Matti Kanervo

OHJELMISTOJEN KONFIGURAATION- JA VERSIONHALLINNAN ONGELMAT JA RATKAISUT MOBIILIPELITUOTANNOSSA

Tietojärjestelmätieteen
pro gradu -tutkielma
4.7.2008

Jyväskylän yliopisto
Tietojärjestelmätieteiden laitos
Jyväskylä

TIIVISTELMÄ

Kanervo, Matti Olavi
Ohjelmistojen konfiguraation- ja versionhallinnan ongelmat ja ratkaisut mobiilipelituotannossa / Matti Kanervo
Jyväskylä: Jyväskylän yliopisto, 2008
156 s.
Tietojärjestelmätieteen pro gradu -tutkielma

Tutkielman tutkimusalue on konfiguraation- ja versionhallinta mobiilipelituotannossa. Tutkielmassa kartoitetaan mobiilipelituotannon ongelmia konfiguraation- ja versionhallintaan liittyen ja sovelletaan ohjelmistotuotantoon suunnattuja ratkaisumalleja kyseisiin ongelmiin. Mobiilit laitteet kehittyvät jatkuvasti, joten mobiilipeleistä voidaan tehdä entistä monimutkaisempia. Tästä johtuen mobiilipelituotannolla on, omiin peleihin liittyvien kehitysprojektien lisäksi, suuret paineet versionpäivityksiin laitteistovalmistajien taholta. Tutkielma muodostuu käsitteellisteoreettisesta osasta ja empiirisestä osasta. Tutkielma pohjautuu tieteellisten lähteiden sisältämien ratkaisuiden soveltamiseen mobiilipelituotannossa koettuihin ongelmiin. Empiiriseen tutkimukseen kuuluu mobiilipelituotannon tarkempi kuvaaminen, konfiguraation- ja versionhallinnan ongelmien kartoitus ja ratkaisumallien hahmottaminen.

Tutkielman käsitteellisteoreettisessa osassa rakennettiin empiirisen tutkimuksen tutkimuskehys, jota sovellettiin kuuteen kohdeorganisaatioon. Tutkimuskehys osoittautui toimivaksi. Empiirisen tutkimuksen tuloksena saatiin asetetut tutkimusongelmat ratkaistua. Tutkimuksen perusteella tyypilliset ohjelmistotuotannon konfiguraation- ja versionhallintaan liittyvät ongelmat esiintyvät myös mobiilipelituotannossa ja tieteellisten lähteiden määrittelemät ratkaisumallit soveltuvat niiden ratkaisemiseen myös mobiilipeliteollisuudessa. Ratkaisumallien rakentamisessa täytyi kuitenkin ottaa huomioon mobiilipelituotannon erityispiirteet.

AVAINSANAT: Ohjelmisto, versionhallinta, konfiguraationhallinta, mobiilipeli, mobiilipelituotanto

SISÄLTÖ

KUVIOT

TAULUKOT

1 JOHDANTO

Tässä tutkielmassa on aiheena mobiilipelituotannon version- ja konfiguraationhallintaan liittyvät ongelmat (software version management ja software configuration management). Mobiilipelituotannolla tarkoitetaan tässä yhteydessä matkapuhelimiin pelejä valmistavia yhtiöitä. Mobiilit laitteet kehittyvät jatkuvasti ja mobiilipelituotannon täytyy pysyä mukana jatkuvassa kehityksessä. Täten mobiilipelituotannolla on omiin peleihin liittyvien kehitysprojektien lisäksi suuret paineet päivityksiin laitteiston valmistajien puolesta (Damsgaard & Marchegiani 2004, 639). Ohjelmistotuotannossa standardit tarjoavat huomattavat apukeinot kehittäjille ja ylläpitäjille (Tuohey 2002, 48). Mobiilipeliteollisuuden näkökulmalta standardointia voidaan lähtökohtaisesti pitää hyvänä asiana. Valitettavasti instituutioita, jotka toteuttavat sitä on vähän (Sirén 2004).

Tutkielman kannalta tärkeitä käsitteitä ovat versio, konfiguraatio, ohjelmisto-objekti, versionhallinta ja konfiguraationhallinta. **Versio** esittää kehittyvän ohjelmisto-objektin tilan (Conradi & Westfechtel 1998, 238). **Konfiguraatio** taas esittää tiettyyn ympäristöön tai käyttöön tarkoitetun ohjelmiston osat ja niistä koostuvan rakenteen (ATK-sanakirja 4.0, 2003). **Ohjelmisto-objekti** voi olla mikä tahansa ohjelmistoon liittyvä tekijä, joka voidaan laittaa versionhallinnan alaisuuteen (Conradi & Westfechtel 1998, 238).

Konfiguraation- ja versionhallintaan liittyen löytyy paljon kirjallisuutta, mutta kyseisten termien eroavaisuuksia käsittelevää artikkelia ei ole julkaistu. Termiä **konfiguraation-hallinta** käytetään usein, kun tarkastellaan yksittäisiä muutoksia ja muutoksiin liittyviä hallinnollisia toimia, kuten muutosten hyväksymiskäytänteitä. Termiä **versionhallinta** taas käytetään kun tarkastellaan muutostenhallintaa tasoa korkeammasta näkökulmasta. Versionhallinta tarkastelee muutostenhallintaa kokonaisuutena, kun taas konfiguraation-hallinnan näkökulma on enemmän yksittäisten muutosten hallinnassa. Yksinkertaistettuna, konfiguraationhallinta on kiinnostunut ohjelmistotuotteen rakenteellisesta koostumuksesta ja versionhallinta on kiinnostunut ohjelmistotuotteen elinkaarellisesta kehityksestä.

Mobiileihin järjestelmiin keskittyvien toimialojen version- ja konfiguraationhallintaan liittyvät ongelmat ovat nousseet suuremmissa määrin esille viimeisten viiden vuoden aikana. Tämä johtuu muun muassa nopeasta teknologisesta kehityksestä ja laitteistojen erilaisuudesta (jopa saman valmistajan mallien välillä). Mobiilipelituotanto sopii hyvin version- ja konfiguraationhallintaan liittyvien ongelmien tutkimiseen, sillä peleillä voi olla ainakin näennäisesti pitkä elinkaari. Tämä johtuu toimialalla nopeasti vaihtuvista kohdelaitteistosukupolvista, minkä takia mobiilipeleihin joudutaan tekemään lyhyillä aikaväleillä uusia versioita. (Son & Tan 2008). Lisäksi mobiilipelituotannossa täytyy ottaa huomioon kohdelaitteistojen tekniset rajoitteet (Akkawi, Schaller, Wellnitz & Wolf 2004, 78).

Tutkimusalueeksi on valittu mobiilipelituotanto, sillä koko mobiilialan tutkiminen olisi ollut liian suuri kokonaisuus. Päätavoitteena on tutkia, ilmeneekö mobiilipelien version- ja konfiguraationhallinnassa samankaltaisia ongelmia kuin perinteisten ohjelmistojen version- ja konfiguraationhallinnassa. Lisäksi on tarkoitus tutkia, pystyykö mobiilipelien version- ja konfiguraationhallintaan soveltamaan samoja ratkaisumalleja ongelmien ratkaisemiseksi/ennaltaehkäisemiseksi. Tutkielman tavoitteen voi tiivistää kahteen tutkimusongelmaan:

- Esiintyykö mobiilipelituotannossa ohjelmistojen konfiguraation- ja versionhallinnan yleisimpiä ongelmia?
- Miltä osin ohjelmistojen konfiguraation- ja versionhallinnan ratkaisumallit soveltuvat mobiilipeliteollisuuteen?

Tutkimus sisältää sekä käsitteellisteoreettisen että empiirisen osion. Käsitteellisteoreettinen osio pohjautuu pääosin tieteellisiin lähteisiin. Se koostuu ohjelmistotuotannon kuvauksesta tutkielman kannalta tärkeiltä osilta, konfiguraation- ja versionhallinnan kartoittamisesta ja mobiilipelituotannon kuvauksesta. Lisäksi käsitteellisteoreettista osuutta käytetään tärkeimpänä tukirankana empiirisen tutkimuksen tutkimuskehyksen muodostuksessa ja tulosten analysoinnissa.

Empiiriseen tutkimukseen kuuluu mobiilipeliteollisuuden rakenteen ja tutkielman kannalta oleellisten organisatoristen tekijöiden vertaaminen, aikaisemman tutkimuksen pohjalta tehtäviin määritelmiin. Empiirisen tutkimuksen pääpaino on konfiguraation- ja versionhallintaan liittyvien ongelmien kartoituksessa. Ratkaisumallien soveltuvuus mobiilipelituotantoon suoritetaan tieteellisiin lähteisiin perustuvalla analysoinnilla. Empiirinen tutkimus on laadullinen, sillä tutkielmassa käsiteltävät ongelmat ovat suurimmilta osin monisyisiä. Empiirisessä tutkimuksessa ei keskitytä siis tilastotieteellisen materiaalin keräämiseen, vaan syvennytään tarkasteltavien avainprosessien osalta mobiilipelituotannon ongelmiin.

Empiirisen tutkimuksen toteutustavaksi valittiin haastattelu, sillä haastattelun avulla päästään paremmin laadulliseen tavoitteeseen kuin esimerkiksi lomaketta tai sähköpostikyselyä käyttämällä. Riittävän aineiston saamiseksi rakennetaan vaihtoehtoista vastaustapaa varten haastattelua vastaava lomakekysely Korppi-järjestelmään. Aikataulullisista syistä suurin osa vastaajista joutui käyttämään Korppi-järjestelmään rakennettua lomakekyselyä. Vastaajat ovat asiantuntijoita organisaatioiden eri tasoilta: sillä vastaajat ovat organisaation johtajia, teknisiä päälliköitä ynnä muita organisatorisista tehtävistä vastaavia henkilöitä. Tutkittavia organisaatioita on yhteensä 6 kappaletta.

Luvussa 2 muodostetaan kokonaiskuva perinteisestä ohjelmistotuottajasta ja keskitytään konfiguraation- ja versionhallintaan liittyen tärkeimpiin organisatorisiin alueisiin. Luvussa 3 syvennytään luvussa 2 rajattuihin avainprosesseihin. Toisin sanoen kartoitetaan perinteisten ohjelmistotuottajien version- ja konfiguraationhallinnan ongelmia ja ratkaisumalleja. Tietoa haetaan perinteisestä ohjelmistotuotannosta, koska ohjelmistojen kehitykseen ja ylläpitoon liittyviä ongelmia on tutkittu useiden vuosikymmenten ajan nimenomaan tällä toimialalla. Luvussa 4 selvennetään mobiilipelituotannon ominaispiirteitä. Luku koostuu mobiilijärjestelmien, mobiilipelien, mobiilipeliteollisuuden rakenteen ja mobiilipelituottajien kuvauksista. Luvussa 5 rakennetaan empiirisen tutkimuksen tutkimuskehys. Tutkimuskehys pohjautuu kirjallisuuskatsauksessa läpikäytyihin tieteellisiin lähteisiin. Luvussa kerrataan tutkimuksen tavoitteet, luodaan selkeä rakenne ja viitekehykset tutkimukselle, valitaan ja perustellaan käytettävät tutkimusmenetelmät ja esitetään empiirisen tutkimuksen eteneminen alusta loppuun. Luku 6 sisältää

empiirisen tutkimuksen tulokset ja tulosten analysoinnin. Tulosten analysointi keskittyy esiintyviin ongelmiin ja ratkaisumalleihin. Luvussa käydään ensin tulokset läpi yksityiskohtaisesti esitetty kysymys kerrallaan ja lopuksi tärkeimmät havainnot kootaan luvun päättävään yhteenvetoon. Luvussa 7 käsitellään koko tutkielmaa koskevat johtopäätökset ja yhteenveto. Luku koostuu tutkielman tavoitteiden kertaamisesta, kirjallisuuskatsauksen ja empiirisen tutkimuksen yhteenvedosta ja perinteisestä johtopäätökset -osiosta. Johtopäätökset sisältävät tutkielmassa saavutetut keskeiset tulokset, empiirisen tutkimuksen luotettavuuden arvioinnin ja jatkotutkimusaiheet.

2 OHJELMISTOTUOTTAJA

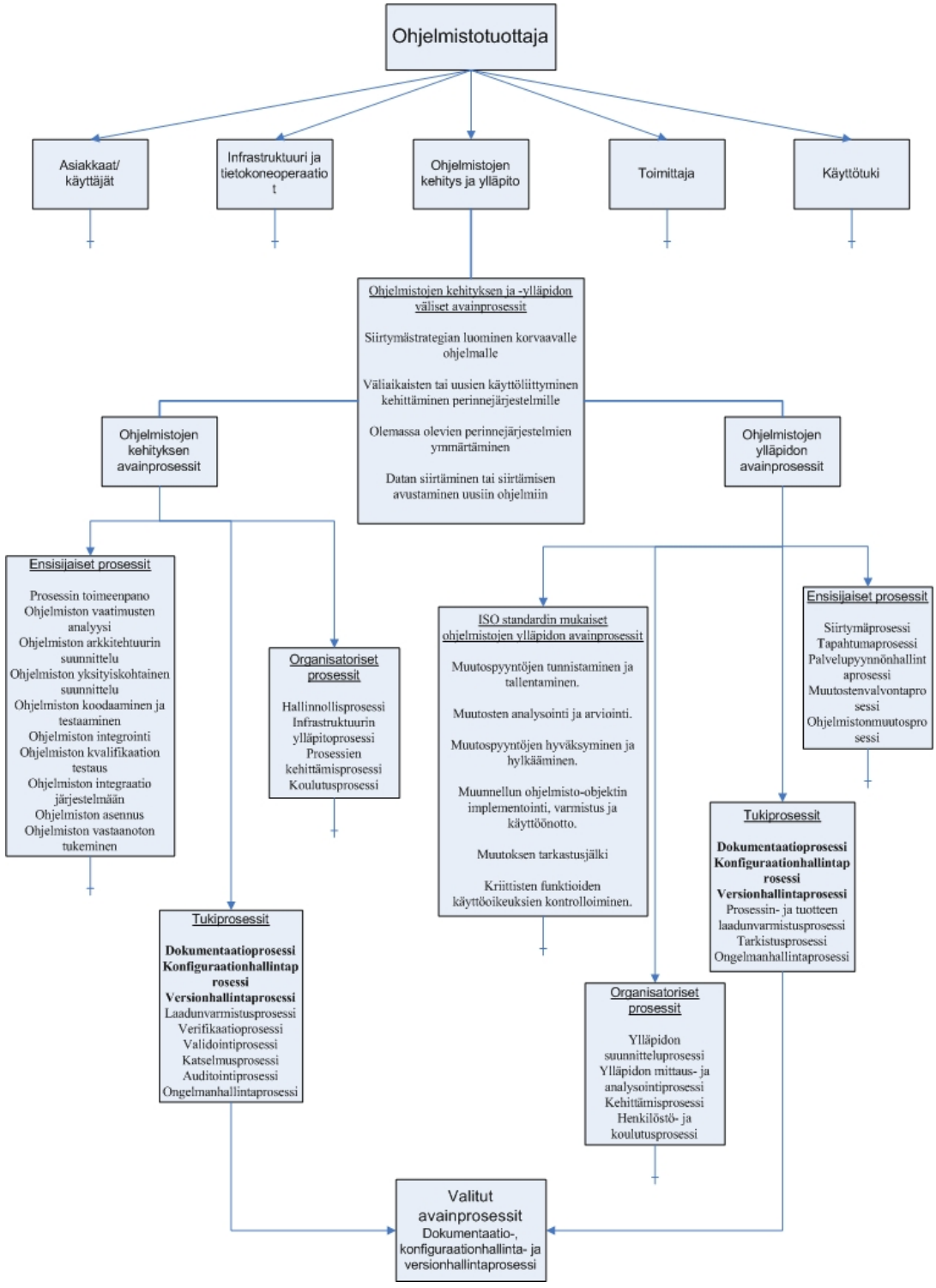
Tässä luvussa keskitytään ohjelmistotuottajan toimialaan. Kohdassa 2.1 esitetään tiivistetysti tässä luvussa tarkasteltavat osa-alueet. Kohdassa 2.2 kuvataan ohjelmistotuottajan nykyistä tilannetta toimialallaan, kohta myös selventää vertailukohteen sopivuutta tutkimuskohteeseen. Kohdassa 2.3 käydään läpi pääosin ohjelmistotuottajan tehtäväalueet ja tarkennetaan mille organisaation tehtäväalueille tutkielma suuntautuu. Kohdassa siis luokitellaan tärkeimmät tehtäväalueet ja rajataan tutkimuskohdetta. Rajauksena toimii se peruste, että tutkielma kohdentuu ohjelmistotuotteen konfiguraation- ja versionhallintaan. Luvun viimeisissä kohdissa syvennytään niihin organisaation tehtäväalueisiin, jotka valittiin tarkasteltaviksi kohdassa 2.3. Näitä tehtäväalueita ovat ohjelmistojen ylläpito ja kehitys. Myös kyseiset kohdat toimivat rajaavina, sillä kohtien avulla tarkennetaan ne tehtäväalueisiin liittyvät avainprosessit, joita tullaan tarkastelemaan luvussa 3. Tämän luvun kohdissa ei siis syvennytä tarkemmin avainprosesseihin, vaan syvällisempi tarkastelu tutkielman kannalta oleellisimpiin avainprosesseihin tehdään seuraavassa luvussa. Kohdan 2.4 tarkoitus on selventää, mitä avainprosesseja ohjelmistojen ylläpitoon kuuluu ja rajata, mitä niistä tarkastellaan tutkielmassa. Kohdassa 2.5 taas käydään läpi ohjelmistojen kehityksen avainprosessit ja rajataan tutkielman kannalta niistä oleellisimmat. Ohjelmistojen kehityksen avainprosessit ovat osittain yhteneväiset ohjelmistojen ylläpidon avainprosessien kanssa. Yhteneväiset avainprosessit jätetään kuvaamatta kohdassa 2.5, sillä kyseiset avainprosessit ovat kuvattu edellisessä kohdassa. Kohdassa 2.6 koostetaan yhteen luvun tulokset. Toisin sanoen luvussa esitetään, perusteluiden kera, tarkasteltaviksi valitut avainprosessit.

2.1 Tarkasteltavat osa-alueet

Ohjelmistotuottajalla tarkoitetaan yritystä, joka valmistaa ohjelmistotuotteen. Tällaisia yrityksiä ovat esimerkiksi Adobe Systems (http://www.adobe.com/) ja Microsoft (http://www.microsoft.com/). Ohjelmistotuottajaa tarkastellaan erityisesti konfiguraation- ja versionhallinnan alueilta. Mobiilipelituotanto on valittu vertailukohteeksi, koska tutkielmassa tarkastellaan mobiilipeliteollisuutta pelien tuottajan näkökulmasta. Valinta perustuu siihen, että mobiilipelien tuottaja on myös periaatteessa ohjelmistotuottaja, tosin

huomattavasti tuoreemmalla toimialalla, jonka markkinoilla on oltava erittäin nopea-liikkeinen. Toisin sanoen sekä perinteinen ohjelmistotuottaja että mobiilipelien tuottaja ovat ohjelmistotuottajakentän osajoukkoja. Sellaiset tekijät, kuten laitteistovalmistajat ja käyttäjäorganisaatioiden tietohallinto rajataan pois tutkielmasta, koska taustaa haetaan nimenomaan mobiilipelien tuottajan konfiguraation- ja versionhallinnan ongelmiin.

Alla olevaan kuvioon (kuvio 1) on koottu tämän luvun kohdissa esille tulevat tehtäväalueet ja avainprosessit. Lisäksi kuviosta selviää seuraavaan lukuun jatkotarkasteluun valitut avainprosessit ja tämän luvun rakenne, joka on edetä yleisestä erityiseen. Kyseisen kuvion sisältö on koottu useista tieteellisistä lähteistä. Seuraavissa kohdissa, tehtäväalueiden ja avainprosessien käsittelyn yhteydessä, myös perustellaan miksi jokin kokonaisuus rajataan pois tai otetaan mukaan tutkielmaan. Rajaukset on esitetty kuviossa yhteyden lopettavalla poikkiviivalla, kuten on tehty esimerkiksi Asiakkaat/käyttäjät kokonaisuudessa. Tätä logiikkaa noudattamalla, kuvion alimpaan kokonaisuuteen (Valitut avainprosessit) saadaan kerättyä tutkielmaan valitut avainprosessit. Kyseisten avainprosessien tarkempi sisältö käydään läpi luvussa 3.



KUVIO 1. Ohjelmistotuottaja organisaationa ja tutkielman kannalta oleellisimmat avainprosessit organisaatiossa (Perustuen: April, Hayes, Abran & Dumke 2005; Conradi & Westfechel 1998; ISO 1995)

2.2 Ohjelmistotuottajien nykyinen tilanne

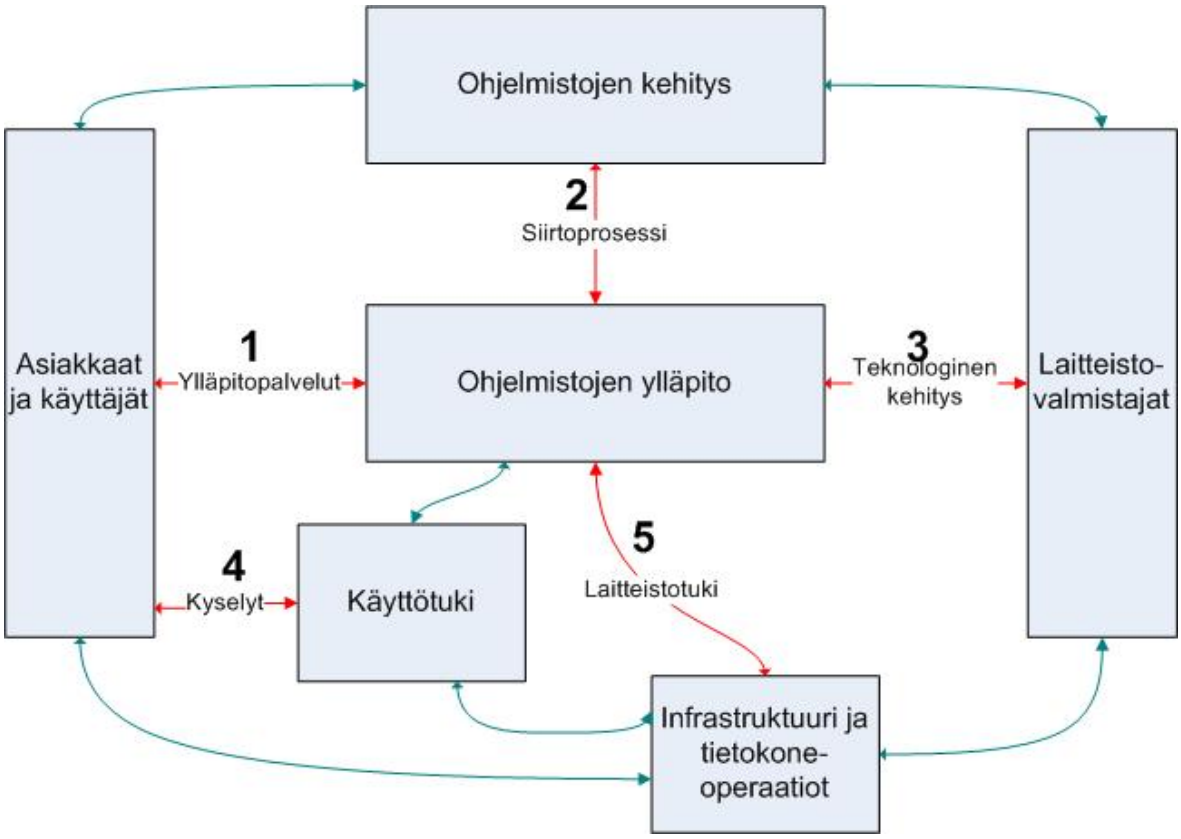
Tilanne ohjelmistotuottajien toimialalla on tiukentunut huomattavasti viime aikoina. Ohjelmistotuottajat ovat joutuneet kohtaamaan globalisaation tuoman kilpailun ja lisäksi ohjelmistotuottajien asiakkaat ovat entistä vaativampia, osaamistason noustua (April ym. 2005, 197). Ohjelmistotuottajien pitää pystyä tarjoamaan korkealaatuisia tuotteita ja palveluita, kilpailukykyiseen hintaan (April ym. 2005, 197). Aprilin ym. (2005, 197) mukaan, tästä ongelmasta selvitäkseen, yrityksen täytyy pystyä tarjoamaan asiakkaan vaatimusten mukaisia tuotteita/palveluita, lisäksi yrityksellä täytyy olla omia liiketoimintaprosesseja tukevia ohjelmistoja. Tukemalla omia liiketoimintaprosesseja voidaan muun muassa ennaltaehkäistä turhaa uudelleen rakentamista ja tukea kommunikointia (Forte 1997, 120).

Riskit liittyvät oleellisena osana nopeasti kehittyviin aloihin, kuten ohjelmistotuotantoon. Ohjelmistotuotantoon liittyvistä riskeistä voidaan kuitenkin selvitä aktiivisella muutoksenhallinnalla (Forte 1997, 120). Muutoksenhallinnalla Forte (1997) tarkoittaa ohjelmistojen kehityksen ja ylläpidon hallintaa. Edellä mainittujen tekijöiden pohjalta voidaan päätellä, että ohjelmistotuottajien nykyinen tilanne vaatii liiketoimintaprosessien tukemista. Liiketoimintaprosesseilla tarkoitetaan organisaation ensisijaisia prosesseja ja, näitä tukevia, tukiprosesseja ja organisatorisia prosesseja (April ym. 2005, 204; ISO 1995).

2.3 Ohjelmistotuottaja organisaationa

Ohjelmistotuottajan organisaatiorakenteella tarkoitetaan ihmisistä ja muista ohjelmistotuotannollisia avainprosesseja suorittavista resursseista koostuvaa, muun muassa tehtäväalueiden välisiä suhteita kuvaavaa, konfiguraatiota (Seaman 1993, 315). Organisaatiorakenne saadaan esitettyä selkeästi graafisessa muodossa organisaatiomallin avulla (Seaman 1993, 316). Alla oleva kuvio (kuvio 2) on esimerkki organisaatiomallista, tosin kuvio käsittelee vain ohjelmistojen ylläpidon kanssa toimivia tehtäväalueita. Kuvion tarkoitus on selventää tyypillisen ohjelmistotuottajan organisaatiorakennetta liittyen ohjelmistojen kehitykseen ja ylläpitoon. Kuvion mukaisten tehtäväalueiden välisten rajapintojen sisältö tuodaan esille tärkeimpien avainprosessien muodossa. Tutkielman kannalta tärkeimmät tehtäväalueet ovat ohjelmistojen kehitys ja ohjelmistojen ylläpito. Kuviossa

olevat numerot esittävät rajapintoja eri tehtäväalueiden välillä. Rajapinnoilla tarkoitetaan tehtäväalueiden välisiä yhteyksiä, joiden avulla esitetään tehtäväalueiden kanssakäymiseen liittyvät avainprosessit. Rajapinnat ovat oleellisia ohjelmistotuottajalle, sillä esimerkiksi ohjelmistojen ylläpitoon liittyy, omien sisäisten avainprosessien lisäksi, muiden tehtäväalueiden kanssa yhteistoimintaa vaativia avainprosesseja.



