

Jussi Koutonen

**KETTERÄT MENETELMÄT PELITEOLLISUUDESSA:
SUOMALAISIA PELISTUDIOITA KOSKEVA KYSELY-
TUTKIMUS**



JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIEDEIDEN LAITOS

2011

TIIVISTELMÄ

Koutonen, Jussi

Ketterät menetelmät peliteollisuudessa: suomalaisia pelistudioita koskeva kyselytutkimus

Jyväskylä: Jyväskylän yliopisto, 2011, 129 s.

Tietojärjestelmätiede, pro gradu

Ohjaaja(t): Leppänen, Mauri

Peliteollisuus on erittäin kilpailtu ala, jossa yksittäinen epäonnistuminen voi johtaa jopa konkurssiin. Markkinat vaativat yhä kiinnostavampia ja hauskempia pelejä, joiden tulee hyödyntää nopeasti kehittyvän teknologian tarjoamia mahdollisuuksia. Laadullisten haasteiden ja kehittyvän teknologian ohella monen alan toimijoiden mukana olo tekee pelien kehittämisestä erittäin haastavaa. Pelistudion kehitysprosessin tulee olla huippuluokkaa. Pelikehitysprosessi on luonteeltaan iteratiivinen, ja pelistudiot hyödyntävät siinä enenevässä määrin ketteriä menetelmiä.

Tämän tutkimuksen tarkoituksena on valottaa pelien, peliteollisuuden, pelikehityksen ja ketterien menetelmien taustoja kirjallisuuskatsauksella sekä empiirisesti selvittää ketterien menetelmien hyödyntämistä suomalaisissa pelistudioissa. Tutkimus on toteutettu kyselytutkimuksena käyttäen sähköistä kyselylomaketta.

Ketterät menetelmät ovat mullistaneet ohjelmistokehityksen ja niitä pyritään hyödyntämään myös pelikehityksessä. Kyselytutkimus osoitti, että ketterät menetelmät ovat laajasti pelistudioiden käytössä ja niiden vaikutukset nähdään pääasiassa myönteisinä. Suosituimpia menetelmiä ovat Scrum ja XP, josta hyödynnetään keskeisimpiä käytänteitä. Pelistudion koolla ei ollut vaikutusta ketterien menetelmien hyödyntämiseen. Saatuja tuloksia voidaan hyödyntää jatkotutkimuksen pohjana, esimerkiksi vastaavan kansainvälisen tutkimuksen tekemisessä, ja osoittamaan pelistudioille suuntaa kehitysprosessin muodostamisessa.

Asiasanat: ketterät menetelmät, peliteollisuus

ABSTRACT

Koutonen, Jussi

Agile Methods in Gaming Industry: a survey into Finnish game studios

Jyväskylä: University of Jyväskylä, 2011, 129 p.

Information Systems, Master's Thesis

Supervisor(s): Leppänen, Mauri

Gaming industry is a highly competitive field, where even a single failure can lead to bankruptcy. The market demands more interesting and fun games that need to make use of possibilities offered by quickly evolving technology. In addition to challenges with quality and evolving technology the diversity of stakeholders involved in the development of games makes it very challenging. Game development process of a game studio must be top quality. Game development process is iterative by nature, and game studios are gradually shifting towards full-fledged usage of agile methodologies.

The purpose of this thesis is, firstly, to shed light onto the background of games, gaming industry, game development and agile methodologies, and secondly, to research into the usage of agile methodologies in Finnish game studios. The research was conducted as an electronic survey.

Agile methodologies have revolutionized software development, and they are also being used in game development. The survey proved that agile methodologies are largely used in game studios providing mainly positive impact on game development. The most popular methods are Scrum and XP, with its core practices. Agile methodologies are used regardless of the size of the game studio. The results of the survey can be used as basis for further research. For example, a similar research in the international context, or as a starting point for a game studio to improve its game development process.

Keywords: agile methodologies, gaming industry

KUVIOT

KUVIO 1 Scrum prosessi Cheesmania mukaillen (Cheesman, 2010)	39
KUVIO 2 XP-prosessi Beckiä ja Andersta (2004) mukaillen.....	47
KUVIO 3 Ketterän pelikehityksen prosessimalli (Musil, Schweda, Winkler & Biffel, 2010a)	62
KUVIO 4 Tutkimusasetelma	76
KUVIO 5 Pelistudioiden ikäjakauma (N = 20)	82
KUVIO 6 Pelistudioiden kokojakauma työntekijöiden määrän perusteella (N = 20).....	82
KUVIO 7 Pelistudioiden pelien laitealustat (N = 20)	83
KUVIO 8 Pelistudioiden pelien genret (N = 20)	83
KUVIO 9 Pelistudioiden projektien keskimääräinen kesto (N = 20)	84
KUVIO 10 Yhden projektin keskimääräinen henkilöstömäärä (N = 20)	84
KUVIO 11 Samanaikaisten projektien määrä suomalaisissa pelistudioissa (N = 20).....	85
KUVIO 12 Konseptivaiheen tehtävien suorittaminen.....	87
KUVIO 13 Esituotantovaiheen tehtävien suorittaminen	89
KUVIO 14 Tuotantovaiheen tehtävien suorittaminen	90
KUVIO 15 Jälkituotantovaiheen tehtävien suorittaminen	91
KUVIO 16 Ketterien menetelmien myönteiset vaikutukset kehittämistyöhön (N = 16)	97
KUVIO 17 Ketterien menetelmien vaikutukset projektin hallintaan (N = 16) ...	99

TAULUKOT

TAULUKKO 1 Pelin määritelmän ominaisuudet ja tasot	13
TAULUKKO 2 Pelaajarakenne	18
TAULUKKO 3 Flow-elementtien ja pelikirjallisuudessa esiintyvien elementtien kytkökset (Sweetser & Wyeth, 2005).....	22
TAULUKKO 4 Pelikehityksen vaiheet eri lähteiden mukaan	60
TAULUKKO 5 Pelialan ongelmat ja niihin vaikuttavat ketterien menetelmien käytänteet tai toimintatavat	69
TAULUKKO 6 Tietyn kokoisten yritysten keskimääräinen ikä	85
TAULUKKO 7 Yhteenveto pelistudioiden taustatiedoista	86
TAULUKKO 8 Projektin keskimääräinen kesto suhteessa prototyyppien valmistukseen.....	87
TAULUKKO 9 Projektin keskimääräinen henkilöstömäärä suhteessa prototyyppien valmistukseen.....	87
TAULUKKO 10 Eri vaiheissa käytettävät ketterät menetelmät (N = 17).....	92
TAULUKKO 11 Ketterien menetelmien hyödyntäminen eri vaiheissa suhteessa ikään (N = 20)	93

TAULUKKO 12 Scrumin käytänteiden hyödyntäminen pelikehityksen vaiheissa (N = 16).....	94
TAULUKKO 13 XP-käytänteiden hyödyntäminen pelikehityksen vaiheissa (N= 16).....	95

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
KUVIOT	4
SISÄLLYS.....	6
1 JOHDANTO.....	8
2 PELIT JA PELITEOLLISUUS.....	11
2.1 Peliharrastuksen yleistymisen.....	11
2.2 Pelin määritelmä	12
2.3 Peligenret	13
2.3.1 Genrejako.....	14
2.3.2 Pelitypologia	16
2.4 Pelikokemus	18
2.5 Pelikehitys ja sen ominaispiirteet	25
2.6 Pelialan keskeiset toimijat	27
2.7 Pelien julkaisumallit	29
2.8 Yhteenveto	32
3 KETTERÄT MENETELMÄT	33
3.1 Agile-manifesti	33
3.2 Ketterien menetelmien määritelmä.....	34
3.3 Scrum.....	35
3.3.1 Arvot	35
3.3.2 Käytänteet.....	36
3.3.3 Roolit	38
3.3.4 Prosessimalli	39
3.4 XP	40
3.4.1 Arvot	40
3.4.2 Periaatteet.....	41
3.4.3 Käytänteet.....	42
3.4.4 Roolit	45
3.4.5 Prosessimalli	47
3.5 Lean-ohjelmistokehitys	48
3.5.1 Periaatteet.....	48
3.5.2 Työkalut.....	50
3.6 Kanban.....	53
3.7 Yhteenveto	54
4 KETTERÄ PELIKEHITYS	55

4.1	Pelikehityksen ongelmat.....	55
4.2	Agile-arvot pelikehityksessä	58
4.3	Ketterän pelikehityksen vaiheet	59
4.4	Ketterien menetelmien hyödyntäminen pelikehityksen eri vaiheissa.....	61
4.5	Ketterät käytänteet pelikehityksessä	63
4.6	Tiimit	65
4.7	Roolit.....	67
4.8	Ketterä kehittäminen ratkaisuna pelialan ongelmiin.....	68
4.9	Yhteenveto	72
5	TUTKIMUSMENETELMÄ JA -PROSESSI	73
5.1	Tutkimusmenetelmän valinta.....	73
5.2	Tiedon keruutavan valinta	74
5.3	Otoksen valinta	75
5.4	Tutkimusasetelman luominen ja kyselylomakkeen teko.....	75
5.5	Tiedon keruu ja käsittely	78
6	TULOKSET.....	81
6.1	Pelistudion taustatiedot	81
6.2	Yrityksen pelikehitysprosessi	86
6.3	Yrityksen käyttämät ketterät menetelmät.....	92
6.4	Yrityksen hyödyntämät käytänteet.....	93
6.5	Ketterien menetelmien positiiviset ja negatiiviset vaikutukset	96
6.6	Yhteenveto	100
7	TULOSTEN TARKASTELO	101
7.1	Johtopäätökset.....	101
7.2	Tutkimuksen reliabiliteetti ja validiteetti	104
8	YHTEENVETO	106
	LÄHTEET	108
	LIITE 1 SAATE 1	113
	LIITE 2 SAATE 2.1.....	114
	LIITE 3 SAATE 2.2.....	115
	LIITE 4 KETTERÄT MENETELMÄT PELITEOLLISUUDESSA KYSELY	116

1 JOHDANTO

Peliteollisuudessa liikkuu suuret määrät rahaa (FIGMA ry, 2011). Peliharrastus on kovaa vauhtia yleistymässä, ja pelaajien keski-ikä kasvaessa heillä on myös varaa käyttää enemmän rahaa harrastukseensa (Peltoniemi, 2009). *Pelillä* tarkoitetaan tässä tutkielmassa kaikkia digitaalisilla laitteilla pelattavia pelejä. Markkinoiden kasvaessa peleille asetetut odotukset kasvavat. Vaaditaan teknisesti monimutkaisempia, visuaalisesti näyttävämpiä ja sisällöllisesti merkityksellisempiä pelejä, jotka tuottavat ”tajunnan räjäyttäviä” pelikokemuksia (Arhipainen, 2009). Tällaisten pelien kehittäminen on kallista ja haastavaa (Blow, 2004).

Pelikehittäjät voidaan jakaa karkeasti *itsenäisiin kehittäjiin*, jotka voivat olla yksittäisiä henkilöitä tai pieniä ryhmiä, ja *pelistudioihin*, jotka ovat pelejä kehittäviä yrityksiä. Pelistudiot toimivat useimmiten ulkopuolisen rahoituksen varassa, kun itsenäiset kehittäjät puolestaan rahoittavat itse toimintansa tai tekevät pelejä vapaa-ajallaan. Pelialalla kilpailu on kovaa (Manninen, Kujanpää, Vallius, Korva & Koskinen, 2006). Julkaisija saattaa sijoittaa kymmeniä miljoonia euroja pelistudioon, jonka tehtävä on tuottaa erinomainen peli, joka erottuu edukseen kilpailijoiden joukossa. Mikäli pelistudio epäonnistuu tehtävässään, se maksaa pahimmassa tapauksessa epäonnistumisestaan konkurssin hinnan. Tämän takia pelistudioilla on valtavat paineet pelikehitysprojekteissaan.

Pelit ja niiden kehitystyö poikkeaa ohjelmistokehityksestä, sillä pelit ovat viihdetuotteita, joissa yhdistyy monen alan osaajien tuotokset saumattomaksi pelikokemukseksi (Manninen ym., 2006). Tämä tuo huomattavan lisähaasteen pelikehitykseen ja toisaalta asettaa myös kehitysmenetelmille toisenlaisia vaatimuksia (esimerkiksi eri alojen osaajien työnkulkujen yhteensovittamisen, eri alojen osaajien kommunikaation ylläpitämisen ja yhteisen vision toteutumisen takaamisen) kuin ohjelmistokehitys (Keith, 2010; Manninen ym., 2006; Potanin, 2010).

Pelikokemuksen tärkeyden ja pelikehityksen luonteen takia pelikehitysprosessin tulisi olla iteratiivinen. On ensiarvoisen tärkeää, että peli-ideoita päästään testaamaan jo prosessin alkuvaiheessa, ja muutostarpeisiin voidaan reagoida nopeasti. Tämän iteratiivisen luonteensa vuoksi on luonnollista, että peli-

kehityksessä pyritään hyödyntämään ketterän kehittämisen periaatteita ja käytäntöjä (Agile Alliance, 2001).

Ketterän kehittämisen tueksi on julkaistu useita ketteriä menetelmiä. Tällaisia ovat esimerkiksi Scrum (Schwaber & Beedle, 2001), XP (eXtreme Programming) (Beck & Andres, 2004), Lean (Poppendieck & Poppendieck, 2003) ja Kanban (Kniberg & Skarin, 2009). Ketteriä menetelmiä yhdistää menetelmän helppous, muutoksiin reagoiminen, panostaminen yhteistyöhön ja kehittämisen inkrementaalisuus (Abrahamsson, Salo, Ronkainen & Warsta, 2002).

Kirjallisuudessa on julkaistu paljon pelikehittämiseen liittyviä tutkimuksia muun muassa DIGRAn (Digital Games Research Association) Level Up konferenssijulkaisuissa ja ACM:n Computers in Entertainment lehdessä. Suuri osa niistä käsittelee erityisaiheita liittyen esimerkiksi pelimekaniikkaan tai pelikokemukseen. Joitakin myös pelikehitysprosesseja käsitteleviä julkaisuja löytyy, kuten Bethken Game Development and Production (Bethke, 2003).

Ketterää pelikehitystä koskevia tutkimuksia on tehty kuitenkin hyvin vähän. Keith on kirjoittanut aiheesta käsittelevän kirjan Agile Game Development with Scrum (Keith, 2010). Musil ym. ovat yhteistyössä itävaltalaikeiden pelistudioiden kanssa laatineet esityksen ketterästä pelikehitysprosessista (Musil, Schweda, Winkler & Biffel, 2010a). Kyseiset esitykset ovat yksittäisiä näkemyksiä ketterästä pelikehityksestä. Laajempaa näkemystä ketteryydestä pelikehityksessä on selvitetty Itävallan peliteollisuudessa tutkimuksessa (Musil, Schweda, Winkler & Biffel, 2010b), jossa todetaan, että valtaosa pelistudioista hyödynsi ketteriä menetelmiä. Tutkimus oli hyvin yleisluonteinen. Tarkempia tutkimuksia esimerkiksi siitä, millaisia ketteriä käytäntöjä pelikehityksessä käytetään ja millaisia ei, ei ole tehty.

Tämän tutkielman tavoitteena on selvittää, kuinka paljon ja millä tavalla ketteriä menetelmiä hyödynnetään pelikehityksessä. Tutkimus rajataan koskemaan suomalaisia pelistudioita, joissa on vähintään viisi työntekijää. Tavoite voidaan esittää tutkimusongelmana seuraavasti:

Missä laajuudessa ja muodossa ketterän lähestymistavan periaatteita ja käytänteitä käytetään peliteollisuudessa?

Tutkimusongelma voidaan jakaa kolmeksi tutkimuskysymykseksi:

Millaisia erityispiirteitä pelikehitykseen liittyy?

Millaisia ketteriä menetelmiä hyödyntäviä pelikehitysprosesseja kirjallisuudessa esitetään?

Miten ketterää pelikehitystä tehdään Suomessa?

Ensimmäisen kohdan tavoitteena on selvittää, mikä on peli ja mitä on pelikehitys sekä sen erityispiirteet. Toisena tavoitteena on esitellä tutkielman kanalta keskeiset menetelmät, ja tämän jälkeen selvittää, missä määrin kirjallisuudessa käsitellään ketterien menetelmien hyödyntämistä pelikehityksessä. Kol-

mantena tavoitteena on selvittää, hyödyntävätkö suomalaiset pelistudiot ketteriä menetelmiä.

Tutkimus koostuu käsitteellis-teoreettisesta osuudesta ja empiirisestä osuudesta. Ensiksi mainitulla pyritään vastaamaan kahteen ensimmäiseen tutkimuskysymykseen. Empiirinen osuus toteutetaan suomalaisille pelistudioille suunnattuna kyselytutkimuksena, jossa selvitetään kehitysprosessin vaiheiden toimia, niissä hyödynnettäviä ketteriä menetelmiä ja niiden käytänteitä sekä ketterien menetelmien vaikutuksia kehitystyöhön ja projektinhallintaan. Tämän laatuista tutkimusta ei ole juuri aiemmin tehty. Tiedettävästi ainoastaan Musil ym. (2010) ovat tutkineet peliteollisuudessa hyödynnettäviä kehitysmenetelmiä, mutta hekin vain yleisellä tasolla.

Tämä tutkielma jakautuu seitsemään lukuun. Toisessa luvussa esitellään pelejä ja peliteollisuutta. Kolmannessa luvussa kuvataan tämän tutkielman kannalta relevantit ketterät menetelmät ja niiden yhteinen tausta. Neljäs luku käsittelee ketterää pelikehitystä aiheesta julkaistuun kirjallisuuteen perustuen. Tämän jälkeen luvussa 5 esitellään empiirisen osuuden tutkimusmenetelmä ja -asetelma. Luvussa 6 esitetään kyselytutkimuksen tulokset ja luvussa 7 tuloksista vedetyt johtopäätökset. Tutkielma päättyy lukuun 8, jossa esitetään yhteenveto ja jatkotutkimusaiheita.

2 PELIT JA PELITEOLLISUUS

Tässä luvussa taustoitetaan pelejä, pelaamista ja peliteollisuutta. Luvussa lähdetään liikkeelle pelaamisesta harrastuksena ja tarkastellaan sen suosiota. Tämän jälkeen esitetään pelin määritelmä ja miten pelejä luokitellaan. Kolmantena kokonaisuutena käsitellään pelikokemusta ja sen suunnitteluun liittyviä haasteita. Lopuksi luodaan lukijalle yleiskuva peliteollisuuden toimijoista, heidän välisistä suhteista ja vuorovaikutuksista sekä tutustutaan pelien julkaisumalleihin.

2.1 Peliharrastuksen yleistyminen

Digitaalisten pelien myynti on ollut pieniä notkahduksia lukuun ottamatta kasvussa koko 2000-luvun ajan. FIGMA on suomalainen peliohjelmisto- ja multi-mediayhdistys, joka kerää vuosittain tilastoa pelien myyntimäärästä ja sen rahallisesta arvosta. FIGMAN tilastojen mukaan vuonna 1999 FIGMAN jäsenet myivät pelejä noin 1410000 kappaletta yhteensä 45000000 euron arvosta. Vuonna 2010, jolloin FIGMA ilmoitti Suomen kokonaispelimarkkinoiden tilastot, vastaavat lukemat ovat 3045000 ja 108600000 (FIGMA ry, 2011). Erilaisista digitaalisista latauspalveluista ei kuitenkaan tällä hetkellä saada tilastoja kerättyä, joten etenkin viime vuosien todelliset lukemat ovat todennäköisesti ilmoitettuja suurempia.

Peltoniemen (2009) useista lähteistä kokoamien tilastotietojen mukaan Yhdysvalloissa pelimyynnin arvo dollareissa vuonna 1996 oli noin 2,6 miljardia, kun se vuonna 2007 oli kasvanut jo 9,5 miljardiin. Voidaankin turvallisesti päätellä pelien myyntimäärien ja -arvojen kehityksen perusteella, että pelaamisen suosio on kasvanut. Tähän ovat luultavasti osaltaan vaikuttaneet pelikonsolien uusi sukupolvi, jonka myötä pelaamisesta ja peleistä on tullut yhä helpommin lähestyttävää ajanvietettä.

60% amerikkalaisista pelaa säännöllisesti, ja pelaajien keski-ikä on 35 vuotta. Iso-Britanniassa puolestaan 100% 6-10 -vuotiaista pelaa jonkinlaisia digitaali-

sia pelejä mutta vain 18% 51-65 -vuotiaista. Peltoniemen (2009) tulkinnan mukaan tilastoista voi päätellä, että nuoruudessa pelanneet jatkavat harrastusta vanhemmalla iällä, jolloin voidaan olettaa pelaamisen lisääntyvän myös vanhemmissa ikäluokissa ja tämä edelleen saattaa nostaa pelaajien keski-ikää.

Viime vuonna julkaistussa Suomen pelialan strategiaraportissa (2010), jonka työstivät yhteistyössä Neogames ja Suomen Pelikehittäjät ry, löytyy vastaavia havaintoja pelaamisen yleistymisestä sekä pelimyynnin ja -alan kasvusta. Raportissa esitetään, että syitä peliharrastuksen yleistymiseen ovat muun muassa internetin ja pelaamisen konvergenssi eli sulautuminen, uudet pelilaitteet ja jakelukanavat sekä verkkopelaamisen ja -jakelun voimakas yleistyminen. Myös pelaamisen linkittyminen sosiaalisuuteen nähtiin yhdeksi tekijäksi. Raportissa kuitenkin huomautetaan, että pelkästään teknologiset edistysaskeleet eivät selitä pelaamisen kasvua, vaan teknologian myötä pelien sisältö on kehittynyt ja on voitu tarjota elämyksiä myös uusille kohderyhmille. (Kuittinen ym., 2010)

Tilastojen valossa peliharrastus näyttäisi saavuttaneen yleisen hyväksynnän. Se on siirtynyt pienen piirin hämärästä makuuhuoneista suurten massojen olohuoneisiin, joissa niin lapset kuin aikuisetkin pelaavat. Pelit ovat valtavirran viihdettä siinä missä elokuvat ja musiikkikin (Peltoniemi, 2009).

2.2 Pelin määritelmä

Pelille on annettu kirjallisuudessa useita erilaisia määritelmiä aikojen saatossa. Usein peli on rinnastettu leikkiin ja usein määritelmät ovatkin pyrkineet kattamaan niin pelin, leikin kuin myöhemmin digitaalisen pelin. Eri määritelmät myös keskittyvät eri näkökulmiin; toisissa pyritään määrittämään peli ja joissakin pelaaminen toimintana (Juul, 2003).

Juul (2003) on esittänyt oman määritelmänsä pelille. Hänen määritelmänsä perustuu seitsemään aiemmin esitettyyn määritelmään, joiden yhtäläisyydet poimimalla ja tekemällä tarpeelliset muutokset ja selvennökset hän muodosti oman määritelmänsä. Aiemmat määritelmät jakautuvat melko tasaisesti aikavälille 1950-luvulta vuoteen 2003. Määritelmä kuuluu seuraavasti:

"Peli on sääntöihin perustuva formaali systeemi, jolla on vaihtuvia ja mitattavia lopputulemia, joilla on erilaisia arvoja. Pelaaja pyrkii aktiivisesti vaikuttamaan lopputulemaan ja on näin sidoksissa siihen. Pelillä voi olla seurauksia reaali maailmassa, jos näin on sovittu." (Juul, 2003, s.35)

Määritelmässä on kolme tasoa (taulukko 1): peli formaalina systeeminä, peli ja pelaaja, sekä peli ja muu maailma. Määritelmä voidaan jakaa kuuteen ominaisuuteen, jotka puolestaan voivat kuulua yhteen tai useampaan kolmesta eri tasosta. Ominaisuudet ovat (1) säännöt, (2) vaihtuvat, mitattavat lopputulemat, (3) arvotetut lopputulemat, (4) pelaajan vaikutus, (5) pelaajan side lopputulemaan ja (6) sovittu seuraus reaali maailmassa. (Juul, 2003)

TAULUKKO 1 Pelin määritelmän ominaisuudet ja tasot

Määritelmän ominaisuus	Peli formaalina systeeminä	Peli ja pelaaja	Peli ja muu maailma
Säännöt	X		
Vaihtuvat, mitattavat lopputulemat	X		
Arvotetut lopputulemat	X	X	
Pelaajan vaikutus		X	
Pelaajan side lopputulemaan	X	X	
Sovittu seuraus reaali maailmassa			X

Määritelmän kannalta ongelmallisia ovat tietyt peleiksi luonnehdittavat ohjelmaletut, kuten Sims tai SimCity, joissa ei ole pelin sisällä määriteltyjä tavoitteita, joiden täytyttyä peli päättyisi (Juul, 2003). Sen sijaan esimerkiksi SimCity-pelissä on tarkoitus rakentaa ja ylläpitää kaupunkia näennäisesti loputtomiin. SimCityyn haastetta toivat erilaiset skenaariot ja satunnaisesti kaupunkia vaivaavat luonnonkatastrofit. Pelissä ei kuitenkaan ollut selkeää määritelmää voittamiselle tai häviämislle (Juul, 2003).

Määritelmässä on otettu huomioon myös pelin vaikutus reaali maailmaan, koska muun muassa uhkapelien lopputulemalla on hyvin usein rahallista merkitystä pelaajalle. Samoin Juul mainitsee yhä kasvavan ammattipelaamisen mahtuvan määritelmän alaisuuteen. Edellä mainituissa tapauksissa on sovittu ennen peliä reaali maailman seurauksista. (Juul, 2003) Entäpä tapaukset, joissa pelaaja ei välttämättä tiedosta, että pelin pelaamisella on seurauksia reaali maailmassa? Näin voisi käydä esimerkiksi Habbo Hotel -pelissä, jonka pelaajakunta on iältään hyvin nuorta ja jossa on mahdollista kalustaa huonettaan huonekaluilla tilaamalla ne tekstiviestillä (Wikipedia, 2010a). On tosin huomattava, että Habbo Hotel ei sovi Juulin määritelmään pelistä.

Tässä työssä käytetään Juulin määritelmää pelistä sillä poikkeuksella, että rajatapaukset, jotka eivät täysin täytä määritelmää, katsotaan peleiksi. Usein rajatapaukset ovat pelaajien silmissä pelejä siinä missä kaikki määritelmän täyttävätkin pelit. Rajatapauksia edustavien pelien kehitys ei todennäköisesti oleellisesti poikkeaa määritelmän täyttävien pelien kehityksestä.

2.3 Peligenret

Genre tarkoittaa tyyli suuntaa, ja genrejä on käytetty työkaluna kategorisoinnissa ja analysoinnissa (Ye, 2004). Esimerkiksi musiikin eri tyylit, kuten reggae ja country, ovat genrejä samoin kuin elokuvissa draama ja komedia. Eri tyyliellä on usein tietyt sisällölliset ominaispiirteensä, jotka ovat leimallisia kaikille kyseisen tyylin edustajille. Myös pelejä voidaan luokitella eri genreihin.

Pelien genremäärittelyjen suhteen on olemassa kaksi koulukuntaa: (1) kerontaa, ulkoasua ja tyyliä painottava koulukunta, joka lähestyy pelejä samoilla käsitteillä kuin musiikkia ja elokuvia, ja (2) pelien vuorovaikutuksellisuutta painottava koulukunta, johon nykyiset pelitutkijat kuuluvat. (Apperley, 2006) Yen (2004) mukaan pelaajien ja pelin tekijöiden luomat genret ovat muotoutuneet luonnollisella tavalla kahden viime vuosikymmenen kuluessa. Hänen mukaansa suurin osa näistä genremäärittelyistä perustuu pelaajan ja pelin vuorovaikutukseen ja pelikokemukseen ennemmin kuin sisältöön tai ilmaisuun ja tyylin muotoon (Ye, 2004). Apperley (2006) puolestaan väittää, että nykyiset peligenret painottavat edelleen kuvausta (engl. representation) enemmän kuin mitään käsitystä vuorovaikutuksesta.

On siis epäselvää kumman koulukunnan näkemystä ajan kuluessa muotoutuneet, yleisesti kuluttajien keskuudessa hyväksytyt peligenret lopulta edustavat. Ne eivät myöskään ole millään tavalla ehdottomia genrejä, koska hyvin usein tilanne on sellainen, että peli määritellään kuuluvaksi useampaan genreen. Jos kuluttajien hyväksymät genret mielletään heidän ja kehittäjien väliseksi sopimuksiksi kokemuksesta, joita tietyn genren edustajat tarjoavat, niin luokittelu on sekä kehittäjien että kuluttajien näkökulmasta riittävä, koska molemmilla osapuolilla on riittävän samanlainen näkemys tyypillisestä genren tai genrehybridin edustajasta. Peliummikolle genreistä ei ole apua. Ye myös huomauttaa, että genret ovat jatkuvassa muutoksen tilassa kehittyvän teknologian vuoksi (Ye, 2004).

Seuraavaksi esitellään lyhyesti yleisimpiä peligenrejä ja sen jälkeen eräs kattavimmista typologioista, jolla pelejä voidaan systemaattisesti luokitella.

2.3.1 Genrejako

Vaikka edellä mainittiin ajan kuluessa muotoutuneet, hyväksytyt peligenret, on vaikeaa löytää kattavaa listausta nykyisistä peligenreistä tieteellisistä lähteistä. Peligenret vaikuttavatkin olevan pelistudioiden ja -julkaisijoiden sekä pelaajien välistä hiljaista tietoa. Erilaisia peligenrejen listauksia löytyy internetistä useita. Näissä listauksissa muutamat genret esiintyvät aina ja niiden rinnalla esitetään ala- tai hybridigenrejä. Seuraavaksi esitellään muutamia, keskeisiksi koettuja peligenrejä niitä käsittelevästä Wikipedia-artikkelista ja suomalaiselta pelitieto.net -sivustolta.

Räiskintäpelit (engl. shooter) ovat usein kolmiulotteisia pelejä, joissa pelaajahahmo erilaisia aseita hyödyntäen eliminoi vihollisia ja etenee pelimaailmassa vaihtelevien tavoitteiden mukaan (Mäyrä ym., 2009). Räiskintäpelit ovat hyvin reaktio- ja refleksivetoisia haasteensa puolesta (Mäyrä ym., 2009). Wikipedia-artikkeli tunnistaa räiskintäpeleille useita aligenrejä, joista tätä nykyään tärkeimmät lienevät ensimmäisen persoonan ja kolmannen persoonan räiskintäpelit. Näiden kahden keskeinen ero on kuvakulman vaihtuminen ensimmäisestä persoonasta kolmanteen persoonaan, jolloin kuvakulma yleensä vedetään pelihahmon taakse laajemman näkymän ja monipuolisempien pelaajahahmon toi-

mintojen saavuttamiseksi. Edellä mainituista on olemassa niin yhden kuin usean pelaajan toteutuksia. (Wikipedia, 2010b)

Wikipediassa lasketaan räiskintäpeleihin kuuluviksi myös historialliset räiskintäpelit (engl. shoot 'em up tai shmup), kuten Space Invaders, valopistoolipelit, joissa ruutua tähtäillään erillisellä aseohjaimella ja peli kuljettaa pelaajahahmoa ikään kuin raiteilla ennalta määriteltyjen pisteiden välillä, ja taktiset räiskintäpelit, joissa lähestytään räiskintää taktisesta, realistisesta näkökulmasta, jolle on ominaista sotilasryhmän osana toimiminen tai sellaisen komentaminen sen sijaan, että pelaajahahmo taistelisi yksin. (Wikipedia, 2010b)

Tietokoneroolipelit (engl. RPG, role-playing [video] games) ovat alun perin kehittyneet perinteisistä kynä ja paperi -roolipeleistä, kuten Dungeons & Dragons -roolipelistä (Wikipedia, 2010b). Tietokoneroolipelit alkavat usein pelaajahahmon, avatarin, luonnilla. Avatarilla voi olla paljon erilaisia ominaisuuksia, kykyjä ja piirteitä, joihin pelaajalla on mahdollisuus vaikuttaa luontivaiheessa. Tietokoneroolipeleissä pelaajahahmo edelleen kehittyy ja muuttuu pelin edetessä. Peliympäristönä toimii yleensä jokin kuvitteellinen maailma, jonka tapahtumat vaikuttavat toisaalta pelaajahahmoon ja toisaalta pelaajahahmo pyrkii vaikuttamaan pelimaailman tapahtumiin ja tätä kautta pelin juonen kulkuun. (Mäyrä ym., 2009)

Kulttuuriset erot tietokoneroolipeleissä ovat tuottaneet kaksi suuntausta, joilla on muutamia muuttumattomia ominaisuuksia. Länsimaisissa roolipeleissä yleensä luodaan avatar ja seikkaillaan maailmassa seuraten epälineaarista juonta, kun taas itämaisissa roolipeleissä pelaajahahmo on ennalta määrätty ja peli etenee lineaarista juonipolkua (engl. storyline) seuraten. (Wikipedia, 2010b) Molemmissa suuntauksissa pelaajahahmo usein kerää myös seurueen, jonka jäsenillä on erilaisia, toisiaan täydentäviä kykyjä.

Eräs tietokoneroolipeli-aligenre on massiivisesti usean pelaajan verkkoroolipeli (engl. MMORPG tai MORPG eli [massively] multiplayer online role-playing game), johon muun muassa World of Warcraft kuuluu, mutta genre on ollut olemassa teknologisesti hieman vaatimattomammin toteutettuina tekstipohjaisina MUDeina (Multi-User Dungeon) jo 1980-luvulla. Verkkoroolipelit toivat sosiaalisen elementin aidosti takaisin tietokoneroolipeleihin. (Mäyrä ym., 2009)

Seikkailupeleissä (engl. adventure) keskitytään ongelmanratkaisuun ja tarinankerrontaan eikä niissä yleensä ole varsinaista taistelumekaniikkaa tai jatkuvasti uhkaavia vihollisia. Pelillisen ongelman ratkaistuaan, joko puhumalla tai esineitä käyttämällä, pelaajahahmo pääsee tarinassa eteenpäin. Usein pelitilanteet ovat verkkaisia ja ongelmien ratkaisua voi pohdiskella kiireettömästi. (Mäyrä ym., 2009) Seikkailupeleiksi saatetaan kutsua myös toimintapelejä, joissa on seikkailullisia elementtejä. Tämä on kuitenkin virhe, joka johtuu seikkailupeli-puuttumisesta konsolialustoilta ja konsoleille tyypillisen toimintaseikkailu -genren (engl. action-adventure) nimittämisestä seikkailupeliksi. Yksinkertaistetusti toimintaseikkailu on seikkailupeli, jossa taistelu on keskeinen elementti. (Wikipedia, 2010b) Seikkailupeli-elementtejä käytetään nykyään

useissa muissa peligenreissä, kuten aiemmin esitellyissä räiskintäpeleissä ja roolipeleissä. (Mäyrä ym., 2009)

Strategiapelit (engl. strategy games) kaksi tai useampi osapuoli kamppailevat selviytymisestä tai jonkin voittotavoitteen saavuttamisesta. Kamppailun keskiössä ovat taktinen ajattelu ja rajallisten resurssien tehokas hyödyntäminen. Strategiapelien katsotaan olevan lautapelien johdannaisia. (Mäyrä ym., 2009; Wikipedia, 2010b) Strategiapelit ovat joko vuoropohjaisia tai reaaliaikaisia, mikä voi tehdä kahden tämän genren pelin erot melko suuriksi. Useimmiten strategiapelit käsittelevät sotimista ja niiden mittakaava voi vaihdella yksittäisten sotilaiden käskyttämisestä kokonaisen valtakunnan johtamiseen. Jälkimmäisessä tapauksessa ero strategiapelin ja simulaation välillä on häilyvä, sillä tapaus voidaan käsittää myös valtakuntasimulaationa. (Mäyrä ym., 2009)

Simulaatio (engl. simulation) on genrenä monimuotoinen. Siihen sisältyvät niin urheilu-, ajoneuvo-, rakentelu- kuin elämä-, jumal- ja valtakuntasimulaatiot. Simulaatioissa yleensä jäljitellään jotain reaali maailman tapahtumaa, esinettä tai ilmiötä, mikä onkin niiden yhdistävä tekijä. Ulkoasun, tarinan tai juonen osalta simulaatioissa ei ole yhteistä nimittäjää. Niinpä genreen kuuluvat niin siviili- kuin sotalentokoneiden lentämistä mallintavat pelit, kaupunkien rakenteluun keskittyvät ohjelmalelut, elämysimulaatiot ja autoilupelit, joissa ajaminen on pääasia eikä vain mauste. (Mäyrä ym., 2009) Simulaatioista löytyy useita esimerkkejä, jotka eivät täytä Juulin määritelmää pelistä.

Edellä esiteltujen genrejen lisäksi on olemassa muun muassa hyöty-, bile-, musiikki- ja kuntoilu- sekä oppimispelejä, joista osa voidaan luokitella samaan genreen kuuluviksi. (Mäyrä ym., 2009; Wikipedia, 2010b)

Genreihin sisältyy pelaajan taholta tiettyjä odotuksia, jotka pelikehittäjän tulee ottaa huomioon. Toisaalta nämä odotukset voivat olla myös taakka, koska ne voivat jossain määrin rajoittaa kehittäjien luovuutta. Kehittäjien tuleekin esitellä uusia ideoita rikkomatta liikaa pelaajan odotuksia ja käsitystä genrestä. (Ye, 2004)

2.3.2 Pelitypologia

Aarseth ym. (2003) esittävät kolmannen näkemyksen peligenreihin (Aarseth, Smedstad & Sunnan, 2003). Heidän mukaansa aiemmat yritykset tyypitellä tai luokitella pelejä ovat epäonnistuneita, koska niissä on ollut taipumus sisällyttää liian monta, epämääräisesti määriteltyä genreä, jotka eivät ole olleet yhteensopivia tai ne ovat olleet osittain päällekkäisiä (Aarseth ym., 2003). He ovatkin esittäneet oman typologiansa, joka ottaa huomioon viisi ulottuvuutta, joiden sisällä on useita muuttujia, jotka voivat saada eri arvoja riippuen siitä, miten kuhunkin ulottuvuuteen liittyvät osa-alueet on tyypiteltävässä pelissä toteutettu (Aarseth ym., 2003). Seuraavaksi esittelen typologiaa hieman tarkemmin. Esittely perustuu kokonaan Aarseth ym. (2003) artikkeliin, mutta suluissa esitetyissä esimerkeissä on harkinnan mukaan myös omia lisäyksiä.

Typologian viisi ulottuvuutta ovat:

- Tila (engl. space), jolla tarkoitetaan pelien tapaa esittää peliympäristö
- Aika (engl. time), jolla tarkoitetaan pelimaailman sisäisen ajankulun vaikutusta pelihahmoihin, ympäristöön ja edelleen pelaamiseen ja pelikokemukseen
- Pelaajarakenne (engl. player structure), jolla tarkoitetaan pelin mahdollisia, erilaisia pelaajamääriä ja niiden jakautumista kilpaileviin ryhmiin
- Ohjaus (engl. control), jolla tarkoitetaan pelien ominaisuuksia, jotka ohjaavat pelaajan pelitapaa ja toimia pelissä
- Säännöt (engl. rules), jossa kartoitetaan, minkä tyyppisiä sääntöjä pelissä on, eikä niinkään jaotella pelejä varsinaisten sääntöjen kautta.

Tila-ulottuvuudessa on kolme muuttujaa. Näistä ensimmäinen on näkökulma tilaan. Se voi olla kaikkialla läsnä oleva (engl. omnipresent), jolloin pelaaja voi aina tarkastella mitä tahansa osaa pelialueesta (Warcraft, Farmville) tai näkeä sen kokonaisuudessaan aina (shakki, tetris), tai kulkuri (engl. vagrant), jolloin kuvakulma seuraa aina pelaajahahmoa (Call of Duty). Toinen muuttuja on pelimaailman topografia (engl. topography). Se voi olla joko geometrinen (engl. geometrical), jolloin pelaajahahmo voi liikkua maailmassa saumattomasti ja liikkeen suunnalle on jokaisella hetkellä tuhansia vaihtoehtoisia suuntia, tai topologinen (engl. topological), jolloin pelimaailmassa on rajallinen määrä diskreettejä, ei-päällekkäisiä sijainteja, joiden välillä pelaajahahmo voi liikkua. Joissakin peleissä tekninen toteutus on kahden esitetyn arvon välimaastossa, mutta useimmiten pelielementit eivät voi olla päällekkäin näissä peleissä, joten ne luokitellaan topologisiksi. Viimeinen muuttuja tila-ulottuvuudessa on ympäristö tai oikeastaan peliympäristön muuttuminen pelin aikana. Peliympäristö voi olla staattinen (engl. static), jolloin pelin aikana peliympäristön alue pysyy samanlaisena läpi pelin (shakki, Warcraft), tai dynaaminen (engl. dynamic), jolloin pelaajahahmo voi muokata peliympäristöä (Lemmings, Minecraft). Ympäristön muokkaamiseksi ei kuitenkaan lasketa esimerkiksi Warcraftissa tuontorakennuksen pystyttämistä, sillä se ei varsinaisesti muokkaa ympäristöä vaan vain muuttaa hieman pelin tilaa.

Aika-ulottuvuudessa muuttujia on kolme. Pelin tahti voi olla joko vuoropohjainen (engl. turnbased) tai reaaliaikainen (engl. realtime). Ajan kuvaus (engl. representation) voi olla jäljittelevä (engl. mimetic), jolloin teot pelimaailmassa ovat kestoltaan suunnilleen saman mittaisia kuin reaali maailmassa, tai epämääräinen (engl. arbitrary), jolloin pelimaailman tapahtumat eivät pyri matkimaan todellisuuden tapahtumia. Teleologialtaan pelit voivat olla joko rajattomia (engl. infinite) tai rajallisia (engl. finite), mikä tarkoittaa, että peleillä joko on tavoiteltava tila, jossa pelin "voittaa", tai tällaista tilaa ei ole.

Pelaajarakenne-ulottuvuus on yksi muuttuja. Pelaajarakenne koostuu itse asiassa kahdesta erillisestä ulottuvuudesta, jotka ovat vastustajarakenne ja ryhmärakenne. Pelissä ei joko ole vastustajia (tetris) tai niitä on yksi tai useampia. Nämä kolme vaihtoehtoa muodostavat vastustajarakenteen. Ryhmärakenteessa on kaksi vaihtoehtoa, yksilö ja ryhmä. Nämä kaksi ulottuvuutta yhdistyvät vaivattomasti pelaajarakenne-ulottuvuudeksi, jossa on kuusi kategorialla,

jotka ovat taulukon 2 oikeanpuoleisimmat sarakkeet. Usein pelit toteuttavat useamman pelaajarakenne muuttujan arvon, tällöin valitaan monimutkaisin rakenne, jonka mukaan peli luokitellaan.

TAULUKKO 2 Pelaajarakenne

Vastustajat	Yksilö	Ryhmä
Ei vastustajia	Yksi pelaaja	Yksi ryhmä
Yksi vastustaja	Kaksi pelaajaa	Kaksi ryhmää
Useita vastustajia	Useita pelaajia	Useita ryhmiä

Ohjaus-ulottuvuudessa on kolme muuttujaa. Muuttuvuus (engl. mutability) tarkoittaa pelaajahahmon muuttuvuutta. Pelaajahahmo voi olla muuttumaton (engl. static), saavuttaa väliaikaisia voimistimia (engl. power up) tai saavuttaa pysyviä kokemustasoja (engl. experience levels). Tallennettavuudeltaan (engl. savability) pelitilanne voi olla ei-tallennettava (engl. non-saving), ehdollisesti tai rajattomasti tallennettava. Pelin kulun deterministisyys (engl. determinism) voi saada kaksi arvoa: epädeterministinen ja deterministinen. Deterministisessä pelissä tietty pelaajan syöte tietyssä tilanteessa tuottaa aina saman tuloksen; epädeterministisessä pelissä on satunnaisuustekijä mukana, joten tietyssä tilanteessa tietty syöte voikin antaa erilaisen lopputuloksen.

Sääntö-ulottuvuudessa on kolme muuttujaa, jotka saavat joko kyllä- tai ei-arvon riippuen pelissä esiintyvien sääntöjen luonteesta suhteessa topologiaan, aikaan ja tavoitteisiin. Pelissä on topologiasääntöjä, jos jokin pelin sääntö riippuu pelaajahahmon paikasta maailmassa. Samalla tavoin pelissä on aikäsääntöjä, jos pelkästään ajan kuluminen muuttaa pelin tilaa. Pelissä on tavoiteperustaisia sääntöjä, jos pelin edistyminen tai lopputulos riippuu tietyn ehdon täytymisestä.

Typologian kehittäjät huomauttavat, että typologia ei ole missä tapauksessa lopullinen tai ehdoton menetelmä pelien tyypittelyyn ja että siihen voi jokainen tehdä muokkauksia tai parannuksia (Aarseth ym., 2003).

Tämän tutkielman puitteissa harkittiin mahdollisuutta rajata pelikehityksen tutkimus käsittelemään tiettyjen genrien edustajia. Tämä ei kuitenkaan ollut järkevää, koska pelikehitys ei ole niin merkittävästi erilaista eri genren edustajien välillä. On myös huomattava, että saman genren edustajat voivat olla hyvin erilaisia riippuen niitä kehittäneen pelistudion koosta ja budjetista.

2.4 Pelikokemus

Pelit tarjoavat erilaisen kokemuksen kuin muut joka kodin viihdemuodot. Television tai elokuvien katselu, kirjojen lukeminen ja musiikin kuuntelu ovat passiivisia viihdemuotoja, joissa ei tapahdu konkreettista vuorovaikutusta viihteen kuluttamisen aikana. Peleissä pelaaja puolestaan on aktiivinen toimija, jonka

toimista pelitapahtumat riippuvat. Pelejä voitaisiin lähestyä puhtaasti käyttäjäkokemuksen näkökulmasta, mutta silloin pelin taiteellinen sisältö jää vähäisemmälle huomiolle. Käyttäjäkokeuksiin liittyvät lainalaisuudet pitävät kuitenkin paikkaansa myös peleissä, tavallaan pelit seisovat jalka kummassakin leirissä. Sanders (2001) on esittänyt ajatuksen käyttäjäkokemukseen liittyen, joka pitää paikkaansa myös pelikokemuksen suhteen (Manninen ym., 2006):

”Ei ole olemassa asiaa, jota kutsutaan kokemuksen suunnitteluksi. Kokemusta ei voi suunnitella, koska kokeminen tapahtuu ihmisessä. Voidaan kuitenkin suunnitella kokemista varten. Suunnitellaan rakennusteline tai infrastruktuuri, jota ihmiset voivat hyödyntää muodostaessaan omia kokemuksiaan.” (Sanders, 2001)

Käyttäjäkokemuksen määrittelemisen on vaikeaa. Arhippaisen (2009) väitöskirjassa käyttäjäkokemuksen määritelmää pohdiskellaan useista näkökulmista eikä hän lopulta päädy yksittäiseen määritelmään. Keskeistä määritelmän ongelmallisuudessa on käyttäjäkokemuksen kaikenkattavuus. Käyttäjäkokemus muodostuu käyttäjän ja laitteen välillä ainakin käytettävyydestä, käyttäjän tunteista ja odotuksista laitetta kohtaan sekä vuorovaikutuksesta laitteen kanssa. (Arhippainen, 2009)

Vuorovaikutuksen eli oikeastaan pelien suunnittelua helpottamaan on useiden Game Developers Conference -työpajojen yhteydessä luotu MDA-malli, joka pyrkii selittämään, mistä osista pelikokemus koostuu ja miten pelikokemusta eli pelaajan ja pelin vuorovaikutusta voidaan suunnitella paremmin. (Hunicke, LeBlanc & Zubek, 2004) Seuraavaksi tarkastellaan tätä mallia lähemmin.

Pelejä kulutetaan samoin kuin elokuvia ja muita viihdetuotteita, jotka käytön jälkeen hyljätään. Hunicke ym. (2004) ovat havainneet, että pelien ja muiden viihdemuotojen ero on, että pelien pelaaminen on suhteellisen ennalta arvaamattontaa. Peli on alusta, joka tarjoaa käyttäjälle avaimet omaan kokemukseen, jonka suuntaviivat on asetettu pelikehittäjien toimesta heidän pyrkiessä tarjoamaan tietynlaista kokemisen mahdollisuutta. Pelaajan erikoisen aseman vuoksi hänellä on suuri vaikutus omaan pelikokemukseensa; samaa peliä voidaan pelata periaatteessa monella täysin erilaisella tavalla.

Pelitapahtumien sarjaa ja tapahtumien lopputulosta ei tiedetä, kun varsinaisen pelikehitys on tullut päätökseensä. MDA-mallissa kulutustapahtuma pilkotaan erillisiin komponentteihin, joita ovat säännöt (engl. rules), järjestelmä (engl. system) ja ”hauskuus” (engl. fun). Näitä vastaavat suunnittelussa käytettävät käsitteet ovat mekaniikka (engl. mechanics), dynamiikka (engl. dynamics) ja estetiikka (engl. aesthetics), joiden englanninkielisten nimitysten alkukirjaimista malli on saanut lyhenteensä. (Hunicke ym., 2004)

Mallia käsittelevässä artikkelissa todetaan, että pelin sisältö on sen käyttäytyminen (engl. behavior), eikä niinkään kuva ja ääni, joka virtaa pelaajaa kohti (Hunicke ym., 2004). Vaikka pelit usein pyrkivät olemaan ulkomuodoltaan mahdollisimman viimeistellyjä, monet pelaajat kokevat sisällön eli käyttäytymisen tärkeämmäksi kuin pelin ulkoasun ja äänimaailman. Mainioita esimerkkejä tästä ovat viime aikoina valtaisaan suosioon nousut palikkagrafiikal-

la toteutettu Minecraft, jota ei hyvällä tahdollakaan voi kutsua kauniiksi peliksi, ja lukuiset retropelien nykyaikaisilla laitteilla toimivat uudelleenjulkaisut.

