

**Elina Keränen**

**Pelisuunnitteludokumentointi Jyväskylä Game Lab  
-peliprojekteissa**

Tietotekniikan pro gradu -tutkielma

15. lokakuuta 2017

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

**Tekijä:** Elina Keränen

**Yhteystiedot:** elinakera@gmail.com

**Ohjaajat:** Jonne Itkonen, Antti-Jussi Lakanen ja Vesa Lappalainen

**Työn nimi:** Pelisuunnitteludokumentointi Jyväskylä Game Lab -peliprojekteissa

**Title in English:** Game design documenting in JGL game projects

**Työ:** Pro gradu -tutkielma

**Suuntautumisvaihtoehto:** Pelit ja pelillisuus

**Sivumäärä:** 57+9

**Tiivistelmä:** Pelisuunnitteludokumentti (Game Design Document, GDD) on dokumentti pelisuunnittelusta, ja sen tarkoitus on organisoida pelin kehittämisprosessia monialaisen tiimin yhteistyönä. Sen pitäisi koota taiteellisen tuotteen visio ja kuvailla pelin ominaisuudet niin kattavasti että sitä voidaan käyttää luomaan ohjelmistotuote. Ongelmat GDD:ssä saattavat aiheuttaa projektitiimissä yhteisymmärryksen puutetta sekä uudelleentehtävää työtä ja yrityksessä investointien menetystä. Siten GDD:llä on olennainen rooli peliprojektin onnistumisessa.

Jyväskylä Game Lab (JGL) oli projekti jonka tavoitteena on kehittää uusia koulutuksellisia metodeja pelikoulutukseen Jyväskylässä. Tämä tutkielma tutkii pelisuunnitteluprosessia JGL-projektiryhmissä, tarkoituksena analysoida miten ryhmät käyttivät suunnitteludokumentteja pelin kehityksen tukena, miten yleiset ongelmat ilmenivät sekä vertailla prosessia yleisiin kirjallisuudessa esitettyihin käytäntöihin. Tutkimuksen tuloksena syntyy jatkoideoita siitä millaisia asioita peliprojekteissa pitäisi tutkia dokumentoinnin kannalta.

**Avainsanat:** GDD, pelit, pelisuunnittelu

**Abstract:** Game Design Document (GDD) is a key document about game design, and its purpose is to organize game development in order to facilitate the collaboration of multidisciplinary teams. It should capture the overall vision of an artistic product, and describe all the features of the game so exhaustively that it can be used to create a software product.

Problems in GDD may cause lack of shared understanding in a project team, rework and loss of investments. Therefore GDD has an essential role to success of the game development.

Jyväskylä Game Lab (JGL) was a project which aimed to develop new educational methods for game education in Jyväskylä. This thesis studies the game design process of JGL project groups, in purpose to analyze, how the documenting is used for the benefit of the game development, what common problems occur and compare the process to the common practices presented in literature. As a result of research is generated information how the documenting affects to the game project and how common practices and problems should be taken into account.

**Keywords:** GDD, games, game design

## Termiluettelo

Asset	Pelin käyttämät ulkoiset resurssit, kuten grafiikka ja äänet.
Elävä dokumentti	Dokumentti, jota päivitetään koko kehitysprosessin ajan.
Flowdock	Keskustelusovellus.
GDD	Game Design Document, pelisuunnitteludokumentti.
Hybridimenetelmä	Ohjelmistokehityksen menetelmä, joka yhdistää adaptiivisia ja lineaarisia menetelmiä.
Iteratiivinen	Suunnittelua ja toteutusta tehdään pienemmissä osissa ja tätä prosessia toistetaan päätyen lähemmäs valmista tuotetta.
JGL	Jyväskylä Game Lab. Jyväskylässä järjestetty opetuksellinen projekti joka tarjosi tiloja ja koulutusta pelinkehitystiimeille.
Jäätynyt dokumentti	Dokumentti, jota on lakattu päivittämästä huolimatta suunnitelmien muuttumisesta, ja siten sisältää vanhentunutta tietoa ja on ristiriidassa pelin kanssa.
Ketterä kehitys	Projektimalli, jossa ohjelmistokehitystä suoritetaan iteratiivisesti.
Konseptitaide	Luonnosvaiheen grafiikkaa tai ääniä.
TDD	Technical Design Document, tekniset asiat sisältävä suunnitteludokumentti.
Trello	Projektinhallintajärjestelmä, jossa tehtävien edistymistä hallinoidaan ja tarkastellaan.
Vesiputousmalli	Ohjelmisto- ja pelinkehityksen projektimalli, jossa suunnittelu- ja tuotantovaiheet suoritetaan peräkkäin eikä suoritettuun vaiheeseen enää palata.

## Kuviot

Kuvio 1. Ennakoiva prosessimalli. Mukailten Al-Azawi, Ayesha & Al-Obaidy (2014) .....	9
Kuvio 2. Adaptiivinen prosessimalli. Mukailten Al-Azawi ym. (2014) .....	9
Kuvio 3. Peliprojektin vaiheet. Mukailten Keith (2010).....	10
Kuvio 4. Aineiston keruu .....	20
Kuvio 5. Miten aineisto vastaa tutkimuskysymyksiin .....	22
Kuvio 6. Pointy End .....	26
Kuvio 7. Emberlands .....	27
Kuvio 8. APPrenticen ruudukkonäkymä .....	27
Kuvio 9. APPrenticen taistelunäkymä.....	28
Kuvio 10. Millä tavalla olet toiminut pelialalla? .....	54
Kuvio 11. Mitä kommunikointivälineitä ryhmällänne oli käytössä ja kuinka säännöllisesti käytit sitä? .....	55
Kuvio 12. Kuinka paljon aikaa vietit labilla? .....	55
Kuvio 13. Missä vaiheissa luit dokumentteja? .....	56
Kuvio 14. Mitä dokumentoiduista osa-alueista olet lukenut dokumenteista? .....	56
Kuvio 15. Jos olit projektin aikana epävarma mitä olitte sopineet jonkin tietyn asian ratkaisusta, miten tarkistit suunnittelupäätöksen? .....	57
Kuvio 16. Kävikö jossain vaiheessa niin että tuottamasi sisältö ei vastannut jonkun muun suunnitelmiin? .....	58
Kuvio 17. Miten eroavaisuudet suunnittelussa ja tehdyssä työssä kävivät ilmi? .....	58
Kuvio 18. Mistä eroavaisuudet suunnittelussa ja tehdyssä työssä johtuivat? .....	59
Kuvio 19. Miten eroavuuksiin reagoitiin .....	60

## Sisältö

1	JOHDANTO .....	1
2	PELI OHJELMISTOPROJEKTINA .....	3
3	DOKUMENTOINTI PELIALALLA .....	7
3.1	Pelituotannon vaiheet .....	8
3.2	Pelisuunnitteludokumentin sisältö .....	12
3.3	Pelisuunnitteludokumentointiin liittyvät ongelmat .....	14
4	TUTKIMUSMENETELMÄ .....	18
4.1	Tutkimusmenetelmän valinta .....	18
4.2	Aineiston keruu .....	19
4.3	Validiteetti .....	22
5	TUTKIMUSKOHTEET .....	25
5.1	Pointy End .....	25
5.2	Emberlands .....	25
5.3	APPrentice .....	26
6	TULOKSET .....	29
6.1	Dokumentin käyttäminen .....	29
6.2	Dokumentoimatta jääneet toteutetut ominaisuudet .....	35
6.3	Pelistä puuttuvat dokumentoidut ominaisuudet .....	39
7	JOHTOPÄÄTÖKSET .....	46
	KIRJALLISUUTTA .....	49
	LIITTEET .....	52
A	Pelisuunnittelijoiden haastattelu .....	52
B	Kysely .....	52
C	Kyselyn vastaukset .....	54

# 1 Johdanto

Pelisuunnitteludokumentti on dokumentti, joka kuvailee pelin suunnittelua: sen toiminnallisuutta, visiota ja taiteellista näkemystä. Se toimii kommunikointivälineenä monitieteisissä tiimissä sekä auttaa pitämään tavoitteen selkeänä ja säilyttämään yhteisymmärryksen visios-  
ta (Nevelsteen & Gayoso, 2011; Bethke, 2003). Vähimmillään suunnitteludokumentin täytyy kuvata pelin ominaisuudet eli toiminnalliset vaatimukset riittävän tarkasti (Bethke, 2003).

Yhtä tiettyä yleisesti hyväksyttyä formaattia dokumentille ei ole (Rouse III, 2005), vaan se kirjoitetaan pelin tavoitteisiin soveltuena. Jopa 11% ohjelmistoprojektien kustannuksista käytetään dokumentointiin (Zhi, Garousi-Yusifoglu, Sun, Garousi, Shahnewaz & Ruhe, 2015), joten on luonnollista, että dokumentoinnilla odotetaan olevan jotain hyötyjä. Hyötyjä ei kuitenkaan voida kartoittaa, ennen kuin tiedetään nykytilanne. Koska peli on ohjelmistotuote, ohjelmistoalan käytänteistä voi olla hyötyä myös pelialalla (Callele, Neufeld & Schneider, 2005). Perinteinen ohjelmistoalan dokumentointi ei kuitenkaan täysin toimi, vaan myös esimerkiksi ulkoiset resurssit kuten grafiikka ja äänet sekä pelin prototyypit ovat olennainen osa suunnittelua (Winget & Sampson, 2011).

Osa pelialan yleisimmistä ongelmista liittyy pelisuunnitteludokumentteihin (Petrillo, Pimenta, Trindade & Dietrich, 2009): dokumentoinnin puutteet aiheuttavat ongelmia tiimin jäsen-  
ten välisessä yhteisymmärryksessä sekä turhaa uudelleentehtävää työtä (Salazar, Mitre, Lemus & Sánchez, 2012), ja projektin laajuuden kasvaminen viivästyttää projektia (Kanode & Haddad, 2009). Ohjelmistoalalla ongelmien syntymistä ehkäistään parhailla käytänteillä jotka syntyvät tutkimuksen ja kokemuksen tuloksena, mutta peliala on hyvin nuori ala, ja siksi tutkimustietoa ei ole paljon saatavilla. Peliprojekteja tutkimalla voidaan selvittää, mitä käytäntöjä dokumentoinnissa omaksutaan ja miten ongelmat ilmenevät dokumenteista johtuen tai niistä riippumatta.

Jyväskylä Game Lab oli tutkimushanke, jonka tavoitteena oli kehittää uusia opetusmenetelmiä Jyväskylän yliopiston ja ammattikorkeakoulun peliopetukseen. Hankkeen osallistujista muodostettiin ryhmiä, joiden tarkoituksena oli pelialan ammattilaisten tuella kehittää peliprototyyppi viiden kuukauden aikana (JGL, 2016). Tämän tutkimuksen tarkoituksena on

selvittää, miten aloittelevat pelinkehittäjät käyttävät suunnitteludokumentointia pelin kehityksen tukena ja mitä yleisiä dokumentointiin liittyviä ongelmia projektin aikana ilmenee.

Tutkimus toteutetaan eksploratiivisena tapaustutkimuksena siten, että JGL-ryhmien tuottamia dokumentteja analysoidaan vertaillen niitä kirjallisuudessa esiteltyihin parhaisiin käytänteisiin, valmistuneita pelejä tarkastellaan dokumenttien valossa sekä ryhmiä haastatellaan miten dokumentit tuotettiin, miten niitä käytettiin projektin aikana sekä mitä ongelmia ilmeni. Tutkimuksen tuloksena syntyy tietoa siitä, miten dokumentointia käytetään ja mitä ongelmallisia osa-alueita saattaa olla. Tavoitteena on myös selvittää, mitkä aiheet dokumentoinnin tiimoilta nousevat tärkeiksi ja mitä tältä alalta kannattaisi tutkia jotta saataisiin tietoa dokumentoinnin hyödyistä ja suosituksista.

Seuraavaksi käsittelen pelisuunnitteludokumentoinnin erityispiirteitä vertaillen ohjelmistoprojektien dokumentoinnin käytänteisiin. Sen jälkeen käsittelen dokumentointia peliprojekteissa. Sitten esittelen tutkimusmenetelmän ja analysoin JGL-ryhmistä kerättyä informaatiota. Lopuksi teen yhteenvedon ja pohdin tulosten merkitystä.



## 2 Peli ohjelmistoprojektina

Peli on ohjelmisto, mutta se kuitenkin eroaa monin tavoin ohjelmistoprojektista. Pelin kenties näkyvin piirre ohjelmistona on se, että pelit käyttävät enemmän muita kuin ohjelmistoresursseja, esimerkiksi grafiikkaa ja ääniä (Kanode & Haddad, 2009). Peli on olennaisesti luova tuotos, ja pelin kehitys on ohjelmistokehitystä, johon on lisätty visuaalisuus, taiteellisuus ja luova suunnittelu (Kasurinen, Laine & Smolander, 2013). Mikäli pelin kehitystä kuitenkin voidaan käsitellä ohjelmistokehityksenä, pelin kehittäjät voivat hyötyä ohjelmistokehityksen käytännöistä. Seuraavaksi käsittelen peli- ja ohjelmistokehityksen eroavaisuuksia ja samankaltaisuuksia.

Zhi ym. (2015) määrittelevät ohjelmiston kehitysvaiheeseen liittyvän dokumentoinnin seuraavasti: 1) se on kirjoitettu kuvaus ohjelmistosta, 2) sen odotetaan tuottavan tarkkaa informaatiota ohjelmistosta, 3) se voi viitata käyttöoppaaseen joka on kirjoitettu ei-kehittäjille, 4) dokumentaatio on luotu kommunikointivälineeksi ohjelmistoinsinöörien välille, 5) dokumentaatio voi viitata artefakteihin kuten ohjelmistovaatimuksiin, suunnitteluun, ohjelmakoodin kommentteihin, testitapauksiin ja niin edelleen, 6) dokumentaatio voidaan esittää erilaisissa formaateissa, tekstin lisäksi esimerkiksi graafisina malleina. Suhteessa peliprojekteihin tärkein ero on kohdassa kaksi.

Ohjelmistokehityksessä ohjelmiston vaatimukset jaetaan kahteen luokkaan: toiminnallisiin ja laadullisiin. Toiminnalliset vaatimukset tarkoittavat toimintoja joita ohjelmisto suorittaa, esimerkiksi tekstinmuokkausohjelmassa tekstin muotoilu. Peleissä toiminnallinen vaatimus on esimerkiksi pelihahmon liikuttaminen näppäinpainallusten avulla. Laadulliset vaatimukset ohjelmistokehityksessä puolestaan kattavat esimerkiksi suorituskyvyn, ylläpidettävyyden, turvallisuuden ja luotettavuuden (Bourque & Fairley, 2014).

Pelien kohdalla nämä vaatimukset eivät kuitenkaan riitä. Pelien tapauksessa tärkein tavoiteltava laatuvaatimus on hauskuus: pelin kehittäjät kehittävät tunnekokemusta (Murphy-Hill, Zimmermann & Nagappan, 2014). Tämä tarkoittaa, että vaatimukset ovat subjektiivisempia (Washburn, Sathiyarayanan, Nagappan, Zimmermann & Bird, 2016). Vaatimus pelissä voi olla, että pelaajan täytyy viihtyä erilaisilla aikaskaaloilla (Murphy-Hill ym., 2014), mi-

hin päädytään eri tavalla erilaisissa peleissä. Siten pelien vaatimuksia on hankala määrittellä etukäteen ja kehitysprosessia on vaikea ennakoida (Murphy-Hill ym., 2014).

Dokumentoinnin suhteen tärkeä ero pelien ja ohjelmistojen välillä on Zhin määrittelyn kohta kaksi: tarkkaa informaatiota peleistä on vaikea tuottaa etukäteen, koska pelikokemusta on vaikea ennakoida. Tämän vuoksi pelin kehityksessä on parempi tuottaa korkean tason ohjeita, ja yksityiskohtaisen suunnitelman sijaan kuvailla, mitä kyseisellä ominaisuudella tavoitellaan (Murphy-Hill ym., 2014).

Dokumentointiin vaikuttaa myös projektiin valittu prosessimalli, josta tärkein aspekti on se, missä vaiheessa ohjelmiston vaatimukset määritellään ja missä vaiheessa dokumentit kirjoitetaan. Tämän ominaisuuden perusteella ohjelmistotuotantoprosessit voidaan jakaa kahteen eri tyyppiin: lineaarisiin ja adaptiivisiin malleihin (Bourque & Fairley, 2014). Lineaarissa mallissa, esimerkiksi vesiputousmallissa, projektin vaiheet suoritetaan peräkkäin: ensin määritellään ohjelmiston vaatimat ominaisuudet, sitten toteutetaan nämä vaatimukset (Bourque & Fairley, 2014). Mahdolliset muutokset vaatimuksiin toteutetaan formaalien prosessien kautta, mutta tavoitteena on määrittellä ne kokonaisuudessaan ennen tuotannon aloittamista (Bourque & Fairley, 2014). Dokumentointi siis kirjoitetaan ennen ohjelmiston tuottamista.

Adaptiivisissa malleissa puolestaan on olennaista kehittäminen sykleissä. Adaptiivisten mallien tärkeä piirre on iteratiivisuus, joka tarkoittaa että kehitys aloitetaan yksinkertaisesta implementaatiosta ja sitten tätä tuotetta parannetaan iteratiivisesti, lisäten ominaisuuksia tai muokaten olemassaolevia ominaisuuksia (Basili & Turner, 1975). Prosessi suoritetaan sykleissä, ja jokainen sykli sisältää suunnitteluvaiheen, toteutuksen ja tämän osittain valmiin tuotteen analyysin, sekä vaatimusten muokkaamisen tarpeen mukaan (Basili & Turner, 1975). Olennaista siis on, että vaatimusten suunnittelu jatkuu toteuttamisen ajan.

Esimerkki adaptiivisesta mallista on ketterä kehitys (*engl. agile*). Ketterän kehityksen kahdestatoista periaatteesta dokumentointiin liittyviä ovat erityisesti seuraavat (Ketterän ohjelmistokehityksen julistus, 2016).

*“2. Otamme vastaan muuttuvat vaatimukset myös kehityksen myöhäisessä vaiheessa.”* Dokumentoinnin kannalta tämä tarkoittaa, että dokumenttia ei kohdella valmiina tuotteena vaan

sitä voidaan muuttaa.

*“6. Tehokkain ja toimivin tapa tiedon välittämiseksi kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu.”* Tämä voi tarkoittaa, että dokumentointia on vähän tai ei lainkaan, tai se on epätäydellistä, jos kasvokkainen keskustelu voi olla korvaamassa sitä.

*“10. Yksinkertaisuus - tekemättä jätettävän työn maksimointi - on oleellista.”* Myös tämä tarkoittaa, että dokumentti ei ole valmis tuote: jos ominaisuus tai virheen korjaus valmistuu ilman dokumentointia, sen kirjaaminen ylös olisi ylimääräistä työtä ja siten se kannattaisi jättää tekemättä.

Ketterissä ohjelmistoprojekteissa dokumentoinnilla ei siis välttämättä katsota olevan hyötyjä, vaan painotetaan enemmänkin suoraa kommunikaatiota ja muutoksiin vastaamista (Stettina & Heijstek, 2011).

Dokumentoinnilla myös nähdään olevan ongelmia, mm. se on kallista (11% ohjelmistoprojektin kuluista käytetään dokumentointiin (Zhi ym., 2015)), jolloin kustannukset eivät välttämättä ole suhteessa hyötyihin mikäli dokumentointi ei ole tehokasta. Erityisesti startup-yritykset jotka toimivat rajallisten resurssien maailmassa, näkevät dokumentoinnin ylimääräisenä työnä.