KUVIO 2. Organisaatiomalli ohjelmistojen ylläpidolle (April ym. 2005, 199)

Seuraavaksi käydään läpi kaikki kuviossa olevat tehtäväalueet ja oleellisimmat tehtäväalueiden väliset rajapinnat. Lisäksi perustellaan, miksi tarkastelun alla oleva tehtäväalue tai rajapinta jää pois tai tulee mukaan jatkotutkimukseen. Rajapinnat **1** (asiakkaat ja käyttäjät) ja **3** (laitteistovalmistajat) toimivat usein myös organisaation rajoina, sillä vain ohjel-

mistojen kehitys, käyttötuki, infrastruktuuri ja operaatiot, sekä ohjelmistojen ylläpito, ovat yleensä organisaation sisäisiä tehtäväalueita.

Rajapinta **1** koskee ohjelmistojen ylläpidon ja organisaation asiakkaiden/ohjelmiston käyttäjien välistä suhdetta. Kyseinen rajapinta sulkeutuu lähes kokonaan gradun tutkimusalueen ulkopuolelle, sillä siihen ei suoranaisesti liity ohjelmistotuotteen konfiguraation- tai versionhallintaa. Tästä syystä kyseisestä rajapinnasta tarkastellaan vain oleellisimmat piirteet. Tyypillisesti rajapintaan kuuluvat avainprosesseina ohjelmistojen ylläpitopalvelut. Kyseiset ylläpitopalvelut dokumentoidaan yleensä palvelutasosopimukseen (Service Layer Agreement) (April ym. 2005, 200).

Rajapinta **2** koskee ohjelmistojen kehityksen ja -ylläpidon välistä suhdetta ja on tärkein rajapinta gradun tutkimusalueen kannalta. Kyseistä rajapintaa ei ole kaikissa ohjelmistotuottajaorganisaatioissa. Tämä johtuu siitä, että joissakin organisaatioissa ei ole erikseen ohjelmistojen kehittäjiä ja ylläpitäjiä, vaan tietyn ohjelmiston kehittäjät jatkavat kyseisen ohjelmiston ylläpitäjinä (April ym. 2005, 200). Kun tämä rajapinta on olemassa, suuri osa ylläpidon ongelmista voidaan jäljittää ohjelmistojen kehitykseen asti (April ym. 2005, 201). Suurimmat ongelmat johtuvat siitä, että ylläpitäjät eivät pääse tarpeeksi aktiivisesti mukaan ohjelmiston kehitykseen ja siirtoprosessiin kehityksestä ylläpitoon. Aprilin ym. (2005, 200) esittämistä neljästä tärkeästä avainprosessista kaikkiin liittyy olennaisesti konfiguraation- tai versionhallinta. Kyseiset avainprosessit käsittelevät ohjelmistojen kehityksen ja -ylläpidon välistä suhdetta. Avainprosesseja ovat:

- siirtymästrategian luominen korvaavalle ohjelmalle
- väliaikaisten tai uusien käyttöliittyminen kehittäminen perinnejärjestelmille
- olemassa olevien perinnejärjestelmien ymmärtäminen
- datan siirtäminen tai siirtämisen avustaminen uusiin ohjelmiin.

Laitteistovalmistajan (supplier) ja ohjelmistotuottajaorganisaation välisellä rajapinnalla, eli rajapinnalla **3**, ei ole merkitystä tarkasteltaviin konfiguraation- tai versionhallintaan. Raja-

pinta pitää sisällään muun muassa alihankkijat, jotka auttavat ohjelmistojen ylläpito-prosesseissa tai suorittavat ylläpitopalveluita organisaatiolle.

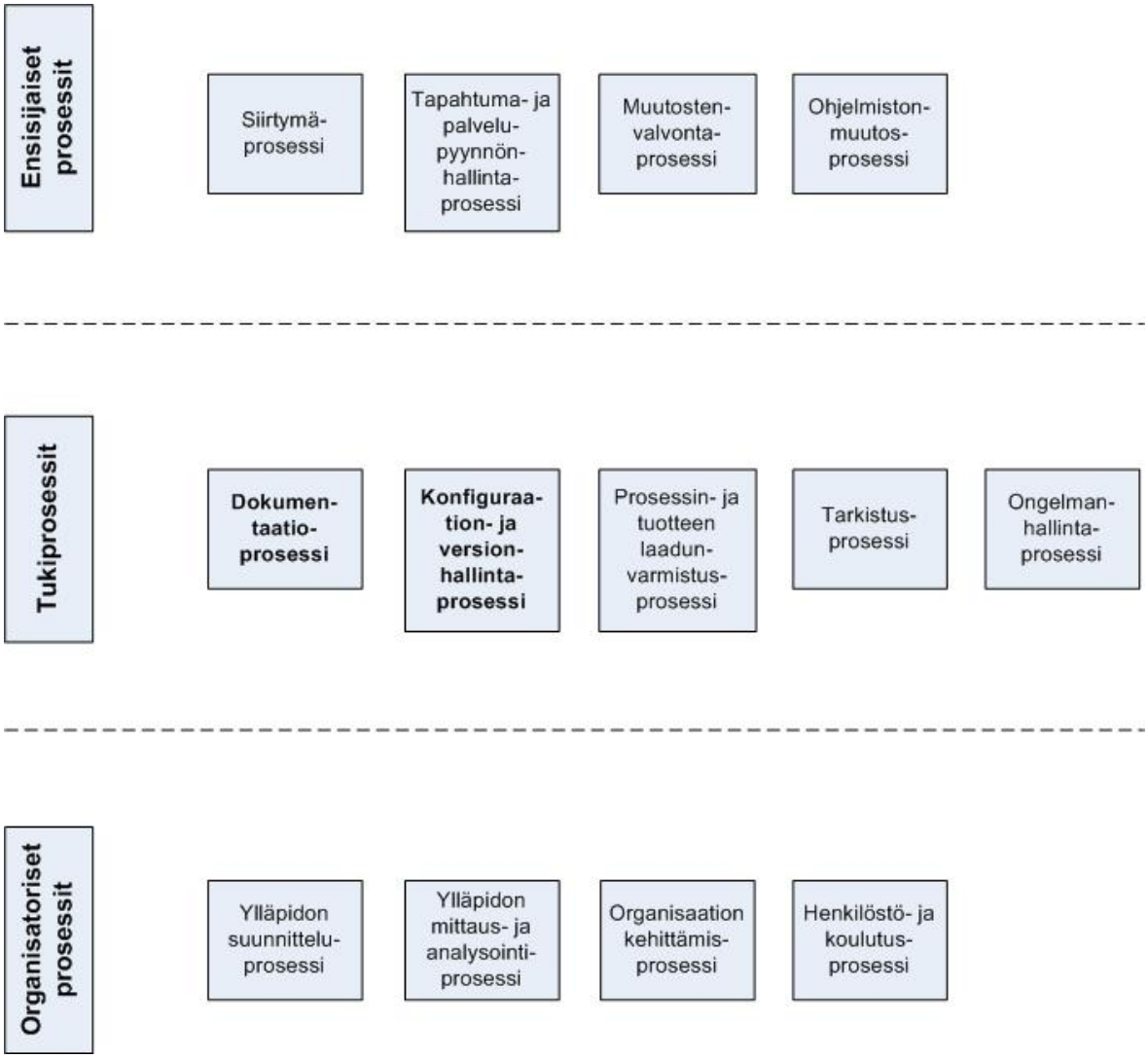
Rajapinta **4** on samantyyppinen kuin rajanpinta **1**, sillä se kuvaa käyttäjän ja käyttötuen välistä suhdetta. Vaikka käyttäjän ja käyttötuen välinen suhde usein synnyttää muutos-ehdotuksia, kuten rajapinta **1**, niin se ei suoranaisesti liity tarkasteltaviin konfiguraation- tai versionhallintaan.

Rajapinta **5** edustaa infrastruktuurin ja tietokoneoperaatioiden ja ohjelmistojen ylläpidon välistä rajapintaa. Aprilin ym. (2005, 200) mukaan sen rooli on toimia ohjelmistoa tukevan infrastruktuurin vartijana. Tehtäviin kuuluu muun muassa hoitaa kaikki operaatiot, tuki, ja ylläpito liittyen työpisteisiin, verkkoihin ja alustoihin (April ym. 2005, 200). Kyseinen rajapinta koskee laitteistoihin liittyviä tehtäviä, joten gradun tutkimusalueen kannalta rajapinnan tarkempi tarkasteleminen ei ole tarpeen.

Tässä kohdassa tehdyn rajauksen perusteella ohjelmistojen kehitys ja ylläpito ovat tärkeimmät tehtäväalueet tutkielman kannalta. Lisäksi näiden tehtäväalueiden välinen rajapinta on oleellinen osa tutkimusaluetta. Tästä johtuen seuraavissa kohdissa syvennytään valittuihin tehtäväalueisiin yksityiskohtaisemmin ja tuodaan muun muassa esille tehtäväalueiden sisältämiä avainprosesseja. Avainprosessit tarkentavat, mitä kyseinen tehtäväalue pitää sisällään.

2.4 Ohjelmistojen ylläpidon avainprosessit

Ohjelmistojen ylläpidon sisältö on koostettu pääosin Aprilin ym. (2005), Reichenbergerin (1989) ja ISO standardin (1995) sisältämistä määritelmistä. Kuvio 3 (April ym. 2005, 204) antaa hyvän kokonaiskuvan ohjelmistojen ylläpidon avainprosesseista. April ym. (2005) eivät tosin tarkastele kuvion sisältöä tarpeeksi syvällisesti, joten tutkielmaan mukaan rajattujen avainprosessien tarkempi sisältö on etsitty muista, samaa aihetta käsittelevistä, artikkeleista.



KUVIO 3. Ylläpidon avainprosessien luokittelukaavio (April ym. 2005, 204)

April ym. (2005, 204) jakavat avainprosessit kolmeen luokkaan. Ensisijaiset prosessit, joita April ym. (2005, 204) kutsuvat myös operationaalisiksi prosesseiksi, sisältävät siirtymäprosessin, tapahtuma- ja palvelupyynnönhallintaprosessin, muutostenvalvonta-prosessin ja ohjelmistonmuutosprosessin. Ensisijaiset prosessit eivät sisällä konfiguraation- tai versionhallintaa, mutta kyseiset prosessit aiheuttavat usein muutoksia ohjelmistoon ja näiden muutoksien hallinnan tukena on konfiguraation- ja versionhallinta. Ensisijaiset

prosessit eivät kuulu tutkielman tutkimusalueeseen, mutta tutkimusalueeseen kuuluvien tukiprosessien tavoite on tukea ja kehittää ensisijaisten prosessien toimivuutta.

Tukiprosessit sisältävät dokumentaatioprosessin, konfiguraationhallinta- ja versionhallinta-prosessin, prosessin- ja tuotteen laadunvarmistusprosessin, tarkistusprosessin ja ongelmanhallintaprosessin. Kyseinen luokka on oleellisin luokka tutkielman kannalta. Tutkielmassa syvennytään nimenomaan luokassa oleviin dokumentaatio-, konfiguraation-hallinta- ja versionhallintaprosesseihin. Esimerkiksi Reichenberger (1989, 137) käsittelee artikkelissaan versionhallintaa konfiguraationhallinnan alaisena prosessina, mutta tutkielmassa pidättäydytään muun muassa Aprilin ym. (2005, 204) määrityksessä, jossa konfiguraationhallinta ja versionhallinta ovat erillisiä avainprosesseja.

Organisatoriset prosessit eivät ole tutkielman kannalta oleellisia. Myös April ym. (2005, 205) mainitsevat, että kyseiset prosessit tarjoavat yleensä muut osastot. Kuitenkin organisatorisilla prosesseilla on vaikutus konfiguraation- ja versionhallintaan, sillä muun muassa ylläpidon suunnitteluprosessin avulla kehitetään tukiprosesseja. Organisatorisiin prosesseihin kuuluu myös ylläpidon suunnitteluprosessia tukeva ylläpidon mittaus- ja analysointiprosessi, organisaation kehittämisprosessi ja henkilöstö- ja koulutusprosessi.

ISO standardin (1995) esille tuomat avainprosessit ovat eri näkökulmasta kuin April ym. (2005, 205) mainitsemat avainprosessit. ISO standardin (1995) mukaan ohjelmistojen ylläpitoon kuuluvat seuraavat avainprosessit:

- muutospyyntöjen tunnistaminen ja tallentaminen
- muutosten analysointi ja arviointi
- muutospyyntöjen hyväksyminen ja hylkääminen
- muunnellun ohjelmisto-objektin implementointi, varmistus ja käyttöönotto
- muutoksen tarkastusjälki, jonka avulla jokainen muutos, muutoksen syy ja muutoksen hyväksyjä voidaan jäljittää

- kriittisten funktioiden käyttöoikeuksien kontrolloiminen.

ISO standardi esittää avainprosessit loogisessa järjestyksessä ja selkeinä kokonaisuuksina, mutta se ei tarkenna ollenkaan sitä, miten näistä avainprosesseista selvitään. Lisäksi kokonaisuudet ovat niin laajoja, että ne voivat kuvata sekä ohjelmistojen kehitystä että ohjelmistojen ylläpitoa. ISO standardin mukainen lista tuo kuitenkin hyvin esille sen, mistä ohjelmistojen ylläpidossa on kyse.

Ohjelmistojen ylläpidosta rajautui kolme tutkielman kannalta oleellista avainprosessia: konfiguraationhallinta, versionhallinta ja dokumentointi. Kaikki valitut avainprosessit kuuluivat tukiprosessien alaisuuteen.

2.5 Ohjelmistojen kehityksen avainprosessit

Ohjelmistojen kehitykseen kuuluvat avainprosessit on otettu ISO standardista (ISO 1995). Kyseinen standardi on tehty ohjelmistojen elinkaaren avainprosessien kuvaamista varten. Standardi käyttää samankaltaista avainprosessien kolmeen luokkaan jakamista kuin April ym. (2005, 204).

Ensisijaiset prosessit eivät ole tutkimusongelman kannalta oleellisia, joten niitä ei listata tässä kohdassa. Standardin mukaiset (ISO 1995) ohjelmistojen kehitykseen liittyvät ensisijaiset prosessit löytyvät sen sijaan taulukkona (taulukko 8) liitteestä 2. Taulukossa on kaikki ensisijaiset prosessit liittyen ohjelmistojen kehitykseen. Epäoleellisuuden takia listasta on karsittu muun muassa hankinta-, toimitus-, operaatio- ja ylläpitämisprosessit. Ensisijaiset prosessit eivät sisällä konfiguraation- tai versionhallintaa, mutta kyseisten prosessien tukena on konfiguraation- ja versionhallinta.

Ohjelmistojen kehitykseen kuuluu myös tukiprosessit ja organisatoriset prosessit, kuten ohjelmistojen ylläpitoonkin. Tukiprosesseista monet ovat luonteeltaan samankaltaisia kuin ohjelmistojen ylläpidon puolella olevat tukiprosessit. Tukiprosesseihin kuuluu ISO standardin (1995) mukaan dokumentaatio-, konfiguraationhallinta-, laadunvarmistus-, verifikaatio-, validointi-, katselmus-, auditointi- ja ongelmanhallintaprosessi. Lisäksi tukiprosesseihin voidaan laskea versionhallinta, sillä Conradi ja Westfechtel (1998)

käsittelevät artikkelissaan versionhallintaa ja kyseisen artikkelin mukaan myös ohjelmistojen kehityksessä on versionhallinta yhtenä tukiprosessina. Tukiprosesseista tutkielman kannalta oleellisimmat ovat dokumentaatio-, konfiguraationhallinta- ja versionhallintaprosessit.

Organisatorisiin prosesseihin kuuluu hallinnollis-, infrastruktuurin ylläpito-, prosessien kehittämis- ja koulutusprosessi (ISO 1995). Ohjelmistojen kehityksen organisatoriset prosessit eivät ole tutkielman kannalta oleellisia, kuten eivät ole ohjelmistojen ylläpidon organisatoriset prosessitkaan. Täten kyseiset prosessit jäävät pois tarkemmasta tarkastelusta.

Ohjelmistojen kehityksestä tutkielman kannalta oleellista avainprosesseja löytyi kolme, konfiguraation- ja versionhallinta sekä dokumentointi. Kaikki valitut avainprosessit kuuluivat tukiprosessien alaisuuteen.

2.6 Tarkasteltavat avainprosessit

Tähän mennessä on käsitelty ohjelmistotuottajien nykyistä tilannetta, organisaatiorakennetta oleellisten tehtäväalueiden osalta ja näiden tehtäväalueiden avainprosesseja. Avainprosesseista on valittu tutkielman kannalta oleellisimmat, eli dokumentaatio, versionhallinta- ja konfiguraationhallinta. Kyseiset avainprosessit löytyvät sekä ohjelmistojen kehityksen että ylläpidon puolelta. Vaikka ohjelmistojen kehitys ja ylläpito eroavat toisistaan, niin tarkasteltavat avainprosessit eivät muutu oleellisesti ohjelmistojen kehityksen ja ylläpidon välillä (Abran, Bourque, Dupuis & Moore 2004). Tästä johtuen luvussa 3 tarkastellaan valittuja avainprosesseja sekä ohjelmistojen kehityksen että ylläpidon kannalta erottamatta niitä omiksi kohdiksi.

Valitut avainprosessit kuuluvat tukiprosessien alaisuuteen. Lisää sisältöä tukiprosesseista löytyy muun muassa ITIL kirjasarjasta teoksesta Service Support (ITIL 2005), mutta näkökanta kyseisessä teoksessa on tutkielman kannalta epäsopiva. Teoksessa tarkastellaan palveluiden tarjoajan ja -ylläpitäjän näkökannalta tukiprosesseja, kun tutkielmassa tarkastellaan tukiprosesseja ohjelmistotuottajan näkökannalta (ITIL 2005, 4). Kyseistä teosta ei kuitenkaan voi kokonaan ohittaa, sillä siinä tarkastellaan monia ohjelmistotuottajan tuki-

prosessien kanssa yhteneviä aihealueita (esimerkiksi versiokirjaston rakennetta), joten yhteneviltä osin sisältöä on otettu myös kyseisestä teoksesta muiden tieteellisten lähteiden tueksi.

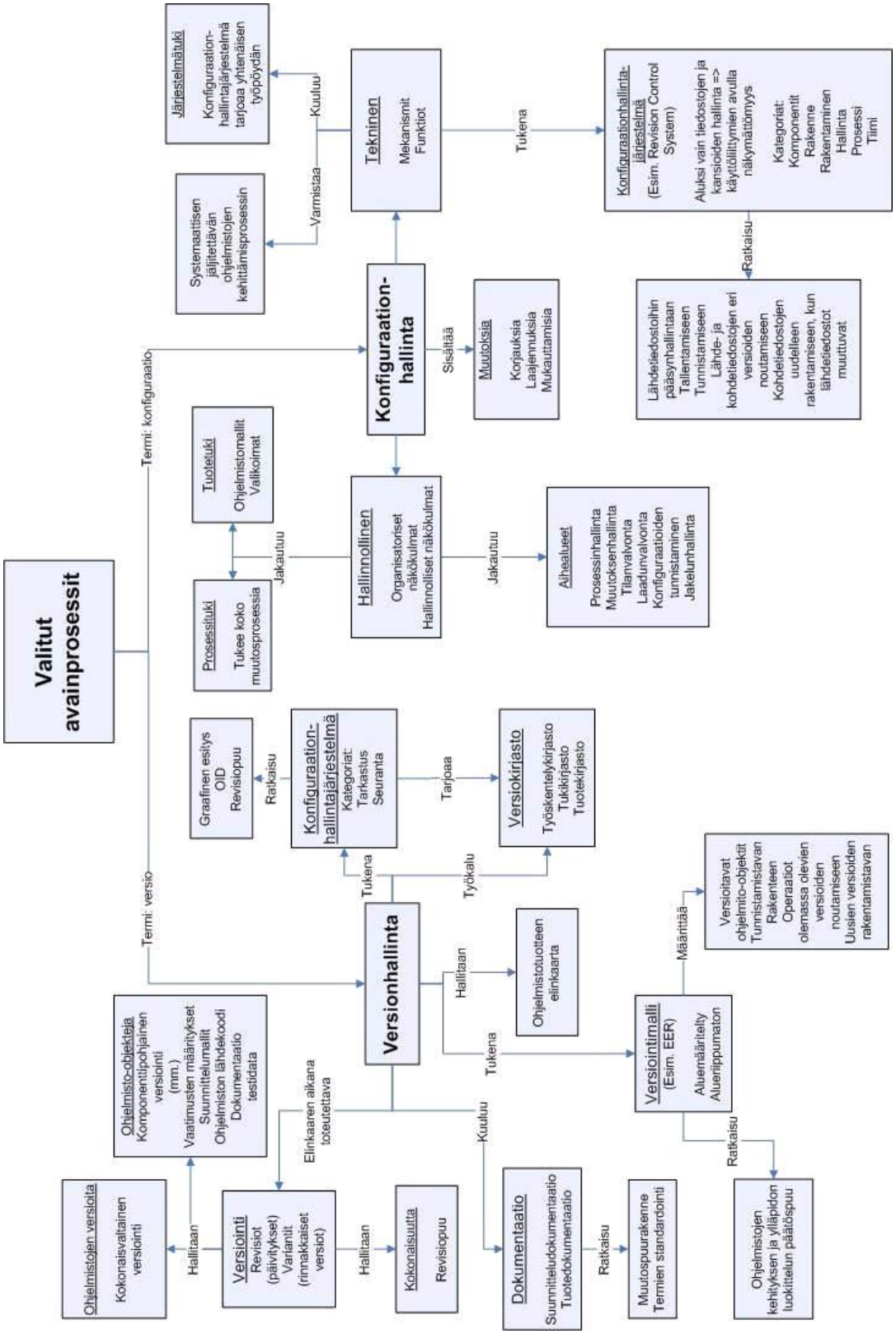
3 OHJELMISTOTUOTTEEN KONFIGURAATION- JA VERSIONHALLINTA

Luvun kohdissa syvennytään edellisessä luvussa rajattuihin avainprosesseihin. Tarkasteltavia avainprosesseja ovat siis konfiguraation- ja versionhallinta sekä dokumentointi. Kohdassa 3.1 esitetään tiivistetysti luvun sisältö erottamalla konfiguraationhallinta ja versionhallinta omiksi kokonaisuuksiksi. Avainprosessit jakautuvat kohtien 3.2 ja 3.3 alle seuraavasti: Kohta 3.2 käsittelee konfiguraationhallintaa ja kohta 3.3 versionhallintaa. Molemmissa kohdissa käsitellään kyseinen avainprosessi yhtenäisenä kokonaisuutena ohjelmistojen kehityksen ja ohjelmistojen ylläpidon osilta. Tämä johtuu lähtöolettamuksesta, että versionhallinnassa ja konfiguraationhallinnassa ei ole oleellista eroa ohjelmistojen kehityksen ja ohjelmistojen ylläpidon välillä (Abran ym. 2004). Lisäksi dokumentaatio avainprosessina on tarkasteltuna molemmissa kohdissa.

Alakohdassa 3.2.1 selvitetään mitä ovat konfiguraationhallintajärjestelmät ja mitkä ovat konfiguraationhallintajärjestelmien tärkeimmät ominaisuudet. Alakohdassa 3.2.2 käsitellään, miten konfiguraationhallinta toimii ohjelmistotuotannon tukena. Toisin sanoen alakohta 3.2.1 sisältää konfiguraationhallinnan teknisen näkökulman ja alakohta 3.2.2 konfiguraationhallinnan hallinnollisen näkökulman. Alakohdassa 3.3.1 keskitytään versionhallinnan osalta versiohistoriaan. Alakohdassa 3.3.2 selvitetään versiointimallin poistamia ongelmia ja tuomia etuja. Lisäksi alakohdassa käsitellään esimerkin avulla versiointimalleja. Alakohdassa 3.3.3 määritellään tyypillisimmät versiokirjastot ja käydään läpi tärkeimmät tekijät versiokirjaston suunnittelussa. Alakohdassa 3.3.4 käydään esimerkin avulla läpi ohjelmiston elinkaarta. Alakohta 3.3.5 käsittelee dokumentointiin liittyviä ongelmia terminologian osalta ja alakohdassa esitetään ongelmaan yksi mahdollinen ratkaisumalli. Alakohta 3.3.6 tuo esille sen, miten laaja konfiguraationhallintajärjestelmä voi tukea versionhallintaa.

3.1 Konfiguraation- ja versionhallinnan jako

Konfiguraationhallinta ja versionhallinta osoittautuivat laajoiksi kokonaisuuksiksi. Tästä syystä alla olevaan kuvioon (kuvio 4) on koostettu tämän luvun kohtien sisältö tärkeimmiltä osilta.



KUVIO 4. Valitut avainprosessit ja avainprosessien sisältö (Perustuen: taulukko 1)

Kuvioon on koottu kaikki oleelliset prosessit, termit, ongelmat ja ratkaisut, jotka tulevat esille tässä luvussa. Kaikki edellä mainitut kokonaisuudet on käyty tarkemmin läpi luvun kohdissa. Kuviosta selkenee myös valittujen avainprosessien rakenne graafisessa muodossa ja kuvio toimii pohjustuksena empiirisen tutkimuksen tutkimuskehykseen. Kuvion sisältö on koostettu useista lähteistä, selkeyden vuoksi lähteet on koostettu seuraavaan taulukkoon (taulukko 1), eikä lisätty kuvioon esimerkiksi jokaisen kokonaisuuden perään. Taulukosta löytyvät kuviossa olevat kokonaisuudet ja suhteiden määritykset. Lisäksi taulukosta selviää, millä sivulla tutkielmassa kokonaisuus on käsitelty ja mihin lähteisiin kyseinen kokonaisuus perustuu. Esimerkiksi ensimmäinen rivi: Versionhallinnalla hallitaan ohjelmisto-objektien ja ohjelmistojen versioita, käsitelty tutkielmassa sivulla 44, pohjautuen Conradiin ja Westfecheliin (1998) ja Westfecheliin, Bjorniin ja Conradiin (2001).