Mekaniikka kuvaa pelin yksittäiset komponentit datana ja algoritmeina. *Dynamiikka* kuvaa yksittäisen mekaniikan komponentin suorituksen aikaista käytöstä se toimiessa pelaajan ja muiden komponenttien syötteisiin reagoiden. *Estetiikka* kuvaa pelaajan pelin aikaisia, haluttuja emotionaalisia reaktioita. Pelisuunnittelijan näkökulmasta mekaniikka mahdollistaa järjestelmän dynamiikan, mikä johtaa esteettiseen kokemukseen. Pelaajan näkökulmasta estetiikalla on jokin sävy, joka kumpuaa havaittavasta dynamiikasta ja lopulta taustalla toimivasta mekaniikasta. (Hunicke ym., 2004)

Pelikokemuksen halutaan usein olevan "hauska". Hunicken ym. artikkelissa pelikokemuksesta käytetään nimitystä estetiikka. Estetiikkaa kuvaamaan he esittelevät sanaston, joka sisältää mutta ei ole rajoittunut seuraaviin:

1. Aistikokemus, jolla tarkoitetaan pelin tarjoamaa aistikokemuksellista mielihyvää
2. Mielikuvitus, jolla tarkoitetaan peliä satuna
3. Kerronta, jolla tarkoitetaan peliä draamana
4. Haasteellisuus, jolla tarkoitetaan peliä esteratana
5. Yhteisöllisyys, jolla tarkoitetaan pelissä olevia ja sen ympärille rakentuvia sosiaalisia rakenteita
6. Löytäminen, jolla tarkoitetaan löytämisen ja tutkimisen iloa
7. Ilmaisu, jolla tarkoitetaan pelin tarjoamia itseilmaisun mahdollisuuksia
8. Alistuminen, jolla tarkoitetaan pelin rajoitteiden ja sääntöjen noudattamista.

Monet pelit ovat "hauskoja" mutta on havainnollisempaa luokitella, mitkä esteetiikan osa-alueet tuottavat jonkin pelin pelikokemuksen. Usein peleissä yhdistyy useita osa-alueita eriasteisesti toteutettuna. Samaan tapaan kuin pelitypologian avulla voi luokitella pelejä niiden havaittavien ominaisuuksien avulla, MDA-mallin estetiikka-ajattelu pyrkii luomaan kuvan niistä paloista, joista pelikokemus koostuu, ja näin valottamaan, miten ja miksi erilaiset pelit ovat hyviä. (Hunicke ym., 2004)

Dynamiikka pyrkii tuottamaan esteettisen kokemuksen. Esimerkiksi haaste tulee vastustajista ja aikarajoitteista. Yhteisöllisyys tulee voittotavoitteista, joita on hankala saavuttaa yksin. Kehittämällä malleja, jotka ennustavat ja kuvailevat pelin dynamiikkaa, voidaan pelisuunnittelua ohjata haluttuun suuntaan. Esimerkiksi Monopolista voidaan tehdä malli, jossa arvioidaan yhteen pelilaudan kierrokseen vaaditut heitot. Samalla tavoin voidaan tunnistaa palauttejärjestelmät pelistä, jotka määrittävät miten tilanne tai muutos vaikuttavat pelin kokonaistilaan. Monopolissa rikkaat rikastuvat ja köyhät köyhtyvät. Kun rikkauden ja köyhyyden ero kasvaa, pelistä loppuu mielenkiinto. Käyttäen ymmärrystä esteettisistä tavoitteista ja dynamiikasta, voidaan kuvitella keinoja Monopolin "korjaamiseen" joko palkitsemalla häviämässä olevia pelaajia tai hankaloittamalla voittamassa olevia pelaajia. (Hunicke ym., 2004)

Mekaniikkaan kuuluvat erilaiset toimet, käyttäytymismallit ja ohjausmekanismit, jotka ovat pelaajan käytettävissä. Yhdessä pelin sisällön kanssa se tukee pelin dynamiikkaa. Säättämällä pelimekaniikkaa hienosäädetään pelin dynamiikkaa ja muutokset dynamiikkaan vaikuttavat estetiikkaan. MDA-mallin keskeinen idea on, että sama toimii myös päinvastoin. (Hunicke ym., 2004) Vaikka MDA-malli pyrkiikin helpottamaan pelikehittäjien työtä, on vaikea enustaa pelin estetiikkaa. Tämän takia pelitestausta on keskeisessä asemassa koko pelikehityksen ajan (Sydow, 2008).

Hieman toisenlaisen näkökulman pelien hyvyyden rakentumisesta tarjoaa GameFlow -malli (Sweetser & Wyeth, 2005). Mallissa on koostettu yhteen useita kirjallisuudessa esitettyjä, pelien eri osa-alueiden arviointiin tarkoitettuja heuristiikkoja, joiden avulla on pyritty ymmärtämään, mikä pelissä tuottaa pelaajalle nautintoa. Mallin jäsentely perustuu flow-teoriaan, joka on oppi täydellisestä kokemuksesta.

Flow-teorian kehitti Mihalyi Csikszentmihalyi, ja se perustuu perusteelliselle havainnoinnille, jossa usean vuosikymmenen aikana tutkittiin ensin ihmisiä, jotka kuluttivat aikaa haastaviin tehtäviin, joista ei ollut suoranaista hyötyä, ja myöhemmin ihmisiä arkipäivän askareiden parissa. Haastateltavia pyydettiin kuvailemaan elämää parhaimmillaan ja sitä tilannetta, kun heidän tehtävänsä oli kaikkein nautinnollisinta. Lopputuloksena esitettiin täydellisen kokemuksen kahdeksan elementtiä, jotka useimmiten mainitaan, kun kuvaillaan erittäin onnistunutta kokemusta: (Csikszentmihalyi, 1990)

1. Kohdataan suoritettavissa oleva tehtävä
2. On mahdollista keskittyä tekemiseensä
3. Tehtävällä on selkeät tavoitteet
4. Tehtävästä saa välitöntä palautetta
5. Toimitaan erittäin syventyneesti mutta vaivattomasti siten, että tietoisuudesta katoaa jokapäiväiset murheet
6. Tunnetaan, että tilanne on hallinnassa
7. Huoli itsestä katoaa, mutta palaa voimakkaampana flow-kokemuksen jälkeen
8. Ajantaju katoaa.

Flow-tilaan pääsyä edesauttavat tietyt asiat, jotka eivät kuitenkaan ole välttämättömän tärkeitä sen saavuttamiseksi. Sopiva tasapaino tehtävän haastavuuden ja suorittajan kykyjen välillä usein edeltää flow-kokemusta ja eräs avainelementti flow-tilaan on tehtävän itsetarkoituksellisuus (engl. autotelic) eli tehtävää tehdään vain tehtävän itsensä takia tai huvikseen, mikä on totta lähes kaikkien pelien kohdalla. (Sweetser & Wyeth, 2005)

GameFlow -mallin idea on arvioida, miten hyvin pelin mekaniikka ja muu toteutus mahdollistavat flow-tasaisen pelikokemuksen. Tätä tarkoitusta varten tekijät tutkivat peleihin liittyvää käytettävyyttä ja käyttäjäkokemuskirjallisuutta selvittääkseen, miten flow-elementit ilmenevät peleissä. GameFlow -malli luotiin perustuen flow-elementteihin ja todisteisiin flow-kokemuksista peleissä. GameFlow -mallissa on kahdeksan ydin osaa (taulukko 3): keskittyminen, haas-

te, taidot, hallinta, selkeät tavoitteet, palaute, immersio ja sosiaalisuus. Osan niistä voi suoraan yhdistää Csikszentmihalyin flow-elementteihin, vaikkakin muutamien yhteys pitää selkeyttää. (Sweetser & Wyeth, 2005)

TAULUKKO 3 Flow-elementtien ja pelikirjallisuudessa esiintyvien elementtien kytkökset (Sweetser & Wyeth, 2005)

Pelikirjallisuus	Flow
Peli	Täytettävissä oleva tehtävä
Keskittyminen	Mahdollisuus keskittyä
Haaste ja pelaajan taidot	Taitojen tulisi vastata haastetta ja molempien tulisi ylittää tietty taso
Hallinta	Tunne tilanteen ja toiminnan hallitsemisesta
Selkeät tavoitteet	Tehtävällä on selkeät tavoitteet
Palaute	Tehtävä tarjoaa välitöntä palautetta
Immersio	Syventynyt, vaivaton osallistuminen, vähentynyt huoli itsestä ja taju ajasta
Sosiaalisuus	Ei liity flow'uun

Flow-elementtien ja niitä vastaavien peli-ilmentymien yhteensovittamisessa täytettävää tehtävää vastaa Sweetserin ym. (2005) mukaan itse peli, eivätkä he täten ota sitä GameFlow-malliin elementiksi. Cowley ym. (2008) kuitenkin kritisoi väitettä varsin aiheellisesti omassa pelien flow-kokemusta tutkivassa artikkelissaan. Heidän mielestään tehtävällä tarkoitetaan enemminkin yksittäistä peli-istuntoa (Cowley, Charles, Black & Hickey, 2008). Cowley ym. (2008) järkeilee, että tavoitteen ei tarvitse olla konkreettinen tavoite eli tehtävän lopputulos, kuten pelin läpäisy, vaan se voi olla pelkästään halu paneutua johonkin toimeen toimen itsensä takia.

GameFlow'n sosiaalisuuselementtiä vastaavaa elementtiä ei löydy flow-teoriasta, mutta se esiintyy usein pelien käyttäjäkokemuksia käsittelevässä kirjallisuudessa (Sweetser & Wyeth, 2005). Cowley näkee sosiaalisuuselementin turhana tässä yhteydessä ja sisällyttää sen peli-istuntoon sisäänrakennetuksi; peli-istunnossa voi olla mikä tahansa määrä osallistujia (Cowley ym., 2008). Perusteluna hän käyttää sosiaalisen elementin uupumisen flow-teoriasta vaikka Csikszentmihalyi havainnoi alkuperäisessä tutkimuksessaan muiden muassa shakin pelaajia eli usean pelaajan peliä (Cowley ym., 2008). Sweetser tunnistaa, että sosiaalisuus ei ole flow-elementti ja se voi olla jopa haitaksi flow-kokemukselle, mutta sosiaalisuus on hänen mukaansa erittäin olennainen osa pelien nautinnosta (Sweetser & Wyeth, 2005). Nykypäivän trendi on sosiaalisuus; on olennainen osa peliä, että siinä on tarjolla usean pelaajan samanaikainen osallistuminen tai joukkuepelaaminen ja yhtä tärkeää on voida pelin ulkopuolella keskustella pelistä tai omasta edistymisestä (Sweetser & Wyeth, 2005).

Muut GameFlow -mallin elementit ovat toisiinsa liittyviä ja toisistaan riippuvaisia. Pelit pitävät pelaajan keskittymisen yllä suurella työtaakalla, joka rakentuu riittävän haasteellisista ja tämän vuoksi nautinnollisista tehtävistä, koska pelaajan kyvyt kuitenkin riittävät. Tehtävillä tulee olla selkeät tavoitteet ja

niiden edistymisestä tulee saada riittävästi palautetta. Jos pelaaja on riittävän taitava, tehtävien tavoite on selkeä ja tehtävästä saa palautetta, pelaaja tuntee hallitsevansa tehtävää ja tilannetta. Tästä seuraa immersoituminen peliin, jolloin huoli itsestä katoaa ja ajantaju heikkenee. (Sweetser & Wyeth, 2005)

Jennett ym. (2008) esittävät immersiolle kolme tasoa. Ensimmäinen taso on sitoutuminen (engl. engagement), jolloin pelaajalla on halu käyttää aikaa, jakamista ja huomiotaan opetellakseen pelaamaan peliä ja tutustuakseen siinä käytettäviin ohjaimiin. Toinen taso on uppoutuminen (engl. engrossment), jolloin pelaajan tunteet vaihtelevat pelitilanteen mukaan eikä pelaaja enää ajattele hallitsevansa peliä peliohjaimen kautta vaan on sisällä pelissä. Täydellisessä immersoitumisessa (engl. total immersion) on kyse tilasta, jossa ollaan irtauduttu todellisuudesta ja pelin tapahtumat saavat pelaajan täydellisen huomion. (Jennett ym., 2008)

Kuvausten perusteella voidaan päätellä, että jo toisella immersion tasolla lähestytään flow-tasoista pelikokemusta. Immersio ei kuitenkaan ole flow'n synonyymi jo pelkästään sen takia, että immersiota on useamman asteista ja flow on tietty, täydellisen kokemuksen tila (Jennett ym., 2008). Lisäksi on huomattava, että Sweetserin GameFlow -mallin elementeistä muodostuu polku kohti immersiota ja flow-kokemusta, koska eri elementit ovat toisistaan riippuvaisia. Mikäli kaikki polun varrella esitetyt vaatimukset toteutuvat, saavutetaan suurella todennäköisyydellä flow-tila.

GameFlow -mallin elementit ovat keskittyminen, haaste, pelaajan taidot, hallinta, selkeät tavoitteet, palaute, immersio ja sosiaalisuus. Mallissa on määritetty kullekin elementille tavoite ja vaatimukset tavoitteen täyttymiselle. Sweetser ym. (2005) ovat koostaneet vaatimukset useista pelejä käsittelevistä lähteistä, joten seuraavien kappaleiden lähde on Sweetser paitsi kritiikin osalta, jolloin lähde mainitaan.

Pelin tulee vaatia keskittymistä ja pelaajan tulee voida keskittyä peliin. Mitä enemmän pelaaja joutuu keskittymään peliin, sitä paremmin peli pitää pelaajan otteessaan. Tulee kuitenkin huolehtia, ettei pelaajan työtaakka pelatessa kasva yli sopivan tason suhteessa havainnoinnin, ajatustoiminnan ja muistin rajoitteisiin. Ei myöskään pidä rasittaa pelaajaa tehtävillä, jotka eivät tunnu tärkeiltä. Näiden lisäksi tulisi peliin kuulumattomat toimet pitää mahdollisimman vähäisinä pelaamisen aikana (esimerkiksi asetusten säätäminen) ja häivyttää pelin käyttöliittymä mahdollisimman pieneksi, jotta varsinainen pelinäköymä hallitsee ruutua.

Haaste tunnustetaan johdonmukaisesti tärkeimmäksi osa-alueeksi hyvässä pelisuunnittelussa. Pelien tulisi olla riittävän haastavia suhteessa pelaajan taitoihin. Pelien tulisi olla vaikeustasoltaan vaihtelevia pelin edistyessä ja jaksoitukseltaan sopivia. Haasteen ja taitojen sovittaminen on olennaista, sillä taitojen suhteen liiallinen haastavuus aiheuttaa ahdistusta ja vähäinen haastavuus apatiaa. Pelaajien erilaisiin taitotasoihin vastatakseen peleissä tulisikin olla useita vaihtoehtoisia vaikeustasoja. Vaikeustasosta riippumatta pelin haasteellisuuden pitäisi vaihdella peliä pelattaessa kasvaen asteittain pelin edetessä ja pelaajan

taitotason noustessa. Pelin edetessä voidaan esitellä uusia pelielementtejä ja uudenlaisia haasteita sopivin väliajoin, jotta vaikeustaso säilyy sopivana.

Pelien tulee tukea pelaajan taitojen kehittymistä. Pelaajan taitojen kehittymistä voidaan tukea useilla erilaisilla tavoilla. Usein pelien alussa on opastusosio (engl. tutorial), jonka aikana pelaaja opetetaan pelin perustoiminnot. Vaihtoehtoisesti - tai opastusosion jatkeena - pelaaja voi oppia pelaamalla. Pelaaja oppii uusia taitoja ja harjoittaa niitä osana normaalia etenemistä pelissä. Palkitseminen on keskeinen osa pelatessa oppimista, ja palkintojen tulisikin vastata pelaajan panostusta. Pelien tulisi alkaa siten, ettei mitään ylimääräistä tietoa tarvita ennen pelaamisen aloittamista, koska pitkät selostukset tylsistytävät pelaajan ja ohjekirjojen tulisi tarjota tieto tiiviisti ja ymmärrettäväsi. Peleissä tulisi olla sisäänrakennettu apu- tai vinkkitoiminto, jotta pelaamista ei tarvitse keskeyttää. Se ei kuitenkaan saisi rikkoa immersiota. Muita tapoja pelaajan taitojen kehittymisen helpottamiseksi on noudattaa pelialustan konventioita ja pelisuunnittelun trendejä, jolloin toteutus on lähempänä pelaajan odotuksia. Myös toimintonäppäinten asettelun tulisi olla johdonmukainen ja pelin käyttöliittymän huomaamaton ja yksinkertainen.

Pelaajien tulee tuntea hallitsevansa omia toimiaan. Pelaajahahmon liikuttelun tulee tuntua luonnolliselta ja pelaajan tulee kyetä välittämään aikeensa pelimaailmaan sujuvasti. Myös käyttöliittymän ja toimintonäppäinten tulisi olla luontevat. Pelin aloittamisen, lopettamisen ja tilanteen tallentamisen tulee olla helppoa, koska ne tarjoavat pelaajalle mahdollisuuden edetä pelissä oman tahdinsa mukaan. Pelaajien ei tulisi olla mahdollista tehdä virheitä, jotka rikkovat pelin ja pelien tulisi auttaa pelaajia kiertämään mahdolliset virhetilanteet. Pelaajan hallinnan tunteelle on tärkeää nähdä heidän tekojensa vaikutus pelimaailmassa pysyvinä muutoksina, ja pelaajalle tulisi tarjota päätöstilanteita, joiden lopputulokset vaikuttavat maailmaan ja pelin läpäisyyn eri tavoilla. Pelaajalle tulisi olla mahdollista pelata peliä valitsemallaan pelityylillä, eikä hänen tulisi tuntea olevansa rajoitettu vain pelikehittäjien tarkoittamiin tapoihin pelata peliä. Pelien tulisi naamioida lineaarinen luonteensa mahdollisimman hyvin eikä pelin voittamiseen saisi olla yksittäistä, tehokasta strategiaa.

Pelien tulisi antaa pelaajalle selkeitä tavoitteita tarvittavissa kohdissa. Yleensä pelin taustatarina kerrotaan pelin alussa lyhyellä välianimaatiolla, joka myös esittelee pelaajalle pelin lopullisen päämäärän. Usein peleissä on myös rakenne, jossa tehtävänannoissa esitetään seuraavan tehtävän tavoitteet ja mahdolliset välitavoitteet seuraavalle peliosiolle yhdessä mahdollisten esteiden kanssa.

Pelaajien tulisi saada palautetta tekemisestään ja edistymisestään tehtävissä. Pelien tulisi tarjota välitöntä palautetta pelaajahahmon toimista, palautetta kokonaisvaltaisesta ja eri tehtävien edistymisestä. Onnistumisista ja epäonnistumisista tulee saada palautetta, onnistumisista tulee saada jonkinlainen palkinto ja epäonnistumisesta tulee tietää syy. Tietyissä tapauksissa pelissä on syytä olla erilaisia pisteytysjärjestelmiä pelaajan kannustamiseksi. Cowley huomauttaa, että palautejärjestelmä tulee hienosäätää pelikohtaisesti pelitestaamalla, koska yleispeitevää suunnitteluratkaisu tähän ei ole (Cowley ym., 2008).

Immersiossa pelaaja on syvästi mutta vaivattomasti uppoutuneena peliin. Immersio, uppoutuminen (engl. engagement) ja sulautuminen (engl. absorption) ovat käsitteitä, joista keskustellaan tiheään ja jotka ovat tärkeitä pelisuunnittelulle ja -tutkimukselle. Useissa tutkimuksissa on havaittu eritasoista immersoitumista pelejä pelatessa, ja näitä havaintoja on kuvattu juuri samoin kuin Csikszentmihalyin flow-kokemuksen todellisuudesta irtaantumista ja vaivatonta, syvää keskittyneisyyttä. Pelien tulisikin kaapata pelaajan koko keskittyminen ja saada hänet unohtamaan olevansa vuorovaikutuksessa laitteen ja käyttöliittymän kautta pelin kanssa. Immersiota luodaan muun muassa nokkelan äänimaailman ja kerronnan avulla. Kerronta tarjoaa pelaajalle juonen kulun ja taustat; valottaa keitä pelin hahmot ovat ja mitä pelissä tapahtuu saaden pelaajan tuntemaan itsensä osaksi tarinaa.

Pelikokemus on käyttäjäkokemus, jossa uutena osana on itse peli. MDA-mallin estetiikka-ajattelu tarjoaa meille käsityksen, mitä peli voi sisältää, ja muut osat havainnollistavat, mitkä pelin osat luovat estetiikan. GameFlow-malli tarjoaa näkemyksen siitä, miten sisältö tulee toteuttaa, jotta se on nautinnollista. Yhdessä mallit auttavat ymmärtämään, mistä pelikokemus koostuu.

2.5 Pelikehitys ja sen ominaispiirteet

Pelikehitykseen liittyy monia normaalista ohjelmistokehityksestä poikkeavia piirteitä. Ohjelmistokehityksestä pelikehityksen erottaa erilaiset tuotannon vaiheet ja lopputuotteen laatu. Pelikehitystä on verrattu elokuva- ja televisiosarjatuotantoon, sillä niiden lopputuote on viihdettä, ja myös samankaltainen luovuuden ulottuvuus on läsnä kaikkien tuotannoissa. (Manninen ym., 2006)

Manninen ym. (2006) ovat koostaneet kirjallisuuden pohjalta mallin, joka tiivistää pelikehitykselle tyypilliset vaiheet. Pelikehityksessä ei ole yhtä vallalla olevaa menetelmää tehdä pelejä vaan useimmiten pelistudiolla on oma, mukautettu menetelmänsä (Manninen ym., 2006). On kuitenkin havaittu, että ketterät menetelmät ovat suosittuja peliteollisuudessa (Musil, Schweda, Winkler & Biffel, 2010b). Manninen ei ota kantaa menetelmään, vaan esittää ainoastaan pelikehityksen eri vaiheet, jotka ovat pelikehityksessä läsnä menetelmästä riippumatta. Pelikehitys on usein iteratiivista (Potanin, 2010), ja pelikehityksessä iteraatiot ovat suunnittelu-toteutus-testaus -syklejä (Manninen ym., 2006). Nämä syklit voivat levittäytyä jopa useamman vaiheen välille, mikä tulisi ottaa huomioon suunnittelussa ja aikataulutuksessa (Manninen ym., 2006).

Mannista ym. (2006) mukailten pelikehityksen vaiheet ovat:

1. Konseptin valmistelu
2. Esituotanto
3. Tuotanto
4. Laadunvarmistus ja testaus
5. Julkaisu
6. Ylläpito

Pelikehitys lähtee liikkeelle peli-ideasta, jota saatetaan testata prototyypeillä ennen varsinaista kehitystä ja joka esitetään pelisuunnitteludokumentissa (Mussil, Schweda, Winkler & Biffel, 2010c). Peli-idea on pysyvä ajatus, joka säilyy läpi koko pelin kehityskaaren. Idea voi kuitenkin muuttua ja kehittyä (Manninen ym., 2006). Peli-idea kehittyy ja kypsyy jo *konseptia valmistellessa*; ideaa koetellaan, muokataan ja hiotaan (Manninen ym., 2006). Pelisuunnitteludokumentti (engl. game design document) on esituotantovaiheen tuotos, joka sisältää kaiken ydintiedon kehitettävästä pelistä niin ulkoasun, juonen, pää- ja sivuhenkilöiden kuin pelimekaniikan osalta (Callele, Neufeld & Schneider, 2005). Pelimekaniikka voidaan ymmärtää pelin määritelmässä mainituiksi säännöiksi. Tässä on eräs pelikehityksen keskeinen ongelma. Vaikka säännöt on luotu valmiiksi, ei voida tietää, millainen esteettisen kokemuksen varsinainen pelaaminen tuottaa. Pelikehityksessä lopputuote on vuorovaikutukseen perustuva viihdetuote. Vuorovaikutuksen suunnittelu ja toteutus voi olla hankalaa, pelikokemuksen suunnittelu on mahdotonta (Manninen ym., 2006).

Konseptin luontia seuraa *esituotanto*, jonka perusidea on suunnitella, testata ja arvioida kaikki mahdollinen ennen varsinaisen tuotannon aloittamista (Manninen ym., 2006). Parhaimmillaan esituotannon lopputulos on pelattava demo tai prototyyppi pelistä. Keithin (2010) mukaan keskeisimmät osat pelimekaniikasta tulisi olla valmiina esituotantovaiheen jälkeen. Esituotantoon kuuluu pelin vaatimusten tiivistäminen ja vaatimusten seurausten analysointi, jolloin karsitaan vaatimukset vastaamaan liiketoiminnan rajoitteita, sekä säilyneiden vaatimusten yksityiskohtainen suunnittelu niin teknisen toteutuksen, äänen kuin taiteen osalta (Manninen ym., 2006). Callelen ym. (2005) havaintojen mukaan peliteollisuudessa on merkittäviä ongelmia pelisuunnitteludokumentin muuttamisessa konkreettisiksi vaatimuksiksi. Heidän mukaansa ongelma saattaa piillä siinä, ettei peliteollisuudessa käytetä yleisesti hyväksytyjä vaatimusmäärittelyn periaatteita.

Esituotannossa syvennetään pelin maailmaa määrittämällä pelille lähtökohdan ja teeman, joiden taustoittamista varten tutkitaan kirjallisuutta ja elokuvia tai tehdään retkiä pelimaailmaa muistuttaviin ympäristöihin. Muita esituotannon toimia ovat markkinoiden ja kilpailevien tuotteiden analysointi, kehitystyökalujen rakentaminen ja käsikirjoituksen laatiminen ja hiominen, joka voi jatkua läpi pelin kehityskaaren. (Manninen ym., 2006)

Tuotantovaihe on raskain vaihe pelikehityksessä, jos ajatellaan aika- ja henkilöstöresursseja, sillä tuotantovaiheessa luodaan kaikki pelissä käytettävät audiovisuaaliset ja pelimekaaniset elementit. Tuotantovaiheessa luodaan konseptitaide, jonka perusteella luodaan niin pelin ympäristöt kuin kaikki ympäristöissä esiintyvät objektit, jotka voivat olla joko 3- tai 2-ulotteisia. Objekteille luodaan tarvittaessa animaatiot, ja peleissä on usein myös pelaamista rytmittäviä ja tarinaa kuljettavia välianimaatioita. Ympäristöt ja objektit tarvitsevat myös ääntä, suuri osa pelikehitystä on myös pelihahmojen vuorosanojen äänittäminen. Pelielementit sitoo lopulta yhteen kaiken taustalla ajettava ohjelmakoodi, joka luodaan tuotantovaiheessa. Yhdessä kaikkien edellä mainittujen tehtävien kanssa suoritetaan pelitestaamista. (Manninen ym., 2006)

Laadunvarmistus ja testaus läpäisevät koko pelikehityskaaren mutta muodostavat toisaalta alfa- ja beetatesteinä yhtenäisen vaiheen tuotannon ja julkaisu välissä. Pelin ohjelmakoodia testataan samoilla periaatteilla kuin ohjelmistotuotannossa, ja testaus voidaan jakaa yksikkö-, koonti-, järjestelmä-, konfiguraatio- ja taantumistestaukseen. Pelitestauksella tarkoitetaan pelin pelaamista. Se jaetaan alfa- ja beetatestaukseen, joista ensimmäistä tehdään siinä vaiheessa, kun peli on niin sanotusti ominaisuuksiltaan valmis, ja jälkimmäistä tehdään siinä vaiheessa, kun pelissä ei ole tiedossa olevia virheitä. Beetatesti voidaan jakaa suljettuun ja avoimeen, joiden erona on testaukseen osallistujien rajoittaminen. Testauksen yhteydessä löydetty virheet korjataan niiden kiireellisyyden ja vakavuuden perusteella. Korjauksia tehdessä on pidettävä huoli, ettei korjaus itsessään aiheuta lisää virheitä. (Manninen ym., 2006)

Julkaisu on pelikehityksen huippukohta. Julkaisuvaiheen tarkoituksena on viimeistellä peli ja toimittaa se julkaisijalle monistusta ja jakelua varten. Julkaisun valmistelussa, mahdollisesti jo tuotantovaiheen aikana, luodaan käyttäjälle tarkoitettu dokumentaatio, jakelua tukevat demot ja muu materiaali, tukipalvelut ja lokalisointi eri markkina-alueita varten. (Manninen ym., 2006)

Ylläpitovaiheessa julkaistaan korjauksia, jotka poistavat pelistä virheitä, korjaavat pelielementtien tasapainoa tai estävät huijaamista. Ylläpitovaiheessa voidaan myös julkaista uutta sisältöä peliin korjauksenomaisesti tai joko ilmaisena tai maksullisena lisäosana. Kehittäjät voivat myös osallistua edustustehtäviin peliään markkinoivissa tapahtumissa. Kehittäjien tulee myös huolehtia pelinsä yhteisöstä ja kuunnella yhteisön mielipiteitä. Myös kehittäjien oma yhteisö on kiinnostunut jakamaan kokemuksiaan pelikehityksestä ja oppimaan muiden kokemuksista. (Manninen ym., 2006)

2.6 Pelialan keskeiset toimijat

Pelialalla on kolme keskeistä toimijaa: pelikehittäjä, julkaisija ja kuluttaja. *Pelikehittäjä*, josta tässä tutkielmassa käytetään termiä *pelistudio*, on vastuussa pelin suunnittelusta ja kehityksestä. *Julkaisija* on osittain vastuussa pelikehityksen rahoittamisesta. Julkaisijan rooli on määritelty tarkemmin kehittäjän ja julkaisijan välisessä sopimuksessa. Yleensä julkaisija hoitaa myös pelin markkinoinnin, painatuksen ja jakelun. *Kuluttaja* on pelaaja tai muu asiakas, joka ostaa pelin tai pelipalvelun. Kuluttaja on harvoin suoraan yhteydessä kehittäjiin. (Manninen ym., 2006)

Manninen ym. (2006) tunnistaa pelialalla olevan myös muita toimijoita. Lähemmässä tarkastelussa voidaan havaita, että alalla vaikuttavat myös alustat tai *pelimoottorikehittäjä* (engl. platform developer), rahoittaja, jakelija ja jälleenympäjä. *Alustakehittäjällä* tarkoitetaan pelilaitteen, kuten konsolin, tai pelimoottorin kehittäjää. Valmiin pelimoottorin hyödyntäminen helpottaa pelikehittäjän työtä, kun tämä voi keskittyä varsinaisen pelisisällön luomiseen eikä tarvitse kehittää matalan tason teknologiakeskeisiä ratkaisuja. *Rahoittaja* voi olla mikä tahansa kolmas osapuoli, joka sijoittaa rahaa pelikehitykseen. *Jakelija* huolehtii

pelin jakelusta kuluttajille tai jälleenmyyjille sekä saattaa osallistua pelikehityksen rahoittamiseen. *Jälleenmyyjät* ostavat pelejä julkaisijalta tai jakelijalta ja myyvät niitä edelleen voitolla kuluttajille.

Pelikehitys on ammattina riskaabeli, hektinen ja hyvin kilpailuhenkinen. Se houkuttelee pariinsa ihmisiä, jotka pitävät peleistä, ja usein pelikehittäjät rakastavat työtään. Pelikehitys koostuu suunnittelusta, ohjelmoinnista, audiovisuaalisen materiaalin tuottamisesta sekä testaamisesta. Sitä verrataan usein elokuvateollisuuteen taiteellisten, luovien osa-alueidensa vuoksi. Pelistudio arvioidaan pelikonseptin, kehitystiimin, prosessimallin, budjetin hallinnan ja laadunvarmistuksen perusteella. Pelistudion kaikkien osa-alueiden tulee olla laadukkaita. Sopimusten solmimisen kannalta tärkeitä ovat vakuuttavat referenssit eli pelistudion aiemmat pelit. (Manninen ym., 2006)

Pelistudio kehittää peliä julkaisijan maksamien rojaltien tai ennakkomaksujen varassa, koska sillä itsellään ei usein ole riittävästi pääomaa, jakelukanavia, markkinointiresursseja tai halua julkaista peliä itsenäisesti. Tämän seurauksena pelistudiot eivät yleensä itse päättä, millaista peliä he kehittävät, ja he ovat julkaisijoiden armoilla, mikäli nämä päättävät peruuttaa pelin kehityksen (Manninen ym., 2006; Potanin, 2010). Pelistudiot eivät myöskään usein omista tekijänoikeuksia peliin, vaan ne ovat julkaisijan hallussa muutamien suurien pelistudioiden poikkeuksia lukuun ottamatta. Pelistudiot eivät siis yleensä myy tuotettaan suoraan kuluttajille, mutta kuluttajat ovat osana pelikehitystä beetatestien ja kehitystä seuraavan pelaajayhteisön palautteen muodossa. (Manninen ym., 2006)

Pelistudiot vastaavat pelikehityksestä, mutta usein julkaisija, joka rahoittaa kehityksen, määrää millainen peli on kehitettävä ja milloin pelin tulee olla valmis (Kanode & Haddad, 2009; Potanin, 2010). Julkaisija rahoittaa pelejä ja pyrkii tekemään voittoa niitä julkaisemalla (Manninen ym., 2006). On siis luonnollista, että julkaisija huolehtii pelin markkinoinnista, PR-kampanjoista, monistuksesta, jakelusta ja peleihin liittyvästä asiakaspalvelusta (Manninen ym., 2006).

Julkaisija kantaa suuren osan taloudellisesta vastuusta, joten he vaativat pelistudioilta laadukkaita pelejä, jotka ovat aikataulun mukaisesti valmiita. On kuitenkin havaittu, että vain pieni osuus kaikista peleistä tuottaa voittoa, minkä seurauksena julkaisijat pyrkivät tuottamaan pelistudioilla suosittujen genrejen pelejä tai hyvin myyneiden pelien kaltaisia pelejä. Julkaisija ja kehittäjä pyrkivät minimoimaan riskit tekemällä suosituille peleille jatko-osia tai matkimalla muita menestyviä nimikkeitä. Tämä näkyy pelistudioiden suuntaan entistä rajoituneempina mahdollisuuksina innovoida ja luoda uutta. Julkaisijoiden toimintatapa on ristiriidassa suhteessa tutkimushavaintoihin, joiden mukaan ne harvat uudet nimikkeet, jotka julkaistiin saivat parempia arvosanoja peliarvosteiluissa, joiden puolestaan on havaittu korreloivan pelin menestyksen kanssa. (Potanin, 2010)

Tiukka julkaisija-kehittäjä-suhde on kuitenkin muuttumassa internetin pelijakelupalveluiden myötä. Pelijakelupalvelut, kuten Steam, Gamersgate, Impulse, Good Old Games sekä konsoli- ja puhelinvalmistajien alustakohtaiset

nettijakelupalvelut ja muut vastaavat, myyvät pelejä digitaalilatauksina kuluttajille. Tällaisessa tapauksessa suuri osa normaalin monistamisen ja jakelun kuluista sekä voiton jakavista osapuolista jää pois. Pelialalla vallinnut "suuruuden ekonomia" ei muutoksen myötä pakota kehittämään sisällöltään niin suuria pelejä. Alentuneiden kustannusten myötä pelikehittäjät voivat kehittää pelejä ilman perinteistä julkaisijaa suoraan pelijakelupalvelussa jaeltavaksi, jolloin tekijänoikeudet jäävät kehittäjille itselleen ja mahdollistavat sen hyödyntämisen myös jatkossa (Kuittinen ym., 2010). Eräs tällainen menestystarina on suomalaisen Rovio Mobilen Angry Birds -peli (Kuittinen ym., 2010).

Kuluttajien eli pelaajien osalta peliteollisuus poikkeaa ohjelmistoteollisuudesta, koska pelejä ei pelata tai hankita käytännöllisistä syistä. Peliteollisuus onkin osa viihdeteollisuutta enemmän kuin ohjelmistoteollisuutta. Kuluttajien ostokäyttäytymistä ohjaa eniten kunkin oma maku mutta markkinointi on keskeisessä asemassa kuluttajien tietoisuuden kasvattamisessa. Paraskaan peli ei myy, jos kukaan ei ole koskaan kuullut siitä. Myös tiettyjen pelistudioiden osallistuminen kehitykseen saattaa vaikuttaa ostokäyttäytymiseen. (Manninen ym., 2006) Sotamaa (Sotamaa, 2007) on tutkinut pelaajatyyppisiä ja huomauttaa, että pelistudioille voisi olla hyödyllistä ottaa pelikehitykseen mukaan pelaajia vielä kiinteämmin kuin nykyään pelitestaajina.

2.7 Pelien julkaisumallit

Pelin *julkaisumallilla* tarkoitetaan tapaa, jolla pelillä on tarkoitus tuottaa voittoa. Aiheesta on vaikea löytää tieteellisiä artikkeleita eikä pelikehittäjien internetyhteisöissäkään juuri keskustella asiasta. Ainoa löytämäni tieteellinen lähde (Grigg, 2005) käsittelee pelin julkaisemista osissa tv-sarjan tapaan, mikä on julkaisumallina saavuttanut jonkinlaista suosiota viime vuosina. Internetissä on myös julkaistu jonkin verran artikkeleita liittyen satunnaispelaajille suunnattujen pelien ansaintamalliin. Tämän vuoksi esittelenkin havaitsemani vallalla olevat julkaisumallit yhden pelinimikkeen kautta ja koetan valottaa pelin tyyppin, tuotannon ja pelistudion avulla, miksi julkaisumalli on valittu ja mitä mahdollisia etuja valinnalla saavutetaan.

Ennen internetin yleistymistä pelit julkaistiin aina fyysisessä muodossa ja niitä hankittiin yleensä kauppojen hyllyistä valitsemalla ja ostamalla. Tätä nykyä on todennäköisesti poikkeus, mikäli peli julkaistaan ainoastaan fyysisessä muodossa. Lähes kaikki pelit ovat saatavilla niin kaupan hyllyllä kuin internetissä digitaalisena. On epäselvää, mikä on hyllymyynnin ja digijakelun suhde, koska digijakelusta ei ole luotettavaa, kattavaa tietoa saatavilla.

Yleinen ansainta- ja julkaisumalli on perinteisen hyllymyynnin uusi muoto, jossa kertamaksua vastaan saa joko pelin fyysisen tai digitaalisen kopion, minkä jälkeen kuluttaja omistaa pelin ja voi pelata sitä periaatteessa rajoittamattomasti. Usein julkaisusta vastaa erillinen julkaisija, joka on sopinut pelin kehityksestä ja rahoituksesta pelistudion kanssa.

Eräs tunnettu räiskintäpeli, joka on julkaistu tätä mallia hyödyntäen, on Call of Duty: Black Ops. Pelin kehitti Treyarch, jolla on noin 250 työntekijää (Wikipedia, 2011g), ja julkaisi Activision (Wikipedia, 2011b). Peli on saatavilla niin hyllymyynnissä kuin pelijakelupalvelu Steamissa (Wikipedia, 2011b). Call of Duty edustaa suosittua peligenreä, ja nimike on pelaajien keskuudessa kuuluisa. Pelistudion koko viittaa korkeisiin kehityskuluihin ja odotukseen laadusta. Luultavasti Activision myös panosti pelin markkinointiin voimakkaasti.

Edellisen julkaisumallin laajennus ovat kuukausimaksulliset pelit, jotka ovat yleensä massiivisia verkkoroolipelejä (engl. MORPG). Julkaisumallin idea on myydä peli pelaajille ja tämän jälkeen myydä peliaikaa kuukausi tai muu vaihtoehtoinen ajanjakso kerrallaan. Kerätyllä rahalla rahoitetaan pelin jatkokehitystä, palvelimien ylläpitoa ja muita toimintoja, joita kyseisen genren peleihin liittyy. Osa rahasta on tietysti voittoa pelistudiolle ja julkaisijalle. Kuukausimaksullisuuden vuoksi pelin tulee olla erityisen koukuttava ja tarjota riittävästi sisältöä. Usein uutta sisältöä julkaistaankin tasaisin väliajoin ostettavina laajennusosina ja myös näiden laajennusosien välissä peliin julkaistaan muutoksia päivitysten muodossa. Julkaisumalli on haastava ja useat mallia hyödyntävät massiiviset verkkoroolipelit, kuten Tabula Rasa, ovatkin epäonnistuneet ja sulkeneet palvelimet pelaajilta (Wikipedia, 2011f).

Kuuluisin edellä mainittua julkaisumallia ja genreä edustava peli lienee Blizzard Entertainmentin julkaisema ja kehittämä World of Warcraft. Blizzard Entertainmentilla on noin 4600 työntekijää (Wikipedia, 2011a). Vuoden 2009 Game Developers Conferencessä pidetyn esitelmän uutisoinnin mukaan varsinaisen kehitystiimin koko oli tuolloin noin 450 henkeä, mikä oli tuolloinkin vain murto-osa peliin liittyviä toimintoja tuottavasta ja ylläpitävästä henkilöstöstä (Gamasutra, 2009). Massiivisia verkkoroolipelejä pidetään suurimpina, monimutkaisimpina projekteina, joita pelialalla tällä hetkellä toteutetaan (Blow, 2004).

Nettijulkaisujen porkkana on alentuneet kustannukset, koska fyysisten kopioiden painattaminen ja jakelu jäävät pois. Tähän mennessä ei vielä olla nähty Call of Dutya vastaavan mittakaavan peliä, joka olisi julkaistu pelkästään pelijakelupalveluissa. Voidaankin väittää, että pelkästään digitaalisessa muodossa jaeltavien pelien budjetti - ja myyntihinta per kappale - ovat pienempiä kuin perinteistä mallia käyttävillä peleillä. Tällöin myöskään pelaajien odotukset peliä kohtaan eivät ole samaa luokkaa.

Eräs lähinnä pelijakelupalveluissa myyty peli on Trine, joka on suomalaisen Frozenbyten kehittämä. Pelin julkaisusta vastasivat eri tahot eri puolilla maailmaa. PC:llä Frozenbyte sopi jakelusta suoraan jakelupalveluiden kanssa. Playstation3 -konsolin PSN -jakelupalveluun Euroopassa ja Yhdysvalloissa pelin julkaisi Nobilis Games, Japanissa Square Enix Downloadable Games. Pienessä mittakaavassa peli julkaistiin myös hyllymyyntiin, mistä vastasi Nobilis Games Euroopassa ja SouthPeak Games Yhdysvalloissa. Frozenbyten kehitystiimi oli projektin alussa kolme henkeä ja parhaimmillaan studio työllisti 16 henkeä. Pelin budjetti oli 850 000 € ja kehitys kesti noin 18 kuukautta. (Hyvärinen & Kinnunen, 2010) Näyttäisi siis siltä, että pääasiassa digitaalijakeluun tar-

koitettujen pelien tuotanto ja budjetti ovat pienempiä kuin myös fyysisesti julkaistavilla peleillä.

Digitaalijakelun erikoistapaus ovat ilmaisapelit. Yleensä ne ovat tuotannoltaan ja budjetiltaan lähes samaa luokkaa kuin muutkin pelijakopalveluiden pelit. Pelin hankkiminen ja pelaaminen on yleensä ilmaista, joten pelistudio ja -julkaisija ansaitsevat joko peliin liittyvillä mainostuloilla tai pelielementtien myynnillä eli niin sanotuilla mikromaksuilla. Jälkimmäisellä tarkoitetaan joko peliä helpottavien esineiden, täysin pelimekaniikasta irrotettujen, visuaalisten elementtien, kuten pelaajahahmon nahkojen (engl. skin), tai muulla tavoin peliin liittyvien, pelaajaa ilahduttavien elementtien myyntiä. Toisinaan digitaalijakelussa näkee "maksu omantuntosi mukaan" -kampanjoita (<http://www.humblebundle.com/>), joilla yksityiset pelikehittäjät ovat saavuttaneet huomattavia parannuksia sekä kappalemääräiseen myyntiin että myynnin rahalliseen arvoon.

Riot Gamesin PC:lle kehittämä ja julkaisema League of Legends on ilmaisopeli, jonka ansaintamalli perustuu mikromaksuihin. Kuka tahansa voi ladata pelin ja pelata sitä ilmaiseksi. Lisämateriaalin ostaminen mikromaksuilla on täysin vapaaehtoista. League of Legends on usean pelaajan reaaliaikainen strategiapeli, jossa pelaajat ovat jakautuneet kahteen joukkueeseen. Peliä pelataan ainoastaan internetissä. Riot Games työllistää internetsivujensa mukaan 71 työntekijää, joista noin 40 henkeä osallistui varsinaiseen pelikehitykseen loppujen huolehtiessa pelipalvelinten ylläpidosta, asiakaspalvelusta, yhteisöviestinnästä ja markkinoinnista. Peliä kehitetään ja ylläpidetään edelleen. (Riot Games, 2010) League of Legendsin takana olevan tiimin koko saattaa olla hieman suurempi kuin muilla ilmaispeleillä, koska Riot Games vastaa itse myös kaikista julkaisijan toimista. Ilmaispelein käsite on myös hyvin laaja; sen alle mahtuu tuotannoltaan monen kokoisia pelejä, joten esimerkki ei ole kovin hyvin yleistettävä.

Viimeinen käsiteltävä julkaisumalli on pelin julkaiseminen jaksoissa televisiosarjan tapaan. Episodipohjainen julkaiseminen (engl. episodic gaming) on saavuttanut jonkinasteista suosiota 2000-luvulla. Tällä tarkoitetaan sarjaa pelijulkaisuja, jotka toimivat myös itsenäisinä peleinä ja jotka lopulta yhdessä muodostavat yhden pelin sekä tarinallisesti että pelillisesti. Yksittäisen jakson hinta on yleensä noin 5 - 10 €, ja jaksoja pyritään julkaisemaan tasaisin, noin kolmen kuukauden, väliajoin. Yleensä jaksoittainen julkaisu tapahtuu pelijakelupalveluiden kautta ja ylipäätään digitaalisena. (Grigg, 2005) Tähän mennessä jaksoina julkaistuista peleistä on nähty myös kaikki jakson yhteen pakettiin ko koavia kausina, joita on saatettu myydä myös hyllyjulkaisuina. Pelit-lehdessä noteeratuista episodipohjaisista julkaisuista kaikki ovat tähän mennessä kuuluneet seikkailupelien genreen. Syy tähän on luultavasti pelityypin sopivuus jaksoittaiseen julkaisuun, sillä ne nojaavat kerrontaan ja tarinaan lähes yhtä paljon kuin televisiosarjat.

Telltale Games on pelistudio, joka on keskittynyt pääasiassa kehittämään seikkailupelejä, jotka julkaistaan episodipohjaisesti (Wiltshire, 2009). Studion Sam & Max Save the World -pelin jaksot julkaistiin PC:lle aluksi pelistudion

itsensä ja GameTap -jakelijan toimesta digitaalijakeluna ja myöhemmin Pohjois-Amerikassa The Adventure Companyn sekä Euroopassa JoWood Productionsin toimesta hyllyjulkaisuina (Wikipedia, 2011e). Peli on julkaistu myös Wii- ja Xbox360-konsoleille (Wikipedia, 2011e). Pelin kehitystiimi oli kooltaan noin 30 henkeä (Wiltshire, 2009), ja jaksot julkaistiin noin kuukauden välein (Wikipedia, 2011e).

2.8 Yhteenveto

Tässä luvussa kuvattiin pelejä ja peliteollisuutta. Ensin käsiteltiin pelejä harrastuksena, ja huomattiin peliharrastuksen jatkuvasti yleistyvän sekä peliharrastajien keski-ään kasvavan. Aliluvussa 2.2 esiteltiin pelin määritelmä, johon tässä tutkielmassa tukeudutaan. Tämän jälkeen tehtiin katsaus pelien jaottelutapoihin, joita on kaksi: peligenret ja jaottelu pelitypologian avulla. Neljäs kokonaisuus oli pelikokemus, johon pureuduttiin esittelemällä MDA-malli, joka erottaa pelikokemuksen mekaniikan, dynamiikan ja estetiikan yhteensulautumaksi, ja GameFlow-malli, jossa pyritään selittämään hyvän pelikokemuksen vaatimukset Flow-teoriaan nojautuen. Pelikokemuksesta siirryttiin käsittelemään varsinaista pelikehitystä, josta esiteltiin Mannisen ym. (2006) havaitsemat vaiheet ja niiden sisältö. Toiseksi viimeinen kokonaisuus oli pelialan keskeiset toimijat, joita ovat pelikehittäjä, julkaisija ja kuluttaja. Viimeiseksi käytiin lävitse tutkijan havaitsemat pelijulkaisumallit ja kerrottiin taustoja kustakin julkaisumallia hyödyntäneestä esimerkkipelistä.

3 KETTERÄT MENETELMÄT

Tässä luvussa esitellään ketterien menetelmien tausta Agile-manifestin avulla, määritellään ketterän menetelmän käsite ja kuvataan tutkielman piiriin kuuluvat ketterät menetelmät. Menetelmät ovat Scrum, XP, Lean ja Kanban. Ne on valittu sen perusteella, että ne esiintyivät ketterää pelikehitystä käsittelevässä kirjallisuudessa.

3.1 Agile-manifesti

Agile-manifestin takana on joukko ohjelmisto- ja menetelmäkehittäjiä, jotka olivat tahoillaan luoneet kevyempiä kehittämismenetelmiä vastatakseen ohjelmistokehityksen haasteisiin, joihin perinteiset dokumentointia painottavat, raskaat kehitysmenetelmät eivät kyenneet vastaamaan, sekä palauttaakseen ”ihmisarvon” ohjelmistokehitykseen (Agile Alliance, 2001). On huomattava, että ketterät menetelmät olivat olemassa ennen manifestia. Agile-manifestissa esitetään neljä arvoa ja kaksitoista periaatetta, joiden tulisi toteutua ketterässä kehityksessä (Agile Alliance, 2001).