Jako lineaarisen ja adaptiivisen prosessin välillä ei ole selkeä vaan se on enemmänkin jatkumo, ja ohjelmistoprojekteissa voidaan soveltaa erilaisia elementtejä eri prosessimalleista tarpeen mukaan. Pelinkehityksessä ketterät menetelmät ovat yleisiä: O’Hagan, Coleman & O’Connor (2014) tekivät kirjallisuuskatsauksen jonka mukaan 47% peliprojekteista käytti ketteriin menetelmiin kuuluvaa mallia, ja 53% yhdistelmää perinteisistä ja ketteristä menetelmistä.

Ketterillä ohjelmistokehitysmenetelmillä siis vaikuttaa olevan olennainen rooli pelikehityksessä. Dokumentoinnin kannalta tämä tarkoittaa erityisesti dokumentoinnin vähäisyyttä. Silmäkin kuitenkin on ongelmansa.

Stettina & Heijstek (2011) esittelivät ohjelmistokehitystä koskevissa tuloksissaan, että lähes puolet ketterien menetelmien käyttäjistä kokee sisäistä dokumentaatiota olevan liian vähän, vaikka suurin osa kokee dokumentaation olevan tärkeää. Tästä saattaa muodostua ongelma

erityisesti, kun tuote on toimitettu tai tiimistä lähtee jäseniä, jolloin menetetään projektin aikana kerättyä dokumentoimatta jäänyttä tietoa ja osaamista (Stettina & Heijstek, 2011).

Kun projektin koko kasvaa ja tuote kehittyy, dokumentaation rooli kasvaa, kun on tarpeen muistaa motivaatio ominaisuuksien taustalla, jotta voidaan ohjata uusien vaatimusten mukaantuloa (Bourque & Fairley, 2014). Lisäksi suulliseen kommunikaatioon nojatessa ollaan alttiita muistiongelmille ja ajan kuluessa on yhä vaikeampaa muistaa suunnittelupäätösten syitä (Stettina & Heijstek, 2011). Sama pätee myös kirjallisen viestinnän muotoihin, jotka eivät ole loogisesti vaan ajallisesti organisoituja: pitkistä sähköposti- tai keskusteluhistoriasista on sitä vaikeampi löytää relevanttia informaatiota, mitä enemmän informaatiota on. Tämä kaikki puoltaa organisoidun dokumentoinnin käyttöä.

Pelin kehityksen tärkeä piirre ohjelmistokehitysprojektina on siis pelisuunnittelun luova aspekti ja laadun subjektiivisuuden aiheuttama kehityksen ennakoimattomuus, mikä puolestaan voi johtaa iteratiivisuuden tarpeeseen. Pelin kehitys voi siten hyötyä erityisesti ohjelmistokehityksen ketterien mallien käytännöistä, ottaen kuitenkin huomioon tähän projekti-malliin liittyvät dokumentoinnin ongelmat.

### 3 Dokumentointi pelialalla

Pelisuunnitteludokumentointi sisältää kaksi käsitettä: suunnittelu ja dokumentointi. Pelisuunnittelu määrittelee pelaamisen: mitä pelaaja tekee, millainen pelimaailma on, mitä haasteita pelissä on, miten peli hävitään tai voitetaan (Rouse III, 2005). Pelisuunnittelua voidaan tehdä kirjoittamalla, tekemällä grafiikasta ja äänistä konseptitaidetta, samoin keskustelu, sähköpostien vaihto ja chat-ohjelmissa keskustelu voivat olla pelisuunnittelua.

Dokumentointi puolestaan tarkoittaa jonkinlaista järjestelmällistä asioiden tietovarastoon sijoittamista ja ylläpitoa. Dokumentointia voidaan tehdä eri formaateissa: se voi olla tekstiä, kaavioita, laskentataulukoita tai konseptitaidetta. Pelisuunnitelman lisäksi peliprojektissa mahdollisia dokumentteja ovat esimerkiksi markkina-analyysiin liittyvät dokumentit, monetisaatiosuunnitelmat, dokumentit joiden avulla peli-ideaa esitellään esimerkiksi mahdollisille rahoittajille, sekä projektin hallinnointiin liittyvät aikataulut, budjetit ja projektisuunnitelmat (Rouse III, 2005). Nämä eivät kuitenkaan käsittele itse pelin suunnittelua, vaan peliprojektiin liittyviä muita asioita.

Siis: kaikki dokumentit eivät ole suunnitteludokumentteja, ja kaikki suunnittelu ei ole dokumentointia. Tämä tutkielma käsittelee suunnittelun ja dokumentoinnin yhtymäkohtaa, eli dokumentteja, jotka liittyvät pelisuunnitteluun. Kun pelisuunnittelu lisätään luvun 2 dokumentin määritelmään (Zhi ym., 2015), pelisuunnitteludokumentin määritelmä voidaan tiivistää seuraavasti:

1. Pelisuunnitteludokumentti on kirjoitettu kuvaus pelin suunnittelusta ja se voi sisältää muita tiedon esitystapoja, esimerkiksi grafiikkaa.
2. Pelisuunnitteludokumentti tuotetaan projektiryhmän käytettäväksi projektin aikana.

Seuraavaksi käsittelen dokumentointiin liittyvää kolmea osa-aluetta: 1) missä vaiheessa dokumentointi tehdään, 2) mitä dokumentointi sisältää, 3) mitä ongelmia dokumentointiin liittyy.

### 3.1 Pelituotannon vaiheet

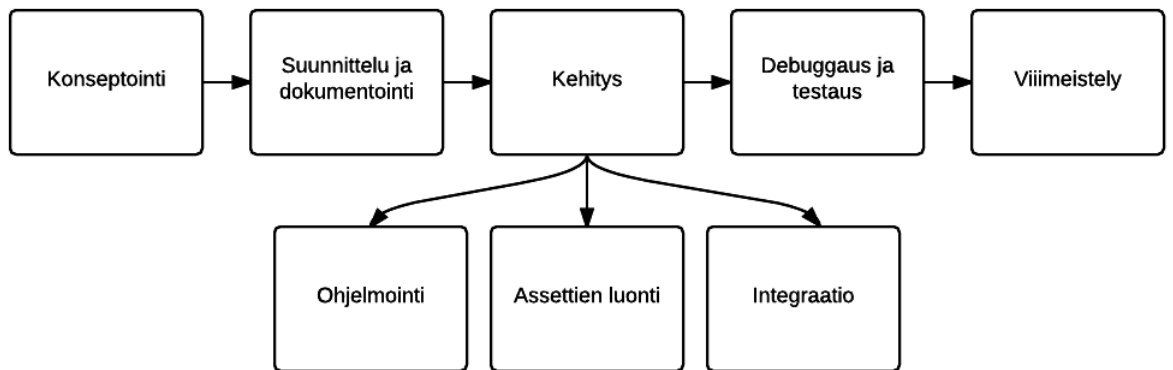
Pelikehitysprosessi voidaan jakaa neljään vaiheeseen: 1) konseptointi, 2) esituotanto, 3) tuotanto, 4) jälkituotanto (Keith, 2010). Konseptivaiheessa ideoidaan, kehitetään ja hylätään useita peli-ideoita (Keith, 2010). Esituotantoon valitaan yksi konseptointivaiheessa esitetty peli-idea, keskitytään sen suunnitteluun (Mitre-Hernández, Lara-Alvarez, González-Salazar & Martin, 2016), ja voidaan myös tuottaa kenttiä ja muita resursseja (Keith, 2010). Esituotantovaiheen tuloksena syntyy pelisuunnitteludokumentti.

Tuotantovaihe pohjautuu esituotantovaiheessa tuotettuun dokumenttiin (Winget & Sampson, 2011; Salazar ym., 2012; Kanode & Haddad, 2009). Tuotantovaiheessa luodaan pelin ohjelmisto sekä grafiikka, äänet ja muut resurssit (Mitre-Hernández ym., 2016). Tämän prosessin aikana ohjelmoijat, graafikot, äänisuunnittelijat ja muut kehittäjät käyttävät suunnitteludokumenttia oman työnsä pohjana ja tarvittaessa päivittävät dokumenttia (Almeida & da Silva, 2013). Projektin tuottajat tai projektipäälliköt puolestaan käyttävät dokumenttia aikataulutuksen perustana (Rouse III, 2005). Jälkituotannossa peliä testataan ja viimeistellään (Keith, 2010), tai jälkituotannolla voidaan viitata myös peli julkaisuun ja ylläpitoon (Mitre-Hernández ym., 2016).

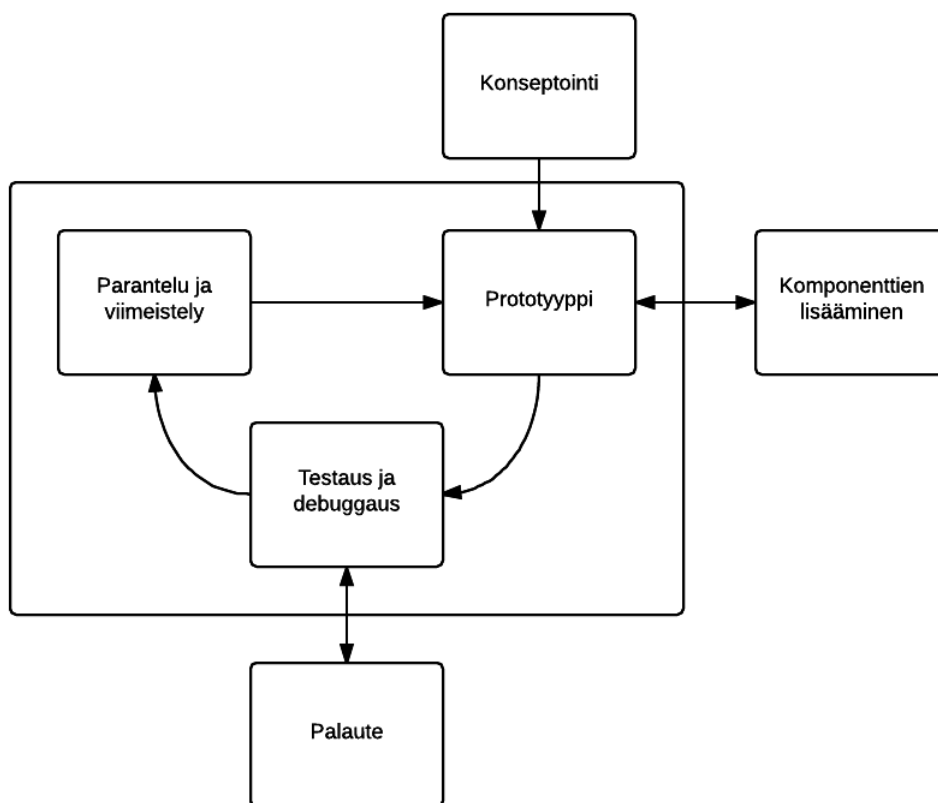
Kuten luvussa 2 kuvailtiin, ohjelmistokehitysprosessit voivat olla lineaarisia tai adaptiivisia. Kasurinen, Maglyas & Smolander (2014) hahmottelevat samankaltaisen jaottelun pelialalla. Linearisessa mallissa, kuten vesiputousmallissa, edellämäinitut neljä pelikehitysvaihetta ovat peräkkäisiä (kuva 1). Pelisuunnitteludokumentti tuotetaan kokonaisuudessaan esituotantovaiheessa, ja tuotantovaiheessa hyödynnetään tätä dokumenttia tuotannon perusteena, mutta uutta suunnittelua tai dokumentointia ei ole tarkoitus tapahtua (Callele ym., 2005; Winget & Sampson, 2011).

Adaptiivisiin malleihin lukeutuvat ketterät mallit. Ero lineaarisiin menetelmiin on, että vaiheet eivät tapahdu peräkkäisinä ja erillisinä, vaan iteratiivisesti: lyhyitä syklejä toistetaan ja sykleissä parannetaan tuotetta (Politowski, Fountoura, Petrillo & Guéhéneuc, 2016). Kuvassa 2 näkyy adaptiivisen prosessimallin eteneminen.

Ketterissä malleissa on siis olennaista, että erityyppisiin projektin vaiheisiin liittyviä tehtäviä suoritetaan koko kehitysprosessin ajan, ja jokainen sykli sisältää suunnittelua, tuotantoa

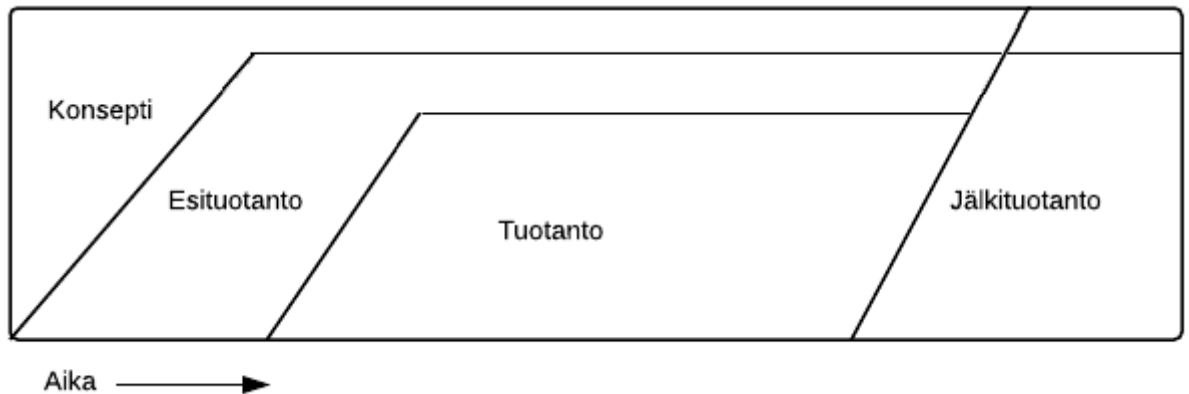


Kuvio 1. Ennakoiva prosessimalli. Mukailten Al-Azawi, Ayesha & Al-Obaidy (2014)



Kuvio 2. Adaptiivinen prosessimalli. Mukailten Al-Azawi ym. (2014)

ja jälkituotantoon liittyvää viimeistelyä. Keith (2010) esittää peliprojektin vaiheet yhtäaikaista tapahtuvina, pinottuna aluekaaviona, jossa käytetyn ajan osuutta kuvaa alueen pinta-ala ja x-akselilla on aika projektin alusta loppuun (kuva 3).



Kuvio 3. Peliprojektin vaiheet. Mukaillen Keith (2010)

Ketterässä mallissa siis esituotantovaiheelle tyypillisen suunnittelun määrä vähenee ajan kuluessa tuotantovaiheelle tyypillisen ohjelmoinnin, resurssien tuottamisen ja muun kehittämisen viedessä enemmän aikaa, mutta suunnittelu ei kuitenkaan lopu täysin.

Adaptiivisia ja lineaarisia menetelmiä voidaan myös yhdistellä. Näissä *hybridimenetelmissä* iteratiivisuus ilmenee erityisesti tuotantovaiheen aikana (Politowski ym., 2016), ja vaiheiden välillä pyritään mahdollisimman vähään määrään iteraatioita (O'Hagan ym., 2014), eli tuotantovaiheeseen siirryttyä ei enää ole tarkoitus palata suunnitteluvaiheeseen.

Ideaalisti ketterät menetelmät painottavat toimivan tuotteen toimittamista enemmän kuin kaikenkattavaa dokumentointia, ja peliprojektissa voidaan painottaa varsinaisen pelin tekemistä sekä prototyyppien kehittämistä. Silti myös ketterät projektit käyttävät jonkinlaista dokumentointia. Ketterälle kehitykselle on ominaista, että dokumentteja muokataan koko projektin ajan (Keith, 2010), jolloin dokumentti on *elävä* dokumentti (engl. living document) (Nevelsteen & Gayoso, 2011). Tämä tarkoittaa, että dokumenttia ja tuotetta muokataan yhtäaikaaisesti (Winget & Sampson, 2011).

Yleisesti hyvä dokumentointi voi johtaa esimerkiksi lyhyempään tehtävänteko-aikaan, parempaan ohjelmakoodin laatuun sekä korkeampaan tuottavuuteen (Zhi ym., 2015), kun taas



hallinnoimattomat muutokset voivat johtaa projektin laajuuden kasvamiseen, sekä ongelmiin aikataulutuksessa ja resursseissa (Kanode & Haddad, 2009). Vaatimusten tehokas kerääminen projektin alussa vähentää tarvetta iteraatioiden määrälle (Kanode & Haddad, 2009) ja siten myös ketterissä menetelmissä tavoitteena voi olla dokumentoinnin muokkaamisen pitäminen minimissään.

Toisaalta kuten luvussa 2 todettiin, peliprojekteille ominainen piirre on vaikeus tuottaa tarkkaa informaatiota etukäteen. Dormans (2012) esittelee käsitteen emergenssi: pelin toiminta on seurausta kompleksisesta sääntöjärjestelmästä, ja toisistaan riippumattomat pelimekaniikat aiheuttavat, että pelissä voi olla valtava määrä erilaisia mahdollisia tiloja. Esimerkiksi kysymykseen “kuinka korkealla kentän tasot voivat olla jotta pelaajahahmo voi hypätä sen päälle” on helppo vastata ilman pelaamista määrittelemällä pelaajahahmon hyppykorkeus ja sijoittamalla kentän tasot sopiville korkeuksille. Mutta mitä enemmän tekijöitä pelikokemukseen vaikuttaa, sitä vaikeampi on ennustaa, miten nämä vaikuttavat toisiinsa. Entä jos pelissä on  $x$  määrä  $y$  tyyppisiä vihollisia ympäristössä jossa on  $z$  välein ansoja, ja pelaajalla on  $n$  erityyppisiä taitoja? Kaikkia näitä voi olla vaikea määritellä aluksi, koska ne mahdollisesti havaitaan vasta myöhemmässä vaiheessa kehitystä.

Erilaisten tilojen määrän lisäksi toinen merkittävä seikka on jo luvussa 2 mainittu hauskuusfaktori. Peli tähtää kokemuksen tuottamiseen: pelaaja suorittaa tehtäviä joiden vaikeustaso kasvaa vähitellen ja tavoitteena on pelaamisen haasteellisuus sekä pelaajan tyytyväisyys (O’Hagan ym., 2014). Näitä voi olla vaikea arvioida ennen kuin peli on jossain määrin pelattavissa. Mikäli peli on tylsä tai ei riittävän viihdyttävä, suunnitelmia saatetaan joutua muuttamaan. Siten peliprojektin dokumentointi on tasapainoilua muutoksiin vastaamisen ja tehokkuuden välillä.

Se, millaista projektimenetelmää kannattaa käyttää, riippuu projektin tarpeista. O’Hagan ym. (2014) tekivät systemaattisen kirjallisuuskatsauksen peliprojekteista. 404 tutkimuksesta identifioitiin 356 ohjelmistoprosessia, jotka jaettiin 23 prosessimalliin. Näistä malleista 47% luokiteltiin ketteräksi menetelmäksi, ja 53% hybridimenetelmäksi. Politowski ym. (2016) puolestaan tutkivat 20 peliprojektia, joista 30% käytti vesiputoustyyppisiä prosesseja.

Kasurinen ym. (2013) tutkivat pienille ohjelmistoprojekteille suunnatun standardin sovel-

tuvuutta pelialalle, ja he tulivat tutkimuksensa perusteella siihen lopputulokseen, että erittäin pienet tiimit (alle 10 henkilöä) hyötyisivät iteratiivisesta lähestymistavasta ja suunnittelea ei pitäisi lopettaa ennen implementaatiota. Pienempien tiimien on helpompi nojata suoraan kommunikaatioon (Nevelsteen & Gayoso, 2011) ja ne ovat nopeampia mukautumaan muutoksiin (Kasurinen ym., 2013). Projektin koon kasvaessa taas myös kompleksisuus kasvaa, pelisuunnitteludokumentin tärkeys kommunikointivälineenä kasvaa ja sen myötä voidaan tarvita formaalimpaa prosessia (Callele ym., 2005; Nevelsteen & Gayoso, 2011).