TAULUKKO 1. Avainprosessien lähteet

Avainprosessi	**Suhteen määritys**	**Kokonaisuus**	**Käsitelty tutkielmassa sivulla**	**Lähde**
Versionhallinta	Hallitaan	Ohjelmisto-objektien/Ohjelmistojen versioita	42	Conradi & Westfechtel 1998; Westfechtel ym. 2001
	Hallitaan	Kokonaisuutta	41	Conradi & Westfechtel 1998; Reichenberger 1989
	Hallitaan	Ohjelmistotuotteen elinkaarta	48	Eick ym. 2002; IEEE 1990
	Tukena	Versiointimalli	44	Conradi & Westfechtel 1998
	Ratkaisu	Päätöspuu	46	Chapin ym. 2001

(jatkuu)

TAULUKKO 1. (jatkuu)

	Ratkaisu	Muutospuurakenne	50	Reichenberger 1989
	Ratkaisu	Termien standar-dointi	50	Chapin ym. 2001
	Elinkaaren aikana toteutettava	Versiointi	40	Conradi & West-fechtel 1998; Reichenberger 1989
	Tukena	Konfiguraation-hallintajärjestelmä	52	Dart 1991; ITIL 2005
	Ratkaisu	Graafinen esitys	43	Conradi & West-fechtel 1998; Reichenberger 1989; Eick ym. 2002
	Työkalu	Versiokirjasto	46	Abran ym. 2004; Conradi & West-fechtel 1998; Dart 1991; IEEE 1990; ITIL 2005; Tichy 1985
	Ratkaisu	OID ja Revisiopuu	53	Conradi & West-fechtel 1998; Reichenberger 1989

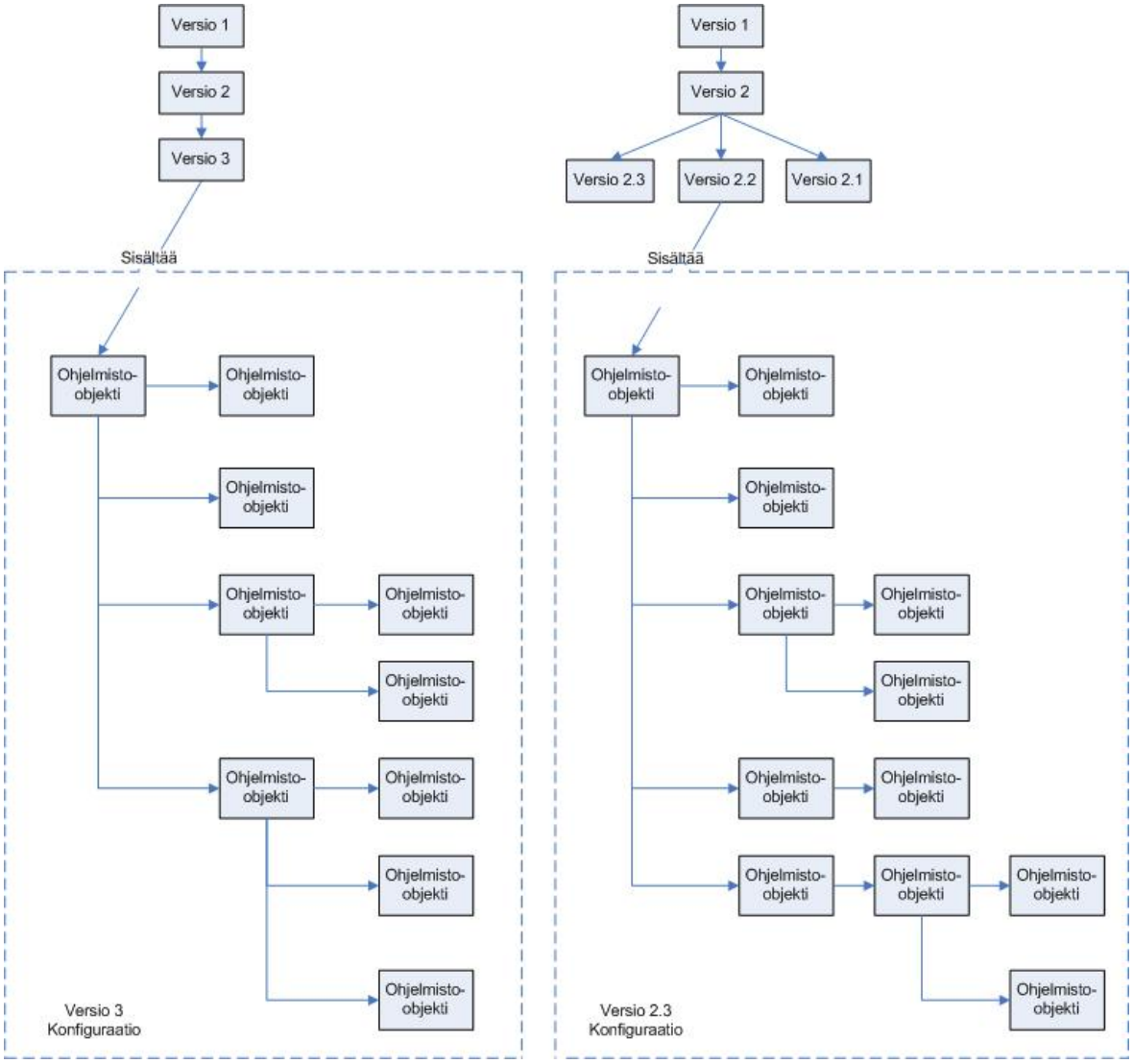
(jatkuu)

TAULUKKO 1. (jatkuu)

Konfiguraation-hallinta	Sisältää	Korjauksia, laajennuksia, mukauttamisia	31	Estublier ym. 2005
	Tukena	Konfiguraation-hallintajärjestelmä	33	Dart 1991
	Sisältää	Hallinnollinen näkökulma	34	Abran ym. 2004; Humphrey 1989; Joeris 1997; IEEE 1983 & IEEE 1988; Conradi & Westfechtel 1998
	Sisältää	Tekninen näkökulma	32	Humphrey 1989; Joeris 1997; Conradi & Westfechtel 1998
	Varmistaa	Systemaattisen jäljitettävän ohjelmistojen kehitysprosessin	31	Estublier ym. 2005
	Kuuluu	Järjestelmätuki	33	Estublier ym. 2005
	Jakautuu	Tuotetuki	39	Estublier ym. 2005; ITIL 2005
	Jakautuu	Prosessituki	39	Estublier ym. 2005; Bersoff ym. 1980 mukaan, Conradi & Westfechtel 1998
	Jakautuu	Aihealueet	35	IEEE 1983 & IEEE 1988 mukaan, Conradi & Westfechtel 1998; Abran ym. 2004

Konfiguraation- ja versionhallintaan liittyen löytyy paljon kirjallisuutta. Valitettavasti kyseisten termien eroavaisuuksia käsittelevää artikkelia ei ole julkaistu, joten termien välinen ero on johdettu useiden eri artikkelien määritysten perusteella. Termiä konfiguraationhallinta käytetään usein, kun tarkastellaan yksittäisiä muutoksia ja muutoksiin liittyviä hallinnollisia toimia, kuten muutosten hyväksymiskäytänteitä. Termiä versionhallinta taas käytetään, kun tarkastellaan muutostenhallintaa tasoa korkeammasta näkökulmasta. Eli versionhallinta tarkastelee muutostenhallintaa kokonaisuutena, kun taas konfiguraationhallinnan näkökulma on enemmän yksittäisten muutosten hallinnassa. Toisin sanoen jos esimerkiksi otetaan jonkin ohjelmiston konfiguraation- ja versionhallinta. Konfiguraationhallinta on kiinnostunut mistä ohjelmisto-objekteista kyseinen ohjelmisto koostuu, versionhallinta taas on kiinnostunut kyseisen ohjelmiston kehittymisestä eri versioihin. Sachweh ja Schafer (1995, 25) tuovat artikkelissaan esille kuvion (kuvio 5), joka käsittelee selkeästi ja yksinkertaisesti konfiguraationhallinnan ja versionhallinnan välistä suhdetta. Kyseinen artikkeli on varsin iäkäs ja täten terminologiassa on pieniä eroavaisuuksia. Esimerkiksi kuvion alaosaa Sachweh ja Schafer (1995) kutsuvat inkrementtitasoksi, mutta heidän määritys inkrementtitasosta vastaa esimerkiksi Ambriolan, Bendixin ja Ciancarinin (1990, 304) määritystä konfiguraationhallinnan kohdealueesta.

Kuvion yläosassa on esitetty kahden ohjelmistotuotteen versiopuut. Ohjelmistotuotteet ovat kehittyneet useiksi eri versioiksi, kuten näistä versiopuista selviää. Versionhallinta keskittyy tähän kuvion yläpuoliseen osaan, jota Sachweh ja Schafer (1995) kutsuvat versiotasoksi. Kuvion alaosaan on otettu ohjelmistotuotteiden versioiden 3 ja 2.3 konfiguraatiot esille. Konfiguraatioiden avulla saadaan esitettyä kyseisen ohjelmistotuotteen tietyn version sisältämät ohjelmisto-objektit ja ohjelmisto-objektien väliset suhteet. Konfiguraationhallinta keskittyy kuvion alapuoliseen osaan. Seuraavissa alakohdissa näitä kahta kokonaisuutta (kuvion ylä- ja alaosaa) tarkkaillaan omina kokonaisuuksina.



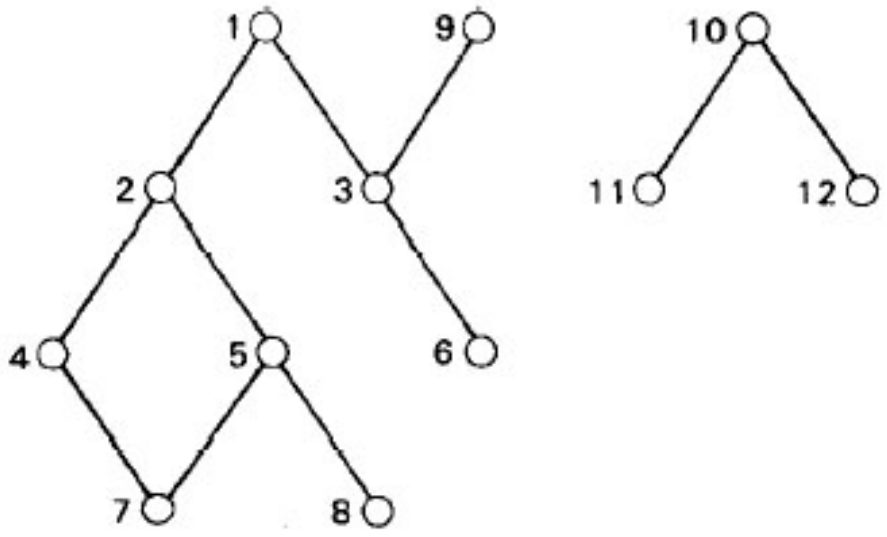
KUVIO 5. Ohjelmistotuotteen versiograafi (Sachweh & Schafer 1995, 25)

3.2 Ohjelmistotuotteen konfiguraationhallinta

Konfiguraationhallinta sai alkunsa 1950-luvulla, kun avaruusalusten tuotanto koki vaikeuksia riittämättömän suunnittelu- ja tuotedokumentaation johdosta (Estublier ym. 2005, 388). Ohjelmistotuotteiden puolella konfiguraationhallinta nousi esille erillisenä avainprosessina 1970-luvun lopulla, jolloin ohjelmistotuotteet alkoivat kärsiä samoista ongelmista kuin avaruusalusteollisuus 1950-luvulla (Estublier ym. 2005, 388). Tässä

tutkielmassa konfiguraationhallinnalla tarkoitetaan nimenomaan ohjelmistotuotteen konfiguraationhallintaa.

Konfiguraatio esittää tiettyyn ympäristöön tai käyttöön tarkoitetun ohjelmiston osat ja niistä koostuvan rakenteen (ATK-sanakirja 4.0, 2003). Konfiguraatiosta siis näkee mitä ohjelmisto-objekteja kyseiseen versioon ohjelmistosta tarvitaan ja ohjelmisto-objektien väliset riippuvuudet. Tästä esimerkkinä on alla oleva Ambriolan ym. (1990, 304) esittämä kuvio (kuvio 6). Kyseinen kuvio on yksinkertaistettu esimerkki konfiguraatiosta, joka koostuu ohjelmisto-objekteista 1–12. Konfiguraatiosta selviää, mitkä ohjelmisto-objektit ovat vuorovaikutuksessa toistensa kanssa. Esimerkiksi ohjelmisto-objektit 10–12 muodostavat ohjelmistoon oman kokonaisuuden. Lisäksi konfiguraatio helpottaa ohjelmistotuotteen kyseisen version vertaamista toiseen versioon, kun konfiguraatiosta näkee heti miltä osin niiden rakenne tai ohjelmisto-objektit eroavat.



KUVIO 6. Kuvaus konfiguraatiosta (Ambriola ym. 1990, 304)

Konfiguraationhallinnalla tarkoitetaan siis yksittäisen kokonaisuuden hallintaa ohjelmisto-objektien ja ohjelmiston rakenteen osalta, esimerkiksi kuvitteellisen ohjelmiston C version 2.3 osalta. Estublier ym. (2005, 387) tarkentavat artikkelissaan, että ohjelmistojen kokemat muutokset ovat useimmiten korjauksia, laajennuksia tai mukauttamisia, joita suoritetaan ohjelmiston elinkaaren aikana. Konfiguraationhallinnan tarkoitus on tukea ohjelmisto-tuotantoa, jotta ohjelmistotuotteeseen saadaan muutokset aikaan koordinoidusti (Babich 1986 mukaan, Conradi & Westfechtel 1998, 233). Estublier ym. (2005) määrittelevät artikkelissaan konfiguraationhallintaa samantyyppisesti: konfiguraationhallinnan keskeisenä tehtävänä on varmistaa systemaattinen ja jäljitettävä ohjelmistotuotantoprosessi. Kyseisessä prosessissa jokainen muutos täytyy olla konfiguraationhallinnan mukaisesti suoritettu, sillä näin järjestelmä säilyy hyvin määritellyssä tilassa koko ajan. (Estublier ym. 2005, 387.)

Konfiguraationhallinta määritellään avainprosessiksi, jonka avulla hallitaan monimutkaisen ohjelmiston tuottamista (Tichy 1988 mukaan, Westfechtel ym. 2001, 1111). Estublier ym. (2005, 384) taas kuvaavat konfiguraationhallintaa seuraavasti: sen avulla kontrolloidaan suurten ja monimutkaisten ohjelmistojen muutosta. Estublierin ym. (2005, 384) määritelmä on tärkeä, sillä siinä täsmennetään, että konfiguraationhallinnalla kontrolloidaan nimenomaan ohjelmistojen muutosta. Tämä tukee aikaisempaa versionhallinnan ja konfiguraationhallinnan välistä rajausta. Tichyn (1988) määritys taas koskee laajempaa kokonaisuutta, sillä sen mukaan konfiguraationhallinnalla tuetaan koko ohjelmisto-tuottamista. Konfiguraationhallinta on hankala määritellä yhdellä lauseella, sillä konfiguraationhallintaan kuuluu paljon erityyppisiä prosesseja. Tässä ja edellisessä kappaleessa esitetyt määritelmät tuovat esille tieteellisten lähteiden määritelmät konfiguraationhallinnasta kokonaisuutena.

Konfiguraationhallinta on tunnistettu suureksi osaksi hyvin määritettyä ohjelmistotuotanto-prosessia (Humphrey 1989 mukaan, Joeris 1997, 126). Tähän kuuluu ohjelmistotuotannon kannalta sekä tekninen, että hallinnollinen näkökulma. Tekninen näkökulma pitää sisällään mekanismit ja funktiot, joiden avulla hallitaan ohjelmistojen ja ohjelmisto-objektien konfiguraatioita. Näin saadaan rakennettua polveutuvat versiot ja yhtenäiset konfiguraatiot. (Joeris 1997, 126.) Hallinnollinen näkökulma taas keskittyy konfiguraationhallinnan

organisatorisiin ja hallinnollisiin näkökantoihin (Joeris 1997, 125). Conradin ja Westfechtelin (1998, 233) näkemyksen mukaan konfiguraationhallinta sisältää toisistaan riippuvaisten ohjelmisto-objektien yhtenäisyyden ylläpidon, tarvittaessa aikaisemmin rakennettujen versioiden uudelleenrakentamisen, polveutuvien ohjelmisto-objektien rakentamisen lähdekoodista (toisin sanoen kääntää ohjelmakoodista tietokoneen ymmärtämäksi ohjelmaksi) ja uusien versioiden rakentamisen. Kyseinen rajaus koskee varsin selkeästi konfiguraationhallinnan teknistä näkökulmaa. IEEE standardien (IEEE 1983 & IEEE 1988 mukaan, Conradi & Westfechtel 1998, 233) määritelmä konfiguraationhallinnasta sisältää ohjelmisto-objektien ja niiden versioiden tunnistamisen, muutoksenhallinnan, tilanvalvonnan ja laadunvalvonnan. Kyseinen IEEE standardien määritelmä ottaa selkeämmin kantaa konfiguraationhallinnan hallinnolliseen näkökulmaan. Konfiguraationhallinnan jakaminen kahteen osaan, hallinnolliseen ja tekniseen näkökulmaan, selkeyttää konfiguraationhallinnan sisältämiä prosesseja. Tästä syystä seuraavissa kahdessa alakohdassa on käsitelty konfiguraationhallinnan teknistä ja hallinnollista kantaa omina kokonaisuuksina.

3.2.1 Konfiguraationhallinta – tekninen näkökulma

Konfiguraationhallinnan työkalujen historia selittää suurelta osin sen, minkä tyyppisiä nykyiset työkalut ovat. Ensimmäiset konfiguraationhallinnan työkalut keskittyivät tiedostojen ja kansioiden hallintaan (Estublier ym. 2005, 387). Tämän takia suuri osa nykyisistäkin työkaluista pohjautuu itse tiedostojärjestelmän näkemiseen ja sitä kautta tiedostojen muokkaamiseen (Estublier ym. 2005, 387). Viime aikoina kehittyneimmät työkalut (esimerkiksi Eclipse) pyrkivät eroon tästä manuaalisuudesta muun muassa kehittyneiden käyttöliittymien avulla (Estublier ym. 2005, 388). Tämä kehityssuunta perustuu siihen, että työkaluista saadaan näkymättömiä ja siten ohjelmistotuotannossa voidaan keskittyä itse tekemiseen eikä työkalujen käyttöön (Introna & Edgar 1997, 35). Estublier ym. (2005, 391) toteavat, että melkein kaikki nykyiset konfiguraationhallinnan työkalut ovat riippumattomia muun muassa ohjelmointikielistä. Myös tämä tekijä edesauttaa näkymättömyyden saavuttamisessa, sillä käyttäjän ei tarvitse opetella laajaa valikoimaa eri ohjelmointikieliä pystyäkseen käyttämään eri työkaluja. Kehittyneimmissä työkaluissa on niin paljon ominaisuuksia, että niitä kutsutaan useimmin konfiguraationhallinta-

järjestelmiksi kuin konfiguraationhallinnan työkaluiksi. Tästä johtuen käytän tutkielmassa ainoastaan termiä konfiguraationhallintajärjestelmä.

Teknisestä näkökulmasta Estublier ym. (2005, 392) nostavat esille järjestelmätuen. Järjestelmätuki tarkoittaa sitä, että konfiguraationhallintajärjestelmä antaa yhtenäisen työpöydän, jossa kehittäjä/ylläpitäjä voi tehdä tarvittavat tehtävät, käyttäen joko konfiguraationhallintajärjestelmän ominaisuuksia tai ulkoisia työkaluja. Lisäksi esimerkiksi hajautetuissa järjestelmissä ei tarvitse keskittyä siihen, kenellä on uusin versio jostakin tietystä ohjelmisto-objektista tai miten muokattavana ollut ohjelmiston-objekti palautetaan takaisin konfiguraationhallintajärjestelmään työn jälkeen, vaan järjestelmä hoitaa tämän kaiken, viemättä liikaa ohjelmistotuotantoon varattua aikaa. (Estublier ym. 2005, 392.)

Konfiguraationhallintajärjestelmän tarkoitus on säilyttää ja hallita ohjelmistotuotannon aikana tehtyjä muutoksia. Kyseisillä järjestelmillä haetaan ratkaisua muun muassa lähdetiedostoihin pääsynhallintaan, tallentamiseen, tunnistamiseen, lähde- ja kohdetiedostojen eri versioiden noutamiseen ja kohdetiedostojen uudelleen rakentamiseen, kun lähdetiedostot muuttuvat. (Korel ym. 1991, 161.) Esimerkki konfiguraationhallintajärjestelmästä on Unixissa toimiva Revision Control System (Tichy 1982 mukaan, Korel ym. 1991, 161).

Dart (1991, 7) esittää artikkelissaan konfiguraationhallinnan kannalta 6 tärkeää kategoriaa. Kategoriat kuvaavat käyttäjän tarpeita konfiguraationhallintajärjestelmältä. Nämä kategoriat on lueteltu alla ja jokaisesta on lisäksi kevyt kuvaus:

- Ohjelmisto-objektit. Konfiguraationhallintajärjestelmä tukee käyttäjiä ohjelmisto-objektien tunnistamisessa, tallentamisessa ja käyttämisessä.

- Rakenne. Konfiguraationhallintajärjestelmän avulla pystytään selvittämään miten eri ohjelmisto-objektit liittyvät toisiinsa.

- Rakentaminen. Konfiguraationhallintajärjestelmän avulla kyetään rakentamaan ohjelmistotuote järjestelmän versioimista lähdetiedostoista. Lisäksi ohjelmisto-objektien vanhojen versioiden käyttö on oltava mahdollista.

- Hallinta. Konfiguraationhallintajärjestelmän avulla pystytään ymmärtämään muutoksen aiheuttamat vaikutukset ohjelmistotuotteeseen. Lisäksi järjestelmä tarjoaa vianseurantaan ja muutospyyntöjenkäsittelyyn tarvittavat työkalut.

- Prosessi. Konfiguraationhallintajärjestelmä auttaa käyttäjiä selviämään konfiguraationhallinnan prosesseista.

- Tiimi. Konfiguraationhallintajärjestelmä tukee sekä yksilö- että ryhmätyötä. Esimerkiksi ryhmätyötä tuetaan muun muassa konfliktien tunnistamisessa ja ratkaisemisessa.

Dartin (1991) määritys konfiguraationhallintajärjestelmästä vastaa laajaa ja kehittynyttä järjestelmää. Myös Estublier ym. (2005, 391) toteavat, että vasta viimeajan kehittyneimmät konfiguraationhallintajärjestelmät kykenevät vastaamaan kaikkien kategorioiden asettamiin haasteisiin. Konfiguraationhallintajärjestelmät ovat kehittyneet versioinnin tueksi rakennetuista kotitekoisista järjestelmistä, kaupallisiin moniin erityyppisiin ohjelmistotuotantoprosesseihin sopiviksi järjestelmiksi, jotka tukevat muun muassa ryhmätyötä ja hajautettua työskentelyä (Estublier ym. 2005, 389). Kaikki kategoriat eivät kuitenkaan ole aina tarpeellisia, esimerkiksi kategoria rakentaminen saatetaan hoitaa jollakin toisella työkalulla. Tällöin käyttötarpeisiin voi riittää jokin avoimen koodin (open source) järjestelmä. Yleisimmät avoimen koodin konfiguraationhallintajärjestelmät ovat CVS ja Subversion (Estublier ym. 2005, 385). Konfiguraationhallintajärjestelmän hallinta, prosessi ja tiimi -kategoriat eroavat versionhallinnan näkökulmasta. Konfiguraationhallinnan puolella jokaisen kategorian näkökulma painottuu yksittäisten muutosten tasolle. Toisin sanoen, kun konfiguraationhallintajärjestelmä tarjoaa hallinnan avuksi muun muassa muutosten valvonnan, vianseurannan ja muutospyyntökyselytyökalut, niin versionhallinnan puolella hallinta keskittyy koko ohjelmistotuotteen elinkaaren hallintaan.

3.2.2 Konfiguraationhallinta – hallinnollinen näkökulma

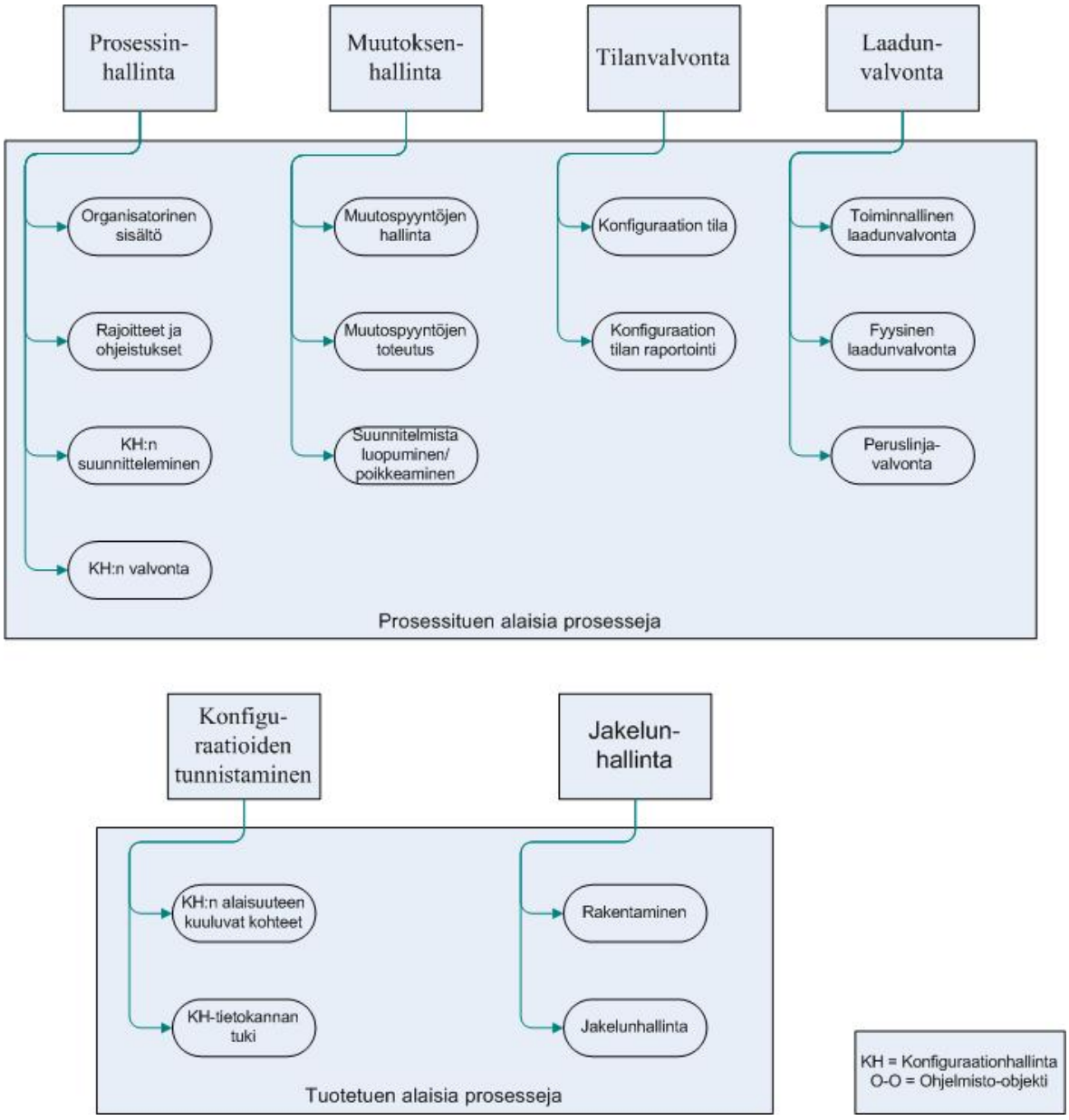
Abran ym. (2004) käsittelevät konfiguraationhallinnan sisältämiä prosesseja hallinnollisesta näkökulmasta. Toki kyseisestä artikkelista löytyy prosesseja myös tekniseltä kannalta, kuten konfiguraationhallintajärjestelmän implementointi ja ohjelmistotuotteen

rakentaminen, mutta näkökulma prosesseihin on vahvasti hallinnollinen. Abranin ym. määritelmää konfiguraationhallinnan prosesseista voidaan pitää erittäin luotettavana, sillä se on koostettu useista tieteellisistä lähteistä ja tehty yhteistyöllä IEEE Computer Societyn kanssa. Myös Estublierin ym. (2005, 392) artikkelissa käsitellään konfiguraationhallintaa hallinnolliselta kannalta, tosin hieman laajemmasta näkökulmasta, sillä artikkelissa hallinnollinen näkökulma jaetaan kahteen osaan, tuotetukeen ja prosessitukeen. Pääosin näiden kahden artikkelin pohjalta on koostettu alla oleva kuvio (kuvio 7), joka käsittelee konfiguraationhallinnan hallinnollista näkökulmaa.