Manifestin arvot esitetään seuraavalla tavalla. Manifestin luojat arvostavat:

Yksilöitä ja vuorovaikutusta enemmän kuin prosesseja ja työkaluja.

Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentointia.

Yhteistyötä asiakkaan kanssa enemmän kuin sopimusneuvottelua.

Muutoksiin reagoimista enemmän kuin suunnitelman seuraamista.

Arvojen esityksessä on huomattava niiden molemmat puolet. Myös oikealla olevilla asioilla on arvoa, vaikka manifestin laatijat arvostavat vasemmalla olevia käsitteitä enemmän. Manifestiin kuuluu edelleen kaksitoista periaatetta, joista kukin tukee jotakin edellä esitetyistä neljästä arvosta. (Agile Alliance,

2001) Tämän tutkielman puitteissa ei ole oleellista esittää periaatteita eikä näiden sidoksia arvoihin.

Nykyään ketterät menetelmät ovat yleisesti käytössä ohjelmistoteollisuudessa ja niiden käytänteitä pyritään hyödyntämään sopivissa tilanteissa. VersionOnen rahoittaman ja Analysis.Net Researchin vuonna 2010 toteuttaman kyselytutkimuksen mukaan 90% kaikista kyselyyn osallistuneista ohjelmistokehitysalalla työskentelevistä henkilöistä työskentelivät organisaatioissa, joissa käytettiin ketterien menetelmien käytänteitä ainakin jossain määrin. (Analysis.Net Research, 2010)

3.2 Ketterien menetelmien määritelmä

Highsmithin ja Cockburnin (2001) mukaan ketteryydessä on kyse muutoksen luomisesta ja muutokseen vastaamisesta. Heidän mukaansa ketterissä menetelmissä ei ole uutta niissä esiintyvät käytänteet, vaan ihmisten tunnistaminen menestymisen avaintekijöiksi yhdessä tehokkuuteen (engl. effectiveness) keskittymisen ja ohjattavuuden (engl. maneuverability) kanssa. Edellisistä saadaan yhdistelmä arvoja ja periaatteita, jotka määrittävän "ketterän" maailmankatsomuksen. (Highsmith & Cockburn, 2001)

Agile-manifesti ei määrittele tarkasti, millainen ketterä menetelmä on. Sen sijaan manifestin esittelemät arvot, ja käytänteet voidaan ymmärtää suuntaviivoiksi, joiden sisällä toimia. Arvot ovat suuren mittakaavan kriteerit, joiden pohjalta voidaan arvioida näkemänsä, ajattelemansa ja tekemänsä asiat (Beck & Andres, 2004). Periaatteet ovat arvojen alakohtaiset kriteerit (Beck & Andres, 2004). Manifestissa ei määritellä käytänteitä, jotka puolestaan ovat arvoja ja periaatteita kuvastavia jokapäiväisiä toimia (Beck & Andres, 2004).

Abrahamsson, Salo, Ronkainen ja Warsta (2002) tutkivat ketteriä menetelmiä tarkoituksenaan muun muassa esittää määritelmä ketterälle lähestymistavalle. Heidän mukaansa inkrementaalisuus, yhteistyö, helppous ja mukautuvuus määrittelevät menetelmän ketteräksi. *Inkrementaalisuudella* tarkoitetaan, että ohjelmisto julkaistaan pieninä osina, jotka kehitetään nopeissa sykleissä. *Yhteistyöllä* tarkoitetaan, että asiakas ja kehittäjät työskentelevät jatkuvasti yhdessä tiiviisti kommunikoiden. *Helppoudella* tarkoitetaan, että menetelmä itsessään on helppo oppia sekä muokata ja se on hyvin dokumentoitu. *Mukautuvuudella* tarkoitetaan, että menetelmä mahdollistaa viime hetken muutokset ja muutoksiin reagoimisen ylipäättään. (Abrahamsson ym., 2002)

Määritelmä on osuva tässä tutkielmassa käsiteltävien, alunperinkin ketterien menetelmien perheeseen kuuluvien menetelmien osalta. Näitä ovat XP (Beck & Andres, 2004) ja Scrum (Schwaber & Beedle, 2001). Vaikka Lean (Poppendieck & Poppendieck, 2003) ja Kanban (Kniberg & Skarin, 2009) eivät Abrahamssonin ym. (2002) määritelmään sovikaan ovat ne tämän tutkielman puitteissa oleellinen osa ketteriä menetelmiä, ja ideologisesti lähellä Agile-manifestia. Molemmat ovat pelikehitystä käsittelevässä kirjallisuudessa esiintyneet enemmänkin XP:tä ja Scrumia vahvistavina menetelminä. On myös ole-

massa yhdistelmämenetelmiä, kuten Scrumban (Ladas, 2009), joka on Scrumin ja Kanbanin yhteen sulauma. Seuraavissa alaluvuissa kuvataan Scrum, XP, Lean ja Kanban.

3.3 Scrum

Scrum on kehitetty ohjelmistokehitysprosessin hallinnointiin (Schwaber & Beedle, 2001). Se ei ota kantaa siihen, millä tekniikoilla varsinaista kehitystyötä tulisi tehdä (Abrahamsson ym., 2002). Termi "scrum" on lainattu rugbyista, jossa se tarkoittaa pelaamattomissa olevan pallon hankkimista pelattavaksi tiimityön voimalla ((Schwaber & Beedle, 2001).

Scrumissa on arvot ja käytänteet, joihin lasketaan tämän tutkielman puitteissa niin käytänteet, työkalut kuin artefaktit, sekä roolit. Scrumin arvot ovat Scrumia hyödyntävässä organisaatiossa ilmentyviä arvoja, joita ei välttämättä ole havaittavissa ennen prosessimallin käyttöönottoa (Schwaber & Beedle, 2001). Ne muodostuvat kuitenkin väistämättä Scrum-organisaation arvoiksi (Schwaber & Beedle, 2001). Seuraavassa kuvataan Scrumin arvot, käytänteet, roolit ja prosessimalli.

3.3.1 Arvot

Schwaber ja Beedle (2001) puhuvat Scrumin arvoista Scrumin hyödyntämisen myötä ilmaantuvina arvoina sen sijaan, että ne olisivat edellytys Scrumille. Heidän mukaansa Scrumin arvot tulevat organisaation arvoiksi vääjäämättä, vaikka Scrum perustuukin näillä samoille arvoille. Scrumin arvoiksi he tunnistavat seuraavat: sitoutuminen (engl. commitment), keskittyminen (engl. focus), avoimuus (engl. openness), kunnioitus ja rohkeus.

Sitoutumisella tarkoitetaan sitoutumista tiettyyn tavoitteeseen ja selkeiden rajojen tai määritelmän asettamista tavoitteelle. Scrumin perusidea on helpottaa ihmisiä sitoutumaan tavoitteisiin ja tarjota heille työkalut, joilla he voivat saavuttaa tavoitteensa. (Schwaber & Beedle, 2001)

Keskittyminen on paneutumista olennaiseen. Scrumissa tiimi määrittelee tavoitteen, johon se saa rauhassa pyrhdyksen ajan keskittyä ilman häiriöitä. Ylipäättään työskennellessä tulee olla keskittynyt ainoastaan käsillä olevaan ongelmaan. Niinpä työympäristöstä tulee karsia keskittymistä vaikeuttavat tekijät pois. Ohjelmistokehitys on riittävän haastavaa ilman turhia häiriötekijöitäkin. (Schwaber & Beedle, 2001)

"Scrumissa kaikki projektiin liittyvä on näkyvää kaikille (Schwaber & Beedle, 2001, s. 151)." *Avoimuus* on läsnä kaikissa Scrumin käytänteissä, ja Scrumissa pyritään tarjoamaan riittävästi totuudenmukaista tietoa projektin tilasta jokaisena ajan hetkenä (Schwaber & Beedle, 2001).

Kunnioitus on oleellista, jos tiimin halutaan toimivan. Scrumissa tiimi on vastuussa tavoitteen saavuttamisesta. Näin ollen vaikeinakin hetkinä tiimin

tulee yhdessä ratkaista ongelmat eikä sortua syyttämään yksittäistä henkilöä, joka ei ole kyennyt samanlaiseen suoritukseen kuin muut. Kaikki tekevät parhaansa ja mikäli se ei riitä, tiimin tulee yhdessä miettiä, miten ongelma ratkaistaan. (Schwaber & Beedle, 2001)

Scrum vaatii *rohkeutta*, sillä se on täysin erilainen työympäristö kuin mihin ollaan totuttu. Scrum-organisaation arvot ovat erilaiset, ja vaatii rohkeutta havaita, että ympäristö todella toteuttaa arvoja. Vielä enemmän rohkeutta vaatii luottaa omaan harkintaansa. Se on kuitenkin oleellista Scrumissa, sillä tiimillä ja sen yksilöillä on valta päättää asioista. (Schwaber & Beedle, 2001)

3.3.2 Käytänteet

Scrumin käytänteet ovat kehittyneet, kun niitä on ajan mittaan hyödynnetty tuhansissa ohjelmistokehitysprojekteissa (Schwaber & Beedle, 2001). Voidaan väittää, että käytänteet ja niiden yhteistoiminta ovat Scrum-menetelmä. Yksittäinen käytänte ei todennäköisesti ole hyödyllinen. Seuraavaksi kuvataan Scrum-käytänteet.

Tuotteen työlista (engl. product backlog) on "kehittyvä, järjestetty jono liike-toiminnallista ja teknistä toiminnallisuutta, joka pitää toteuttaa ohjelmistona (Schwaber & Beedle, 2001, s. 32)." Se on priorisoitu lista kaikista tehtävistä ja työstä, jotka tulee tehdä ohjelmiston tuottamiseksi. Kehitystyön aloittamiseksi tarvitaan tehtäviä vain ensimmäisen pyrhdyksen tarpeisiin. Tuotteen työlista ei ole koskaan valmis johtuen muuttuvista vaatimuksista ja itse listan kehityvästä luonteesta. Lista on olemassa niin kauan kuin ohjelmistoa kehitetään. Listan vaatimukset voivat tulla mistä tahansa, mutta listaa hallinnoi *tuotteen omistaja* (engl. product owner). Listan vaatimuksien kehittämiseen kuluva aika tulee arvioida jokaiselle vaatimukselle erikseen aloittaen tärkeimmistä vaatimuksista. Arvioinnin tekee *Scrum-tiimi* yhdessä tuotteen omistajan kanssa. (Schwaber & Beedle, 2001)

Päivittäinen Scrum -tapaaminen (engl. daily Scrum meeting) on jokaisena työpäivänä pidettävä lyhyt, 15-minuuttinen kokous, jonka tarkoituksena on parantaa kommunikaatiota, sitouttaa työhön ja tarjota ajantasaista tietoa työn edistymisestä. Siihen osallistuvat kehittäjät ja Scrum -mestari sekä kuuntelijoina kaikki halukkaat. Päivittäinen Scrum alkaa joka päivä samaan aikaan samassa paikassa. Tapaamisessa jokainen kehittäjä puhuu vuorollaan vastaten kolmeen kysymykseen: "Mitä olen tehnyt viime tapaamisen jälkeen?", "Mitä teen ennen seuraavaa tapaamista?" ja "Oliko työskentelylläni esteitä tai ongelmia?". Scrum-mestarin tehtävä on kirjata esteet ja poistaa ne. Mikäli Scrum-tapaamisessa ilmenee tarvetta, voidaan sopia tapaaminen välittömästi Scrum-tapaamisen jatkoksi, jossa voidaan keskustella ja suunnitella välittämättä Scrum-tapaamisen tiukasta protokollasta. Jatkotapaamiseen osallistuvat vain ne henkilöt, joiden ehdottomasti täytyy, jolloin muut voivat jatkaa työskentelyä. (Schwaber & Beedle, 2001)

Pyrähdys (engl. sprint) on ajanjakso, jonka aikana tiimi työskentelee tuotukseen sovitun palan ohjelmistoa. Pyrhdyks voi olla kestoaltaan viikosta kuu-

kauteen tai jopa kahteen. Tuona ajanjaksona Scrum-tiimi on vastuussa ohjelmiston kehityksestä. Tavoitteet eivät voi muuttua ulkoisten tekijöiden takia kesken pyrähdysten, ja tiimille on suotava työskentelyrauha. (Schwaber & Beedle, 2001)

Ennen pyrähdystä sen tavoite ja sisältö sovitaan *pyrähdyksen suunnittelutapaamisessa* (engl. sprint planning meeting), johon osallistuvat tiimi, Scrum-mestari, tuotteen omistaja ja asiakas sekä muut sidosryhmät tarpeen mukaan. Se on jaettu kahteen erilliseen tapaamiseen. Ensimmäisessä suunnittelutapaamisessa asetetaan pyrähdykselle tavoite ja päätetään ohjelmistoon tuotettavasta toiminnallisuudesta. Käytännössä valitaan tuotteen työlistalta pyrähdykseen sopiva määrä tehtäviä. (Schwaber & Beedle, 2001)

Toisessa suunnittelutapaamisessa tiimi suunnittelee, miten se toteuttaa pyrähdykseen valitun toiminnallisuuden. Tuotteen työlistan tehtävät pilkotaan pienemmiksi tehtäviksi, jotka asetetaan pyrähdysten työlistalle (engl. sprint backlog). Kun pyrähdysten työlistalla olevat tehtävät on tehty, pitäisi pyrähdysten tavoitteen olla täyttynyt. Pyrähdysten työlistaan voi pyrähdysten aikana tulla muutoksia, jos havaitaan, että jokin tehtävä on jäänyt listaa laadittaessa huomaamatta tai listalle on pantu tehtäviä, joita ei tarvitsekaan toteuttaa pyrähdysten tavoitteen täyttymiseksi. (Schwaber & Beedle, 2001)

Pyrähdyksen aikana työn edistymistä seurataan *pyrähdyksen edistymiskäyrän* (engl. sprint burndown) avulla. Käyrä alkaa pyrähdysten tehtävien työmääräarvion yhteenlasketusta summasta, ja sitä päivitetään päivittäin jäljellä olevan työmääräarvion mukaan. Kun kaikki työ on tehty, käyrä osoittaa, että jäljellä olevien tehtävien työmäärä on nolla. Tästä tulee käyrän englanninkielinen nimitys ”burndown”. Käyrän kulmakerroin vaihtelee sen mukaan, miten eri tehtävien työmääräarviot tarkentuvat, havaitaanko uusia tehtäviä, jotka ovat välttämättömiä tavoitteen saavuttamiseksi, ja saadaanko tehtäviä valmiiksi. (Schwaber & Beedle, 2001)

Pyrähdyksen lopussa pidetään *pyrähdyksen katselmointi* (engl. sprint review). Katselmoinnissa Scrum-tiimi esittelee pyrähdyksessä tuottamansa palan ohjelmistoa johdolle, asiakkaille, käyttäjille ja tuotteen omistajalle, joiden tehtävä on arvioida tuotosta. Katselmoinnissa tiimi kertoo, miten se tuotti kyseisen palan ohjelmistoa, millaisia ongelmia tai ratkaisuja se joutui tekemään ja mihin suuntaan heidän mielestään ohjelmistoa tulisi seuraavaksi kehittää. Katselmoinnissa eri sidosryhmät näkevät, miten tiimi suoriutui ja miltä lopputulos näyttää. Kukin sidosryhmä on kiinnostunut tiimistä tai lopputuloksesta omasta näkökulmastaan. Katselmoinnissa selviää projektin todellinen tila kyseisen pyrähdysten lopussa. Tämän vuoksi katselmoinnissa ei saa käyttää diaesityksiä eikä siihen tule erityisesti valmistautua, vaan tarkoituksena on, että uusi, toimiva pala ohjelmistoa puhuu puolestaan ja katselmointi ylipäätään on suhteellisen vapaamuotoinen tilaisuus. (Schwaber & Beedle, 2001)

Pyrähdyksen katselmoinnin jälkeen ennen seuraavan pyrähdysten alkua pidetään *pyrähdyksen retrospektiivi* (engl. sprint retrospective). Se on aikarajattu kokous, johon osallistuvat Scrum-mestari ja tiimi. Kokouksessa on tarkoitus tunnistaa asiat, jotka menivät hyvin, ja asiat, joita voidaan parantaa. Nämä asiat voivat liittyä esimerkiksi tiimin jäsenistöön, kokouksiin, työkaluihin, prosessiin

ja kommunikaatioon. Kokouksen aikana parannuskohteet tulisi tunnistaa ja laatia niihin ratkaisu, jonka toimivuutta kokeillaan seuraavassa pyrhdyksessä. (Schwaber & Sutherland, 2008)

3.3.3 Roolit

Scrumissa määritellään erilaisia rooleja ohjelmistokehitykseen tavalla tai toisella osallistuville työntekijöille. Rooleja ovat Scrum-mestari, tuotteen omistaja, Scrum-tiimi, johto ja asiakkaat.

Scrum-mestari (engl. Scrum master) on "jokapaikanhöylä", joka osallistuu jollain tapaa lähes kaikkiin tehtäviin Scrum-organisaatiossa ja on vastuussa Scrumin onnistumisesta. Scrum-mestarin vastuulla on huolehtia, että Scrumin arvoja, periaatteita ja sääntöjä noudatetaan, sekä toimia alkuun panevana voimana käytänteiden suhteen. Hän myös seuraa tiimin jäseniä ja projektin edistymistä. Ongelmatilanteissa Scrum-mestari auttaa tiimiä ongelman ratkaisemisessa. (Schwaber & Beedle, 2001)

Projektin alussa Scrum-mestari valitsee tuotteen omistajan yhdessä johdon ja asiakkaiden kanssa ja määrittelee Scrum-tiimin jäsenet yhdessä johdon kanssa. Scrum-mestari on mukana, kun tuotteen omistaja ja tiimi(t) määrittelevät tuotteen työlistan. Pyrhdyksen alussa Scrum-mestari suunnittelee pyrhdyksen yhdessä tiimin tai tiimien kanssa ja käynnistää pyrhdyksen. Pyrhdyksen aikana Scrum-mestari vastaa päivittäisestä tapaamisesta ja huolehtii esteiden poistamisesta sekä sutjakkaasta päätöksenteosta. (Schwaber & Beedle, 2001)

Johtuen Scrum-mestarin keskeisestä asemasta, rooliin valitaan usein henkilö, jolla on sekä kunnioitusta että valtaa organisaatiossa. Scrum-mestarilta vaaditaan lisäksi sitoutumista ja keskittymistä auttaa Scrum-tiimiä keinoja kaihdamatta. Luonteeltaan Scrum-mestarin ei pidä olla ujo keskeisen asemansa takia, ja esteiden poistaminen vaatii järeäpäisyyttä. (Schwaber & Beedle, 2001)

Tuotteen omistaja (engl. product owner) on yksinään vastuussa tuotteen työlistasta ja sen hallinnoinnista. Tuotteen omistaja vastaa virallisesti projektista. Tuotteen omistajana voi olla vain yksi henkilö. Tämän henkilön kautta tai hänet vakuuttamalla voidaan muuttaa tuotteen työlistaa tai listan kohtien tärkeysjärjestystä. Näin ollen on vain yksi näkemys työlistasta ja sen tärkeysjärjestyksestä, jolloin epäselvyyksiä tai ristiriitaista tietoa ei pitäisi olla. Tuotteen omistajan päätöksen ovat näkyviä, koska ne heijastuvat välittömästi tuotteen työlistaan. Tiimi mukautuu vain muutoksiin, jotka tulevat tuotteen työlistaan. (Schwaber & Beedle, 2001)

Scrum-tiimi valitsee yhdessä Scrum-mestarin kanssa pyrhdyksen tehtävät, joiden toteuttamiseen tiimi omistautuu. Scrum-tiimillä on vapaat kädet toteuttaa tehtävät parhaaksi katsomallaan tavalla organisaation rajoitteiden puitteissa. Scrum-tiimi on poikkitoiminnallinen. Tällä tarkoitetaan sitä, että tiimissä on kaikki osaaminen, joka tarvitaan pyrhdyksen tavoitteen täyttymiseen. Tiimissä kaikki ovat samanarvoisia eikä kenelläkään ole erityisiä nimikkeitä. Tiimin koko voi vaihdella kolmen ja kahdeksan välillä. Kahdeksaa henkeä suuremmat tiimit tulisi jakaa kahdeksi tiimiksi, sillä esimerkiksi päivittäinen Scrum-

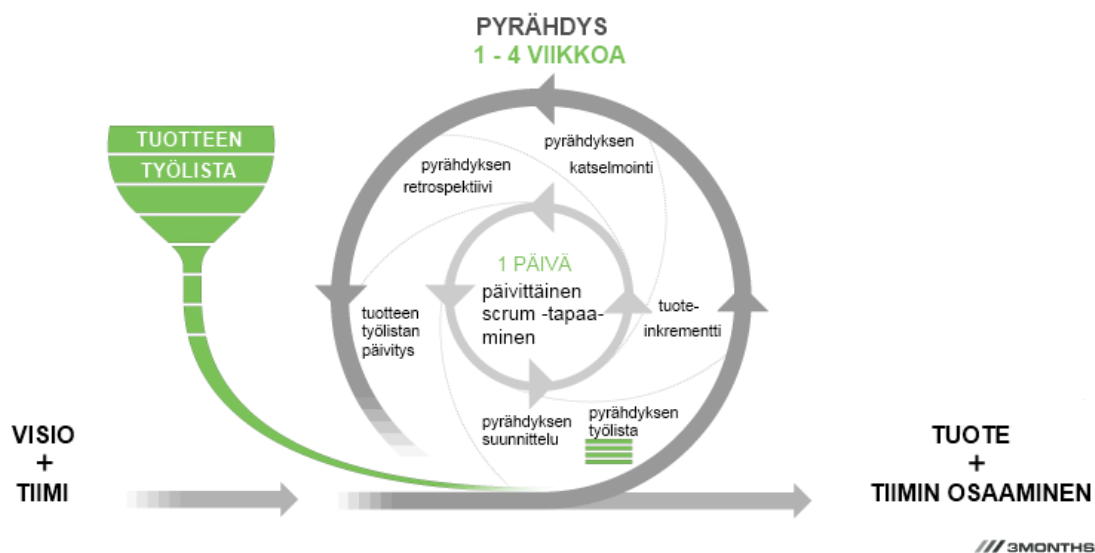
tapaaminen voi muodostua raskaaksi. Tiimin jäsenistö voi vaihtua pyrähdysten välissä tarpeen mukaan. Samoin jotkin jäsenistä voivat toimia tiimissä vain osan aikaa, mikäli heidän osaamisalueensa on hyvin kapea. Nyrkkisääntönä voidaan kuitenkin pitää, että jokainen jäsen osallistuu ohjelmiston tuottamiseen ja tekee parhaansa. (Schwaber & Beedle, 2001)

Johdon rooli Scrumissa on auttaa tiimiä mahdollisuuksien mukaan ja tehdä tärkeitä päätöksiä projektin suunnan suhteen. Johto on mukana pyrähdysten katselmoinnissa arvioimassa tuotosta suhteessa annettuihin resursseihin. (Schwaber & Beedle, 2001)

Asiakkaat tuottavat vaatimuksia tuotteen työlistaan ja pyrkivät tarjoamaan alakohtaista tietoa, mikäli sellaiseen on tarvetta. He myös päättävät ennen jokaista pyrähdystä, mitä pyrähdyksessä halutaan tiimin tuottavan. He ovat mukana pyrähdysten katselmoinnissa arvioimassa, onko pyrähdysten tuotos tyydyttävä. (Schwaber & Beedle, 2001)

3.3.4 Prosessimalli

Scrumin prosessia on havainnollistettu kuviossa 1. Tiimi tai tiimit hyödyntävät Scrumia toteuttaakseen tuotteen, jonka asiakas visioi. Tuote on kuvattu konkreettisina ominaisuuksina tuotteen työlistalla. Tuotteen työlistalta otetaan sopiva määrä ominaisuuksia pyrähdystä varten pyrähdysten suunnittelukokouksen ensimmäisessä osassa ja ne pilkotaan pienemmiksi tehtäviksi suunnittelukokouksen toisessa osassa. Valitut tehtävät muodostavat pyrähdysten työlistan. (Schwaber & Beedle, 2001)



KUVIO 1 Scrum prosessi Cheesmania mukailten (Cheesman, 2010)

Pyrähdysten lopputuloksena syntyy uusi, toimiva inkrementti ohjelmaa. Pyrähdysten päätteeksi tämä palanen esitellään sidosryhmille pyrähdysten katselmoinnissa, jonka yhteydessä ilmenneet kehitysehdotukset päivitetään tuotteen työlistaan. Pyrähdysten päätteeksi tiimi kokoontuu tuotteen omistajan ja

Scrum-mestarin kanssa pyrähdyn retrospektiiviin keskustelemaan toimintatapojen kehittämisestä. Kuten kuvassa 1 on havainnollistettu, Scrumin ytimessä on päivittäinen Scrum -tapaaminen, jossa tiimi motivoituu työhön ja kartoittaa esteet. (Schwaber & Beedle, 2001)

3.4 XP

Kuten Agile-manifestissa, myös XP:ssä on arvot ja periaatteet. Näiden lisäksi XP:ssä on myös käytänteitä, joista varsinainen menetelmä käytännössä muodostuu, sekä rooleja, joilla on tietyt vastuut. Tässä tutkielmassa XP kuvataan lähinnä Beckin ja Andreksen (2004) mukaan.

XP:ssä on viisi arvoa, neljätoista periaatetta sekä kolmetoista pääkäytäntä ja yksitoista muuta käytännettä, joita Beck ja Andres (2004) nimittävät seurauskäytänteiksi (engl. corollary practices) ja joita tässä tutkielmassa kutsutaan jatkossa täydentäviksi käytänteiksi.

3.4.1 Arvot

XP:ssä arvostetaan kommunikaatiota, yksinkertaisuutta, palautetta, rohkeutta ja kunnioitusta. Beck ja Andres (2004) toteavat, että *kommunikaatio* merkitsee eriten tiimipohjaisessa ohjelmistokehityksessä. On ongelmia, jotka johtuvat huonosta kommunikaatiosta. Toisaalta kommunikaation parantuessa siitä tulee yksi apuväline ongelmanratkaisussa. Kommunikaatio on avainasemassa ryhmähengen luonnissa ja tehokkaan yhteistyön saavuttamisessa. Kommunikaatio ei kuitenkaan yksinään riitä vaikuttavaan ohjelmistokehitykseen. (Beck & Andres, 2004)

Yksinkertaisuudella tarkoitetaan pyrkimystä ratkaista ongelmat yksinkertaisimmalla tavalla, joka toimii. Beck ja Andres (2004) huomauttavat, että tällä tarkoitetaan tarpeettoman monimutkaisuuden eliminointia painottamalla yksinkertaisuutta ratkaisuisissa. Heidän mukaansa yksinkertaisuus ja kommunikaatio ovat toisiaan tukevia ja tasapainottavia arvoja. Yhden tavoittelu helpottaa toisen saavuttavista. Yksinkertaistaminen vähentää kommunikoitavan informaation määrää, ja hyvä kommunikaatio puolestaan välittää vain tilanteen kannalta oleellisia asioita, jolloin yksinkertaisuus säilyy. (Beck & Andres, 2004)

"Muutos on väistämätöntä mutta muutos luo tarpeen *palautteelle*. (Beck & Andres, 2004, s. 19)" Palaute ohjaa tiimin toimintaa, ja se voi olla monessa muodossa. XP:ssä ryhmä pyrkii tuottamaan mahdollisimman nopeasti niin paljon palautetta kuin se kykenee käsittelemään. Palaute tulisi saada mahdollisimman nopeasti, jotta voidaan reagoida mahdollisimman nopeasti, mikäli siihen on tarvetta. Palaute on tärkeä osa kommunikaatiota. (Beck & Andres, 2004)

Rohkeudella tarkoitetaan rohkeutta toimia ja puuttua ongelmiin. Sillä tarkoitetaan myös rohkeutta kommunikoida asioista rehellisesti ja avoimesti, rohkeutta hylätä huono ratkaisu yksinkertaisemman löytyessä ja rohkeutta etsiä

oikeita vastauksia, mikä luo palautetta. Silmitön rohkeus ilman muita arvoja on kuitenkin ongelma. (Beck & Andres, 2004)

Kunnioitus on pohjana kaikille muille neljälle arvolle. Tiimin tulee välittää jäsenistään ja projektista, jossa työskentelevät. Jokaisen panos ohjelmistokehityksessä on yhtä arvokas ja sitä tulee kunnioittaa. (Beck & Andres, 2004)

Nämä viisi arvoa ovat XP:n tärkeimmät ajurit. Tiimin käytöksen tulisi heijastaa valittuja arvoja. Arvot eivät kuitenkaan tarjoa mitään vastauksia käytännön tasolla. Arvot ovat yleismaailmallisia ajureita periaatteiden ja käytänteiden taustalla. (Beck & Andres, 2004)

3.4.2 Periaatteet

XP:ssä periaatteet luovat sillan yleisen tason arvojen ja käytännönläheisten käytänteiden välille (Beck & Andres, 2004). Seuraavaksi esitellään lyhyesti kukin XP:n neljästätoista periaatteesta.

Inhimillisyydellä (engl. humanity) tarkoitetaan inhimillisten olosuhteiden tarjoamista ihmisille, jotka osallistuvat ohjelmistokehitykseen. Heidän perustarpeidensa tulee olla täytettyjä, jotta he voivat olla hyviä ohjelmistokehittäjiä. (Beck & Andres, 2004)

Taloudellisuus (engl. economics) on periaate, jolla tarkoitetaan taloudellisten realiteettien ja riskien tunnistamista ohjelmistokehityksessä. Ohjelmistokehitys on arvokkaampaa, kun se tuottaa rahaa aikaisin ja kuluttaa sitä myöhemmin. (Beck & Andres, 2004)

Yhteisen hyödyn periaatteen mukaan kaikkien toiminta tulisi hyödyttää kaikkia osapuolia. Beckin ja Andreksen (2004) mukaan yhteisen hyödyn periaate on XP:n tärkein periaate mutta samalla vaikein toteutettava.

Itsetoisteisuudella (engl. self-similarity) tarkoitetaan hyvien ratkaisujen toistamista eri tilanteissa. Aina sama ratkaisu ei tietenkään toimi, mutta siitä huolimatta se on hyvä lähtökohta. (Beck & Andres, 2004)

Parantamisella tarkoitetaan pyrkimystä tehdä aina parhaansa ja asteittain nostaa oman parhaan tekemisensä tasoa. (Beck & Andres, 2004)

Monimuotoisuudella käsitetään ohjelmistokehitystiimien monimuotoisuutta, sitä että niissä on erilaisia jäseniä, joissa kullakin on omat vahvuutensa ja ajattelutapansa. Tästä seuraa positiivisia konflikteja, kun esimerkiksi ongelmiin keksitään useita kilpailevia ratkaisuja. Se ei kuitenkaan ole ongelma vaan mahdollisuus. (Beck & Andres, 2004)

Reflektoinnilla (engl. reflection) tarkoitetaan oman tekemisen ja sen motiivien analysointia sekä parannusten ehdottamista niin virallisissa kuin epävirallisissa tilaisuuksissa. (Beck & Andres, 2004)

Virralla tarkoitetaan vakaata arvokkaan ohjelmiston virtaa, joka saadaan aikaan ryhtymällä kaikkiin kehitystoimiin samanaikaisesti. XP:n käytänteet painottavat toimintojen jatkuvaa virtaa sen sijaan, että kehityksessä olisi erilaisia vaihteita. (Beck & Andres, 2004)

Mahdollisuuden periaatteella tarkoitetaan tapaa nähdä ongelmat ennemmin mahdollisuutena muutokselle, oppimiselle ja kehittymiselle. (Beck & Andres, 2004)

Redundanssilla tarkoitetaan esimerkiksi ongelman ratkaisemista usealla keinolla, jolloin yhden ratkaisun epäonnistuessa on toinen ratkaisu valmiina. Redundanssi, joka ei palvele tarkoitustaan, on hukkaa. (Beck & Andres, 2004)

Epäonnistuminen on periaatteena muistutus siitä, että epäonnistuminenkin on sallittua. On myös parempi epäonnistua yrittäessään kuin pelätä yrittää. Epäonnistumisesta tulee kuitenkin oppia. (Beck & Andres, 2004)

Laatu-periaatteella tarkoitetaan panostamista laatuun ohjelmistokehityksessä. Se ei Beckin ja Andreksen (2004) mukaan ole puhtaasti taloudellinen kysymys, vaan ihmisten tulee olla ylpeitä työnsä laadusta. Laatu ei myöskään saa olla tekosyy olla keksimättä jonkinlaista ratkaisua johonkin ongelmaan - on vain tehtävä parhaansa. (Beck & Andres, 2004)

Pienet askelet tarkoittavat muutoksen tekemistä vähitellen. Pienin askelin ihmiset pysyvät muutoksessa mukana ja muutoksia voi tehdä enemmän. (Beck & Andres, 2004)

Hyväksytyt vastuu-periaate tarkoittaa, että vastuuta ei voi jakaa. Sen voi vain hyväksyä. (Beck & Andres, 2004)

3.4.3 Käytänteet

XP:n käytänteet jaotellaan ensisijaisiin ja täydentäviin käytänteisiin. Beckin ja Andreksen (2004) mukaan jaotteluun ollaan päädytty, koska täydentävien käytänteiden hyödyntäminen ennen ensisijaisten käytänteiden hyödyntämistä ja hallitsemista voi olla vaikeaa. Ensisijaiset käytänteet sen sijaan ovat hyödyllisiä riippumatta tilanteesta.

Ensisijaisia käytänteitä ovat yhdessä istuminen, kokonaisvaltainen tiimi, informatiivinen työtila, tarmokas työ, pariohjelmointi, tarinat, viikkosykli, vuosineljännessykli, löysä, kymmenen minuutin ohjelmakooste, jatkuva integrointi, testilähtöinen ohjelmointi ja inkrementaalinen suunnittelu. (Beck & Andres, 2004)

Yhdessä istuminen käytänteenä tarkoittaa, että tiimi on työtä varten järjestäytyneenä samaan tilaan. Yksityisyyttä varten voi olla erillisiä tai omia tiloja tiimin jäsenille yhteisen tilan läheisyydessä. Käytänteellä pyritään parantamaan kommunikaatiota. Kommunikaatiota arvostamaan oppineet tiimit järjestyvät itsenäisesti yhteen tilaan, kun heille annetaan siihen mahdollisuus. (Beck & Andres, 2004)

Kokonaisvaltainen tiimi koostuu ihmisistä, joilla on kaikki kyvyt ja näkökulmat, jotka vaaditaan projektin onnistumiseen. Tiimi on siis poikkitoiminnallinen. Kokonaisvaltaisuudella tarkoitetaan myös tiimin jäsenten sitoutumista tiimiin, heidän yhteistä "me"-henkeä ja keskinäistä luottamusta. Tiimin koko on merkittävä tekijä luottamusta rakennettaessa, sillä tiimin kasvaessa se hankaloituu. (Beck & Andres, 2004)

Informatiivisella työtilalla tavoitellaan tilannetta, jossa kuka tahansa pelkää työtilaan saapumalla ja ympärilleen katsomalla voi todeta projektin tilan. Tarkemman tarkastelun pitäisi tarjota lisää tietoa. Tähän tavoitteeseen usein pyritään asettelemalla tarinakortit seinälle tarinan tilan mukaan lajiteltuna. Muita tapoja tarjota tietoa ovat esimerkiksi erilaiset seurantakuvaajat. (Beck & Andres, 2004)

Tarmokas työ -käytännön mukaan tulee tehdä töitä vain niin pitkään kuin kykenee olemaan tuottava ja jaksaa tehdä töitä (Beck & Andres, 2004). Beckin ja Andreksen mukaan väsyneenä työskentelyä pitää välttää, koska silloin tekee helposti huonoja ratkaisuja, jotka vähentävät ohjelmiston arvoa. Myöskään sairaana ei pidä saapua työpaikalle.

Pariohjelmoinnilla tarkoitetaan työskentelytapaa, jossa kaksi henkilöä ohjelmoivat yhdessä saman tietokoneen ääressä. Molempien tulee kyetä olemaan mukavasti, ja tämän takia on syytä kunnioittaa toisen henkilökohtaista tilaa. (Beck & Andres, 2004)

Tarinat ovat lappusia tai tarinakortteja, joissa kuvataan toiminnallisuus, joka kehitettävässä ohjelmistossa tulee olla. Tarinalla on nimi ja lyhyt kuvaus tai havainnollistava piirros. Tarinan toteuttamiseen kuluva aika arvioidaan mahdollisimman pian kirjoittamisen jälkeen ja arvio kirjataan tarinakorttiin. Kortit asetetaan oikealle paikalle seinälle, jossa kaikki muut olemassa olevat tarinat ovat lajiteltuina. (Beck & Andres, 2004)

Viikkosykli on kehityksen tahti. Viikko aloitetaan kokouksella, jossa katselmoidaan edistyminen, asiakas valitsee viikon tarinat ja tarinat pilkotaan tehtäviin, joiden toteuttamiseen kuluva aika arvioidaan. Suunnittelun jälkeen tehdään automaattiset testit viikon tarinoita varten. Tämän jälkeen on helppo lähteä toteuttamaan tarinoita, jolloin testit yksi toisensa jälkeen menevät läpi. Viikko on valittu tahdiksi, koska puolesta välissä voidaan helposti havaita, jos kaikkea ei saadakaan toteutettua. Tällöin keskitytään tärkeimpiin tarinoihin ja niiden toteutukseen. Viikkosykli on myös hyvä aikaväli erilaisille kokeiluille. Viikkosykliä suunniteltaessa näkökulma on tiimi- ja projektitasolla. (Beck & Andres, 2004)

Vuosineljännessykli on toinen kehityksen tahti. Joka vuosineljänneksessä reflektoidaan tiimiä, projektia ja sen suhdetta suuremman mittakaavan tavoitteisiin. Vuosineljänneestä suunniteltaessa tunnistetaan mahdolliset pullonkaulat, aloitetaan korjaustoimet, suunnitellaan teema seuraavalle neljännekselle ja valitaan teeman mukaiset tarinat toteutettavaksi. Vuosineljänneestä suunniteltaessa näkökulma on projekti- ja organisaatiotasolla. (Beck & Andres, 2004)

Löysällä tarkoitetaan suunnittelua, jossa otetaan mukaan toissijaisia tehtäviä, jotka voidaan pudottaa tavoitteista pois, mikäli kaikkea suunniteltua ei ole mahdollista saada toteutettua. Sillä myös tarkoitetaan rehellisyyttä tehtävien toteuttamiseen kuluvan ajan arvioinnissa. (Beck & Andres, 2004)

Kymmenen minuutin ohjelmakoosteella tavoitellaan palautteen vasteajan lyhentämistä. Koko ohjelma koostetaan ja kaikki testit ajetaan automaattisesti kymmenessä minuutissa, koska ajan pidentyessä koostamista vältellään ja palautetta ei saada. Nopea ohjelmakooste vähentää työntekijöiden stressiä, koska

muutosten jälkeen he voivat helposti todeta ohjelman toimivuuden kääntämällä uuden koosteen. (Beck & Andres, 2004)

Jatkuvaassa integroinnissa uusi ohjelmakoodi integroidaan koodikantaan ja ajetaan automaattiset testit vähintään parin tunnin välein. Tällä pyritään minimoimaan ongelmat, jotka seuraavat mittavien muutosten integroinneista. Jos integrointi epäonnistuu, tulee se korjata välittömästi. (Beck & Andres, 2004)

Testilähtöinen ohjelmointi tarkoittaa testien kirjoittamista ennen varsinaisen ohjelmakoodin tuottamista. Testit eivät mene läpi ennen kuin ohjelmakoodi on toteutettu. (Beck & Andres, 2004) Beckin ja Andreksen (2004) mukaan testilähtöinen ohjelmointi lisää luottamusta ohjelmakoodia kohtaan, pitää työssä hyvän rytmin, paljastaa suunnitteluvirheitä, kun testien kirjoittaminen on hankalaa, ja estää turhan ohjelmakoodin kirjoittamista tarjoamalla selkeän päämäärän.

Inkrementaalinen suunnittelu on käytänne, jonka mukaan suunnittelua ja ohjelmiston arkkitehtuurimuutoksia sisältyy päivittäiseen työhön. Suunnittelu ja arkkitehtuurisuunnittelu ei ole tehtävä, joka tehdään ennen kehitystä kerralla oikein, vaan järjestelmää suunnitellaan uudelleen ja toteutusta muutetaan pienin askelin jatkuvasti tiedon lisääntyessä. Ohjelmakoodin uudelleen kirjoittamista ilman toiminnallisuuden muutoksia kutsutaan refaktoroinniksi. Inkrementaalinen suunnittelu helpottaa muutoksiin reagoimista. (Beck & Andres, 2004)

Täydentäviä käytänteitä ovat asiakkaan todellinen osallistuminen, inkrementaalinen käyttöönotto, tiimin jatkuvuus, kutistuvat tiimit, ongelman alkuperän analysointi, yhteinen ohjelmakoodi, yksi ohjelmakoodikanta, ohjelmakoodi ja testit, päivittäinen käyttöönotto, neuvoteltavan mittakaavan sopimus ja käyttöön perustuvat maksut. (Beck & Andres, 2004)

Asiakkaan todellisessa osallistumisessa on kyse heidän konkreettisesta osallistumisesta ohjelmistokehitykseen tai siihen liittyviin päätöksiin. Tavoitteena on, että asiakas on osa ohjelmistokehityksestä vastaavaa tiimiä ja hänellä on valta päättää jossain määrin tiimin tehtävistä suoraan. (Beck & Andres, 2004)

Inkrementaalisen käyttöönoton mukaan ohjelmiston valmistuneet osat tulee ottaa käyttöön sitä mukaa, kun ne valmistuvat. Näin tehdään, koska niin sanottu "big bang" -käyttöönotto onnistuu hyvin harvoin. (Beck & Andres, 2004) Voidaan väittää, että inkrementaalinen käyttöönotto osaltaan tuottaa palautetta nopeammin ja edesauttaa ongelmien nopeaa ratkaisua.

Tiimin jatkuvuudella viitataan toimivan tiimin yhdessä pitämiseen. Yksittäisen jäsenen vaihtaminen tarvittaessa tai vaihtuvuus pitkällä aikavälillä on tietysti mahdollista. Kyse on pohjimmiltaan tiimin jäsenten välisistä suhteista ja luottamuksesta, joita ei pidä vähätellä. Edelliseen liittyy *kutistuvan tiimin* käytänne. Kun tiimin kyvyt kasvavat, tulee sen työmäärä pitää vakiona mutta pienentää tiimin kokoa sitä mukaa, kun henkilöille ei riitä työtä. Idea taustalla on, että tiimi hioo omaa toimintaansa koko ajan tehokkaammaksi, jotta se voi puodottaa "ylimääräisen" jäsenen pois. Jos tiimi kutistuu liian pieneksi, se yhdistetään toiseen liian pieneen tiimiin. (Beck & Andres, 2004)

"Jokainen kerta, kun ohjelmistosta löydetään virhe kehityksen jälkeen, virhe ja sen lähde tulee korjata. Tavoitteena ei ole ainoastaan estää saman vir-

heen toistuminen ohjelmistossa, vaan ettei tiimi koskaan toista samankaltaista virhettä. (Beck & Andres, 2004, s. 64)" Tämä on perusajatus *ongelman alkuperän analysoinnissa*. Ohjelmistovirheen synty ja syy, miksi sitä ei havaittu, tulee selvittää. Tähän tarkoitukseen voidaan soveltaa viiden miksi -kysymyksen sarjaa. Riittävän monen kysymyksen jälkeen perimmäinen - yleensä ihmisiin liittyvä - syy paljastuu. (Beck & Andres, 2004)

Yhteinen ohjelmakoodi tarkoittaa, että kaikki tiimin jäsenet yhdessä omistavat sen ja siksi kuka tahansa heistä voi koska tahansa parantaa mitä tahansa osaa ohjelmistosta. Tästä seuraa, ettei kukaan yksinään ole vastuussa tietystä osasta, jolloin on mahdollista, ettei kukaan kannu vastuuta. Tästä syystä yhteinen ohjelmakoodi on täydentävä käytänne. Kun tiimille on kehittynyt kollektiivinen vastuu ohjelmakoodista, edellä kuvattua ongelmaa ei enää ole. Vastuun kehittämisessä auttaa pariohjelmointi. Edellytyksenä yhteiselle ohjelmakoodille on jatkuva integrointi.

Projektilla on vain *yksi ohjelmakoodikanta*, johon kaikki uusi toiminnallisuus kehitetään. Useamman ohjelmakoodikannan käyttäminen lisää turhan työn määrää. *Ohjelmakoodi ja testit* ovat ainoat pysyvät, ylläpidettävät artefaktit, koska kaikki muut ovat turhia, sillä asiakasta kiinnostaa vain ohjelmiston toiminnallisuus. Muut dokumentit voidaan luoda näiden kahden pohjalta. (Beck & Andres, 2004)

Päivittäinen käyttöönotto on tärkeää, koska erot tuotantokoodin ja ohjelmoijan työstettänä olevan ohjelmakoodin välillä kasvattavat riskiä tehdä päätöksiä, jotka eivät perustu todenmukaiselle palautteelle. Päivittäisellä käyttöönotolla on monta vaatimusta ja estettä ennen kuin on realistista harkita käytänteen hyödyntämistä. Vaatimuksiin kuuluvat muun muassa käyttöönoton automatisointi ja esteisiin ohjelmistovirheiden liiallinen määrä. (Beck & Andres, 2004)

Neuvoteltavan mittakaavan sopimuksella tarkoitetaan sopimuksia, joissa aika, hinta ja laatu on kiinteitä mutta ohjelmiston tarkka mittakaava on neuvoteltavissa. Riskit vähenevät, kun solmii sarjan sopimuksia yhden suuren sopimuksen sijaan. Tähän liittyen on hyödyllistä perustaa hinnoittelu *käyttöön perustuville maksuille*. Tällöin nähdään suoraan, onko toiminnallisuus ollut hintansa väärsti, ja saadaan tuottoa aiemmin. (Beck & Andres, 2004)

3.4.4 Roolit

XP:n kokonaisvaltaisen tiimin käsitteen mukaan tiimissä pitää olla kaikki osaaminen, joka tarvitaan projektin onnistumiseen. Kokonaisvaltainen tiimi koostuu testaajista, interaktiosuunnittelijoista (engl. interaction designer), ohjelmistoarkkitehdeista, projektipäälliköistä, tuotepäälliköistä, johtajista, teknisistä kirjoittajista, käyttäjistä ja ohjelmoijista. Henkilöllä voi olla useita rooleja. Kypsässä XP-tiimissä roolit eivät ole kiinteitä, vaan jokaisen tulee osallistua parhaansa mukaan, mitä tiukasti roolin tehtävissä pitäytyminen saattaa hankaloittaa. (Beck & Andres, 2004) Seuraavaksi esitellään kukin rooli XP:n näkökulmasta.