Dokumentointiin liittyvät projektimenetelmän piirteet ovat siis missä vaiheessa kehitystä dokumentointi tapahtuu ja onko kyseessä elävä dokumentti vai suoritetaanko suunnittelu etukäteen. Tästä muodostuu tutkimuksen ensimmäinen teema: miten projektimenetelmän valinta tai piirteet vaikuttavat dokumentoinnin käyttämiseen ja missä vaiheissa dokumentointia tehdään?

### **3.2 Pelisuunnitteludokumentin sisältö**

Pelisuunnitteludokumenttiin viitataan yleisesti termillä GDD, Game Design Document. Sille ei ole yleisesti hyväksyttyä standardiformaattia, ja sen muoto vaihtelee pelistudioiden kesken (Nevelsteen & Gayoso, 2011). Pelisuunnitteludokumentti kuvailee vision pelille (Almeida & da Silva, 2013) ja sen tarkoitus on toimia kommunikointivälineenä tiimin jäsenten kesken sekä toimia muistiinpanoina (Schell, 2008).

Salazar ym. (2012) tekivät pelialan kirjoista ja tutkimuksista muodostuvan kirjallisuuskatsauksen pohjalta koosteen siitä, mitä yleisiä elementtejä GDD:n tulisi sisältää, ja siihen pohjautuen ehdotuksen pelisuunnitteludokumentoinnin parantamisesta ohjelmistokehityksen parhailla käytänteillä. Bethke (2003) ja Rouse III (2005) ovat paljon viitattuja kirjoja. Näistä yhteenvetona saadaan seuraavanlainen sisällysluettelo.

1. Yleiskatsaus, sisällysluettelo
2. Pelielementit
3. Ydinmekaniikat
4. Kontekstuaalinen pelaaminen: valikot, tallentaminen, ohjaimet
5. Resurssit: grafiikat, äännet, musiikki, erikoisefektit

## 6. Pelin eteneminen, pelikokemus ja tarina

## 7. Tekninen yleiskatsaus

Yleiskatsaus on korkean tason yhteenveto pelistä, ja se on tarkoitettu luettavaksi uusille tiiminjäsenille, ei niinkään kehittäjille, jotka tarvitsevat yksityiskohtaisempaa tietoa (Rouse III, 2005). Sisällysluettelon tarkoitus on helpottaa navigointia ja yksityiskohtaisemman tiedon löytämistä (Rouse III, 2005). Salazar ym. (2012) lisäsivät navigointia helpottamaan viitteet mahdollisiin muihin dokumentteihin sekä listat määritelmistä, lyhenteistä ja termeistä.

Salazar ym. (2012) nimittävät mekaniikoiksi pelin elementtejä, kuten pelaajaa tai vihollista. Rouse III (2005) nimittää samaa asiaa pelielementeiksi, mikä tarkoittaa hahmoja sekä pelin esineitä, joiden kanssa voi vuorovaikuttaa, kuten ovia tai poimittavia esineitä.

Rouse III (2005) nostaa pelimekaniikat tärkeimmäksi osioksi. Pelimekaniikkoja ovat esimerkiksi pelihahmon liikkuminen ja miten pelihahmo on vuorovaikutuksessa pelimaailman kanssa. Tämä osio kuvailee, miten peliä pelataan, ja sen on tarkoitus hyödyttää niitä, jotka tekevät peliä - ohjelmoijia, tarinankirjoittajia, graafikoita. Pyrkimyksenä on siis yksityiskohtaisuus yleiskatsauksenomaisuuden sijaan. Bethke (2003) sisällyttää siihen vain ydinpelimekaniikat, kun taas Salazar ym. (2012) sisällyttävät siihen pelin kaikki interaktiot, mukaanlukien valikot, tasot ja tekoälyn.

Bethke (2003) sisällyttää valikot kontekstuaalisen pelaamisen alle, johon sisältyy kaikki ydinpelimekaniikkojen ulkopuolinen pelin toiminta. Samoin Rouse III (2005) pitää omassa luvussaan pelin valikot, tallentamisen, peliohjaimet ja niin edelleen.

Pelin eteneminen on pelistä riippuen pisin osio (Rouse III, 2005). Tässä osiossa kuvaillaan, mitä pelaaja kokee, miten hän kehittyy, miten häneen vaikutetaan, mitä tasot sisältävät, millaisia haasteita on, millainen peli on visuaalisesti. Sen tarkoituksena on toimia oppaana graafikoille sekä kenttäsuunnittelijoille.

Salazar ym. (2012) ja Bethke (2003) jakavat tämän aihealueen kahteen osaan. Salazar ym. (2012) erottaa omaksi luvukseensa estetiikat, eli visuaalisen ilmeen sekä äänimaailman. Bethke (2003) nimittää tätä resursseiksi, sisältäen grafiikan, äänet, musiikin ja erikoiseffektit. Estetiikkaluku on usein erillinen dokumentti ja se sisältää konseptitaidetta visuaalisille elemen-

teille (Rouse III, 2005).

Toinen osio on pelikokemus: kuvaillaan mitä pelaajan oletetaan kokevan (Salazar ym., 2012). Bethke (2003) painottaa tässä osiossa pelin tarinaa, johon hän sisällyttää myös tasot ja hahmojen taustatarinat. Myös tämä voi olla erillinen dokumentti. Se sisältää myös taustatarinan, joka ei välttämättä näy pelissä suoraan (Rouse III, 2005).

Rouse III (2005) erottelee suunnitteludokumentista teknisen suunnitteludokumentin (Technical Design Document, TDD), jonka pohjalta ohjelmistotuotanto tehdään. Se sisältää ohjeistuksen pelin toteutukseen: koodin rakenne, grafiikan renderöinti, tekoäly ja niin edelleen (Rouse III, 2005). GDD:n ei siis ole sinänsä ohjelmistodokumentti, vaan sen perusteella kirjoitetaan vaatimusspesifikaatio formaalimmassa muodossa Callele ym. (2005). Salazar ym. (2012) kuitenkin lisäisi myös GDD:hen TDD:n yleiskatsauksen ja tekniset oletukset ja rajoitteet, mikäli ne vaikuttavat suunnittelupäätöksiin.

### **3.3 Pelisuunnitteludokumentointiin liittyvät ongelmat**

Yhden tietyn formaatin luominen suunnitteludokumentille pelialalla on osoittautunut hankalaksi (Rouse III, 2005). Ei ole yhteisymmärrystä miten, milloin ja mihin tarkoitukseen dokumentteja pitäisi kirjoittaa, vaikka käytännön työssä dokumentteja kirjoitetaankin (Dormans, 2012). Liian formaalit vaatimukset dokumentoinnille eivät kuitenkaan välttämättä toimi ja voivat jopa estää luovuutta (Callele ym., 2005).

Syynä sille että GDD:tä ei lueta voi olla, että GDD kasvaa liian isoksi (Nevelsteen & Gayoso, 2011). Useat pelialan ammattilaiset väittävät, että peliprojekteissa ei haluta lukea dokumentteja (Rouse III, 2005), mikä vastaa myös tutkimustuloksiin: esimerkiksi Winget & Sampson (2011) huomasivat, että 10/12 haastatelluista kertoivat dokumentinkäyttöaktiivisuuden laskevan tuotantovaiheen edetessä. Jos kukaan ei lue dokumentteja, ne eivät käytännössä ole kommunikointiväline.

Dokumenttien käyttöä kuitenkin puoltaa muun muassa se, että pelkkään suulliseen kommunikointiin ei voi nojata, vaan päätökset tulisi kirjata muistiin, jotta jokainen voi tarkistaa ne myöhemmin (Schell, 2008). GDD:tä ei ole tarkoitus lukea kuin kirjaa yhtenä kokonaisu-

tena vaan hyödyntää käyttäen kulloinkin tarvittuja osioita puhelinluettelon tavoin (Adams, 2007). Sitä käytetään suunnittelupäätösten dokumentointina ja lähteenä, josta tarkistetaan pelin kehityksessä kulloinkin tarvittavat asiat sitä mukaa kuin kyseisiä asioita tarvitaan.

Standardoitu dokumentointimalli johtaa kontrolloidumpaan suunnitteluprosessiin, ja samaa mallia voidaan käyttää esimerkiksi eri yrityksissä sekä yliopistoissa (Dormans, 2012). Standardoitu malli siten voisi parantaa tehokkuutta, kun aina ei tarvitse omaksua uutta käytäntöä. Huono pelisuunnitteludokumentti johtaa tavoitteen laajuuden kasvamiseen mikä voi aiheuttaa viivästyksiä tuotantovaiheessa (Kanode & Haddad, 2009). Hallinnoimattomat muutokset voivat aiheuttaa ongelmia liittyen toiminnallisuuteen, aikataulutukseen ja resurssien käyttöön (Kanode & Haddad, 2009).

Dokumentin ongelmana voi olla, että se kuvailee peliä niin yleistasolta, että siitä ei ole hyötyä toteutusvaiheessa (Rouse III, 2005). Washburn ym. (2016) tutkivat peliprojektien onnistumisia 155 projektissa. Niissä tapauksissa kun kehittäjät kokivat dokumentoinnin epäonnistuneen, se oli yleensä siksi että kunnollista dokumentaatiota ei ollut tarpeeksi. Esimerkiksi eräs tiimeistä vietti projektin loppuvaiheessa paljon aikaa yrittäessään ymmärtää miten jonkin asian oli tarkoitus toimia, koska siitä ei ollut dokumentaatiota.

Petrillo ym. (2009) tutkivat yleisimpien ongelmien esiintymistä 20 peliprojektissa. Suunnitteluvaiheen ongelmista eräs oli, että suunnittelua ei ollut lainkaan paperilla mikä aiheutti viivästyksiä (Petrillo ym., 2009). Tästä voi aiheutua ongelma, että vain yksi henkilö voi tehdä jonkun tietyn tehtävän (Petrillo ym., 2009), kun muilla jäsenillä ei ole riittävästi tietoa kyseisen tehtävän vaatimuksista, ongelmista tai tavoitteista. Tämä luonnollisesti aiheuttaa ongelmia erityisesti silloin, jos tiimiin ja kyseiseen tehtävään saapuu uusia jäseniä. Ongelmat ja ominaisuudet ovat ehkä ryhmän tiedossa, mutta dokumentoinnin puuttuessa tiedon siirtyminen uusille jäsenille on epävarmaa.

Dokumentti voi olla liian yksityiskohtainen, keskittyen asioihin jotka voidaan määritellä vasta esimerkiksi animointivaiheessa tai ohjelmoitaessa (Rouse III, 2005). Suunnitteluvaiheen ongelma on myös se, että suunnitellaan etukäteen liikaa: dokumentointia voi olla niin paljon että hyödyllistä tietoa on vaikea löytää (Rouse III, 2005). Myös rakenteen puute voi aiheuttaa, että dokumentista on vaikea löytää hyödyllistä tietoa ja tärkeät ja vähemmän tärkeät asiat

kirjoitetaan samalla prioriteetilla.

Fossiloitunut tai jäänyt dokumentti tarkoittaa, että dokumenttia lakattiin päivittämästä ja se sisälsi asioita jotka eivät päätyneet peliin (Rouse III, 2005). Jos samaa asiaa käsitellään useammassa paikassa, päivitysten tullessa mahdollisesti ei päivitetä kaikkia, jolloin GDD sisältää vanhentunutta tietoa (Rouse III, 2005). Tämä voi olla ongelma esimerkiksi jos kehittäjät implementoivat suunnitteludokumentin perusteella jotain vanhentunutta jota ei ollut tarkoitus implementoida, mikä tarkoittaa, että jo tehtyä työtä joudutaan korjailemaan tai sitä ei voi käyttää lainkaan.

Jäätyminen on ongelma erityisesti silloin, kun peliprojektissa pyritään elävään dokumenttiin (Winget & Sampson, 2011). Adaptiivisissa projekteissa elävä dokumentti vastaa suunnittelussa tapahtuviin muutoksiin ja päivittyy koko projektin ajan. Päivittämisen aktiivisuus vähenee huomattavasti projektin edetessä, jolloin dokumentin tarkkuus vähenee ja pelissä ilmenneet ongelmat eivät päädy suunnitteludokumentteihin (Winget & Sampson, 2011).

Jääntyneen dokumentin määritelmä riippuu siis osittain valitusta projektityylistä. Mikäli noudatetaan lineaarista mallia ja suunnitelmat eivät muutu suunnitteluvaiheen jälkeen, kyseessä ei ole jäänyt dokumentti vaan projektityylin ominaisuus. Jos suunnitelmien on tarkoitus muuttua ja pyritään adaptiivisten menetelmien mukaisesti elävään dokumenttiin, on mahdollista että elävyys ei onnistu ja dokumentti jäätyy: vanhentuneita suunnitelmia ei poisteta tai kaikkia uusia ominaisuuksia ei kirjata ylös.

Syy dokumentoinnin vanhentumiselle voi olla lennosta suunnittelu: pelisuunnitteluun liittyviä päätöksiä tehdään ohjelmoinnin yhteydessä ilman etukäteissuunnittelua, ja suunnitelmat vanhenevat sen vuoksi (Winget & Sampson, 2011). Perinteisessä mallissa ongelmana voi olla että epäonnistutaan pyrkimyksessä suunnitella kaikki etukäteen, ja vaikka suunnittelua tosiasiallisesti tapahtuu, ne eivät päädy virallisiin dokumentteihin.

Jos käytännön työn yhteydessä tehtävä suunnittelu on tarpeellista, ja dokumentit päivitetään jälkikäteen vain dokumenttien itsensä vuoksi eikä kukaan lue niitä, dokumentointi on muuttunut työkalusta itsetarkoitukseksi. Voiko asian, jota tavoitellaan dokumenttien lukemisella ja kirjoittamisella, korvata jollain muulla toiminnalla?

Asiassa on siis kaksi ääripäätä: toisessa ääripäässä dokumentointia ei tehdä lainkaan, mikä aiheuttaa vaikeutta kontrolloida resursseja ja kommunikointia, ja toisessa ääripäässä on formaali dokumentointi, joka saattaa aiheuttaa ylimääräisen työvaiheen kehitystyön jälkeen ja dokumentin käyttämisen epäkäytännöllisyyden. Tähän välille jää kaikki erilaiset yhdistelmät siitä, miten dokumentointia päivitetään: joko dokumenttiin jää ominaisuuksia joita ei ole tarkoitus implementoida, tai toteutetaan ominaisuuksia mutta ei dokumentoida niitä. Mikäli kommunikointi hoidetaan jotenkin muuten kuin dokumenteilla, dokumentoimattomuus tai päivittämättömyys eivät välttämättä ole ongelmia.

Tästä päästään tutkimuksen toiseen teemaan: millä alueilla esiintyy dokumentoinnin puutetta, onko päivittämättömyys ongelma vai ominaisuus, ja korvataanko dokumentointi jollain muulla menetelmällä?

## 4 Tutkimusmenetelmä

Edellinen luku käsitteli kolmea aihealuetta: peliprojektityylejä, pelisuunnitteludokumentointia sekä niihin liittyviä ongelmia. Näiden lukujen tuloksena kävi ilmi, että ongelmallisina alueina on päivittämättömyys. Projektityylejä sekä dokumentointia tutkimalla saadaan selville, miten päivittämättömyys ilmenee JGL-peliprojekteissa. Päivittämättömyys voi liittyä myös välittuun projektityyliin ja sen piirteisiin: se ei välttämättä ole ongelma, vaan kyseisen projektin ominaisuus.

Ongelman tutkiminen jakautuu kolmeen alakysymykseen seuraavasti:

1. Miten dokumenttia käytettiin peliprojektissa ja mihin tarkoitukseen?
2. Mitä vaihtoehtoisia tapoja suunnittelussa käytettiin dokumenttien sijasta?
3. Miten dokumenttien päivittämättömyys ilmeni projektin aikana?

### 4.1 Tutkimusmenetelmän valinta

Tämä tutkielma kartoittaa, miten ja mitä tarkoitusta varten pelisuunnitteludokumentointia tehdään, keskittyen tarkastelemaan prosesseja, jotka johtivat dokumentoinnin syntymiseen tai syntymättä jäämiseen. Tavoitteena on tutkia ilmiötä syvällisesti tilastollisen yleistettävyyden sijasta, minkä vuoksi tutkimusstrategiaksi valikoitui tapaustutkimus. Koska tutkimus pelialasta on vielä alkuvaiheessa, valittiin eksploratiivinen tapaustutkimus: tarkoituksena on tarkastella tapahtumia ja tuottaa ideoita ja hypoteeseja uutta tutkimusta varten (Runeson & Höst, 2009).

Jyväskylä Game Lab (jatkossa: JGL) oli EU-rahoitteinen Jyväskylän yliopiston ja Jyväskylän ammattikorkeakoulun 1.1.2015-31.1.2017 toteuttama hanke, jonka tarkoituksena oli kehittää uusia opetusmenetelmiä pelialan opetukseen yliopistossa ja ammattikorkeakoulussa (JGL, 2016). Projekteihin osallistui ensisijaisesti opiskelijoita ja työttömiä, ja heistä muodostettiin ryhmiä, joiden tarkoituksena oli viiden kuukauden aikana tuottaa pelattava peliprototyyppi. Ryhmille tarjottiin työtilat ja -välineet sekä koulutusta, apua ja tukea pelialan käytäntöjen suhteen, mutta tarkempia vaatimuksia projektityylille tai dokumentoinnille ei ollut.



Koska JGL-ryhmiltä ei vaadittu tiettyä dokumenttia, peliprojektissa käytetty dokumentointi syntyi ryhmien omista tarpeista käsin. Toisaalta ryhmät eivät olleet ammattimaisia pitkän aikaa yhdessä työskennelleitä ryhmiä, joten tämä tutkimus ei kuvaa pelialan yrityksiä, vaan vastikään muodostuneita ryhmiä, joille ei ole vielä syntynyt vakiintuneita käytänteitä.

