on kerännyt viime vuosina laajasti huomiota pelinkehityksen ja pelitutkimuksen osalta. Historiallisesti pelinkehitys ei ole ollut datavetoista, mutta muutoksia on tullut, kun data-analytiikan omaksumisen avulla on saatu hyötyjä päätöksenteon tueksi kaikilla pelituotannon toimialaan kuuluvissa osissa. (El-Nasr, Drachen & Canossa, 2016, s. 3.) Vaikka data-analytiikan käytännöille löytyy rooleja kaikilta peliyrityksen osa-alueilta, pelinkehityksessä analytiikalla pyritään saamaan parempia tietoja käyttäjistä – pelaajista. Tarve käyttäjätietojen hyödyntämiselle on korostunut nopean sosiaalisten verkkopelien ja Free-to-Play -liiketoimintamallien myötä, sillä ne nojaavat kattavan käyttäjätiedon analyysin ajamaan liikevaihtoon. Pelaajista on tullut kiinteämpi osa pelinkehitysprosessissa käyttäjätutkimusmenetelmien myötä, mikä perinteisten pelitestausten lisäksi hyödyntää palautetta välittömästi suunnitteluun. Pelianalytiikasta on myös tullut tärkeämpi osa toimialan liiketoimintatiedon hallintaa, kun määrällistä dataa on saatu päätöksenteon tueksi telemetriasta, markkinatiedotteista, laadunhallinnasta, suorituskykyjen testauksesta sekä monista muista lähteistä. (El-Nasr, Drachen & Canossa, 2016, s. 4.) Hallinnoinnin näkökulmasta verkkopelien analytiikan tarkastelulla voidaan ymmärtää vuorovaihtusmenetelmien parantamismahdollisuuksia, jotta pystytään tekemään parempia pelejä (Koskenvoima & Mäntymäki, 2015, s. 327).

Analytiikan menetelmien perusteet koostuvat tilastoista, datanlouhinnasta, matemaattisista menetelmistä, ohjelmoinnin ja toiminnan tutkimuksesta sekä datan visuaalisesta esityksestä, jotta tiedoista voidaan viestittää edelleen oleellisille sidosryhmille. Organisaatiossa analytiikka voi myös haarautua moneksi kokonaisuudeksi, kuten verkko-, riski- ja markkina-analyysiksi sekä tällä toimialalla pelianalyysiksi. Analytiikka onkin yleisnimitys, joka kattaa erilaiset menetelmät, joilla voidaan löytää ja viestiä erilaisista malleista, joita data muodostaa. Yrityksissä datan analytiikka muodostaa tärkeän osan liiketoimintatiedon hallintaa yrityksen koosta riippumatta. Yrityksen liiketoimintatiedon hallinnan tavoitteena on tuottaa kerätystä, raakasta datasta oleellista tietoa päätöksenteon tueksi. Pelituotannon toimialan yrityksissä liiketoimintatiedon hallinnan datalähteisiin kuuluu yrityksen peliohjelmistojen käyttäjät, pelien pelaajat, erilaisten käyttäjätestausten ja käyttäjätutkimusten kautta. (El-Nasr, Drachen & Canossa, 2016, s. 14.) Erilaisiin käyttäjätesteihin lukeutuu myös laadunvarmistustestaukset, joilla varmistetaan pelin audiovisuaalisen sisällön miellyttävyyttä käyttäjien näkökulmasta. Käyttäjiin kohdistetaan myös pelitestejä, joissa hyödynnetään käyttäjiä pelikokemuksen testaamisessa ja jolla pyritään löytämään ongelmia pelattavuuden kannalta. (Peuron, 2017, s. 19.)

Peleistä mitattua dataa voidaan käsitellä raakana datana, jota saadaan esimerkiksi tarkastelemalla, miten pelaaja toimii asiakasohjelman kautta, ohjelmistovikojen korjausten etenemisestä sekä liiketoimintatiedon verkkomaksusta. Asiakasohjelman kautta saadusta pelaajan toimintaan liittyvän datan keräämisen kohdalla asiakasohjelma välittää datan palvelimelle, josta data saadaan ke-

rättyä. (El-Nasr, Drachen & Canossa, 2016, s. 17.) Koskenvoiman ja Mäntymäen (2015, s. 328) mukaan erityisesti verkkopeleistä voidaan kerätä dataa lähes kaikista pelaajan toiminnoista pelissä, jota pelinkehittäjät voivat hyödyntää pelisuunnittelussa esimerkiksi etsiessään syitä pelaajien turhautumisille, mikä voi johtaa pelin pelaamisen vähentymiseen.