Testaajat auttavat asiakasta laatimaan ja valitsemaan ohjelmistotason testejä, jotka kirjoitetaan ennen varsinaista toteutusta. Testaajat myös valmentavat ohjelmoijia testaamistekniikoiden suhteen. Testaajien vastuulla on laatia ohjelmistotason testeistä sellaisia, jotka menevät läpi vain, kun toiminnallisuus on kokonaisuudessaan toteutettu. (Beck & Andres, 2004)

Interaktiosuunnittelijat kirjoittavat tarinoita yhdessä asiakkaan kanssa ja arvioivat käyttöön otettua ohjelmistoa löytääkseen uusia tarinoita. He vastaavat myös käyttöliittymästä. (Beck & Andres, 2004)

Ohjelmistoarkkitehdit vastaavat järjestelmän arkkitehtuurista. He etsivät ja toteuttavat tarvittavia suuren mittakaavan refaktorointeja, laativat ohjelmiston arkkitehtuuria rasittavia testejä ja ovat mukana tarinoiden toteuttamisessa. Vastoin perinteistä näkemystä ohjelmistoarkkitehdit työstävät arkkitehtuuria vähän kerrallaan koko kehityskaaren ajan. (Beck & Andres, 2004)

Projektipäälliköt ovat tiimin sisäisen kommunikoinnin helpottajia ja ulospäin suuntautuvan kommunikoinnin koordinoijia. Tämän ohella heidän tehtäviinsä kuuluu projektin etenemisen seuranta ja viestittäminen eri sidosryhmille. Projektipäällikkö on hyvässä asemassa suunnitteluprosessin kehittämisen suhteen. (Beck & Andres, 2004)

Tuotepäälliköt kirjoittavat tarinoita, valitsevat teemat ja tarinat vuosineljännessykliin, valitsevat tarinat viikkosykliin ja vastaavat kehityksessä ilmenneisiin kysymyksiin. Tuotepäällikkö on aktiivinen toimija tilanteissa, joissa tarinointa joudutaan ajan puutteen takia karsimaan tai muokkaamaan. He rohkaisevat ohjelmoijia ja asiakkaita keskustelemaan keskenään, jotta asiakkaan kannalta tärkeimmät asiat saavat huomiota. (Beck & Andres, 2004)

Johtajien tehtävä on valaa uskoa tiimiin ja huolehtia, että tiimi panostaa myös itsensä kehittämiseen. Johtajan tulee odottaa tiimiltä järkeenkäyviä selityksiä ohjelmiston tilasta ja eri vaihtoehdoista kaikissa päätöksentekotilanteissa, jotta hän voi tehdä oikean ratkaisun. (Beck & Andres, 2004)

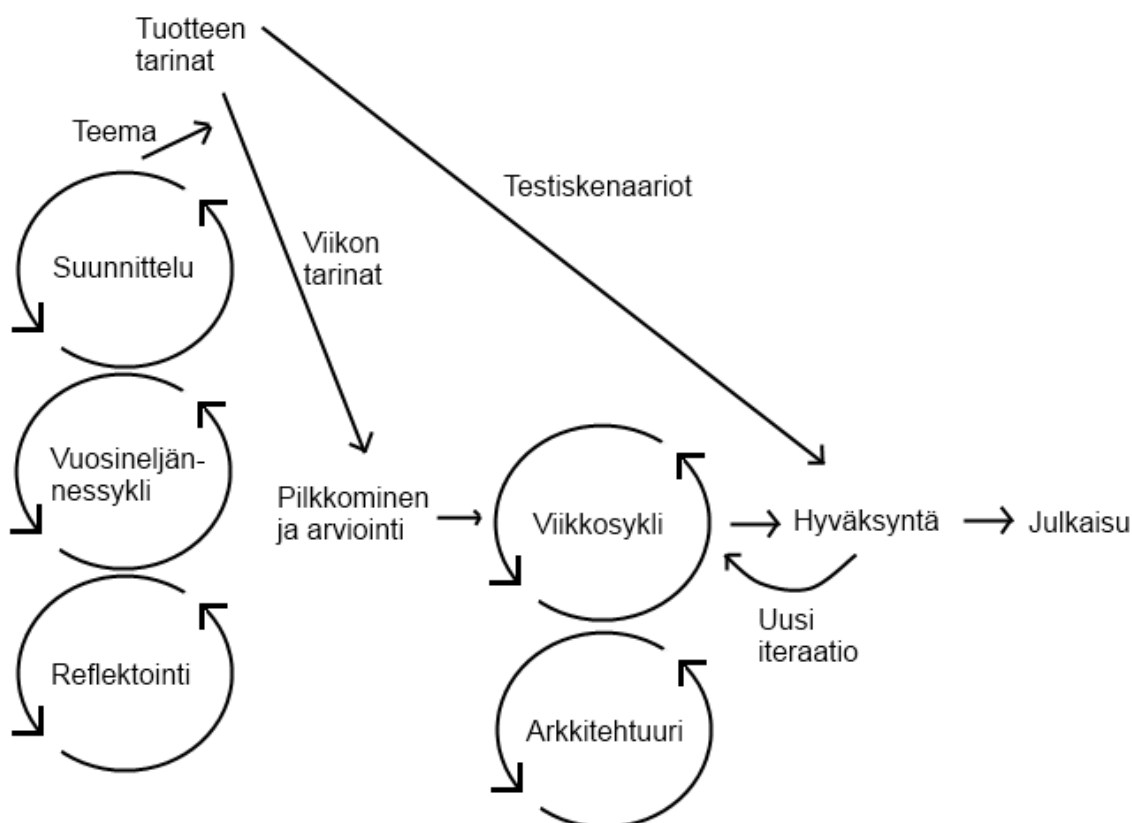
Teknisen kirjoittajan tuotoksilla on kaksi tavoitetta. Niiden tulee tarjota palautetta ohjelmiston toiminnallisuudesta jo varhaisessa vaiheessa ja luoda läheinen suhde käyttäjiin. Ensimmäinen tavoite helpottaa ohjelmoijien työtä, sillä he saavat palautetta työstään nopeasti. Jälkimmäinen vastuu helpottaa käyttäjiä omaksumaan ohjelmisto käyttöönsä. (Beck & Andres, 2004)

Käyttäjän tehtävä on auttaa tarinoiden kirjoittamisessa ja valitsemisessa sekä tehdä omaan toimialaansa liittyviä päätöksiä kehittämisen aikana. Käyttäjän tulisi kuvastaa koko käyttäjäkuntaa, siksi hänen tulee keskustella muiden tulevien käyttäjien kanssa ennen suurten päätösten tekemistä. (Beck & Andres, 2004)

Ohjelmoijat arvioivat tarinoiden ja tehtävien koon, purkavat tarinat tehtäviksi, kirjoittavat testejä ja ohjelmakoodia, automatisoivat työläitä kehitysprosesseja ja asteittain hiovat ohjelmistoa. Ohjelmoijien tulee olla sosiaalisia ja yhteistyökykyisiä. (Beck & Andres, 2004)

3.4.5 Prosessimalli

Beck ja Andres (2004) eivät kuvaa XP:stä prosessimallia. XP voidaan kuitenkin esittää projektinhallintaan liittyvien käytänteiden yhteistoimintaa kuvaavana kaaviona kuvion 2 tapaan.



KUVIO 2 XP-prosessi Beckiä ja Andersta (2004) mukailleen

Vasemmalla kuviossa 2 on neljännesvuosisykli, jolla tarkoitetaan vuosineljänneksen suunnittelemista korkeammalla tasolla sekä organisaation ja tiimin tasolla reflektointia. Ohjelmiston kehitykseen valitaan teemoja, joiden mukaisesti neljännekseen valitaan tarinoita. (Beck & Andres, 2004)

Viikkosykliä suunniteltaessa tarinat pilkotaan tehtäviksi ja kuhunkin tehtävään kuluva aika arvioidaan. Tarinoita vastaavat hyväksymistestit kirjoittaa asiakas apunaan joku tiimin jäsen. Viikkosyklin lopussa ohjelmistoa arvioidaan hyväksymistestejä vasten. Uusia tarinoita voi syntyä missä vaiheessa kehitystä tahansa eikä niiden ehdottamista ole rajoitettu mitenkään. (Beck & Andres, 2004)

XP:ssä arkkitehtuuri muotoutuu ensimmäisten viikkosykliden myötä, kun ohjelmistossa on havaittavissa sen luonnolliset murtumakohdat, joiden mukaan se voidaan pilkkoa arkkitehtonisiksi kokonaisuuksiksi. Arkkitehtuuri muotoutuu viikkosykliden myötä vähitellen ja osa tarinoista voi olla suoraan arkkitehtuurin muutoksiin liittyviä. (Beck & Andres, 2004)

Jokaisen viikon jälkeen tulisi olla uusi, toimiva palanen ohjelmistosta valmiina. Inkrementaalisen käyttöönoton hengessä jokainen valmistunut osa tulisi antaa heti käyttäjien hyödynnettäväksi. Kun edellinen on mahdollista, voidaan koettaa päivittäistä käyttöönottoa, jolloin julkaisujen määrä kasvaa ja koko pienee entisestään. (Beck & Andres, 2004)

3.5 Lean-ohjelmistokehitys

Lean-ohjelmistokehitys pyrkii hyödyntämään muilta aloilta ammennettuja Lean-tuotannon *periaatteita* ohjelmistokehityksessä. Näin ollen sen juuret ovat Lean-ajattelussa ja Toyota Production Systemissä. (Poppendieck & Poppendieck, 2003) Poppendieck ja Poppendieck (2003) ovat esittäneet kirjassaan, miten Lean-periaatteita sovelletaan ketterien käytänteiden räätälöimisessä yrityksessä. Tätä kutsutaan Lean-ohjelmistokehitykseksi. Leanissa on seitsemän periaatetta ja periaatteiden soveltamisen tukena 22 työkalua (engl. thinking tools).

3.5.1 Periaatteet

Lean-ohjelmistokehitykseen liittyvät periaatteet ovat samoja kuin teollisen tuotannon Lean-ajattelussa. Koska ne ovat periaatteita, ne eivät ole käytännön työtä sanelevia ohjeita eli käytänteitä. Periaatteina ne ovat kaikki siirrettävissä muille aloille ja sovellettavissa ohjelmistokehityksessä, kunhan se tehdään asianmukaisesti. (Poppendieck & Poppendieck, 2003) Seuraavaksi kuvataan kukin seitsemästä periaatteesta.

Karsi turha. Turhan karsimisella tarkoitetaan kaiken sellaisen karsimista, mikä ei tuota lisäarvoa asiakkaalle. Tuotannon kontekstissa turhaa ovat muun muassa varastot, tavaroiden siirtely paikasta toiseen ja odottaminen. Autoteollisuudessa Toyota vei tämän ajattelun niin pitkälle, että he näkivät tuotekehittelyn turhana mutta pakollisena, koska ennen varsinaista tuotantoa asiakas ei kokenut saavansa arvoa. Myös ohjelmistokehityksessä on paljon turhaa, mikä voidaan karsia. Turhan karsimiseen liittyy kaksi työkalua – turhan havaitseminen ja arvoketjun kartoittaminen, jotka esitellään seuraavassa aliluvussa ja joiden puitteissa turhaa käsitellään syvällisemmin ohjelmistokehityksen näkökulmasta. (Poppendieck & Poppendieck, 2003)

Vahvista oppimista. Ohjelmistokehitys poikkeaa tuotantoteollisuudesta kolmella tavalla, joita ovat laadun käsite ja suhtautuminen variaatioon sekä iteraatioon. Tuotannossa laadulla tarkoitetaan sitä, että lopputulos on täydellisesti vaatimusten mukainen. Ohjelmistokehityksessä laadulla tarkoitetaan Poppendieckien (2003) mukaan lopputuotteen sopivuutta käyttötarkoitukseensa. Sekä variaatio että iteraatio nähdään tuotannossa ei-toivottuina seikkoina. Ohjelmistokehityksen lopputuotteen ei pidäkään olla aina samanlainen, koska sen on tarkoitus olla asiakkaiden ainutlaatuisiin ongelmiin sopiva toimiva ratkaisu. Ohjelmistokehityksessä lopputuote on aina erilainen. Ohjelmistokehitysprojekti

voidaan nähdä kompleksina ongelmana. Paras tapa ratkaista kompleksi ongelma on käyttää tieteellistä menetelmää, mikä ohjelmistokehityksen yhteydessä tarkoittaa ohjelmiston kehittämistä iteraatioissa. Iteraatiot ja refaktorointi luovat oppimisilmukoita, jotka tuottavat tehokkaasti tietoa, jota voidaan hyödyntää kompleksien ongelmien ratkaisussa. Iteraatioiden yhteydessä tulee kuunnella asiakasta, joka parhaiten tuntee oman liiketoimintaympäristönsä, ja pyrkiä esittämään ohjelmisto sellaisessa muodossa, joka on asiakkaalle helppotajuinen. Näin luodaan tietoa siitä, mihin suuntaan asiakas haluaa ohjelmiston kehittyvän. Oppimisen vahvistamiseen liittyy neljä työkalua – palaute, iteraatiot, synkronointi ja rajoiteperusteinen kehittäminen. (Poppendieck & Poppendieck, 2003)

Päätä mahdollisimman myöhään. Lean-ohjelmistokehityksessä toteutusratkaisut tehdään mahdollisimman myöhään. Tämä poikkeaa vanhasta, vaiheistetusta kehitystavasta, jossa kaikki suunniteltiin tarkasti ennen varsinaista toteutusta. Vaiheistetun kehitystavan takia muutosten tekeminen on kallista ja niitä on vaikea tehdä. Iteratiivinen kehitystapa tuottaa ohjelmakoodia, jota on helppompi muuttaa ja sen myötä luontaisesti pidättäytytään tekemästä päätöksiä, joita ei vielä tarvitse tehdä. Toimiakseen iteratiivinen kehitystapa vaatii läheistä yhteistyötä asiakkaiden kanssa ja ammattitaitoa kehittäjältä kyseisen liiketoiminta-alueen tuntemuksen muodossa. Iteratiivisen kehityksen aikana kohdat, joihin tulee muutoksia paljastavat ne kohdat, joihin myös ylläpitovaiheessa todennäköisesti tulee muutoksia. Avainseikka myöhään päättämisen mahdollistamiseksi on tuottaa ohjelmisto, jota on vaivatonta – ja halpaa – muokata jälkikäteen. Päätösten tekemiseen myöhään liittyy kolme työkalua – vaihtoehtojen puntarointi, viimeinen vastuullinen hetki ja päätösten tekeminen. (Poppendieck & Poppendieck, 2003)

Toimita mahdollisimman nopeasti. Asiakas pitää nopeasta toimituksesta, koska usein nopeasti toimitettu ohjelmisto näkyy parantuneena joustavuutena liiketoiminnassa. Nopea toimitus tuotantoon paljastaa ongelmat ja virheet ohjelmistossa, jolloin niihin voidaan myös reagoida nopeasti. Toimittamatta jättäminen on riski, koska silloin virheet saattavat jäädä paljastumatta. Mahdollisimman nopeasti toimittaminen tarkoittaa myös mahdollisimman myöhään päättämistä, sillä mikäli yrityksellä kestää viikko toteuttaa ja toimittaa muutos, sen pitää lukita toteutusratkaisu vain viikkoa ennen muutoksen toteuttamista. Mahdollisimman nopeasti toimittamiseen liittyy kolme työkalua – vetojärjestelmät, jonoteoria ja viivästyksen hinta. (Poppendieck & Poppendieck, 2003)

Valtuuta tiimi. Tiimin valtuuttamisella tarkoitetaan, että tiimi saa päättää omista toimintatavoistaan ja toteutusratkaisuihinsa. Tämä on lähes täysi vastakohta perinteiselle tieteelliselle johtamiselle (engl. scientific management), jossa esimiehet sanelevat, miten tulee toimia. Vaikka tieteellistä johtamista sellaisenaan ei ole tietyvästi juuri havaittu ohjelmistokehityksessä, on siitä johdettuja standardeja tai kypsyyssmalleja, kuten CMM ja CMMI, hyödynnetty ohjelmistokehityksessä. Lean-ohjelmistokehityksessä esitetään oma kaksikohtainen kypsyyssmalli vastineena CMMI:lle. (Poppendieck & Poppendieck, 2003)

1. Kypsä organisaatio katsoo järjestelmää kokonaisuutena, eikä pyri optimoimaan eriytettyjä palasia järjestelmästä. (Poppendieck & Poppendieck, 2003)
2. Kypsä organisaatio keskittyy tehokkaaseen (engl. effective) oppimiseen ja valtuuttaa työntekijät päätöksen tekoon. (Poppendieck & Poppendieck, 2003)

Organisaatio, joka kunnioittaa kehittäjien ammattitaitoa, odottaa heidän päättävän omasta tekemisestään riittävästi tuettuna ja tekevän parannuksia työtapoihinsa osana oppimista, mikä edelleen auttaa heitä tekemään työnsä hyvin. Tiimin valtuuttamiseen liittyy neljä työkalua - itsemäärääminen, motivaatio, johtajuus ja asiantuntemus. (Poppendieck & Poppendieck, 2003)

Tuota laatua. Laatu (engl. integrity) jaetaan sisäiseen (engl. conceptual integrity) ja ulkoiseen laatuun (engl. perceived integrity). Ulkoisella laadulla tarkoitetaan tuotteen toiminnallisuuksien, käytettävyyden, luotettavuuden ja muiden havaittavien ominaisuuksien välistä tasapainoa, joka ilahduttaa asiakasta. Sisäisellä laadulla tarkoitetaan tuotteen rakennuspalikoiden (engl. concepts) - ohjelmistokomponentit ja arkkitehtuuri - sujuvaa toimintaa yhtenäisenä kokonaisuutena. Sisäinen laatu on edellytys ulkoiselle laadulle mutta ei ole sen tae. Avainasemassa sisäisen laadun tuottamiseen on erinomainen tiedonkulku kehitystiimin sisällä kehittäjien välillä. Ulkoisen laadun tuottamiseksi vaaditaan erinomainen tiedonkulku käyttäjien ja muiden sidosryhmien sekä kehitystiimin välillä. Laadun tuottamiseen liittyy neljä työkalua - ulkoinen laatu, sisäinen laatu, refaktorointi ja testaaminen. (Poppendieck & Poppendieck, 2003)

Näe kokonaisuus. Järjestelmä koostuu toisistaan riippuvista, vuorovaikutteisista osista, jotka on yhdistetty jotakin tarkoitusta varten. Järjestelmä ei ole vain osiensa summa, vaan osiensa yhteistoiminnan tuotos. Järjestelmäajattelua (engl. systems thinking) voidaan soveltaa myös organisaatioihin. Usein organisaatiot jaetaan pienempiin yksiköihin, joiden toimintaa tarkastellaan, mitataan ja pyritään optimoimaan. Tämä ei kuitenkaan tuota parasta tulosta, sillä paikallisilla päätöksillä voi olla kokonaisuuden kannalta hyvin haitallisia vaikutuksia. Järjestelmäajattelun mukaan prosessia rajoittavat tekijät tulee löytää ja poistaa. Tällöin tulee kuitenkin pitää mielessä, että aina syntyy uusia rajoitteita. Tämän takia rajoitteiden eliminointi on jatkuva prosessi. Järjestelmäajattelu pyrkii myös ratkaisemaan ongelmien syyt niiden aiheuttamien oireiden hoitamisen sijaan. Tähän voidaan pyrkiä viiden "miksi?" kysymyksen avulla. Tärkein järjestelmäajattelun huomio liittyy kuitenkin osioptimointiin (engl. suboptimization), jota esiintyy, kun optimoidaan vain järjestelmän osia yksinään. Kokonaisuuden näkemiseen liittyy kaksi työkalua - mittaaminen ja sopimukset. (Poppendieck & Poppendieck, 2003)

3.5.2 Työkalut

Tässä osiossa käydään lyhyesti lävitse ne Poppendieckien (2003) esittämistä työkaluista, jotka sekä katsotaan olennaisiksi pelikehitystä ajatellen että ovat

tulleet mainituiksi ketterää pelikehitystä käsittelevissä lähteissä. Nämä ovat ne käytänteet, jotka tulee taustoittaa myöhempien lukujen ymmärtämiseksi.

Turhan havaitseminen on ensimmäinen turhan karsimiseen tarkoitettu työkalu. Turhan havaitsemisella tarkoitetaan kykyä tunnistaa, mikä on turhaa, ennen kuin aletaan kitkeä turhaa. Ohjelmistokehityksen alalla on seitsemää erilaista turhaa. *Osittain tehdyllä työllä* on tapana osoittautua vanhentuneeksi, siinä voi olla ongelmia, jotka paljastuvat vasta loppukäytössä, eikä voida osoittaa sen ratkaisevan liiketoimintaongelmaa. *Ylimääräiset prosessit* estävät tarkoituksellisen työn tekemistä. Esimerkiksi paperityö voi olla täydellistä ajanhukkaa, mikäli kukaan ei oikeastaan tarvitse tuotettuja dokumentteja mihinkään mutta prosessi niitä vaatii. *Ylimääräisten ominaisuuksien* ohjelmointi ei vaikuta vaaralliselta mutta kaikki ylimääräinen ohjelmakoodi lisää virheiden mahdollisuutta ja kasvattaa ohjelmiston kompleksisuutta. *Tehtävien vaihtelu* eri tehtävien välillä hidastaa kaikkien tehtävien tekemistä. *Odottamista* voi esiintyä monissa tilanteissa ja sen aiheuttama viivästys voi pahimmillaan tehdä projektista kokonaan vanhentuneet tai myöhästyttää koko kehitysaikaa. *Liikkuminen* on turhaa, liittyi se sitten ihmisten liikkumiseen tai töiden liikkumiseen ihmisten välillä. Ensimmäisessä tapauksessa liike aiheuttaa keskittymisen herpaantumisen ja hidastuttaa tiedon kulkua. Jälkimmäisessä tapauksessa töiden liikuttelussa osa hiljaisesta tiedosta saattaa jäädä siirtymättä, jos ihmiset eivät voi keskustella välillään siirtävästä työstä. *Virheiden* aiheuttama haitta on virheen vaikutusten ja virheen havaitsemiseen kuluvan ajan yhteistuotos. Mitä myöhemmin virhe havaitaan, sitä enemmän turhaa työtä se aiheuttaa. Näiden varsinaiseen työhön liittyvien tuhan lajien lisäksi myös mittavat työn edistymisen mittaamiseen tarkoitettut järjestelmät saattavat olla turhia ja aiheuttaa haittaa. (Poppendieck & Poppendieck, 2003)

Arvoketjun kartoittaminen on toinen turhan karsimiseen liittyvä työkalu. Arvoketjua kartoitettaessa selvitetään, missä vaiheessa prosessia tehdään työtä, joka tuottaa asiakkaalle lisäarvoa. Tämä voidaan tehdä esimerkiksi seuraamalla asiakkaan esittämän yksittäisen vaatimuksen kulkua prosessin lävitse valmiiksi ominaisuudeksi tuotteessa. Arvoketju havainnollistetaan piirroksena, jossa piirretään viivaa tason yläpuolelle, kun tehdään työtä, ja tason alapuolelle, kun ei tehdä työtä. Työhön tai odottamiseen kuluva aika merkitään viivalle sopivana ajanyksikkönä. Havainnollistetun arvoketjun perusteella prosessia on helppo parantaa; vähennetään turhaa odottelua ja pyritään tekemään arvoa lisäävää työtä. (Poppendieck & Poppendieck, 2003)

Oppimisen vahvistamiseen liittyvistä työkaluista palaute ja iteraatiot käsitellään seuraavaksi tarkemmin. *Palaute ja palautteen hyödyntäminen* on ainoa keino ohjata hankalassa tilanteessa ohjelmistoa oikeaan suuntaan, tämä johtuu valtavasta määrästä ohjelmistokehitykseen liittyviä muuttujia, joita on mahdoton ennakoita. Ohjelmistokehityksessä tulee olla riittävästi palautesilmukoita, joiden kautta saadaan oikeaa tietoa ratkaisujen toimivuudesta. Palautesilmukoita ovat niin testilähtöinen ohjelmistokehitys, testien ajaminen kuin läheinen yhteistyö asiakkaan kanssa. (Poppendieck & Poppendieck, 2003)

Iteraatio on aikarajattu jakso, jonka aikana toteutetaan pieni, toimiva palanen lopullista tuotetta. Iteraatioiden puitteissa voidaan antaa ja hyödyntää enemmän palautetta kuin ilman iteraatioita. Iteraatioilla saavutetaan myös kolme muuta etua. Pienen ohjelmiston palasen toteuttaminen on nopeampaa kuin suuren. Iteraatio tarjoaa rajoiteperusteisen lähestymistavan, jolloin voidaan kehittää ohjelmistoa faktojen eikä ennusteiden perusteella. Iteraatiot synkronoivat kehittämisen tiimien ja asiakkaan välillä; iteraation lopussa tuote on aina lähes julkaistavassa kunnossa. Tämä synkronointi mahdollistaa tiimien itsenäisen työnteon mutta estää niitä ajautumasta liian kauaksi yhtenäisestä visiosta, jonka niin tiimit, asiakkaat kuin käyttäjätkin jakavat. (Poppendieck & Poppendieck, 2003)

Nopeaan toimitukseen liittyviä työkaluja ovat veto-järjestelmät, jonoteoria ja viivästyksen hinta. Näistä käsitellään ainoastaan veto-järjestelmät. *Veto-järjestelmissä* asiakkaan tarpeet ovat signaali tehdä työtä ja työntekijöillä on vapaat kädet tehdä työnsä parhaaksi katsomallaan tavalla. Näihin kahteen seikkaan on päädytty, koska nopeasti muuttuvissa tilanteissa ei ole aikaa odotella tiedon kulkeutumista johtoportaan saati edelleen odotella johtoportaan päätöksen tekoa ja määräysten kulkua takaisin työntekijöille. Kanban on veto-järjestelmä, joka käsitellään seuraavassa aliluvussa. (Poppendieck & Poppendieck, 2003)

Tiimin valtuuttamiseen liittyvistä työkaluista itsemääräämistä, motivaatioita ja johtajuutta ei käsitellä tässä tutkielmassa. Tiimin valtuuttamiseen liittyvistä työkaluista asiantuntemus mainitaan pelikehitystä käsittelevässä kirjallisuudessa (Keith, 2010). *Asiantuntemusta* tulee vaalia ja kehittää. Tähän tarkoitukseen voidaan muun organisaatorakenteen ohelle perustaa erityisiä ammatti- tai pikemminkin asiantuntemuskuntia, joiden tehtävä on jakaa ja vaalia tietoa sekä parantaa asiantuntemuskunnan osaamista. Nämä asiantuntemuskunnat toimivat tiimien taustalla eikä niillä ole varsinaisia liiketoiminnallisia tehtäviä. (Poppendieck & Poppendieck, 2003)

Laadun tuottamiseen liittyvistä neljästä työkalusta – ulkoinen laatu, sisäinen laatu, refaktorointi ja testaaminen – vain refaktorointi käsitellään. Leanissa *refaktorointi* tarkoittaa samaa asiaa kuin XP:ssä. Refaktoroinnissa on kyse ohjelmakoodin parantamisesta muuttamatta varsinasta toiminnallisuutta. Refaktorointia tarvitaan iteratiivisessä kehittämisessä pitämään arkkitehtuuri terveenä ja ylläpitämään erilaisia standardeja liittyen ohjelmakoodin laatuun ja ulkoasuun. Refaktorointi on myös tapa välttää turhaa ylimääräistä työtä kehityksen loppuvaiheessa. Refaktorointia tulisi aina tukea automaattisin testeillä. (Poppendieck & Poppendieck, 2003)

Kokonaisuuden näkemiseen liittyvistä työkaluista – mittaaminen ja sopimukset – käsitellään vain *sopimukset*. Erilaisista sopimustyypeistä aika- ja materiaalikulu-, monivaihe-, tavoitekulu- ja jaetun edun sopimukset sopivat ketterään ohjelmistokehitykseen. Tämän tutkielman puitteissa ei ole olennaista kuvata näitä sopimuksia tarkemmin. Sopimuksille yhteistä on mahdollisuus olla lukitsematta ohjelmiston kaikkia yksityiskotia. Näin voidaan kehityksen edetessä toteuttaa vain tärkeimmät ominaisuuksista ja jättää tarpeettomat ko-

konaan pois. Tämä on eduksi myös asiakkaalle, joka ei joudu maksamaan turhasta. (Poppendieck & Poppendieck, 2003)

3.6 Kanban

Kanban on eräs ketterän kehittämisen työkalupakkiin kuuluva menetelmä. Sen juuret ovat tietävästi Toyota Production Systemissä, joka on kortteihin perustuva vetojärjestelmä auton rakennuksessa käytettäville osille, ja on sittemmin levinnyt myös ohjelmistokehitykseen. Kanban on hyvin yksinkertainen, ja se määrittelee vain kolme käytännettä. Arvoja ja periaatteita Kanbanissa ei ole lainkaan, eikä se myöskään määrittele rooleja tai kuvaa mitään määrättyä prosessia. Kanbanin määrittelemät käytänteet ovat työn edistymisen visualisointi, meneillään olevan työn määrän rajoittaminen (engl. Limit work in progress, WIP) ja yksittäisen tehtävän läpikulkuajan (engl. cycle time tai lead time) keskiarvon seuraaminen (Kniberg & Skarin, 2009).

Usein esimerkiksi Scrumia hyödyntävässä organisaatiossa pyrhdyksen työlista esitetään Kanban-työtilan seinällä. Taulu on jaettu sarakkeisiin, jotka osoittavat kunkin tehtävän sen hetkisen tilan. Aloittamattomat tehtävät ovat omassa sarakkeessaan, samoin työn alla olevat, ja riippuen organisaation työtavoista, valmiit tehtävät voidaan siirtää katselmointia odottavien sarakkeeseen tai muuhun vastaavaan tai valmiiden sarakkeeseen. Tällöin kyseessä on *työn edistymisen visualisointi*. (Kniberg & Skarin, 2009)

Edelleen kullekin sarakkeelle voidaan asettaa arvo, joka kertoo, kuinka monta tehtäväkorttia sarakkeessa voi enimmillään olla samanaikaisesti eli *meneillään olevan työn määrää rajoitetaan*. Tätä voidaan verrata Scrumiin, jossa rajoitetaan pyrhdyksen otettavien tehtävien määrää. Kanbanissa rajoitetaan jokaisen tunnistetun työvaiheen tehtävien määrää, jotta prosessiin ei pääsisi syntymään pullonkauloja. Sopiva tehtävien enimmäismäärä saraketta kohti selviää vain käytännössä kokeilemalla. (Kniberg & Skarin, 2009)

Kanbanissa mitataan *läpikulkuaikaa*, joka yksittäiseltä tehtävältä kuluu kulkea prosessin jokaisen vaiheen läpi valmiiksi asti. Tehtävän läpikulkuajan mittaaminen aloitetaan, kun tehtävä tulee Kanban-työtilalle vasemman puolimmaisesta sarakkeeseen, ja päätetään, kun tehtävä on Kanban-työtilan oikean puolimmaisessa sarakkeessa, eli valmis. Läpikulkuaika on muiden Kanban-käytänteiden ohella työkalu, jota käytetään organisaation prosessin optimointiin. (Kniberg & Skarin, 2009)

Kanbania voidaan hyödyntää Scrumin yhteydessä (Kniberg & Skarin, 2009). Erästä tällaista Scrumin ja Kanbanin hybridiä kutsutaan Scrumbaniksi. Siinä on Kanbanin kaikki käytänteet sekä tietyt hyödyllisiksi havaitut Scrumin käytänteet. Scrumista on jäljellä pyrhdykset ja niiden suunnittelukokoukset, joissa ainoastaan lisätään Kanban-työtilan tyolistasarakkeeseen uusia tehtäviä, sekä katselmoinnit ja retrospektiivit. Päivittäinen Scrum -tapaaminen on räätälöity Scrumbania varten erilaiseksi, koska Scrumin tiimejä ei enää ole jäljellä. (Ladas, 2009)

3.7 Yhteenveto

Tässä luvussa esiteltiin Agile-manifesti, ketterän menetelmän määritelmä ja neljä ketterää menetelmää, jotka ovat Scrum, XP, Lean ja Kanban. Agile-manifestin neljä arvoa toimivat lähtökohtana kaikille ketterille menetelmille. Abrahamssonin ym. (2002) määritelmän mukaan ketterät menetelmät ovat inkrementaalisia, yhteistyötä painottavia, helppoja oppia ja mahdollistavat muutoksiin mukautumisen.

Scrum ja XP esiteltiin arvojen, käytänteiden, roolien ja prosessimallin mukaan. Scrum keskittyy enemmän projektinhallintaan ja XP puolestaan tarjoaa enemmän näkemystä varsinaiseen kehitystyöhön. Lean-ohjelmistokehitys esiteltiin periaatteiden ja työkalujen kautta. Lean-ohjelmistokehitys on ketterän organisaation työkalupakki, joka ei varsinaisesti esitä tiettyä prosessimallia vaan joka ainoastaan tarjoaa monipuoliset lähtökohdat olemassa olevan prosessin parantamiseen. Kanban on erittäin kevyt vetojärjestelmä, joka tarjoaa jatkuvasti ajantasaista tietoa työn edistymisestä ja asettaa työnkululle yksinkertaiset säännöt.

Seuraavassa luvussa käydään läpi ketterää pelikehitystä. Aluksi paneudutaan pelikehityksessä esiintyviin ongelmiin, ja sitten esitellään Agile-manifesti sovellettuna pelikehitykseen. Tämän jälkeen käydään läpi ketterän pelikehityksen vaiheet ja tutustutaan ketterien menetelmien hyödyntämiseen eri vaiheissa, minkä jälkeen käydään läpi käytänteiden tasolla niiden hyödyntämistä. Lopuksi esitellään ketterässä pelikehityksessä esiintyviä rooleja ja ketterän pelikehityksen vaikutuksia pelialan ongelmiin.

4 KETTERÄ PELIKEHITYS

Edellisissä luvuissa on tarkasteltu pelejä, pelaamista ja peliteollisuutta sekä ketterää lähestymistapaa ja ketteristä menetelmistä Scrumia, XP:tä, Leania ja Kanbania. Tämän luvun tarkoituksena on selvittää, minkälaisia esityksiä ketterän lähestymistavan soveltamisesta pelikehitykseen on tehty ja millaisia kokemuksia ketterästä pelikehityksestä on saatu. Lähtökohdaksi otetaan pelikehityksessä koetut ongelmat. Tämän jälkeen esitellään ketterän pelikehityksen arvoja, vaiheita, menetelmiä, käytänteitä ja organisointia. Tämän osuuden pääasiallisina lähteinä käytetään Keithin (2010) kirjaa ja Musilin ym (2010) artikkelia. Clinton Keith on eräs pelialan kuuluisimmista agile-konsulteista. Hän oli mukana, kun High Moon Studios (tuolloin Sammy Studios) tiittävästi ensimmäisenä pelistudiona ryhtyi hyödyntämään ketteriä menetelmiä Mike Cohnin avustamana vuonna 2003. Keith pitää internetissä ketterää pelikehitystä käsittelevää blogia (<http://blog.agilegamedevelopment.com/>). Luvun lopuksi pohditaan kirjallisuuteen nojautuen, miten ketterä lähestymistapa voisi auttaa pelikehitystä sen ongelmissa.

4.1 Pelikehityksen ongelmat

Pelikehitykseen liittyy paljon haasteita ja erilaisia ongelmia, joita on myös teettellisesti tutkittu. Petrillo, Pimenta, Trindade ja Dietrich (2009) vertailivat ohjelmistotuotannon ja pelikehityksen ongelmia. Heidän tutkimuksessaan havaittiin, että pelikehityksessä on samat ongelmat kuin ohjelmistokehityksessä mutta siihen liittyy erityisiä ongelmia.

Ohjelmistotuotannossa keskeisiä ongelmia ovat aikataulut, budjetointi ja laatu sekä hallinnolliset ongelmat. Aikataulutongelma johtuu huonoista arviointitekniikoista, ylioptimismista ja pelosta; ei uskalleta antaa rehellisiä arvioita, kun asiakas tahtoo ohjelmiston olevan valmis tiettyyn päivämäärään mennessä. Budjetoinnin ongelmat ovat heijastumia aikataulutuksesta ja heikosta arviointitekniikoista. Laatu on heikkoa, kun lopputuotos ei vastaa odotettua

lopputulosta. Tekniseltä kannalta ongelma on laadunvarmistuksessa. Virheiden etsiminen ja korjaus on myös eräitä kalleimmista osa-alueista ohjelmistotuotannossa. Hallinnolliset ongelmat kumpuavat huonosta kommunikoinnista, tiimin kouluttamiseen panostamattomuudesta ja riittämättömästä projektin seurannasta. (Petrillo ym., 2009)

Petrillon ym. (2009) mukaan pelikehityksen erityisiä ongelmia ovat ongelmat projektin mittakaavan (engl. scope) määrittelyssä, epärealistinen mittakaava, ominaisuuksien kasvu (engl. feature creep), ominaisuuksien karsiminen kehityksen aikana, aikataulutusetongelmat, loppurutistus (eng. crunch tai crunch time), tekniset ongelmat, ongelmat suunnitteluvaiheessa, viivästykset, dokumentaation puute, kommunikaatio-ongelmat, työkaluongelmat, testausongelmat, ohjelmistovirheiden määrä, ongelmat ryhmäkoostumuksessa, budjetin ylitys ja kehittäjien menettäminen. Useat edellä mainituista ongelmista saattavat johtua ongelmista pelikonseptin valmistelussa ja esituotantovaiheessa sekä pelialan ristiriitaisesta kulttuurista.

Ongelmat projektin mittakaavan määrittelyssä johtuvat usein siitä, ettei pelin ominaisuuksia lukita riittävän aikaisin tai missään vaiheessa kehitystä. Jos uusia ominaisuuksia lisätään suunnittelematta kesken kehityksen, *puhutaan ominaisuuksien kasvusta* (engl. feature creep). On kuitenkin huomautettava, että toisinaan juuri ominaisuuksien kasvun myötä peli kehkeytyy kaupalliseksi menestykseksi huolimatta ongelmista, joita sen myötä kohdataan kehitysvaiheessa. Toisaalta voidaan jo projektin alkuvaiheessa olla liian kunnianhimoisia ja hahmotella ominaisuuksiltaan liian haastava peli suhteessa kehitystiimin kykyihin, jolloin pelillä on *epärealistinen mittakaava*. Tällöin seurauksena on *usein ominaisuuksien karsiminen*, joka voi kohdistua joko toteutettuihin tai vasta suunniteltuihin ominaisuuksiin. (Petrillo ym., 2009) Joskus ominaisuuksia joudutaan karsimaan myös jakeluformaatin rajoitteiden vuoksi (Petrillo ym., 2009), mutta luultavasti internetjakelun ja DVD:n myötä tätä ongelmaa ei juuri ole.

Aikataulutusetongelmat johtuvat osin tiimien monialaisuudesta, sillä pelikehitys perustuu yhteistyölle vankemmin kuin mikään muu viihdetuotanto (Petrillo ym., 2009; Potanin, 2010). Eri alojen edustajien työpanosten keskinäiset riippuvuussuhteet hankaloittavat aikataulutusta ja aikataulussa pysymistä, minkä lisäksi aikataulujen suunnittelu on vaikeaa uutta teknologiaa kehitettäessä etenkin tavoitteiden vaihdellessa (Petrillo ym., 2009; Potanin, 2010).

Loppurutistus on terminä peräisin koripallosta, jossa ennen peliajan loppua taistellaan kiivaasti ottelun voitosta. Pelialalla on vallalla kulttuuri, jossa hyväksytään valtaisa määrä ylityötunteja hyvin pitkälläkin aikavälillä, esimerkiksi joulumarkkinoiden alla. Ilmiötä kutsutaan alalla rutistukseksi tai rutistusajaksi. Rutistusta ruokkii osaltaan alalle ominainen toiminnallisuuden kasvaminen (engl. feature creep) ja tiukat, joustamattomat aikataulut (Musil, Schweda, Winkler & Biffel, 2010a; Potanin, 2010). Pelikehitysraporteissa on ilmoitettu neljän kuukauden ja jopa kahdeksan kuukauden rutistusjaksoja, jolloin kehittäjät ovat tehneet töitä lähes kellon ympäri. Pitkä rutistus heikentää ajan mittaan työkykyä ja edelleen hankaloittaa järkevien päätösten tekoa, mikäli ylitöiden

määrä kasvaa niin suureksi, että kehittäjät eivät saa edes riittävää määrää lepoa. (Petrillo ym., 2009)

Tekniset ongelmat, joita raportoitiin postmortemeissa, johtuvat pääosin kolmannen osapuolen komponenttien ohjelmointirajapintojen heikkouksista. Tämän lisäksi pelistudiot työskentelevät aina viimeisimpien teknologisten innovaatioiden parissa, jolloin kehittäminen on vaativampaa ja vie enemmän aikaa. (Petrillo ym., 2009)

Suunnitteluvaiheen ongelmat johtuvan joko liian vähäisestä tai liiallisesta etukäteissuunnittelusta (Petrillo ym., 2009). Pelistudio koostuu muun muassa ohjelmoijista, tuottajista ja taiteilijoista, jotka kaikki osallistuvat pelisuunnitteluun tai kokevat sen oikeudekseen (Musil, Schweda, Winkler & Biffel, 2010; Potanin, 2010). Tämä yhdistettynä Petrillon ym. (2010) mainitsemaan ”lennosta suunnitteluun” (engl. design on the fly), jossa tehdään pelisuunnittelua teknologian ehdoilla ja suunnitelmat vanhenevat aina niiden laatimisen jälkeen, aiheuttaa varmasti ongelmia.

Petrillo ym. (2009) raportoivat *viivästyksistä*, joita eri pelistudiot ovat kokeneet. Viivästykset ovat myöhästymisiä alkuperäisestä aikataulusta, joka ainakin yhdessä tapauksessa oli julkaisijan sanelema. Aikatauluja myös luonnehdittiin epärealistisiksi eikä niitä ensimmäisten ongelmien ilmaannuttua suinkaan muutettu, mistä seuraa viivästysten paheneminen.

Dokumenttaation puute tarkoittaa, ettei dokumentaatiota ole ollut riittävästi, jolloin ainoastaan jonkin ominaisuuden toteuttaja on kyennyt muokkaamaan tai laajentamaan ominaisuutta. Vaikka pelisuunnitteludokumentti laaditaan, tulee pitää mielessä, että sitä voidaan aina muuttaa. On kuitenkin kiistatta selvää, että jonkinlainen yhteinen ohjenuora pelikehitykseen olisi oltava. (Petrillo ym., 2009)

Kommunikaatio-ongelmat liittyvät toisaalta pelistudion sisäisiin tekijöihin ja toisaalta pelistudion ja sidosryhmien välisiin tekijöihin. Pelistudion sisällä on nähtävissä kaksi rajapintaa, jotka tuottavat ongelmia. Johdon ja kehittäjien välillä on kiista, joka kumpuaa väärinymmärryksistä ja epäluottamuksesta (Potanin, 2010). Toinen juopa on ohjelmoijien ja taiteilijoiden välillä. Aiemmin pelistudioiden työntekijät jakautuivat tiimeihin työn tyyppin perusteella, jolloin ohjelmoijat olivat yksi tiimi, äänistä vastaa toinen, grafiikasta vastaa kolmas ja niin edelleen. Tämä jako oli omiaan vaikeuttamaan kommunikaatiota. Ketterien menetelmien suosion myötä ollaan alettu suosia monialaisia ryhmiä, joissa on jäseniä kultakin pelikehitykseen osallistuvalla alalla. Näissäkin ryhmissä on kuitenkin havaittavissa jakoa ”ohjelmoijien” ja ”taiteilijoiden” välillä. (Kanode & Haddad, 2009; Petrillo ym., 2009) Potanin huomauttaa myös, että peliala voi olla luonteeltaan sellainen, joka houkuttaa itsenäisiä, jopa anarkistisia persoonia (Potanin, 2010).

Pelistudion ja julkaisijan keskinäinen kommunikaatio ei usein ole ongelmatonta (Petrillo ym., 2009). Ongelmia saattavat aiheuttaa muun muassa maantieteellinen ja ajallinen etäisyys (Petrillo ym., 2009). Näiden lisäksi pelistudion ja julkaisijan välillä ei usein vallitse luottamuksen ilmapiiri, mikä osittain johtuu

etäisyydestä ja osittain pelialan epäterveestä kulttuurista (Petrillo ym., 2009; Potanin, 2010).

Työkaluongelmat johtuvat yksinkertaisesti siitä, ettei pelialaa varten räätälöityjä kehitystyökaluja ole olemassa (Blow, 2004; Petrillo ym., 2009). Pelikehityksessä ohjelmoijat hyödyntävät samoja työkaluja kuin ohjelmistokehityksessä ja taiteilijat luovat kolmiulotteisia animaatiota työkaluilla, jotka on kehitetty alun perin animaatioelokuvia varten. Työkalujen skaalautuvuus pelikehitykseen on heikkoa. Olisi hyödyllistä, jos useat taiteilijat voisivat työstää samaa ympäristön osaa samanaikaisesti. Tämä ei kuitenkaan ole mahdollista nykyisillä kaupallisilla työkaluilla. (Blow, 2004)

Testausongelmat johtuvat pääasiassa aikataulujen lipsumisesta, mikä supistaa testivaiheita ajallisesti, ja kehityksen loppuvaiheeseen sijoittuvien testivaiheiden vaatiman ajan ja resurssien aliarvioinnista. Osa syystä vieritetään jälleen julkaisijan harteille, koska myös he ovat aliarvioineet laadunvarmistustoimiin vaadittavat resurssit. Toisaalta testaaminen on ollut myös onnistunutta, sillä sen avulla peliprojektien aikana löydetään huomattava määrä *ohjelmistovirheitä*, joiden korjaaminen on syönyt valtaosan kehityksen loppuvaiheista. (Petrillo ym., 2009)

Ongelmat ryhmäkoostumuksessa liittyvät ongelmiin palkata kehittäjiä ylipäänsä, ongelmiin tiimin yhteen hitsautumisessa ja ongelmiin, jotka aiheutuvat tiimin jäsenten liiallisesta kokemattomuudesta. *Budjetin ylitys* ei ole yleinen ongelma peliteollisuudessa. Petrillo ym. (2009) raportoivatkin vain kahdesta tapauksesta, joissa pelistudion on ylittänyt budjettinsa. *Kehittäjien menettämisellä* tarkoitetaan ongelmia, jotka seuraavat, kun muut eivät kykenekään jatkamaan työpaikkaa vaihtaneen tai sairastuneen kehittäjän töitä (Petrillo ym., 2009). Ilmeisesti pelikehitykseen liittyvän hiljaisen tiedon ja pelin ohjelmakoodin tuntemuksen määrä on huomattava yksilötasolla.

Yllä mainittuja ongelmia käsitellään luvussa 4.8, kun tarkastellaan ketterien menetelmien vaikutuksia pelikehitykseen.

4.2 Agile-arvot pelikehityksessä

Keith (2010) on tutkinut Agile-manifestin arvojen vastaavuutta pelikehityksen arvoihin. Hän on esittänyt seuraavia päätelmiä ja muokauksia.

Ensimmäinen arvo "yksilöitä ja vuorovaikutusta arvostetaan enemmän kuin prosesseja ja työkaluja (Agile Alliance, 2001)" säilyy muuttumattomana. Koska peli muotoutuu usean eri alan osaajien panoksesta, on erityisen tärkeää, että alojen ja osaajien välistä kommunikaatiota ei hankaloiteta tarpeettomalla hierarkialla, jollainen tavallisesti vallitsee yli 100 hengen pelistudiossa. Jos pienimmästäkin asiasta joudutaan kysymään hierarkiassa ylempänä olevalta, kuuluu kaikkien aika viestien kulkemiseen hierarkiaa pitkin. Ketterissä menetelmissä tiimi on valtuutettu päättämään asioista itse ja se kommunikoi sisäisesti välttämättä hierarkian. Tällöin hierarkiassa ylempänä olevat voivat keskittyä yle-

sempiin asioihin, ja vähän kerrallaan tiimi kasvaa kantamaan yhä suurempaa vastuuta. (Keith, 2010)

Pelialalle sovellettuna toinen arvo (Agile Alliance, 2001) kuuluu seuraavasti: "toimivaa peliä arvostetaan enemmän kuin kaiken kattavaa pelisuunnitteludokumenttia (Keith, 2010, s. 27)." Peli on enemmän kuin pelkkä ohjelmisto. Jonkinlainen pelisuunnitteludokumentti on kuitenkin hyvä olla eri sidosryhmiä ja niiden tarpeita varten. (Keith, 2010)

Pelistudion asiakas on kehityksen aikana julkaisija, joka rahoittaa pelin (Keith, 2010). Kolmas arvo pelialalle sovellettuna säilyy na: "yhteistyötä asiakkaan kanssa arvostetaan enemmän kuin sopimusneuvottelua (Agile Alliance, 2001)."

Arvon toteutumisessa on kuitenkin vielä parantamisen varaa, sillä pitkään pelistudion ja julkaisijan väliset sopimukset ovat perustuneet virstanpylväisiin, jotka ovat etukäteen neuvoteltuja vaatimuksia pelikehityksen edistymisestä. Virstanpylvään saavutettuaan pelistudio saa lisää rahoitusta julkaisijalta. Kun virstanpylväät ovat etukäteen sovittuja, ne estävät pelistudiota kehittämästä peliä parempaan suuntaan eikä julkaisija voi muuttaa virstanpylvästä enää jälkikäteen, vaikka peli voisi hyötyä siitä. Kompromissina virstanpylväät määritellään nykyään joustavimmiksi, jotta julkaisija ja pelistudio voivat tehdä yhteistyötä edes jollakin tasolla. (Keith, 2010)

Myös neljäs Agile-manifestin arvo, "Muutokseen reagoimista arvostetaan enemmän kuin suunnitelman seuraamista (Agile Alliance, 2001)", säilyy muuttumattomana. Pelikehityksessä teknologia kehittyy jatkuvasti ja lopputuloksen tulisi olla innovatiivinen, hauska peli, joten etukäteen tehtyjä suunnitelmia ei ole järkevää eikä niitä usein voida noudattaa. (Keith, 2010)

4.3 Ketterän pelikehityksen vaiheet

Luvussa 2 esiteltiin yleisesti pelikehityksessä tunnistettavissa olevat vaiheet. Pelialaa käsittelevässä kirjallisuudessa eri lähteet esittävät pelikehityksen vaiheet karkeasti ottaen samalla tavalla. Mannisen ym. esittämät vaiheet kattavat pelin elinkaaren konseptin synnyttämisestä aina ylläpidon päättämiseen saakka (Manninen ym., 2006). Seuraavaksi käsitellään Keithin (2010) ja Musilin ym. (2010) esittämät vaiheet suhteessa Mannisen ym. (2006) esittämiin vaiheisiin. Yleiskuva vaihteista ja niiden välisistä suhteista on havainnollistettu taulukossa 4.

Keith (2010) tunnistaa neljä vaihetta, jotka ovat konsepti-, esituotanto-, tuotanto- ja jälkituotantovaihe. Konseptivaiheessa kehitellään pelikonsepteja, joista yksi valitaan kehitettäväksi. Esituotantovaiheessa hiotaan pelimekaniikkaa ja etsitään pelin hauskuus sekä tuotetaan vaiheen tarpeiden mukaan taideartefakteja, minkä puitteissa myös hiotaan taiteen tuotantolinjaan seuraavaa vaihetta varten. Tuotantovaiheessa tuotetaan pelin pelattava sisältö, ja jälkituotantovaiheessa peli hiotaan julkaisukuntoon. Hän huomauttaa, että ketterissä menetelmissä pyritään eroon tällaisesta vesiputousmaisesta työn pilkkomisesta

vaiheisiin, mutta perustelee vaihejakoa niiden tarpeellisuudella sidosryhmiä varten ja havainnollistamaan tiimin muuttuvia työskentelytapoja, jotka riippuvat kulloisestakin vaiheesta. Vaiheet ovat jossain määrin päällekkäisiä eivätkä vaiheiden rajat ole selkeitä, vaan niissä on väistämättä liukumaa, jonka aikana siirrytään asteittain vaiheesta toiseen. Esimerkiksi konseptivaihe läpäisee lähes koko kehitysajan. (Keith, 2010)

Musil ym. (2010) puolestaan tunnistavat vain kolme vaihetta Scrumiin perustuvassa pelikehitysprosessissaan. Nämä vaiheet ovat esituotanto- ja tuotantovaiheet sekä projektin päättäminen. Esituotantovaiheessa etsitään toteuttamiskelpoinen peli-idea rakentamalle prototyyppejä. Esituotantovaiheessa myös tehdään riskianalyysi ja hankitaan rahoitus. Tuotantovaiheessa tuotetaan valmis peli. Projektia päätettäessä huolehditaan pelin jakelusta, analysoidaan mennyt projekti ja kootaan peliin liittyvät artefaktit paketiksi, jonka avulla pelin ylläpito on mahdollista. Vaiheet ovat ajallisesti erillisiä, ja vaiheiden välillä siirrytään ilman liukumaa.