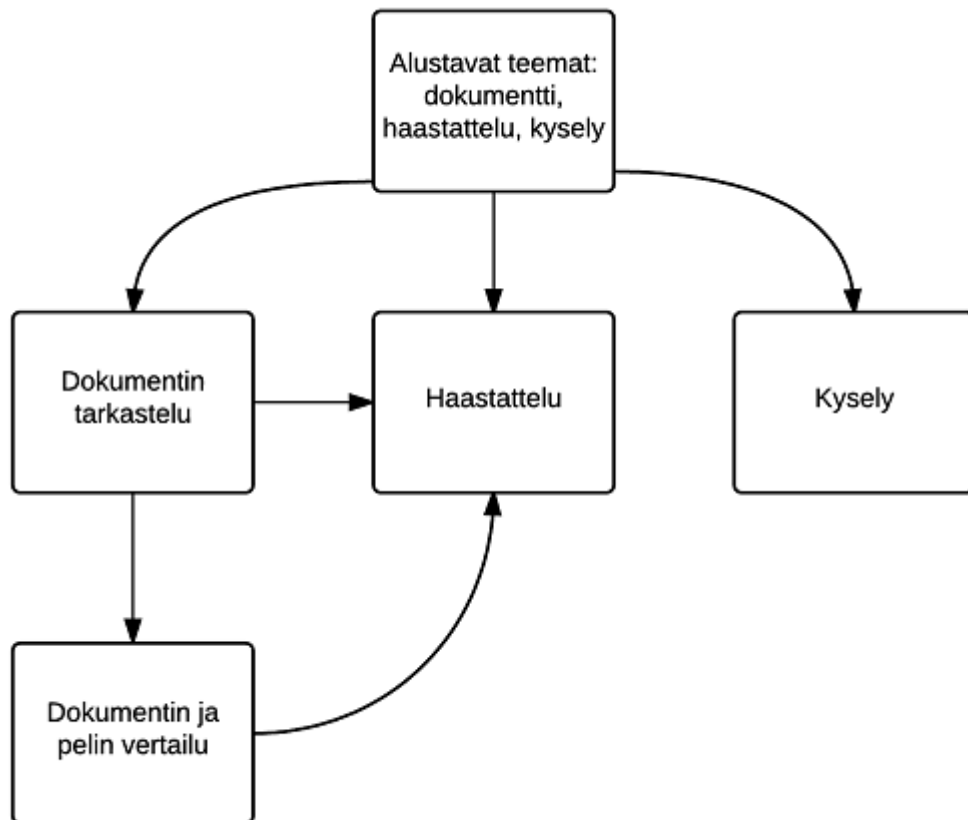
Tämä tutkielma on holistinen monitapaustutkimus, eli useaa samankaltaista tapausta tutkitaan kokonaisuuksina (Yin, 2009). Tapaukset ovat JGL:ssä tammikuussa 2016 aloittaneet kolme ryhmää: Emberlands, Pointy End ja APPrentice. Keväällä 2016 JGL:ssä oli näiden kolmen lisäksi myös kaksi muuta ryhmää, joista toinen aloitti edellisen Game Labin aikana, ja toinen oli aloittanut pelin tekemisen itsenäisesti aiemmin. Pidemmän yhdessä vietetyn ajan voi olettaa vaikuttavan olennaisesti muun muassa kommunikointiin ja työtapoihin, joten lähtökohtien erilaisuuden vuoksi nämä kaksi ryhmää suljettiin tutkimuksesta ulos.

## **4.2 Aineiston keruu**

Tutkittavina kohteina ovat pelisuunnitteludokumenttien lisäksi dokumenttien vertailu valmistuneeseen peliin, dokumenttien kirjoittamisesta päävastuussa olleen henkilön haastattelu sekä kysely muille ryhmän jäsenille. Aineiston keruu valmisteltiin muodostamalla kirjallisuuden avulla alustavat teemat haastattelulle, kyselylle ja dokumenttien tarkastelulle. Aineiston keruussa oli kolme vaihetta: dokumenttien tutkiminen ja niiden vertailu peliprototyyppeihin, haastattelun teemojen tarkentaminen tämän vertailun pohjalta, sekä haastattelun ja kyselyn suorittaminen (kuva 4).

Ensimmäisessä vaiheessa tarkastelin dokumenttien sisältöä kirjallisuudessa esiteltyjen sisältöjen valossa sekä vertailin sitä peliin. Luvussa 3 annetun määritelmän mukaisesti tässä tutkimuksessa pelisuunnitteludokumentiksi määritellään ryhmän jäsenten käyttöön tarkoitettu suunnittelua käsittelevä dokumentti. Pois jätetään siis esimerkiksi yksittäisten henkilöiden itselleen tekemät muistiinpanot jotka eivät ole koko ryhmän saatavissa, sekä projektin hallinnointia käsittelevät dokumentit, kuten projekti- ja markkinointisuunnitelmat.

Dokumentit ja pelit toimitettiin tutkittaviksi toukokuun 2016 alussa. Dokumentin ja pelin vertailu oli kaksiosainen: löytyikö pelistä ominaisuuksia, joita ei ollut mainittu dokumentissa, ja löytyikö dokumenteista ominaisuuksia, joita ei oltu toteutettu peliin. Eroavaisuudet



Kuvio 4. Aineiston keruu

tarkistettiin kirjallisuudessa esiteltyihin sisällysluetteloihin perustuen sen varmistamiseksi, että kaikki olennaiset kategoriat käsitellään.

Toisessa vaiheessa räätälöin pelisuunnittelijoiden haastattelut (liite A). Haastattelun runko sisälsi kolme teemaa: projekti, dokumentointi sekä pelin ja dokumentin erot. Teemat räätälöitiin kullekin ryhmälle erikseen pelistä ja dokumentista tehtyjen havaintojen pohjalta. Tavoitteena oli saada taustatietoa projektimenetelmistä sekä etsiä syitä dokumentin ja pelin eroavaisuuksille saadakseni selville, millä vaihtoehtoisilla tavoilla peliä oli suunniteltu dokumentin lisäksi ja mikä prosessi johti kyseisten erojen esiintymiseen.

Kysely dokumenttien käytämisestä (liite B) osoitettiin ryhmän kaikille jäsenille. Kyselyn tarkoituksena oli varmistaa että tutkimuksessa oletetut taustatiedot pitävät paikkansa, sekä tarjota lisäinformaatiota miten muut ryhmän jäsenet käyttivät dokumenttia. Myös kyselyssä oli kolme teemaa: projekti, dokumentin käyttö sekä muu työskentely.

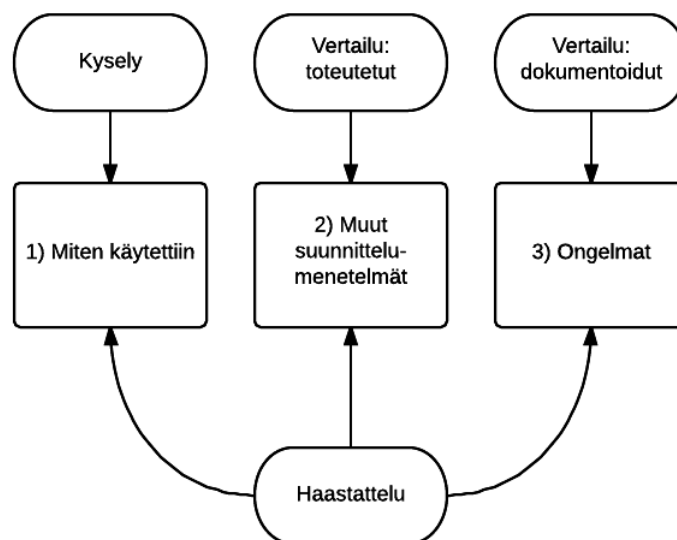
Kolmannessa vaiheessa toteutin haastattelut. Pelisuunnittelijan haastattelu toteutettiin puolistrukturoituna kasvokkain toukokuussa dokumenttien ja pelien analysoinnin jälkeen. Puolistrukturoitu haastattelu valittiin, koska kaikki tärkeät aiheet eivät välttämättä nouse esille pelkästään dokumentin ja pelin perusteella, jolloin puolistrukturoidun haastattelun aikana voidaan vapaammin syventyä aiheisiin jotka vaikuttavat kyseisen ryhmän kohdalla olennaisilta. Haastateltaviksi valittiin ainoastaan dokumenttien kirjoittamisesta päävastuussa olevat, joka oli jokaisessa ryhmässä suunnittelijan roolissa toimiva henkilö. Tämä siksi, että pääsuunnittelija dokumenttien vastuuhenkilönä tietää dokumenttien syntymisestä eniten, ja vain yhtä, asiantuntevinta, henkilöä kustakin ryhmästä haastatteleamalla aineiston määrä pysyy hallittavan kokoisena.

Muiden ryhmän jäsenten mahdollinen panostus dokumentointiin selvitettiin kyselyn avulla, joka toteutettiin nimettömän ryhmäkohtaisen verkkolomakkeen avulla projektin päättymisen jälkeen toukokuun 2016 lopussa. Kyselyihin vastasi Emberlandsilta 7/9 henkilöä, APPrenticelta 3/5, ja Pointy Endiltä 5/6.

### **Analyysi ja tulokset**

Aineisto vastaa tutkimuskysymyksiin seuraavasti (liite 5). Tutkimuskysymykseen 1, miten dokumenttia käytettiin peliprojektissa, saadaan vastaukset pelisuunnittelijan haastattelusta. Kyselystä puolestaan saadaan selville miten muut ryhmän jäsenet käyttivät dokumenttia.

Toiseen ja kolmanteen kysymykseen (vaihtoehtoiset suunnittelutavat ja miten päivittämättömyys ilmeni) saadaan vastaus tarkastelemalla dokumentin ja pelin eroja ja jäljittämällä haastattelun avulla, mikä prosessi johti näiden erojen olemassaoloon. Näin saadaan syvällisemmin selville, mistä syystä kyseiset eroavaisuudet esiintyivät: johtuivatko ne yllättävistä ongelmista vai olivatko ne tiedostettuja päätöksiä. Erot jaetaan kahteen kategoriaan. Syyt, miksi pelissä näkyviä ominaisuuksia ei ole dokumentoitu, antavat tietoa siitä, käytettiinkö jotain muuta keinoa suunnittelussa. Dokumenteissa olevat ominaisuudet, jotka eivät päätyneet peliin asti, antavat tietoa siitä, onko kyseessä dokumentin jäätyminen vai jokin muu syy, ja mistä asioista nämä johtuvat.



Kuvio 5. Miten aineisto vastaa tutkimuskysymyksiin

### 4.3 Validiteetti

Validiteetti jaetaan neljään pääluokkaan: rakenteelliseen, sisäiseen ja ulkoiseen validiteettiin sekä reliabiliteettiin (Runeson & Höst, 2009). Rakenteellinen validiteetti tarkoittaa, edustavatko mittarit todella sitä, mitä ollaan tutkimassa (Yin, 2009). Esimerkiksi, jos haastattelukysymykset tulkitaan eri tavalla, tämä on uhka rakenteelliselle validiteetille. Tätä pyrittiin ehkäisemään erityisesti valitsemalla puolistrukturoitu haastattelu, jotta haastateltavilla on mahdollisuus kysyä lisäkysymyksiä.

Sisäinen validiteetti tarkoittaa kausaliteetin luotettavuutta: olivatko tulokset seurausta niistä tekijöistä kuin oletettiin, vai vaikuttiko niihin joku kolmas tekijä jota ei otettu huomioon (Runeson & Höst, 2009). Tässä tutkimuksessa sisäistä validiteettia vahvistaa se, että tapauksia on useampi kuin yksi. Näin tapausten väliltä voidaan etsiä samankaltaisuuksia. Sisäistä validiteettia parantavat myös erilaiset triangulaation tavat. Tässä tutkimuksessa käytetään metodologista triangulaatiota, eli samaa aihetta käsittelevää informaatiota kerätään useilla eri tavoilla (Yin, 2009): dokumentaatiosta ja pelistä tehdyt havainnot varmistetaan suunnitteli-jahaastattelusta, ja haastatteluvastauksia tukemassa ovat myös tiimin muiden jäsenten näkemykset kyselyn kautta.

Ulkoinen validiteetti tarkoittaa tutkimuksen yleistymistä tapausten ulkopuolelle. Tämä tut-

kimus ei vähäisen tapausmäärän vuoksi ole tilastollisesti yleistettävissä, mutta se voi yleistyä analyyttisesti (Yin, 2009), eli tässä tutkimuksessa tuotettuja havaintoja voidaan kokeilla muiden samankaltaisten tapausten analysoimisessa. Reliabiliteetti tarkoittaa tutkimuksen toistettavuutta: saisiko eri tutkija samat tulokset (Runeson & Höst, 2009). Tätä on pyritty varmistamaan kuvailemalla tutkimuksen kulku edellisissä luvuissa, jotta tutkimuksen voisi toistaa mahdollisimman tarkasti.

## **Rajoitteet**

Tapaustutkimuksen yhtenä rajoitteena on se, edustaako tapaus riittävästi tutkittavaa ilmiötä. JGL-ryhmien tapauksessa on otettava huomioon seuraavat seikat.

JGL oli palkaton opiskelijoille ja työttömille suunnattu projekti johon osallistujat käyttivät aikaa oman kiinnostuksensa ja muiden sitoumusten puitteissa, eikä suurin osa työskennellyt kokoaikaisesti (liite 12), mikä todennäköisesti rajoitti kaikkien paikallaoloa yhtä aikaa ja siten mahdollisesti vaikutti kommunikaatioon.

Tiimit olivat varsin pienet, projektin alkaessa 6-9 henkilöä. Projektin aikaväli oli alle viisi kuukautta, ja varsinaiseen julkaisuun ei tähdätty. Osallistujat olivat melko kokemattomia pelialalla (liite 10) sekä toisilleen tuntemattomia ja ryhmäytyivät vasta projektin aikana. Siten tutkimustulokset todennäköisesti eroaisivat esimerkiksi peliyrityksistä, joilla saattaa olla suurempi tiimi, formaalimmat projektikäytänteet, enemmän kokemusta ja enemmän aikaa käytettävissä.

Suunnittelupäätöksiä saattaa syntyä myös kokouksissa sekä epäformaaleissa keskusteluissa, mitkä on mahdollista huomioida vain suoralla havainnoinnilla. Epäformaalien keskustelujen seuraamiseksi tarvittaisiin jatkuvaa seuraamista, joten tapahtumien ennakoimattomuuden sekä ajanpuutteen vuoksi suoraa havainnointia ei kuitenkaan ollut tämän tutkimuksen puitteis-  
sa mahdollista toteuttaa. Siksi nojataan ainoastaan toissijaisiin lähteisiin haastattelun kautta, joissa pyritään kattamaan suunnitteluprosessia kokonaisuutena.

Haastattelujen ongelmana voi olla haastateltavan refleksiivisyys, eli totuuden sijasta yritetään vaikuttaa hyvältä (Yin, 2009). Tässä tutkimuksessa se saattaa vaikuttaa siten, että dokumentteja saatetaan tehdä tai viimeistellä tutkimusta silmälläpitäen ja haastatteluun ja kyselyyn

saatetaan vastata refleksiivisesti. Tätä pyrittiin ehkäisemään korostamalla haastattelussa, että erilaisilla dokumenttityyleillä ei ole paremmuusjärjestystä. Lisäksi kyselyyn vastaaminen tapahtui nimettömästi, jolloin mahdollisiin sensitiivisiin kysymyksiin voidaan vastata rehellisemmin. Kolmanneksi JGL-projekteissa refleksiivisyys saattaa vaikuttaa myös siinä, että projektin koulutuksellinen tavoite ohjaa liikaa projekteja, ja näin se ei vastaa tilannetta jossa ensisijaisena tavoitteena on tuottaa peli.

Tutkijan puolueellisuus voi vaikuttaa tutkimuskysymysten suunnitteluun. Erityisen kriittinen puolueellisuuden ongelma on tässä tutkielmassa siksi, koska toimin itse yhdessä ryhmässä (Pointy End) kenttäsuunnittelijana ja ohjelmoijana, ja siten kyseisen ryhmän dokumentit ja peli olivat minulle tutumpia kuin muiden ryhmien. Puolueellisuutta pyritään ehkäisemään johtamalla päättelyketjut suoraan kirjallisuudesta sekä haastateltavien vastauksista sekä mainitsemalla analyysissä eksplisiittisesti kohdat, joihin minun läsnäoloni projektissa on saattanut vaikuttaa.

## 5 Tutkimuskohteet

Jyväskylä Game Lab (JGL) oli Jyväskylän yliopiston ja Jyväskylän ammattikorkeakoulun toteuttama hanke, jossa kehitettiin opetusmenetelmiä Jyväskylän korkeakoulujen peliopetusta varten (JGL, 2016). Osallistujista muodostettiin tiimit, jotka kehittivät viiden kuukauden aikana pelin, hankkeen henkilökunnan tuella. Koulutusta ja ohjausta oli tarvittaessa saatavilla, mutta hankkeen puolesta ei vaadittu mitään tiettyjä käytänteitä dokumentointiin liittyen.

Tammikuussa 2016 JGL:ssä aloitti 20 osallistujaa, jotka aloittivat kehittämällä peli-ideoita. Näistä ideoista projektin henkilökunta valitsi kolme toteutuskelpoisinta ja osallistujat jaettiin kolmeen ryhmään kehittäjäroolien sekä kiinnostuksen perusteella. Tämän jälkeen tiimit saivat melko vapaasti muodostaa peliprojektinsa aikataulut ja muut projektinhallinnolliset sekä pelisuunnittelulliset seikat. Näin syntyi kolme ryhmää, joissa oli kussakin 6-9 henkilöä. Osa ryhmistä tähtäsi projektin jatkamiseen JGL:n jälkeen, joten pelin prototyyppi toimi osittain välituloksena, mikä on tulosten analysoinnissa huomioitu.

### 5.1 Pointy End

Pointy End (kuva 6) on sivusta kuvattu taistelu-tasohyppely-moninpeli, jossa pelihahmot taistelevat toisiaan vastaan miekoilla, keihäillä ja muilla vastaavilla aseilla. Pelihahmot kuolevat joko toisten pelaajien aseisiin, kentässä oleviin ansoihin tai kentästä pudottuaan, ja voittajan selvittyä siirrytään uuteen kenttään kaikilla pelaajilla. Tarinaa ei ole lainkaan; teemana on fantasiatyypinen keskiaika. Toukokuun 2016 alkupuolella tuotetussa prototyyppissä oli valmiina neljä kenttää ja pelimekaniikat, eli hahmojen liikkuminen sekä taistelu. Ryhmän tavoitteena oli luoda toimiva pelidemonstraatio JGL:n aikana ja jatkosuunnitelmia ei ollut.

### 5.2 Emberlands

Emberlands (kuva 7) on sivusta kuvattu tasohyppely-tarinapeli, jossa pelaajahahmo kulkee pelimaailmassa ratkoen kenttämekaniikoista muodostuvia ongelmanratkaisutehtäviä sekä tutustuen pelimaailmaan. Emberlands on vahvasti tarinavetoinen: tapahtumapaikkana on post-



Kuvio 6. Pointy End

apokalyptinen fantasiamaailma, jossa päähahmo etsii päävihollista ja löytää samalla vihjeitä maailman historiasta. Tarina ilmaistaan dialogin sekä pelistä löytyvien muistiinpanojen avulla pelaajan kulkiessa pelimaailmassa. Emberlandsin tavoitteena oli jatkaa peliä myös JGL:n jälkeen, joten pelisuunnitelma koski laajempaa kokonaisuutta, ja toukokuussa tuotetun prototyypin tavoitteena oli lähinnä esitellä tärkeimpiä pelimekaniikkoja kuten kentässä liikkumista ja kenttämekaniikkoja.

### 5.3 APPrentice

APPrentice on vuoropohjainen mobiililaitteella toimiva pulmapeli. Pelissä on kaksi eri kenttää: 8x8 -ruudukosta (kuva 8) etsitään samanvärisiä palloja samalta riviltä. Yhdistettyjen pallojen väri määrää loitsun tyyppin ja määrä loitsun voimakkuuden. Kolmen loitsuvalinnan jälkeen siirrytään taistelemaan taistelukenttään, jossa valitut loitsut suoritetaan (kuva 9) tekoälyn avulla toimivaa velhoa vastaan. Tätä peliä oli suunniteltu ennen JGL:ää, mutta lähinnä ideatasolla.





Kuvio 7. Emberlands



Kuvio 8. APPrenticen ruudukkonäkymä



Kuvio 9. APPrenticen taistelunäkymä

## 6 Tulokset

Tässä kappaleessa esitellään tutkimuksen tulokset. Ensin käsitellään tutkimuskysymys 1, miten dokumenttien tuotantoprosessi peliprojektiryhmissä tapahtui. Kaksi seuraavaa lukua käsittelevät dokumentoinnin päivittämistä tarkastelemalla kahta kategoriaa: mitkä dokumentoidut ominaisuudet eivät päätyneet peliin ja mitkä toteutetut ominaisuudet eivät olleet dokumenteissa.