4.2 Käyttäjätiedon käyttötavat pelinkehityksessä

Käyttäjätiedon ja pelianalytiikkaa voidaan hyödyntää pelisuunnittelun tukena, pelin julkaisussa sekä julkaisijoiden ja tuottajien viestinnässä eri tavoin. Pelin toimittajat voivat myös myydä käyttäjistä kerättyä dataa edelleen kolmansille osapuolille. (Koskenvoima & Mäntymäki, 2015, s. 327.) Täydellisen hyödyn saaminen kerätystä datasta kuitenkin edellyttää sitä, että päätöksentekijät saavat tarvitsemansa datan vaivattomasti mitä voidaan helpottaa datan keskittämällä, jossa kerätty data kootaan yhteen tietokantaan (Vielma, 2017, s. 17).

Käyttäjätiedon tukeman peliprojektin suunnitteluvaiheessa ilmenevä ongelma huomataan mittareissa, mitä pelisuunnittelijat yrittävät ratkaista tutkimalla kerättyä dataa ja jonka johdosta laaditaan muutoksia peliin ja mitataan sen vaikutuksia (Koskenvoima & Mäntymäki, 2015, s. 332). Käyttämällä analytiikkaa viestintävälineenä pelin sijoittajat ja julkaisijat kykenevät seuraamaan ennalta määrättyjä mittareita, jotka antavat keskusteluille jaetun näkökulman. Analytiikka toimii myös päätöksenteon työkaluna, jolla pyritään tukemaan suunnittelun päätöksiä niiden ohjaamisen sijaan. Analytiikka vaikuttaa kaikkiin pelin kehitysvaiheisiin, joissa on jatkuva muutosten, arviointien ja kehittämisen kehä. (Koskenvoima & Mäntymäki, 2015, s. 333.) Pelinkehityksen aikana testejä tulisikin toteuttaa jo varhaisten prototyyppien tuotannon aikana, jotta testien tuloksia voitaisiin hyödyntää pelinkehityksessä niin pian kuin mahdollista (Dörner, Göbel, Effelsberg & Wiemeyer, 2016, s. 94).

Pelinkehityksen aikana pelaajista kerätty käyttäjätiedon, kuten kirjallinen palaute, auttaa pelinkehittäjiä arvioimaan pelaajien kokemaa käyttäjäkokemusta (O'Hagan, Coleman & O'Connor, 2014, s. 2). Pelianalytiikan ei kuitenkaan katsota olevan pelisuunnittelua ohjaava tekijä eikä siitä saada suoria vastauksia, mutta käyttäjätiedon analytiikalla voidaan saada tukea päätöksentekoon myös suunnitteluvaiheessa. Pelisuunnittelu saa tukea data-analytiikasta, ja mittauksen avulla voidaan tuottaa käyttäjille viihtyisämpiä pelejä. (Koskenvoima & Mäntymäki, 2015, s. 330.) Pelinkehittäjien tavoitteena on tavanomaisesti kehittää pelin pelaamisesta viihtyisä kokemus audiovisuaalisen ja narratiivisen sisällön sekä ohjelmiston toiminnan ominaisuuksien avulla (Dörner, Göbel, Effelsberg & Wiemeyer, 2016, s. 4-5).

Pelinkehittäjät tyypillisesti pyrkivät ylläpitämään pelaajan sitoutumista pelaamiseen, joten Hännisen (2017, s. 21) mukaan turhautumista halutaan välttää, sillä se johtaa usein pelaamisen keskeytymiseen, kun pelaaja ei kykene selvittämään pelin asettamia haasteita. Pelattavuuden kehittämisen haasteisiin liittyy myös eroavaisuudet pelaajien taitojen välillä: Mikäli peli ei tarjoa tar-

peeksi haasteita kokeneelle pelaajalle tai jos pelin haastetaso on liian korkea kokemattomalle pelaajalle, käyttäjä voi turhautua peliin (Dörner, Göbel, Effelsberg & Wiemeyer, 2016, s. 74). Pelin pelaamisen aikana voidaan valvoa turhautumisen syntymistä esimerkiksi ohjainten käytön tarkastelulla. Tarkastelun avulla pystytään arvioimaan tilanteita, joiden aikana pelaajat kykenevät selviytymään eri pelitilanteista omatoimisesti sekä havaitsemaan pelitilanteita, joiden aikana pelaajat turhautuvat ja siten hyötyisivät parannetusta pelattavuudesta. (Hänninen, 2017, s. 21.)