Nämä kolme vaihejakoa on koottu yhteen taulukkoon (taulukko 4), jotta ketterän pelikehityksen vaiheita voidaan verrata suhteessa Mannisen esittämiin yleisiin pelikehityksen vaiheisiin. Keithin esittämät vaiheet vastaavat karkeasti Mannisen ym. esittämiä vaiheita konsepti-, esituotanto- ja tuotantovaiheiden osalta. Keithin mukaan jo konseptivaiheessa voidaan tehdä prototyyppi konseptin esittelyä varten. Keithin jälkituotantovaihe sisältää Mannisen ym. esittämistä vaiheista laadunvarmistus- ja testausvaiheen ja julkaisuvaiheen. Ylläpito-vaihetta ei tarkastella lainkaan. (Keith, 2010; Manninen ym., 2006)

TAULUKKO 4 Pelikehityksen vaiheet eri lähteiden mukaan

	Pelikehityksen vaiheet					
Manninen ym.	Konseptivaihe	Esituotantovaihe	Tuotantovaihe	Laadunvarmistus- ja testausvaihe	Julkaisuvaihe	Ylläpito-vaihe
Keith	Konseptivaihe	Esituotantovaihe	Tuotantovaihe	Jälkituotantovaihe		
Musil ym.	Esituotantovaihe		Tuotantovaihe		Projektin päättäminen	

Musilin ym. (2010) esituotantovaihe sisältää Mannisen ym. vaiheista konseptin luomisen ja esituotantovaiheen, tuotantovaihe sisältää Mannisen ym. tuotantovaiheen sekä laadunvarmistus- ja testausvaiheen. Musilin ym. projektin päättäminen vastaa Mannisen ym. julkaisuvaihetta. (Manninen ym., 2006; Musil, Schweda, Winkler & Biffel, 2010a)

Asioiden käsittelyn selkiyttämiseksi tullaan tässä luvussa käyttämään jatkossa neljää vaihetta, jotka ovat konsepti-, esituotanto-, tuotanto- ja jälkituotantovaihe. Tähän Keithin (2010) esittämään vaihejakoon päädyttiin, koska kirjallisuudessa ei ole esitetty ketterien menetelmien hyödyntämistä pelien ylläpito-vaiheessa, vaan niiden hyödyntäminen rajoittuu kehitysvaiheeseen. Lisäksi

Keithin (2010) vaihejako on uskottavampi, koska se perustuu käytännön kokemuksiin pelikehityksestä.

Ketterien menetelmien soveltaminen eri vaiheissa on hyvin erilaista Musilin ym. (2010) ja Keithin (2010) esityksissä. Koska pelikehityksen eri vaiheissa tapahtuvat varsinaiset pelikehitystoimet on esitelty jo luvussa 2, keskitytään seuraavaksi niiden sijaan eri vaiheissa hyödynnettäviin menetelmiin (alaluku 4.4) ja käytänteisiin (alaluku 4.5).

4.4 Ketterien menetelmien hyödyntäminen pelikehityksen eri vaiheissa

Keithin (2010) esityksessä *konseptivaiheessa* hyödynnetään Scrumia, mutta pyrähdysten kestot ovat lyhyempiä. Pyrähdysten työlistat ovat pieniä ja suurin osa tehtävistä on aikarajattuja (engl. time boxed). Tällaisia tehtäviä kutsutaan piikeiksi (engl. Spikes). (Keith, 2010) Musil ym. (2010) esittävät konseptivaiheen iteraatiosilmukkana, jossa toimivaa konseptia etsitään prototyyppien avulla, mutta he eivät eksplisiittisesti ehdota ketterien menetelmien hyödyntämistä tässä vaiheessa.

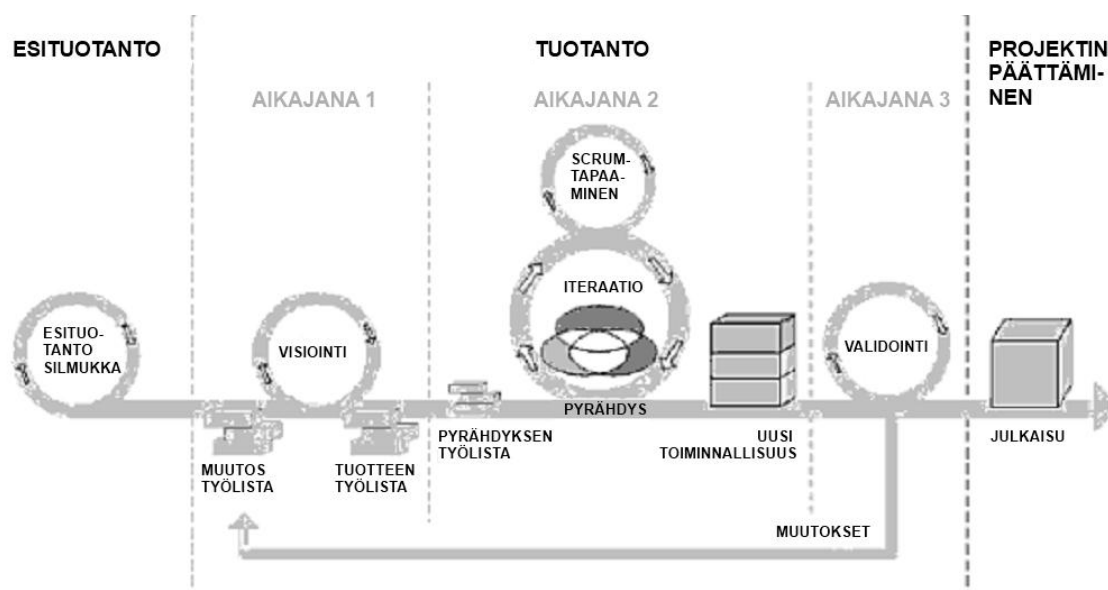
Pelikonseptin kehittämiseen on Cook (2002) esittänyt evoluutionäristä suunnittelua (engl. evolutionary design). Siinä on ideana luoda säännöt, testata niitä käytännössä ja muokata sääntöjä havaintojen perusteella kohti halutun kaltaista peliä (Cook, 2002). Kuvattu menetelmä muistuttaa pelikehitykselle ominaista suunnittelu-toteutus-testaus -sykliä (Manninen ym., 2006) ja soveltuu sellaisenaan hyödynnettäväksi Scrumin pyrähdyksissä.

Keithin (2010) mukaan *esituotantovaiheessa* Scrumia pystytään hyödyntämään puhtaimmillaan tiimin toteuttaessa keskeisimpiä osia pelimekaniikasta niin taiteen kuin ohjelmakoodin osalta. Scrumin lisäksi ohjelmoijat hyödyntävät XP:n käytänteitä kattaen koko pelin kehityskaaren. Esituotantovaiheen aikana tiimi tarkentaa jokaisen pyrähdysten myötä arviotaan tuotantovaiheeseen varattavasta ajasta, kun yksittäisten taidepiikkien aikarajat tarkentuvat, tieto tarvittavista taideartefakteista tarkentuu ja niiden tuottamiseen tarvittava tuotolinja hioutuu. (Keith, 2010)

Musil ym. (2010) esittävät esituotanto- ja tuotantovaiheessa hyödynnettäväksi pelikehitykseen muokattua Scrumia, jossa kolme rinnakkaista aikajanaa jakavat saman pyrähdysten. Musilin ym. (2010) esittämä Scrum -prosessimallia on havainnollistettu kuviossa 3, joka on heidän artikkelistaan lainattu ja käännetty suomenkieliseksi. Aikajanat ovat visiointi, pyrähdys ja validointi.

Visioinnin tehtävät ovat verrattavissa tuotteen omistajan tehtäviin normaalissa Scrumissa, mutta niihin ottavat osaa myös tekniset ja taiteelliset sidoryhmät. Visiointi huolehtii projektin kokonaiskuvasta ja on vastuussa tuotteen työlistan ylläpitämisestä. Tuotteen työlistan lisäksi pidetään yllä myös muutostyölistää, joka kootaan validointiprosessin muutosehdotuksista. Suuret muutokset kokeillaan toteuttaa ensin prototyyppinä ja lisätään muutostyölistalle.

Pienemmät priorisoidaan ja lisätään listalle tai hylätään. (Musil, Schweda, Winkler & Biffel, 2010a)



KUVIO 3 Ketterän pelikehityksen prosessimalli (Musil, Schweda, Winkler & Biffel, 2010a, s. 89)

Musilin ym. (2010) pyrhdyks on muutoin tavanomainen Scrumin pyrhdyks, mutta eri alat integroidaan osaksi pyrhdyks omilla tyonkuluillaan, jotka suoritetaan rinnakkain pyrhdyksen aikana. Esimerkiksi ohjelmoijille tyonkulku voi olla testilahttoista kehitysta. Taidealojen tyonkulkujen kehittaminen on vaikeampaa. Usein niissa sovelletaan vetojarjestelmiä (engl. pull systems) tai Scrum-johdannaisia, kuten Scrumbania (Ladas, 2009). Paivittaisilla Scrum-tapaamisilla pidetaan tiimin jaset ajan tasalla pyrhdyksen edistymisesta ja koordinoidaan eri alojen toimijat. (Musil, Schweda, Winkler & Biffel, 2010a)

Validointi on vastuussa laadunvarmistuksesta ja virstanpylvaiden hyvaksymisesta. Prosessissa hoidetaan myos tavalliseen pyrhdykseen ajallisesti liian suuret testustoimet. Pyrhdyksen lopuksi kaikki testitulokset annetaan validointiprosessille. (Musil, Schweda, Winkler & Biffel, 2010a)

Keith (2010) on havainnut, etta Scrum ei sovi hyodynnettavaksi *tuotantovaiheessa* tuotantolinjojen paaasiallisena prosessina. Tämän takia hän esittääkin Lean-ajatusmallien (Poppendieck & Poppendieck, 2003) ja Kanbanin (Kniberg & Skarin, 2009) hyodyntamista Scrumin rinnalla. Tuotantovaiheessa tiimit, jotka ovat osa tuotantolinjaa, siirtyvat hyodyntamaan laajennettua Kanban -taulua pyrhdyksen tyolistan sijaan ja pyrkivat parantamaan prosessiaan jatkuvasti Leanin esittamien oppien mukaisesti. Kanban -taulun kaikki tehtavat ovat piikkeja. (Keith, 2010) Nämä tiimit toimivat kuitenkin edelleen samassa tahdissa muiden tiimien pyrhdyksien kanssa ja jatkavat joidenkin Scrumin kaytanteiden, kuten paivittaisen Scrum -tapaamisen, hyodyntamista. He myos esittelevat pyrhdyksen katselmoinnissa aikaansaannoksensa. Tuotantotiimit eivat kuiten-

kaan suunnittele pyrhdyistä eivätkä näin ollen laadi pyrhdyksen työlisterää, vaan he työskentelevät kunnes Kanban-taulun tehtävälisterä on tyhjä, jonka jälkeen se täytetään jälleen sovitulla määrällä uusia tehtäviä. Mikäli tiimi ei ole puhtaasti tuotantotiimi, se käyttää pyrhdyksen aikana sekä Kanban-taulua tuotantoon osallistuville että tavallista pyrhdyksen työlisterää ja Kanban-taulua muille jäsenille. Tällöin pyrhdyks täytyy myös suunnitella ja pyrhdyksen työlisterä laatia niiden henkilöiden toimesta, joita ne koskevat. (Keith, 2010)

Jälkituotantovaiheessa työnkuva Keithin (2010) mukaan muuttuu jälleen. Kun taideartefaktien tuotanto on takana ja peli on ominaisuuksiltaan valmis, keskitytään enemmän hiomiseen, säätämiseen ja virheiden korjaamiseen. Tässä vaiheessa tehtävät tunnistetaan lyhyemmällä aikavälillä kuin kokonaiset pyrhdykset. Niinpä tehtävät lisätään päivittäiseen työlisterään, kun niitä havaitaan. Pyrhdyksen katselmointi jatkuu edelleen. (Keith, 2010) Musilin ym. (2010) tuotantovaiheeseen esittämä prosessi jatkuu samanlaisena jälkituotantovaiheessa.

4.5 Ketterät käytänteet pelikehityksessä

Eri lähteet nimeävät yksittäisiä, pelikehityksessä hyödynnettäviä ketterien menetelmien käytänteitä vaihtelevasti. Musil ym. (2010) eivät mainitse omassa Scrumiin perustuvassa prosessimallissaan kaikkia Scrumin käytänteitä. Koska prosessimalli on Scrumia laajentava, voidaan olettaa, että Scrumissa määriteltyjä käytänteitä hyödynnetään samalla tavoin kuin normaalisti, ellei toisin mainita. Keith (2010) puolestaan mainitsee useita käytänteitä niin XP:stä, Scrumista, Leanista kuin Kanbanistakin. Osa käytänteistä hyödynnetään sellaisenaan, mutta toisia on muokattu pelialan vaatimuksia vastaaviksi ja osa räätälöidään pelikehityksen eri vaiheisiin eri tavalla.

XP:n käytänteistä informatiivinen työtila, jota Keith nimittää Scrum-tauluksi, pariohjelmointi, jatkuvat integrointi ja testilähtöinen kehitys sekä tarinat ovat keskeisessä asemassa ketterässä pelikehityksessä (Keith, 2010). Myös tiheät julkaisut (engl. release) pelikehityksessä juontuvat oletettavasti XP:n käytänteistä. Osa käytänteistä toimii sellaisenaan pelikehityksessä, mutta esimerkiksi jatkuvan integroinnin tulee käsittää myös taideartefaktien integrointi (Keith, 2010).

Scrumin käytänteistä kaikki ovat Keithin (2010) mukaan hyödynnettävissä ketterässä pelikehityksessä. Osa käytänteistä räätälöidään vastaamaan kehitysvaiheen tarpeita, kuten aiemmin on mainittu. Esimerkiksi pyrhdyksen pituudet vaihtelevat; jälkituotantovaiheessa pyrhdyksen jokainen päivä suunnitellaan erikseen ja tuotantolinjan osana olevat tiimit korvaavat tuotantovaiheessa pyrhdyksen työlisterän Kanban-taululla, jonka odottavien tehtävien saraketta täydennetään aina sen tyhjennyttyä. (Keith, 2010)

Tobey (2008) raportoi organisaationsa hyödyntävän pyrhdyksen puolesta välissä puolipyrhdyksetapaamista (engl. mid-sprint meeting), jossa varmistetaan, että tiimi saa pyrhdyksen tehtävät valmiiksi. Samassa tapaamisessa on mahdollista arvioida tarinoiden toteutusta tuotteen omistajan sekä julkaisijan

tai muun asiakkaan edustajan kanssa ja muuttaa kurssia, mikäli tarve vaatii. Toinen saman organisaation sovellus on tuotteen työlistan tarinoiden priorisointi ja sijoittaminen kalenteriin pitkän aikavälin suunnittelun havainnollistamiseksi (Tobey, 2008).

Pelistudio voi olla kooltaan suuri, jolloin tiimien määrä kasvaa, ja sen myötä kokonaisprojektin hallinta vaikeutuu ja tiimien väliseen kommunikaation pitää panostaa. Scrumissa mekanismi tiimien väliseen kommunikaation ja tiimien hallinnointiin on Scrumien Scrum -tapaaminen, jolla tarkoitetaan sovittuina aikoina järjestettäviä tapaamisia, joihin osallistuu yksi jäsen kustakin tiimistä. Tämä tapaaminen muistuttaa hieman päivittäistä Scrum-tapaamista mutta on säännöiltään poikkeava. Scrumien Scrum -tapaamisessa tarkoituksena on selvittää, miten kunkin tiimin työ vaikuttaa muiden tiimien toimintaan. Tapaamisessa on myös pyrittävä ongelmien ratkaisuun, mikäli sellaisia ilmenee. Tämän takia tapaamisella ei ole tiukkaa aikarajaa, vaan se jatkuu niin kauan kuin on tarvetta. (Keith, 2010)

Seuraavaksi tarkastellaan ketterien käytänteiden käyttöä tuotantovaiheessa Keithin (2010) mukaan. Hän ehdottaa, että tuotantolinjaan kuuluvat tiimit hyödyntävät Kanbania kokonaisuudessaan tuotantovaiheessa. Kuten aiemmin mainittiin, tuotantotiimit hylkäävät pyrähdysten työlistan ja korvaavat sen Kanban-taululla, jonka odottavien tehtävien sarake täytetään sarakkeen tyhjentymistä. Varsinaista tuotantolinjaa parannetaan aina esituotantovaiheen kokeiluista läpi koko tuotantovaiheen Lean-periaatteiden mukaisesti. Tehtävän eteneminen tuotantolinjalla visualisoidaan Kanban-taululla, jossa kukin työvaihe esitetään omana sarakkeenaan. Koska työvaiheiden kestot vaihtelevat, tehtävät eivät solju linjan läpi tasaisesti. Tämän ratkaisemiseksi kunkin työvaiheen aikaraja (engl. timebox) selvitetään. (Keith, 2010)

Työvaiheen aikaraja on optimaalinen aika, jonka puitteissa työvaihe saadaan valmiiksi sopivat laatukriteerit täyttyen. Näin ollen aikarajaa säätelemällä säädellään työvaiheen lopputuloksen laatua. Aikarajojen ei ole tarkoitus pakottaa työntekijöitä ”kuolonmarssiin” saadakseen tehtävän valmiiksi! Usein alustavat aikarajat eri työvaiheille saadaan selville jo esituotantovaiheen artefakteja luotaessa, ja tuotantovaiheessa, kun tuotannon tehokkuutta kehitetään edelleen, aikarajat tarkentuvat entisestään. Kun jokaisen työvaiheen aikaraja on selvillä, voidaan tehtävien kulku linjan läpi tasata siten, että eri vaiheet tasapainotetaan lisäämällä resursseja vaiheisiin, jotka aiheuttavat pullonkaulan. Työvaiheeseen resursoitujen henkilöiden määrän mukaan määräytyy Kanban-taulun sarakkeiden tehtävämäärän rajoitukset. (Keith, 2010)

Lean-periaatteiden mukaisesti tuotantolinjan hukkaa voidaan vähentää ja tehtävien läpimenoaikaa parantaa useilla tavoilla. Keith (2010) on tunnistanut neljä keskeistä tekijää, jotka vaikuttavat tuotantolinjan läpimenoaikaan. Nämä ovat artefaktin koko, erä koko, hukka ja tiimin kyvyt ja valtuuttaminen.

Työstettävän *artefaktin koko* (engl. asset size) vaikuttaa läpimenoaikaan, palautesilmukan pituuteen ja iteraatioon. Mitä suurempia artefaktit ovat, sen hitaampi on läpimenoaika. Koska artefaktien tuottaminen kestää pitempään, niiden sopivuudesta saadaan palautetta myöhemmin ja parannusten tekeminen eli

iteraatioiden tekeminen hidastuu. Mikäli mahdollista, artefaktit tulee pitää pieninä. Esimerkiksi pelimaailman eri alueet voidaan pilkkoa pienempiin osiin tuotantoa varten. Tällöin esimerkiksi ensimmäisen osan tuotannossa havaitut huonot ratkaisut osataan välttää seuraavia osia tuotettaessa. (Keith, 2010)

Eräkoolla (engl. batch size) tarkoitetaan työvaiheessa kerralla työstettävien artefaktien määrää. Yleensä eräkoko on yksi. Jos eräkoko kuitenkin pidetään mahdollisimman pienenä ja jossakin työvaiheessa ilmenee ongelmia, ongelma heijastuu kaikkiin seuraaviin työvaiheisiin. Tällaisen tilanteen estämiseksi voidaan Kanban-tauluun lisätä työvaiheiden väliin puskurisarakkeet, joiden tehtävämäärä tulee rajoittaa mahdollisimman pieneksi, jolloin eräkoko ei voi päästä kasvamaan. Suuret eräkoot kasvattavat läpimenoaikaa ja työ saattaa alkaa kasautua. (Keith, 2010)

Tuotantolinjalla *hukkaa* (engl. waste) on kaikki työ, joka ei suoraan lisää lopputuloksen arvoa. Hukkaa esiintyy monessa muodossa. Artefaktin hyväksyttämisen prosessi on hukkaa, jos sitä joudutaan odottamaan. Puutteelliset tai hitaat työkalut ovat hukkaa. Huonot työskentelytavat ja työn siirtely ovat hukkaa. Keith (2010) kertoo tilanteesta, jossa konseptitaiteilija tuotti useita piirroksia istuessaan erillään tämän tuotoksia käyttävistä taiteilijoista. Kun taiteilijat istutettiin yhteen työtilaan, konseptitaiteilija tuotti vähemmän piirroksia ja lopullisten artefaktien laatu parani, koska hän sai parempaa palautetta heidän keskustellessaan enemmän yhdessä konseptien ja lopullisten artefaktien tarkoituksesta. (Keith, 2010)

Tiimin *valtuuttaminen* päätöksien tekoon ja omista työskentelytavoista sekä käytänteistä päättämiseen edesauttaa sitoutumista työhön, mikä suoraan näkyy tiimin kyvyssä parantaa tuotantolinjaa. Kanbanin ja Lean-periaatteiden käyttö tuo läpinäkyvyyttä tuotantolinjaan niin sen haasteiden kuin vahvuuksien osalta. Sama läpinäkyvyys paljastaa erot tiimien *kyvyssä*, jolloin heikompien tiimien opastamiseen ja mentorointiin voidaan panostaa. (Keith, 2010)

4.6 Tiimit

Vaikka sekä XP että Scrum vaativat poikkitoiminnallisia tiimejä (Beck & Andres, 2004; Schwaber & Beedle, 2001), ei pelikehityksessä tällaisten luominen aina ole mahdollista saati paras ratkaisu (Keith, 2010). Niinpä Keith (2010) on esittänyt erityisiä tiimityyppejä, jotka ovat vastaus pelikehityksen ominaispiirteisiin ja esiintyvät normaalien poikkitoiminnallisten tiimien rinnalla.

Toimintotiimi (engl. feature team) on XP:n ja Scrumin käsityksen mukainen poikkitoiminnallinen tiimi, joka on vastuussa yhden keskeisen pelitoiminnon toteuttamisesta. Usein nämä tiimit joutuvat jakamaan muiden tiimien kanssa tietyn alan osaajan, koska käytännössä ei aina ole mahdollista, että tiimiin kuuluisi aina osaaja kaikilta toiminnon rakentamiseen vaadittavilta aloilta. (Keith, 2010)

Eriyisosaajatiimi (engl. functional team) on pääosin yhden alan osaajista koostuva tiimi, joka on vastuussa keskeisestä pelikehityksen osa-alueesta.

Yleensä heidän tehtävänä on olla vastuussa osa-alueesta, joka vaatii erityisesti tiettyä erikoisosaamista ja joka olisi muille tiimeille liian vaikea. Esimerkki tällaisesta tiimistä on ohjelmakoodin optimoinnista tietyllä alustalla vastaava tiimi, joka koostuu paljon kyseiselle alustalle ohjelmoineista ohjelmoijista. (Keith, 2010)

Tuotantotiimi (engl. production team) on tiimi, jonka tehtävänä on tuottaa taideartefakteja peliä varten. Tiimillä on määritelty tuotantolinja, ja he hyödyntävät sen optimoinnissa Lean-ajatusmalleja sekä Kanbania. Tuotantotiimit muodostetaan usein esituotantovaiheen jälkeen toimintotiimistä, kun toiminto on saatu valmiiksi. Tällöin tiimin jäsenistö vaihtuu, ja jäsenistö koostuu pääasiassa tuotantolinjaan osallistuvista taiteilijoista. Pyrähdysten tarpeista riippuen tuotantotiimit voivat pyrähdysten välissä vaihtaa jäsenistöä keskenään. (Keith, 2010)

Jaettu tukitiimi (engl. shared infrastructure team) on usealle peliprojektille tukitoimintoja tarjoava tiimi, joka voi keskittyä esimerkiksi pelimoottorin muokkaamiseen tai tuottamaan välianimaatioita tai ääntä. Näillä tiimeillä tulisi erilaisten riippuvuuksien takia olla oma työlista ja tuotteen omistaja, jolla on päätäntävaltaa ratkaista ristiriitatilanteet riippuvuuksien välillä, ja tiimin tulisi osallistua heidän palveluitaan tarvitsevien tiimien julkaisujen suunnittelukokouksiin (näitä voidaan verrata XP:n vuosineljännessyklin kokoukseen). Tukitiimin tulisi myös aikatauluttaa ylläpitotehtäviä pyrähdyksiin. Jaettu tukitiimi toimii mallina julkaistuja pelejä ylläpitävälle tiimille. (Keith, 2010)

Työkalutiimi (engl. tool team) luo työkalut pelikehitystä varten tuotantolinjojen tuotantotiimeille ja muille tiimeille. Myös työkalutiimin asiakkaat saattavat tulla eri pelejä työstävistä projekteista. (Keith, 2010)

Varatiimi (engl. pool team) koostuu yhden, usein taiteellisen, alan osaajista. Varatiimit ovat sananmukaisesti varalla, ja heidän tehtävänä on tukea muita tiimejä. Varatiimillä ei ole omaa työlistää. Varatiimejä muodostetaan yleensä vasta esituotannon loppuvaiheessa, kun kapasiteettia tuotantolinjoille tarvitaan siirryttäessä tuotantovaiheeseen. (Keith, 2010)

Integraatiotiimillä (engl. integration team) tarkoitetaan tiimiä, joka on vastuussa lopullisesta pelikokemuksesta. Se on samanlainen kuin toimintotiimi. Tämä tiimi integroi eri pelitoiminnot ja hioo niiden yhteistoiminnan oman näkemyksensä mukaiseksi. Näin tehdään yhtenäisen vision toteutumiseksi, koska eri toimintotiimien näkemykset lopullisesta pelistä voivat kehityksen edetessä ajautua erilleen. Integraatiotiimi huolehtii, että lopullisessa pelissä tällaista ei havaita. (Keith, 2010)

Edellä kuvatut tiimit saattavat vaatia useamman kuin yhden tuotteen omistajan hallitsemaan erillisiä työlistoja ja tarinoiden priorisointia. Tästä syystä Keith (2010) ehdottaakin ratkaisuksi mallia, jossa joukko tuotteen omistajia muodostavat matalan hierarkian, jossa yksi tuotteen omistaja on päävastuussa ja delegoi muille tuotteen omistajille vastuuta tiimeistä. Myös päävastuussa oleva tuotteen omistaja toimii tuotteen omistajana jollekin määrälle tiimejä. (Keith, 2010)

Vaikka poikkitoiminnalliset tiimit parantavat kommunikaatiota, voi ongelmaaksi muodostua saman alan osaajien välisen kommunikaation puute. Eri tiimien grafiikkaohjelmoijat voivat painia saman ongelman parissa toisistaan tietämättä ja ratkaista ongelman eri tavalla. Vaikka tämä ei varsinaisesti olekaan ongelma, olisi silti hyödyllistä jakaa ratkaisuun ja ongelmiin liittyvää tietoa muiden saman alan osaajien välillä. Saman alan osaajien välisen kommunikaation parantamiseksi on ehdotettu kaikki tiimit läpäiseviä ammattiyhteisöjä, jotka voivat keskinäisten sopimusten mukaisesti kokoontua käsittelemään erityisesti oman alan työskentelyyn ja ongelmiin liittyviä kysymyksiä. (Keith, 2010)

4.7 Roolit

Ketterien menetelmien soveltamista pelialalla kuvaavissa lähteissä ei ole esitelty uusia rooleja. Poikkitoiminnallinen tiimi koostuu useamman alan osaajista kuin ohjelmistokehityksessä mutta näyttäisi siltä, että uusia rooleja ei ole ollut tarvetta määrittellä. Keith (2010) on kuitenkin keskustellut erilaisten pelialalla perinteisesti esiintyvien nimikkeiden soveltumisesta eri rooleihin, joita ketterien menetelmien hyödyntäminen tuo tullessaan, ja näiden nimikkeiden työnkuvan muuttumisesta ketterien menetelmien myötä.

Perinteisesti pelialalla on eri osaamisalueita johtaneet henkilöt, jotka ovat johtaneet päivittäistä työskentelyä. Ketterien menetelmien myötä näiden henkilöiden rooli on muuttunut, koska tiimit päättävät päivittäisestä tekemisestään itse. Johtohenkilöiden uusi rooli on johtaa projektia korkeammalla tasolla, osallistua pelisuunnitteluun ja opettaa tai mentoroida oman osaamisalansa työntekijöitä. Taiteellinen johtaja (engl. art lead) huolehtii lisäksi Lean-ajatusmallien tuottamien tuotantolinjaparannusten jakamisesta kaikille tuotantotiimeille. (Keith, 2010)

Tuotteen omistajan rooliin Keith (2010) kelpuuttaa useiden eri nimikkeiden haltijoita mutta varoittaa tiettyjen nimikkeiden mahdollisista heikkouksista. Esimerkiksi tuottaja (engl. producer), joka vastaa hyvin pitkälti elokuva-alan tuottajaa, sopii hyvin tuotteen omistajaksi etenkin, jos tällä on hyvä näkemys markkinoiden tarpeista. Tuottaja sopii myös tuotteen omistajaa avustavaan rooliin, jolloin tuottaja keskittyy enemmän lisenssi-, budjetti- ja resurssiasioiden hoitamiseen. (Keith, 2010)

Myös pääsuunnittelija voi jossain määrin soveltua tuotteen omistajaksi, koska tämä osaa kommunikoida visionsa kaikille kehitykseen osallistuvien alojen osaajille, tämä edustaa pelaajan näkökulmaa peliin ja pelisuunnittelu on tuotevisiota eteenpäin ajava voima. Usein pääsuunnittelijalta kuitenkin puuttuu kykyjä liittyen taloudelliseen ajatteluun ja projektin hallintaan. (Keith, 2010)

Tuottaja sopii myös Scrum-mestariksi, koska hän osaa kommunikoida eri alojen kanssa ja tällä on luonnostaan roolin tarvitsemää valtaa ja mahdollisesti myös aikaa ajaa tiimien asioita. Sudenkuoppana tuottajan asettamisessa Scrum-mestariksi on tuottajan vanhan roolin mukanaan tuoma tapa alaisten komentamiseen, mikä ei istu ketterään organisaatioon. (Keith, 2010)

4.8 Ketterä kehittäminen ratkaisuna pelialan ongelmiin

Alaluvussa 4.1 esiteltiin useita ongelmia, joita on havaittu ohjelmistokehityksessä ja pelikehityksessä. Perinteisesti ohjelmistokehityksessä havaitut ongelmat vaikuttavat enemmänkin vesiputousmalliin tai muun vaiheistettuun kehittämiseen liittyviltä ongelmilta, koska lähes kaikki Petrillon ym. (2009) käyttämät lähteet ovat ajalta ennen Agile-manifestia ja ketterien menetelmien suosion nousua. Sama tilanne pätee pelikehityksen ongelmia käsittelevän osan lähteisiin. Tästä voidaan siis päätellä, että hyvin suurella todennäköisyydellä kummastakaan alasta, jota Petrillo ym. (2009) ovat tutkimuksessaan käsitelleet, ei ole saatu tutkimuksen ilmestymisvuotta vastaavaa tulosta. Tämä ei kuitenkaan ole ongelma, sillä se tarjoaa erinomaisen mahdollisuuden selvittää ongelmien olemassa oloa nykyaikaisessa pelikehityksessä. Missään nimessä ei kuitenkaan pidä väittää, etteikö Petrillon ym. (2009) tutkimuksessaan havaitsemat pelialan ongelmat olisivat olleet todellisia tutkimuksen kohteena olleena aikakautena (1999 - 2002).

Ohjelmistokehityksessä havaitut ongelmat liittyvät aikatauluun, budjettiin, laatuun ja projektinhallintaan, jotka on havaittu myös pelialalla (Petrillo ym., 2009). Voidaan väittää, että kaikki neljä riippuvat toisistaan. Pelialalla jäykät virstanpylväät ja ennakkoon sovitut aikataulut rampauttavat pelin kehittämisen oikeaan suuntaan, jolloin suora seuraus on laadun kärsiminen (Keith, 2010). Laatuun vaikuttavat myös laadunvarmistustoimet, jotka ketterissä menetelmissä ovat olennainen osa ohjelmistokehitystä testilähtöisen kehityksen ja jatkuvan integroinnin muodossa, sekä läheinen yhteistyö asiakkaan kanssa (Beck & Andres, 2004). Petrillon ym. (2009) mukaan projektinhallinnan ongelmat johtuivat huonosta kommunikaatiosta, huonosta projektin seurannasta ja siitä, että tiimiin ei panostettu. Kaikki nämä ovat asioita, joihin ketterissä menetelmissä erityisesti keskitytään (Agile Alliance, 2001; Beck & Andres, 2004; Schwaber & Beedle, 2001).

Petrillon ym. (2009) mukaan pelikehitykselle ominaisia ongelmia ovat ongelmat projektin mittakaavan (engl. scope) määrittelyssä, epärealistinen mittakaava, ominaisuuksien kasvu (engl. feature creep), ominaisuuksien karsiminen kehityksen aikana, aikataulutusergelmat, loppurutistus (eng. Crunch tai crunch time), teknologiset ongelmat, ongelmat suunnitteluvaiheessa, viivästykset, dokumentaation puute, kommunikaatio-ongelmat, työkaluongelmat, testausongelmat, ohjelmistovirheiden määrä, ongelmat ryhmäkoostumuksessa, budjetin ylitys ja kehittäjien menettäminen.

Mistä Petrillon ym. (2009) tutkimuksessaan havaitsemat pelialan ongelmat kumpuavat? Syitä on useita. On hyvin todennäköistä, että esimerkiksi Potaninin (2009) havainnot kehittäjän ja julkaisijan välisestä suhteesta ovat osa ongelmaa. Voidaan kuitenkin väittää, että tuo suhde on vain jäävuoren huippu, joka on seurausta muista perustavanlaatuisista ongelmista. Jos tarkastellaan pelialalla perinteistä julkaisijan ja pelistudion välistä sopimusmallia, jossa pelistudio sitoutuu ennen kehityksen alkua sovittuihin virstanpylväisiin (Keith, 2010), al-

kaa ongelmien lähde paljastua. Tällainen sopimus pakottaa myös pelistudion laatimaan aikataulunsa päivämäärän tarkasti pitkälle tulevaisuuteen. Miten kumpikaan osapuoli voi ennustaa kuukausia tai vuosia eteenpäin virstainpylväitä, kun pelikehitys on luonteeltaan iteratiivista, luovaa työtä?

Näyttäisi siltä, että hyvin monet Petrillon ym. (2009) tunnistamista pelialan ongelmista kumpuavat pelikehitysmenetelmien ja itse pelikehityksen luonteen huonosta yhteensopivuudesta, mihin pelikehityksen osapuolet pakottavat toisensa. Seuraavaksi käydään läpi Petrillon ym. (2009) mainitsemat pelikehityksen erityiset ongelmat ja pohditaan, miten niitä voitaisiin lieventää ketterän kehittämisen keinoin. Tulokset on koottu taulukkoon 5.

TAULUKKO 5 Pelialan ongelmat ja niihin vaikuttavat ketterien menetelmien käytänteet tai toimintatavat

Pelialan ongelma	Ongelmaan vaikuttava ketterä käytänte tai toimintatapa
Projektin mittakaavan määrittely	Tuotannon resurssien ja aikataulun arviointi esituotantovaiheessa kerättävän tietämyksen perusteella (Keith, 2010).
Epärealistinen mittakaava	Läpinäkyvyys (Beck & Andres, 2004; Poppendieck & Poppendieck, 2003; Schwaber & Beedle, 2001).
Ominaisuuksien kasvu	Tuotteen työlistan priorisointi (Schwaber & Beedle, 2001).
Ominaisuuksien karsiminen	
Aikataulutusetongelmat	Ketterä pelikehitys Keithin (2010) mukaan.
Tekniset ongelmat	Teknologioiden kokeilu esituotantovaiheessa (Keith, 2010).
Suunnitteluvaiheen ongelmat	Ketterä pelikehitys Keithin (2010) mukaan.
Viivästykset	Aikataulun neuvottelu, projektin mittakaavan muutokset, virstainpylväiden järjeistäminen (Keith, 2010). Estimointi (Schwaber & Beedle, 2001).
Dokumentaation puute	Ketterä pelikehitys Keithin (2010) mukaan.
Kommunikaatio-ongelmat	Poikkitoiminnalliset tiimit (Beck & Andres, 2004; Poppendieck & Poppendieck, 2003; Schwaber & Beedle, 2001). Ammattiyhteisöt (Keith, 2010).
Ongelmat ryhmäkoostumuksessa	
Kehittäjien menettäminen	
Ohjelmistovirheet	Pariohjelmointi ja testilähtöinen kehittäminen (Beck & Andres, 2004).
Testausongelmat	
Työkäluongelmat	Työkäluutiimi (Keith, 2010).

Ongelmat *projektin mittakaavan määrittelyssä* ja *epärealistinen mittakaava* ovat ongelmia, jotka todennäköisesti liittyvät konsepti- ja esituotantovaiheeseen. Kuten Keith (2010) on esittänyt, molempien tulisi ratketa esituotantovaiheessa, kun pelitoiminnot ja tuotantoon vaadittavat resurssit tarkentuvat (taulukko 5). Ketterien menetelmien tarjoama läpinäkyvyys helpottaa sen arvioimista (Beck & Andres, 2004; Poppendieck & Poppendieck, 2003; Schwaber & Beedle, 2001), voiko pelistudio ylipäätään saavuttaa tavoitteen alustavan aikataulun ja budjetin puitteissa. Tämä puolestaan vähentää *loppurutistuksen* tarvetta.

Petrillon ym. (2009) mainitsemista ongelmista *ominaisuuksien kasvu* (engl. scope creep) ja *ominaisuuksien karsiminen* ovat ongelmia, joihin ketterät menetelmät ainakin osittain tarjoavat vastauksen. Koska uudet tarinat priorisoidaan, tärkeimmät tulevat seuraavien pyrähdysten työlistalle (Schwaber & Beedle, 2001). Näin ollen vain tärkeät ominaisuudet tulevat toteutetuksi, mikäli aika ei riitä kaikkeen. Tämä vähentää ominaisuuksien kasvun aiheuttamia ongelmia ja samalla mekanismilla ominaisuuksien karsimisesta koituvat ongelmat pienenevät, koska karsitut ominaisuudet eivät alun perinkään olleet tärkeitä.

Aikataulutusetongelmat johtuvat aikataulun orjallisesta seuraamisesta esimerkiksi siirtymällä esituotantovaiheesta tuotantovaiheeseen. Mikäli pelimekaniikkaa ja keskeisiä pelitoimintoja ei vielä ole saatu valmiiksi, saattavat niihin kohdistuvat muutokset aiheuttaa merkittävää hukkatyötä ja lisätyötä taiteellisten artefaktien osalta. Tämä vaikuttaa suoraan kehitykseen kuluvaan aikaan, kun aikataulu venyy, ja kehitykseen kuluvaan rahaan, kun artefakteja joudutaan rakentamaan uudelleen. (Keith, 2010)

Keith (2010) onkin esittänyt näkemyksensä ketterän pelikehityksen vaiheista ja niiden merkityksestä aikatauluun ja budjettiin. Konseptivaiheessa etsitään piikkien avulla sopiva konsepti, joka yleensä hyväksytetään julkaisijalla ja jonka kehittämistä jatketaan esituotantovaiheessa. Esituotantovaiheessa kehitetään pelin keskeisimmät toiminnot ja niiden esittelemiseen vaadittavat artefaktit. Tämän vaiheen aikana saadaan tietoa niin pelin lopullisista toiminnoista ja hyödynnettävästä teknologiasta kuin niiden asettamista rajoituksista taideartefakteille. Esituotantovaiheessa myös kasvatetaan tietämystä eri artefaktien tuottamiseen kuluvista resursseista. (Keith, 2010) Voidaan sanoa, että esituotantovaihe on arvaamattomin, vaikeasti ennustettavin vaihe, jonka aikana pelin hauskuus etsitään.

Esituotantovaiheessa kerätyn tiedon avulla tarkennetaan arviota varsinaisen tuotantovaiheen vaatimista resursseista, erityisesti aikataulusta ja budjetista. Myös esituotantovaiheen jälkeen peli saatetaan hyväksyttää julkaisijalla. Esituotantovaiheesta siirrytään tuotantovaiheeseen sitä mukaa, kun se on luontevaa. Tuotantovaiheessa pelitoimintoihin ei saisi tulla enää merkittäviä muutoksia, jotka vaikuttaisivat jo luotuihin artefakteihin. Näin ollen tuotantovaiheen aikataulu voidaan arvioida tarkemmin. Tuotantovaiheen jälkeen peli etenee jälkituotantoon, jossa peli hiotaan julkaisukuntoon. (Keith, 2010)

Teknisillä ongelmilla Petrillo ym. (2009) viittaavat kolmannen osapuolen ohjelmistojen hyödyntämisessä havaittuihin ongelmiin. Ketterät menetelmät eivät tarjoa suoraa ratkaisua ongelmaan, mutta Keithin (2010) esittämä esituotantovaihe tarjoaa hyvän mahdollisuuden kokeilla eri teknologioita ennen lopullisen päätöksen tekemistä sen sijaan, että päätettäisiin teknologian hyödyntämisestä etukäteen ilman riittävää tietoa.

Petrillon ym. (2009) havaitsemat *ongelmat suunnitteluvaiheessa* vaikuttavat ongelmilta, jotka ilmenevät, kun yritetään etukäteen suunnitella tietämättä riittävästi suunnitteluun vaikuttavista muista seikoista. Juuri tähän Keithin (2010) esittämä ketterä pelikehitysprosessimalli pyrkii tarjoamaan vastauksen. Suun-

nitteluvaiheen ongelmia sivutaan myös dokumentaation puutteita käsittelevässä kappaleessa alempana.

Viivästykset olivat Petrillon ym. (2010) havaintojen mukaan yleensä seurausta epärealistisesta aikataulusta ja kehittämistehtäviin kuluvan ajan aliarvioimisesta. Keithin (2010) mukaan aikataulun tulisi olla neuvoteltavissa. Mikäli aikataulu ei anna periksi, voidaan toisaalta vaikuttaa projektin mittakaavaan, jolloin vähiten tärkeitä ominaisuuksia karsitaan (Keith, 2010). Kehittämistehtäviin kuluvan ajan arviot tarkentuvat mitä pidemmälle kehityksessä edetään, koska tiimin pyrähdyksessä toteuttama työmäärä tarkentuu (Schwaber & Beedle, 2001).

Sitoutuminen ennalta sovittuihin virstanpylväisiin ja aikatauluun tuottaa ongelmia erityisesti, mikäli ilmenneet muutokset eivät vaikuta millään tavalla aikatauluun saati virstanpylväisiin (Keith, 2010). Suurin ongelma hyvän pelin kehittämisen kannalta on pakonomainen sitoutuminen virstanpylväisiin, josta sekä Potanin (2010) että Keith (2010) raportoivat. Kun virstanpylvään saavuttaminen takaa pelikehityksen rahoituksen jatkuvuuden, sen saavuttamiseksi tehdään töitä, vaikka pelin menestymisen kannalta olisi muutettava suuntaa (Keith, 2010). Tämä on vahingollista pelille, jonka kehitys ei etene oikeaan suuntaan, pelistudiolle, joka ei lopulta saakaan tuotettua julkaisijalle menestyvää peliä, ja julkaisijalle, joka kantaa taloudellisen taakan pelin epäonnistumisesta.

Tobey (2008) raportoi artikkelissaan muutoksesta parempaan suuntaan ketterien menetelmien hyödyntämisen myötä. Julkaisijan osallistuminen pyrähdyn katselmointiin helpotti neuvotteluja muutosten aikaan saamiseksi ja tarjosi myös julkaisijalle mahdollisuuden ehdottaa muutoksia. Samalla luottamus pelistudioon kasvoi ja projekti nähtiin vähemmän riskialttiina. Tasaisin väliajoin julkaistavat julkaisut myös vähensivät julkaisijan kokemaa yllätystä virstanpylväiden kohdalla, mikä vähensi puolin ja toisin painetta ennalta sovitujen vaatimuksien saavuttamiseksi. Julkaisujen suunnittelu paljastaa ongelmat jo varhaisessa vaiheessa projektia ja mahdollistaa suunnan muuttamisen ennen kriittisiä ongelmia. (Tobey, 2008)

Dokumentaation puutteeseen ratkaisu ei ole kaiken dokumentointi. Keith (2010) kertoo esimerkiksi nähneensä massiivisia pelisuunnitteludokumentteja, joissa on otettu kantaa kehitettävän scifiräiskintäpelin eri aseiden lippaiden ammuskapasiteettiin. Voidaan kysyä, kuinka moni dokumenttiin kirjatuihin suunnitteluratkaisuihin säilyy muuttumattomana kehityksen loppuun saakka? Liiallisen tarkkuuden sijaan pelisuunnitteludokumentin tulisi sisältää tieto, joka on varmaa ja tunnistaa asiat, jotka vaativat tarkennusta (Keith, 2010). Pelisuunnitteludokumentin ei pitäisi yrittää vastata kaikkiin kysymyksiin toteutuksesta (Keith, 2010). Toisaalta Callele ym. (2005) väittävät, että pelialalla on puutteita vaatimusmäärittelyssä.

Miksi dokumentointi on ollut niin tärkeässä asemassa pelialalla? Keith (2010) esittää erääksi syyksi julkaisijan tarvetta tulla vakuutetuksi, että peli on toteutuskelpoinen, ja valheellisesti kuvittelee perinpohjaisen etukäteissuunnittelun ja dokumentoinnin varmistavan tämän. Syy julkaisijan varovaisuuteen on pelikehityksen suurissa kustannuksissa (Keith, 2010).

Ketterien menetelmien poikkitoiminnalliset tiimit ja ammattiyhteisöt puolestaan parantavat kommunikaatiota (Beck & Andres, 2004; Keith, 2010; Schwaber & Beedle, 2001), joka on voidaan nähdä ratkaisuna Petrillon ym. (2009) esittelemiin ongelmiin *kommunikaation* ja *ryhmäkoostumuksen* suhteen. Myöskään *kehittäjien menettämisestä* ei pitäisi koitua suurta haittaa, koska kukaan ei ole yksin ollut vastuussa mistään yksittäisestä pelin osasta. XP:n käytänteet, kuten parikoodaus ja testilähtöinen kehitys, ovat ratkaisu *ohjelmistovirheiden* esiintymistiheyteen ja *testausongelmiin* (Beck & Andres, 2004).

Työkaluongelmat johtuvat pelikehitystä huonosti tukevista yleisistä työkaluista (Blow, 2004) ja sisäisesti kehitettyjen työkalujen puutteista. Ensimmäiseen ongelmaan ketterät menetelmät eivät tarjoa vastausta, mutta jälkimmäiseen todennäköisesti auttaa työkalutiimi, joka vastaa pelistudion sisäisten työkalujen kehittämisestä (Keith, 2010).

Useat pelialan ongelmista näyttäisivät johtuvan huonosti alalle sopivien kehittämismenetelmien hyödyntämisestä. Ketterien menetelmien periaatteet ja käytänteet näyttäisivät tarjoavan ratkaisun suureen osaan havaituista ongelmista. Ketterien menetelmien hyödyntämistä pelialalla tarkastelevassa tutkimuksessa ollaan tultu samanlaiseen lopputulokseen (Petrillo & Pimenta, 2010).

4.9 Yhteenveto

Tässä luvussa esiteltiin ensin pelialalla koettuja ongelmia. Sen jälkeen kuvattiin kirjallisuudessa esitettyjä näkemyksiä ketterien menetelmien hyödyntämisestä pelikehityksessä. Eri lähteet kuvasivat ketterien menetelmien hyödyntämistä erilaisella tarkkuustasolla ja hieman erilaisesta lähtökohdasta. Esimerkiksi Keith (2010) lähestyy asiaa kokeneena pelikehittäjänä ja pelialan ketterien menetelmien pioneerina, josta syystä hän osaa kiinnittää huomiota moniin sellaisiin asioihin, joihin muissa lähteissä ei puututa. Musilin ym. (2010) lähestymistapa on akateemisempi. Molemmilla on samankaltaisia ehdotuksia liittyen ketterien menetelmien hyödyntämiseen pelikehityksessä.

Pelialan ongelmat saattavat olla seurausta huonosti innovatiiviselle alalle sopivista kehitysprosesseista. Luvun lopuksi pohdittiin sitä, miten ketterät menetelmät ja niiden käytänteet voisivat vähentää pelistudion ongelmia.