Kuviossa konfiguraationhallinta jaetaan useisiin aihealueisiin. Aihealueita ovat prosessinhallinta, konfiguraation tunnistaminen, muutoksenhallinta, tilanvalvonta, laadunvalvonta ja jakelunhallinta. Jokainen aihealue sisältää useita eri prosesseja ja joillakin prosesseilla, kuten konfiguraationhallinnan suunnittelulla, on omia aliprosesseja. (Abran ym. 2004, 7 – 3.) Jokainen prosessi ja aliprosessi kuulu joko tuotetuen tai prosessituen alaisuuteen. Abranin ym. (2004) aihealueet ovat prosessinhallintaa ja jakelunhallintaa lukuun ottamatta yhteneväiset IEEE standardien määrityksen kanssa (IEEE 1983 & IEEE 1988 mukaan, Conradi & Westfechtel 1998, 233). Jakelunhallinnan puuttuminen kyseisistä IEEE standardeista johtuu standardien iästä, sillä jakelunhallintaan on keskitytty omana aihealueena vasta viimeaikoina.

Prosessinhallinta sisältää konfiguraationhallinnan suunnitteluun ja valvontaan liittyviä prosesseja. Organisatorista sisältöä suunnitellessa täytyy olla tiedossa organisatoriset elementit ja niiden väliset suhteet, sillä konfiguraationhallinta on vuorovaikutuksessa useiden muiden avainprosessien kanssa. Konfiguraationhallinnan rajoitteet ja ohjeistukset määrittelevät muun muassa sen, millä tarkkuustasolla konfiguraationhallintaa suoritetaan (Conradi & Westfechtel 1998, 235). Lisäksi rajoitteiden ja ohjeistusten avulla taataan yhtenäinen konfiguraationhallinta. Konfiguraationhallinnan suunnitteleminen sisältää vastuualueiden määrittämisen, resurssien ja aikataulujen kartoittamisen, työkalujen valinnan, alihankkijoiden konfiguraationhallinnan kartoittamisen ja kausaalisuudenhallinnan (Abran ym. 2004, 7 – 4; ITIL 2005, 233). Konfiguraationhallinnan suunnitteludokumentti on konfiguraationhallinnan suunnitteluprosessin tuotos. Konfiguraationhallinnan valvontaprosessin tarkoitus on valvoa suunnitteludokumentin toimivuutta ja noudattamista ja sitä

kautta hankkia pitkän tähtäimen tietoa prosessin toimivuudesta. (Abran ym. 2004, 7 – (2–5).)



KUVIO 7. Konfiguraationhallinnan aihealueet ja prosessit hallinnollisesta näkökulmasta (Abran ym. 2004, 7 – 3; Estublier ym. 2005, 392)

Muutoksenhallinta valvoo, nimensä mukaisesti, ohjelmistotuotteen elinkaaren aikana tapahtuvia muutoksia ohjelmistotuotteessa. Tähän kuuluu muutospyyntöjenhallinta, muutospyyntöjen arviointi ja muutospyyntöjen hyväksyntä. Laadukkaan konfiguraationhallinnan tuloksena muutosten vaikutukset pystytään ennakoimaan ja kustannukset arvioimaan tehokkaasti – kyseiset tekijät vaikuttavat osaltaan ohjelmistotuotannon tehokkuuteen (Abran ym. 2004, 7 – 7; ITIL 2005, 126). Lisäksi muutosten implementointi ja sovitusta muutoksesta luopuminen ja poikkeaminen kuuluvat muutoksenhallintaan. Muutosten implementoinnilla ei tarkoiteta varsinaista muutoksen toteutusta, vaan sen avulla hallitaan esimerkiksi useiden samanaikaisten muutosten toteutumista oikeisiin versioihin ja peruslinjoihin. (Abran ym. 2004, 7 – (7–8).)

Tilanvalvonta on hallinnolliselta kannalta tärkeä aihealue, sillä sen vastuulla on oleellisten tietojen kerääminen ja raportoiminen (IEEE 1988, 7). Tätä kautta konfiguraationhallinnan tehokas toiminta ja konfiguraatioiden pitäminen määritellyssä tilassa on mahdollista (Abran ym. 2004, 7 – 8; Estublier ym. 2005, 387). Tietojen kerääminen hoidetaan yleensä konfiguraationhallintajärjestelmän avulla automaattisesti, joten kyseinen prosessi säilyy näkymättömänä ohjelmistojen kehitykselle ja ylläpidolle. Raportoinnista on hyötyä useille eri organisaation tehtäväalueille, esimerkiksi projektin johdolle ja ohjelmistotuotantotiimeille. Raporteista projektin johto voi valvoa muun muassa ohjelmisto-objektien kokemien muutosten määrää ja muutoksiin tarvittua aikaa. (Abran ym. 2004, 7 – 8.)

Laadunvalvonnan avulla varmistetaan, että ohjelmistotuotteet ja ohjelmistotuotantoprosessit ovat yhdenmukaisia määriteltyjen sääntöjen, standardien, rajoitusten, ohjeistuksien ja suunnitelmien kanssa (IEEE 1997, 25). Konfiguraationhallinnassa laadunvalvonnan tarkoitus on valvoa fyysisten ja funktionaalisten ominaisuuksien toteutumista ennalta määritellyllä tasolla. Tästä johtuen laadunvalvonta konfiguraationhallinnassa on jaettu kahteen osaan, fyysiseen konfiguraation valvontaan (PCA) ja funktionaaliseen konfiguraation valvontaan (FCA). Ensiksi mainitussa huomio kohdistuu suunnittelu- ja viitedokumentaation yhdenmukaisuuteen ohjelmistotuotteen kanssa. Jälkimmäisessä valvotaan ohjelmisto-objektien ja spesifikaatioiden yhdenmukaisuutta. Esimerkiksi peruslinjojen tulisi läpäistä sekä fyysinen että funktionaalinen konfiguraation valvonta. (Abran ym. 2004, 7 – 9.)

Konfiguraatioiden tunnistamiseen kuuluu konfiguraationhallinnan alaisten ohjelmisto-objektien määrittäminen ja konfiguraationhallintatietokannan käyttö. Konfiguraation-hallinnan alaisuuteen määritetään joko kaikki tai vain tarpeelliset ohjelmisto-objektit (IEEE 1983 & IEEE 1988 mukaan, Conradi & Westfechtel 1998, 233; Abran ym. 2004, 7 – 6). Lisäksi itse ohjelmistotuotteen konfiguraatiot täytyy sisällyttää konfiguraation-hallinnan alaisuuteen. Konfiguraatioiden tunnistamiseen kuuluu oleellisena osana myös versioiden tunnistaminen, sekä ohjelmisto-objektien välisten suhteiden tunnistaminen (Abran ym. 2004, 7 – 6; Dart 1991, 7). Konfiguraatioiden tunnistamisen apuna toimii peruslinja. Peruslinjalla tarkoitetaan ohjelmistotuotteesta tietyssä pisteessä otettua konfiguraatiota, joka esittää sekä konfiguraation rakenteen että tarvittavat yksityiskohdat (ITIL 2005, 123). Tätä peruslinjaa voidaan käyttää jatkossa vertailukohtana peruslinja-valvonnassa, tarkkailtaessa esimerkiksi seuraavien versioiden konfiguraatioiden muutoksia. Konfiguraationhallintatietokannan täytyy tukea konfiguraatioiden tunnistamisen asettamia vaatimuksia. Tämän takaaminen kuuluu konfiguraationhallinnan tekniseen näkökulmaan, mutta valvominen on yksi konfiguraatioiden tunnistamiseen liittyvistä prosesseista. Edellä mainitut prosessit tuottavat perustan konfiguraationhallinnan toimivuudelle. (Abran ym. 2004, 7 – (6–7).)

Jakelunhallinnassa keskitytään ohjelmistotuotteiden ja -objektien julkaisemiseen liittyvään toimintaan. Jakelunhallinnan alaisia prosesseja ovat rakentaminen ja julkaisunhallinta (release management). Prosessilla rakentaminen ei tarkoiteta koodaamista, vaan konfiguraatiotietojen perusteella oikeanlaisen version rakentamista yhdistämällä ohjelmisto-objektit kokonaisuudeksi. Hallinnolliselta kannalta rakentamisprosessiin kuuluu rakentamisohjeiden luominen ja valvonta, sekä rakentamisprosessin toimivuuden valvonta Esimerkiksi pystytään rakentamaan ohjelmistotuotteesta aikaisemmin julkaistu konfiguraatio täysin identtisenä kokonaisuutena. Julkaisunhallintaprosessin kohdalla menee Abranin ym. (2004, 7 – 9) määritelmä osittain päällekkäin versionhallinnan kanssa. Versionhallinnan ollessa avainprosessina niin laaja, tutkielmassa pidättäydytään Aprilin ym. (2005, 204) määritelmässä jakamalla konfiguraation- ja versionhallinta omiksi avain-prosesseiksi. Konfiguraationhallinnan puolella julkaisunhallintaprosessi pitää sisällään

oikeanlaisten konfiguraatioiden tunnistamisen ja toimittamisen ohjelmistotuotannosta eteenpäin. (Abran ym. 2004, 7 – 9.)

Estublier ym. (2005, 392) purkavat artikkelissaan konfiguraationhallintaa yhtenä tuki-prosessina. Hallinnollisesta näkökulmasta konfiguraationhallinnan tarjoaman tuen he jakavat kahteen osaan, tuotetukeen ja prosessitukeen (Estublier ym. 2005, 392). Estublierin ym. edellä mainittu jako konfiguraationhallinnasta, hallinnollisesta näkökulmasta, on otettu huomioon kuviossa 7, sisällyttämällä Abranin ym. (2004) esittämät prosessit joko tuotetuen tai prosessituen alaisuuteen.

Tuotetuki pitää sisällään ohjelmistomallit ja valikoimat. Tämä sen takia, että projektin-hallinta tiedosto tiedostolta ei ole kovin tehokasta. Täten on tärkeää tukea ohjelmisto-tuotantoa erilaisten konfiguraatioiden luomisessa. Estublier ym. (2005) puhuvat yhdistelmä artefakteista konfiguraatioiden sijaan, mutta tutkielmassa käytetään sanaa konfiguraatio selkeyden vuoksi. Konfiguraatioiden avulla ohjelmistotuotantoprosessi selkeytyy varsinkin laajoissa projekteissa, sillä konfiguraatiosta selviävät myös ohjelmisto-objektien väliset suhteet ja ominaisuudet. (Estublier ym. 2005, 392.) Käytännön esimerkkinä tästä voisi pitää tilannetta, jossa mielikuvituksellista alemman tason ohjelmisto-objektia muokataan. Tällöin konfiguraationhallintajärjestelmä selvittää, mihin ylemmän tason ohjelmisto-objekteihin muokattu ohjelmisto-objekti vaikuttaa (ITIL 2005, 233).

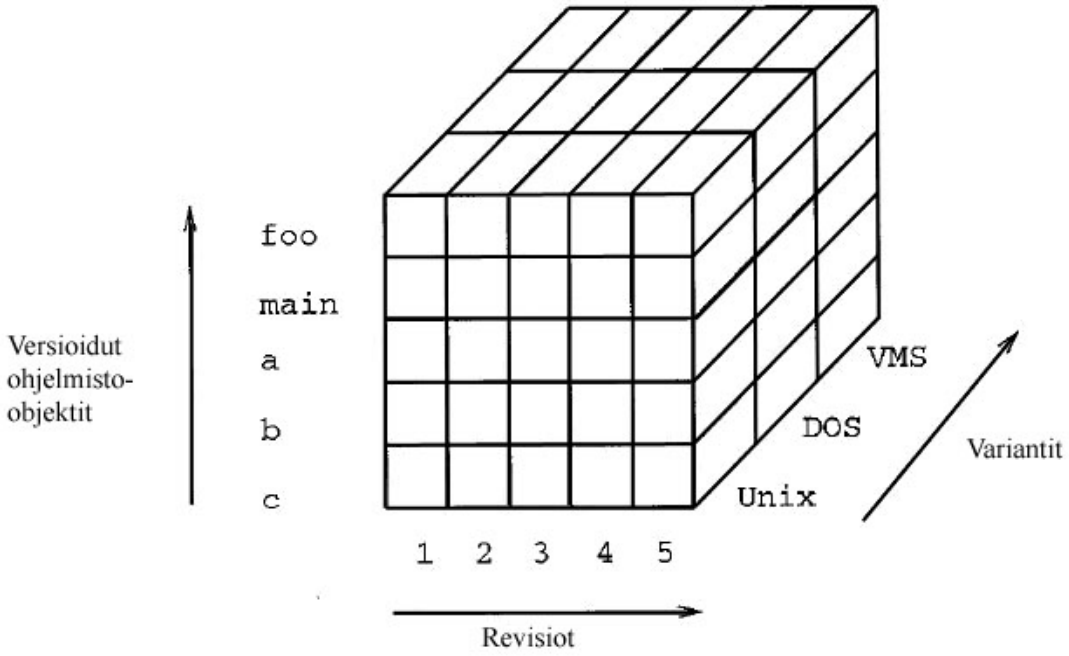
Prosessituki tarjosi aluksi apua vain muutoksenhallintaan, mutta nykyään kehittyneimmät konfiguraationhallintajärjestelmät tukevat yleisesti koko muutosprosessia (Estublier ym. 2005, 393). Prosessituen avulla organisaatio pystyy tarvittaessa suunnittelemaan ja implementoimaan uusia muutosprosesseja ohjelmistoihin (Estublier ym. 2005, 393). Myös Bersoff, Henderson ja Siegel (Bersoff ym. 1980 mukaan, Conradi & Westfechtel 1998, 233) ottavat kantaa prosessitukeen toteamalla, että prosessituen tarkoitus on kontrolloida ohjelmistotuotteen muutoksia.

3.3 Ohjelmistotuotteen versionhallinta

Aluksi selvennetään tämän kohdan kannalta tärkeitä termejä versio, ohjelmisto-objekti ja versiointi. Versio esittää kehittyvän ohjelmisto-objektin tilan. Ohjelmisto-objekti voi olla

mikä tahansa ohjelmistoon liittyvä tekijä joka voidaan laittaa versionhallinnan alaisuuteen. Versiointia voidaan soveltaa ohjelmisto-objekteihin riippumatta niiden monimutkaisuudesta, versioinnin kohteena voi siis olla minkälainen tahansa ohjelmisto-objekti tekstirivistä ohjelmistotuotteeseen. Versioidut ohjelmisto-objektit kuuluvat versionhallinnan alaisuuteen, vastakohtana versioimattomat ohjelmisto-objektit eivät kuulu versionhallinnan alaisuuteen. Versioimattomalla ohjelmisto-objektilla tarkoitetaan ohjelmisto-objektia, josta säilytetään vain yksi versio ja muutokset tehdään päällekirjoittamalla. (Conradi & Westfechtel 1998, 238.)

Ohjelmistotuotannon kannalta tärkeä huomioon otettava ongelma on versionhallinta (Tichy 1982 mukaan, Korel ym. 1991, 161). Tähän yhtyy myös Joeris (1997, 125), joka nostaa artikkelissaan versionhallinnan jopa yhdeksi ohjelmistotuotannon ydinongelmista. McNurlin ja Sprague (2002, 3) perustelevat versionhallinnan tärkeyttä seuraavasti: Teknologian hallintaprosessi organisaatiossa monimutkaistuu jatkuvasti ja samanaikaisesti versionhallinta tulee yhä tärkeämmäksi. Täten monista versioista ja konfiguraatioista koostuvan ohjelmistotuotteen pitäminen hyvin organisoituna on yksi versionhallinnan ongelmista (Tichy 1982 mukaan, Korel ym. 1991, 161). Versionhallinta termin lisäselvennykseksi käydään läpi Conradin & Westfechtelin (1998, 250) esittämä kuvio (kuvio 8), joka käsittelee ohjelmistotuotteen versiointia, esittelemällä tuotepohjaisen versioinnin. Kyseinen artikkeli käsittelee versiointimallien käyttöä konfiguraationhallintaan, mutta artikkelissa käsitellään versiointia niin tarkalla tasolla, että sieltä saa hyvin sisältöä myös versionhallinnan puolelle. Myös Reichenberger (1989, 138) käyttää artikkelissaan samanlaista graafista esitystä, tuodessaan esille versiokirjaston. Tosin käytännössä kyseisen graafisen esityksen käyttö versionhallinnan tukena on kyseenalaista, sillä revisioiden ja varianttien määrä on harvoin vakio. Tällöin kyseisestä graafista ei tule säännöllinen kuutio, kuten kuvion mukainen esimerkki ja siten revisioiden ja varianttien määrän hahmottaminen vaikeutuu (Reichenberger 1989).



KUVIO 8. Ohjelmistotuotteen versiointi (Conradi & Westfechtel 1998, 250; Reichenberger 1989, 138)

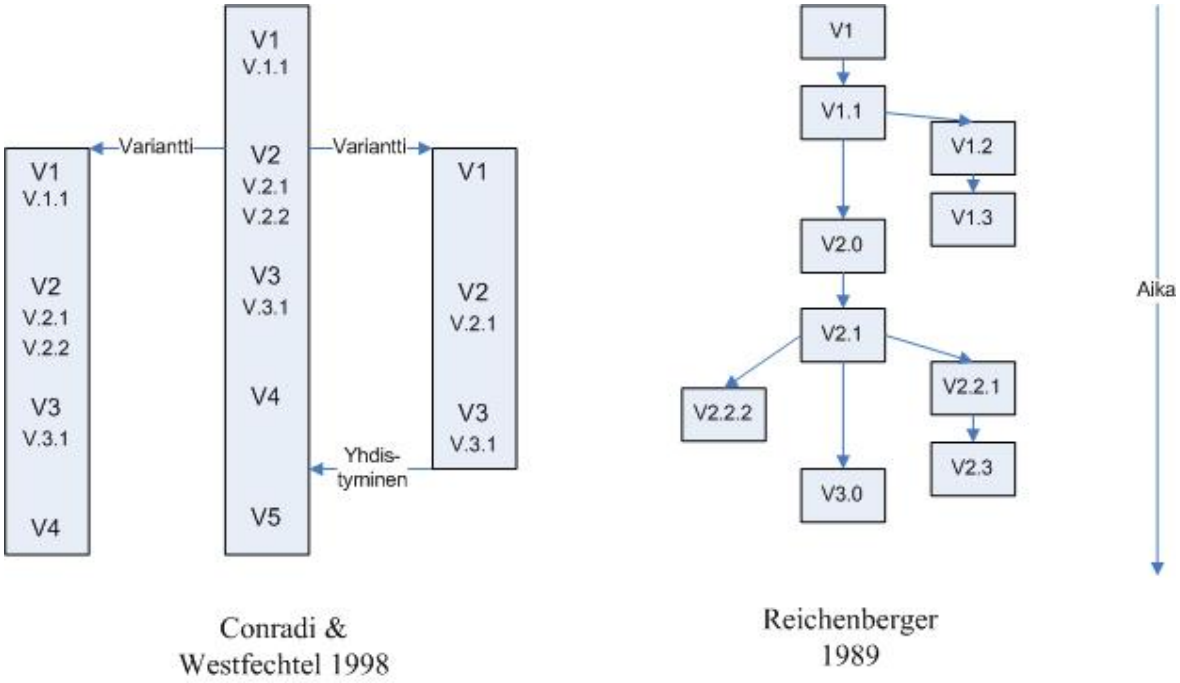
Kuvion pystyakselissa ovat versioidut ohjelmisto-objektit, vaaka-akselissa kyseisten ohjelmisto-objektien päivitykset (revisions) ja z-akselissa eri käyttöliittymille tarkoitetut versiot (variants). Abran ym. (2004, 7 – 6), Conradi ja Westfechtel (1998, 240) sekä Reichenberger (1989, 138) tarkentavat termiä versio samanlaisella määritelmällä seuraavanlaisesti. Versio, jonka tarkoitus on korvata edeltäjänsä, on nimeltään revisio. Versio, jonka tarkoitus on toimia rinnakkain, ei syrjäyttää edeltäjäänsä, on variantti. Erilaisia variantteja voidaan joutua tekemään esimerkiksi suorituskyvyn, talletustilan, käyttöoikeuksien tai eri käyttöjärjestelmien takia (Conradi & Westfechtel 1998, 240). Selkeyttämisen vuoksi kutsun tutkielmassa tästä eteenpäin edeltäjänsä korvaavia versioita revisioiksi ja rinnakkain toimivia versioita varianteiksi. Vaikka kyseessä on vain yksi lähestymistapa versiointiin, antaa kuvio hyvän käsityksen versionhallinta termistä. Versionhallinnalla tarkoitetaan nimenomaan kuvion esittämää kokonaisuuksien hallintaa elinkaaren aikana. Kyseisen kuvion esittämä versioinnin lähestymistapa on Conradin ja

Westfechtelin (1998, 249) mukaan tuotepohjainen versiointi (product versioning). Muut Conradin ja Westfechtelin (1998, 249) esittämät versiointitavat ovat komponenttipohjainen versiointi (component versioning) ja kokonaisvaltainen versiointi (total versioning).

3.3.1 Versiohistoria

Elinkaarensa aikana monimutkaiseen ohjelmistotuotteeseen liittyvät ohjelmisto-objektit, kuten vaatimusten määritykset, suunnittelumallit, ohjelmiston lähdekoodi, dokumentaatiot, ja testidata kehittyvät moniksi versioiksi (Westfechtel ym. 2001, 1111). Samat ohjelmisto-objektit mainitaan myös Conradin ja Westfechtelin (1998, 235) aikaisemmassa artikkelissa ohjelmisto-objektin määrityksen yhteydessä. Versionhallinnan alaisiin ohjelmisto-objekteihin voi tietenkin lukeutua myös muita ohjelmisto-objekteja, sillä Westfechtelin ym. esittämä määritys pitää sisällään vain yleisimmät ohjelmisto-objektit. Versiohistorian hallinnan apuna on useita erilaisia graafisia esitysmuotoja. Toinen hyvä tapa esittää ohjelmistotuotteen useat revisiot ja variantit graafisessa muodossa, verrattuna kuvion 8 mukaiseen esitystapaan, on revisiopuu (kuvio 9) (Conradi & Westfechtel 1998, 242; Reichenberger 1989, 137). Revisiopuun (eli versiopuun) avulla saadaan paremmin esille revisioiden aikaulottuvuus, eikä varianttien ja ohjelmisto-objektien esittäminen siitä huolimatta kärsi, verrattuna tuotepohjaisen versioinnin graafiseen esitykseen.

Kuviossa käsitellään kaikki kolme ITIL:n (2005, 205) määrittelemää revisioluokkaa. Tärkeä (major) revisio usein syrjäyttää edelliset revisiot ja aloittaa revisionumeroinnin pienten (minor) revisioiden kohdalta alusta. Esimerkiksi revisiot 2.0 ja 3.0 ovat tärkeitä revisioita ja revisiot 2.1 ja 3.1 ovat pieniä revisioita. Lisäksi on olemassa kolmas revisioluokka; pikakorjaus (fix). Kyseisen luokan revisiot sisältävät usein vain pieniä korjauksia. Esimerkiksi revisiot 2.2.1 ja 2.2.2 kuuluvat pikakorjauksiin. (ITIL 2005, 205–207.) Varianttien määrä kannattaa pitää mahdollisimman pienenä, sillä se tekee versionhallinnasta huomattavasti helpompaa ja luotettavampaa. Lisäksi revisiopuut hajoavat vaakasuunnassa hallitsemattoman kokoisiksi liian suuren varianttimäärän takia. (ITIL 2005, 239; Reichenberger 1989.)

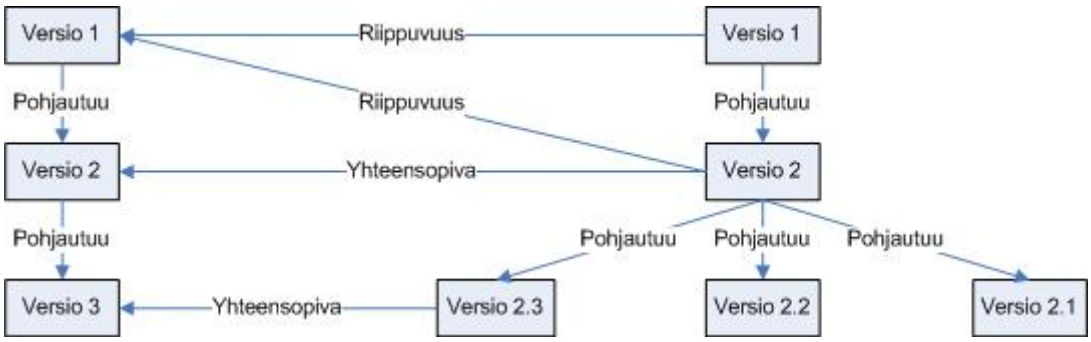


KUVIO 9. Revisiopuumallit (Conradi & Westfechtel 1998, 242; Reichenberger 1989, 137)

Perustuen kuvioihin 5 ja 9, versionhallinnan keskeinen aspekti on aikaulottuvuuden seuranta ohjelmistotuotteen tai ohjelmisto-objektin elinkaaressa. Tämä tarkoittaa sitä, että tarvittaessa voidaan palata tiettyyn ajalliseen pisteeseen ja tarkastella sen hetkistä ohjelmistotuotteen tilaa konfiguraationhallinnan avulla. Conradin ja Westfechtelin (1998, 242) ja Reichenbergerin (1989, 137) esittämät revisiopuumallit ovat ominaisuuksiltaan yksinkertaistettuja. Sachweh ja Schafer (1995, 25) lisäävät revisiopuun avulla esitettäviin kuvauksiin mukaan tarkemmat versioiden väliset suhteet. Näitä suhteita ovat pohjautuvuus (based on), riippuvuus (depend on) ja yhteensopivuus (consistent with). Alla oleva kuvio (kuvio 10) tuo esille edellä mainitut Sachwehin ja Schaferin (1995) esittämät versioiden väliset suhteet.