Pelianalytiikkaa voidaan käyttää pelinkehityksen työkaluna riskien vähentämisessä, etenkin kehitysvaiheessa analytiikan hyödyntäminen voi vähentää julkaisuvaiheen täydellisen epäonnistumisen riskiä (Koskenvoima & Mäntymäki, 2015, s. 334). Pelin menestys on myös riippuvainen sen käytön vaivattomasta omaksumisesta sekä pelattavuudesta, johon voidaan vaikuttaa ohjelmiston toimivuuden lisäksi pelin audiovisuaalisella sisällöllä ja pelille ominaisilla toiminoilla, jotka sitouttavat pelaajaa pelin käyttöön (Dörner, Göbel, Effelsberg & Wiemeyer, 2016, s. 38). Käyttäjät eivät usein kykene ilmaisemaan omia odotuksiaan peliä kohtaan selvästi ennen pelin käyttöä, mutta käyttökokemuksen jälkeen käyttäjät tyypillisesti osaavat sanoa vastaako peli käyttäjän odotuksiin ja vaatimuksiin. Tämän vuoksi pelinkehitysprosessit usein hyötyvät käyttäjädatabasta, jota saadaan pelien prototyyppien testauksista sekä käyttäjäpalautteesta. (Scacchi, 2017, s. 114.)

5 JOHTOPÄÄTÖKSET

Tässä tutkielmassa käsiteltiin käyttäjätietojen merkitystä sekä ohjelmisto- että pelinkehitykselle kirjallisuuskatsauksena, jossa pelinkehitystä tarkastellaan yhtenä ohjelmistokehityksen muotona, alustaen tutkimusta kartoittamalla käyttäjätietojen merkitystä ohjelmistokehitykseen. Tutkielman aineisto koostui tieteellisistä julkaisuista, jonka käytössä priorisoitiin vuoden 2010 jälkeisiä julkaisuja, jotta tutkielma hyötyisi mahdollisimman ajankohtaisesta aineistosta. Käyttäjätietojen tarkastelussa hyödynnettiin sekä ohjelmisto- että pelinkehityksen piirissä kahta tutkimuskysymystä, joista ensimmäinen kysymys tarkasteli käyttäjätietojen merkitystä ohjelmistokehityksessä. Toisessa tutkimuskysymyksessä tarkasteltiin, miten ohjelmistokehitys ilmenee pelinkehityksenä ja miten pelinkehitys hyötyy käyttäjätiedoista. Tutkielmassa tuotiin esille myös haasteita, joita liittyy käyttäjätietojen tietoturvaan, sekä ajankohtaista uutisointia ja tietosuojasetuksen tuomia muutoksia käyttäjätietojen käyttöön asetuksen tullessa voimaan 25.5.2018 korostaakseen tietoturvan merkitystä käyttäjätiedoille, vaikka tutkimus keskittyy ensisijaisesti käyttäjätietojen merkityksen ja käyttötapojen tarkasteluun.

Tuloksena ensimmäiseen tutkimuskysymykseen, ”Mikä merkitys käyttäjätiedoilla on ohjelmistokehityksessä?”, käyttäjätiedoille löytyi monia merkityksiä ohjelmistokehityksessä: Käyttäjätieto muodostaa tärkeän liiketoiminnallisen resurssin, koska käyttäjätietojen tarkastelun avulla voidaan löytää keinoja saavuttaa haluttuja käyttäjäkokemuksia ohjelmiston avulla, kartoittaa käyttäjien odotuksia ja vaatimuksia sekä huomata puutteita ohjelmistossa. Käyttäjätietoja syntyy monessa eri muodossa esimerkiksi Big Datana sekä kirjallisina palautteina, joten käyttäjätieto tulee analysoida, suodattaa sekä tiivistää oleelliseksi tiedoksi, jotta käyttäjätiedoista on hyötyä ohjelmistokehittäjille.