5 TUTKIMUSMENETELMÄ JA -PROSESSI

Tämän tutkimuksen empiirisen osan tavoitteena on selvittää, missä määrin ja millä tavalla suomalaisessa peliteollisuudessa hyödynnetään ketteriä menetelmiä ja onko ketterien menetelmien hyödyntämisellä vaikutusta aiemmin havaittuihin pelialaa vaivaaviin ongelmiin. Tutkimustuloksista toivotaan selviävän ketterien menetelmien suosio Suomen peliteollisuudessa ja saatavan vahvistus kirjallisuudessa esitetyille väitteille ketterien menetelmien positiivisesta vaikutuksesta peliteollisuuden ongelmiin. Tulosten toivotaan myös auttavan pelistudioita pelikehitysprosessien- ja käytänteiden parantamisessa. Seuraavissa alaluvuissa kuvataan tutkimusmenetelmä, tiedonkeruutapa ja tutkimuksen otos. Sen jälkeen kuvaillaan tutkimusasetelma ja kyselylomakkeen teko, joita seuraa kuvaus tiedon keruusta ja käsittelystä.

5.1 Tutkimusmenetelmän valinta

Tämän tutkielman tavoitteet huomioon ottaen tutkimus olisi ollut mahdollista toteuttaa tapaus- eli case-tutkimuksena tai survey-tutkimuksena. Tapaustutkimuksessa tarkastellaan yhtä tapausta tai useita tapauksia, joista kerättävä tieto voi olla sekä kvantitatiivista että kvalitatiivista. Tietoa voidaan kerätä niin kyselyllä, haastattelulla, havainnoinnilla kuin arkistomateriaalia käyttämällä. (Järvinen & Järvinen, 2004) Hirsjärven, Remeksen ja Sajavaaran (2004) mukaan tapaustutkimuksella kerätään yksityiskohtaista, intensiivistä tietoa tutkimuskohteesta. Tapaustutkimuksessa kerätyn aineiston avulla pyritään yleensä kuvailemaan ilmiötä (Hirsjärvi ym., 2004).

Survey-tutkimuksessa kerätään tietoa käyttäen otosta ja standardisoituja tiedonkeruun tekniikoita. Survey-tutkimuksessa kysytään strukturoituja, ennalta määriteltyjä kysymyksiä. (Järvinen & Järvinen, 2004) "Survey-tutkimus on halpa tapa saada nopeasti tietoa laajasta joukosta (Järvinen & Järvinen, 2004, s. 57)." Hirsjärven ym. (2004) mukaan survey-tutkimuksella kerätään tietoa tutkimuskohteesta standardoidussa muodossa. Survey-tutkimuksessa kerätyn ai-

neiston avulla pyritään kuvailemaan, vertailemaan ja selittämään ilmiöitä (Hirsjärvi ym., 2004).

Tutkimusmenetelmän valintaan vaikuttavia tekijöitä tämän tutkielman osalta ovat muun muassa tutkimuksen deskriptiivinen luonne, perusjoukon koko ja tavoitteet. Tämän tutkielman tavoitteena on saada yleistettävää tietoa ketterien menetelmien hyödyntämisestä Suomen peliteollisuudessa, jolloin tutkimus on deskriptiivistä (Pinsonneault & Kraemer, 1993). Tutkimuksen kohteena oleva perusjoukko, suomalaiset pelistudiot, on niin suuri, että kokonaisvaltaista tietoa ei voida kerätä tapaustutkimuksella. Näistä syistä johtuen tämän tutkimuksen tutkimusmenetelmäksi valittiin kvalitatiivinen ja kvantitatiivinen survey-tutkimus. Survey on vaivaton tapa kerätä tietoa suuresta ryhmästä tutkimuskohteita, joista valitaan survey -tutkimukseen vastaava otos (Hirsjärvi ym., 2004; Järvinen & Järvinen, 2004).

5.2 Tiedon keruutavan valinta

Survey-tutkimus voidaan toteuttaa kyselylomakkeen tai strukturoitujen haastattelujen avulla (Hirsjärvi ym., 2004). Edellä esitettyjen syiden vuoksi ja koska haastattelujen tekeminen kaikissa näissä pelistudioissa olisi liian työlästä, tämän tutkimuksessa tiedon kerääminen toteutettiin elektronisella kyselylomakkeella. Tämä todennäköisesti alentaa kynnystä osallistua tutkimukseen ja sitä kautta tutkimustulosten yleistettävyyttä parane.

Kyselyn etuja ovat mahdollisuus kerätä laaja tutkimusaineisto ja menetelmän tehokkuus sekä halpuus etenkin, kun kysely toteutetaan verkossa (Hirsjärvi ym., 2004). Kyselyn heikkouksia ovat Hirsjärven ym. (2004) mukaan:

- Mahdotonta tietää, miten vakavasti vastaajat suhtautuvat kyselyyn.
- Ei ole selvää, ovatko annetut vastausvaihtoehdot onnistuneita vastaajien näkökulmasta.
- Ei tiedetä, ovatko vastaajat perehtyneitä kyselyn aihealueeseen.
- Kyselylomakkeen laatiminen on vaativaa.
- Kato voi olla suuri.

Tämän survey-tutkimuksen kohteena ovat pelistudiot, joiden toimintaa ja jokapäiväistä elämää tutkimus koskettaa. Tästä syystä voidaan olettaa, että kyselyyn vastataan vakavissaan ja että vastaajat ovat enimmäkseen perehtyneitä kyselyn aihealueeseen. Tämä saattaa vähentää katoa. Vastausvaihtoehtojen ja kysymysten asettelussa on pyritty käyttämään ymmärrettävää sanastoa ja erilaiset termit esitetään myös englanniksi väärinkäsitysten välttämiseksi.

5.3 Otoksen valinta

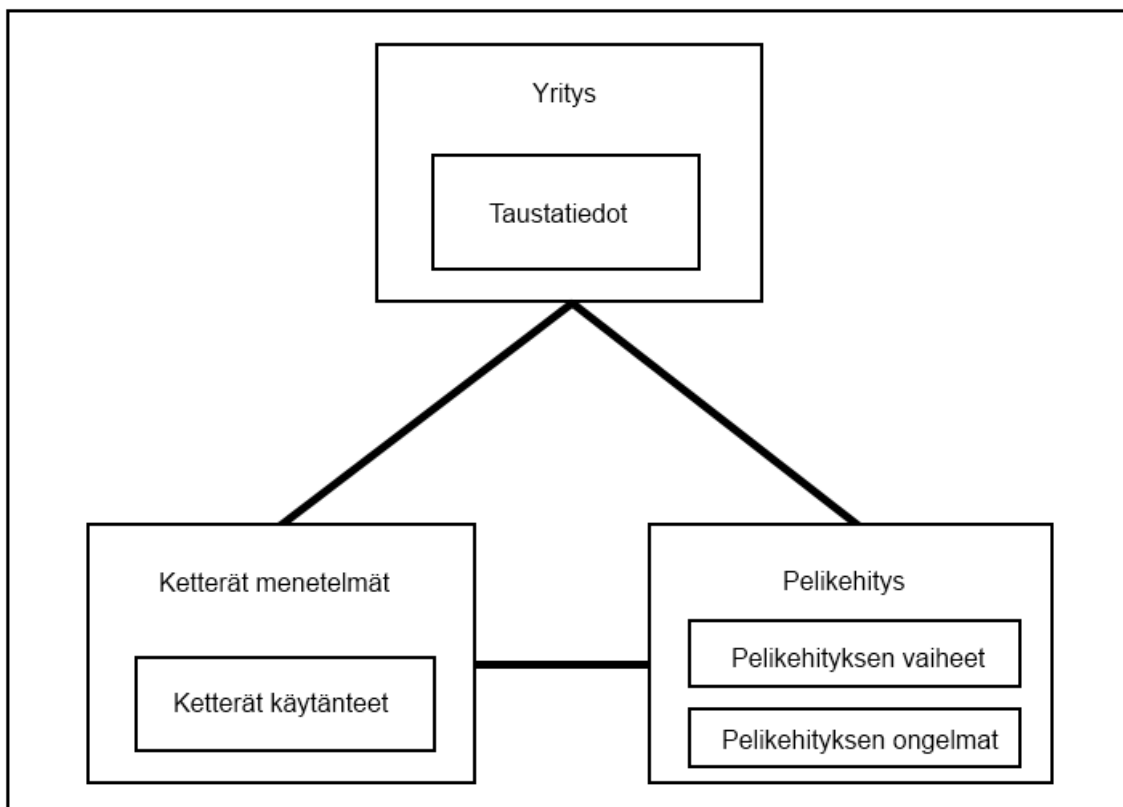
Tutkimuksen perusjoukkoon kuuluvat kaikki suomalaiset pelistudiot. Suomessa pelistudioiden koko oletettavasti vaihtelee kansainvälisessä mittakaavassa pienestä keskisuureen. Suomen suurimpia pelistudioita ovat Remedy Entertainment, joka työllistää 45 henkeä (Wikipedia, 2011c), ja Rovio Mobile, joka työllistää noin 90 henkeä (Wikipedia, 2011d). Ne ovat kuitenkin pieniä, kun niitä verrataan esimerkiksi Treyarchiin, joka työllistää kokonaisuudessaan noin 200 henkeä (Wikipedia, 2011g).

Suomalaiset pelistudiot kehittävät pelejä niin PC:lle, konsoleille kuin mobiililaitteillekin. Verkkojakelun ja verkossa toimivien kauppapaikkojen sekä konsolien omien verkkokauppojen, ja niissä myytävien pikkupelien, myötä yhä pienemmät pelistudiot voivat kehittää pelejä PC:lle ja konsoleille, joita on tavallisesti pidetty suurten studioiden alustoina. Kaiken kaikkiaan suomalaiset pelistudiot muodostavat monenkirjavan joukon. (Kuittinen ym., 2010)

Kvalitatiivisen tutkimuksen ominaispiirteitä on muun muassa kohdejoukon valinta tarkoituksen mukaisesti (Hirsjärvi ym., 2004). Jotta kohdejoukko olisi kattava otos suomalaisia pelistudioita, tutkija otti yhteyttä kahteen suomalaiseen pelialan kattojärjestöön. Suomen Pelikehittäjät ry (<http://www.pelinkehittajat.fi/>) on suomalaisten pelistudioiden kattojärjestö, jonka jäsenenä on noin kolmekymmentä suomalaista pelistudiota. Neogames (<http://www.hermia.fi/neogames/>) on Suomen pelialan kattojärjestö, jonka jäsenistöön kuuluvat niin Suomen Pelikehittäjät ry:n jäsenet kuin muut pelialalla eri tavoilla vaikuttavat yritykset. Kolmas Suomessa toimiva pelialan järjestö on FIGMA ry (<http://www.figma.fi/index.htm>), joka on digitaalisten mediataallenteiden maahantuojien voittoa tuottamaton aatteellinen yhdistys, jonka jäseniä ovat pelejä Suomeen tuovat yritykset. FIGMA ei ollut tämän tutkimuksen kannalta relevantti. Tutkimuksen otos muodostui lopulta sekä Suomen Pelikehittäjät ry:n jäsenistä että Neogamesin websivuilta löytyvän suomalaisten pelistudioiden listan (http://www.hermia.fi/neogames/alan_toimijat/) yrityksistä. Kyselyn vastaajaksi pyrittiin saamaan kustakin pelistudiosta henkilö, jonka työnkuva ja tietämys (esimerkiksi Scrum-mestari) takaavat parhaat lähtökohdat vastata kyselyyn.

5.4 Tutkimusasetelman luominen ja kyselylomakkeen teko

Tämän tutkimuksen tutkimusasetelma perustuu kirjallisuuden pohjalta luotuihin pelialaa ja pelikehitystä koskeviin rakenteisiin ja jäsenyyksiin. Asetelma koostuu viidestä osiosta: taustatiedot, pelikehityksen vaiheet, ketterät menetelmät, ketterät käytänteet ja ketterän kehittämisen vaikutukset pelikehityksen ongelmiin (Kuvio 4).



KUVIO 4 Tutkimusasetelma

Yrityksen taustatiedot -osion kysymysten tarkoituksena on selvittää pelistudion ominaisuuksia. Lähinnä tutkijaa kiinnostavat pelistudion työntekijöiden määrä, kehitettävien pelien genre ja samanaikaisesti meneillään olevien peliprojektien määrä, koska ne osaltaan vaikuttavat pelistudion tarpeeseen ja tapaan hyödyntää ketteriä menetelmiä. Näiden tietojen valossa voidaan myös tarkastella eri tavalla organisoituneiden yritysten tarvetta hyödyntää ketteriä menetelmiä, miten yrityksen koko ja kehitettävän pelin genre sekä muut taustatekijät vaikuttavat kehittämisprosessiin ja prosessissa hyödynnettävien menetelmien soveltamiseen.

Pelikehityksen vaiheet -osiossa selvitetään, mitä kehittämistehtäviä pelistudiolla kuuluu pelikehityksen eri vaiheisiin. Keith (2010) esitti kirjassaan, mitä tehtäviä ketterien menetelmien myötä pelikehityksen vaiheisiin kuuluu. Manninen ym. (2006) puolestaan on esittänyt menetelmistä riippumattomasti pelikehityksessä havaittavat vaiheet. Osion kysymysten avulla selviää, mitä vaiheisiin käytännössä kuuluu riippumatta hyödynnettävistä menetelmistä, ja sitä voidaan myöhemmin käyttää vertailukohtana eri menetelmien ja toisaalta esiintyvien ongelmien yhteydessä.

Ketterät menetelmät eri kehitysvaiheissa -osion tarkoituksena on selvittää, mitä ketteriä menetelmiä eri vaiheissa hyödynnetään. Kirjallisuudessa Musil ym. (2010) ja Keith (2010) ovat esittäneet näkemyksensä, missä eri vaiheissa tulisi mitään ketteristä menetelmistä hyödyntää. Keithin (2010) näkemyksen mukaan Scrumia käytetään kokonaisvaltaisesti konsepti- ja esituotantovaiheissa, mutta

tuotantovaiheessa on tuotantotiimien parempi siirtyä Kanbania hyödyntävään tuotantolinjaan ja jälkituotantovaiheessa Scrum tulee muokata tukemaan noin päivän mittaisia pyrähdyksiä. Musilin ym. (2010) mielestä esituotanto- ja tuotantovaiheessa hyödynnetään kokonaisvaltaisesti Scrumia. Tämän osion kysymysten avulla selviää, miten suomalaisten pelistudioiden kehitysmenettelmät vaihtelevat eri vaiheiden välillä ja noudattavatko vaihtelut kirjallisuudessa esitettyjä näkemyksiä.

Ketterät käytänteet eri kehitysvaiheissa -osio tarkentaa edellisen osion tarkastelua käytänteiden tasolle. Tämän osion kysymysten tarkoituksena on selvittää yksittäisten käytänteiden hyödyntämistä eri vaiheissa. Esimerkiksi Keith (2010) on esittänyt tuotantovaiheessa Scrumin hylkäämistä tuotantotiimien osalta ja näiden tiimien siirtymistä hyödyntämään Kanbania, koska taideartefaktien luomisprosessi poikkeaa ohjelmoinnista.

Ketterät menetelmät suhteessa pelialan ongelmiin -osio pyrkii kartoittamaan, ovatko ketterät menetelmät tarjonneet vastauksia kirjallisuudessa havaittuihin pelialaa vaivaaviin ongelmiin (Petrillo ym., 2009). Kysymykset kattavat niin projektinhallinnan ongelmat kuin pelikehityksen eri osa-alueiden ongelmat. Tämän osion tulokset ovat erityisen mielenkiintoisia, sillä ne kertovat ketterien menetelmien hyödyllisyydestä peliteollisuudessa. Pelialan ongelmia ovat käsitelleet Petrillo ym. (2009), joiden artikkeliin kysymykset perustuvat. Myös muut lähteet ovat sivunneet ongelmia, kuten Keith (2010), Potanin (2010) ja Callele ym. (2005).

Survey -tutkimusta varten luotiin sähköinen kyselylomake Jyväskylän yliopiston Korppi -järjestelmää käyttäen. Lomake koostuu tutkimusasetelman mukaisesti viidestä osiosta, joista kukin sisältää useampia kysymyksiä. Kyselylomake on esitetty liitteessä 4.

Yrityksen taustatiedot -osiossa kysytään yrityksen ikää, työntekijöiden määrää, pelien kohdealustoja, pelien genrejä, projektien keskimääräistä kestoja, projektien keskimääräistä henkilömäärää ja samanaikaisten projektien määrää. Taustatiedoista pelien kohdealustat ja pelien genret vaikuttavat muun muassa kehitystoimien haasteellisuuteen, tarvittavien taideartefaktien määrään, jotka edelleen vaikuttavat itse kehitysprosessiin. Muut kysyttävät tiedot vaikuttavat yrityksen hyödyntämiin kehitysmenettelmiin ja edelleen ketterien menetelmien soveltamiseen.

Pelikehityksen vaiheet -osio on jaettu neljään osaan perustuen kirjallisuudessa esitettyihin neljään pelikehityksen vaiheeseen: konsepti-, esituotanto-, tuotanto- ja jälkituotantovaihe. Kunkin vaiheen kohdalla esitetään kehitystehtäviä, jotka kirjallisuuden perusteella esiintyvät kyseisessä vaiheessa. Näiden lisäksi esitetään kehitystehtäviä, jotka toistuvat jokaisen osion kohdalla riippumatta siitä, onko kirjallisuudessa esitetty sitä kuuluvaksi kyseiseen osioon. Näiden kehitystehtävien osalta pyritään selvittämään, tehdäänkö joitakin kehitystehtäviä kirjallisuuden näkemyksen vastaisesti. Tämän osion kysymyksillä selvitetään, missä määrin yritysten pelikehitysprosessi vastaa kirjallisuudessa esitettyä näkemystä.

Ketterät menetelmät eri kehitysvaiheissa -osiossa vastaajat voivat kunkin kehitysvaiheen kohdalla valita, mitä ketterää tai muuta menetelmää kyseisessä kehitysvaiheessa yritys käyttää. Aiemmin mainittujen neljän vaiheen lisäksi on kohta ”Muut vaiheet”, mikäli yrityksen kehitysprosessiin kuuluu sellaisia vaiheita, joita ei kirjallisuudessa ole mainittu. Tämän osion tarkoituksena on selvittää menetelmä- ja kehitysvaihetasolla ketterien menetelmien hyödyntämistä.

Ketterien menetelmien käytänteet eri vaiheissa -osiossa kysytään yksittäisten käytänteiden hyödyntämistä eri vaiheissa, jolloin niitä voidaan verrata kirjallisuudessa esitettyihin näkemyksiin eri vaiheissa hyödynnettävistä käytänteistä. Kullakin ketterällä menetelmällä (Scrum, XP ja Kanban) on oma osionsa, joissa on oma kohta kullekin menetelmän tutkimuksen kannalta oleellisille käytänteille.

Ketterät menetelmät suhteessa pelialan ongelmiin -osiossa esitetään väittämiä, joissa ketterillä menetelmillä on jonkinlainen positiivinen vaikutus pelikehitykseen tai projektinhallintaan, ja kysytään avoimena kysymyksenä, millaisia negatiivisia vaikutuksia ketterillä menetelmillä on ollut pelikehitykseen. Tämän osion vastausten perusteella voidaan arvioida, onko ketterillä menetelmillä vaikutusta kirjallisuudessa havaittuihin pelialan ongelmiin ja onko joitakin uusia ongelmia syntynyt ketterien menetelmien hyödyntämisen seurauksena.

Kyselylomake luotiin iteratiivisen prosessin avulla. Tutkija loi ensimmäisen version kyselystä, joka käytiin läpi yhdessä ohjaajan kanssa. Läpikäynnin jälkeen tutkija muokkasi kyselyä palautteen mukaan. Tutkija valmisteli kyselyä usean iteraation verran ennen kuin neljä ulkopuolista henkilöä testasi kyselyn ja heiltä saadun palautteen mukaan kyselyä muokattiin. Testaajien palautteen perusteella kysymyksissä havaittua johdattelua pyrittiin vähentämään, joitakin sanoja ja termejä muutettiin ja erään osion rakennetta muokattiin vastaajaystävällisemmäksi. Tämän jälkeen kysely käytiin vielä kertaalleen läpi tutkijan ja ohjaajan toimesta ennen kuin lopullinen kysely lähetettiin vastaajille. Kyselylomakkeeseen liittyi liitteessä 1 oleva saatekirje.

5.5 Tiedon keruu ja käsittely

Kyselyn laatimisen rinnalla selvitettiin mahdollisia kanavia, joiden tuella kyselylle saataisiin uskottavuutta ja saavutettaisiin helposti mahdollisimman moni suomalainen pelistudio. Tutkija tiedusteli ensin pelialan kattojärjestö Neogamesin kiinnostusta kyselyn avustamiseen. Valitettavasti Neogames on linjannut, ettei se välitä kyselyjä jäsenilleen. Myöhemmin tutkija otti vielä yhteyttä Suomen Pelikehittäjät ry:hyn, joka suostui jakelemaan kyselyä jäsenilleen. Suomen Pelikehittäjät ry on kyselyn kannalta erittäin sopiva järjestö, koska sen jäsenistö koostuu pelkästään pelikehitykseen keskittyvistä yrityksistä toisin kuin Neogamesin jäsenistö.

Kysely saatteineen lähetettiin Suomen Pelikehittäjät ry:lle 7.6.2011. Kyselyn edelleen lähettäminen viivästyi, koska Suomen Pelikehittäjät ry:n puheenjohtaja oli E3-messuilla, joka on eräs pelialan keskeisimmistä messuista ja järjestettiin juuri 7. -9.6.2011. Valitettavasti tämä oli jäänyt tutkijalta huomioimatta. Kyselyyn pystyi vastaamaan 23.6.2011 asti ja Suomen Pelikehittäjät ry:n kautta yritettiin lähettää vielä 20.6.2011 muistutus vastaamisesta kaikille jäsenille. Muistutuksen välittämisestä eteenpäin ei ole varmuutta.

Edellä kuvattu ensimmäinen kyselykierron tuotti valitettavasti vain neljä vastausta. Määrä oli liian pieni. Tämän takia tutkija päätyi ohjaajan kanssa neuvoteltuaan hyödyntämään Neogamesin internetsivuilta löytyvää listaa suomalaisten pelikehitysalan yritysten kotisivuista. Tutkija kävi läpi kaikki listalla olevat kotisivut ja keräsi yritysten yhteystiedoista ensisijaisesti puhelinnumerot ja toissijaisesti sähköpostiosoitteet. Jo tässä vaiheessa kaikista 66 yrityksestä osa karsiutui pois, koska ne eivät joko olleet riittävän suuria (vähintään viiden hengen yrityksiä) tai ne eivät olleet pelikehitykseen keskittyviä yrityksiä. Listan läpikäynnin jälkeen tutkijalla oli kaiken kaikkiaan 51 yrityksen yhteystiedot.

51 yrityksestä 17 antoi kotisivuillaan ainoastaan sähköpostiosoitteen, 32 puhelinnumeron ja kaksi yritystä tarjosi yhteydenoton ainoastaan internetissä täytettävän lomakkeen kautta. Kaikille yrityksille, jotka eivät antaneet kotisivuillaan puhelinnumeroa, lähetettiin sähköisesti saatekirje (liite 3), jossa kerrottiin kyselyn tilanne, varmistettiin vastaamisen tila ja yrityksen koko. Tämän jälkeen pyydettiin vastaamaan tutkijalle sähköpostitse kyselyn kannalta sopivimman henkilön sähköpostiosoite. Sähköposteihin saatiin kolme vastausta, joista kaksi kertoi sopivan vastaajan sähköpostiosoitteen ja yksi kertoi yrityksen olevan liian pieni.

Yrityksistä, joita tavoiteltiin puhelimitse, kuutta ei tavoitettu lainkaan ja viisi osoittautui puhelinsoiton myötä liian pieniksi. Kaksi tavoitettua yritystä myös ilmoitti vastanneensa jo aiemmin ja kaksi yritystä jättäytyi tutkimuksen ulkopuolelle muihin syihin vedoten. 17 yritystä kertoi puhelimesta vastaamiseen sopivimman henkilön sähköpostiosoitteen ja aikoi vastata kyselyyn.

Sähköpostiosoitteisiin, jotka tutkija sai vastauksena yrityksiltä, lähetettiin jokaiseen erikseen lyhyen saatekirjeen (liite 3) kera linkki kyselylomakkeeseen. Tutkija päätti lähettää kyselylomakkeen saatekirjeen (liite 2) kera myös kaikkiin sähköpostiosoitteisiin, joista ei saatu vastausta, ja niihin puhelinsoittoon vastaamattomiin, jotka kertoivat myös sähköpostiosoitteen. Kaiken kaikkiaan kyselylomake lähetettiin 37 yritykseen.

Tämän toisen kyselykierron aikana kysely oli vastattavissa 15.7.2011 saakka ja kaikille 37 yritykselle lähetettiin vielä muistutus kyselyyn vastaamisesta 13.7.2011. Kyselyn vastausajan umpeuduttua kyselyyn oli kertynyt 20 vastausta, joissa kyselyyn oli vastattu ainakin osittain. Yhtään täysin tyhjää vastausta ei tullut.

Kyselystä saatu data oli tutkijan käytettävissä sähköisessä muodossa ja siitä voitiin generoida erilaisia raportteja tulosten käsittelyn tueksi. Asteikolliset kysymykset käytiin läpi kukin erikseen, ja niihin liittyvissä avoimissa kysymyksissä tulleita kommentteja hyödynnettiin tarpeen mukaan. Osa ketterien

menetelmien käytänteisiin liittyvien kysymysten tuloksista jouduttiin tarkastelemaan erityisen varovasti, sillä toteutusteknisistä syistä vastaajien oli mahdollista valita useita, ristiriitaisia vaihtoehtoja. Näiden osalta tuloksia tulkittiin siten, että "en osaa sanoa" ja "ei käytetä" vastaukset jätettiin huomiotta, mikäli samaan kohtaan oli annettu muukin vastaus.

6 TULOKSET

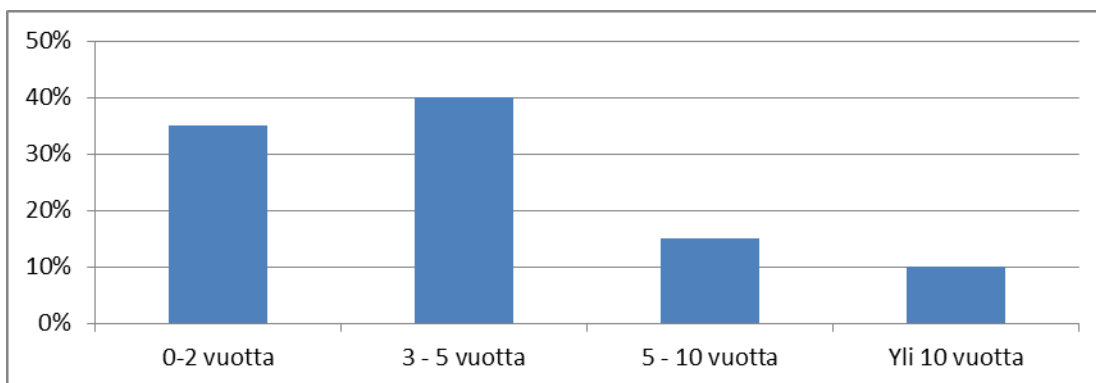
Tässä luvussa esitetään kyselytutkimuksen tulokset. Tulokset esitetään samassa järjestyksessä, jossa niitä kyselylomakkeella kysyttiin. Ensimmäinen aliluku käsittelee yrityksen taustatietoja, toinen pelikehitysprosessia, kolmas ketteriä menetelmiä, neljäs ketterien menetelmien käytänteitä ja viimeinen aliluku ketterien menetelmien positiivisia ja negatiivisia vaikutuksia pelikehitykseen.

Kyselyyn saatiin yhteensä 20 vastausta. Perusjoukko koostui suomalaisista, vähintään viiden hengen pelistudioista, joita tutkimuksen selvityksen mukaan on 45 kappaletta. Tässä joukossa saattaa olla vielä mukana liian pieniä yrityksiä, koska kaikkien yritysten kokoa ei saatu varmistettua. Näistä 37 lähetettiin kyselylomake, ja vastauksia saatiin 20. Vastausprosentti oli 54%, mitä voidaan pitää varsin hyvänä. Neljä vastaajaa oli lopettanut kesken kyselyn. Syynä saattoi olla tekninen ongelma kyselyjärjestelmässä tai kysely oli koettu liian raskaaksi normaalien työkiireiden keskellä. Keskeyttäneiden vastaajien vastauksia hyödynnetään niiltä osin kuin he vastasivat ja loput vastaukset katsotaan tyhjiksi.

6.1 Pelistudion taustatiedot

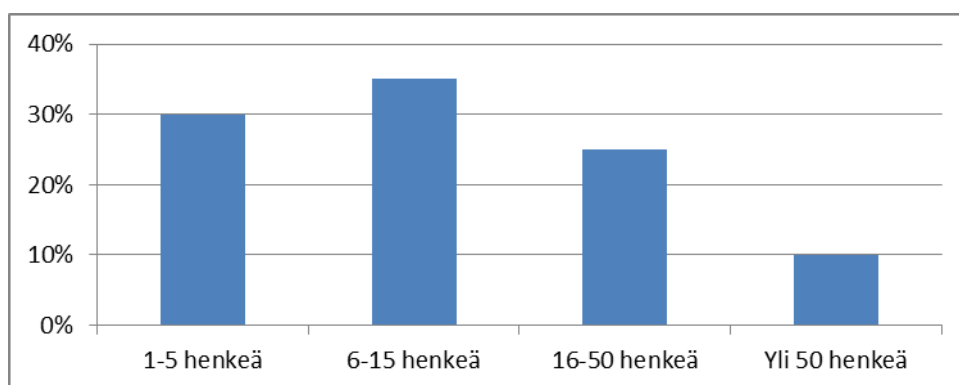
Pelistudion taustatietoina kysyttiin ikää, työntekijöiden määrää, pelien laitealustaa ja genreä sekä peliprojektien keskimääräistä kestoja, peliprojektiin osallistuvien määrää ja samanaikaisten projektien määrää.

Tutkimuksen kohteena olleet pelistudiot ovat pääosin melko nuoria (kuvio 5). Seitsemän (35%) yritystä ilmoitti iäkseen 0-2 vuotta, kahdeksan (40%) ilmoitti iäkseen 3-5 vuotta ja kolmen (15%) ikä oli 5-10 vuotta. Vain kaksi yritystä ilmoitti olevansa yli 10 vuotta vanhoja.



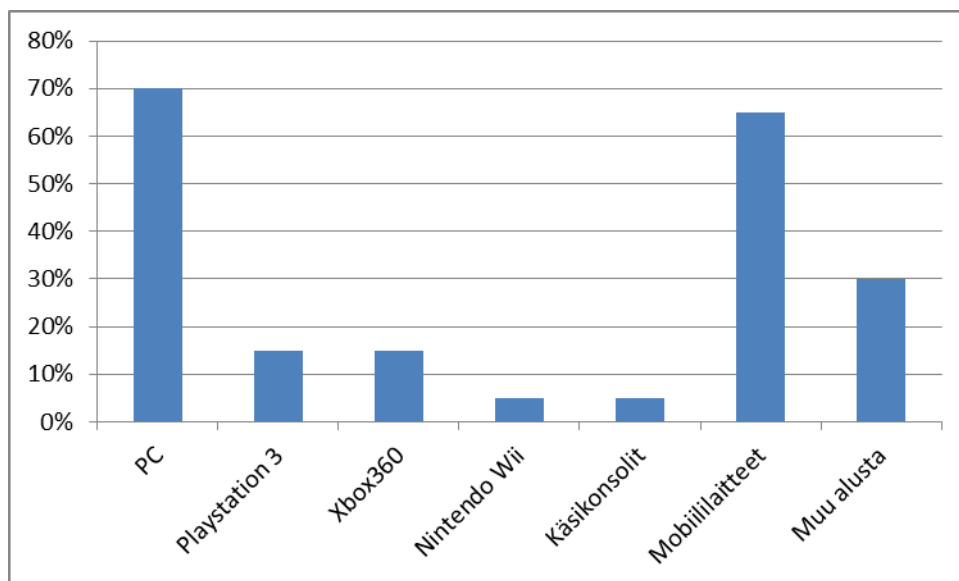
KUVIO 5 Pelistudioiden ikäjakauma (N = 20)

Yli puolet pelistudioista työllistää vähemmän kuin 16 henkeä (Kuvio 6). Kuusi (30%) yritystä ilmoitti työllistävänsä 1-5 henkilöä, mikä tässä kyselyssä tarkoittaa, että yrityksessä on työskennellyt tasan viisi henkilöä, koska tätä pienemmät eivät kuulu kyselyn kohderyhmään. Seitsemän (35%) yritystä ilmoitti työllistävänsä 6-15 henkilöä ja viisi (25%) 16-50 henkilöä. Vain kaksi (10%) yritystä työllistää yli 50 henkilöä.



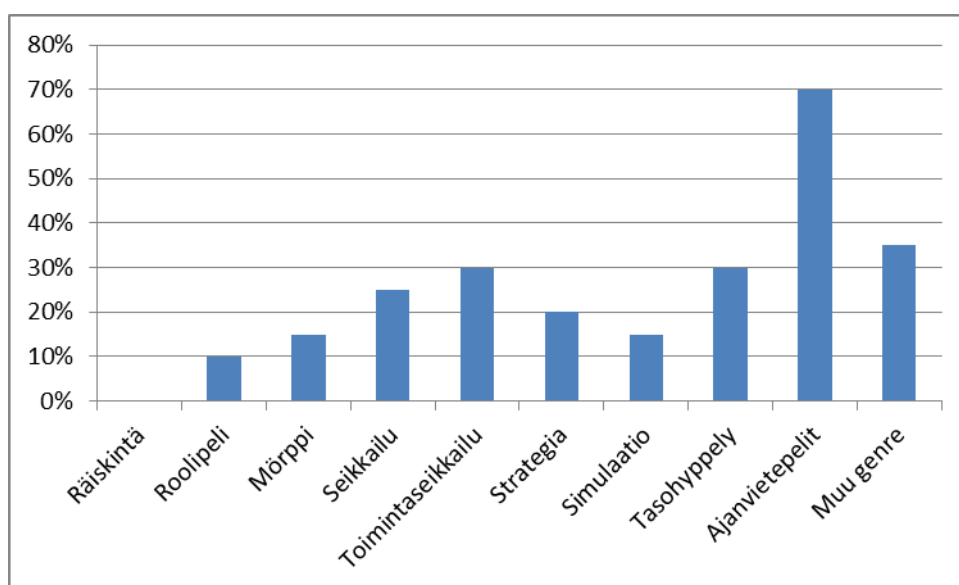
KUVIO 6 Pelistudioiden kokojakauma työntekijöiden määrän perusteella (N = 20)

Laitealustoista PC ja mobiililaitteet ovat suosituimpia (Kuvio 7). PC:lle kehittää pelejä 14 (70%) yritystä ja mobiililaitteille 13 (65%) yritystä. Playstation 3:lle kehittää kolme (15%) yritystä samoin Xbox360:lle. Nintendo Wiille ja käsikonsoleille kehittää molemmille ainoastaan yksi (5%) yritys. Muita mainittuja alustoja ovat Mac (5%), iOS (5%), Facebook (5%) ja lautapelit (15%). Yhdeksän (45%) yritystä keskittyy yhteen alustaan mutta muutoin alustojen määrä vaihtelee kahdesta kuuteen. Laitealustojen määrän keskiarvo yritystä kohden on noin kaksi.



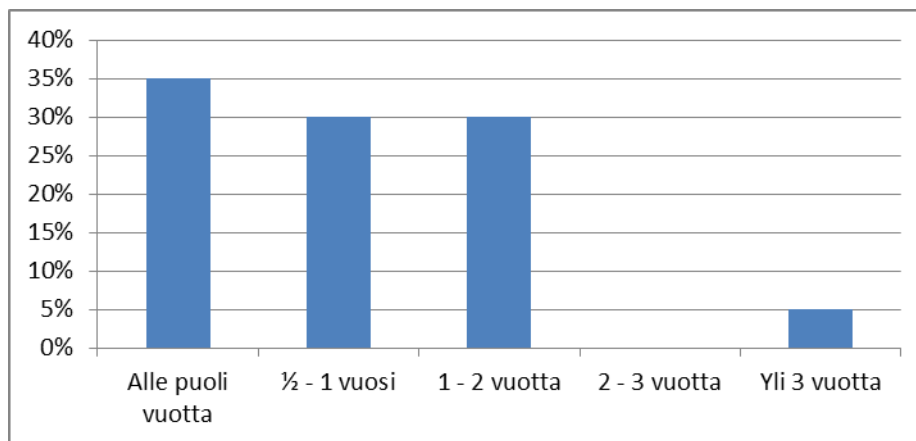
KUVIO 7 Pelistudioiden pelien laitealustat (N = 20)

Ylivoimaisesti suosituin peligenre yritysten keskuudessa on ajanvietepelit, joita 14 (70%) yritystä kertoo kehittävänsä (Kuvio 8). Seuraavaksi eniten kysymykseen on "Muu genre" vastauksia (7 kpl, 35%), joissa mainitaan seuraavat: "Massive Multiplayer Online ilma rpg elementtejä", "... tieto- ja älypelit, sekä sovellukset. Sovellukset ovat hyvin rinnastettavissa peleihin, toiminnallisesti hiukan erilaisia", "Arcade/fysiikka-pohjaiset mobiilipelit", "Musiikkipeligenre", "Hyötypelit ...", "Urheilu (racing)", ja "Musiikkipelit, management pelit ja lautapeleissä eurogame -genre". Muutoin genrejen suosio vaihtelee nollan ja kuuden välillä. Kahdeksan (40%) yritystä kehittää kolmen eri genren pelejä. Genrejen määrän keskiarvo yritystä kohden on noin 2,5.



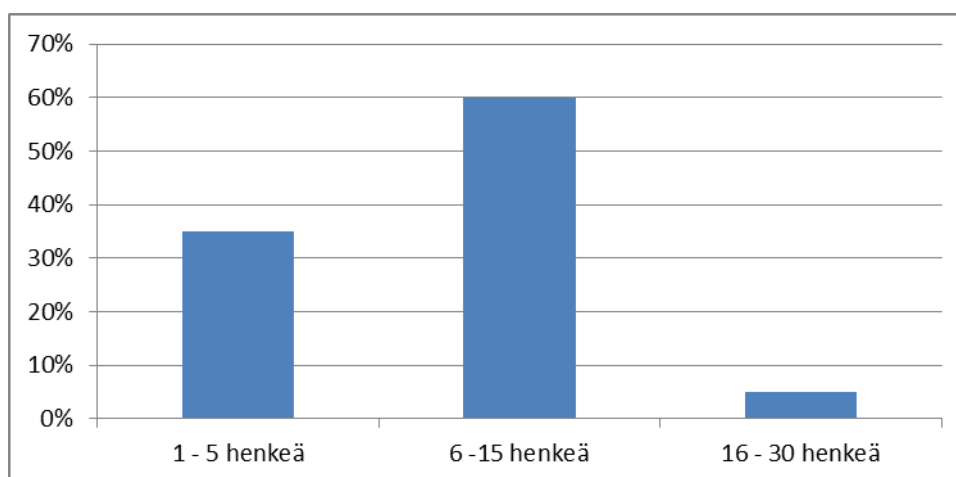
KUVIO 8 Pelistudioiden pelien genret (N = 20)

Peliprojektit kestävät yleensä alle vuoden (Kuvio 9). Seitsemän (35%) yritystä ilmoitti projektien kestävän keskimäärin alle puoli vuotta, kuuden (30%) projektit kestävät puolesta vuodesta vuoteen ja toiset kuusi (30%) ilmoitti kestoksi 1-2 vuotta. 2-3 vuoden haarukkaan ei vastannut yksikään yritys, mutta yksi yritys ilmoitti projektin keskimääräisen keston olevan yli kolme vuotta.



KUVIO 9 Pelistudioiden projektien keskimääräinen kesto (N = 20)

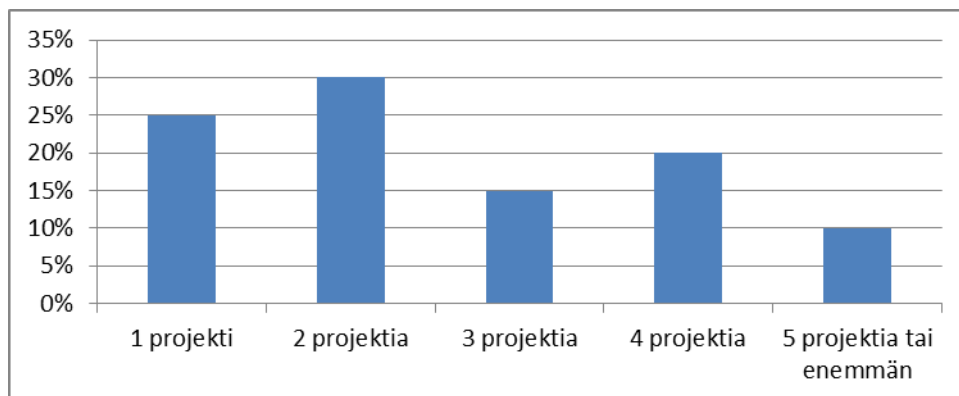
Yhteen projektiin keskimäärin osallistuvien henkilöiden määrä ei millään yrityksellä ylitä 30 henkilöä (Kuvio 10). Seitsemän (35%) yritystä ilmoittaa määräksi 1-5 henkilöä, kaksitoista (60%) 6-15 henkilöä ja yhden (5%) projekteihin osallistuu keskimäärin 16-30 henkilöä.



KUVIO 10 Yhden projektin keskimääräinen henkilöstömäärä (N = 20)

Osion viimeinen kysymys koski samanaikaisten projektien lukumäärää. Noin puolet yrityksistä keskittyy korkeintaan kahteen projektiin samanaikaisesti (Kuvio 11). Viisi (25%) yritystä tekee vain yhtä projektia kerrallaan, kuusi (30%)

kahta projektia, kolme (15%) yritystä kolmea, neljä (20%) neljää ja kaksi (10%) viittä tai useampaa projektia samanaikaisesti.



KUVIO 11 Samanaikaisten projektien määrä suomalaisissa pelistudioissa (N = 20)

Henkilöstömäärän suhde yrityksen ikään selviää seuraavasta taulukosta 6. Näyttäisi siltä, että yrityksen ikä ja henkilöstömäärä eivät ole sidoksissa keskenään. Yli 50 henkisiä yrityksiä vastasi kyselyyn vain kaksi.

TAULUKKO 6 Tietyn kokoisten yritysten keskimääräinen ikä

Yrityksen henkilöstömäärä	Eniten vastauksia kerännyt pelistudion ikähaarukka
1 - 5 henkilöä	3 - 5 vuotta
6 - 15 henkilöä	0 - 2 ja 3 - 5 vuotta
16 - 50 henkilöä	3 - 5 vuotta
Yli 50 henkilöä	0 - 2 ja 5- 10 vuotta

Tämän osion tulokset esitetään koostetusti taulukossa 7, jossa on yrityksen koon suhteen esitetty keskimääräinen laitealustojen, genrejen ja samanaikaisten projektien määrä sekä projektin henkilöstömäärän ja keston moodit. Laitealustojen ja genrejen määrä ei näytä juurikaan vaihtelevan yrityksen koon mukaan. Samanaikaisten projektien määrä näyttäisi hieman kasvavan sen mukaan, mitä suurempi yritys on henkilöstömäärältään. Yhteen projektiin osallistuvien henkilöiden määrän moodi on yritysten koosta riippuen 1- 5 tai 6 - 15 henkilöä. Projektin kesto on useimmiten yritysten koosta riippumatta 1 - 2 vuotta tai vähemmän.

TAULUKKO 7 Yhteenveto pelistudioiden taustatiedoista

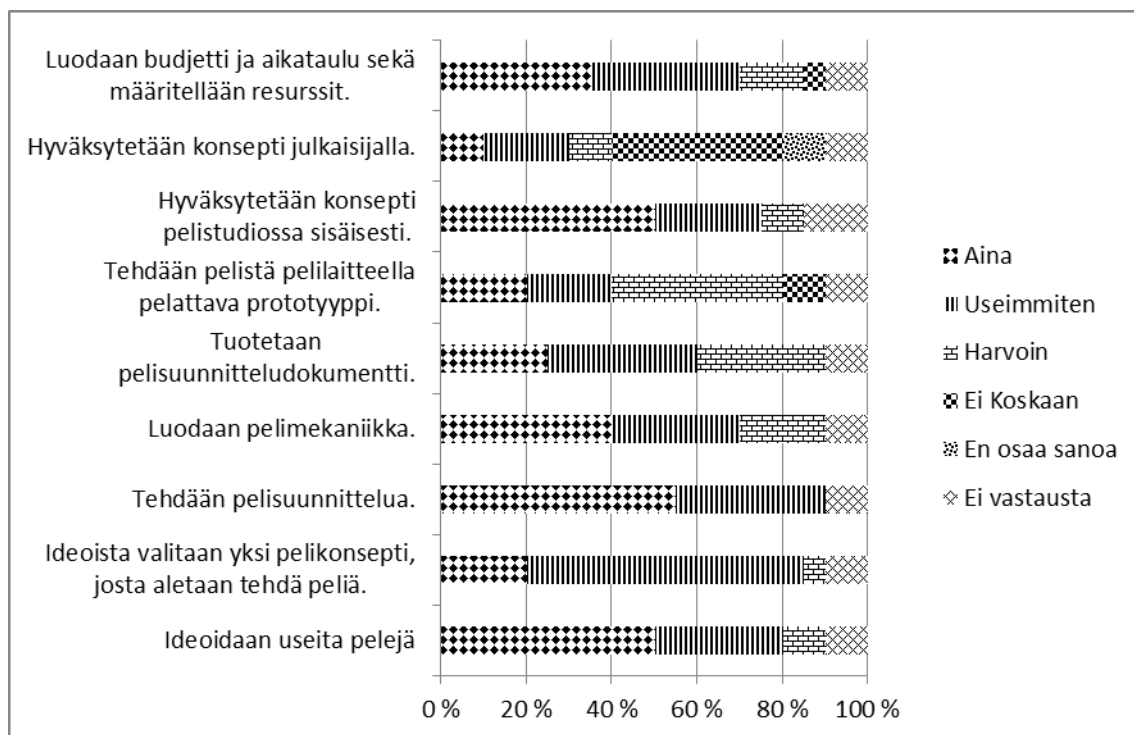
Yrityksen henkilöstömäärä	Keskimääräinen			Eniten vastauksia kerännyt	
	Laitealustojen määrä	Genrejen määrä	Samanaikaisien projektien määrä	Projektin keskimääräinen henkilöstömäärä	Projektin kesto
1 - 5 henkilöä	2	2	2	1 - 5 henkilöä	Alle puoli vuotta
6 - 15 henkilöä	1	3	2	6 - 15 henkilöä	1 - 2 vuotta
16 - 50 henkilöä	3	3	4	6 - 15 henkilöä	½ -1 ja 1 -2 vuotta
Yli 50 henkilöä	2	3	5	1 - 5 ja 6 - 15 henkilöä	Alle puoli vuotta

6.2 Yrityksen pelikehitysprosessi

Seuraavaksi tarkastellaan kysymysosiota, joka sisältää väittämiä liittyen neljään työvaiheeseen ja niiden sisältämiin tehtäviin. Jokaiseen kysymykseen liittyi usean väittämän lisäksi myös avoin kysymys, jossa tiedusteltiin kunkin vaiheen keskeisiä tehtäviä, jotka väittämissä jäivät huomioimatta. Väittämien vastauksissa käytetään asteikkoa "aina", "useimmiten", "harvoin", "ei koskaan" ja "en osaa sanoa".

Konseptivaiheeseen liittyviin väittämiin vastasi 18 vastaajaa kahdestakymmenestä, paitsi yhteen väittämään, johon saatiin vain 17 vastausta. Tähän osioon mennessä kaksi vastaajaa siis keskeytti kyselyyn vastaamisen, joten vastausten prosenttimäärät eivät summaudu tästä eteenpäin 100%:iin.

Konseptivaiheen yhteydessä nimetyistä tehtävistä suoritetaan yli puolessa yrityksistä kaikkia tehtäviä, kahta tehtävää lukuun ottamatta, aina tai useimmiten (kuvio 12). Puolessa yrityksistä hyväksytetään konsepti julkaisijalla ja tehdään pelistä pelilaitteella pelattava prototyyppi harvoin tai ei koskaan. Tutkittaessa tarkemmin prototyypin tekemistä voidaan havaita, että prototyypin tekeminen on sitä todennäköisempää, mitä pidempiä yrityksen projektit keskimäärin ovat (taulukko 8). Projektien henkilöstömäärä ei näyttäisi vaikuttavan prototyypin valmistamiseen (taulukko 9).



KUVIO 12 Konseptivaiheen tehtävien suorittaminen

TAULUKKO 8 Projektin keskimääräinen kesto suhteessa prototyyppien valmistukseen

Projektin keskimääräinen pituus	Pelistä tehdään pelattava prototyyppi konseptivaiheessa			
	Aina	Useimmiten	Harvoin	Ei koskaan
Alle puoli vuotta	0	1	3	2
½ - 1 vuotta	1	2	2	0
1 - 2 vuotta	2	1	3	0
Yli 3 vuotta	1	0	0	0

TAULUKKO 9 Projektin keskimääräinen henkilöstömäärä suhteessa prototyyppien valmistukseen

Projektin henkilöstömäärä	Pelistä tehdään pelattava prototyyppi konseptivaiheessa			
	Aina	Useimmiten	Harvoin	Ei koskaan
1 - 5 henkilöä	1	2	1	2
6 - 15 henkilöä	3	2	6	0
16 - 30 henkilöä	0	0	1	0

Konseptivaiheeseen liittyvään avoimeen kysymykseen annettiin neljä vastausta, joista yhdessä esiteltiin väittämien ulkopuolelle jäänyt konseptivaiheen tehtävä: "erillaiset feasibility studyt". Toinen vastaaja ilmoitti, etteivät konseptivaihetta koskevat väittämät sovellu heidän yrityksensä prosessin selvittämiseen. Lisäksi kaksi vastausta tarkoittaa konseptivaiheen keskeisiä tavoitteita seuraavalla tavalla:

"Tärkein tehtävä on löytää konseptin X, eli ydinjuttu, joka kuvaa peli-idean perusolemusta. Lisäksi on olennaista irroitella erilaisilla ideoilla ja kerätä runsaasti vaihtoehtoisia tapoja viedä ideaa eteenpäin. Myös konseptin havainnollistaminen kaikin mahdollisin keinoin on oleellista."