Kysymys, mitkä toteutetut ominaisuudet eivät ole dokumenteissa, kertoo minkä asioiden dokumentointia ei pidetty tarpeellisena ja/tai niitä suunniteltiin jollain muilla tavoin. Luku 6.2 käsittelee tutkimuskysymystä 2, millä vaihtoehtoisilla tavoilla suunnittelua tehtiin.

Kysymys, mitkä dokumentoidut ominaisuudet eivät ole pelissä, kertoo miksi niitä ei ole toteutettu ja oliko se tietoinen päätös. Mikäli päätös oli tiedostamaton eikä esimerkiksi aikataulutukseen liittyvä, se saattaa olla ongelma. Ongelmana voi olla, että dokumentointi on jäänyt, eli sitä lakattiin päivittämisestä vaikka suunnitelmat muuttuivat. Tällöin voidaan vahingossa implementoida jotain, jota ei ollut tarkoitus implementoida, mikäli kaikki eivät ole tietoisia suunnitelmien muuttumisesta. Luku 6.3 käsittelee tutkimuskysymystä kolme, tapahtuiko dokumentissa jääymistä.

### 6.1 Dokumentin käyttäminen

Dokumentin käyttöön vaikuttaa omaksuttu projektityyli. JGL-ryhmissä projektityylien formaalia omaksumista vaikeuttaviksi asioiksi koettiin muun muassa tiukka aikataulu, ryhmä, jossa osallistujat ovat toisilleen tuntemattomia sekä kokemattomuus projekteista.

*Siihen emme kovin syvälle lähteneet, että rupeamme soveltamaan mitään tekstikirjasta opittua mallia tähän, koska käytännössä puhutaan nimenomaan protosta ja se aikataulu enemmän sanelee että miten kannattaa lähtä asioita tekemään.*  
— APPrentice

*Meillä ei ollu semmosta [projektityyliä], me käytännössä kokeiltiin paljon erilaisia juttuja. Ensin kun oli tuntematon tiimi [...], niin mun lähtökohta oli se että*

*yritin saada ymmärryksen siitä että miten kukin työskentelee henkilökohtaisesti, ja se on ollut kokoajan semmonen prosessi että mitä oon käynyt läpi, oppinu paremmin toisten työtavat.*

— Emberlands

Pointy End kuitenkin tiedosti projektityylien käyttämisen mahdollisuuden, vaikka ei välttämättä toteuttanutkaan sitä kovin formaalisti:

*Ei varsinaisesti [käytetty projektityylejä], koska mullahan ei aikaisemmin projektijohtamisen kokemusta oo ja menin ite vähä sen mukaan mitä pientä vinkkiä [projektin henkilökunnalta] tuli. Mutta käytännössä se on jonkinlainen ketterien metodien sovellus.*

— Pointy End

Formaaleja projektityylejä ei siis noudatettu, mutta erilaisten projektityylien piirteitä oli kuitenkin nähtävissä. Luvun 3.1 mukaisesti ketterän projektin dokumentointiin liittyviä piirteitä ovat tuotannon vaiheiden esiintyminen yhtä aikaa, dokumenttien päivittyminen koko projektin ajan, iteratiivisuus, sekä dokumentoinnin vähäisyys painottaen käytännön työtä.

Emberlandsilla peli-ideaan tuli useita suuria muutoksia. Tällöin suunnittelua tehtiin koko projektin ajan suurella prioriteetilla.

*Preproductionin ideointi ja suunnittelu pysyy aika lailla 50-50 käsi kädessä tuon productionin kanssa. [...] preproduction vaihe on tullu uudestaan ja uudestaan sitte ku se projekti on muuttunu. [...] Tää on ollut just aikalailla kuukauenen mittainen sykli meillä tän osalta. [...] se success-testaus projektin lopussa että mitkä oli onnistumiset ja epäonnistumiset, niin meillä se on tullu jo tän prosessin aikana.*

— Emberlands

Emberlandsin projekti siis vaikuttaa olevan ketterien menetelmien mukainen, vastaten muutoksiin ja sisältäen syklissä myös testauksen.

Adaptiivisten projektimallien mukaista vaiheiden päällekkäisyyttä (kuva 3) voi soveltaa JGL-

ryhmiin vain osittain: konseptivaihe sekä jälkituotantovaihe puuttuivat käytännössä kokonaan. Konseptivaihe, eli useiden peli-ideoiden tuottaminen, oli järjestetty hankkeen henkilökunnan puolesta, ja kaikkien osallistujien tuottamista ideoista he valitsivat toteutettavat ideat ja muodostivat tiimien kokoonpanon ideoiden perusteella. APPrentice ja Emberlands eivät siirtyneet jälkituotantovaiheeseen lainkaan, Pointy End puolestaan keskittyi viimeistelyyn yhden viikon ajan viimeistä esittelyä varten. Näin JGL-projektien ajanjaksolle sijoittuivat siis ainoastaan esituotanto- ja tuotantovaiheet.

Yhteinen piirre kaikille ryhmille projektityylin suhteen oli, että suunnittelua ja peliä tehtiin yhtäaikaan, eli kyse ei ollut vesiputousmallista, jossa vaiheet ovat peräkkäisiä, ensin suunnitellaan ja sitten tuotetaan.

*Matkan varrella tapahtuu sitä suunnittelutyötäkin jonkin verran. Pääpaino on ihan samaan tapaan ku tuossa tuotannossa tottakai, mutta siinä vilahtaa aina sillon tällön justuinsa semmosia pieniä juttuja jotka kehittää sitä peli-ideaakin.*

— APPrentice

Pointy Endillä kuitenkin ydinominaisuuksia lyötiin lukkoon jo projektin puolessa välissä peliesittelyä varten, mutta heilläkään suunnittelu ei loppunut täysin.

*[Suunnittelu jatkui koko projektin ajan] enemmän tai vähemmän. [...] pelimekaniikka on pikkusen muuttunu, mutta pelimekaniikallahan ne isot kuviot lyötiin aika pitkälle lukkoon sillon ku se gathering oli. [...] Key featureita myöskin lyötiin lukkoon siinä vaiheessa, viimeistään gatheringissa pelaajien määrää, miten ohjataan, tällasta. Suunnittelua ei oo sen jälkeen paljo ollu.*

— Pointy End

Tällainen lähestymistapa muistuttaa hybridimallia: joillain osa-alueilla suunnitteluvaihe lopui ja siten vesiputousmallin vaiheiden peräkkäisyyttä voi osittain soveltaa tämän projektin tarkasteluun.

Mikäli projektin tavoitteena on elävä dokumentti, eli suunnitelmat muuttuvat ja dokumentit sen myötä, ongelmana voi olla jäätyminen eli dokumenttia ei päivitetä johdonmukaisesti. Tämä laskee dokumentin tarkkuutta ja voi johtaa dokumentin käyttövaikeuteen ja vanhen-

tuneen tiedon implementointiin. Kaikilla ryhmillä dokumentin päivittymisaktiivisuus laski projektin edetessä, ja suunnittelu siirtyi osittain muihin kanaviin. Dokumentoinnin loppumista käsittelevään kysymykseen Pointy End vastasi seuraavasti:

*Helmikuun ajan tuli päivitettyä vielä ihan hyvin. Maaliskuussa se alko pikku hiljaa hiljenee [...]. Kenttiä on päivitetty sen jälkeen, mutta pelisuunnitelmia, design suunnitelmia, asesuunnitelmia ei oo tullu päivitettyä koska ne oli niitä alkuperäisiä suunnitelmia mitä ei voinu toteuttaa monissa asioissa, niin ei oo sitte vaan välittäny tai muistanu korjata niitä.*

— Pointy End

Tällainen tapaus siis on jäänyt dokumentti, koska uusia suunnitelmia ei päivitetty dokumentteihin.

Myös Emberlandsilla dokumentointi väheni loppua kohden, huolimatta siitä, että suunnitelmat muuttuivat. Dokumentointi nähtiin ylimääräisenä vaiheena joka lisää työmäärää turhaan, käytännön työn ollessa hyödyllisempää.

*Se että dokumentoiaan jos se saattaa vielä muuttua, ei olla nähty tarvetta sille tässä vaiheessa enää, ku se konsepti on käytännössä muuttunu aikasemmin neljä kertaa ja sitä päiviteltiin. [...] Tullu enemmän se käytännöllinen homma ja joutunu jättämään sen dokumentoinnin sivuun siitä. [...] Eli yksinkertaisesti isoin syy miksi ei oo dokumentointia tällä hetkellä niin on ollu se ajankäyttö. [...] se hyödyllisyys siitä niin ei oo nähty sitä semmosena että sitä kannattaa tehdä.*

— Emberlands

Mitä dokumentoinnin päivittämättömyys sitten käytännössä tarkoittaa? APPrentice huomautti, että vaikka tekstidokumentteja ei päivitetty koko projektin ajan, varsinainen suunnittelu ei loppunut, vaan siirtyi konseptitaiteena suunnitteluun sekä Flowdock-keskusteluohjelmaan.

*Käytännössä se ideoitten välittäminen toisilleen [ei] loppunu vaikka siitä ei varsinaisesti tehtykään mitään dokkareita. Käytännössä monia asioita esittelin tiimille grafikan ja animaatioiden kautta, että “tämmönen olis tää ui-model” ja ”tällä tavalla objektien pitäisi liikkua”.*

— APPrentice

Myös muilla ryhmillä oli havaittavissa siirtymistä muihin kanaviin dokumentoinnin sijasta. Emberlands teki suunnittelua valkotalulla ja keskustelemalla.

*Siinä on ollu niin paljon näitä muuttuvia tekijöitä, niin ja se suunnittelu on ollu enemmänkin, sitä on puhtaasti suunniteltu keskustellen enemmän ja konseptointia siinä meidän fläppitalulla*

— Emberlands

Pointy End siirtyi Flowdock-keskusteluohjelmaan:

*Flowdock tais aktivoitua pikkuhiljaa, ihmiset alko käyttää sitä enemmän projektin edetessä. Mutta sitte Trello oli ehkä semmonen ku dokumentteja ei käytetty niin itekki annoin sen Trelon kanssa vähän periksi, en päivittänyt sitä enää sen jälkeen paljo sitäkään. [Kasvokkainen kommunikaatio] pysy aika samana [dokumentoinnin vähentyessä], koska enimmäkseen tuli kerättyä dokumentteihin asioita joita oltiin sovittu kasvokkain tai jossain muissa kanavissa. Se toimi muistiinpanovälineinä siinä mielessä.*

— Pointy End

APPrenticella ryhmän koon koettiin vaikuttaneen kommunikointivalintoihin.

*Alkuun piettiin yhteyttä sähköpostilla, sitte otettiin Flowdock käyttöön, koska tarve nopeammalle, reaaliaikaiselle kommunikoinnille kasvo. Ja yritettiin käyttää Trelloa ku sitä meille [henkilökunta] suositteli, mutta sitten siitä huomaa taas sen että se on sitä parempi työkalu mitä suurempi tiimi on. Sitte ku alkaa tosiaanki mennä alle viiden hengen tiimiin, [...] niin voi tulla hirmu paljon sitä hukkatyötä, kun loppujen lopuksi monesti kyseessä oli asioita joista riittää että vaan sanoo vaikka siellä Flowdockissa tai nähdessä että tämmönen pitäis saaha tähän kohtaan tehtyä.*

— APPrentice

Flowdockin suosio näkyy myös kyselyn tuloksissa: yhteensä 11/15 kyselyn vastaajista käytti

Flowdockia joka päivä (kuva 11). Kuviossa 13 näkyy, että viimeistelyvaiheessa luettiin kokonaisuudessaan vähiten dokumentteja. Tämä vastaa myös tutkimustuloksiin joiden mukaan dokumentinkäyttöaktiivisuus laskee tuotannon edetessä.

Vaikka muitakin kommunikointikanavia käytettiin, dokumentoinnin hyödyllisyys nähtiin kommunikointivälineenä:

*Tottakai se ku kirjottaa sen paperin niin selvittää peli-idea jo itsellekin. Mutta tottakai eniten muille. Että vähentää sitä tarvetta olla kokoajan kertomassa ihmisille että seuraavaksi tämmöstä ja tämmöstä. Että on kuitenkin semmonen peruskuvaus siitä että mikä se homma on, että siihen pystyy palaamaan itsekin aina jos tulee tyhjää ja miettii että mitä seuraavaksi.*

— APPrentice

*Kyllä se tämmöseen tiimin sisäseen viestintään ja kommunikaatioon, että niitä suunnitelmia kirjoitetaan ylös. Että jos tuntuu että pitää varmistaa jotain ja nousee jotain ongelmia niin sen voi dokumentoida ja sitä voi pohtia ja se on kirjoitettuna auki se ratkasuki siihen.*

— Pointy End

Dokumentteja luettiin kaikissa ryhmissä melko suurella prosenttiosuudella (kuva 14). Kyselyn perusteella dokumentit eivät kuitenkaan olleet suosituin kanava suunnittelupäätösten tarkistamiseen, vaan keskustelu, kokoukset ja online-kanavat olivat tässä tarkoituksessa suosituimpia (kuva 15).

Yhteenvedon voisi sanoa, että ajan kuluessa dokumentin päivittymisaktiivisuus laski ja suunnittelu siirtyi muihin kanaviin. Piirteitä, jotka vaikuttivat formaalien projektikäytänteiden epäselkeyteen, oli kolme: 1) projektiryhmän kokemattomuus esti formaalien käytänteiden omaksumisen ja hyödyntämisen, 2) projektiryhmän pieni koko mahdollisti suoran kommunikaation, 3) projektin lyhyt aikaväli leikkasi osan vaiheista pois. Ketterien menetelmien piirteitä oli suunnittelun jatkuminen koko projektin ajan, sekä Emberlandsilla kuukauden mittaiset suunnittelun syklit, kun taas Pointy Endillä oli nähtävissä myös hybridityyppistä vaihteistusta.



Dokumenttien päivittämisen vähenemiseen vaikuttavat tekijät olivat 1) siirtyminen käytännön työhön, 2) suunnittelu muissa kanavissa kuten konseptitaiteena, keskusteluna kasvokkain sekä keskustelusovelluksissa, 3) alkuperäisten suunnitelmien epäkäytännöllisyys.

## 6.2 Dokumentoimatta jääneet toteutetut ominaisuudet

Joskus peliin toteutetaan ominaisuuksia, joita ei kuitenkaan kirjattu suunnitelmiin missään vaiheessa tai kirjattiin hyvin yleistasolta. Suunnitelmia siis tehdään samalla kuin itse peliä tehdään: peliä suunnitellaan lennosta. Tämä voi johtaa suunnitelmien vanhentumiseen, mikä puolestaan voi johtaa vaikeuteen käyttää dokumenttia ja tietää, mitkä suunnitelmat ovat vanhentuneita. Toisaalta voi olla, että suunnittelu on tehty dokumenttien sijasta muilla keinoin.

JGL-ryhmissä se, että suunnittelua tehdään muuten kuin dokumentoinnin avulla, oli erityisen suuressa roolissa grafiikan ja äänisuunnittelun kohdalla. Konseptitaidetta tehtiin kaikissa ryhmissä ja erityisesti Emberlandsilla oli paljon inspiraatiografiikkaa, eli grafiikkaa jonka tarkoituksena ei ole päätyä peliin vaan toimia inspiraationa kehittäjille.

Pointy Endillä ja APPrenticella tekstuaalinen dokumentointi pelin ulkonäön suhteen oli vähäistä, esimerkiksi APPrenticella oli lyhyesti kuvailtu tunnelmaa jota grafiikalla ja äänillä tavoitellaan.

*Audiovisuaaliset mallit: Square Enixin ysäri-JRPG:t(Chrono Trigger, Final Fantasyt, Secret of Manat, yms.)*

*+ retrofilis*

— APPrentice: APPrentice Bible 0.95

Pointy End kuvaili tematiikkaa seuraavasti:

*Temaattisesti peli sijoittuu löyhästi Euroopan 1300-1600 -luvulle: myöhäiskeskiajasta renessanssiin. Periodille ominaista oli aseiden kirjo, jossa varhaisia tuliaseita saatettiin käyttää erilaisten keskiajalta periytyneiden teräksisten aseiden kanssa.*

— Pointy End: Pelisuunnitelma

Valmistuneessa pelissä kuitenkin oli myös fantasiaelementtejä, kuten helvettiä kuvaava tulinen kenttä, ja tätä teeman muutosta ei kuvattu missään. APPrenticella ja Pointy Endillä siis pysyttiin hyvin yleistasolla visuaalisesta tyylistä puhuttaessa. Tämä poikkesi Emberlandsista, jossa kuvailtiin hyvinkin yksityiskohtaisesti kuvien yhteydessä vaikutelmaa, mikä esimerkiksi hahmoista pitäisi syntyä. Päävihollista kuvailtiin seuraavasti:

- *Must look scary and powerful*
  - *Must have a massive weapon that could look like a Scythe*
  - *Must be mysterious and have some Red Spirit*
- Emberlands: Game Design Document

Pointy End ja APPrentice kertoivat suunnitelleensa grafiikkaa lähinnä konseptitaiteena sekä keskustellen.

*Palaverissa siitä kyllä puhuttiin justissa, ja se mikä mulla on alunperinki ollu tämä visio, tämä ois retrohenkinen, tämä käyttäis pixel arttia.*

— APPrentice

*Suullista suunnittelua oli. Mä saatoin laittaa joistain jutuista jotain Paint havainnekuvia, jos mä halusin jotain tiettyä jonkin kentän grafiikan suhteen. Me puhuttiin siitä käyttöliittymästä, niin siitähän mä laitoin sen painttiluonnoksen näyttille. Siinä mielessä oli jotain pieniä juttuja, mutta varsinaista dokumentaatiota ei ollu että mitä tehdään, miten tehdään.*

— Pointy End

Konseptitaiteen ja keskustelun yhdistelmä saattoi olla hyvinkin moniportainen.

*[Graafikko] suunnitteli miltä se hahmo concept artina näyttää, ja niitä käytettiin pohjana sille pikseliukolle. [...] Siinä keskusteltiin alussa tekijöitten kans että millasia [pelihahmot] on persooniltaan, ja millasia ne eleet on. [...] Ja sitten sen idean pohjalta hän loi sen hahmon, ja itse taas sitte yhdessä sovittujen seikkojen ja luonnekuvausten pohjalta tein sitä animaatiota.*

— APPrentice

Äänisuunnittelu poikkesi grafiikasta siten, että siitä ei ollut dokumentaatiota lainkaan missään ryhmässä. APPrepticella syynä oli se että kyseistä osa-aluetta oli tekemässä vain yksi henkilö, jolloin dokumentointia ei koettu tarpeelliseksi:

*Työnjaollisista syistä, että muu tiimi ei periaatteessa kovin paljo tarvinnu tietää että millasta audiota tarvitaan koska se oli kokoajan minun itseni pöydällä audioitten tekeminen.*

— APPretrice

Sama päti grafiikan ja äänisuunnittelun suhteen Pointy Endillä, joka rajoitti kommunikoinnin pelkkään ilmoitukseen siitä, mitä ääniä tai grafiikkaa on valmistunut.

*Yksi äänisuunnittelija ja yksi graafikko käytännössä, että aika pitkälle se meni sitte että se vaan pukas sitä materiaalia driveen, mistä tietysti näkee suoraan jos on tullu jotain. Sitte johonki kommunikaatiokanavaan yleensä viesti siitä mitä on tehty.*

— Pointy End

Kuten luvussa 3.3 todettiin, dokumentoimattomuus ja yhden henkilön vastuulla oleminen voi olla ongelma suuremmissa projekteissa, mikäli jonkun täytyy jatkaa jonkun muun tekemää työtä. Osa Petrillon tutkimuksen tiimeistä kuitenkin oli huomattavasti suurempia kuin JGL-projektit, 5-60 henkilöä, mikä mahdollisesti kasvattaa tarvetta dokumentaatiolle jos päivittäistä kahdenvälistä kommunikaatiota ei ole samalla tavalla mahdollista hyödyntää.