Pohjautuvuudella tarkoitetaan perinteisenkin revisiopuun esiin tuomaa suhdetta, esimerkiksi versio 2.1 pohjautuu versioon 2.0. Riippuvuus suhdetta Sachweh ja Schafer (1995, 22) kuvaavat kahden ohjelmisto-objektin välille rakentuvalla riippuvuudella, jossa ensimmäinen ohjelmisto-objekti tarvitsee resursseja toiselta ohjelmisto-objektilta. Yhteen-

sopivuussuhteen avulla taas saadaan selville ohjelmisto-objektin kyseisen version kanssa yhteensopivat versiot. Sekä riippuvuus- että yhteensopivuussuhde ovat konfiguraationhallinnan puolella käytettyjä suhteita, sillä konfiguraatiot rakentuvat yhteensopivista ohjelmisto-objekteista (Sachweh & Schafer (1995, 23) ja konfiguraatioiden avulla täytyy saada esille ohjelmisto-objektien väliset riippuvuudet (Abran ym. 2004, 7 – 6; Dart 1991, 7). Tällaiset suhteet voidaan ottaa mukaan myös revisiopuuhun, kun halutaan esimerkiksi käsitellä useita revisiopuita yhtä aikaa ja halutaan saada revisiopuusta esiin oleellista tietoa myös konfiguraationhallinnan puolelle (Sachweh & Schafer 1995, 22).



KUVIO 10. Versiotaso (Sachweh & Schafer 1995, 25)

3.3.2 Versiointimalli

Versiointi varsinkin laajoissa ohjelmistoissa ei saa olla satunnaista, vaan sen pitää olla tarkasti annetun versiointimallin mukaisesti suoritettu. Conradin ja Westfechtelin (1998, 233) mukaan versionhallinnan täytyy tarkasti ja luotettavasti tallentaa versioidun ohjelmistotuotteen rakenne, kun ohjelmistotuote kehittyy moniksi eri revisioiksi ja varianteiksi. Erityisesti laajan ohjelmistotuotteen kannalta on tärkeää, että versionhallinta toimii heti alusta asti. Tämä varmistaa mahdollisimman tehokkaan versionhallinnan tuen ohjelmistojen kehitykselle ja myöhemmin ohjelmistojen ylläpidolle. Tästä syystä Conradi ja Westfechtel (1998) tuovat esille versiointimallin (version model). Versiointimalli määrittää ohjelmisto-objektit, jotka kuuluvat versioinnin alaisuuteen, tunnistamistavan, rakenteen,

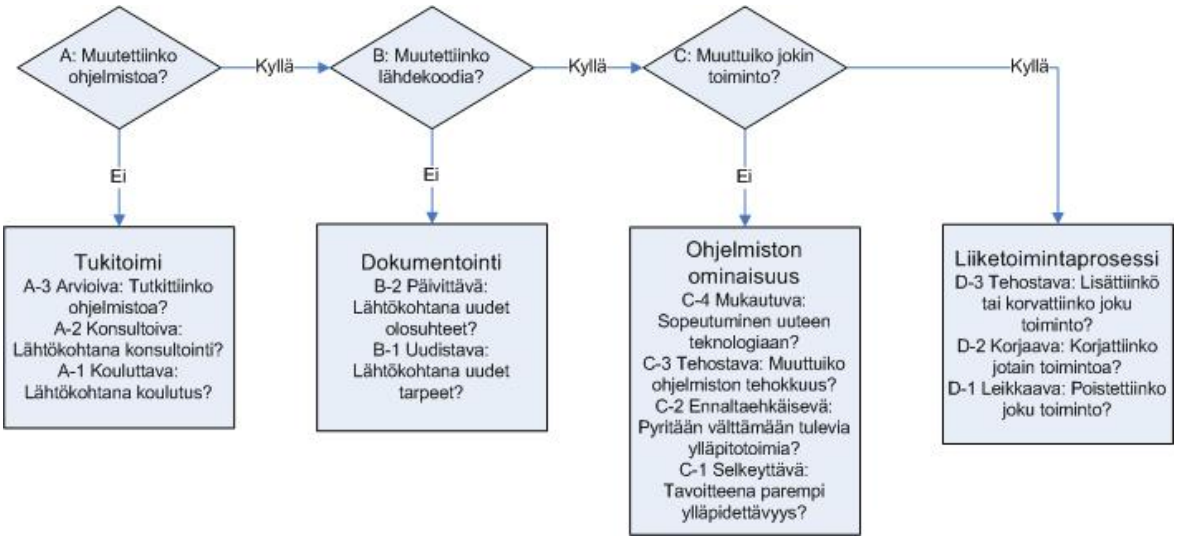
operaatiot miten olemassa olevat versiot noudetaan ja uusien versioiden rakentamistavan (Conradi & Westfechtel 1998, 233). Versioinnin alaisuuteen voidaan määrittää kuuluvaksi kaikki ohjelmisto-objektit, tällöin kyseessä on alueriippumaton versiointimalli (Tichy 1985 mukaan, Conradi & Westfechtel 1998, 235). Vaihtoehtoisesti jos versioitavaksi halutaan vain tietyn tyyppiset ohjelmisto-objektit, on käyttöön otettava aluemääritelty versiointimalli (Conradi & Westfechtel 1998, 235).

Se, miten versiointimallit näkyvät kehittäjälle/ylläpitäjälle, vaihtelee erityyppisten konfiguraationhallintajärjestelmien välillä. Suurin osa kaupallisista konfiguraationhallintajärjestelmistä tukeutuu tiedostopohjaiseen versiointimalliin, jossa versiointia sovelletaan tiedostoihin ja kansioihin. Lisäksi on olemassa ohjelmointikieliin ja tietokantoihin pohjautuvia versiointimalleja. Jälkimmäisistä eli tietokantoihin pohjautuvista versiointimalleista esimerkkeinä mainittakoon EER (Extended Entity-Relationship Model) ja oliopohjaiset versiointimallit. (Conradi & Westfechtel 1998, 234.)

Erilaisia versioita ohjelmistotuotteesta voi olla useita. Versiointimallin tarkoitus on tukea ohjelmistotuotantoa eri versioiden tunnistamisessa, tähän kuuluu myös version kokeman muutoksen tyypin tunnistaminen (toisin sanoen millainen muutos oli tai minkä takia muutos tehtiin). Ratkaisuksi artikkelissa esitetään päätöspuuta (kuvio 11), joka käy läpi erityyppiset kehitys- ja ylläpitotyypit. Päätöspuu on esimerkki yksinkertaistetusta versiointimallista, joka tukee myös kehitys- ja ylläpitotyyppien luokittamisen standardointia.

Kaavan toimintaperiaate on seuraava. Ohjelmistoon tehty toimenpide voi kuulua useampaan ryhmään ja ryhmän sisällä useampaan tyyppiin. Ryhmiä edustavat ohjelmisto, lähdekoodi ja toiminto. Lisäksi kullakin ryhmällä on sisällään ryhmää tarkentava tyypitys. Tyypin valinta tehdään pääosin sillä perusteella, mikä tyyppi on dominoiva. Jos dominoivaa tyyppiä ei löydy, valitaan tyyppi joka sijaitsee korkeimmalla hierarkiassa eli lähimpänä oikeaa ylälaitaa (Chapin ym. 2001). Kaavasta on apua eritoten suurissa ohjelmistoissa, joista saattaa syntyä satojakin eri versioita elinkaarensa aikana. Kaavan avulla versiot saadaan luokiteltua omiin ryhmiinsä. Siten esimerkiksi jonkin tietyn entisen version hakeminen helpottuu, tämä auttaa eritoten ohjelmistojen ylläpidon

konfiguraationhallintaa. Myös Estublier ym. (2005) kuvaavat versionhallinnan antamia etuja tältä osa-alueelta. Versioinnin avulla ylläpidetään historianmukaista arkistoa ohjelmistosta ja sen ohjelmisto-objekteista sitä mukaa, kun nämä kokevat muutoksia, tällainen versiointi toimii perustavanlaatuisena apuna ohjelmistojen konfiguraationhallinnalle (Estublier ym. 2005, 392). Tästä voidaan päätellä, että versionhallinnalla on suuri vaikutus konfiguraationhallinnan toimivuuteen, kun kyseessä on oikein versioitujen ohjelmisto-objektien jatkokäyttö.



KUVIO 11. Ohjelmistojen kehityksen ja ylläpidon luokittelun päätöspuu (Chapin, Hale, Khan, Ramil & Tan 2001, 10)

3.3.3 Versiokirjasto

Versiokirjastoissa (software library) säilytetään ohjelmisto-objektien ja ohjelmisto-tuotteiden eri versioita. Versiokirjasto toimii siten versionhallinnan pääasiallisena työ-kaluna. (IEEE 1990, 68) Versiokirjastoja voi olla useita erityyppisiä. Näistä ITIL (2005, 203) sekä Abran ym. (2004, 7 – 7) mainitsevat työskentelykirjaston (working library), joka toimii perinteisen ohjelmoinnin tukena, tukikirjaston (project support library), joka tukee

esimerkiksi testausta ja tuotekirjaston (master library tai definitive software library), jossa säilytetään ohjelmistotuotteiden ja ohjelmisto-objektien valmiita ja hyväksyttyjä versioita. Kyseisiin versiokirjastoihin tallennettuja versioita kerääntyy suuria määriä ja näiden versioiden hallinta aiheuttaa haasteita ohjelmistotuottajille. Esimerkkinä työskentelykirjaston hyödyistä Abran ym. (2004, 7 – 7) mainitsevat ohjelmistojen kehityksen ja ylläpidon saaman avun suuren ohjelmisto-objektimäärän käytön tukena. ITIL:n (2005, 209) mukaan versiokirjastoa suunnitellessa pitäisi ottaa huomioon seuraavat seikat:

- Versiokirjaston fyysinen sijainti, sekä käytettävät ohjelmistot ja laitteistot: Monet konfiguraationhallintajärjestelmät sisältävät versiokirjastoja tukevia ominaisuuksia, joten kyseisiä järjestelmiä voidaan pitää loogisena osana versiokirjastoa.

- Nimeämiskäytännöt, esimerkiksi ohjelmisto-objektin tunnistimen käyttö (Conradi & Westfechtel 1998, 235)

- Tehtäväalueet, joiden kanssa versiokirjasto toimii yhteistyössä.

- Turvallisuusjärjestelyt liittyen versiokirjaston muutoksiin, varmuuskopioihin ja palauttamisiin.

- Versiokirjaston laajuus. Alueriippuvainen tai alueriippumaton versiointimalli (Conradi & Westfechtel 1998, 235; Tichy 1985 mukaan, Conradi & Westfechtel 1998, 235).

- Vanhojen versioiden säilytykseen liittyvät tekijät, kuten säilytysaika.

- Kapasiteettisuunnitelmat ja käytänteet kapasiteetinvalvontaan.

- Seurantamenetelmät, esimerkiksi konfiguraationhallintajärjestelmän tarjoamana (Dart 1991, 7).

Versiokirjaston lisäksi konfiguraationhallintajärjestelmän avulla voidaan luoda konfiguraationhallintatietokanta. Konfiguraationhallintatietokanta on samantapainen kirjasto kuin versiokirjasto, erona vain se, että sitä käytetään vain konfiguraatioiden

tallentamiseen. Usein versiokirjastoa kuitenkin käytetään sekä versioiden että konfiguraatioiden tallentamiseen, eikä erillistä konfiguraationhallintatietokantaa luoda. (Abran ym. 2004, 7 – 3; ITIL 2005, 124.)

3.3.4 Ohjelmistotuotteen elinkaari

IEEE standardin (1990, 68) mukaan ohjelmistotuotteen elinkaari alkaa siitä, kun ohjelmistotuote saa alkunsa ja loppuu siihen, kun ohjelmistotuotteella ei ole enää mitään käyttöä. Tyypillinen ohjelmistotuotteen elinkaari sisältää konsepti-, vaatimusten määritys-, suunnittelu-, implementointi-, testaus-, ylläpito- ja lopetusvaiheen (IEEE 1990, 68). Ohjelmistotuotteen elinkaari voi kestää useita vuosia ja sen aikana ohjelmistoon voidaan tehdä jopa satoja päivityksiä tai ylläpitotoimenpiteitä. Varsinkin suurissa ohjelmistoissa ohjelmistojen versionhallintaan sijoitetaan paljon resursseja johtuen ohjelmiston pitkästä elinkaaresta. Tämä tarkoittaa sitä, että myös ylläpitoon liittyvä versionhallinta on tärkeä osa-alue ohjelmistotuottajalle.

Esimerkkinä ohjelmiston elinkaaresta käsitellään Eickin ym. (2002) artikkelissaan tutkimaa suurikokoista ohjelmistoa. Ohjelmisto on 15 vuotta käytössä ollut puhelinverkon kytkin. Ohjelmisto koostuu 100 miljoonasta koodirivistä, pääosin kirjoitettuna c/c++ kielillä. Ohjelmiston elinkaaren aikana kehitykseen ja ylläpitoon on osallistunut yli 10 000 henkilöä. (Eick, Graves, Karr, Mockus & Schuster 2002, 397.)

Esimerkin kaltainen ohjelmisto voi kokea elinkaarensa aikana jopa tuhansia ylläpitotoimenpiteitä. Eick ym. (2002, 402) esittävät artikkelissaan kuvion, johon on koottu tietoa ohjelmistossa olevan aliohjelman kokemista muutoksista 15 vuoden ajalta. Kyseinen aliohjelma hoitaa operaattorin ja puhelinverkon kytkimen välistä toimintaa. Kuviossa ei ole eritelty ohjelmistotuotteen läpikäymiä eri vaiheita, vaan ohjelmiston elinkaari on jaettu vuosittain. Silti kuviosta voi päätellä, että ohjelmisto on kokenut paljon muutoksia, on kyseessä ollut siten mikä tahansa IEEE standardin (1990, 68) määrittämistä tyypillisistä ohjelmistotuotteen elinkaaren vaiheista. Eickin ym. (2002) esittämän kuvion muodostamisen mahdollistaa toimiva versionhallinta, sillä versionhallinnan vastuulla on tarkka versiohistorian seuraaminen. Valitettavasti Eickin ym (2002, 402) kuviosta on, ilmeisesti

tietoturvasyiden takia, jätetty vaaka- ja pystyakselien arvot lukukelvottomiksi. Tästä syystä kuviota ei esitetä gradussa, vaan keskitytään kuvion keskeisimpiin ominaisuuksiin. Kuviossa pystyakselissa on muutosten lukumäärä ja vaaka-akselissa ohjelmiston elinkaari jaettuna vuosittain. Muutoksilla kuviossa tarkoitetaan muutettujen koodirivien määrää. Muutokset eivät siis tarkoita aina uutta versiota (revisiota tai varianttia), vaan yhden ohjelmistopäivityksen aikana ohjelmistoon voi kohdistua useita muutoksia. Kuvion käsittelyn avulla pyritään osoittamaan, kuinka monta muutosta ohjelmistoon voi kohdistua sen elinkaaren aikana ja tätä kautta konkretisoida versionhallinnan tärkeyttä ohjelmistotuotannon tukena. Lisäksi kuviosta voi päätellä, että aktiivisessa käytössä olevan ohjelmiston ylläpito jatkuu huolimatta siitä, että ohjelmisto on ehtinyt kypsyä 15 vuoden ajan. Tähän vaikuttaa muun muassa käytettävien teknologioiden kehittyminen ja ohjelmistoon kohdistuvien vaatimusten muuttuminen. Toisin sanoen ohjelmisto kokee jatkuvasti muutospaineita toimintaympäristön muuttuessa.

Eickin ym. (2002, 404) artikkelissa esitetään myös kuvio, joka käsittelee muutosten lukumäärää per versio, oli kyseessä sitten revisio tai variantti. Kuvio jätetään esittämättä samoihin perusteluihin vedoten, kuin edellisessä kappaleessa käsitelty kuvio. Kuvio tuo hyvin esille miten eri versioihin vaadittava työmäärä voi vaihdella. Jokaisen version kohdalla on kaksi palkkia, ensimmäinen esittää lisättyjen ja toinen poistettujen koodirivien määrän. (Eick ym. 2002, 404.) Kuvio ei ota kantaa muihin ohjelmisto-objekteihin kohdistuvaan versionhallintaan, vaan tarkkailee nimenomaan ohjelmistokoodiin kohdistuneita muutoksia. Kuviosta saa silti hyvän kuvan ohjelmistotuotteen elinkaaresta ja elinkaaren aikana tapahtuvasta kehityksestä/ylläpidosta. Lisäksi kyseiset muutokset nimenomaan aiheuttavat usein tarpeen julkaista/tehdä uusi versio ohjelmistotuotteesta (ITIL 2005, 12). Versioiden kokemien muutosten määrän seurannasta saadaan tärkeää tietoa versionhallinnalle. Varsinkin tulevien projektien suunnittelun ja työmäärän ennakoinnin pitäisi olla huomattavasti helpompaa Eickin ym. (2002) esittämien versiotietojen keräämisen avulla.

3.3.5 Versionhallinnan ratkaisut dokumentoinnin ongelmiin

Ohjelmistotuotannossa pidetään suunnitteludokumentaatiota ja tuotedokumentaatiota (joka on dokumentaatiota toteutetuista muutoksista). Suunnitteludokumentaatiota käytetään ohjelmistojen kehityksen versionhallinnassa, kun taas muutoksia käsittelevää tuotedokumentaatiota käytetään ohjelmistojen ylläpidon yhteydessä. Dokumenttien versionhallinta hoidetaan yleensä hallitsemalla yhden dokumenttityypin eri versioita, käyttämällä muutospuurakennetta (Reichenberger 1989, 137). Muutospuurakenteessa dokumenttien eri revisiot säilyvät kronologisessa järjestyksessä. Täten muutospuurakenteen avulla ohjelmistotuottajalla säilyy selkeä kuva tiettyjen dokumenttien eri versioista. Muutospuurakenne auttaa sekä dokumenttimallien hallinnassa että itse dokumenttien versioinnissa. Jos kyseessä on pysyvä dokumenttimalli, auttaa muutospuurakenne myös selventämään syyt, miksi dokumenttimallia piti muuttaa. Toisaalta, dokumenteista ja dokumenttimalleista säilyy vanhat versiot oikeassa järjestyksessä ja vanhempaan versioon palaaminen on helppoa.

Seuraavaksi käsitellään esimerkki osittain ylläpitoon liittyvästä ongelmasta dokumentoinnin saralla. On selvää, että ohjelmistoa on sen elinkaaren aikana ylläpitämässä useita eri henkilöitä. Kuitenkin, kun uusi ylläpitäjä ottaa työn alle oman projektin, joutuu hän käymään läpi kyseiseen osa-alueeseen aikaisemmin tehdyt muutokset ja kyseisen osa-alueen määritykset. Tässä vaiheessa kohdataan usein ongelmia. Dokumentoinnissa käytettäville termeille on tehty useita standardeja, jotka eivät vastaa täysin toisiaan, eli joillakin tärkeillä termeillä voi olla useita eri merkityksiä. Täten kookkaissa ohjelmistoissa jo niin pieni tekijä, kuin käytettävien termien standardointi voi tehostaa versionhallintaa (Chapin ym. 2001, 4). Chapinin ym. (2001) artikkelissa on tehty tutkimus versionhallintaan liittyvien tehtäväkuvausten vastaavuudesta eri standardien kesken (taulukko 2). Kuten taulukosta näkee, niin ainoastaan yksi vertailuun otettu tehtäväkuvaus määritellään samalla termillä jokaisessa standardissa (Liiketoimintaprosessi – Korjaava). Nämä eroavaisuudet voivat tuoda sekaannuksia ja ongelmia versionhallinnassa useissa eri tilanteissa. Esimerkiksi jos ohjelmistoon liittyvä dokumentointi on tehty käyttäen IEEE standardia ja eri standardia (esimerkiksi ESF/EPSOM) käyttävä toinen organisaatio ottaa ohjelmiston ylläpidon vastuulleen. Ensinnäkin dokumentointi täytyisi tehdä tai vähintään järjestellä

uusiksi, sillä ryhmityksiin liittyvät määritykset eroavat toisistaan. Lisäksi konsultoinnin saaminen toista standardia käyttävältä organisaatiolta ohjelmiston ylläpitoa varten saattaisi osoittautua tehottomaksi eri termistön takia, ellei eroavaisuuksia oteta huomioon.

TAULUKKO 2. Termien vastaavuudet eri standardeissa (Chapin ym. 2001, 6)

Tyyppipohjainen määritys (Chapin ym. 2001)		**Tarkoituspohjaiset määritykset**			**Toimintapohjaiset määritykset**	
Ryhmä	**Tyyppi**	**Swanson (1976)**	**IEEE (1990 & 1998)**	**ISO (1999)**	**Kitchenham ym. (1999)**	**ESF/ EPSOM (Harjani & Queille 1992)**
Tukitoimi	Arvioiva	Ei sisällä	Ei sisällä	Epäsuorasti kaikissa tyypeissä	Epäsuorasti kaikissa tyypeissä	Epäsuorasti kaikissa tyypeissä
	Konsultoiva	Ei sisällä	Ei sisällä	Epäsuorasti kaikissa tyypeissä	Ei sisällä	Käyttötuki
	Kouluttava	Ei sisällä	Ei sisällä	Epäsuorasti kaikissa tyypeissä	Ei sisällä	Käyttötuki
Dokumentointi	Päivittävä	Viimeistelevä	Viimeistelevä	Viimeistelevä	Epäsuorasti kaikissa tyypeissä	Ennaltaehkäisevä
	Uudistava	Viimeistelevä	Viimeistelevä	Viimeistelevä	Epäsuorasti kaikissa tyypeissä	Ennaltaehkäisevä
Ohjelmiston ominaisuus	Mukauttava	Viimeistelevä	Viimeistelevä	Tehostava viimeistely	Tehostava	Viimeistelevä
	Tehostava	Viimeistelevä	Viimeistelevä	Viimeistelevä tai Tehostava viimeistely	Korjaava tai Muutoksen implementointi	Ennakoiva tai Viimeistelevä
	Ennaltaehkäisevä	Viimeistelevä	Viimeistelevä tai Ennaltaehkäisevä	Ennaltaehkäisevä tai Tehostava viimeistely	Ennaltaehkäisevä	Ennakoiva tai Ennaltaehkäisevä

(jatkuu)

TAULUKKO 2. (jatkuu)

	Selkeyttävä	Selkeyttävä	Selkeyttävä	Tehostava viimeistely	Olemassa olevien vaatimusten muuttaminen	Ennakoiva tai Selkeyttävä
Liiketoiminta-prosessi	Tehostava	Viimeis-televä	Viimeis-televä	Tehostava viimeistely	Vaatimusten lisääminen	Kehittävä
	Korjaava	Korjaava	Korjaava	Korjaava	Korjaava	Korjaava
	Vähentävä	Viimeis-televä	Viimeis-televä	Tehostava viimeistely	Olemassa olevien vaatimusten muuttaminen	Kehittävä

3.3.6 Konfiguraationhallintajärjestelmä versionhallinnan tukena

Versionhallinnan mahdollisimman tehokkaan toimivuuden takaamiseksi versionhallinnan prosessien tulee toimia yhteistyössä konfiguraationhallinnan prosessien kanssa (ITIL 2005, 12). Versionhallintaa voidaan tehostaa esimerkiksi lisäämällä konfiguraationhallinta-järjestelmään versionhallintaa tukevia ominaisuuksia. Dartin (1991, 7) esittämät kategoriat kuvaavat käyttäjän tarpeita konfiguraationhallintajärjestelmältä. Yksi kategorioista selventää, miltä osin konfiguraationhallintajärjestelmän pitää tukea myös versionhallintaa:

- Seuranta. Konfiguraationhallintajärjestelmästä voidaan kerätä tarvittavaa tilasto-tietoa ohjelmistotuotteesta ja ohjelmistotuotantoprosessin etenemisestä.

Varsinkin monimutkaisessa ohjelmistotuotantoprosessissa on tärkeää, että ohjelmiston rakenne säilyy selkeänä. Seurannan avuksi monet konfiguraationhallintajärjestelmät tar-joavat mahdollisuuden tehdä graafisia esityksiä ohjelmiston rakenteesta. Graafisten esitysten avulla voidaan versioidun ohjelmisto-objektikannan (versioned object base) rakenne esittää selkeänä kokonaisuutena (Conradi & Westfechtel 1998, 234). Hyvä esimerkki ohjelmiston rakennetta selkeyttävästä graafisesta esityksestä, on Conradin ja Westfechtelin (1998, 250) sekä Reichenbergerin (1989, 138) esittämä kuvio (kuvio 8), joka käsittelee ohjelmistotuotteen versiointia. Lisäksi, graafiset esitykset tukevat ohjelmisto-

tuotantoprosessin analysointia. Tätä kautta esille voi nousta sellaisia ohjelmisto-tuotantoprosessia tehostavia tekijöitä, jotka olisivat muuten jääneet huomioimatta (Eick ym. 2002, 401).

Konfiguraationhallintajärjestelmät tukevat myös ohjelmistojen ylläpidon aikaista version-hallintaa. Tätä väitettä on tukemassa yksi Dartin (1991, 7) esittämistä kategorioista konfiguraationhallintajärjestelmiltä vaadittavista ominaisuuksista:

- Tarkastus. Konfiguraationhallintajärjestelmän avulla käyttäjä kykenee palaamaan mihin tahansa aikaisempaan versioon, mistä tahansa ohjelmisto-objektista ja tarkastamaan mitä muutoksia on tehty, kuka muutokset teki ja miksi kyseiset muutokset tehtiin.