Toisen tutkimuskysymyksen ensimmäisen osan, ”Miten ohjelmistokehitys ilmenee pelinkehityksenä?”, avulla tarkasteltiin pelinkehitystä ohjelmistokehityksen muotona, tuoden esille jaettuja ominaisuuksia, sekä huomattavia eroja kehitysten välillä. Ohjelmistokehityksen kehittyessä saadaan uusia mahdollisuuksia jalostaa kehityksen menetelmiä, jossa pelinkehityksestä on muovautunut yksi merkittävimmistä alueista ohjelmistokehityksessä. Sekä ohjelmisto-

että pelinkehityksen aikana kehittäjien tulee kommunikoida keskenään kehityksen edistämiseksi, mikä korostuu etenkin monialaisissa pelinkehitysprojekteissa. Ohjelmistojen kehittäminen voidaan toteuttaa sekä ohjelmisto- että pelinkehityksessä erilaisilla vaiheilla, joita voidaan painottaa tai karsia projektista riippuen. Kuitenkin verrattuna ohjelmistokehityksen vaiheisiin pelinkehityksessä korostetaan erilaisia testauksia testaajahenkilöstöllä, jotta pelinkehittäjät kartoittaisivat subjektiivisten käyttäjävaatimusten toteuttamista. Ohjelmistokehitykset voivat noudattaa monia erilaisia kehitysmenetelmiä, kuten perinteisiin menetelmiin kuuluvaa vesiputousmenetelmää sekä erilaisia ketteriä menetelmiä, kuten Scrumia. Pelinkehityksessä hyödynnetään selvästi enemmän erilaisia ketteriä menetelmiä perinteisten sijaan, sillä pelinkehityksessä tulee huomioida erilaisten vaatimusten täyttäminen ja yllättäviin muutoksiin sopeutuminen. Muutoksia ja vaatimuksia saattaa ilmaantua kehityksen eri vaiheissa, mikä tekee perinteisistä menetelmistä sopimattomia pelinkehitykselle. Vaikka ohjelmisto- ja pelinkehityksissä on paljon yhteneväisyyksiä ja eroja, pelinkehitys hyötyy ohjelmistokehityksessä käytettyjen tekniikoiden ja menetelmien soveltamisesta, mutta toisaalta ohjelmistokehityksen käytänteistä on hyötyä pelinkehityksessä vain, kun käytänteet ymmärretään tarpeeksi kattavasti.

Toisen tutkimuskysymyksen toisen osan, ”Miten pelinkehitys hyötyy käyttäjädatasta?”, vastauksen kartoittamisella löytyi menetelmiä, miten pelinkehityksissä hyödytään käyttäjädatasta, samalla tukien vastauksia ensimmäiseen tutkimuskysymykseen, jossa käyttäjädataa tarkasteltiin ohjelmistokehityksessä. Käyttäjädatta tukee pelinkehitystä samoilla periaatteilla kuin ohjelmistokehitystä, mutta aineistosta ilmeni, että pelinkehityksessä hyödynnetään käyttäjädatan käsittelyssä pelianalytiikkaa, joka toimii käyttäjädatan tulkinnan välineenä. Käyttäjädatan avulla pelinkehityksessä voidaan huomata esimerkiksi keinoja parantaa ohjelmiston vuorovaikutusmenetelmiä ja käyttäjädatan analytiikkaa voidaan hyödyntää myös päätöksenteon tukena. Pelinkehityksen aikana tulee siis muodostua käyttäjien ja kehittäjien välistä yhteistyötä erilaisten menetelmien, kuten testaamisen, käytön, sekä kirjallisen palautteen avulla, jotta kehittäjät kykenevät vastaamaan käyttäjien erilaisiin, myös hyvin subjektiivisiin, odotuksiin, parantaen pelin toimivuutta ja lisäten liiketoiminnallisia tuottoja.

Aineiston keräämisessä ilmeni, että tieteellistä tutkimusta ei ole juuri toteutettu pelinkehityksen aiheesta verrattuna ohjelmistokehitystä käsittelevän aineiston saatavuuteen. Myös käyttäjädataa käsitteleviä aineistoja oli haastavaa löytää, mikä voi osin johtua siitä, että yritykset käsittelevät käyttäjädataa organisaation sisäisenä tietona, eivätkä mieluusti jaa tietojaan ulkopuolisille tahoille. Aineiston haussa vuosiluvun lisäksi yhtenä aineiston arviointikriteerinä oli myös aineistoon kohdistuvien viittausten määrä Google Scholarissa, mikä synnytti haasteita aineiston keräämisessä. Osa artikkeleista oli julkaistu vasta vuonna 2017, eivätkä omanneet kovin suurta viittauss määrää, vaikka niiden tarkastelun jälkeen ne osoittautuivat hyvin kootuiksi tieteellisiksi julkaisuiksi.