"Konseptivaiheessa määritellään pelin idea, oleellisimmat featurit ja ne dokumentoidaan. Määritellään budjetti, aikataulu ja käytettävät resurssit ylätasolla."

Esituotantovaiheeseen liittyviin väittämiin vastasi 18 vastaajaa kahdestakymmenestä, paitsi yhteen väittämään "Tehdään pelistä pelilaitteella pelattava prototyyppi", johon saatiin vain 17 vastausta.

Esituotantovaiheeseen liittyvistä tehtävistä vähintään 60% yrityksistä suorittaa kaikkia nimettyjä tehtäviä aina tai useimmiten (kuviokuva 13). Hieman alle kolmannes rakentaa itse kehitystyökaluja harvoin tai ei koskaan. Neljännes yrityksistä tekee tässä vaiheessa pelitestausta ja tarkentaa budjettia, aikataulua ja tarvittavia resursseja tuotantolinjan suorituskyvyn perusteella harvoin tai ei koskaan.

Lähes kolme neljästä yrityksestä rakentaa taiteen tuotantolinjan esituotantovaiheessa aina tai useimmiten. Tähän väittämään liitettyyn avoimeen kysymyksen annettussa vastauksessa on huomautettu: "Esimerkiksi taiteen tuotantolinja on jo valmiina eikä sitä tarvitse erikseen joka kerta luoda."

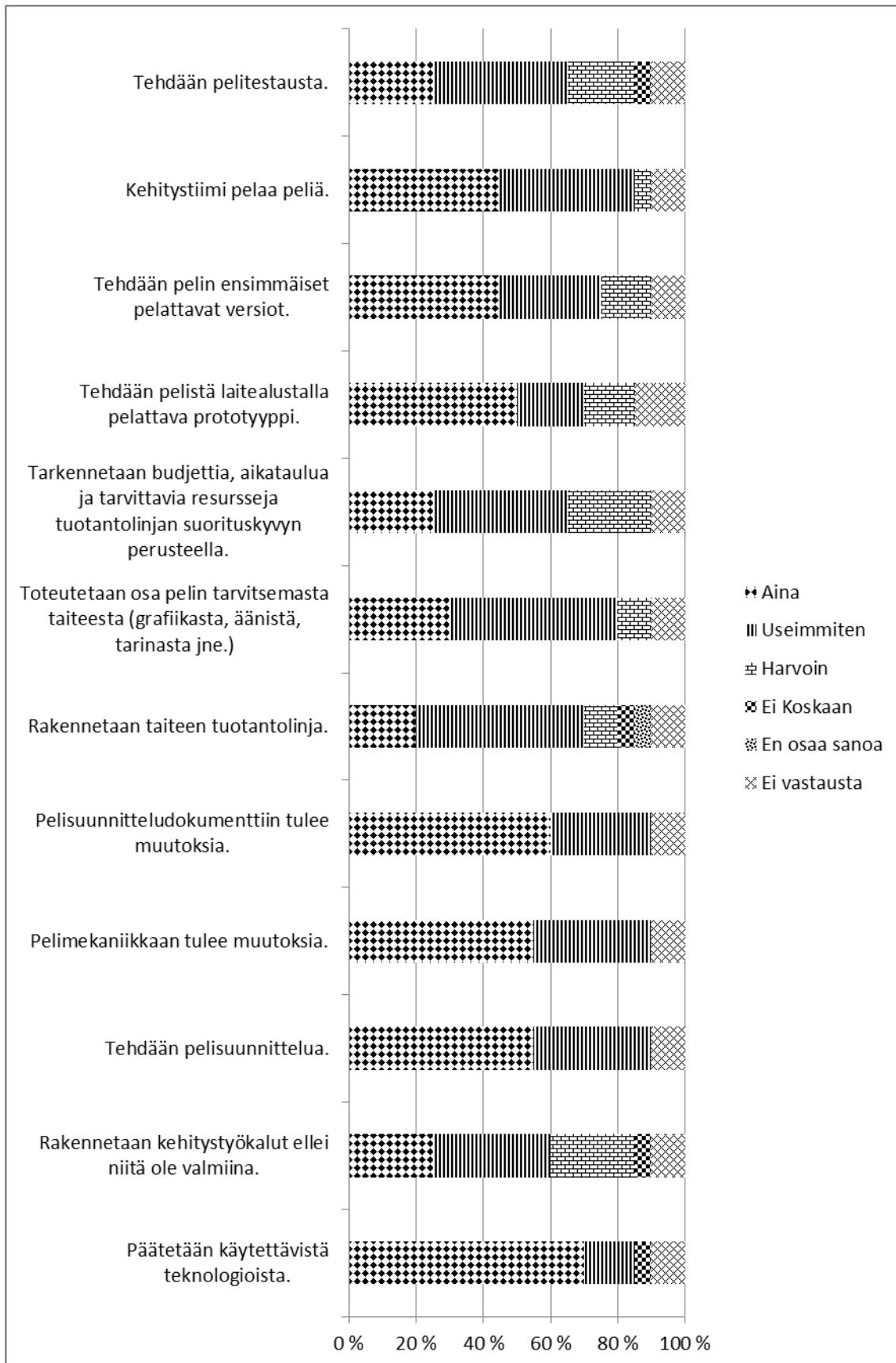
Reilu 60% yrityksistä tekee pelitestausta esituotantovaiheessa aina tai useimmiten. Eräs avoimen kysymyksen vastaus tarkentaa esituotantovaiheessa tapahtuvaa pelitestausta: "Vaikka peli ei ole valmis varsinaiseen pelitestaukseen, on tärkeää silti "testata" peliä käyttäjillä. Konseptitasolla voidaan kysellä ihmisiltä, että pelaisitko sinä peliä jossa... jne."

Muista avoimeen kysymykseen saaduista vastauksista kaksi on yleisesti konseptivaihetta luonnehtivia ja yksi vastaus ruotii koko pelikehitysprosessin onnistumiseen vaikuttavia tekijöitä.

"Esituotantovaiheen keskeisin tavoite on löytää, tunnistaa ja testata pelin ydimekaniikka. Lisäksi esituotantovaiheessa viimeistään olisi varmistettava projektin rahoitus, jotta varsinainen tuotanto saadaan rullaamaan täydellä kapasiteetilla."

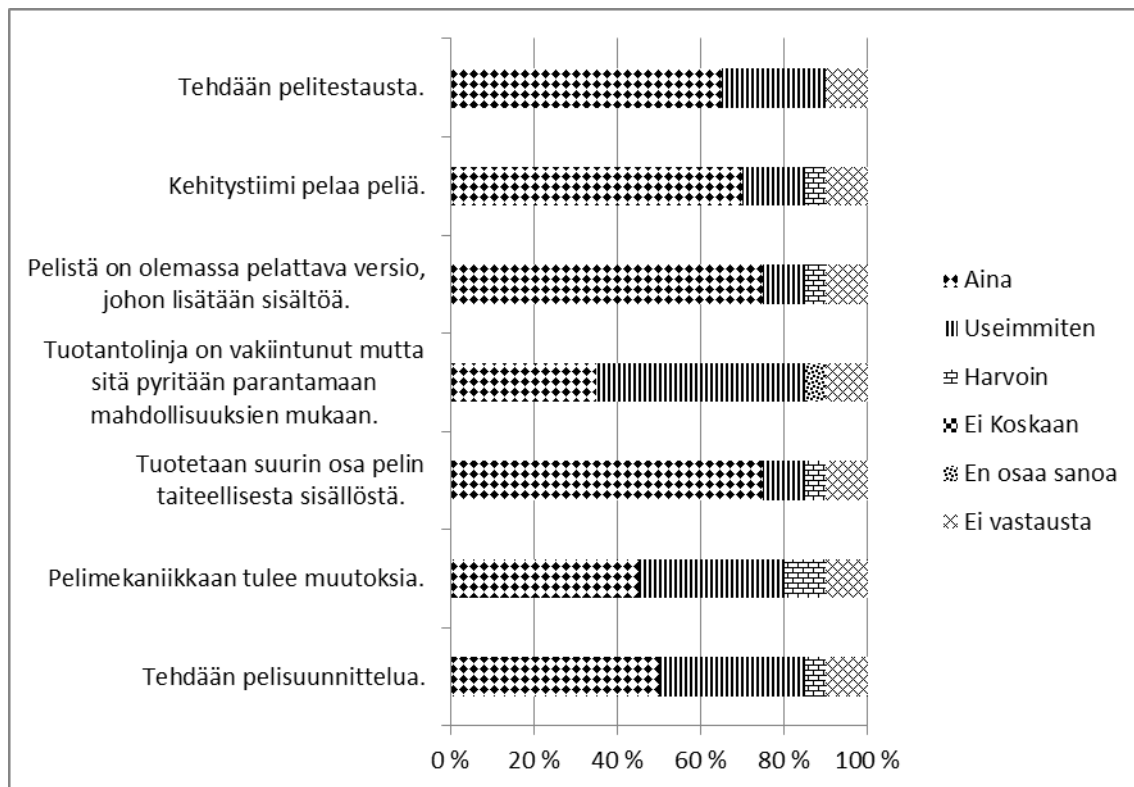
"Konseptivaiheessa saatuja tuloksia kehitetään ja tarkennetaan."

"Peliprojektien keskeinen tekijä on aina raha ja rahoitus. Jos tämä ei ole turvattu, niin hyvätkin pelit ja ideat jäävät vajavaisiksi. Koko tuotantoprosessin ydin ainakin meidän yrityksessä on peli-idea ja sen mahdollisuudet sekä mikä on kilpailutilanne markkinoilla. Tuotantoprosessi on toimiva silloin kun koko peliprojektin osaaminen on prokettiryhmän hallinnassa. Mikäli käytetään ali-hankkijoita tilanteeseen syntyy haasteita, jotka eivät aina ole mieluisia ja liiketoimintaa heikentäviä. Katson innavasti tuottamisen olevan erityistä lahjakkuutta olevan henkilökunnan panosta. Työ tehdään kuitenkin tiimissä ja jokainen peliprojektissa antaa oman panoksen."



KUVIO 13 Esituotantovaiheen tehtävien suorittaminen

Tuotantovaiheeseen liittyviin väittämiin vastasi 18 vastaajaa kahdestakymmenestä. Tuotantovaiheeseen liittyviä tehtäviä suoritetaan 80% yrityksistä aina tai useimmiten (kuvio 14).



KUVIO 14 Tuotantovaiheen tehtävien suorittaminen

Avoimen kysymyksen vastauksista kaksi liittyi pelimekaniikkaan tuleviin muutoksiin ja niihin reagoimiseen:

"Tuotantovaiheessa [...] tärkeintä on taloudellinen turva. Mikäli taustat ovat kunnossa, tuotanto on helppoa toteuttaa ja noudattaa aikatauluja. Innostunut tiimi sekä hyvä projektijohto on avain onnistuneisiin projekteihin. Tuotantovaiheen muutokset tulee aina sallia peliprojekteissa, jopa tehdä radikaali muutoksiakin."

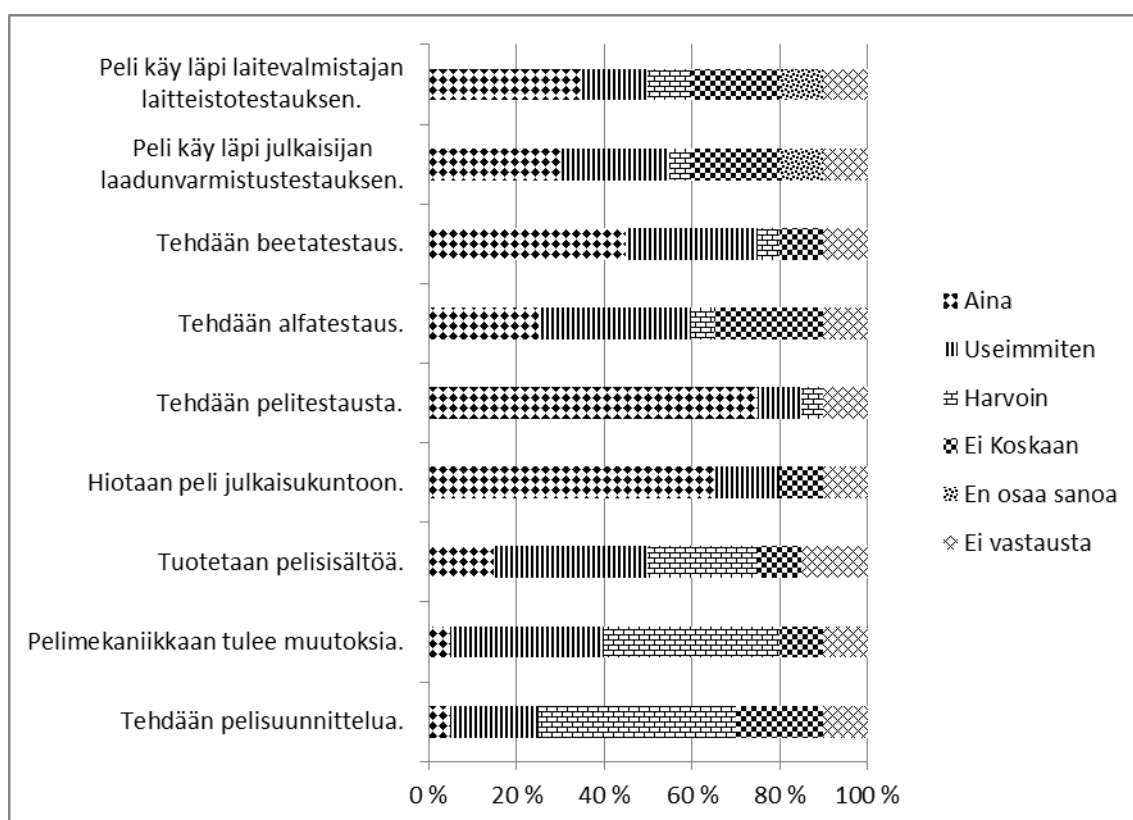
"Tuotantovaiheessa on aina kiire, joten hommien tulee pysyä aikataulussa. Projektihallinto ja projektin johtaminen ovat kriittisiä tehtäviä. Pelimekaniikka on usein vielä tuotantovaiheessakin jalostumatta, joten sen kehittämiseen tulee varata riittävästi resursseja."

Kolmas vastaus koski yleisesti tuotantovaihetta.

"Keskeisin tehtävä on tuottaa valmis peli suunnitelman mukaisesti. Suunnitelmasta poiketaan vain erittäin perustellusta syystä, jotta aikataulu ei kärsisi."

Jälkituotantovaiheeseen liittyviin väittämiin vastasi 18 vastaajaa kahdestakymmenestä. Vähintään puolet yrityksistä suorittaa jälkituotantovaiheen tehtäviä

kahta tehtävää lukuun ottamatta aina tai useimmiten (kuvio 15). Tehtävät, joita alle puolet yrityksistä suorittaa aina tai useimmiten, ovat pelimekaniikan muuttaminen ja pelisuunnittelun tekeminen, joista molempien kohdalla voidaan kirjallisuuteen tukeutuen väittää, että näitä tehtäviä ei tulisi enää jälkituotantovaiheessa esiintyä (Keith, 2010). Näiden lisäksi puolet yrityksistä tuottaa pelisisältöä jälkituotantovaiheessa aina tai useimmiten, mikä ei myöskään ole aivan se, mitä tutkija odotti tämän väittämän kohdalla. Osittain syy voi olla väittämän monimerkityksellisyydestä.



KUVIO 15 Jälkituotantovaiheen tehtävien suorittaminen

Pelisisällön tuottamiseen ja pelin julkaisukuntoon hiomiseen liittyen avoimen kysymyksen vastauksissa todetaan: "Enää vain pelin hiomista, ei uutta sisältöä."

Pelitestaukseen liittyen eräässä avoimen kysymyksen vastauksessa tarkennetaan, että: "Pelitestaaminen on valitun alustan osalta tehtävä aina eri versioiden laitteilla. Mikäli alustoja on runsaasti työ pitää tehdä jokaisella erikseen. Testaamista on turha vähätellä."

Jälkituotantovaiheeseen liittyvään avoimeen kysymykseen kertyi kolme vastausta, joista viimeinen luonnehtii jälkituotantovaiheen tarkoitusta: "Projektin huolellinen paketointi ja jokaisen yksityiskohdan viimeistelyn varmistaminen. Laadunvarmistus yleensäkin kattaa koko tuotannon."

6.3 Yrityksen käyttämät ketterät menetelmät

Tässä alaluvussa tarkastellaan vastauksia, joita annettiin kysymykseen, mitä menetelmää yritykset käyttävät kussakin pelikehityksen vaiheessa (taulukko 10). Kyselyssä pelikehitys jaettiin neljään vaiheeseen, jotka ovat konsepti-, esituotanto-, tuotanto- ja jälkituotantovaihe. Näiden lisäksi on "Muut vaiheet" -kohta. Jokaisessa vaiheessa vastausvaihtoehdoiksi esitettiin Scrum, XP, Lean, Kanban ja muu menetelmä. Osioon jätti vastaamatta kolme yritystä.

TAULUKKO 10 Eri vaiheissa käytettävät ketterät menetelmät (N = 17)

	Scrum	XP	Lean	Kanban	Muu menetelmä
Konseptivaihe	7	0	2	1	5
Esituotantovaihe	11	0	2	0	4
Tuotantovaihe	12	0	2	0	3
Jälkituotantovaihe	10	0	1	0	5
Muut vaiheet	2	0	0	0	4

Scrum on suosituin menetelmä kaikissa vaiheissa lukuun ottamatta "Muuta vaihetta", jossa "Muu menetelmä" on suosituimpi. Esituotanto-, tuotanto- ja jälkituotantovaiheissa yli puolet yrityksistä hyödyntää Scrumia. Konseptivaiheen osalta reilu kolmannes yrityksistä vannoo Scrumin nimeen. Toiseksi suosituin menetelmä on "muu menetelmä", jota hyödyntää noin neljännes yrityksistä vaiheesta riippumatta.

Leania hyödyntää noin yksi kymmenestä yrityksestä kaikissa vaiheissa lukuun ottamatta jälkituotantovaihetta. Kanbania hyödyntää yksi yritys ja sekin vain konseptivaiheessa.

XP:tä ei vastausten mukaan hyödynnetä missään yrityksessä missään vaiheessa. Todennäköisesti tässä kohtaa vastaajat eivät joko ole tunnistanee XP:n käytänteitä tai eivät ole ajatelleet hyödyntävänsä XP:tä prosessimallina. Tämän lisäksi on pieni mahdollisuus, että lyhenteen XP käyttäminen Extreme Programmingin sijaan on aiheuttanut sekannusta. Myöhemmin esiteltävät tulokset osoittavat, että XP:n käytänteitä hyödynnetään runsaasti pelikehityksessä. Usein ketteriä menetelmiä hyödynnettäessä ei oteta käyttöön vain yhtä menetelmää vaan valitaan parhaat tai sopivimmat osat eri menetelmistä. On mahdollista, että muu menetelmä vastauksissa on tällaisia hybridejä tai omiksi miellettyjä menetelmiä, joissa on kuitenkin pääosin aineksia olemassa olevista menetelmistä.

Tuloksista käy ilmi, että yritysten lukumäärään suhteutettuna nuoret yritykset käyttävät vähiten ketteriä menetelmiä (taulukko 11). Tämä saattaa johtua kehitysprosessin kypsyttömyydestä tai vähäisestä tarpeesta muodolliselle prosessille, jos myös henkilöstömäärä on alhainen. On tosin huomattava, että kyselyyn kokonaan vastanneista yrityksistä yksi ilmoitti eksplisiittisesti, ettei

käytä ketteriä menetelmiä, ja tällä yrityksellä oli kuitenkin 6 – 15 henkilöä töissä. Tätä yritystä lukuun ottamatta kaikilla vastanneilla yrityksillä oli hyödynnetty vähintäänkin muutamia käytänteitä jostakin ketterästä menetelmästä. Tämä viittaisi räätälöityyn tai itse kehitettyyn menetelmään, johon on otettu käytänteitä ketteristä menetelmistä.

TAULUKKO 11 Ketterien menetelmien hyödyntäminen eri vaiheissa suhteessa ikään (N = 20)

Yrityksen ikä	Vastan- neiden yritysten lukumää- rä	Yritys hyödyntää ketteriä menetelmiä				Oma, ketteriin mene- telmiin perustu- va mene- telmä
		Konsepti- vaihe	Esituotan- tovaihe	Tuotanto- vaihe	Jälkituotan- tovaihe	
0 - 2 vuotta	6	0	3	3	3	2
3 - 5 vuotta	7	5	6	7	5	0
5 - 10 vuotta	2	2	2	2	2	0
Yli 10 vuotta	2	1	1	2	1	1

"Muut vaiheet" ja "Muu menetelmä" vaihtoehtoja selittävään avoimeen kysymykseen on kolme yritystä selittänyt hyödyntävänsä räätälöityä tai muokattua Scrumia tai muuta ketterää menetelmää. Yksi yritys toteaa: "Ei hyödynnetä ketteriä menetelmiä." Ja toinen yritys kertoo: "Konseptivaiheen menetelmät tapauskohtaisia. Joskus scrum, joskus yhden miehen innovointia."

6.4 Yrityksen hyödyntämät käytänteet

Tämän osion kysymykset koskivat yksittäisten ketterien käytänteiden hyödyntämistä pelikehityksen eri vaiheissa. Osion ensimmäinen kysymys käsitteli Scrumin käytänteitä, toinen XP:n käytänteitä ja viimeinen kysymys käsitteli Kanbanin kolmea käytännettä. Kysymyksissä oli vastausvaihtoehtoina "Kaikissa vaiheissa", "Konseptivaihe", "Esituotantovaihe", "Tuotantovaihe", "Jälkituotantovaihe", "Ei käytetä" ja "En osaa sanoa". Scrumin käytänteitä käsittelevään kysymykseen jätti kokonaan vastaamatta neljä yritystä, ja sekä XP että Kanban kysymykset jättivät väliin kahdeksan yritystä. Useimmissa tapauksissa nämä osiot jätettiin väliin siksi, että yrityksessä ei hyödynnetty tai tiedetty hyödynnettävän mitään käytännettä näistä menetelmistä. Tämän osion vastausten prosentuaalisista jakaumista saadaan summaksi 100%, kun huomioidaan kokonaan vastaamatta jättäneet 20% tai 40%, "Ei käytetä" vastauksen antaneet ja "En osaa

sanoa" vastauksen antaneet. Loput ovat vastanneet joko "Kaikissa vaiheissa" tai jonkinlaisen kombinaation eri vaiheista. Kysymyslomakkeen teknisen toteutuksen takia vastaajien oli mahdollista vastata sekä jokin vaihe että kieltävä vastaus vastaukseksi yhteen kohtaan. Tällaiset ristiriitaiset valinnat on eliminoitu käytämällä positiivista vastausta enemmän kuin negatiivista.

Scrum ja sen käytänteet olivat ylivoimaisesti suosituimpia tämän tutkielman käsittelemästä ketterien menetelmien joukosta (taulukko 12).

TAULUKKO 12 Scrumin käytänteiden hyödyntäminen pelikehityksen vaiheissa (N = 16)

	Kaikissa vaiheissa	Konseptivaiheessa	Esituvantovaiheessa	Tuotantovaiheessa	Jälkituvantovaiheessa	Ei käytetä	En osaa sanoa
Päivittäinen Scrum tapaaminen	5	0	3	4	3	6	0
Pyrähdys	5	2	7	7	4	4	0
Pyrähdysten suunnittelu 1	2	3	6	7	3	6	0
Pyrähdysten suunnittelu 2	1	3	5	7	3	6	0
Pyrähdysten katselmointi	3	1	4	6	4	6	1
Pyrähdysten retrospektiivi	1	1	4	5	2	7	1
Pyrähdysten edistymiskäyrä	0	1	3	6	2	7	1
Tuotteen työlista	4	2	7	8	5	4	0
Pyrähdysten työlista	4	1	7	8	5	4	0

Tästä huolimatta yksi viidestä yrityksestä ilmoitti kaikkien käytänteiden kohdalla, että sitä ei käytetä. Yksikään käytänte ei ollut kaikkien vastaajien keskuudessa käytössä "Kaikissa vaiheissa". Kaikkein eniten "Kaikissa vaiheissa" hyödynnetään päivittäistä Scrum-tapaamista ja pyrähdyksiä. Lisäksi useimpia

Scrum-käytänteitä hyödynnetään mieluummin esituotanto- ja tuotantovaiheessa kuin konsepti- ja jälkituotantovaiheessa.

Scrumin käytänteisiin liittyvään avoimeen kysymykseen "Jos käytänteitä räätälöidään eri vaiheisiin, kerro siitä lyhyesti." vastasi yksi yritys:

"Koska 'projektimme' ovat keskimäärin lyhyitä parannuksia olemassa olevaan virtuaalimaailmaan, ovat lähestulkoon kaikki vaiheet yleensä niputettu yhteen. Ts. 1 sprint keskimäärin tuottaa jotain joka on näkyvässä loppukäyttäjille. Vastauksen eri vaiheiden käyttämiin tekniikoihin sen mukaan miten ne keskimäärin jakautuvat isommissa projekteissa."

XP:n käytänteistä kysyttiin vain yleisimmistä ja pelikehitykseen kirjallisuuden perusteella hyvin istuvista käytänteistä. Vaikka edellisen osion tuloksien perusteella XP:tä ei näyttänyt käytettävän koskaan, ovat XP:n käytänteet yleisesti käytössä. Syynä voi olla joko XP -käytänteiden heikko tunnistaminen tai näkemys XP:stä enemmän kokoelmana käytänteitä kuin varsinaisena menetelmänä, jota voitaisiin hyödyntää suoraan prosessimallina. Vastaukset on koottu taulukkoon 13.

TAULUKKO 13 XP-käytänteiden hyödyntäminen pelikehityksen vaiheissa (N= 16)

	Kaikissa vaiheissa	Konseptivaiheessa	Esituotantovaiheessa	Tuotantovaiheessa	Jälkituotantovaiheessa	Ei käytetty	En osaa sanoa
Yhdessä istuminen	4	3	2	1	1	6	0
Poikki-toiminnallinen tiimi	5	2	3	3	2	4	0
Informatiivinen työtila	4	2	3	3	2	4	2
Tarinat	2	1	2	2	0	7	1
Pariohjelmointi	1	1	2	1	1	7	2
Vuosineljännes-sykli	0	0	0	1	0	11	1
10 minuutin ohjelmakooste	0	0	0	0	1	9	2
Jatkuva integrointi	3	0	4	4	4	4	0
Testilähtöinen ohjelmistokehitys	2	0	4	1	0	6	3

XP -käytänteistä suosituimpia ovat poikkitoiminnalliset tiimit, informatiivinen työtila ja jatkuva integrointi, sillä neljä viidestä yrityksestä hyödyntää käytänteitä ainakin jossain pelikehityksen vaiheessa. Informatiivisen työtilan hyödyntämisestä yksi kymmenestä yrityksestä oli epävarma. Vähiten suosittuja käytänteitä ovat vuosineljännessykli ja 10 minuutin ohjelmakooste. Syy voi osittain olla käytänteiden heikossa tunnistamisessa, mihin viittaisi jatkuvan integroinnin suosio, sillä jatkuva integrointi ja 10 minuutin ohjelmakooste voidaan nähdä toisiaan täydentävinä käytänteinä. Pariohjelmoinnin ja testilähtöisen ohjelmistokehityksen alhainen suosio - 35% ja 30% "Ei käytetä" - yllätti tutkijan, koska ne ovat XP:ssä ensisijaisia käytänteitä (Beck & Andres, 2004).

Virheellisiä vastauksia koskien yhdessä istumista on yksi kappale niin konseptivaiheesta, esituotantovaiheesta, tuotantovaiheesta ja jälkituotantovaiheesta, koska vastaaja oli vastannut myös kaikissa vaiheissa. Sama virhe toistuu kahdessa seuraavassa kohdassa (poikkitoiminnallinen tiimi ja informatiivinen työtila). Lisäksi poikkitoiminnallisia tiimejä koskevaan kohtaan on jättänyt aiemmin mainitun kahdeksan sijaan vastaamatta yhdeksän.

XP:n käytänteiden räätälöintiä koskevaan avoimeen kysymykseen ei vastannut yksikään yritys. Kanbanin käytänteitä käytetään hyvin vähän peliteollisuudessa. Meneillään olevan työn rajoittamista hyödyntää kaksi yritystä kuudestatoista kaikissa kehitysvaiheissa. Työn visualisointia hyödyntää yksi yritys kuudestatoista kaikissa kehityksen vaiheissa ja yksi kuudestatoista hyödyntää visualisointia konseptivaiheessa. Läpikulkuajan mittaamista ei ilmoita hyödyntävän yksikään kyselyyn osallistuneista yrityksistä.

Avoimeen kysymykseen eräs yritys on kirjoittanut:

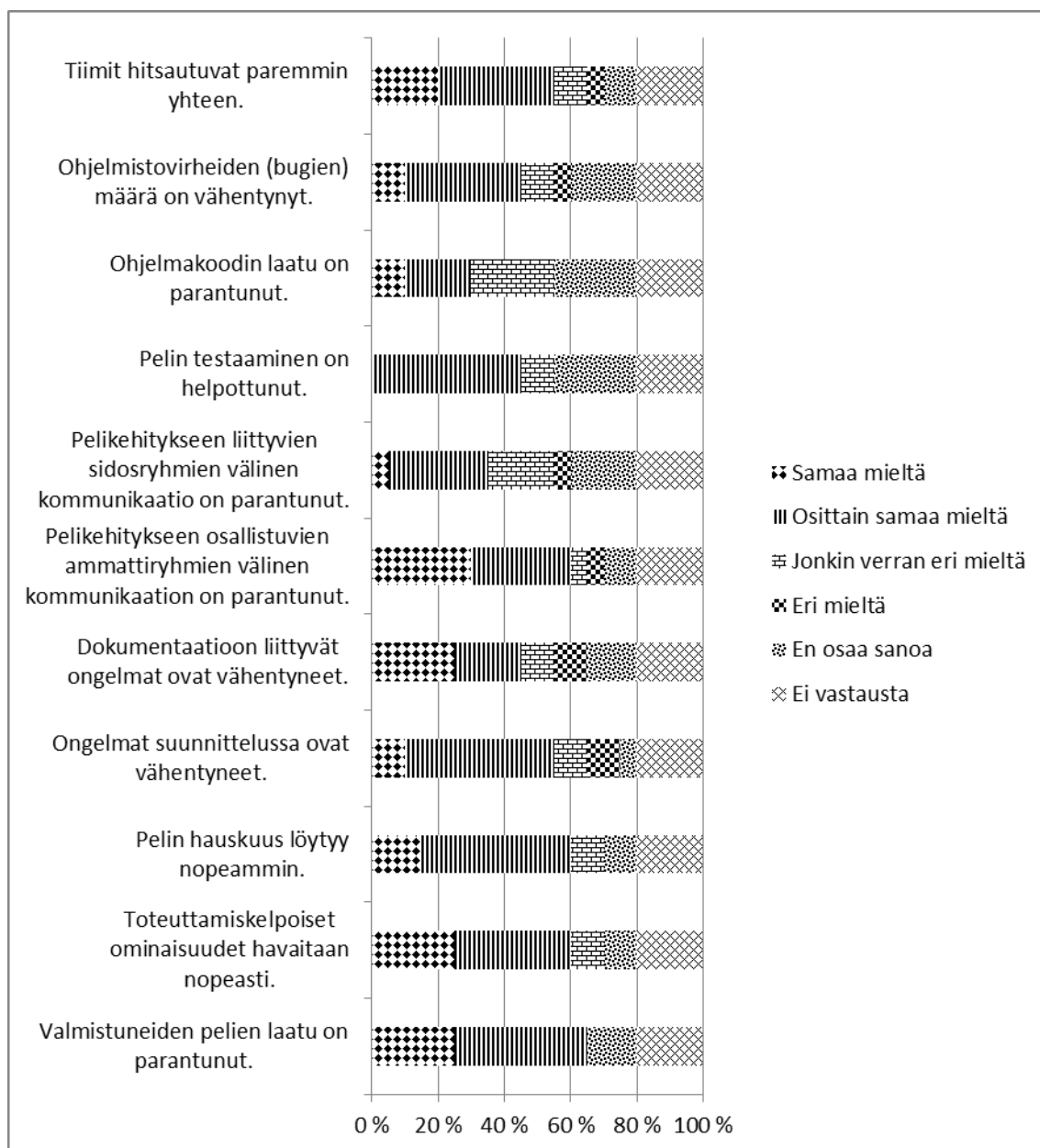
"Jokaisen pelin ja projektin ollessa tuotantovaiheessa, usein joudutaan lähes kaikkia osa-alueita räätälöimään. On muistettava että osa yrityksistä saa tulo-rahoitusta myös muualta kuin pelistä peleistä. Tuotantovaiheiden ja lopputyön välillä on runsaasti eri muuttujia, joita ei voida terminoida tietyn kaavan puitteissa."

6.5 Ketterien menetelmien positiiviset ja negatiiviset vaikutukset

Tämän osion kysymyksissä tiedusteltiin vastaajien mielipiteitä ketterien menetelmien hyödyntämisestä seuraavista positiivisista ja negatiivisista vaikutuksista. Osio sisältää kolme kysymystä, joista ensimmäinen koskee lähinnä varsinaista pelikehitystä, toinen projektinhallintaa ja viimeinen havaittuja negatiivisia vaikutuksia ylipäätään. Viimeinen kysymys on muodoltaan avoin ja muut tämän osion kysymykset ovat väittämiä, joiden vastausasteikko kuuluu "Samaa mieltä", "Osittain samaa mieltä", "Jonkin verran eri mieltä", "Eri mieltä" ja "En osaa sanoa". Neljä (20%) vastaajaa ei antanut vastausta tässä osiossa.

Osion ensimmäisen kysymysryhmän yhteydessä kysyttiin "Pitävätkö seuraavat ketterien menetelmien hyödyntämiseen liittyvät väittämät paikkaansa?". Väittämät perustuvat kirjallisuudessa esitettyihin ketterien menetelmien hyö-

dyntämisen myönteisiin seurauksiin (Beck & Andres, 2004; Keith, 2010; Musil, Schweda, Winkler & Biffel, 2010a; Schwaber & Beedle, 2001). Kaikki väittämät keräsivät enemmän puoltavia ("Samaa mieltä" ja "Osittain samaa mieltä") kuin kieltäviä ("Jonkin verran eri mieltä" ja "Eri mieltä") vastauksia, joskin on huomattava, että osaan väittämistä kertyi myös huomattava prosentuaalinen osuus "En osaa sanoa" vastauksia (kuvio 16).



KUVIO 16 Ketterien menetelmien myönteiset vaikutukset kehittämistyöhön (N = 16)

Erityisesti ammattiryhmien välinen kommunikaatio, johon kuusi kymmenestä yrityksestä vastasi myönteisesti, ja pelien laatu, johon annettiin hieman yli 60% prosenttia myönteisiä vastauksia, ovat parantuneet. Vaikka ohjelmakoodin laa-

tu ja pelikehitykseen liittyvien sidosryhmien välinen kommunikaatio ovat enemmistön mielestä parantuneet (30% ja 35% myönteisiä vastauksia), ovat nämä kaksi väittämää keränneet eniten kielteisiä vastauksia (molemmissa 25% vastauksista).

Osion toinen kysymys kuuluu "Pitävätkö seuraavat ketterien menetelmien hyödyntämiseen projektinhallinnassa liittyvät väittämät paikkaansa?". Väittämät perustuivat osin eri lähteiden esittämiin ketterien menetelmien hyödyntämisen myönteisiin seurauksiin (Beck & Andres, 2004; Keith, 2010; Musil, Schweda, Winkler & Biffl, 2010a; Schwaber & Beedle, 2001) ja osin pelikehityksessä havaittuihin projektinhallinnallisiin ongelmiin (Petrillo ym., 2009). Osion väittämät jakoivat vastaajia enemmän kuin edellisen osion väittämät (kuvio 17). Kolme väittämää keräsi enemmän kieltäviä kuin puoltavia vastauksia. Erityisesti väittämät "Tarve ylitöille on vähentynyt erityisesti projektien loppuvaiheissa" ja "Kehityksen aikainen ominaisuuksien kasvu (feature creep) ei aiheuta ongelmia aikatauluun, budjettiin tai laatuun" keräsivät suuren määrän kieltäviä vastauksia. Muiden väittämien osalta enemmistö vastanneista puolsi ketterien menetelmien myönteisiä vaikutuksia projektinhallintaan. Esimerkiksi projektin hallinnan koki helpommaksi ainakin osittain noin kaksi kolmasosaa vastaajista. Lisäksi huomattava osa vastaajista, neljä kymmenestä yrityksestä, ei osannut arvioida, ovatko henkilöstön vaihdoksien vaikutukset pienentyneet.

Osion viimeisessä, avoimessa kysymyksessä pyydettiin vastaajia kertoamaan omin sanoin, millaisia negatiivisia vaikutuksia ketterillä menetelmillä on ollut pelikehityksessä. Kysymykseen kertyi viisi vastausta. Näistä kolme kuvailee jotakin ketterien menetelmien piirrettä tai vaikutusta, josta on jonkinlaista häiriötä.

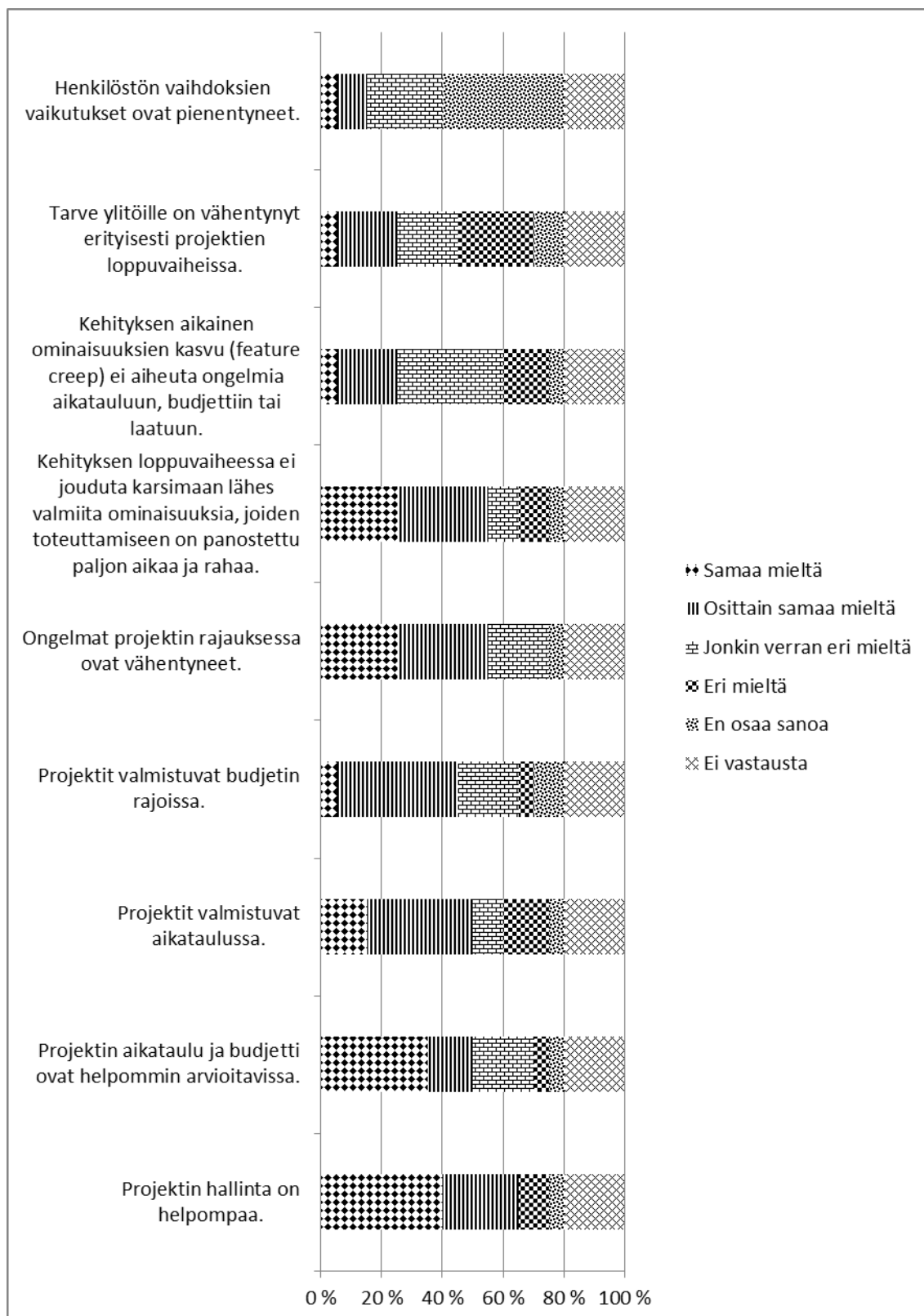
"Ketterät menetelmät siirtävät vastuuta tiimille ja pelisäännöt nousevat tärkeään rooliin. Tämä ei välttämättä loppupelissä sovi kaikille tiimeille."

"Hyväksytään muutosten väistämättömyys hieman liian itsestäänselvyytenä -> otollinen maaperä feature creepille."

"Pahimmillaan ollaan ns. jatkuvan crunchin tilassa, mikäli sprintit otetaan tosissaan. Muita negatiivisia vaikutuksia ei havaittavissa."

Yksi vastaaja kuvailee pelikehityksen eroja tavalliseen ohjelmistokehitykseen ja huomauttaa pelikehityksen olevan luonteeltaan ketterää:

"Olen tehnyt pelejä vain ketterästi, joten on hieman hankala verrata, että miten ketterät menetelmät ovat parempia kuin ei-ketterät nimeomaan peleissä.. (Muuta softaa olen tehnyt ei-ketterästi) On vaikea kuvitella, että pelin voisi tehdä esimerkiksi täysin vesiputousmallilla ei-ketterästi. Negatiivisia vaikutuksia on siis myös hankala listata. Tietyt ketterät ajatukset ja joustavuus tuntuvat olevan "pakollisia" pelienteossa. Itse me seurailimme ketterien menetelmien kehitystä ja firmastamme on mm. edustaja aina Scrum Agile:ssä (valitettavasti en minä joka vastaan), mutta toistaiseksi emme ole ottaneet käyttöön mitään yhtä menetelmää vaan käytössä on tosiaan firman sisäinen "agile" joka ottaa ajatuksia eri menetelmistä."



KUVIO 17 Ketterien menetelmien vaikutukset projektin hallintaan (N = 16)

Yksi vastaus pyrkii muistuttamaan pelikehityksen kompleksisuudesta ja taloudellisten seikkojen vaikutuksesta pelikehitykseen sekä huomauttaa kyselyn puutteista, jotka johtuvat tutkimuksen rajauksesta.

"On muistettava että pelien [...] kehitystyö ei ole suoraviivaista koneelliseen valmistukseen verrattavaa toimintaa, muuttujien määrä on erittäin suuri ja jos yksikin näistä muuttujista toimii eri tavalla kun on suunniteltu. Projektien ketteryys katoaa. Toisaalta Visuaalinen liiketoimintamalli edellyttää paljon ideoita, ajattelua, testausta, mallinnuksia jne. Näiden tehtävien ennakointi peliprojekteissa on erittäin haastavaa. Meidän yrityksessä noudatetaan hyvin avointa ohjelma formaattia ja tulokset sekä aikataulut ovat toteutuneet aiottua nopeammin. Haluan samalla muistuttaa tutkimuksen tekijää siitä että tutkimuksessa ei ole yhtään kysymystä, joka olisi viitannut rahoitukseen tai näiden ongelmiin. Tutkija on luultavasti tehnyt kysymys rakenteet siinä uskossa, ettei taloudellista negatiivista vaikutusta ole. Totuus Suomen peliteollisuuden taustoissa on vain aivan toinen. Jos rahoitusta on riittävästi, mikä tahansa järjestelmä on toimiva. Mutta jos sitä rahaa on niukalti tai eletään yhtiön omalla tulo-rahoituksella, monasti hienot kehitystyökalut jäävät osittain käyttämättä tai peräti kokonaan. Valitettavasti Suomessa ei ole yhtään sellaista rahoitustahoa, joka keskittyisi pelkästään peli-alan kehitykseen ja rahoitukseen. (HUOM. Tekes ei ole sellainen organisaatio). Kirjoittajana huomioi: Halusin tuoda rahoitusongelman esille, koska kehitys suunnittelussa rahalla ja yhtiön taloudellisella asemalla on erittäin suuri merkitys myös Pelien sisältöihin jne."

6.6 Yhteenveto

Tässä luvussa esiteltiin yksityiskohtaisesti tuloksia tutkimuksesta, joka koski ketterien menetelmien ja käytänteiden hyödyntämistä suomalaisissa pelistudioissa. Kyselyllä pyrittiin valottamaan pelistudioiden taustaa, kehitysprosessia, ketterien menetelmien ja käytänteiden hyödyntämistä eri pelikehitysvaiheissa sekä näkemyksiä siitä, millaisia vaikutuksia, positiivisia tai negatiivisia, ketterien menetelmien käytöllä on ollut pelikehitykseen. Seuraavassa luvussa tuloksien pohjalta esitetään johtopäätöksiä.

7 TULOSTEN TARKASTELU

Tässä luvussa esitetään johtopäätöksiä tutkimustuloksista ja suhteutetaan ne kirjallisuudessa esitettyihin näkemyksiin ketterästä pelikehityksestä. Käsittely lähtee liikkeelle tärkeimmästä havainnosta, ja sen jälkeen edetään samassa järjestyksessä kuin tuloksia esittelevässä luvussa. Toiseksi luvussa pohditaan tutkimuksen reliabiliteettia ja validiteettia.

7.1 Johtopäätökset

Tutkimuksen mukaan suomalaiset pelistudiot käyttävät laajasti ketteriä menetelmiä pelikehityksessä. Ketteriä menetelmiä käsittelevään osuuteen vastanneista yrityksistä yhtä lukuun ottamatta kaikki hyödynsivät ketteriä menetelmiä pelikehitysprosessissaan. Samansuuntaiseen päätelmään ovat tulleet myös Musil ym. (2010) Itävallan pelialaa tutkiessaan. Tulosten perusteella ei voi väittää, että yrityksen taustatekijöillä, kuten koolla tai samanaikaisten projektien määrällä, olisi vaikutusta ketterien menetelmien hyödyntämiseen. On kuitenkin huomattava, että tutkimukseen vastanneet yritykset ovat pieniä tai korkeintaan keskisuuria pelistudioita, joiden voi olla helpompi hyödyntää ketteriä menetelmiä kuin suurien.

Tutkimus paljasti, että Suomessa on kaiken ikäisiä pelistudioita. Niiden koko vaihtelee pienestä keskisuureen, ja niiden pääasialliset kehitysalustat ovat PC ja mobiililaitteet. PC:n suosio kehitysalustana selittyy todennäköisesti selinpelien suosiolla. Mobiililaitteiden suosion taustalla on eri matkapuhelinalustoilla tapahtuvan pelaamisen ja niille myytävien pelien määrän kasvu (Kuittinen ym., 2010). Kaikkein suosituin genre on ajanvietepelit, sillä niiden kohderyhmä on suurin ja toisaalta niiden kehittäminen ei ole aikaa vievää ja kallista (Kuittinen ym., 2010). Tätä heijastelee keskimääräinen projektien pituus, joka tutkimustulosten perusteella pyritään pitämään lyhyenä. Yli puolet projekteista kestää alle vuoden. Kolmella yrityksellä neljästä on vähintään kaksi projektia käynnissä samanaikaisesti. Samanaikaisten projektien määrä viitanee

riskien minimointiin. Yksittäisen projektin epäonnistumisen seuraukset yrityksen talouteen ovat pienemmät, kun projekteja on samanaikaisesti käynnissä useampia.

Konseptivaiheessa suurin osa pelistudioistaideoi pelejä ja valitsee ideoiduista peleistä yhden pelikonseptin kehitettäväksi. Konsepti hyväksytään pelistudiossa sisäisesti. Konseptivaiheessa myös tehdään pelisuunnittelua, luodaan pelimekaniikka ja tuotetaan pelisuunnitteludokumentti. Pelistudiot luovat budjetin ja aikataulun sekä määrittelevät resurssit projektille konseptivaiheessa. Vähemmistö pelistudioista hyväksyttää konseptinsa julkaisijalla ja tehdä pelistä pelilaitteella pelattavan prototyypin.

Edellä kuvatut toimet on koostettu väittämiksi kyselyyn Keithin (2010) ja Musilin ym. (2010) ketterää pelikehitystä kuvaavan esityksen pohjalta. Konseptivaiheen osalta Keithin (2010) esityksestä poiketaan ainoastaan konseptin julkaisijalla hyväksyttämisen suhteen. Syynä tämän toimen puuttumiseen enemmistön tekemistä konseptivaiheen toimista saattaa olla pelistudion valitsema pelien julkaisumalli, jossa rahoitusta ei haeta julkaisijalta ja peliä jaellaan digitaalisesti pelistudion oman palvelun tai pelijakelupalvelun kautta (Kuittinen ym., 2010). Musil ym. (2010) puolestaan painottivat prototyyppien valmistamista omassa esityksessään, mitä enemmistö pelistudioista ei vielä konseptivaiheessa tee. Tämä saattaa johtua siitä, ettei julkaisijaa olla haettu rahoittajaksi peliprojektiin, ja näin ollen pelattavalle prototyypille ei ole tarvetta, kun konseptia esitellään mahdolliselle julkaisijalle (Kuittinen ym., 2010). Näyttäisi siis siltä, että pääosin konseptivaiheen toimet noudattavat kirjallisuudessa esitettyjä näkemyksiä. Digitaalisen pelijakelun mahdollistamana pelit kuitenkin julkaistaan itse tai jakelupalvelun kautta, jolloin julkaisija yhtenä välikätenä jää välistä.