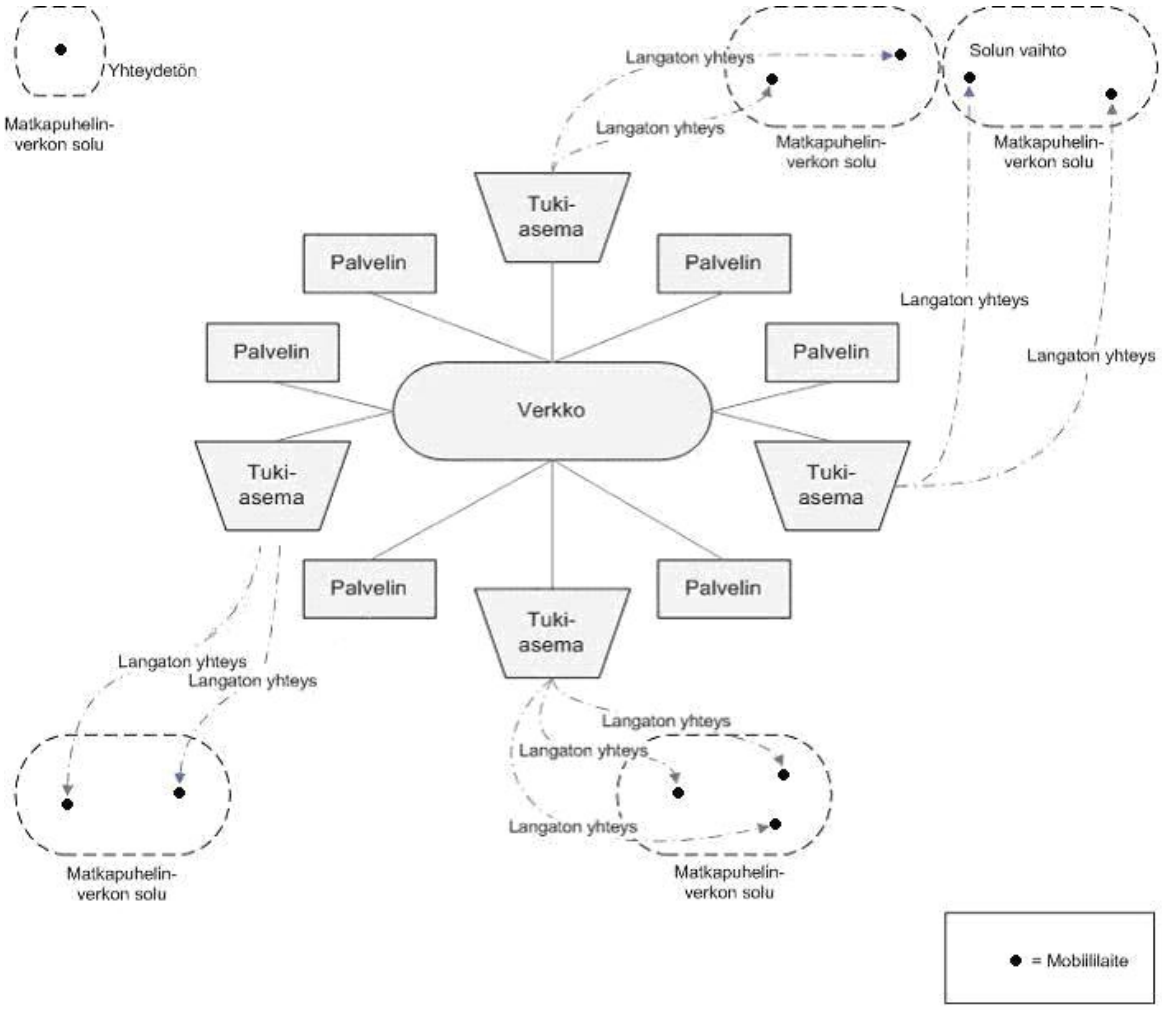
Aikaisempien versioiden käyttäminen ja tarkastaminen vaikeutuu, jos ohjelmisto-objekteissa ei ole jotain millä sen voi tunnistaa. Tällainen tilanne tulee eteen varsinkin jos kyseisestä ohjelmisto-objektista on tehty paljon eri revisioita, jotka eivät juuri eroa toisistaan. Ratkaisu kyseiseen ongelmaan on ohjelmisto-objektin tunnistin (OID eli object identifier). Conradi ja Westfechtel (1998, 235) jakavat artikkelissaan ohjelmisto-objektin tunnistimet kahteen eri tyyppiin. Ulkoinen ohjelmisto-objektin tunnistin on ylläpitäjän itse nimeämä, kun taas sisäinen ohjelmisto-objektin tunnistin on konfiguraationhallinta-järjestelmän luoma (Conradi & Westfechtel 1998, 235). Ulkoinen tunnistin on selkeä, kun ohjelmistojen ylläpitoa ei ole suorittamassa useita eri henkilöitä. Mutta jos kyseessä on suuri ohjelmisto, jonka ylläpitoon vaaditaan paljon ryhmätyötä tai hajautettua työskentelyä, vaatii ulkoisen ohjelmisto-objektin tunnistimen käyttö liikaa koordinointia ja täten sisäinen ohjelmisto-objektin tunnistin on järkevämpi valinta. Aikaisempien versioiden tarkastamisessa apuna toimii hyvin suunniteltu ja toteutettu versiointimalli.

4 MOBIILIPELIEN TUOTTAJAN TOIMIALA

Tässä luvussa käsitellään mobiilipelien tuottajan toimialaa. Kahdessa ensimmäisessä kohdassa kerrotaan mobiilijärjestelmistä ja niiden ominaispiirteistä, tarkentuen lopulta mobiilipelien kuvaamiseen yhtenä mobiilijärjestelmänä. Luvun ensimmäisessä kohdassa käydään läpi mobiilijärjestelmät. Kohdan tarkoitus on antaa selkeä kuva mobiilijärjestelmistä ja järjestelmien nykytilanteesta. Mobiilipelituotanto on yksi ohjelmistoja tuottava toimiala, joka käyttää mobiilijärjestelmiä hyväksi. Tämä on yksi peruste, miksi mobiilipelituotanto on rajattu tutkimuksen kohteeksi. Toisessa kohdassa selvitetään, mikä on mobiilipeli, syventyen muun muassa pelaamisen, moninpelaamisen ja mobiilipelaamisen historiaan. Luvun kahdessa viimeisessä kohdassa käsitellään mobiilipelien tuottamista organisaatiotasolla. Ensiksi kohdassa 4.3 kuvataan mobiilipeliteollisuuden toimialaa. Kohdassa rajataan muun muassa mitä tarkoitetaan mobiilipelien tuottajalla, tarjoajalla, käyttäjällä ja laitteistovalmistajalla. Kohdassa 4.4 tarkennetaan, mikä on mobiilipelien tuottajan rooli omalla toimialallaan. Kohtaan kuuluu muun muassa tarkempi kuvaus mobiilipeleistä, käsittelemällä pelimaailmaa yhtenä sen komponenttina.

4.1 Mobiilijärjestelmät

Tutkielmassa keskitytään mobiilijärjestelmiin, joissa päätelaitteena toimii matkapuhelin. Ensimmäiset matkapuhelimiin perustuvat mobiilijärjestelmät käynnistyivät 1980-luvun alkupuolella Euroopassa, Japanissa, Yhdysvalloissa ja muissa teollistuneissa valtioissa. 1990-luvun puoleen väliin mennessä, suurin osa teollistuneista valtioista oli siirtynyt analogisesta digitaaliseen teknologiaan. Vuoden 2002 loppuun mennessä maailmassa oli jo yli miljardi matkapuhelinliittymää. (Funk 2004, 19.) Dunham ja Helal (1995, 5) esittelevät artikkelissaan tyypillisen mobiilijärjestelmän rakenteen (kuvio 12). Dunhamin ja Helalin (1995, 5) mukaan kyseinen malli on tieteellisissä lähteissä hyvin yleisesti käytetty ja hyväksytty. Perusteena väitteelle, artikkelissa luetellaan useita lähteitä, jotka ovat käyttäneet kuvion mukaista mallia kuvaamaan mobiilijärjestelmän arkkitehtuuria.



KUVIO 12. Mobiilijärjestelmän arkkitehtuuri (Dunham & Helal 1995, 5)

Mobiilijärjestelmillä tarkoitetaan yllä esitetyn kuvion kaltaisia järjestelmiä, jotka käyttävät mukana kuljetettavia, eli mobiileja, langattomia päätelaitteita. Kuvion mukainen mobiilijärjestelmä koostuu mobiililaitteista, palvelimista ja tukiasemista. Mobiililaitteet ovat yhteydessä tukiasemiin langattoman yhteyden avulla. Tukiasemat toimivat rajapintana mobiililaitteiden ja palvelinten välillä. Palvelimet taas ovat yhteydessä toisiinsa perinteisen verkon kautta. (Dunham & Helal 1995, 5.) Mobiilijärjestelmien rakenne voidaan Fritschin, Ritterin ja Schillerin (2005, 1) mukaan jakaa kolmeen luokkaan, luokkia ovat palvelintekniikka (client – server), vertaisverkko (peer – to – peer) ja hybridimalli (hybrid model).

Myös Akkawi ym. (2004, 78) mainitsevat artikkelissaan, että mobiilijärjestelmien rakenne jaetaan kyseisiin kolmeen luokkaan. Kuvion 12 esittämä arkkitehtuuri edustaa palvelintekniikkaa. Lisäksi mobiilijärjestelmä voi käyttää tällä hetkellä muun muassa WLAN, Bluetooth tai GPRS/UMTS tekniikkaan pohjautuvaa tiedonsiirtoa (Fritsch ym. 2005, 1). Son ja Tan (2008, 36) ovat koonneet artikkeliinsa taulukon, jossa on tarkempaa tietoa kyseisten tekniikoiden tiedonsiirtokyvyistä ja muista ominaispiirteistä.

Useat uusimmat matkapuhelinmallit, kuten Nokian N95 (http://www.nokia.fi/A4359045), tukevat kaikkia Fritschin ym. (2005) määrittämiä mobiilijärjestelmien rakenteita ja tiedonsiirtotapoja. Silti mobiilijärjestelmien käytössä tulee ottaa huomioon järjestelmän fyysiset rajoitteet, kuten saantiviive (latency) ja kaistan leveys (Akkawi ym. 2004, 78). Lisäksi käyttäjät vaativat matkapuhelimiin entistä nopeampia prosessoreita, parempaa äänentoistoa, laadukkaampia näyttöjä, loogisempia käyttöliittymiä ja enemmän muistia (Funk 2004). Matkapuhelimen käyttö viihdekeskuksena yleistynyt ja ajanvietettä tarjoavat palvelut ovat nykyään suurin sisällön tuottaja matkapuhelimia käyttävissä mobiilijärjestelmissä (Funk 2004, 59). Ajanvietteellisiin palveluihin kuuluu muun muassa pelit, näytönsäästäjät, soittoäänet ja musiikki (Funk 2004). Matkapuhelimet voivat perustua usealle erilaiselle alustalle (platform). Alustoista mainittakoon esimerkkeinä S60, S80 ja S90. Käyttöjärjestelmänä näissä alustoissa käytetään Symbian OS:a, josta uusin versio tällä hetkellä on 9.5. Jokainen näistä alustoista voi aiheuttaa omanlaatuisia ongelmia mobiilijärjestelmille. Esimerkiksi Huebscher, Price ja Dulay (2006, 52) mainitsevat artikkelissaan heidän ohjelmistonsa toimineen vain tietyissä versioissa S60 alustasta.

Mobiilijärjestelmien tilannetta kuvaa hyvin se, että mobiilia liiketoimintaa löytyy laajasta valikoimasta eri toimialoja, ulottuen terveydenhuollosta rahtipalveluihin. Mobiilin liiketoiminnan sovellusten määrä on kasvussa. Sovellukset voidaan jakaa neljään kategoriaan: yrittäjältä asiakkaalle (B2C), yrittäjältä yrittäjälle (B2B), yrittäjältä työntekijälle (B2E) ja asiakkaalta asiakkaalle (C2C). Kategoriaan B2C kuuluu muun muassa vähittäismyynti, sisällön tarjoaminen, mobiilit pankkipalvelut ja portinvartijapalvelut (concierge services). (Chen & Skelton 2005, 3.) Mobiilipelituotanto on yksi sisältöä tarjoavista toimialoista. B2C puolella mobiili kaupankäynti alkoi teollistuneissa valtioissa vuonna 1999 (Funk 2004, 19).

B2C liiketoiminnasta on saatu kolme tärkeää havaintoa:

- Onnistunut mobiili liiketoiminta on riippuvainen asiakasryhmien käsityksestä palvelun arvokkuudesta.
- Mobiili kaupankäynti sopii parhaiten impulssiostoihin.
- Tarjouksen merkityksellä asiakkaan tilanteeseen on suurempi vaikutus kuin tarjouksen hinnalla. (May 2001 mukaan, Chen & Skelton 2005, 3.)

Mobiilin liiketoiminnan kannalta B2B ja B2E tuottavat suurimmat liikevaihdot. Tämä johtuu siitä, että kyseisiin kategorioihin kuuluu muun muassa omaisuuden ja henkilöstön hallinta, tiimityö, inventaarion hallinta, markkinointikanavien hallinta, ja yrityksen tietovarantoihin pääsy. C2C kategoriaan kuuluu yleisesti ottaen henkilökohtaiset viestintäsovellukset, kuten puheluita ja lyhytviestejä tukevat sovellukset. Lisäksi kyseiseen kategoriaan lasketaan myös tiedostojen vertaisjakelu ja mobiili moninpelaaminen. (Chen & Skelton 2005, 3.)

4.2 Mobiilipelit

Tutkielmassa mobiilipeleillä tarkoitetaan matkapuhelimilla pelattavia pelejä, kuten The Simpsons: Minutes to Meltdown (Electronic Arts 2007) ja Jamdat Bowling 2 (Electronic Arts 2003). Esimerkeiksi on valittu kyseiset mobiilipelit, sillä ne edustavat kahta erityyppistä peliluokkaa. Ensiksi mainittu on uusi, kovat laitevaatimukset omaava peli. Jälkimmäinen taas on hieman vanhempi, mutta mobiilin moninpelaamisen sisältävä peli. Täten kyseiset mobiilipelit tuovat version- ja konfiguraationhallintaan toisistaan eroavia vaatimuksia.

Ensimmäinen tietokonepeli, Space War, ilmestyi vuonna 1961 (Akkawi ym. 2004, 78). Siitä lähtien pelit ovat kehittyneet ja ensimmäinen moninpeli ilmestyi jo vuonna 1969 (Akkawi ym. 2004, 78). Pelien rooli ihmisten vapaa-ajan viihdykkeenä on kasvanut viimeisten vuosien aikana rajusti. 2000-luvun alkupuolella peliteollisuuden liikevaihto ohitti elokuvateollisuuden liikevaihdon ja kehityssuunta on edelleen kasvamaan päin

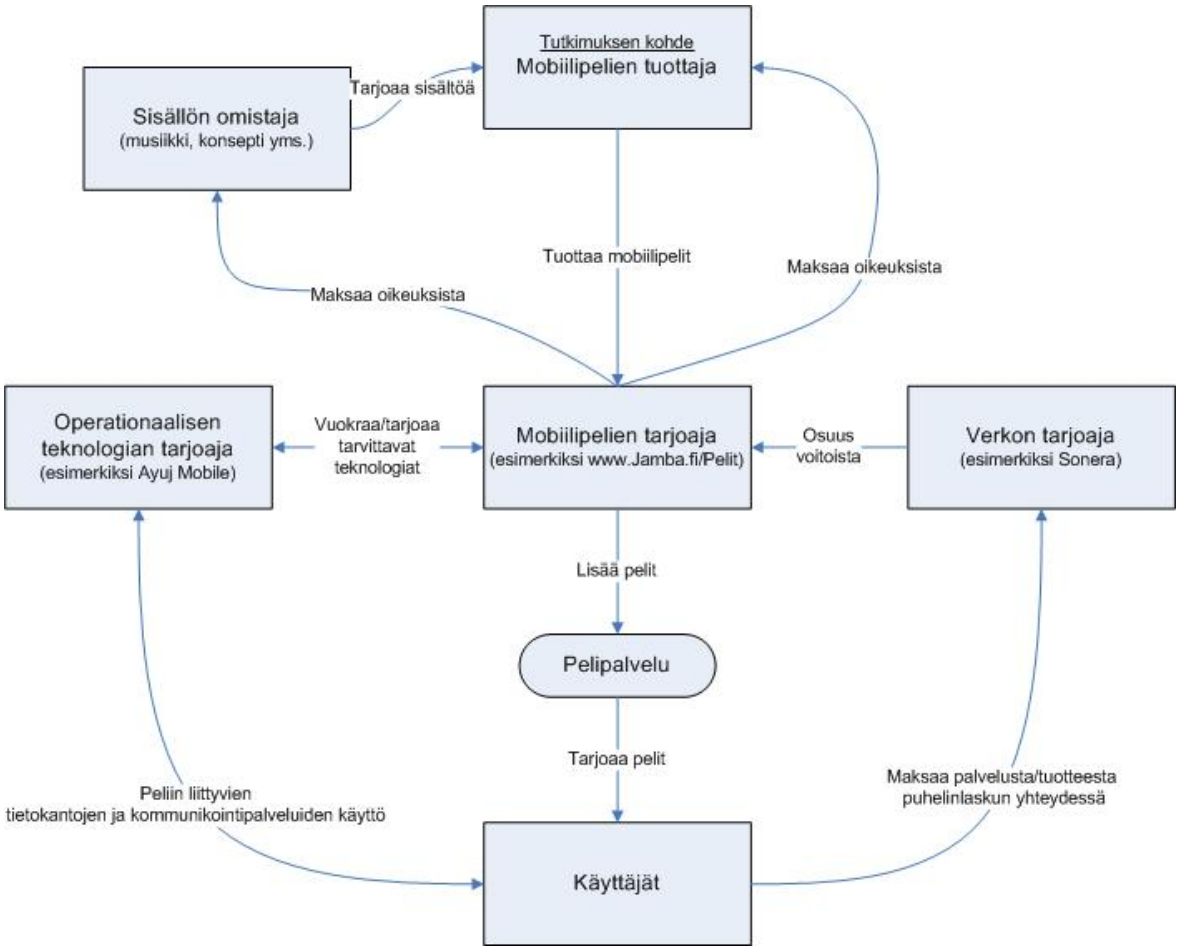
(Fritsch ym. 2005, 1). Pelkästään konsolipelien liikevaihto oli vuonna 2004 yli 10 miljardia dollaria (Funk 2004, 68). Nykyään pelaajat suosivat entistä enemmän moninpelejä verrattuna perinteisiin yksinpeleihin (Fritsch ym. 2005, 1). Moninpelien ääripäätä edustavat MMOG-pelit (Massive Multiplayer Online Game), joissa pelaajamäärä voi olla jopa tuhansia (Son & Tan 2008, 38).

Fritschin ym. (2005, 1) mukaan Euroopassa ja Pohjois-Amerikassa lähes 90 %:lla nuorisosta on vähintään yksi matkapuhelin. Viimeaikaisen teknologisen kehityksen ansiosta matkapuhelimissa on entistä suuremmat laskentatehot, tallennuskapasiteetit sekä laadukkaammat audiovisuaaliset ominaisuudet (Son & Tan 2008, 35). Muun muassa näiden tekijöiden takia mobiilipelien suosio on kasvanut rajusti. Mobiilipelien suosion kasvua kuvastaa hyvin myös se, että pelkästään Euroopassa mobiilipelien liikevaihto kasvoi 0,59 miljardista eurosta vuodessa noin 5,15 miljardiin euroon vuodessa vuosien 2000–2006 välisenä aikana (Son & Tan 2008, 36). Mobiilipelien suosion kasvu oli todella jyrkkä ja vielä muutamaa vuotta aikaisemmin useat arviot mobiilipeleihin liittyvästä liikevaihdosta olivat huomattavasti alakanttiin. Esimerkiksi Hallin, Gordonin, Jamesin ja Newallin (2004, 284) mukaan liikevaihdon odotettiin kasvavan, vuoden 2003 arvion perusteella, 1,65 miljardiin euroon vuodessa, vuoteen 2006 mennessä. Fritschin ym. (2005, 1) vastaava arvio, vuonna 2005, oli 4,6 miljardia euroa vuodessa, vuoteen 2006 mennessä. Akkawi ym. (2004, 77) perustelevat mobiilipelien suosiota sillä, että pelaajat, pelien kehittäjät ja jopa pelien tarjoajat ovat innostuneet peleistä, joita voi pelata missä tahansa. Hall ym. (2005, 284) taas perustelevat mobiilipelien suosiota matkapuhelinten teknologisella kehittymisellä.

4.3 Mobiilipeliteollisuuden rakenne

Mobiilipeliteollisuus koostuu mobiilipelien tuottajasta, tarjoajasta, käyttäjistä, laitteistovalmistajasta, verkon tarjoajasta ja operationaalisen teknologian tarjoajasta (Akkawi ym. 2004, 78; Funk 2004, 77). Tutkielmassa tarkastellaan nimenomaan mobiilipelien tuottajaorganisaatiota, eikä esimerkiksi mobiilipelien tarjoajaa. Alla olevan Akkawin ym. (2004, 78) ja Funkin (2004, 77) artikkeleista koostetun kuvion (kuvio 13) tarkoitus on konkreti-

soida mobiilipeliteollisuuden rakennetta. Kuvio tuo esille mobiilipeliteollisuuden toimijat ja samalla rajaa tutkielman tarkastelualuetta.



KUVIO 13. Mobiilipeliteollisuuden rakenne (Perustuen: Akkawi ym. 2004, 78; Funk 2004, 77)

Mobiilipelien tuottaja toimittaa valmistamansa pelit mobiilipelien tarjoajalle, joka hoitaa mobiilipelit eteenpäin käyttäjille. Tämä on yleinen käytäntö mobiilipeliteollisuudessa, sillä mobiilipelien tuottajilla on harvoin niin laaja jakelukanava, että sen avulla saataisiin mobiilipelille taattua laaja levikki (Son & Tan 2008). Mobiilipelien tarjoajan toiminta-kenttä koostuu pääosin seuraavista komponenteista: Pelipalvelusta, jonka avulla

mobiilipelien tarjoaja julkaisee ja jakaa uudet pelit, sekä toteuttaa peliin liittyvän yleisen tuen. Pelin tietokannasta, jossa on peliin liittyvien tietojen hallintatoiminnot. Kommunikointipalveluista, jotka voidaan toteuttaa usealla erilaisella tekniikalla ja lisäksi mobiililaitteen omaavista käyttäjistä. (Akkawi ym. 2004, 78.) Peliin liittyvät tietokanta- ja kommunikointipalvelut tarjoaa usein operationaalisen teknologian tarjoaja, joka tarjoaa myös pelin toimittamiseen ja mahdolliseen suoratoistoon (streaming) tarvittavan teknologian. Nykyisissä mobiilipeleissä oleellisena toimijana mukana on myös sisällön omistaja, joka pääsee vaikuttamaan usein mobiilipelin sisältöön ja ottaa myös voitoista oman osuuden. Käyttäjiltä tuleva rahavirta menee verkontarjoajan kautta, puhelinlaskun maksun yhteydessä, mobiilipelien tarjoajalle, josta osa tuloista jakautuu eteenpäin operationaalisten palveluiden tarjoajalle, sisällön omistajille ja mobiilipelien tuottajille. (Funk 2004, 77.) Kaikki toimijat eivät välttämättä ilmene mobiilipeliteollisuudessa erillisinä toimijoina, sillä esimerkiksi mobiilipelien tarjoaja voi itse hoitaa operationaalisen teknologian vaatimat tehtävät, tai verkon tarjoaja voi myös toimia mobiilipelien tarjoajana.

4.4 Mobiilipelien tuottaja

Mobiilipelien tuottajaorganisaatio muodostuu kolmesta alasta: ohjelmistotuotannosta, peliteollisuudesta ja mobiilijärjestelmistä. Ohjelmistotuotanto on käyty läpi luvussa 2, mobiilijärjestelmät tämän luvun ensimmäisessä kohdassa ja pelit sekä mobiilipelit tämän luvun toisessa kohdassa. Tässä kohdassa keskitytään mobiilipelien tuottajan toimialaan ja mobiilipelien tuottamiseen. Useat laitevalmistajat, kuten Nokia, pyrkivät pitämään uusien mobiilipelien tuottajien pääsyn toimialalle mahdollisimman helppona. Tämä varmistetaan esimerkiksi tarjoamalla kehityspakkeja (development kit), joita mobiilipelien tuottaja voi käyttää pelien kehityksessä (Son & Tan 2008, 37). Mobiilipelien tuottajalla on harvoin tarpeeksi laaja verkosto pelin tehokkaaseen markkinointiin ja jakamiseen, joten mobiilipelien tuottaja toimii usein tiiviissä yhteistyössä mobiilipelien tarjoajan kanssa. Son ja Tan (2008, 37–38) antavat artikkelissaan tästä useita esimerkkejä.

Perinteisten pelien tuottamiseen verrattuna, mobiilipelien tuottaminen eroaa esimerkiksi mobiilipelin kehitykseen ja markkinointiin liittyvien kustannuksien suuruuden osalta. Sonin ja Tanin (2008, 37) mukaan perinteisen pelin kehityksen ja markkinoinnin

kustannukset saattavat olla jopa miljoonia dollareita, kun mobiilipelin kehityksen ja markkinoinnin kustannukset ovat tyypillisesti 100 000–200 000 dollarin välillä. Informa Media Groupin mukaan mobiilipelituotanto tulee ohittamaan sekä konsolipelituotannon että PC-pelituotannon markkina-arvon vuoteen 2010 mennessä (Son & Tan 2008, 37). Tästä syystä myös perinteisten pelien tuottajat, kuten Electronic Arts (http://www.ea.com/), ovat alkaneet toimia mobiilipelituotannon puolella. Kyseisillä organisaatioilla on selkeä etu mobiilipelituottajiin verrattuna, sillä heillä on organisaatiossa valmiina konsepteja aikaisemmin tuottamistaan peleistä.