Tämän tutkielman pohjalta mahdolliseksi jatkotutkimusaiheeksi voisi kehittyä liiketoiminnallisen tiedon hallinta pelinkehityksessä, sekä data-analytiikan hyödyntäminen organisaatioiden päätöksenteossa, jossa perehdyt-

täisiin syvemmin käyttäjätietojen ja muun organisaation käyttöön syntyneen datan tarkasteluun päätöksenteon näkökulmasta. Koska nyky-yhteiskunta on teknologisesti saturoitunut ja teknologisen kehityksen myötä dataa alkaa syntyä lähes kaikista käyttäjän toiminnoista, kannattaisi tutkia, miten tätä dataa voidaan hyödyntää tehokkaasti ja käyttäjän tietoturvaa kunnioittaen myös pelinkehityksessä. Digitaaliset pelit ovat ohjelmistoina jatkuvasti läsnä ja niiden käyttötarkoitukset laajenevat viihteellisyyden ulkopuolelle kattamaan laajaa käyttäjäkuntaa, mikä kasvattaa pelinkehityksessä mahdollisesti hyödynnettävän käyttäjätietojen määrää ja monipuolisuutta.

LÄHTEET

- Aleem, S., Capretz, L. F., & Ahmed, F. (2016). Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development*, 4(1), 1-30.
- Alpaydin, E. (2016). *Machine learning: the new AI*. Cambridge, MA: MIT Press.
- Alvertis, I., Koussouris, S., Papaspyros, D., Arvanitakis, E., Mouzakitis, S., Franken, S., Kolvenbach, S., Prinz, W. (2016). User Involvement in Software Development Processes. *Procedia Computer Science*, 97, 73-83.
- Brhel, M., Meth, H., Maedche, A., & Werder, K. (2015). Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology*, 61, 163-181.
- Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: from big data to big impact. *MIS quarterly*, 36(4), 1165-1188.
- Despa, M. L. (2014). Comparative study on software development methodologies. *Database Systems Journal*, 5(3), 37-56.
- Dörner, R., Göbel, S., Effelsberg, W., & Wiemeyer, J. (Eds.). (2016). *Serious games: foundations, concepts and practice*. Switzerland: Springer International Publishing.
- El-Nasr, M. S., Drachen, A. & Canossa, A. (2016). *Game analytics*. London: Springer.
- Franken, S., Kolvenbach, S., Prinz, W., Alvertis, I., & Koussouris, S. (2015). CloudTeams: Bridging the gap between developers and customers during software development processes. *Procedia Computer Science*, 68, 188-195.
- Fuggetta, A., & Di Nitto, E. (2014). Software process. In *Proceedings of the on Future of Software Engineering* (s. 1-12). ACM.
- Goode, S., Hoehle, H., Venkatesh, V., & Brown, S. A. (2017). User compensation as a data breach recovery action: an investigation of the Sony Playstation Network breach. *MIS Quarterly*, 41(3), 703-728.
- Greengard, S. (2015). *The internet of things*. Cambridge, MA: MIT Press.
- Hartikainen, J. (2018). Kriisi osuu Facebookin ytimeen – yhtiö menestyy huipputarkoilla tiedoilla ihmisistä, mutta nyt käyttäjät ja viranomaiset saatta-