Esituotantovaiheessa enemmistö pelistudioista päättää käytettävistä teknologioista, rakentaa kehitystyökalut ellei niitä ole valmiina, tekevät pelisuunnittelua, muuttavat pelimekaniikkaa ja pelisuunnitteludokumenttia. Lisäksi ne rakentavat taiteen tuotantolinjan, toteuttavat osan pelin tarvitsemasta taiteesta, tarkentavat budjettia, aikataulua ja tarvittavia resursseja tuotantolinjan suorituskyvyn perusteella. Pelistudiot myös tekevät pelistä laitealustalla pelattavan prototyypin, tekevät pelin ensimmäiset pelattavat versiot, pelaavat peliä ja tekevät pelitestausta. Tämä on täysin linjassa Keithin (2010) esityksen kanssa, jossa esituotantovaiheessa iteroidaan pelimekaniikkaa, sallitaan muutokset pelikonseptiin ja rakennetaan taiteen tuotantolinjaa sekä tarkennetaan kertyvän tiedon perusteella arvioita tuotantovaiheen budjetista, aikataulusta ynnä resursseista ennen kuin siirrytään tuotantovaiheeseen. Pelistudioiden tapa toimia esituotantovaiheessa on siis hyvin lähellä kirjallisuudessa esitettyä näkemystä.

Tuotantovaiheessa lähes kaikki pelistudiot tekevät pelisuunnittelua, muuttavat pelimekaniikkaa, tuottavat suurimman osan pelin taiteellisesta sisällöstä, pyrkivät parantamaan tuotantolinjaa, lisäävät pelisisältöä pelattavissa olevaan versioon pelistä, pelaavat peliä ja tekevät pelitestausta. Kirjallisuudessa Keith (2010) on esittänyt, että tuotantovaiheessa pelimekaniikkaa ei pitäisi enää muuttaa, koska siitä seuraa yleensä taideartefaktien osittainen uudelleen rakentaminen tai muutostyö. Muutokset ovat hyväksyttävissä, mikäli ne parantavat

peliiä ja eivät aiheuta liikaa muutoksia olemassa oleviin artefakteihin (Keith, 2010). Tarve muutokselle saattaa johtua liian nopeasta siirtymisestä esituotantovaiheessa tuotantovaiheeseen. Toisaalta ketterät menetelmät mahdollistavat muutostarpeeseen vastaamisen ja ylläpitävät ilmapiiriä, jossa muutokset ovat suotavia hyvin myöhäisessäkin vaiheessa. Eräs tutkimukseen vastannut huomautti eräässä avoimessa vastauksessa: "Hyväksytään muutosten väistämättömyys hieman liian itsestäänselvyytenä -> otollinen maaperä feature creepille."

Jälkituotantovaiheessa pelimekaniikkaa muutetaan vieläkin mutta huomattavasti vähemmän kuin tuotantovaiheessa. Tämän lisäksi huomattava osa pelistudioista tuottaa uutta pelisisältöä vielä jälkituotantovaiheessa. Keithin (2010) mukaan pelimekaniikkaan ei pitäisi tulla enää muutoksia jälkituotantovaiheessa eikä uutta pelisisältöä tulisi tuottaa enää tuotantovaiheen jälkeen. Syy pelimekaniikan muutoksiin ja pelisisällön tuottamiseen myöhäisessä vaiheessa saattaa olla liian tiukka julkaisuaikataulu, joka pakottaa siirtymään jälkituotantovaiheeseen liian aikaisin.

Pelikehitysprosessissa hyödynnetään pääosin Scrumia, jonka yhteydessä hyödynnetään XP:n käytänteitä varsinaisessa tekemisessä, ja muutamien pelistudioiden osalta myös Lean-ohjelmistokehityksen periaatteita. XP:tä ei tunnustettu erilliseksi kehittämismenetelmäksi prosessin tasolla, vaikka sillä onkin merkittävä rooli päivittäisessä työskentelyssä. Toisaalta pelistudiot käyttävät useimmiten hybridejä, joissa on aineksia useammasta menetelmästä, eikä näin ollen voida sanoa, että käytettäisiin jotain tiettyä, yksittäistä menetelmää.

Keithin (2010) esityksen mukaista Lean-ajatusmallien hyödyntämistä tuotantovaiheessa ei tulosten perusteella juuri esiinny suomalaisissa pelistudioissa ja Kanbanin hyödyntämistä vielä vähemmän. Tutkimustulosten perusteella vaikuttaakin siltä, että Musilin ym. (2010) esittämä Scrumin hyödyntäminen tuotantovaiheessa on suosituimpaa vielä tässä vaiheessa. Luultavasti Keithin (2010) esittämällä ideoilla kestää aikansa löytää tiensä pelistudioihin. Moni kyselyyn vastanneista myös ilmoitti käyttävänsä itse kehittämänsä menetelmää, joka perustuu ketterien menetelmien käytänteille.

Scrumin suosituimpia käytänteitä ovat tutkimuksen mukaan päivittäinen Scrum -tapaaminen, pyrähdykset, pyrähdyn suunnittelupalaverit ja työlistat niin pyrähdykselle kuin tuotteellekin. Ohjelmistokehitysalan työntekijöihin kohdistuneen "State of the Agile 2010" -kyselyn (Analysis.Net Research, 2010) mukaan suosituimpia Scrumin käytänteitä ovat iteraation suunnittelu, päivittäinen Scrum -tapaaminen, retrospektiivit ja pyrähdyn edistymiskäyrä. Näyttäisi siis siltä, että samat käytänteet nähdään hyödyllisiksi molemmilla aloilla.

XP:n suosituimpia käytänteitä ovat tutkimuksen mukaan yhdessä istuminen, poikkitoiminnallinen tiimi, informatiivinen työtila ja jatkuva integrointi. "State of the Agile 2010" -kyselyn (Analysis.Net Research, 2010) suosituimmat XP-käytänteet puolestaan olivat jatkuva integrointi (ja automaattiset ohjelmakoosteet), refaktorointi, koodauskäytänteet ja testilähtöinen ohjelmistokehitys. Voi olla, että erot johtuvat kyselyiden eroista, sillä tähän tutkielmaan liittyvässä kyselyssä ei kysytty hyödyntämisestä kaikkien XP-käytänteiden osalta.

Analysis.Net Researchin kyselyssä (Analysis.Net Research, 2010) käytänteitä ei kysytty menetelmäkohtaisesti. Toisaalta juuri päivittäisessä työssä peli- ja ohjelmistokehitys poikkeavat toisistaan, sillä pelikehityksessä on osajia useammalta eri alalta kuin ohjelmistokehityksessä keskimäärin, joten tämän tutkielman kyselyn vastaukset saattavat myös heijastaa tätä tosiasiaa.

Tämän tutkielman kyselyn mukaan ketterillä menetelmillä on kysytyjen asioiden kohdalla ainoastaan myönteistä vaikutusta tai ei ainakaan haittaavaa vaikutusta. Esimerkiksi ohjelmistovirheiden määrä on vähentynyt ja valmistuneet pelit ovat laadukkaampia. Myös ”State of the Agile 2010” -kyselyssä (Analysis.Net Research, 2010) havaittiin, että enemmistön mielestä ohjelmiston laatu on parantunut ketterien menetelmien hyödyntämisen myötä. Samaisessa tutkimuksessa todettiin, että yksi merkittävin etu ketterien menetelmien hyödyntämisestä on parantuneet mahdollisuudet hallita projektia. Myös tämän tutkielman kyselyssä saatiin samansuuntaisia tuloksia.

7.2 Tutkimuksen reliabiliteetti ja validiteetti

Tutkimusta tulee arvioida reliabiliteetin ja validiteetin suhteen. Hirsjärvi ym. (2004) toteavat tutkimuksen reliabiliteetista:

”Tutkimuksen reliabiliteetti tarkoittaa mittaustulosten toistettavuutta. [...] Reliabiliteetti tutkimus antaa ei-sattumanvaraisia tuloksia. Jos kaksi arvioijaa päätyy samaan tulokseen tai samaa kohdetta tutkitaan eri tutkimuskerroilla ja saadaan sama tulos, tulokset ovat reliabileita. Kvantitatiivisissa tutkimuksissa myös tilastollisilla menetelmillä voidaan osoittaa tutkimuksen reliabiliteetti.” (Hirsjärvi ym., 2004, s. 231)

Järvinen ja Järvinen (2004) toteavat, että suuri otoskoko lisää reliabiliteettia. Tässä tutkimuksessa perusjoukko muodostuu suomalaisista pelistudioista. Kyselytutkimuksen kohteeksi valittiin Suomen Pelikehittäjät ry:n jäsenistöstä ja NeoGamesin pelistudiolistauksesta ne yritykset, joilla on vähintään viisi työntekijää. Otoksen koostuminen kahden eri lähteen pelistudioista todennäköisesti vähentää vinoutumisen vaaraa.

Jokaisesta otokseen kuuluvasta pelistudiosta pyydettiin yhtä pätevää henkilöä vastaajaksi. Tämän takia tutkimuksessa saadaan vain yhden henkilön näkemys studion sisästä, mikä ei tietenkään täydellisesti edusta yrityksen koko henkilöstön mielipiteitä tai näkemyksiä. On kuitenkin huomattava, että vastaajaksi pyydettiin tutkimuksen aihepiiriin hyvin tuntevaa henkilöä, jolloin vastaajan tietämys ja ymmärrys ovat riittävällä tasolla. Sattumanvaraisuuden vähentämiseksi kysymyksiä aseteltiin yksiselitteisyyteen, joka varmistettiin testaamalla kysely neljällä koehenkilöllä.

Tutkimusprosessi on kuvattu tarkasti luvussa 5 ja tulokset luvussa 6, mikä helpottaa tutkimuksen toistettavuutta. Lisäksi kyselylomake ja saatekirjeet on tutkielman liitteissä, jotka myös helpottavat toistettavuutta. Tilastollisia menetelmiä reliabiliteetin arvioimiseen ei käytetty.

Reliabiliteetin ohella tutkimusta tulee arvioida validiuden suhteen. ”*Validius* tarkoittaa mittarin tai tutkimusmenetelmän kykyä mitata juuri sitä, mitä on tarkoituskin mitata (Hirsjärvi ym., 2004, s. 216).” Validiuden arvioimiseen on erilaisia näkökulmia. Tässä tutkielmassa validiutta arvioidaan sisäisen ja ulkoisen validiuden kautta. *Sisäinen validius* tarkastelee tutkimuksen kausaalisuhteita (Hirsjärvi ym., 2004), mikä suomenkielellä tarkoittaa, että tutkimuksessa tutkitaan sitä asiaa, mitä on tarkoituskin tutkia. *Ulkoinen validius* tarkoittaa tulosten yleistettävyyttä (Hirsjärvi ym., 2004).

Sisäisen validiuden takaamiseksi kyselytutkimus rakennettiin pelikehitystä ja ketterää kehittämistä käsittelevään kirjallisuuteen nojaten. Tutkimuksessa käytettiin Mannisen ym. (2006) vaihejäsenystä, joka perustuu laajaan kirjallisuustutkimukseen. Tutkimusalueelle ei ole käytettävissä mitään erityistä teoriaa.

Kyselylomake perustuu tiukasti pelialan todellisuuteen, ja siinä on käytetty alan omia termejä ja kieltä. Nämä seikat vähentävät kysymysten tulkinnanvaraisuutta ja parantavat sisäistä validiteettia. Joitakin kyselyn sanamuotoja olisi tullut hioa vielä paremmiksi, sillä jälkituotantovaihetta koskevassa osuudessa oli väittämä ”tuotetaan pelisisältöä”, jossa termi pelisisältö on liian tulkinnanvarainen ja saattoi aiheuttaa vääristymää tuloksissa. Myös pelistudion pelikehityksen vaiheissa hyödyntämiä ketteriä menetelmiä tiedusteltaessa olisi yhtenä vastausvaihtoehtona pitänyt tarjota Scrumin ja XP:n yhdistelmää, sillä tässä kohtaa saatiin useita vastauksia, joiden mukaan pelistudio hyödyntää muita menetelmiä. Moni vastaaja tarkensi avoimen kohdan vastauksessaan, että heidän menetelmänsä on yhdistelmä eri ketteristä menetelmistä tai oma menetelmä, jossa on vaikutteita ketteristä menetelmistä. Lisäksi eräs vastaaja totesi, ettei kysely sopinut heidän yrityksensä toimintaan, jossa keskitytään tekemään pieniä parannuksia virtuaalimaailmaan ja ”kaikki vaiheet on yleensä niputettu yhteen”. Kysely ei siten välttämättä sovi pelistudiolle, joka ei kehitä uusia pelejä vaan ylläpitää ja kehittää edelleen jo julkaistua peliä.

Tulosten yleistämiseen vaikuttavat otoksen koko ja edustavuus. Suomalaisen pelistudioiden tarkkaa määrää on vaikea arvioida. Varsin luotettavan arvion saa kuitenkin tarkastelemalla Suomen Pelikehittäjät ry:n jäsenluettelo ja Suomen pelialan kattojärjestön, Neogamesin, jäsenluettelo. Tässä tutkimuksessa otettiin lähtökohdaksi nimenomaan nämä jäsenluettelot (66 kpl) ja rajauduttiin niihin pelistudioihin (45 kpl), joissa oli vähintään viisi työntekijää. Yhteyshenkilöiden osoitteet saatiin 37 yrityksestä, joille kysely lopulta lähetettiin. Vastaukset saatiin 20 yritykseltä. Näistä muutamaa lukuun ottamatta kaikki oli vastannut kaikkiin kysymyksiin. Tämän perusteella voidaan sanoa, että tulokset edustavat varsin hyvin suomalaisia pelistudioita. Kustakin yrityksestä saatiin yhden henkilön käsitys asioista. Tällöin mahdolliset eriävät näkemykset eivät välttämättä tulleet julki. Toisaalta yrityksiltä pyydettiin kyselyn vastaajaksi ketteristä menetelmistä ja yrityksen kehitysprosessista hyvin perillä oleva henkilö. Koska kyselyn aihe liittyi niin läheisesti vastaajien omaan toimintaan, voidaan perustellusti olettaa, että vastaajat olivat motivoituneita ja suhtautuivat kyselyyn tosissaan.

8 YHTEENVETO

Tämän tutkielman tavoitteena oli selvittää, kuinka paljon ja millä tavalla ketteriä menetelmiä hyödynnetään pelikehityksessä. Tutkimus rajattiin koskemaan suomalaisia, yli viisi henkeä työllistäviä pelistudioita. Tutkimuskysymyksiksi asetettiin: Millaisia erityispiirteitä pelikehitykseen liittyy? Millaisia ketteriä menetelmiä hyödyntäviä pelikehitysprosesseja kirjallisuudessa esitetään? Miten ketterää pelikehitystä tehdään Suomessa? Tutkimuskysymyksiin etsittiin vastauksia kirjallisuuskatsauksella ja empiirisellä kyselytutkimuksella, joka toteutettiin vuoden 2011 kesän aikana.

Ensimmäisen tutkimuskysymyksen osalta voidaan todeta pelin luonne viihdetuotteena ja sen kehittämisen monet haasteet, jotka johtuvat usean eri alan osaajien tuotosten saumattomasta yhteen liittämisestä ja pelikokemuksen tuottamisen ongelmallisuudesta. Pelikehityksen ominaispiirteitä ovat iteratiivisuus ja monialaisuus. Pelialalla on myös ominaispiirteensä, jotka kumpuavat julkaisija-kehittäjä-kuluttaja kolmikön vuorovaikutuksesta. Tähän liittyen tutkielmassa tehtiin merkittävää selvitystä pelien julkaisumalleista.

Toisen tutkimuskysymyksen vastaamiseksi tutkielmassa kuvattiin ensin ketteristä menetelmistä Scrum, XP, Lean-ohjelmistokehitys ja Kanban sekä niiden yhteinen tausta. Toiseksi kuvattiin kaksi näkemystä ketterästä pelikehityksestä. Keithin (2010) mukaan kehitysmenetelmä räätälöidään jokaiseen vaiheeseen hieman erilaiseksi, jotta menetelmä vastaisi kunkin vaiheen erityisvaatimuksiin. Musilin ym. (2010) mukaan esituotantovaiheessa pelikonsepti kehitetään prototyyppien avulla, minkä jälkeen peli tuotetaan pelikehitykseen räätälöidyn Scrum-prosessin avulla ja lopuksi julkaistaan kehityksen päättämävaiheessa.

Kolmanteen tutkimuskysymyksen vastaamiseksi toteutettiin kyselytutkimus, johon osallistui 20 suomalaista pelistudiota. Tulosten mukaan suomalaiset pelistudiot ovat pieniä tai keski-suuria. Suosituimmat alustat ovat PC ja mobiililaitteet, ja suosituin genre on ajanvietepelit. Pelikehitysprosessi noudattaa suurelta osin kirjallisuudessa esitettyjä malleja. Erojakin esiintyi: konseptivaiheessa ei välttämättä tehdä prototyyppiä tai hyväksytetä konseptia julkaisijalla, tuotantovaiheessa saatetaan muuttaa pelimekaniikkaa ja tehdään pelisuunnitte-

lua ja jälkituotantovaiheessa saatetaan tuottaa pelisisältöä, tehdä pelisuunnitelua ja muuttaa pelimekaniikkaa.

Myös suomalaisissa pelistudioissa käytetään ketteriä menetelmiä laajasti. Scrum on suosituin, ja yleensä sitä on ”maustettu” XP:n keskeisimmillä käytänteillä. Myös hybridit ja omat, ketteristä menetelmistä ammentavat menetelmät ovat suosittuja. Lean-ohjelmistokehitystä ei juuri hyödynnetä, eikä Kanbania senkään vertaa. Kirjallisuudessa Leanilla ja Kanbanilla on suurempi rooli etenkin tuotantovaiheessa. Scrumin ja XP:n keskeisimmät käytänteet olivat odotetulla tavalla käytössä pelikehityksessä, mutta Lean-ohjelmistokehityksen ja Kanbanin käytänteiden hyödyntäminen ei ollut kirjallisuuden esittämällä tasolla.

Ketterillä menetelmillä nähtiin olevan enimmäkseen myönteisiä vaikutuksia niin varsinaiseen kehitystyöhön kuin projektin hallintaan. Harvat kielteiset vaikutukset liittyivät muutosten sallimiseen ja vastuun kantamisen sopivuuteen erilaisilla tiimeillä.

Tämä tutkimus on tiettävästi ensimmäinen, jossa tarkastelun kohteena on ketterien menetelmien ja käytänteiden käyttö suomalaisessa peliteollisuudessa. Itävallassa on tehty ketterien menetelmien käyttöä pelikehityksessä koskeva tutkimus, mutta se on huomattavasti pinnallisempi. Vaikka nyt toteutetussa kyselyssä vastanneiden määrä olikin varsin suuri (suhteessa perusjoukkoon), tulosten luotettavuuden parantamiseksi olisi hyvä saada tilannekuva laajemmalta joukolta pelistudioita.

Tutkimuksen tuloksia voidaan käyttää pohdittaessa, missä laajuudessa ja muodossa ketteriä menetelmiä ja käytäntöjä voitaisiin käyttää yksittäisen pelistudion kehitysprosessissa, sekä toisaalta lähtökohtana pelistudion käyttämän menetelmän vertailussa alan trendeihin. Lisäksi tutkimus tarjoaa tuoretta näkökulmaa entuudestaan ketteriä menetelmiä hyödyntäville pelistudioille.

Tutkimuksen pohjalta voidaan esittää monenlaista jatkotutkimusta. Jatkotutkimusta kaivataan esimerkiksi kansainvälisenä tutkimuksena, jossa mukaan tulisivat myös suomalaisia pelistudioita suuremmat ulkomaiset pelistudiot, samojen teemojen tiimoilta. Toisaalta olisi kiinnostavaa tehdä tapaustutkimus jostakin ketteriä menetelmiä hyödyntävästä pelistudiosta kehitysprosessiin liittyen ja koettaa löytää kirjallisuuden tunnistamat, parannusta vaativat kohdat. Näiden lisäksi ketterien menetelmien käyttö pelien ylläpitovaiheessa tai esimerkiksi massiivisen verkkoroolipelin edelleen kehittämässä voisivat olla mielekkäitä jatkotutkimusaiheita.

LÄHTEET

- Aarseth, E., Smedstad, S. M. & Sunnan, L. (2003). A multidimensional typology of games. Teoksessa Copier, M. & Raessens, J. (toim.), *Level Up: Digital Games Research Conference Proceedings.. Utrecht, Hollanti. 4.-6. marraskuuta.* (s. 48-53). Utrecht: University of Utrecht. Haettu osoitteesta <http://www.digra.org/dl/db/05163.52481.pdf>
- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002). *Agile software development methods: Review and analysis.* Espoo: VTT Electronic. Retrieved from <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>
- Agile Alliance. (2001). *Agile manifesto.* Haettu 23.2.2011 osoitteesta <http://agilemanifesto.org/>
- Analysis.Net Research. (2010). *State of agile survey 2010 VersionOne.* Haettu osoitteesta http://www.versionone.com/pdf/2010_State_of_Agile_Development_Survey_Results.pdf
- Apperley, T. H. (2006). Genre and game studies: Toward a critical approach to video game genres. *Simulation & Gaming*, 37(1), 6-23.
- Arhippainen, L. (2009). *Studying user experience: Issues and problems of mobile services – CASE ADAMOS: User experience (im)possible to catch?* Väitöskirja. Oulun Yliopisto. Haettu osoitteesta <http://herkules oulu.fi/isbn9789514291081/>
- Beck, K. & Andres, C. (2004). *Extreme programming explained: Embrace change.* (2. painos). Boston: Addison-Wesley Professional.
- Bethke, E. (2003). *Game development and production.* Plano, TX, USA: Wordware Publishing.
- Blow, J. (2004). Game development: Harder than you think. *Queue*, 1(10), 28-37.
- Callele, D., Neufeld, E. & Schneider, K. (2005). Requirements engineering and the creative process in the video game industry. Teoksessa Cantarella, J. D. (toim.), *Proceedings of the 13th IEEE International Conference on Requirements Engineering. Paris, France. 29. elokuuta - 2. syyskuuta, 2005.* (s. 240-250). Los Alamitos: IEEE Computer Society.
- Cheesman, K. (2010, 10. Tammikuuta). *Illustrating scrum - A new and improved scrum diagram.* Haettu 7.10.2011 osoitteesta <http://blog.3months.com/2010/01/10/illustrating-scrum-a-new-and-improved-scrum-diagram/>
- Cook, D. (2002, 18. tammikuuta). *Evolutionary design - A practical process for creating great game designs.* Haettu 14.4.2011 osoitteesta <http://archive.gamedev.net/reference/articles/article1661.asp>
- Cowley, B., Charles, D., Black, M. & Hickey, R. (2008). Toward an understanding of flow in video games. *Computers in Entertainment*, 6(2), 1-27.
- Csikszentmihalyi, M. (1990). *Flow : The psychology of optimal experience.* New York: Harper & Row.

- FIGMA ry. (2011, 3. helmikuuta). *Suomen pelimarkkinoiden kehitys*. Haettu 11.2.2011 osoitteesta <http://www.figma.fi/tilastot.htm>
- Gamasutra. (2009, 17. syyskuuta). *GDC austin: An inside look at the universe of warcraft*. Haettu 4.2.2011 osoitteesta http://www.gamasutra.com/php-bin/news_index.php?story=25307
- Grigg, R. J. (2005). Pulp gaming: Taking an episodic approach to interactive entertainment. Teoksessa Pisan, Y. (toim.), *Proceedings of the second Australasian conference on Interactive entertainment*. Sydney, Australia. 23.-25. marraskuuta. (s. 75-82). Sydney, Australia: Creativity & Cognition Studios Press.
- Highsmith, J. & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-127.
- Hirsjärvi, S., Remes, P. & Sajavaara, P. (2004). *Tutki ja kirjoita*. (10. osin uudistettu painos). Helsinki: Kustannusosakeyhtiö Tammi.
- Hunicke, R., LeBlanc, M. & Zubek, R. (2004). MDA: A formal approach to game design and game research. Teoksessa Fu, D., Henke, S. & Orkin, J. (toim.), *Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence*. San Jose, California. 25.-29. heinäkuuta. (s. 1-5). Menlo Park, CA: AAAI Press. Haettu osoitteesta <http://www.cs.northwestern.edu/~rob/publications/MDA.pdf>
- Hyvärinen, L. & Kinnunen, J. (2010, 3. heinäkuuta). *Postmortem: Frozenbyte's trine*. Haettu 4.2.2011 osoitteesta http://www.gamasutra.com/view/feature/5826/postmortem_frozenbyte_s_trine.php
- Järvinen, P. & Järvinen, A. (2004). *Tutkimustyön metodeista*. Tampere: Opinpajan kirja.
- Jennett, C., Cox, A. L., Cairns, P., Dhoparee, S., Epps, A., Tijs, T., ym. (2008). Measuring and defining the experience of immersion in games. *International Journal of Human-Computer Studies*, 66(9), 641-661.
- Juul, J. (2003). The game, the player, the world: Looking for a heart of gameness. Teoksessa Copier, M. & Raessens, J. (toim.), *Level Up: Digital Games Research Conference Proceedings*. Utrecht, Hollanti. 4.-6. marraskuuta. (s. 30-45). Utrecht: Utrecht University. Haettu osoitteesta <http://www.digra.org/dl/db/05163.50560.pdf>
- Kanode, C. M. & Haddad, H. M. (2009). Software engineering challenges in game development. Teoksessa Latifi, S. (toim.), *Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on*. Las Vegas, Nevada. 27.-29. huhtikuuta. (s. 260-265). Los Alamitos: IEEE Computer Society. Haettu osoitteesta <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5070627>
- Keith, C. (2010). *Agile game development with scrum*. (1. painos). Boston: Addison-Wesley.
- Kniberg, H. & Skarin, M. (2009). *Kanban and scrum - making the most of both*. USA: InfoQ.
- Kuittinen, I., Virtala, T., Hyvärinen, L., Heikkinen, H., Lyytikäinen, M., Mäkinen, J., ym. (2010). *Suomen pelitoimialan strategia 2010-2015 - visio 2020*. Hel-

- sinki: Neogames, Suomen Pelinkehittäjät ry. Haettu osoitteesta <http://hermia-fi-bin.directo.fi/@Bin/14f8e6ab8b8283d6b9212aca9887203c/1290493416/application/pdf/725309/Pelistrategia%202010-2015.pdf>
- Ladas, C. (2009). *Scrumban - essays on kanban systems for lean software development*. USA: Modus Cooperandi Press.
- Manninen, T., Kujanpää, T., Vallius, L., Korva, T. & Koskinen, P. (2006). *Game production process: A preliminary study* (Raportti v. 1.0 - 28.2.2006). Oulu: University of Oulu, Department of Information Processing Science, LudoCraft. Haettu osoitteesta http://ludocraft oulu.fi/elias/dokumentit/game_production_process.pdf
- Mäyrä, F., Sihvonen, T., Saarenpää, H., Kultima, A., Paavilainen, J., Stenros, J., ym. (2009). *Pelitieto - pelien peruskurssi*. Haettu 5.1.2011 osoitteesta <http://pelitieto.net/>
- Musil, J., Schweda, A., Winkler, D. & Biffel, S. (2010a). Improving video game development: Facilitating heterogenous team collaboration through flexible software processes. Teoksessa Riel, A., O'Connor, R., Tichkiewitch, S. & Messnarz, R. (toim.), *Proceedings of the 17th European Conference on System, Software and Services Process Improvement. Grenoble, Ranska. 1.-3. syyskuuta, 2010.* (s. 83-94). Berlin: Springer. Haettu osoitteesta <http://books.google.fi/books?id=W2gLUlLoUJAC&printsec=frontcover#v=onepage&q&f=false>
- Musil, J., Schweda, A., Winkler, D. & Biffel, S. (2010b). *A survey on a state of the practice in video game development* (Tekninen raportti IFS-QSE 10/04). Vienna, Itävalta: Institute of Software Technology and Interactive Systems. Haettu osoitteesta <http://qse.ifs.tuwien.ac.at/publication/IFS-QSE-10-04.pdf>
- Musil, J., Schweda, A., Winkler, D. & Biffel, S. (2010c). Synthesized essence: What game jams teach about prototyping of new software products. Teoksessa Kramer, J., Bishop, J., Devanbu, P. & Uchitel, S. (toim.), *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2. Cape Town, South Africa. 2.-8. toukokuuta.* (s. 183-186). New York, NY: ACM. Haettu osoitteesta http://portal.acm.org/ft_gateway.cfm?id=1810325&type=pdf&coll=DL&dl=GUIDE&CFID=9919223&CFTOKEN=53188979
- Peltoniemi, M. (2009). *Industry life-cycle theory in the cultural domain: Dynamics of the games industry*. Väitöskirja. Tampereen Teknillinen Yliopisto. Haettu osoitteesta <http://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/223/peltoniemi.pdf>
- Petrillo, F., Pimenta, M., Trindade, F. & Dietrich, C. (2009). What went wrong?; A survey of problems in game development. *Computers in Entertainment*, 7(1), 1-22.
- Petrillo, F. & Pimenta, M. (2010). Is agility out there?: Agile practices in game development. Anonymous *Proceedings of the 28th ACM International Confer-*

- ence on Design of Communication. Sao Carlos, Sao Paulo, Brazil. (s. 9-15). New York, NY, USA: ACM. Haettu osoitteesta <http://doi.acm.org/10.1145/1878450.1878453>
- Pinsonneault, A. & Kraemer, K. L. (1993). Survey research methodology in management information systems: An assessment. *J.Manage.Inf.Syst.*, 10(2), 75-105.
- Poppendieck, M. & Poppendieck, T. (2003). *Lean software development - an agile toolkit*. Boston: Addison-Wesley.
- Potantin, R. (2010). Forces in play: The business and culture of videogame production. Teoksessa van den Abeele, V., Zaman, B., Obrist, M. & Ijsselsteijn, W. (toim.), *Proceedings of the 3rd International Conference on Fun and Games. Leuven, Belgia. 15.-17. syyskuuta.* (s. 135-143). New York, NY: ACM. Haettu osoitteesta [http://delivery.acm.org/10.1145/1830000/1823833/p135-pota-](http://delivery.acm.org/10.1145/1830000/1823833/p135-potantin.pdf?key1=1823833&key2=2779967921&coll=DL&dl=ACM&CFID=9919223&CFTOKEN=53188979)
[nin.pdf?key1=1823833&key2=2779967921&coll=DL&dl=ACM&CFID=9919223&CFTOKEN=53188979](http://delivery.acm.org/10.1145/1830000/1823833/p135-potantin.pdf?key1=1823833&key2=2779967921&coll=DL&dl=ACM&CFID=9919223&CFTOKEN=53188979)
- Riot Games. (2010). *Company*. Haettu 8.2.2011 osoitteesta <http://www.riotgames.com/company/team>
- Sanders, E. B. N. (2001). Virtuosos of the experience domain. Anonymous *Proceedings of the 2001 IDSA Education Conference. Massachusetts College of Art, Boston. 15.-18. elokuuta.* Haettu osoitteesta [http://www.maketools.com/articles-](http://www.maketools.com/articles-papers/VirtuososoftheExperienceDomain_Sanders_01.pdf)
[papers/VirtuososoftheExperienceDomain_Sanders_01.pdf](http://www.maketools.com/articles-papers/VirtuososoftheExperienceDomain_Sanders_01.pdf)
- Schwaber, K. & Beedle, M. (2001). *Agile software development with scrum*. Prentice Hall PTR.
- Schwaber, K. & Sutherland, J. (2008). *Scrum guide*. Internet: Scrum.org. Retrieved from <http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf>
- Sotamaa, O. (2007). Perceptions of player in game design literature. Teoksessa Baba, A. (toim.), *Situated Play: Proceedings of the 2007 Digital Games Research Association Conference. Tokyo, Japan. 24.-27. syyskuuta.* (s. 456-465). Tokyo: The University of Tokyo. Haettu osoitteesta <http://www.digra.org/dl/db/07311.59383.pdf>
- Sweetser, P. & Wyeth, P. (2005). GameFlow: A model for evaluating player enjoyment in games. *Comput.Entertain.*, 3(3), 3-3.
- Sydow, P. (2008). *Wouldn't It Be Cool If*. Video. Øredev Developers Conference 2008, Malmö. Haettu 14.2.2011 osoitteesta <http://www.viddler.com/explore/oredev/videos/12/>
- Tobey, B. (2008, 29. toukokuuta). *Introducing scrum at large animal games - a look back at the first year of agile development*. Haettu 13.5.2011 osoitteesta http://www.gamasutra.com/view/feature/3677/introducing_scrum_at_large_animal.php
- Wikipedia. (2010a, 6. marraskuuta). *Habbo*. Haettu 15.11.2010 osoitteesta <http://fi.wikipedia.org/wiki/Habbo>
- Wikipedia. (2010b, 31. joulukuuta). *Video game genres*. Haettu 5.1.2011 osoitteesta http://en.wikipedia.org/wiki/Video_game_genres

- Wikipedia. (2011a, 12. helmikuuta). *Blizzard entertainment*. Haettu 12.2.2011 osoitteesta http://en.wikipedia.org/wiki/Blizzard_Entertainment
- Wikipedia. (2011b, 3. helmikuuta). *Call of duty: Black ops*. Haettu 3.2.2011 osoitteesta http://en.wikipedia.org/wiki/Call_of_Duty:_Black_Ops
- Wikipedia. (2011c, 17. huhtikuuta). *Remedy entertainment*. Haettu 8.6.2011 osoitteesta <http://fi.wikipedia.org/wiki/Remedy>
- Wikipedia. (2011d, 2. kesäkuuta). *Rovio mobile*. Haettu 8.6.2011 osoitteesta http://fi.wikipedia.org/wiki/Rovio_Mobile
- Wikipedia. (2011e, 3. helmikuuta). *Sam & max save the world*. Haettu 8.2.2011 osoitteesta http://en.wikipedia.org/wiki/Sam_%26_Max:_Season_One
- Wikipedia. (2011f, 8. helmikuuta). *Tabula rasa*. Haettu 8.2.2011 osoitteesta [http://en.wikipedia.org/wiki/Tabula_Rasa_\(video_game\)](http://en.wikipedia.org/wiki/Tabula_Rasa_(video_game))
- Wikipedia. (2011g, 3. helmikuuta). *Treyarch*. Haettu 3.2.2011 osoitteesta <http://en.wikipedia.org/wiki/Treyarch>
- Wiltshire, A. (2009, 12. toukokuuta). *Interview: Telltale's dan connors*. Haettu 8.2.2011 osoitteesta <http://www.next-gen.biz/features/interview-telltale%E2%80%99s-dan-connors>
- Ye, Z. (2004). Genres as a tool for understanding and analyzing user experience in games. Teoksessa Dykstra-Erickson, E. & Tscheligi, M. (toim.), *CHI '04 extended abstracts on Human factors in computing systems*. Vienna, Austria. 24.-29 huhtikuuta. (s. 773-774). New York, NY: ACM. Haettu osoitteesta http://portal.acm.org/ft_gateway.cfm?id=985929&type=pdf&CFID=9919223&CFTOKEN=53188979

LIITE 1 SAATE 1

Arvoisa vastaanottaja,

Olen tekemässä tietojärjestelmätieteen pro gradu -tutkielmaa aiheesta "Ketterät menetelmät peliteollisuudessa". Ohjaajanani on lehtori Mauri Leppänen, KTT Jyväskylän yliopistosta. Tarkoitukseni on selvittää, missä laajuudessa ja muodossa ketteriä menetelmiä ja niiden käytänteitä käytetään pelikehityksessä. Tutkielmassa tarkastellaan mm. seuraavia asioita: pelikehityksen vaiheet, eri vaiheissa hyödynnettävät ketterät menetelmät ja käytänteet, pelikehityksen ongelmat ja ketterien menetelmien vaikutukset niihin. Tutkielmaan sisältyy kyselytutkimus, jonka kohteena ovat pelistudiot.

Pyytäisin, että yrityksenne osallistuisi kyselyyni. Vastaajana voisi toimia henkilö, jolla on hyvä tuntemus yrityksenne pelikehitysprosessista ja käyttämis-tänne kehittämismenetelmistä, erityisesti ketteristä menetelmistä. Kyselyyn vastaamalla autatte keräämään tärkeää tietoa Suomen peliteollisuuden tilasta ja ketterien menetelmien hyödyllisyydestä pelikehityksessä. Tutkimuksen valmistuttua voitte verrata toimintatapojanne muiden kyselyyn osallistuvien toimintatapoihin.

Vastaaminen tapahtuu täysin anonyymisti digitaalisen kyselylomakkeen välityksellä. Kysely löytyy osoitteesta: <https://korppi.jyu.fi/kotka/r.jsp?qid=6390>. Kyselyyn vastaamiseen kuluu vain 15-20 minuuttia. Vastausaikaa on **23.6.2011 saakka**.

Kyselyn tulokset tulevat nähtäville pro gradu -tutkielman valmistuttua 30.9.2011 mennessä Jyväskylän yliopiston julkaisuarkistoon (<https://jyx.jyu.fi/dspace/>), josta ne löytyvät haettaessa nimelläni tai haulla "Ketterät menetelmät peliteollisuudessa".

Kysymysten ja kommenttien tiimoilta voitte ottaa yhteyttä allekirjoittaneeseen.

Kiittäen,

Jussi Koutonen
jussi.koutonen@jyu.fi
+358505947751

LIITE 2 SAATE 2.1

Arvoisa vastaanottaja,

Olen tekemässä tietojärjestelmätieteen pro gradu -tutkielmaa aiheesta "Ketterät menetelmät peliteollisuudessa". Tarkoitukseni on selvittää, missä laajuudessa ja muodossa ketteriä menetelmiä ja niiden käytänteitä käytetään pelikehityksessä. Tutkielmaan sisältyy kyselytutkimus, jonka kohteena ovat pelistudiot. Kyselyä on jaeltu Suomen Pelikehittäjät RY:n jäsenyrityksille ja sitä kautta kyselyyn on jo saatu vastauksia. Vastauksia tarvitaan kuitenkin vielä lisää.

Mikäli yrityksenne työllistää vähemmän kuin viisi henkilöä tai olette jo vastanneet kyselyyni, voitte lopettaa lukemisen tähän. Kiitän yhteistyöstänne ja toivotan hyvää kesän jatkoa.

Muussa tapauksessa pyytäisin teitä välittämään minulle sellaisen henkilön yhteystiedot (sähköpostiosoite), joka osaa vastata kysymyksiin yrityksenne pelikehitysprosessista. Jos hyödynnätte ketteriä menetelmiä, scrum master on sopiva henkilö vastaamaan. Lähetän linkin kyselyyn suoraan vastaajaksi valitulle henkilölle.

Kyselyyn vastataan netissä, se on suomenkielinen, vastaamiseen kuluu aikaa noin 15 minuuttia ja vastausaikaa on 15.7.2011 saakka.

Pyydän teitä toimimaan ripeästi.

Kysymysten ja kommenttien tiimoilta voitte ottaa yhteyttä allekirjoittaneeseen.

Kiittäen,

Jussi Koutonen

jussi.koutonen@jyu.fi

+358505947751

LIITE 3 SAATE 2.2

Hei etunimi,

Olen tekemässä tietojärjestelmätieteen pro gradu -tutkielmaa aiheesta "Ketterät menetelmät peliteollisuudessa". Tarkoitukseni on selvittää, missä laajuudessa ja muodossa ketteriä menetelmiä ja niiden käytänteitä käytetään pelikehityksessä.

Kysely löytyy osoitteesta:

<https://korppi.jyu.fi/kotka/r.jsp?qid=6390>

Vastaamiseen kuluu aikaa noin 15-20 minuuttia. Kyselyyn vastataan anonyymisti. Kysely on vastattavissa 15.7.2011 saakka.

Gradu löytyy sen valmistuttua (arvioitu syys-lokakuun taitetta) Jyväskylän Yliopiston julkaisuarkistosta (<https://jyx.jyu.fi/dspace/>), kun hakee gradun otsikolla tai nimelläni.

Kiitoksia osallistumisesta ja hyvää kesän jatkoa,

Jussi Koutonen
jussi.koutonen@jyu.fi
+35850594775136

LIITE 4 KETTERÄT MENETELMÄT PELITEOLLISUUDESSA KYSELY

Yrityksen taustatiedot.

1. Yrityksen ikä?

- 0-2
- 3-5
- 5-10
- yli 10 vuotta

2. Kuinka monta työntekijää yrityksenne työllistää?

- 1-5
- 6-15
- 16-50
- Yli 50 henkeä

3. Mille laitealustoille kehitätte pelejä?

- PC
- Playstation 3
- Xbox360
- Nintendo Wii
- Käsikonsolit
- Mobiililaitteet
- Muu alusta

Jos vastasit "Muu alusta", kerro mikä se on.

4. Mitä genrejä kehittämänne pelit edustavat?

- Räiskintä (First Person Shooter)
- Roolipeli (Role-Playing Game)
- Mörppi (Massively Multiplayer Online Role-Playing Game)
- Seikkailu (Adventure)
- Toimintaseikkailu (Action adventure)
- Strategia (Strategy)
- Simulaatio (Simulation)
- Tasohyppely (Platformer)
- Ajanvietepelit (Casual games)
- Muu genre

Jos vastasit "Muu genre", kerro lyhyesti, mikä genre tai genret ovat kyseessä

5. Minkä mittaisia peliprojektit keskimäärin ovat?

- Alle puoli vuotta
- ½ - 1 vuosi
- 1 -2 vuotta
- 2-3 vuotta
- Yli 3 vuotta

6. Kuinka monta henkilöä keskimäärin osallistuu yhteen peliprojektiin?

- 1-5
- 6-15
- 16-30
- 31-50
- Yli 50 henkeä

7. Kuinka monta peliprojektia yrityksessänne on käynnissä samanaikaisesti?

- 1
- 2
- 3
- 4
- 5 tai enemmän

Kehitysprosessin vaiheiden kartoitus.

Tässä osiossa on tarkoitus kartoittaa, millaisia tehtäviä kuuluu pelikehitysprojehtin vaiheisiin.

Yhtenäisyyden vuoksi käytämme jaottelussa kirjallisuuteen pohjautuvaa mallia, jossa pelikehitys jakautuu neljään vaiheeseen:

- 1)Konseptivaihe, jossa keskitytään pelikonseptiin.
- 2)Esituotantovaihe, jossa keskitytään pelimekaniikan ja hauskuuden etsimiseen.
- 3)Tuotantovaihe, jossa pelimekaniikan varaan tuotetaan pelisisältö.
- 4)Jälkituotantovaihe, jossa keskitytään pelitestaukseen ja pelin hiomiseen julkaisua varten.

8. Kuinka usein seuraavat väittämät pitävät paikkansa konseptivaiheessa?

	Aina	Useimmiten	Harvoin	Ei koskaan	En osaa sanoa
Ideoidaan useita pelejä.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ideoista valitaan yksi pelikonsepti, josta aletaan tehdä peliä.

Tehdään pelisuunnittelua.

Luodaan pelimekaniikka.

Tuotetaan pelisuunnitteludokumentti.

Tehdään pelistä pelilaitteella pelattava prototyyppi.

Hyväksytään konsepti pelistudiossa sisäisesti.

Hyväksytään konsepti julkaisijalla.

Luodaan budjetti ja aikataulu sekä määritellään resurssit.

Kerro vapaasti konseptivaiheen keskeisistä tehtävistä, jotka mielestäsi tulee huomioida.

9. Kuinka usein seuraavat väittämät pitävät paikkansa esituotantovaiheessa?

	Aina	Useimmiten	Harvoin	Ei koskaan	En osaa sanoa
Päätetään käytettävistä teknologioista.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rakennetaan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

kehitystyökä-
lut ellei niitä
ole valmiina.

Tehdään peli-
suunnittelua.

Pelimekaniik-
kaan tulee
muutoksia.

Pelisuunnitte-
ludokument-
tiin tulee
muutoksia.

Rakennetaan
taiteen tuotan-
tolinja.

Toteutetaan
osa pelin tar-
vitsemasta
taiteesta (gra-
fiikasta, äänis-
tä, tarinasta
jne.)

Tarkennetaan
budjettia, ai-
kataulua ja
tarvittavia
resursseja
tuotantolinjan
suorituskyvyn
perusteella.

Tehdään pe-
listä laitealus-
talla pelattava
prototyyppi.

Tehdään pelin
ensimmäiset
pelattavat
versiot.

Kehitystiimi
pelaa peliä.

Tehdään peli-
testausta.



Kerro vapaasti esituotantovaiheen keskeisistä tehtävistä, jotka tulee mielestäsi huomioida.

10. Kuinka usein seuraavat väittämät pitävät paikkansa tuotantovaiheessa?

	Aina	Useimmiten	Harvoin	Ei koskaan	En osaa sanoa
Tehdään pelisuunnittelua.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pelimekaniikkaan tulee muutoksia.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tuotetaan suurin osa pelin taiteellisesta sisällöstä.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tuotantolinja on vakiintunut mutta sitä pyritään parantamaan mahdollisuuksien mukaan.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pelistä on olemassa pelattava versio, johon lisätään sisältöä.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kehitystiimi pelaa peliä.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tehdään pelitestausta.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kerro vapaasti tuotantovaiheen keskeisistä tehtävistä, jotka tulee mielestäsi huomioida.

11. Kuinka usein seuraavat väittämät pitävät paikkansa jälkituotantovaiheessa?

	Aina	Useimmiten	Harvoin	Ei koskaan	En osaa sanoa
Tehdään pelisuunnittelua.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pelimekaniikkaan tulee muutoksia.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tuotetaan pelisisältöä.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Hiotaan peli julkaisukuntoon.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tehdään pelitestausta.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tehdään alfa-testaus.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tehdään beetestaus.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Peli käy läpi julkaisijan laadunvarmistustestauksen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Peli käy läpi laitevalmistajan laitteistotestauksen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Kerro vapaasti jälkituotantovaiheen keskeisistä tehtävistä, jotka tulee mielestäsi huomioida.

Yrityksessänne hyödynnettävät ketterät menetelmät.

12. Mitä ketteriä menetelmiä hyödynnetään eri vaiheissa?

	Scrum	XP	Lean	Kanban	Muu menetelmä
Konseptivaihe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Esituotantovaihe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tuotantovaihe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Jälkituotantovaihe	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Muut vaiheet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Jos vastasit "Muut vaiheet" tai "Muu menetelmä", kerro niistä lyhyesti.

Ketterien menetelmien käytänteet yrityksessänne.

13. Scrum käytänteiden hyödyntäminen pelikehityksen eri vaiheissa.

sualisointi
(Visualize
workflow)

Läpikul-
kuajan
mittaami-
nen (Mea-
sure lead
time)



Jos käytänteitä räätälöidään eri vai-
heisiin, kerro siitä lyhyesti.



Ketterien menetelmien vaikutus pelikehityksessä havaittuihin ongelmiin.

16. Pitävätkö seuraavat ketterien menetelmien hyödyntämiseen liit- tyvät väittämät paikkaansa?

	Samaa miel- tä	Osittain sa- maa mieltä	Jonkin verran eri mieltä	Eri mieltä	En osaa sanoa
Valmistuneiden pelien laatu on parantunut.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Toteuttamiskelpoiset ominaisuudet havaitaan nopeasti.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pelin hauskuus löytyy nopeammin.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ongelmat suunnittelussa ovat vähentyneet.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dokumentaatioon liittyvät ongelmat ovat vähentyneet.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pelikehitykseen osallistuvien ammattiryhmien välinen kommunikaatio on parantunut.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pelikehitykseen liittyvien sidosryhmien välinen kommunikaatio on paran-	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

tunut.

Pelin testaaminen on helpot-
tunut.

Ohjelmakoodin laatu on
parantunut.

Ohjelmistovirheiden (bugi-
en) määrä on vähentynyt.

Tiimit hitsautuvat parem-
min yhteen.

17. Pitävätkö seuraavat ketterien menetelmien hyödyntämiseen pro- jektinhallinnassa liittyvät väittämät paikkaansa?

	Samaa mieltä	Osittain sa- maa mieltä	Jonkin verran eri mieltä	Eri mieltä	En osaa sanoa
Projektin hal- linta on hel- pompaa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projektin aika- taulu ja bud- jetti ovat hel- pommin arvi- oitavissa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projektit val- mistuvat aika- taulussa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Projektit val- mistuvat bud- jetin rajoissa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ongelmat projektin ra- jauksessa ovat vähentyneet.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kehityksen loppuvaihees- sa ei jouduta karsimaan lähes valmiita ominaisuuk- sia, joiden toteuttami- seen on pa- nostettu pal- jon aikaa ja rahaa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Kehityksen aikainen omi- naisuuksien	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

kasvu (feature creep) ei aiheuta ongelmia aikatauluun, budjettiin tai laatuun.

Tarve ylitöille on vähentynyt erityisesti projektien loppuvaiheissa.

Henkilöstön vaihdoksien vaikutukset ovat pienentyneet.



18. Kerro omin sanoin millaisia negatiivisia vaikutuksia ketterillä menetelmillä on ollut pelikehityksessä.