Äänisuunnittelijat kuitenkin voivat käyttää dokumentointia sekä muita suunnittelukanavia inspiraation lähteenä, kuten Emberlandsin tapauksessa:

*Musiikkisuunnittelu on menny enimmäkseen sillä pohjalla, että se on ollu [äänisuunnittelijan] vastuulla aikalailla täysin. [...] [Äänisuunnittelija] on lukenu niitä tarinajuttuja, se on kattonu meiän arttia etenkin paljon, mä oon käyny paljon keskusteluja siitä kokonaisuudesta, etenki siitä mitä filistä haetaan mihin alueeseen. Sitä ei oo dokumentoitu meillä sitä. [...] Se on niissä taululla tehdyissä piirustuksissa ja suunnitteluvaiheessa, niissä on käyty läpi sitä filistä.*

— Emberlands

Joskus voi olla tarpeellista tehdä ratkaisu suoraan kehitystyön yhteydessä, sen sijaan että ensin käytettäisiin aikaa dokumentointiin ja sen jälkeen toteutettaisiin. Pointy Endin pelissä oli paljon yksityiskohtia, joita ei dokumenteissa ollut lainkaan, esimerkiksi miten asetet reagoivat ympäristön elementteihin tai pelaajan liikkumiseen. Suunnittelupäätös siis tehtiin toteuttamistyön yhteydessä ja dokumentteja ei päivitetty tämän jälkeen.

*Ne on varmaan aika moni semmosia että osa on tullu testatessa vastaan - tässä on ongelma, tälle pitäis tehdä jotain, [ohjelmoijan] kanssa puhunu siitä, tehty päätös suoraan ja ei oo dokumentoitu mitenkään. Jotkut on vaan semmosia että joku on kysynyt että voidaanko tehdä tällöinen - "joo, tehdään" - ei sen kummempaa pohdintaa sitte asiasta.*

— Pointy End

Tällaista tapausta kutsutaan lennosta suunnitteluksi. Myös aikataulun rajoitteet voivat aiheuttaa tätä: APPrepticella tekoäly koettiin olennaiseksi kokonaisuuden kannalta mutta sitä ei ehditty suunnitella etukäteen, joten suunnittelu annettiin tehtäväksi ohjelmoinnin yhteydessä.

*Olisin tykännyt antaa semmonen lista miten sen AI:n pitäis tiettyissä tilanteissa reagoida, ja muutenki ohjeistaa mallia että sillä on tietty tarvehierarkia, aina hahmosta riippuen. Mutta siinä jouduin antamaan koodareille ihan vapaata kättä että aikataulu aika tiukille, oli tärkeetä saada ylipäänsä jonkinlainen AI.*

— APPreptic

Toisaalta voi olla, että tiedetään muutoksia olevan tulossa, joten säästetään aikaa tarpeettomalta dokumentoinnilta ja siirretään se testauksen jälkeen dokumentoitavaksi.

*Syy minkä takia [tehtävien suorittamisen aikarajoitteet] ei oo dokumenteissa on se että niitä ei oo dokumentoitu tän tietyn vaiheen jälkeen. [...] on ollut tarkoitus että se siinä testivaiheessa sitten katotaan. Eli nyt kun me saahaan tuo demo placeholdaus versio rakennettua täysin, niin se että se kerkee [suorittaa ongelmanratkaisutehtävän], se on tullu testauksen kautta se, tommoset määritteet. Elikkä esimerkiksi kuinka korkeelle hahmo hyppää, miten korkeelle tuplahyppy*

*hyppää, miten pitkälle se hyppää... [...] Niitä ei oo sen takia dokumentoitu erikseen koska me on aateltu että me dokumentoidaan sitten ne mitkä on että “okei nyt löyty se oikee fiilis”, niin se dokumentoidaan vasta. Säästetty just sitä turhaa työtä ja aikaa siitä dokumentoinnilta.*

— Emberlands

Tämä havainnollistaa, että dokumentointi saattaa olla ylimääräinen jälkivaihe: ensin muutetaan peliä ja sitten muutetaan dokumentti vastaamaan peliä. Muista kahdesta ryhmästä poiketen Emberlands kuitenkin aikoi jatkaa pelin kehitystä myös JGL:n jälkeen, jolloin dokumentoinnilla oli myös pitkän tähtäimen tavoite, ja heidän tavoitteenaan oli dokumentoinnin valmius ja jo tehtyjen asioiden dokumentointi.

*Idea on just että noi kaikki dokumentoidaan sitte ku saahaan se greenlight-demo valmiiksi. Sit käydään tarkka detaljoitu dokumentointi kaikelle.*

— Emberlands

Suurimmat syyt dokumentoimatta jättämiseen olivat seuraavat: suunnittelua tehtiin muuten kuin dokumentoinnin avulla (konseptitaide, keskustelu) erityisesti resurssien kohdalla, vain yksi henkilö oli vastuussa sekä suunnittelua tehtiin lennosta, joka tapahtui erityisesti aikarajoitteen vuoksi.

### **6.3 Pelistä puuttuvat dokumentoidut ominaisuudet**

Jäätynyt dokumentti tarkoittaa, että dokumenteissa on ominaisuuksia jotka eivät päätyneet peliin. Taustaksi toteuttamatta jäämiselle kysyn, miksi niitä ei ole toteutettu. Tämä on tärkeää siksi, että saadaan tietää olivatko toteuttamatta jättämiset tietoisia päätöksiä esimerkiksi aikarajoitteeseen liittyen.

Yhtenä syynä jättää ominaisuuksia toteuttamatta suuressa roolissa olivat JGL:n aikarajoitteen vuoksi asiat, jotka eivät ehtineet JGL:n aikana tehtyyn prototyyppiin, mutta ne oli tarkoitus toteuttaa kokonaisempaan versioon.

*Aika paljon on semmosta mitä ei oo varsinaisesti suunnitelmista jätetty pois, mutta aika paljon juttuja on sisällytetty semmoseen kokonaisempaan versioon.*

*Että ne on tiputettu vaan protosta. [...] siinä uusimmassa versiossa jota jossain vaiheessa lähin taas kehittelemään, niin [in-game help] on täysin pöydällä.*

— APPrentice

Myös Emberlandsin tavoite oli jatkaa projektia myös JGL:n jälkeen ja pelisuunnitelmaan tapahtui useita suuria muutoksia. Näin tutkittavaksi toimitettu peli oli lähinnä mekaniikkademonstraatio, josta puuttui lukuisia dokumentoituja asioita, mutta itse suunnitelma toteuttaa dokumentoidut ominaisuudet oli pysynyt samana. Esimerkiksi mekaniikoista taitojen saavuttaminen pelin aikana:

*[Mekaniikat] on justuinsa ne kun mä tein sen päätöksen, [...] nää mekaniikat, ja tässä on säännöt ja piste. Ne on ollu alusta lähtien ja ne on edelleenki. [...] Siinä mekaniikkademosssa puuttuu se skriptaaminen siltä osin että missä vaiheessa sä saat [taidot]. Mutta tähän meidän greenlight demoon tulee skriptattuna se, että aluks pelaaja alottaa pelaamisen, niin se ei pysty tuplahyppään. Sen pitää ensin löytää se tavara jonka jälkeen se saa tuplahypyn.*

— Emberlands

Toteuttamatta jättäminen saattoi johtua siitä, että projektin hallinnoinnissa tai pelin toteuttamisessa oli ongelmia ja ennakoimattomia muutoksia.

*Meillä tuli tämän miehistöpulan takia semmonen skenaario, että se manapankkijärjestelmä oli graafisesti suunniteltu, assetit oli valmiina, niin pois päin, mutta käytännössä meillä ei kukaan pystynyt, ei ehtinyt käytännössä koodaamaan sitä toiminnallisuutta.*

— APPrentice

*Ne interaktiotki, [...] alussa oli paljo isompana asiana, mutta sitte ku niitä alettiin oikeesti tekemään niitä kenttiä niin tuntu että niitten toteuttaminen tai suunnittelu oli yllättävän vaikeata. [...] ehkä se oli semmonen että se tippu siinä, ku suunnitteli niitä kenttiä, niin interaktioita ei tullu suunniteltua siinä mukaan niin paljon.*

— Pointy End

Lisäksi suunnitelmien laajuutta voidaan rajoittaa projektin edetessä aikataulujen pakottaessa priorisoimaan ydinominaisuuksia muiden ominaisuuksien edelle.

*Jouduin tekemään valintoja sen suhteen että pistätkö koodaajan nyt viilaamaan tätä AI:ta (tekoälyä) vai lisäämään näitä loitsuja, ja siinä päädyttiin siihen että se oli parempi ottaa AI koska ne spesiaaliloitsut niin kiva ja olennainen juttu ku ne ois ollu kokonaisuudelle, niin se AI oli tärkeämpi. Aika paljon just tuommosia aikataulullisia ja resurssipohjaisia ratkasuja joutu etenki loppupuolella tekemään.*

— APPrentice

Pointy End puolestaan priorisoi pelimoodit matalammalle aikarajoitteen tullessa vastaan:

*Lähinnä se aikarajoite, ei vaan ehtiny toteuttaa. Kattonu prioriteeteissa ettei oo niin korkeella kuitenkaan.*

— Pointy End

Priorisointi voi liittyä myös suunnitelman kokonaiskuvaan: APPrentice valitsi keskittyä ydinpelimekaniikkoihin.

*Ei, siihen [kokemuspistejärjestelmään] ei lähdetty koska se ois tarkottanu, olis aika nopeessa vaiheessa pitäny jo olla ensinnäkin useampi tappelu kasassa, ja sitte aika loppuvaiheessa vasta oltiin tilanteessa että meillä oli kolme eri vastustajaa ja niin pois päin. Ja siihenki olis pitäny oma materiaalinsa tuottaa, omat hahmonkehitysruudut ja sen semmoset. Niin se oli semmonen yks iso potti jonka olis pystyny sysäämään siihen loppupäähän, niin että se peli tuntuu vielä koherentilta jopa ilman sitä.*

— APPrentice

*Se oli hyvin vahvana toiveena, että [tutoriaali] olis ollu pelissä mukana, [...] mutta ku aikataulun vuoksi tehtiin käytännössä tuommonen kolmen vastustajan tappeludemo, niin se ei enää sen päätöksen jälkeen olisi istunut tuohon kokonaisuuteen.*

— APPrentice

Toisaalta suunnitelmia ei välttämättä muuteta pakon edessä, vaan niitä priorisoidaan jo etukäteen.

*Lopputaistelua ei oo käytännössä dokumentteihin suunniteltu, mutta omassa päässäni olen vähä jo pyöritelly. Sen takia sitä ei nyt oo koska en oo halunnut jatkille turhan paljon antaa hommia, joiden pariin ne ei tule pääsemään tämän proton aikana.*

— APPrentice

*Alkupalaverissa kerroin näistä ideoista, että siinä olis [vastustajien luonteiden kuvailua dialogin avulla]. Mutta kuitenkin myös sillä, että siihen ei aleta kovin paljon syvenemään siinä tuotannon aikana, paitsi jos hyvinkin ehditään. Sillä tavalla tullu tuotannon aikana vähä esiteltyä, ja kyselyä ihmisiltä miten me saatas tähän semmonen hyvin simppeli dialogisysteemi ja niin pois päin. [...] tehä tää pöydällä olevaa hommaa loppuun ennen kuin mietitään tuollaisia höysteitä.*

— APPrentice

Päätös jättää ominaisuus pois JGL:n aikaisesta kehitystyöstä saattoi myös syntyä hyvin aikaisessa vaiheessa, jo ideointivaiheessa.

*[Maailmankartta] oli just näitä alkupalaverien heittoja joka sitten tuli muulta tiimiltä, että sinänsä mä vastustelin sitä ideaa sen takia että ei sitä tähän kevään scopeen oo hyvä työntää.*

— APPrentice

Sama ideoinninomaisuus on näkyvissä myös Pointy Endillä, joka oli sisällyttänyt pelisuunnitelmaan varsinaisten ydinpelimekaniikkojen ulkopuolella olevia asioita, kuten pistemäärän näyttäminen ja voittajan palkitseminen, mutta näitä ei oltu toteutettu.

*Ne jäi lähinnä ideatasolle nuo jutut. [...] Pisteytys on semmonen että se tuskin olis hirveen vaikea toteuttaa, mutta se ei tunnu tällä hetkellä niin tärkeältä.*

— Pointy End

Dokumentointia käytettiin siis myös ideointivälineenä, ja osaa ideoista ei oltu tarkoitettu-



kaan sisällytettäväksi JGL-projektin aikana tehtävään peliin, osa priorisoitiin projektin edessä matalammalle tärkeytasolle ja osa jouduttiin jättämään pakon edessä pois. Useissa tapauksissa ideoinninomaisuus on nähtävissä jo dokumenteista suoraan:

*Tiedon kerääminen: mitä kaikkea tietoa kentistä pitää kerätä? Kuolemat ja tappot? Annetaanko "lopussa" palkintoa voittajalle? Jos ei pelaa loppuun niin näkeekö current scoren jostain?*

— Pointy End: Pelisuunnitelma

*Jos voi aloittaa siitä kentästä mihin viimeksi jäi niin voiko aloittaa myös alusta tai voiko aloittaa toisen pelin eri pelaajamäärällä? Voiko peliä jatkaa jos yksi pelaaja päättää keskeyttää (tappo luonnollisesti helppoa mutta voiko kyseisen pelaajan poistaa kesken pelin ja jatkaa siitä kentästä missä pelaaja viimeksi oli mukana)?*

— Pointy End: Pelisuunnitelma

*(Check this: onko mahdollista muodostua tilanne, jossa hyväksyttäviä yhdistelmiä ei ole laudalla laisinkaan?)*

— APPrentice Bible 07

Ideoinnin lisäksi toinen muistiinpanonomaisuudesta kertova piirre dokumenteissa on se, että kaikki ryhmät käyttivät jonkintyyppistä väri- tai tekstikoodausta erotellakseen eri tarkoitusta varten kirjoitettuja dokumentin sisältöjä. Laajimmin tätä hyödynsi APPrentice, jonka eri väreillä kirjoitetut kategoriat olivat 1) tarkempaa harkitsemista vaativat ominaisuudet, 2) JGL:n ulkopuolelle jäävät ominaisuudet, 3) JGL:n aikana toteutettavat ominaisuudet sekä 4) muistiinpanot.

Emberlandsin dokumenteissa puolestaan yhdessä kohdassa oli merkitty osio, jota ei ollut tarkoitus implementoida sellaisenaan vaan siihen oli tulossa päivitys, lisäksi merkittiin päivitetyt osiot sekä muutoshistoria. Pointy End käytti aiheutunnistetta "pohdi" niiden ominaisuuksien kohdalla jotka vaativat tarkempaa harkitsemista.

Pohdinnan huomaa myös dokumenttien sisällöstä: Pointy Endillä oli esimerkiksi etujen ja haittojen vertailua eri kameratyypeistä sekä listaus kahdesta erilaisesta taistelumeکانیکasta.

*Liikkuva kamera:*

*Edut:*

- *Varsinkin 1v1-tilanteissa parempi luomaan tunnelmaa*
- *Sallii isoissa kartassa scrollaamisen, jolloin ehkä laajemmat kentät toteutettavissa tai kentissä aika-ajoin helpompi havaita tapahtumat*

*Haitat:*

- *Työläämpi toteuttaa, isoissa kartoissa voi zoomata liiankin kauas jolloin kartan yksityiskohtia tai hahmojen liikettä haastavampaa havaita*
- Pointy End: Kenttädesign ja kameran toiminta

Samoin APPrentice vertailee automatisoitua, passiivista manapankkia sekä interaktiivista mallia, niiden haittoja ja hyviä puolia. Näin dokumentteihin päätyi ei pelkästään suunnittelun lopputulos, vaan myös reitti miten kyseiseen ratkaisuun päädyttiin. Erityisesti APPrenticella ja Pointy Endillä dokumenttien kirjoittamista siis leimaa pohdinta ja ideointi yksityiskohtaisuuden ja viimeistellyn tekstin sijasta.

Tämä ideoinninomaisuus viittaa siihen, että dokumenttia ei käytetty valmiina tuotteena vaan väliaikaisena, ei-valmistuvana entiteettinä josta näkee osittain suoraan mikä on vanhentunutta vai ei. Jääntyneen dokumentin ongelmallisuus mahdollisesti siis tiedostettiin, mutta päivittämisen ja poistamisen sijasta säilytettiin vanhentuneet osiot ja merkittiin ne vastaavasti. Tällä koodauksella mahdollisesti siis pyritään ehkäisemään sitä, että toteutettaisiin vahingossa jotain, jota ei ollut tarkoitus toteuttaa.

Dokumenteista ei kuitenkaan kaikissa tapauksissa ole selkeästi erotettavissa, mikä on ideointia ja ei sisällytetty aikatavoitteeseen. Esimerkiksi APPrenticen haastattelusta kävi ilmi, että kaikkia ominaisuuksia ei kyetä toteuttamaan, jolloin päätettiin panostetaan erikoisloitsujen sijasta tekoälyyn. Dokumenteista tätä ei kuitenkaan ollut havaittavissa. Myöskään värikoodausta ei ole käytetty järjestelmällisesti: sekä Pointy Endin että APPrenticen dokumenteissa oli vanhentuneita ominaisuuksia, joihin tätä koodausta ei oltu käytetty.

Aiheuttiko dokumenttien vanhentuminen ongelmia? Kyselyn perusteella eroavaisuudet tehdyssä työssä verrattuna suunnitelmaan eivät johtuneet niinkään dokumentin vanhentumisesta, vaan useammin tapahtui että suunnitelmat joko oltiin dokumentoitu tai niistä oli keskus-

teltu, mutta niistä oli silti eriäviä käsityksiä (kuva 18). Kyseessä vaikuttaa siis olevan enemmänkin vaikeudet tavoitteiden kommunikoinnissa. Eroavaisuuksia ilmeni (kuva 16) keskimäärin muutamia kertoja, jolloin uusi ratkaisu oli hieman useammin yhdistelmä suunnitelmasta ja tuotetusta työstä tai työn korjaaminen, kuin tehdyn työn säilyttäminen (kuva 19), eli se aiheutti lisätyötä.

Suurimmat syyt jättää toteuttamatta aiemmin dokumentoidut ominaisuudet olivat siis seuraavat: dokumentointi oli ideointia, se jäi tavoitelaajuuden ulkopuolelle projektin aikana, toteuttaminen osoittautui vaikeaksi joko suunnitelman luonteen tai projektin hallinnan vuoksi. Dokumentoinnin vanhentumiseen reagoitiin merkitsemällä kategoriat.

## 7 Johtopäätökset

Tämän tutkielman tavoitteena oli tutkia, miten aloittelevat peliprojektit tuottavat ja käyttävät dokumentointia ja miten dokumenttien päivittämättömyys esiintyy, sekä tuottaa hypoteeseja jatkotutkimusta varten.

Projektityylin piirteistä saatiin melko vähän tietoa johtuen siitä, että projektin aikaväli oli niin lyhyt. Vaikka mikään projekteista ei eksplisiittisesti omaksunut tiettyä formaalia projektityyliä, ketterän kehityksen piirteistä oli nähtävissä syklit, suoran kommunikoinnin painottaminen, dokumentin ei-valmius sekä suunnittelun ja tuotannon tapahtuminen yhtäaikaisesti.