Mobiilipelejä pystyttiin aluksi tekemään vain WAP tai c-HTML merkkauskielellä (markup language). Tämä rajoitti huomattavasti normaalien videopelien siirtämistä matkapuhelimiin, sillä kyseisillä merkkauskielillä voitiin tehdä vain yksinkertaisia pelejä. Lisäksi mobiilipelien kehitys oli raskasta ja siten kallista. Muun muassa edellä mainitut tekijät hidastivat suurten pelientuottajaorganisaatioiden siirtymistä mobiilipeliteollisuuden puolelle. Mobiilipelien menestyksen kannalta ratkaiseva tekijä oli kehittyneempien ohjelmointikielien, kuten Javan, siirtäminen myös matkapuhelimiin. Javan ansiosta käyttäjien ei enää tarvinnut ottaa joka pelikerta yhteyttä palveluntarjoajaan, vaan peli pystyttiin lataamaan matkapuhelimeen. Java helpotti myös mobiilipelien kehitystä, sillä sen ansiosta jokaista mahdollista pelaajan tekemää toimintoa ei enää tarvinnut ohjelmoida erikseen, vaan toiminnot perustuivat yksinkertaisiin sääntöihin. Java toi mukanaan myös ominaisuuden, jonka avulla kuvia voitiin liikuttaa. Tämä yksinkertainen ominaisuus edesauttoi entistä kehittyneempien pelien kehityksessä ja houkutteli suuret pelientuottajaorganisaatiot mukaan mobiilipeliteollisuuteen. (Funk 2004, 68–70.) Tällä hetkellä yleisin Java-pohjainen sovelluskehys mobiilipelien tuotannossa on J2ME, C-kielen puolella yleisin mobiilipelien tuotannon sovelluskehys on BREW (Son & Tan 2008, 37). Muista sovelluskehyksistä Son ja Tan (2008, 37) mainitsevat Mophunin, ExEnin ja XNA:n.

Suosituimmat Java-pohjaiset mobiilipelit ovat J2ME sovelluskehyksellä kehitettyjä. Tarkemmin ottaen kyseisissä mobiilipeleissä ajonaikainen ympäristö (runtime environment) on koostunut J2ME:n CLDC (Connected Limited Device Configuration) konfiguraatiosta ja MIDP 1.0 (Mobile Information Device Profile) profiilista. CLDC on matkapuhelimiin käytettävä konfiguraatio J2ME:stä. Nykyisin käytössä on MIDP profiilista versio 2.0.

MIDP 2.0:n tarjoama Game API Java luokkien sarja on yksinkertaistanut mobiilipelien kehitystä huomattavasti verrattuna versioon 1.0. Game API Java luokkien sarja koostuu viidestä Java luokasta ja kyseinen sarja on javax.microedition.lcdui.game paketissa (Williams & Burge 2004, 37).

Muun muassa mobiilit laitteet ja laitteissa käytettävät sovelluskehykset kehittyvät jatkuvasti ja mobiilipelien tuottajien täytyy pysyä mukana kehityksessä. Täten mobiilipelien tuottajilla on omien, peleihin liittyvien kehitysprojektien lisäksi, suuret paineet versionpäivityksiin laitteiston valmistajien puolesta. Esimerkkinä standardoinnista voidaan pitää kolmannen sukupolven mobiilijärjestelmien teknisen spesifikaation saattamista maailmanlaajuisesti yhtenäiseksi (Akkawi ym. 2004, 77). Mobiilipeliteollisuuden puolelta standardoinnin tilannetta taas voidaan kuvata seuraavasti: standardointi on hyvästä. Valitettavasti instituutioita, jotka toteuttavat sitä on vähän (Sirén 2004). Muun muassa tämän ongelman odotetaan tulevan esille empiirisen tutkimuksen aikana.

5 EMPIIRINEN TUTKIMUS

Tähän mennessä tutkielmassa on käsitelty kirjallisuuskatsauksena perinteisen ohjelmistotuottajan organisaatiota, version- ja konfiguraationhallintaa ohjelmistotuotannossa, mobiilijärjestelmiä sekä mobiilipelituotantoa. Kirjallisuuskatsauksen pohjalta on rakennettu tarkka suunnitelma ja tutkimuskehys empiiriselle tutkimukselle. Tieteellisen pohjan rakentaminen empiiriselle tutkimukselle on tärkeää, sillä empiirisen tutkimuksen aineistoa ei pidä kerätä epämääräisen ajatuksen pohjalta, vaan empiirisen tutkimuksen tukena toimii laadukkaasti toteutettu kirjallisuuskatsaus (Hirsjärvi & Hurme 2000, 13). Tässä tutkielmassa on noudatettu Hirsjärven ja Hurmeen (2000, 14) mukaista tutkimusprosessia, jonka avulla on varmistettu empiirisen tutkimuksen tarvitsemat taustatiedot. Luvun ensimmäisessä kohdassa käydään läpi empiirisen tutkimuksen tausta ja tavoitteet. Kohta 5.2 tuo esille empiirisen tutkimuksen rakenteen ja tästä edetään kohtaan 5.3, jonka aiheena ovat tutkimusmenetelmät. Näihin kohtiin ja kirjallisuuskatsaukseen perustuen, on rakennettu empiirisessä tutkimuksessa käytettävä tutkimuskehys. Tutkimuskehys on esitelty luvussa 5.4. Kohdassa 5.5 käydään läpi tutkimusprosessin eteneminen alusta loppuun. Luvun tarkoituksena on esittää empiirisen tutkimuksen suunniteltu kokonaisuus ja toteutuminen. Empiirisen tutkimuksen vastaukset käydään läpi ja analysoidaan seuraavassa luvussa.

5.1 Empiirisen tutkimuksen tausta ja tavoitteet

Perinteistä ohjelmistotuotantoa käsittelevän kirjallisuuskatsauksen pohjalta on kartoitettu version- ja konfiguraationhallinnan ongelmia ja ongelmien ratkaisuja. Tämän kirjallisuuskatsauksen pohjalta rakennettua tutkimuskehystä sovelletaan mobiilipelien tuottajaan, sillä mobiilipelien tuottaja on yksi ohjelmistotuotannon osa-alueista. Empiirisen tutkimuksen kohteeksi olisi hyvin voitu valita myös esimerkiksi ERP-järjestelmiä valmistavat ohjelmistotuottajat, mutta tutkimuskehystä halutaan soveltaa tuoreempaan ja perinteisestä ohjelmistotuotannosta hieman poikkeavaan toimialaan. Mobiilipelien tuottaja sopii tähän tarkoitukseen oivallisesti, sillä mobiilipeliteollisuudessa kohdelaitteistojen (matkapuhelimien) vaihtuvuus on eri luokkaa kuin ERP-järjestelmiä kehittävän organisaation kohdelaitteistojen (asiakkaiden IT-infrastruktuuri). Lisäksi voisi olettaa, että mobiilipelien

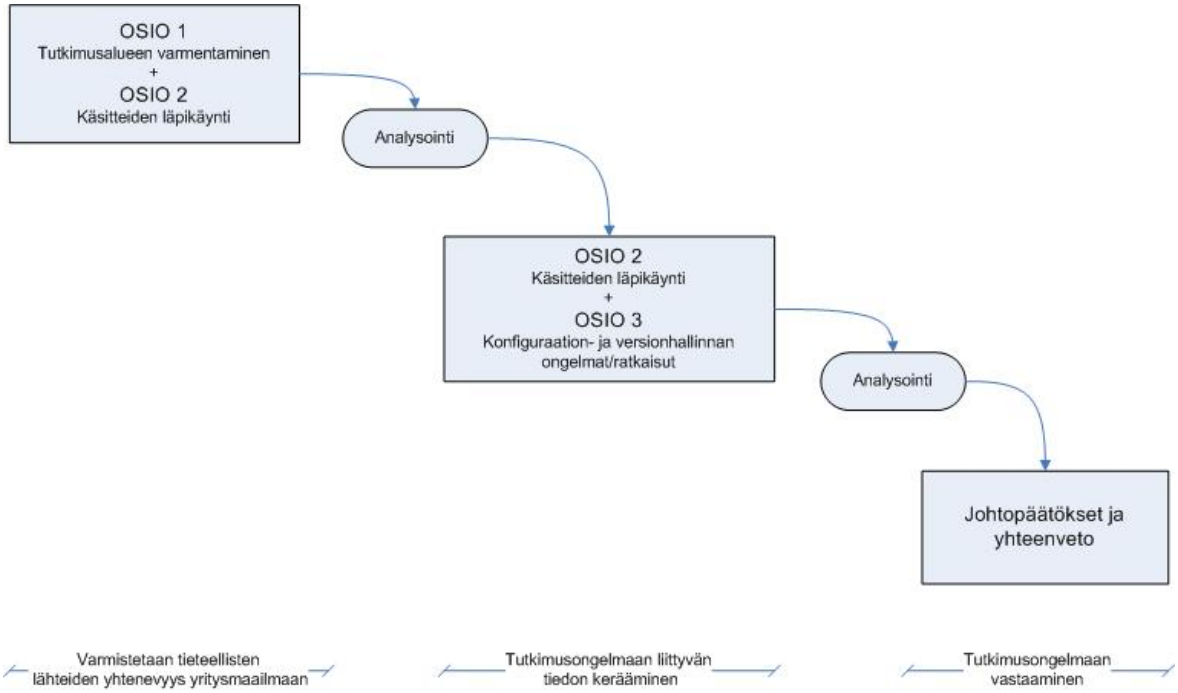
tuottaminen eroaa myös audio-visuaalisen sisällön ja asiakassuhteen takia. Tutkielman tutkimuskehys on rakennettu keskittyen seuraaviin tutkimusongelmiin:

- Esiintyykö mobiilipelituotannossa ohjelmistojen konfiguraation- ja versionhallinnan yleisimpiä ongelmia?
- Miltä osin ohjelmistojen konfiguraation- ja versionhallinnan ratkaisumallit soveltuvat mobiilipeliteollisuuteen?

Empiiriseen tutkimukseen käytettävä tutkimuskehys on tehty kirjallisuuskatsauksen pohjalta, täten tutkielmassa sovelletaan tieteellisistä lähteistä saatavaa tietoa yhteen ohjelmistotuotannon toimialaan (mobiilipelituottajaan). Toisin sanoen tutkielmassa on tavoitteena verrata teoreettiselta pohjalta kerättyä tietoa yhteen rajattuun käytännön kohteeseen. Tutkimustuloksina empiirisestä osuudesta on tavoitteena saada tietoa perinteisestä ohjelmistotuotannosta kirjoitettujen tieteellisten julkaisuiden tulosten soveltuvuudesta mobiilipelituotantoon. Tutkimuksen jälkeen pystytään muun muassa päättelemään, ovatko tämän hetken tieteellisten lähteiden ratkaisut tarpeeksi yleisellä tasolla, jotta ratkaisuja voitaisiin soveltaa esimerkiksi tutkimuskohteena olevaan mobiilipelituotantoon.

5.2 Empiirisen tutkimuksen rakenne

Empiirinen tutkimus koostuu useista osioista ja osiot useista teemoista. Laajuuden takia tässä kohdassa käsitellään empiirisen tutkimuksen koostumus pääosin. Kirjallisuuskatsauksessa edettiin yleisestä erityiseen. Ensiksi tarkasteltiin perinteisen ohjelmistotuottajan organisaatiorakennetta, josta poimittiin tutkielman kannalta oleelliset tehtäväalueet ja tehtäväalueilta oleelliset avainprosessit. Kyseinen rakenne ei ole satunnaisesti valittu, vaan rakenteen suunnittelussa on otettu huomioon empiirisen tutkimuksen looginen eteneminen. Empiirinen tutkimus jakautuu kolmeen osioon (kuvio 14).



KUVIO 14. Empiirisen tutkimuksen rakenne

Ensimmäinen osio on nimeltään tutkimusalueen varmentaminen. Osiossa tutkitaan mobiilipeliteollisuuskentän ja mobiilipelien tuottajaorganisaation rakennetta vertaamalla mobiilipelien tuottajien näkemystä tieteellisiin lähteisiin. Tutkimusalueen varmentaminen (Osio 1) koostuu seuraavista teemoista:

- Verrataan kirjallisuuskatsauksesta saatua mobiilipeliteollisuuden rakennetta (kuvio 13) mobiilipelituottajien näkemykseen mobiilipeliteollisuuden rakenteesta.

- Tutkitaan esiintyykö mobiilipelituottajaorganisaatioissa tutkielman kannalta oleellisimmat perinteisen ohjelmistotuottajaorganisaation tehtäväalueet. Näitä tehtäväalueita olivat ohjelmistojen kehitys ja ylläpito. Lisäksi kartoitetaan, kuinka kohdeorganisaatiot erottelevat toisistaan mobiilipelien kehityksen ja ylläpidon.

- Varmistetaan, että kohdeorganisaatiossa suoritetaan konfiguraation- ja version-hallintaa.

Toisessa osiossa toimitetaan kohdeorganisaatiolle määritelmät tärkeimmistä termeistä. Tämän osion avulla varmistetaan, että osioissa 1 ja 3 käytettävä terminologia on yhtenäistä tieteellisten määritysten kanssa. Alla on lista termeistä, joiden tarkka määritelmä toimitetaan kohdeorganisaatioille.

- Versionhallinta
- Konfiguraationhallinta
- Ohjelmisto-objekti
- Revisio
- Variantti

Toista osiota, eli käsitteiden läpikäyntiä, suoritetaan osioiden 1 ja 3 yhteydessä. Osion 1 jälkeen analysoidaan saadut vastaukset. Analysoitava tieto koostuu siten ensimmäisen osion teemoihin liittyen. Analysointi suoritetaan ennen kolmatta osiota, koska ensimmäisestä osiosta on tarkoitus saada kolmatta osioita tukevaa tietoa.

Kolmannessa osiossa tutkitaan ohjelmistojen konfiguraation- ja versionhallinnan ongelmia ja ratkaisuja mobiilipelituotannossa. Osioon on rakennettu kolme viitekehystä, viitekehyksiä ovat:

- Konfiguraationhallinnan tekninen näkökulma
- Konfiguraationhallinnan hallinnollinen näkökulma
- Versionhallinta

Kolmannen osion avulla pyritään saamaan tietoa tutkimusongelmiin. Osion jokaiseen viitekehykseen kuuluu useita teemoja ja näihin teemoihin pohjautuen kysymyksiä. Kysymyslistan avulla saadaan spesifiä tietoa teemoihin liittyvistä ongelmista, lisäksi vastauksien avulla voidaan päätellä ovatko ratkaisut samankaltaisia ja jos eivät ole, niin

mistä erot johtuvat. Kolmannen osion jälkeen suoritetaan vastausten analysointi, johtopäätökset ja yhteenveto.

5.3 Tutkimusmenetelmät

Kirjallisuuskatsauksen, tutkimusongelmien ja käytännön rajoitteiden pohjalta on tutkielmaan valittu käytettävät tutkimusmenetelmät. Tutkimusmenetelmiä ovat lomakekysely ja teemahaastattelu. Empiirinen tutkimus oli siis jaettu kolmeen osioon. Ensimmäinen osio toteutetaan lomakekyselynä, toinen osio koostuu termien määritelmien yhtenäistämisestä ja viimeinen osio toteutetaan joko lomakekyselynä tai teemahaastatteluna, riippuen kohdeyrityksen toiveista. Ensimmäisessä osiossa käytetään lomakekyselyä, koska kyseisen osion ilmiöt ovat konkreetteja ja ne voidaan siten toteuttaa tehokkaasti ilman haastattelua (Hirsjärvi & Hurme 2000, 37). Kolmannessa osiossa tutkittavat aiheet ovat monimutkaisia ja toisiinsa kietoutuneita. Tässä osiossa tarvitaan menetelmää, jota voidaan tarvittaessa mukauttaa tutkimuksen edetessä. Täten teemahaastattelu sopii hyvin viimeisen osan menetelmäksi. Teemahaastattelu on kvalitatiivinen tutkimus, jossa haastattelija ja haastateltava ovat vuorovaikutuksessa (Hirsjärvi & Hurme 2000, 23). Kuitenkin ottaen huomioon yritysten kiireiset aikataulut, on empiirisen tutkimuksen viimeinen osa suunniteltu toteutettavaksi myös lomakekyselyn avulla, tarpeen vaatiessa. Lomakekyselyn toimivuuden parantamiseksi osioissa 3, on lomakekyselyn rakenne suunniteltu erityisesti sopimaan tähän tutkimusongelmaan.

Kvalitatiivisessa tutkimuksessa tarkkuus ja luotettavuus saavutetaan verifioimalla, tästä syystä kohdeorganisaatioille annetaan mahdollisuus kommentoida tieteellisten lähteiden ratkaisumalleja. Kommentointimahdollisuus on sekä lomakehaastattelussa että teemahaastattelussa. Kvalitatiivisessa tutkimuksessa tuotoksena on usein hypoteeseja ja ankkuroituja teorioita (Hirsjärvi & Hurme 2000, 25). Toisin sanoen tutkielmassa rakennettujen ratkaisumallien toimivuutta päästään testaamaan aikaisintaan tutkielman valmistuttua, mikäli kohdeorganisaatiot haluavat käyttää empiirisen tutkimuksen tuloksia hyödyksi. Kvalitatiivisen tutkimuksen toteutukseen voisi soveltaa myös muita menetelmiä, kuten Nunamakerin Chenin ja Purdinin (Nunamaker ym. 1991 mukaan, Mathiassen 1998, 69) esittelemää osallistuvaa havainnointia. Osallistuvassa havainnoinnissa tutkija menee

yritykseen paikan päälle seuraamaan työskentelyä ja tekee sitä kautta havaintoja. Osallistuva havainnointi hylättiin menetelmänä, koska sen toteutus vaatii liikaa aikaa, varsinkin kun empiiriseen tutkimukseen pyritään ottamaan mukaan useita organisaatioita. Samoin perustein hylättiin päiväkirja ja kirjoitelma menetelmät. (Hirsjärvi & Hurme 2000, 28.)

Lomakekyselyssä lomakkeelle on kirjoitettu joukko kysymyksiä, joihin vastaajiksi valittujen toivotaan vastaavan. Lomakekyselyn etuna teemahaastatteluun on se, että vastaaja voi itse vapaasti valita ajan, jolloin vastaa kysymyksiin. Useisiin ensimmäisessä osiossa tiedusteltaviin ongelmakohtiin ei ole olemassa valmista vastausluokitusta. Tästä syystä tutkielmassa lomakekysely suoritetaan avoimena, eli vastaajalle ei anneta valmiita vastauksia, vaan vastaukset tulee kirjata omin sanoin. (Järvinen & Järvinen 2000, 155.)

Hirsjärven ja Hurmeen (2000, 34) mukaan haastattelu on nykyään käytetyimpiä tiedonkeruumuotoja. Ongelmana he mainitsevat vapaamuotoisten haastatteluiden yleistymisen. Vapaamuotoinen haastattelu voi ennalta suunnittelemattomuutensa takia vaihtua keskusteluksi, vaikka haastattelun pitäisi olla informaation keräämiseen tähtäävää ja ennalta suunniteltua päämäärähakuista toimintaa (Hirsjärvi & Hurme 2000, 42). Tästä syystä tutkielmassa on empiirisen tutkimuksen ensimmäinen osio toteutettu lomakekyselynä, joka on hyvin strukturoitu menetelmä ja viimeinen osio joko lomakekyselynä tai teemahaastatteluna, joka on puolistrukturoitu menetelmä. Vaikka teemahaastattelu onkin vapaamuotoisempi, kuin lomakekysely, saadaan tutkielmassa ensimmäisen osion lomakekyselyistä teemahaastattelua tukevaa tietoa ja täten teemahaastatteluun vahvempi strukturointi.

Teemahaastattelussa haastattelu kohdennetaan tiettyihin teemoihin ja se ei edellytä kokeellisesti aikaansaatua yhteistä kokemusta (Hirsjärvi & Hurme 2000, 48). Teemahaastattelun selvänä etuna lomakekyselyyn, varsinkin monimutkaisissa ja laajoissa aiheissa, on teemahaastattelun tarjoama mahdollisuus suunnata tiedonhankintaa itse haastattelutilanteessa (Hirsjärvi & Hurme 2000, 34). Lisäksi teemahaastattelu sopii tilanteeseen, jossa kohteena on vähän kartoitettu alue tai haastateltavan puhe halutaan sijoittaa laajempaan kontekstiin (Hirsjärvi & Hurme 2000, 34). Haastattelu menetelmänä antaa myös mahdollisuuden motivoida haastateltavia, haastateltava voi pyytää täsmennystä kysymyksiin ja haastattelun

avulla saadaan kuvaavia esimerkkejä (Hirsjärvi & Hurme 2000, 36). Teemahaastattelu sisältää myös haittoja. Haastattelijalla pitäisi olla taitoa, kokemusta ja koulutusta haastattelijan roolista. Teemahaastattelu vie paljon aikaa, kun mukaan otetaan haastatteluiden sopiminen, haastateltavien etsiminen, toteutus ja aineiston analyysi. (Hirsjärvi & Hurme 2000, 35.) Nämä haittapuolet on empiiristä tutkimusta suunniteltaessa otettu huomioon ja arvioitu, että ne eivät estä empiirisen tutkimuksen kolmannen osion toteuttamista teemahaastattelulla. Litterointi ei ole tutkielman empiiristä tutkimusta hidastava tekijä, kuten yleensä, sillä kohdeyritysten määrän pysyessä todennäköisesti alle kymmenessä ja samalla haastattelumateriaalin ollessa spesifi, litterointimateriaalin määrä on suhteellisen vähäinen.

5.4 Tutkimuskehys

Empiirisessä tutkimuksessa edetään kirjallisuuskatsauksen rakenteen mukaisesti, edeten yleisestä erityiseen. Tämä tarkoittaa sitä, että tieteellisten lähteiden pohjalta koostettu tutkimuskehys etenee osioittain kuvion 14 mukaisesti. Ensin suoritetaan tutkimusalueen varmentaminen (Osio 1) ja käsitteiden läpikäynti (Osio 2) tutkimusalueen varmentamisen kannalta oleellisten termien kannalta. Tutkimusalueen varmentamisen jälkeen aloitetaan konfiguraation- ja versionhallinnan ongelmien/ratkaisujen (Osio 3) suorittaminen yhdessä käsitteiden läpikäynnin kanssa, kyseisen osion kannalta oleellisten termien osalta. Tutkimuskehys koostuu siis seuraavista osioista:

1. Tutkimusalueen varmentaminen
2. Käsitteiden läpikäynti
3. Konfiguraation- ja versionhallinnan ongelmat/ratkaisut

Osio 1 suoritetaan yhteydenoton yhteydessä liitetiedostojen avulla. Osion 2 sisältöä on sekä osiossa 1 että osiossa 3, sillä molemmissa osioissa on tarkennusta vaativia käsitteitä. Osio 3 suoritetaan joko teemahaastatteluna tai lomakekyselynä, riippuen kohdeorganisaation valinnasta. Preferenssiksi osioon 3 asetetaan teemahaastattelu, mutta haastateltavien mahdollisen kiireellisyyden vuoksi, annetaan toiseksi vaihtoehdoksi kyselylomake.

5.4.1 Tutkimusalueen varmentaminen

Ensimmäisen osion, eli tutkimusalueen varmentamisen, toteutustapana toimii lomakekysely. Kyselyssä fokusoidutaan organisatorisiin aihealueisiin. Ensimmäiseksi tutkitaan mobiilipeliteollisuuden rakennetta. Tämän jälkeen syvennytään mobiilipelien tuottajaan organisaationa ja viimeiseksi käsitellään mobiilipelien kehitystä ja ylläpitoa ja niihin liittyviä avainprosesseja. Lomakekysely löytyy liitteenä (liite 5).

Mobiilipeliteollisuuden rakennetta tutkittaessa käytetään kuviota 13 hyväksi, verraten tieteellisten lähteiden näkemystä mobiilipelien tuottajien näkemykseen mobiilipeliteollisuudesta. Tätä kautta saadaan vastaus aiheeseen: näkeekö mobiilipelien tuottaja oman roolin mobiilipeliteollisuudessa samanlaisena kuin tieteellisissä lähteissä mobiilipelien tuottajan rooli kuvataan.

Empiirinen tutkimus tarkentuu seuraavaksi organisaatiotasolle, kuten kirjallisuuskatsauksessakin. Tällöin verrataan perinteisen ohjelmistotuottajan organisaatiorakennetta mobiilipelien tuottajan organisaatiorakenteeseen. Vertaaminen suoritetaan vain tutkielman kannalta oleellisten tehtäväalueiden osalta. Tuotoksena tästä on tärkeää tietoa kohdeorganisaation tehtäväalueista, jonka avulla voidaan varmentaa avainprosessien vastaavuus/eroavaisuus toimialojen välillä. Tieteellisten lähteiden perusteella ohjelmistotuotannossa kehitys ja ylläpito tehtäväalueina erotetaan usein toisistaan (April ym. 2005, 199). Tarkemmin sanottuna kohteista selvitetään, miten kyseisissä organisaatioissa erotetaan mobiilipelien kehitys ja ylläpito toisistaan. Suorittavatko kyseisiä tehtäväalueita eri tiimit, onko näille tehtäväalueille määritelty erilaisia avainprosesseja vai kuuluvatko mobiilipelien kehitys ja ylläpito samaan kokonaisuuteen (mobiilipelituotantoon). Osion tärkein tehtävä kuitenkin on selvittää, löytyykö mobiilipelien tuottajaorganisaatiosta konfiguraation- ja versionhallinta avainprosessit (toisin sanoen tutkielmassa tarkkailtavat avainprosessit).

5.4.2 Käsitteiden läpikäynti

Hirsjärven ja Hurmeen (2000, 53) mukaan tutkimuksen suunnittelussa täytyy ottaa huomioon se tekijä, että kieleen ja käsitteisiin liittyvät ongelmat ovat keskeisiä, koska jopa

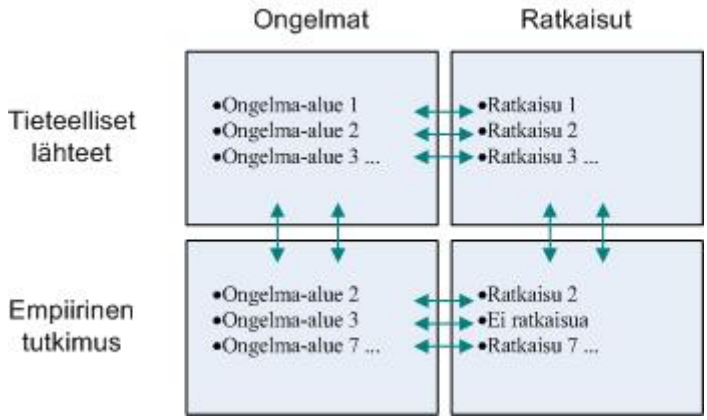
yksinkertaisissa ja tavallisimmissakin ilmauksissa saattaa ilmetä epäselvyyksiä. Toinen osio, eli käsitteiden läpikäynti suoritetaan osioiden 1 ja 3 yhteydessä. Käsitteiden läpikäynti jakautuu siten, että osion 1 lomakekyselyssä esitetään sen kannalta tärkeät käsitteet ja osion 3 alussa esitetään sen kannalta tärkeät käsitteet. Tämä antaa mahdollisuuden haastateltavalle tukeutua molemmissa osioissa ennalta määriteltyihin käsitteisiin ja siten saadaan yhtenäistettyä kohdeorganisaatioiden termistö.