- vat painaa jarrua. *Helsingin Sanomat* (2018, 24. maaliskuuta). Viitattu 10.4.2018 <https://www.hs.fi/talous/art-2000005616319.html>
- Hänninen, J. (2017). Coping to play: the effect of user-driven innovations on user experience in games (Pro gradu -tutkielma) Jyväskylän yliopisto. Haettu osoitteesta <http://urn.fi/URN:NBN:fi:jyu-201704262087>
- Joachims, T., Granka, L., Pan, B., Hembrooke, H., & Gay, G. (2017). Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Forum*, 51(1), 4-11. ACM.
- Kasurinen, J., Palacin-Silva, M., & Vanhala, E. (2017). What Concerns Game Developers? A Study on Game Development Processes, Sustainability and Metrics. In *Emerging Trends in Software Metrics (WETSOM), 2017 IEEE/ACM 8th Workshop* (s. 15-21). IEEE.
- Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2016). The emerging role of data scientists on software development teams. In *Proceedings of the 38th International Conference on Software Engineering* (s. 96-107). ACM.
- Koskenvoima, A., & Mäntymäki, M. (2015). Why do small and medium-size freemium game developers use game analytics?. In *Conference on e-Business, e-Services and e-Society* (s. 326-337). Springer, Cham.
- Kuikkaniemi, K., Laitinen, T., Turpeinen, M., Saari, T., Kosunen, I., & Ravaja, N. (2010). The influence of implicit and explicit biofeedback in first-person shooter games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (s. 859-868). ACM.
- Lesser, E., & Ban, L. (2016). How leading companies practice software development and delivery to achieve a competitive edge. *Strategy & Leadership*, 44(1), 41-47.
- Liu, D., Santhanam, R., & Webster, J. (2017). Toward meaningful engagement: A framework for design and research of gamified information systems. *MIS quarterly*, 41(4), 1011-1034.
- Maalej, W., Nayebi, M., Johann, T., & Ruhe, G. (2016). Toward data-driven requirements engineering. *IEEE Software*, 33(1), 48-54.
- McAllister, G., & White, G. R. (2015). Video game development and user experience. In *Game User Experience Evaluation* (s. 11-35). Springer, Cham.
- McCormick, T. H., Lee, H., Cesare, N., Shojaie, A., & Spiro, E. S. (2017). Using Twitter for demographic and social science research: tools for data collection and processing. *Sociological methods & research*, 46(3), 390-421.

- Murphy-Hill, E., Zimmermann, T., & Nagappan, N. (2014). Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development?. In *Proceedings of the 36th International Conference on Software Engineering* (s. 1-11). ACM.
- Mäyrä, F. (2008). *An introduction to game studies*. London: Sage.
- O'Hagan, A. O., Coleman, G., & O'Connor, R. V. (2014). Software development processes for games: a systematic literature review. In *European Conference on Software Process Improvement* (s. 182-193). Springer, Berlin, Heidelberg.
- Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C. A., Canfora, G., & Gall, H. C. (2015). How can I improve my app? Classifying user reviews for software maintenance and evolution. In *Software maintenance and evolution (ICSME), 2015 IEEE international conference* (s. 281-290). IEEE.
- Papadopoulos, G. (2015). Moving from traditional to agile software development methodologies also on large, distributed projects. *Procedia-Social and Behavioral Sciences*, 175, 455-463.
- Peischl, B., Ferk, M., & Holzinger, A. (2015). The fine art of user-centered software development. *Software Quality Journal*, 23(3), 509-536.
- Peuron, I. (2017). Ohjelmistoarkkitehtuurit peleissä: systemaattinen kirjallisuuskatsaus (Pro gradu -tutkielma) Jyväskylän yliopisto. Haettu osoitteesta <http://urn.fi/URN:NBN:fi:jyu-201706263046>
- Salen, K., & Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. Cambridge, MA: MIT press.
- Scacchi, W. (2017). Practices and Technologies in computer game software engineering. *IEEE Software*, 34(1), 110-116.
- Talus, A., Autio, E., Hänninen, A., Pihamaa, H. T., & Kantonen, S. (2017). *Miten valmistautua EU: n tietosuoja-asetukseen?*. Helsinki: Oikeusministeriö.
- Vielma, S. (2017). Massadata urheiluanalytiikan apuna (Kandidaatin tutkielma) Jyväskylän yliopisto. Haettu osoitteesta <http://urn.fi/URN:NBN:fi:jyu-201712204822>.
- Washburn Jr, M., Sathiyarayanan, P., Nagappan, M., Zimmermann, T., & Bird, C. (2016). What went right and what went wrong: an analysis of 155 postmortems from game development. In *Proceedings of the 38th International Conference on Software Engineering Companion* (s. 1-10). ACM.
- Whitson, J. R. (2017). Voodoo software and boundary objects in game development: How developers collaborate and conflict with game engines and

art tools. *New Media & Society*, 30.06.2017. Haettu osoitteesta
<http://journals.sagepub.com/doi/abs/10.1177/1461444817715020>