Vaihtoehtoiset suunnittelun tavat olivat erityisen tärkeitä grafiikan ja musiikin kohdalla, joissa kaikki ryhmät painottivat dokumentoinnin sijasta konseptitaidetta sekä keskustelua. Näissäkin tapauksissa kuitenkin kyseiset osa-alueiden vastuuhenkilöt käyttivät dokumentteja inspiraation lähteenä. Ryhmän pienen koon koettiin edesauttaneen suoran kommunikoinnin tehokkuutta. Myös aikarajoite aiheutti että suunnittelua tehtiin suoraan työn yhteydessä.

Potentiaalisesti ongelmallista dokumentoimattomuutta tapahtui kun osa-alueesta oli vastuussa yksi henkilö. Dokumentointia äänisuunnittelun kohdalla ei katsottu tarpeelliseksi, koska äänisuunnittelijoita oli vain yksi. On tärkeää kuitenkin huomioida, että projektin koon kasvaessa ja mahdollisesti uusien jäsenten liittyessä projektiin dokumentoimattomuus saattaisi aiheuttaa ongelmia eikä suora kommunikointi enää välttämättä riitä, joten tämä saattaisi pitkällä tähtäimellä muuttua ongelmaksi.

Dokumenttien jääytymistä tapahtui kaikissa ryhmissä: dokumenteissa oli asioita joita ei oltu toteutettu peliin. Kaikki nämä eivät kuitenkaan olleet haitallisia itse peliprojektille. Syitä dokumentoitujen ominaisuuksien toteuttamatta jättämiselle oli kolme: ne sijoitettiin tavoitelaa-juuden ulkopuolelle joko heti alussa tai projektin aikana kun jotain muuta priorisoitiin niiden edelle, vaikeus toteuttaa joko tiimin muutosten tai suunnitelman epäkäytännöllisyyden takia, tai ne olivat ideointityyppistä tekstiä tarkkojen suunnitelmien sijasta.

Dokumentteja siis käytettiin myös tallentamaan reitti valittuihin ratkaisuihin, mikä näkyi ainakin kolmella eri tavalla: vanhentuneet ratkaisut säilytettiin, pohdinnat ja vertailut kirjoitet-

tiin eksplisiittisesti sekä vanhentuneet osiot kategorisoitiin asianmukaisesti väri- tai muulla koodauksella. Mahdollisesti vaara toteuttaa jotain vanhentunutta oli tiedostettu, mutta kategorisointia ei kuitenkaan välttämättä toteutettu kattavasti. Epäselväksi jäi, onnistuivatko muut ryhmän jäsenet huomioimaan kyseisen ongelman ja olivatko kaikki tietoisia siitä.

Johtopäätös siis on, että dokumentteja hyödynnettiin kommunikointivälineenä niissä tapauksissa, joissa kommunikointia tarvittiin, mutta muita suunnittelukanavia käytettiin dokumenttien tukena. Lisäksi dokumenttien sisällölle oli ominaista muistiinpanonomaisuus sekä ketterille menetelmille tyypillisesti väliaikaisuus ja viimeistelemättömyys.

Tämä viittaa siihen, että dokumentointi on laajempi kokonaisuus kuin pelkkä suunnitelma. Dokumentin tarkoituksena oli mahdollisesti olla kokonainen kuvaus pelistä, ei kuvata pelkkää JGL:n aikaista projektia vaan peliä kokonaisuutena riippumatta siitä tullaanko se toteutamaan sekä lisäksi prosessin kulkua pelin lopputuloksen sijasta.

### **Jatkotutkimusaiheet**

On huomioitava, että JGL-projektiryhmille asetettu koulutuksellinen tavoite sekä lyhyt aikarajoite, vain viisi kuukautta, ei vastaa peliprojektia, jonka ensisijainen tavoite on julkaisu. Pidemmällä aikavälillä projektiryhmillä saattaisi olla parempi mahdollisuus hyödyntää projektikäytänteitä ja tämä saattaisi vaikuttaa siihen miten dokumentointia tehdään. Yhtenä tutkimuskohteena voisivat olla peliyritykset.

JGL:ssä oli useita tekijöitä, jotka vaikuttivat siihen, että projektikäytänteitä ei käytetty formaalisti. Olisiko projektikäytänteiden hyödyntäminen suuremmassa määrin avuksi a) lyhyen aikavälin projekteille, b) kokemattomalle projektiryhmälle, c) pienelle projektiryhmälle? Missä vaiheessa aikatavoite tai ryhmän koko alkaa vaikuttaa siten että formaalimpaa dokumentointia tarvitaan?

Toinen merkittävästi vaikuttava seikka on se että jokaisessa ryhmässä oli vain yksi äänisuunnittelija, mikä johti siihen että kommunikointia äänisuunnittelusta ei tarvittu ja siten dokumentointia tästä aiheesta ei koettu tarpeelliseksi. Mikäli samaa aluetta suunnittelevia projektin jäseniä olisi useampia, kommunikointi saattaisi olla suuremmassa roolissa jolloin sama lähtökohta jättää dokumentoimatta voisi toimia eri tavalla. Tämä viittaa siihen, että aloittele-

vien peliprojektien tulisi erityisesti kiinnittää huomiota alueisiin joista vain yksi henkilö on vastuussa ja miettiä voiko pitkällä tähtäimellä tulla ongelmia.

Tässä tutkimuksessa havainnointi rajoittui vain dokumentteihin sekä jälkikäteen tehtyihin haastatteluihin ja kyselyyn. Suunnittelua kuitenkin tehdään dokumenttien lisäksi myös epäformaalisti esimerkiksi keskusteluissa tai erilaisissa verkossa toimivissa kommunikaatiokanavissa. Dokumenttien tutkiminen suhteessa muihin kommunikaatiokanaviin suoraa havainnointia hyödyntävällä tutkimuksella antaisi olennaista tietoa suunnitteluprosessista kokonaisuutena ja dokumenttien asemasta suhteessa koko prosessiin.

## Kirjallisuutta

- Adams, E. 2007. *The Designer's Notebook: Why Design Documents Matter*. Saatavilla WWW-muodossa [http://www.gamasutra.com/view/feature/129946/the\\_designers\\_notebook\\_why\\_.php](http://www.gamasutra.com/view/feature/129946/the_designers_notebook_why_.php), viitattu 1.12.2016
- Al-Azawi, R., Ayesh, A. & Al-Obaidy, M. 2014. *Towards Agent-based Agile approach for Game Development Methodology*. Computer Applications and Information Systems (WCCAIS), 2014 World Congress
- Almeida, M. & da Silva, F. 2013. *Systematic Review of Game Design Methods and Tools*. Entertainment Computing – ICEC 2013
- Basili, V. & Turner, A. 1975. *Iterative Enhancement: A Practical Technique for Software Development*. IEEE Transactions On Software Engineering, VOL. SE-1, No.4
- Bethke, E. 2006. *Game Development and Production*. Wordware Publishing, Inc.
- Bourque, P. & Fairley, R. 2014. *SWEBOK v3.0. Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society
- Callele, D., Neufeld, E., & Schneider, K. 2005. *Requirements Engineering and the Creative Process in the Video Game Industry*. Proceedings of the 2005 13th IEEE International Conference on Requirements Engineering (RE'05)
- Dormans, J. 2012. *Engineering Emergence: Applied Theory for Game Design*. Amsterdam University of Applied Sciences (2012)
- Jyväskylän Game Lab. Saatavilla WWW-muodossa <http://www.mit.jyu.fi/scoma/jklgameLab/> Viitattu 17.2.2016
- Kanode, C. & Haddad, H. 2009. *Software Engineering Challenges in Game Development*. 2009 Sixth International Conference on Information Technology: New Generations, s. 260–265
- Kasurinen, J., Laine, R. & Smolander, K. 2013. *How Applicable Is ISO/IEC 29110 in Game Software Development?* Product-Focused Software Process Improvement 14th International Conference, PROFES 2013
- Kasurinen, J., Maglyas, A. & Smolander, K. 2014. *Is Requirements Engineering Useless in Game Development?* Kirjassa Requirements Engineering: Foundation for Software Quality: 20th International Working Conference, REFSQ 2014, Essen, Germany, April 7-10,

2014. Proceedings. Springer International Publishing
- Keith, C. 2010. *Agile Game Development with Scrum*. Addison-Wesley Signature Series, Pearson Education Inc.
- Ketterän ohjelmistokehityksen julistus. Saatavilla WWW-muodossa <http://agilemanifesto.org/iso/fi/manifesto.html>, viitattu 18.3.2016
- Mitre-Hernández, H., Lara-Alvarez, C., González-Salazar, M. & Martin, D. 2016. *Decreasing Rework in Video Games Development from a Software Engineering Perspective*. Trends and Applications in Software Engineering, s. 295–303
- Murphy-Hill, E., Zimmermann, T., & Nagappan, N. 2014. *Cowboys, Ankle Sprains, and Keepers of Quality: How Is Video Game Development Different from Software Development?* Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)
- Nevelsteen, K. & Gayoso, S. 2011 *GDD as a Communication Medium*. Games and Innovation Research Seminar 2011 Working Papers.
- O’Hagan, A., Coleman, G. & O’Connor, R. 2014 *Software Development Processes for Games: A Systematic Literature Review*. 21th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2014), s. 182-193
- Petrillo, F., Pimenta, M., Trindade, F. & Dietrich, C. 2009 *What went wrong? A survey of problems in game development*. Computers in Entertainment (CIE) - SPECIAL ISSUE: Media Arts and Games archive. Volume 7 Issue 1
- Politowski, C., Fountoura, L., Petrillo, F. & Guéhéneuc, Y. 2016 *Are the Old Days Gone? A Survey on Actual Software Engineering Processes in Video Game Industry*. Games and Software Engineering (GAS), 2016 IEEE/ACM 5th International Workshop
- Rouse, R, III. 2005 *Game Design: Theory and Practice. Second Edition*. Wordware Publishing, Inc.
- Runeson, P. & Höst, M. 2009. *Guidelines for conducting and reporting case study research in software engineering*. Empir Software Eng (2009) 14:131–164
- Salazar, M., Mitre, H., Lemus Olalde, C. & Sánchez, J. 2012 *Proposal of Game Design Document from Software Engineering Requirements Perspective*. The 17th International Conference on Computer Games
- Schell, J. 2008. *Book of Lenses*. Elsevier Inc.
- Stettina, C. & Heijstek, W. 2011. *Necessary and Neglected? An Empirical Study of Internal*



- Documentation in Agile Software Development Teams*. SIGDOC '11 Proceedings of the 29th ACM international conference on Design of communication
- Yin, R. 2009. *Case Study Research: Design and Methods, Fourth Edition*. Applied Social Research Methods Series. Sage Publications Inc.
- Washburn, M., Sathiyarayanan, P., Nagappan, M., Zimmermann, T. & Bird, C. 2016. *"What Went Right and What Went Wrong": An Analysis of 155 Postmortems from Game Development*. 2016 IEEE/ACM 38th IEEE International Conference on Software Engineering Companion
- Winget, M. & Sampson, W. 2011. *Game development documentation and institutional collection development policy*. Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries
- Zhi, J., Garousi-Yusifoglu, V., Sun, B., Garousi, G., Shahnewaz, S. & Ruhe, G. *Cost, benefits and quality of software development documentation: A systematic mapping*. Journal of Systems and Software, Volume 99, January 2015, s.175-198

# Liitteet

## A Pelisuunnittelijoiden haastattelu

- Projektin vaiheet ja suunnittelu
- Dokumentointi
  - Aloittaminen, päivittäminen ja lopettaminen
  - Käyttö ja kirjoittaminen
- Dokumentoidut ominaisuudet joita ei oltu implementoitu peliin
- Peliin implementoidut ominaisuudet joita ei oltu dokumentoitu

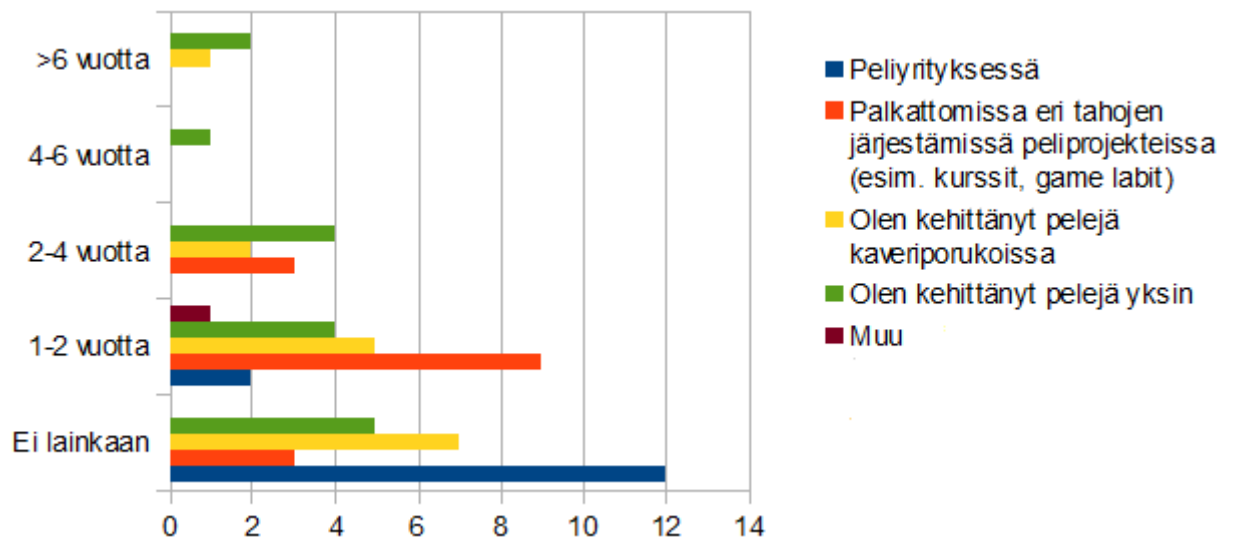
## B Kysely

- Taustatiedot
  1. Missä roolissa toimit pääasiassa?
  2. Millä tavalla olet toiminut pelialalla: peliyrityksessä, palkattomissa peliprojekteissa kuten kurseissa, kaveriporukoissa, yksin
  3. Mitä kommunikointivälineitä ryhmällänne oli käytössä dokumentoinnin lisäksi: tehtävähallintajärjestelmä, online-ryhmäkeskustelu, kokoukset, sähköposti
  4. Lakkasitko jossain vaiheessa osallistumasta projektiin aktiivisesti?
  5. Kuinka paljon aikaa vietit labilla keskimäärin?
- Dokumentointi
  1. Mitä pelinne seuraavista dokumentoiduista osa-alueista olet lukenut dokumenteista?
  2. Tuotitko dokumentteihin sisältöä joihinkin osa-alueihin?
  3. Kuinka usein luit muiden kirjoittamia pelisuunnitteludokumentteja?
  4. Mihin tarkoitukseen luit muiden kirjoittamia dokumentteja?
- Varsinaista työtä koskevat kysymykset
  1. Jos olit projektin aikana epävarma mitä olitte sopineet suunnitteluratkaisuista,

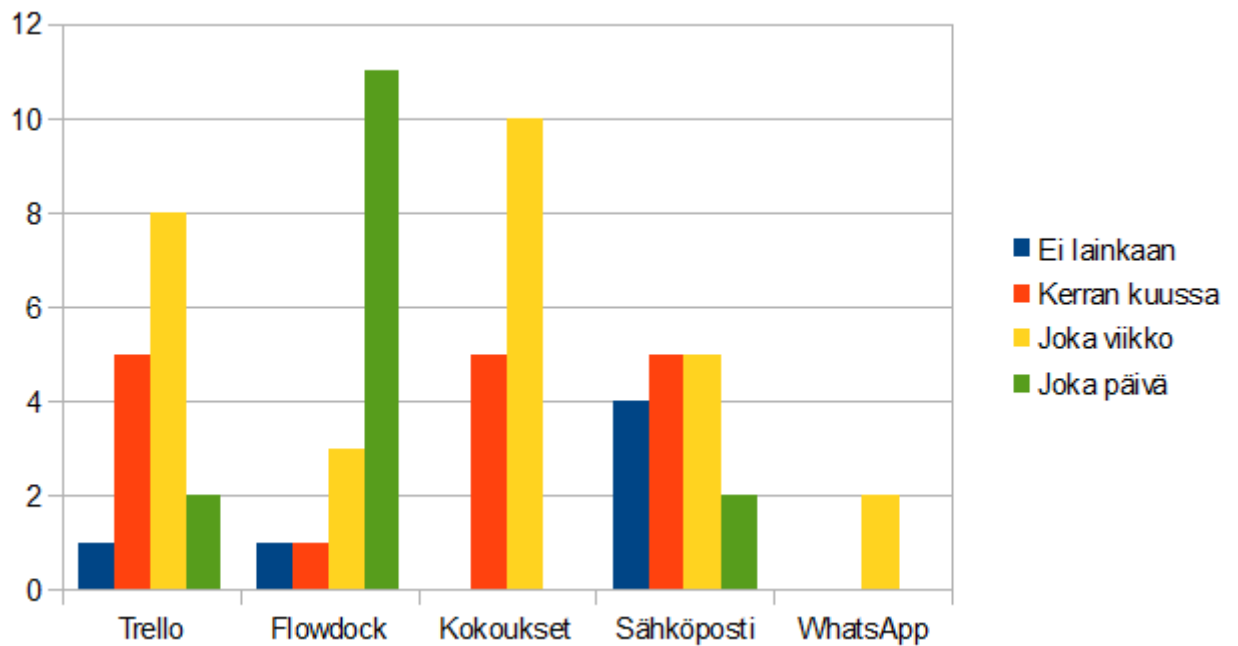
miten tarkistit suunnittelupäätöksen: en tarkistanut, dokumenteista, kokouksessa, keskusteluissa labilla tai muualla, sähköisen kanavan kautta, kahdenvälinen sähköinen kanava

2. Kävikö jossain vaiheessa niin, että tuottamasi sisältö ei vastannut siihen mitä joku muu oli tarkoittanut, esim. koodi ei vastannut suunnitelmiin?
3. Miten eroavaisuudet kävivät ilmi: valmiista tai puolivalmiista työn osasta, kokouksessa, epävirallisessa keskustelussa labilla tai muualla, sähköisen kanavan kautta
4. Missä projektin vaiheessa erimielisyyksiä tapahtui eniten?
5. Mistä eroavaisuudet johtuivat: ominaisuudesta ei oltu keskusteltu ja sitä ei ollut dokumenteissa, ominaisuudesta oltiin keskusteltu mutta eri käsityksiä oli silti, ominaisuus oli dokumenteissa mutta siitä oli silti eri käsityksiä, ominaisuus oli dokumenteissa mutta en ollut lukenut kyseistä dokumenttia, suunnitelmia ei voitu toteuttaa teknisen tai muun syyn takia, suunnitelma oli puutteellinen ja ominaisuus oli lisäys tai korjaus, dokumentti oli vanhentunut ja implementointi tapahtui vahingossa
6. Miten erimielisyydet ratkaistiin: muu kuin suunnittelija teki uudelleen, tehty työ säilytettiin, kombinaatio näistä kahdesta, kokonaan uusi ratkaisu

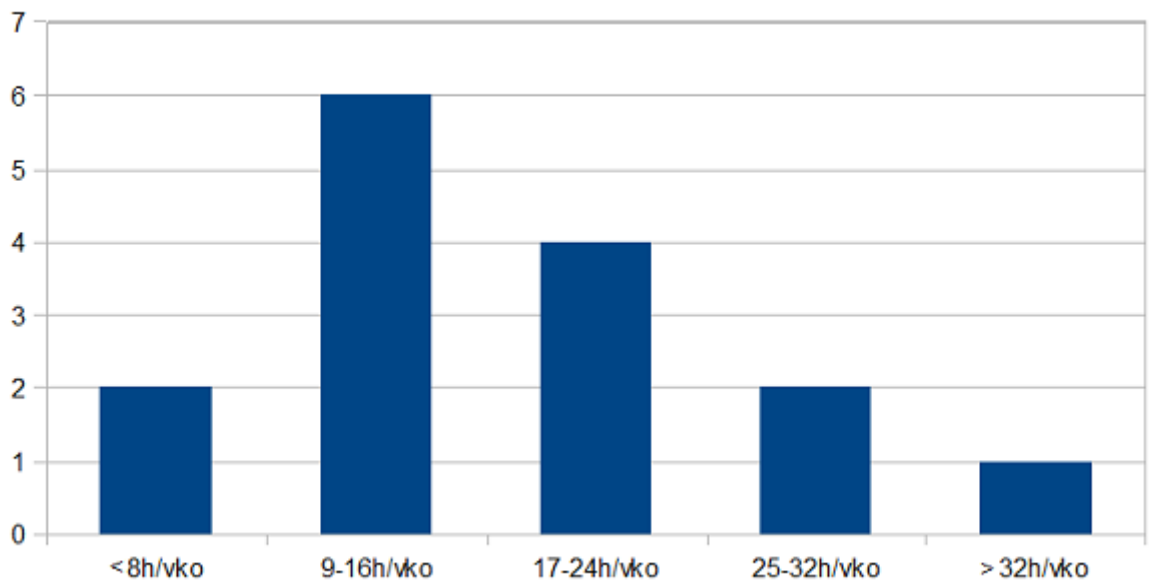
## C Kyselyn vastaukset



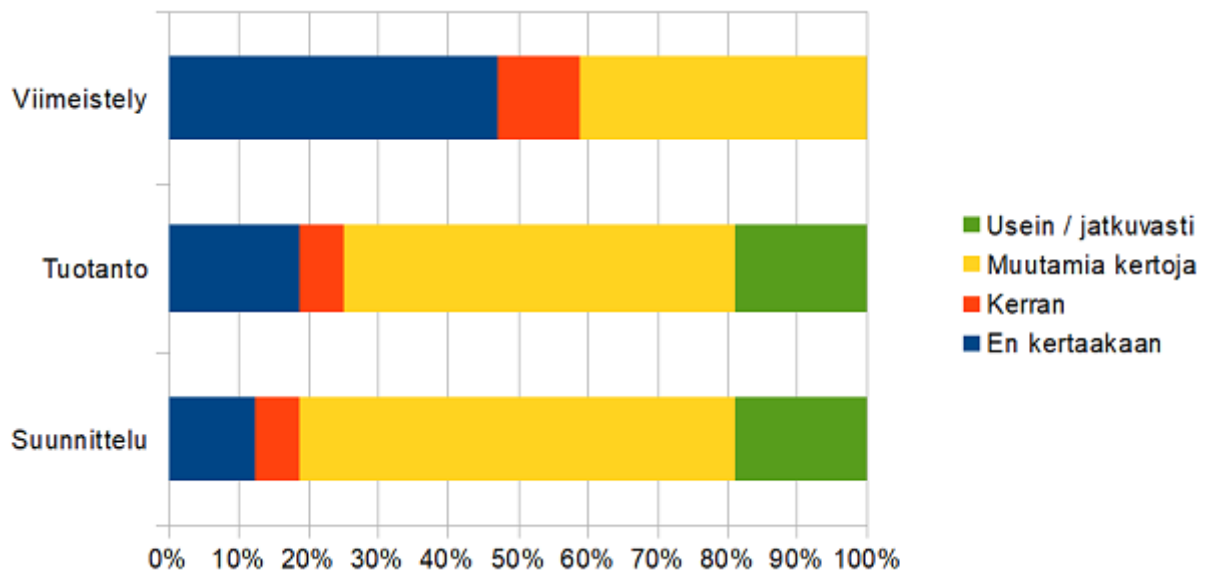
Kuvio 10. Millä tavalla olet toiminut pelialalla?



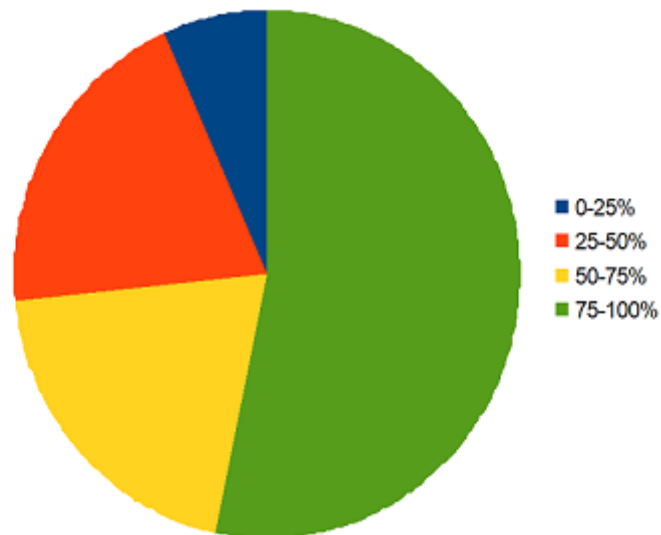
Kuvio 11. Mitä kommunikointivälineitä ryhmällänne oli käytössä ja kuinka säännöllisesti käytit sitä?



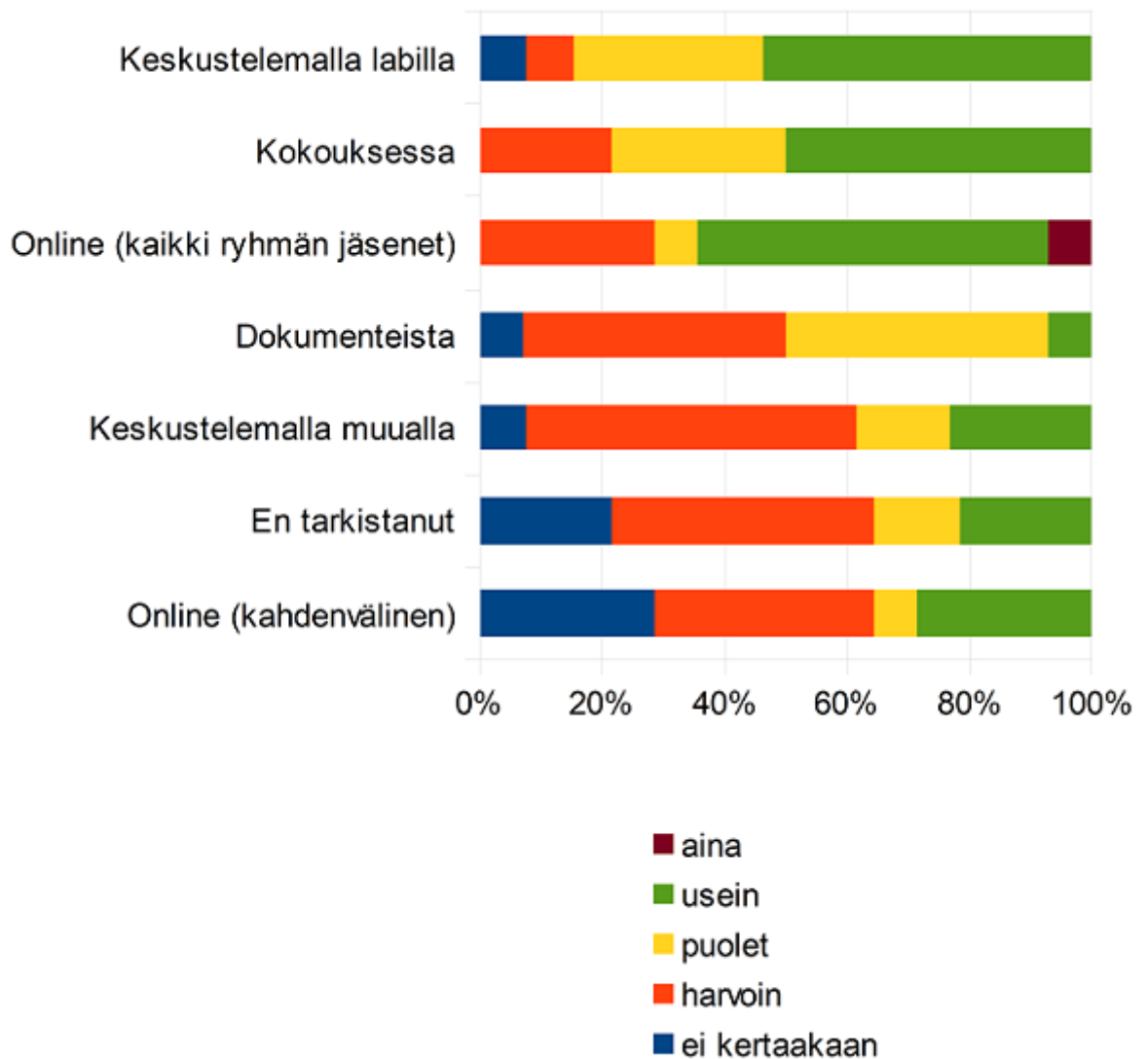
Kuvio 12. Kuinka paljon aikaa vietit labilla?



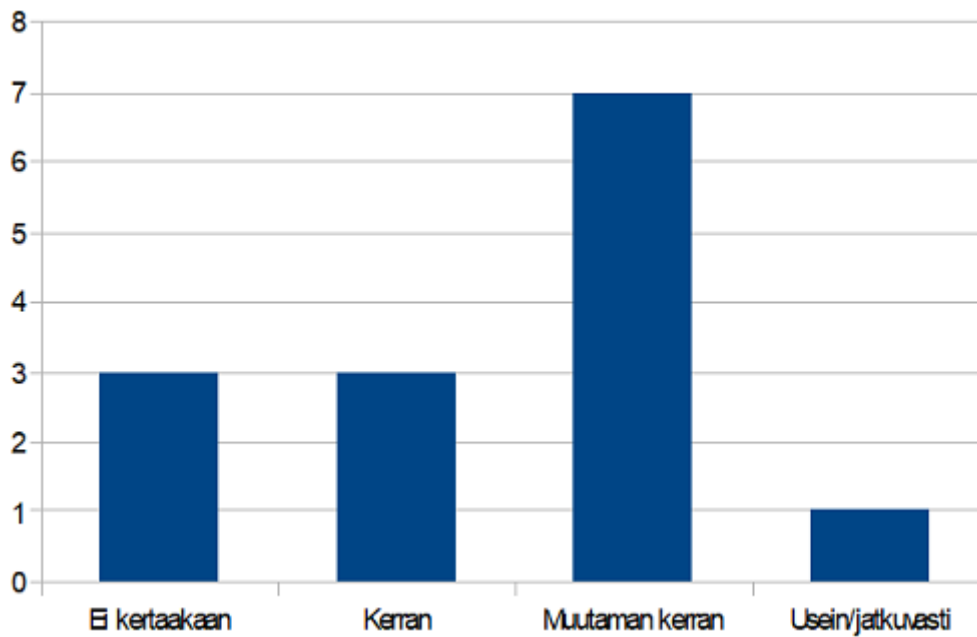
Kuvio 13. Missä vaiheissa luit dokumentteja?



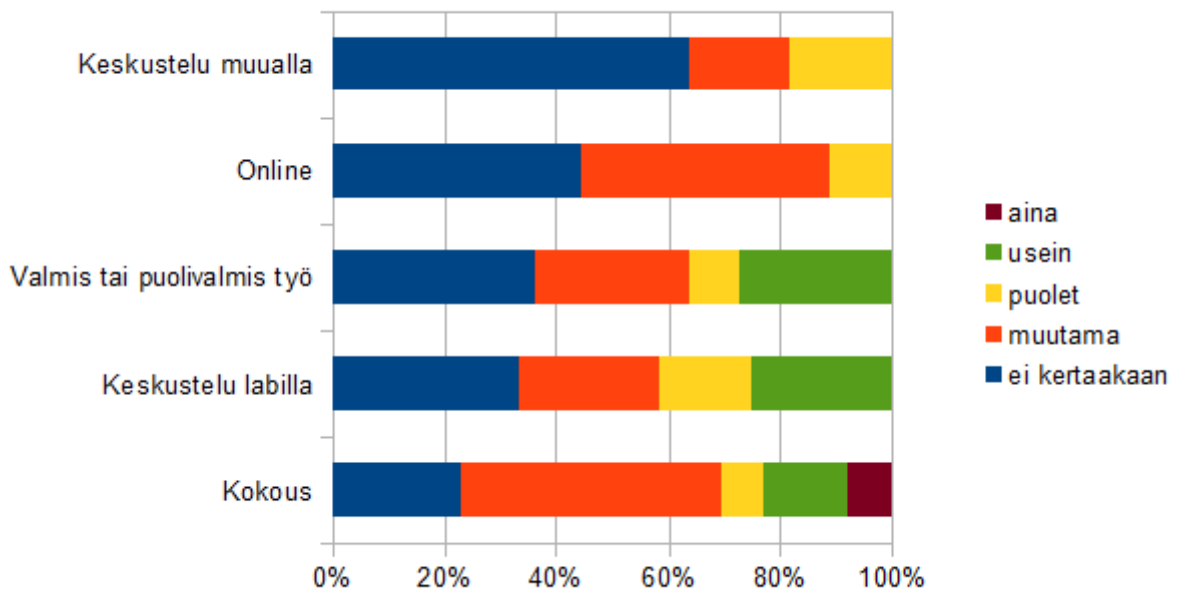
Kuvio 14. Mitä dokumentoiduista osa-alueista olet lukenut dokumenteista?



Kuvio 15. Jos olit projektin aikana epävarma mitä olitte sopineet jonkin tietyn asian ratkaisusta, miten tarkistit suunnittelupäätöksen?

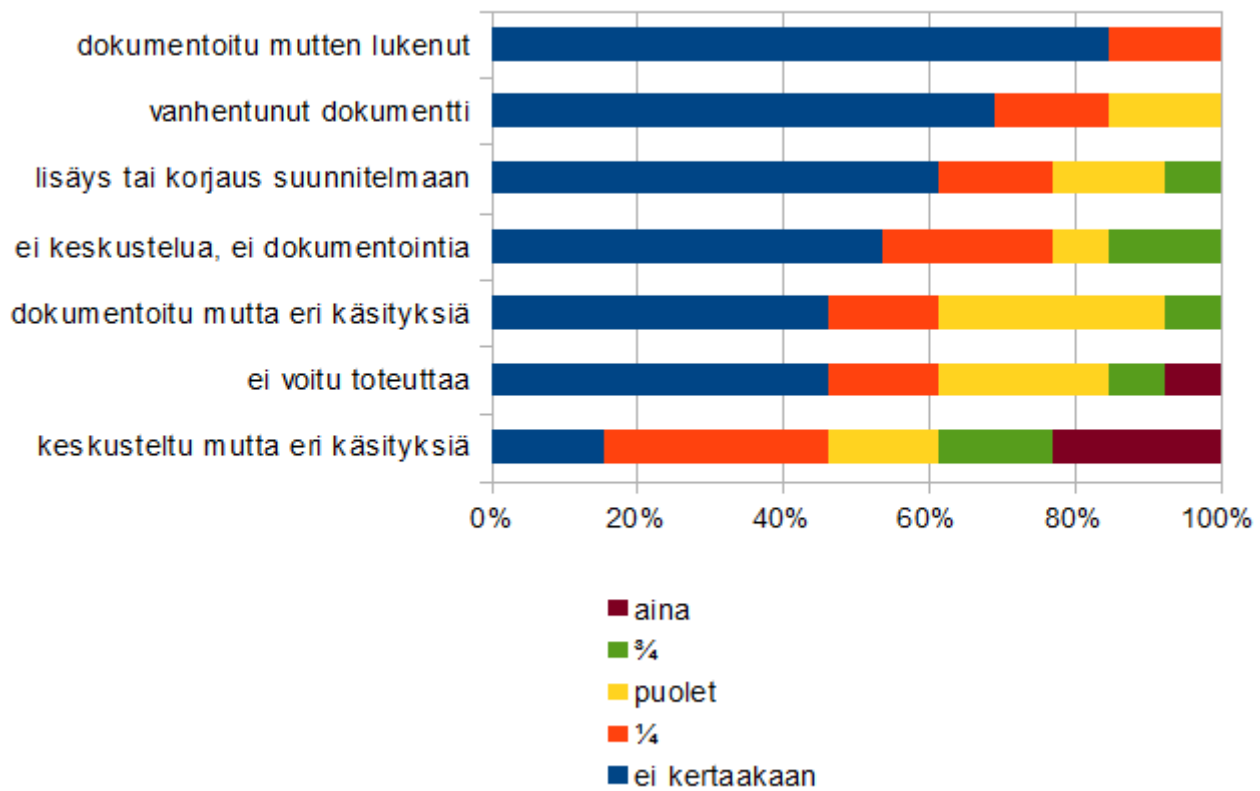


Kuvio 16. Kävikö jossain vaiheessa niin että tuottamasi sisältö ei vastannut jonkun muun suunnitelmiin?

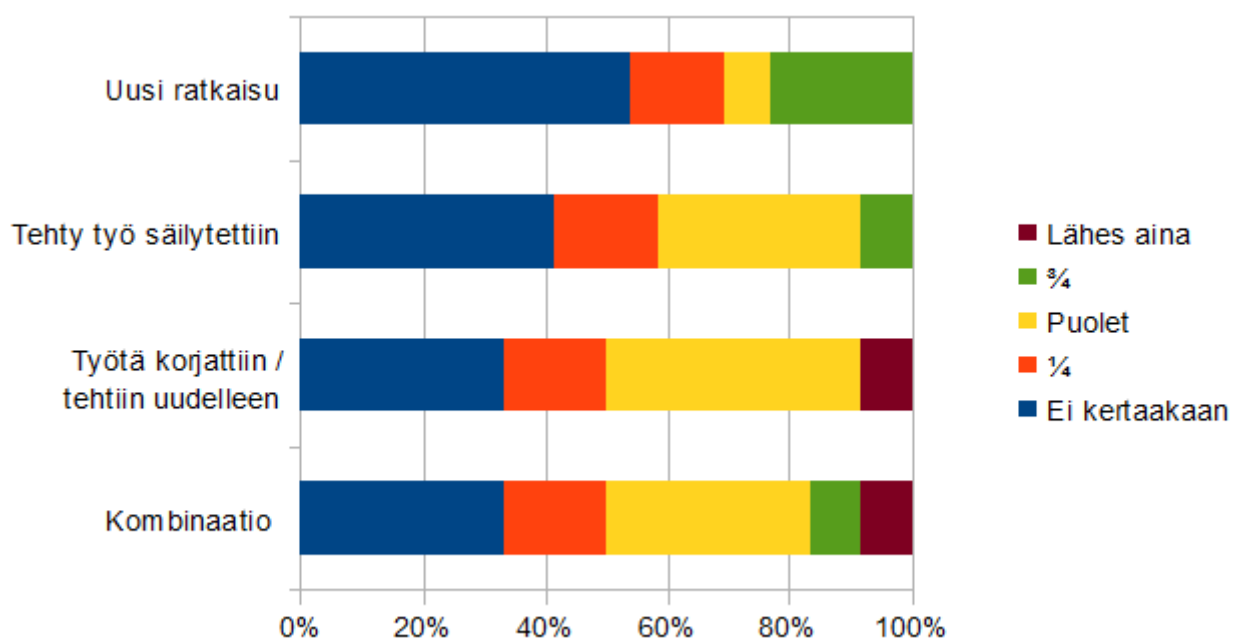


Kuvio 17. Miten eroavaisuudet suunnittelussa ja tehdyssä työssä kävivät ilmi?





Kuvio 18. Mistä eroavaisuudet suunnittelussa ja tehdyssä työssä johtuivat?



Kuvio 19. Miten eroavaisuuksiin reagoitiin