Määriteltyjä käsitteitä osiossa 1 ovat konfiguraation- ja versionhallinta. Osiossa 3 määriteltyjä termejä ovat revisio, variantti, ohjelmisto-objekti sekä muistutukseksi uudelleen konfiguraation- ja versionhallinta. Kyseisten käsitteiden läpikäynti kohdeorganisaatioiden kanssa, ennen itse haastattelun suorittamista, on perusteltua, sillä osa kohdeorganisaatioista saattaa nähdä käsiteltävien termien sisällön erilailla, kuin tieteellisten lähteiden perusteella on määritelty. Tämän osion tavoitteena on varmistaa sekä osion 1 viimeisen kysymyksen että osion 3 kysymysten validius kohdeorganisaatioiden kesken.

5.4.3 Konfiguraation- ja versionhallinnan ongelmat/ratkaisut

Konfiguraation- ja versionhallinnan ongelmat/ratkaisut (Osio 3) on empiirisen tutkimuksen pääkohde. Tässä alakohdassa esitettyjen viitekehysten avulla kartoitetaan ratkaisut tutkielman tutkimusongelmiin. Viitekehysten avulla pyritään minimoimaan haastattelun koostuminen pelkistä irrallisista toteamuksista, toisin sanoen haastatteluun saadaan aikaan diskurssi (Hirsjärvi & Hurme 2000, 51). Viitekehykset käyttävät kuviossa 15 esitetyn esimerkin mukaista logiikkaa.

Tutkielman tieteellisistä lähteistä on viitekehyksiin koostettu perinteiseen ohjelmistotuotantoon liittyvät ongelmat ja ongelmien ratkaisut, liittyen konfiguraation- ja versionhallintaan. Empiirisessä tutkimuksessa kartoitetaan mobiilipelien tuottajien kokemat konfiguraation- ja versionhallinnan ongelmat. Lisäksi selvitetään mobiilipelien tuottajien tämänhetkiset mahdolliset ratkaisut kyseisiin ongelmiin. Kyseisiä ratkaisumalleja voidaan siten verrata tieteellisten lähteiden tarjoamiin ratkaisuihin. Haastattelu (tai lomakekysely) pyritään pitämään mahdollisimman suorana, toisin sanoen haastattelun apuna ei käytetä kuvioita, joita haastateltavan pitäisi tulkita (Hirsjärvi & Hurme 2000, 41).



KUVIO 15. Empiirisen tutkimuksen kolmannen osion ratkaisumalli

Ensimmäisessä viitekehyksessä käsitellään konfiguraationhallintaa teknisestä näkökulmasta. Tekniseltä kannalta ratkaisuina toimivat tieteellisten lähteiden mukaan pääosin konfiguraationhallintajärjestelmät. Jos empiirisen tutkimuksen aikana nousee esille jokin ennalta määritellyistä ongelmista, lisää tarkennusta voidaan tarvittaessa hakea tieteellisistä lähteistä. Viitekehyksen teemoihin kuuluvat Dartin (1991, 7) määrittelemät kategoriat. Jokainen viitekehyksen kysymys (liite 6) pohjautuu johonkin näistä teemoista. Numerot kunkin teeman perässä kertovat, mitkä teknisen näkökulman viitekehyksen kysymykset kuuluvat kyseisen teeman alaisuuteen. Esimerkiksi teeman Rakenne kysymykset ovat teknisen näkökulman viitekehyksen kysymykset 1 ja 7 eli: ”Alalla on esiintynyt ongelmia suurikokoisten mobiilipelien rakenteen hallinnassa?” ja ”Mobiilipelien tuottamisessa on ilmennyt ongelmia tehtyjen/tehtävien muutosten vaikutuksien ymmärtämisessä?”

- Ohjelmisto-objektit: 5
- Rakenne: 1, 7
- Rakentaminen: 3, 6
- Hallinta: 4
- Prosessi: 2

- Tiimi: 8

Toisessa viitekehyksessä käsitellään konfiguraationhallinta hallinnollisesta näkökulmasta. Hallinnollisen näkökannan ongelmat ja ratkaisut pohjautuvat pääosin Abranin ym. (2004) ja Estublierin ym. (2005) esittämiin aihealueisiin ja prosesseihin. Alla olevassa listassa ovat kyseisen viitekehyksen teemat. Viitekehyksen kysymykset ja kohdeorganisaatioille esitetyt ratkaisumallit löytyvät liitteestä 6.

- Prosessinhallinta: 2, 3
- Muutoksenhallinta: 4, 5
- Tilanvalvonta: 6
- Laadunvalvonta: 7
- Konfiguraatioiden tunnistaminen: 1
- Jakelunhallinta: 8

Versionhallintaa ei ole jaettu tieteellisessä lähteissä selkeästi kahteen osaan, tekniseen ja hallinnolliseen, kuten konfiguraationhallinta on jaettu. Erittelyn tekeminen olisi mahdollista, sillä myös versionhallinnassa käytetään teknisiä ratkaisuja (esimerkiksi versiokirjasto ja graafiset esitykset ohjelmistotuotteen elinkaaresta), mutta se vaatisi huomattavan suurta työmäärää tutkielman kokoon nähden. Tästä syystä versionhallinnan teemat koostuvat versionhallinnan suurimmista kokonaisuuksista: versiokirjastosta, ohjelmistotuotteen elinkaaresta ja versiointimallista. Viitekehykseen kuuluvat teemat ovat alla olevassa listassa ja kysymykset ratkaisuineen liitteessä 6.

- Versiokirjasto: 1, 5, 6
- Ohjelmistotuotteen elinkaari: 3, 7
- Versiointimalli: 2, 4, 8

5.5 Tutkimusprosessin eteneminen

Kohdeorganisaatioihin on tutustuttu yleisellä tasolla ennen yhteydenottoa. Tiedot on kerätty kunkin yrityksen www-sivustoilta. Tietojen kerääminen tehtiin yhteydenoton profiloimisen vuoksi, lisäksi tiedoista on apua teemahaastattelun suorittamiseen. Koska kohdeorganisaatiot haluttiin pitää salattuina, ei profiloimiseen liittyviä tietoja esitetä tutkielmassa. Tutkimusta varten kartoitettiin mobiilipelien tuottajaorganisaatiot Suomesta, aktiivisia organisaatioita löytyi yhteensä 12 kappaletta. Organisaatiot kartoitettiin useiden maksuttomien yritysluetteloiden ja yritysten omien www-sivujen avulla. Tutkimukseen otettiin mukaan kuusi organisaatiota. Vastaajiksi haluttiin yrityksen organisatorisista tehtävistä perillä oleva henkilö. Vastaajan ei siten tarvinnut olla välttämättä tietohallintojohtaja, vaan riitti, jos vastaaja oli aktiivisesti mukana tietohallinnossa.

Ensimmäiseen yhteydenottoon otettiin kolme yritystä. Näistä kaksi valittiin maantieteellisen sijaintinsa vuoksi (Jyväskylä ja Tampere) ja kolmas organisaatio valittiin useiden Helsingissä sijaitsevien organisaatioiden joukosta, perusteena suureen kansainväliseen organisaatioon kuuluminen. Yhteydenotto kohdeorganisaatioihin tehtiin 21.2.2008. Yhteydenoton yhteydessä laitettiin mukaan ensimmäisen osion kyselylomake ja toisen osion sisältämät määritykset. Varmuuden vuoksi kyselylomake ja määritykset laitettiin .pdf, .doc ja .jpg -muodoissa. Aikarajaksi vastauksiin asetettiin 3.3.2008. Yhteydenoton avulla sovittavaan teemahaastatteluun asetettiin ajankohdaksi viikot 11–13. Tällä tavoin kohdeorganisaatioille jäi aikaa vastata ensimmäiseen osioon kahdeksan arkipäivää ja teemahaastattelun järjestämiselle kolmen viikon päivät valita sopiva ajankohta.

Kolmesta alkuperäisestä kohdeorganisaatiosta aikataulussa vastasi vain yksi organisaatio. Kyseinen organisaatio varmisti osanottonsa välittömästi sähköpostitse ja toimitti vastaukset ensimmäisen osion kyselylomakkeeseen 3.3.2008. Kuten alustava tutkimus oli paljastanut, yksi ensimmäisistä kolmesta kohdeorganisaatiosta oli viimeisten kahden vuoden aikana panostanut enimmäkseen digitaaliseen mainontaan, joka todennäköisimmin oli syy organisaation jättäytymiselle tutkimuksen ulkopuolelle. Kun toinenkin, kolmesta kohdeorganisaatiosta, jätti vastaamatta, tutkimukseen otettiin mukaan neljäs kohdeorganisaatio

2.3.2008. Vastausaikaa kyseiselle organisaatiolle annettiin 10.3.2008 asti, jolloin teemahaastattelun ajankohta kyettiin säilyttämään viikoilla 11–13. Suostuminen tutkimukseen ja vastaukset neljänneltä organisaatiolta saatiin 3.3.2008, eli reagointi kyseisessä organisaatiossa oli todella nopeaa.

Ensimmäisestä osiosta saatujen vastausten perusteella yrityksillä oli pieniä näkemyseroja mobiilipeliteollisuuden rakenteesta. Koska saturaatio saavutettiin ensimmäisessä osiossa tieteellisiin lähteisiin verrattuna tarpeeksi suurella tasolla ja empiirisen tutkimuksen aikataulua ei haluttu venyttää, ensimmäisen osion suorittamista ei koettu enää oleelliseksi lopuille kohdeorganisaatioille. Tutkimuksen kokonaistulosten kannalta katsottiin parhaaksi laajentaa kuitenkin version- ja konfiguraationhallintaan liittyvien ongelmien/ratkaisujen kartoitusta (eli osiota 3), joten tutkimuksen kohdeorganisaatioiden tavoitemäärää lisättiin 4–5 kappaleeseen. Oleellista ei ollut kohdeorganisaatioita lisäämällä siis saada saturaatiolle lisävarmistusta ensimmäisessä osiossa, vaan lisätä kolmannen osion vastausten validisuutta. 14.3.2008 lähettiin neljälle organisaatiolle haastattelupyynnöt. Haastattelupyyntöjen avulla ei saatu yhtään uutta kohdeorganisaatiota, joten 2.4.2008 täytyi turvautua suoraan yhteydenottoon puhelimitse. Soitot tehtiin kolmelle organisaatiolle, osa näistä organisaatiosta oli saanut jo haastattelupyynnön, mutta ei ollut vastannut. Soittojen toimivuus oli huomattavasti tehokkaampaa kuin haastattelupyynnöt, sillä soittojen avulla tutkimukseen lupautui mukaan kolme organisaatiota. Tosin haastattelupyynnöt saattoivat toimia hyvänä pohjatietona vastaajille ja suostuminen tutkimukseen oli täten helpompaa. Kolmesta kohdeorganisaatiosta vain yksi vastasi ajoissa, joten puhelimitse tehtyjä yhteydenottoja suoritettiin lisää. Yhteydenotot suoritettiin 18.4.2008 ja 23.4.2008. Yhteydenottojen avulla neljä kohdeorganisaatiota lupautui osallistua tutkimukseen. Lopulta kolme näistä neljästä kohdeorganisaatiosta vastasi empiirisen tutkimuksen kysymyksiin.

Haastattelupyyntöjä lähetettiin siis kokonaisuudessaan 13 organisaatiolle, joista kuusi osallistui tutkimukseen ja vastasi esitettyihin kysymyksiin. Haastattelupyyntöjen määrä on 13, eli suurempi kuin mobiilipelituottajien kartoitettu määrä, koska yksi haastattelupyyntö lähti organisaatiolle, joka ei ole enää aktiivinen mobiilipelituottaja. Täten osallistumisosuudeksi saadaan 50 % Suomessa toimivista mobiilipelituottajaorganisaatioista.

Kohdeorganisaatioiden vastaajia olivat muun muassa kehityspäällikkö, tekninen päällikkö ja toimitusjohtaja, joten vastaajiksi onnistuttiin saamaan ammattitaitoiset ja organisatorisista asioista perillä olevat henkilöt.

Ensimmäinen mukaan lähtenyt kohdeorganisaatio halusi suorittaa kolmannen osion teemahaastatteluna, teemahaastattelun ajankohdaksi sovittiin alustavasti 28.3.2008 klo 13. Kyseinen ajankohta sovittiin jo 3.3.2008, eli ensimmäisen osion kyselylomakkeen yhteydessä, kuten oli suunniteltu. Toinen mukaan lähtenyt kohdeorganisaatio halusi suorittaa kolmannen osion kyselylomakkeena Korppi-järjestelmän avulla. Kyselylomake valmistui 13.3.2008 ja täten kohdeorganisaatiolle jäi vastausaikaa 13.3.2008–28.3.2008 välinen aika. Kyselylomake tehtiin alustavasti valmiiksi jo 10.3.2008, mutta kyseessä ollessa laadullinen tutkimus, tarkastutettiin kyselylomake tutkielman ohjaajalla. Lisäksi kyselylomakkeen toimivuus (muun muassa yleinen toimivuus, datan saanti yhdellä kertaa vastatessa ja useilla kerroilla vastatessa) testattiin yliopistonverkon ulkopuoliselta koneelta ja kysymysten ymmärrettävyys testattiin IT-alan ammattilaisella. Kyseisillä toimilla pyrittiin takaamaan laadullisen tutkimuksen onnistuminen sekä toimivuuden kannalta että kysymysten laadun kannalta. Vastaukset kohdeorganisaatioilta saatiin seuraavan listan mukaisesti:

- kohdeorganisaatio A, kyselylomake, 17.3.2008
- kohdeorganisaatio B, teemahaastattelu, 28.3.2008
- kohdeorganisaatio C, kyselylomake, 4.4.2008
- kohdeorganisaatio D, kyselylomake, 21.4.2008
- kohdeorganisaatio E, kyselylomake, 23.4.2008
- kohdeorganisaatio F, kyselylomake, 24.4. ja 28.4.2008.

Suurin osa kohdeorganisaatioista halusi suorittaa konfiguraation- ja versionhallintaan liittyvän haastattelun lomakekyselynä. Kohdeorganisaatio B:ltä teemahaastattelun avulla saadut vastaukset osoittautuivat sisällöltään rikkaammaksi, kuin lomakekyselyn avulla

saadut. Tämä on täysin ymmärrettävää, sillä kysymyksiä lomakekyselyyn tuli yhteensä 24 kappaletta ja kaikkiin näihin kysymyksiin kattavan vastauksen kirjoittaminen vie todella paljon aikaa. Hyvin onnistuneen suunnittelun ja toteutuksen avulla teemahaastattelun tulosten litterointikin sujui kohtuullisen nopeasti. Näiden tekijöiden pohjalta teemahaastattelun suorittaminen vastaavissa tutkielmissa on suositeltavaa, joskin teemahaastattelun valmisteleminen vaatii paljon työtä.

6 EMPIIRISEN TUTKIMUKSEN TULOKSET

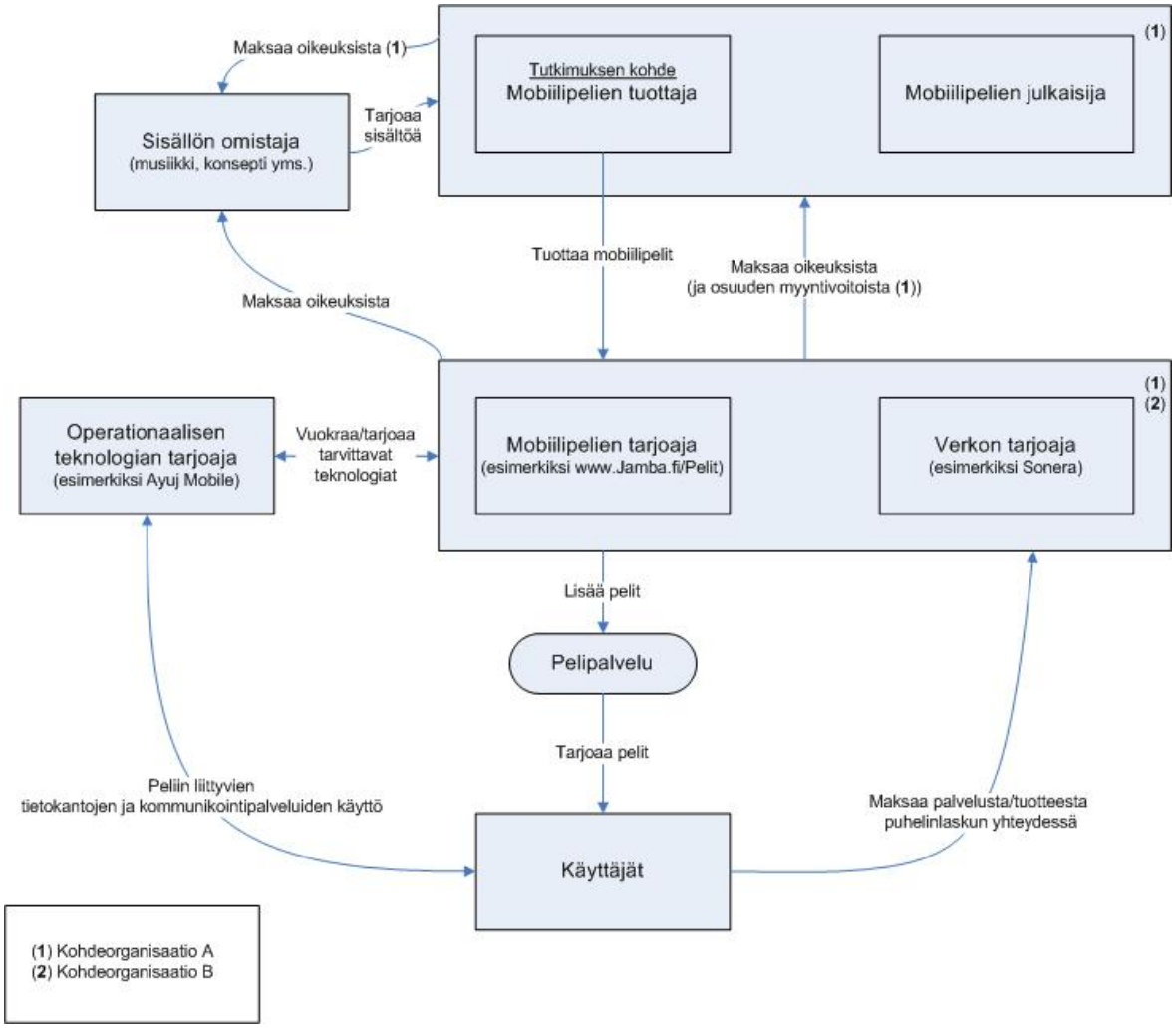
Luvussa käsitellään empiirisen tutkimuksen tulokset, tutkimuskehyksen mukaisessa järjestyksessä. Ensimmäisen osion sisältö käsitellään kohdassa 6.1. Osion avulla verrattiin tieteellisten lähteiden soveltuvuutta mobiilipelituotantoon. Kohdassa 6.2 analysoidaan terminologian yhtenäistämisen vaikutukset empiirisen tutkimuksen vastauksiin, toisin sanoen kohta sisältää empiirisen tutkimuksen toisen osion tulokset. Kolmannen osion sisällön analysointi suoritetaan kohdassa 6.3. Analysointiin sovelletaan kuvion 15 mukaista ratkaisumallia. Osio 1 suoritettiin vain kohdeorganisaatioille A ja B, sillä aikataulu asetti tutkimuksen etenemiselle paineita ja vastaukset olivat tarpeeksi yhtenäisiä. Osion 1 suorittaminen vain kahdelle ensimmäiselle kohdeorganisaatiolle ei ole tutkimuksen onnistumisen kannalta haitallinen tekijä, sillä kyseisen osion tarkoituksena oli varmentaa tieteellisten lähteiden ja tarkkailtavan toimialan kohtaaminen.

6.1 Tutkimusalueen varmentaminen

Tämän kohdan alakohdissa käsitellään empiirisen tutkimuksen ensimmäisen osion vastaukset. Ensimmäinen osio aloitettiin vertaamalla tieteellisten lähteiden pohjalta rakennettua mobiilipeliteollisuuden rakennetta kohdeorganisaatioiden näkemykseen mobiilipeliteollisuuden rakenteesta. Seuraavaksi vuorossa oli tutkielmassa tarkkailtavat tehtäväalueet, tavoitteena selvittää jaetaanko myös kohdeorganisaatioissa kehitys ja ylläpito erikseen toisistaan. Tämän jälkeen tarkennuttiin tutkielman kannalta oleellisimpiin tukiprosesseihin varmentaen, että kohdeorganisaatiossa suoritetaan sekä version- että konfiguraationhallintaa. Alakohtien järjestys noudattaa tätä osion 1 sisältämää järjestystä.

6.1.1 Mobiilipeliteollisuuskentän kuvaaminen

Vastausten perusteella on rakennettu uusi kuvio (kuvio 16) mobiilipeliteollisuudesta. Kuvion avulla tuodaan esille kohdeorganisaatioiden näkemys omasta toimialastaan.



KUVIO 16. Kohdeorganisaatioiden näkemys mobiilipeliteollisuuden rakenteesta

Tieteellisistä lähteistä koostettuun kuvioon mobiilipeliteollisuuden rakenteesta (kuvio 13) ei tullut paljoa muutoksia. Tämä vahvistaa lähteiden vastaavuuden käytännön yritystoimintaan mobiilipeliteollisuudessa. Kohdeorganisaatiot A ja B mainitsivat mobiilipelien tarjoajan ja verkon tarjoajan kuuluvan usein samaan organisaatioon. Näistä mainittiin esimerkkeinä Vodafone (http://www.vodafone.com/) ja Verizon (http://gamesondemand.verizon.net/do/index).

Kohdeorganisaatio A korosti myös mobiilipelien julkaisijan roolia. Toisin sanoen mobiilipeleillä voi olla erikseen julkaisija, jolta mobiilipelien tarjoajat hankkivat pelit. Tällöin A:n

mukaan mobiilipelin tuottaja on useimmiten julkaisijan sisäinen studio tai alihankkija. A nosti esille eroavaisuuden myös rahavirran kulkuun liittyen. A:n mukaan mobiilipelien tarjoaja tilittää myyntivoitot mobiilipelien julkaisijalle, joka hoitaa oikeuksiin (esimerkiksi rojalteihin ja lisenssimaksuihin) liittyvien maksujen suorittamisen sisällön omistajalle. Liiketoiminnasta riippuen eri osa-alueet voivat myös jäädä pois. Esimerkiksi kohde-organisaatio B tuottaa omat mobiilipelinsä yleensä itse konseptista lähtien. Tällöin sisällön omistaja on itse mobiilipelin tuottaja. Mutta myös B myönsi tuottaneesta mobiilipelejä, joissa käytettiin ulkopuolisen toimijan konseptia, jolloin esitetyn kuvion mukainen rakenne on paikkaansa pitävä.

6.1.2 Tutkimuksen alaiset tehtäväalueet

Tehtäväalueiden koostumuksen selvittämiseksi esitettiin kysymys muodossa: ”Erotetaanko mobiilipelituotannossa mobiilipelien kehitys ja ylläpito toisistaan? Jos erotetaan niin millä tasolla, eri henkilöt/tiimit, erilailla määritetyt prosessit?”. Sekä A että B vastasivat myönteisesti kyseiseen kysymykseen. Molempien kohdeorganisaatioiden perustelut olivat samansuuntaisia, käytettävää termistöä myöten.

Vastausten perusteella mobiilipelien kehityksen ja ylläpidon jako on seuraava: Kehitys tähtää ”referenssibuildien” valmistamiseen. ”Referenssibuildeilla” tarkoitetaan valmiita tuotteita, jotka on valmistettu vain muutamalle kohdepuhelimelle. Tämän jälkeen tuote siirtyy jo ylläpidolle (sekä A että B viittasivat tähän termillä jälkituotanto), joka tuottaa mobiilipelin muille käytössä oleville matkapuhelinmalleille. Molemmat kohdeorganisaatiot erottivat ylläpidon puolelta myös mobiilipelien tuottamisen tuleviin matkapuhelinmalleihin ja uusien versioiden toimittamisen asiakkaalle (vrt. Abranin ym. jakelunhallinta (2004, 7 – 9)). B:ssä tämän jakelunhallinnan (sekä pelit että pelien uudet versiot asiakkaille) hoitaa erikseen nimetty Submission-osasto.

Kohdeorganisaatio A:n mukaan ylläpito jatkuu siihen asti, kunnes tuote poistuu ”kaupan hyllyllä”. Tältä osin mobiilipeleillä on selkeä elinkaari, verrattuna perinteiseen ohjelmisto-tuotteeseen. Esimerkiksi Adobe Systemsin Photoshop tuoteperheen aikaisempi malli 7.0 on jo poistunut myynnistä, mutta ylläpitoa kyseiseen versioon hoidetaan yhä päivitysten

muodossa. Lisäksi yritysasiakkaille joudutaan usein säilyttämään Help-desk- ja konsultointitoimintaa tuotekehityksen jo päätyttyä. Aikaisemmin mainittu IEEE standardin (1990, 68) määritelmä ohjelmistotuotteen elinkaaresta sopii siten paremmin perinteiseen ohjelmistotuotantoon kuin mobiilipelituotantoon. Määritelmän mukaan ohjelmistotuotteen elinkaari alkaa siitä, kun ohjelmistotuote saa alkunsa ja loppuu siihen, kun ohjelmistotuotteella ei ole enää mitään käyttöä.

6.1.3 Tutkimuksen alaiset avainprosessit

Tutkimusongelman ollessa ohjelmistotuotannon konfiguraation- ja versionhallinnan ongelmat mobiilipelituotannossa, oli välttämätöntä varmistaa näiden tukiprosessien olemassaolo kohdeorganisaatioissa. Kuten kirjallisuuskatsauksesta selviää, version- ja konfiguraationhallinnan erottaminen toisistaan selkeinä kokonaisuuksina ei ollut yksinkertainen tehtävä. Lisäksi käytännön kokemukset yritysmaailmasta ovat osoittaneet versionhallinta termin usein kattavan myös konfiguraationhallinnan. Tieteellisissä lähteissä tilanne on usein päinvastainen, eli versionhallinta kuuluu useimmin konfiguraationhallinnan alaisuuteen. Näistä johtuen ensimmäisen osion loppuun liitettiin mukaan määritykset version- ja konfiguraationhallinta termeistä.

Vastauksien avulla saatiin varmistettua kohdeorganisaatioiden A ja B suorittavan sekä version- että konfiguraationhallintaa mobiilipelituotannon tukena. Tämän tiedon perusteella ohjelmistotuotannon konfiguraation- ja versionhallinnan ongelmien ja ratkaisujen tarkasteleminen mobiilipelituotannossa oli mahdollista.

6.2 Käsitteiden läpikäynti

Toisen osion tarkoitus oli yhtenäistää vastausten terminologia eri vastaajien kesken. Lisäksi käsitteiden läpikäynnin avulla poistettiin mahdollinen ongelmatilanne, jossa vastaaja ei tiedä mitä kyseisellä termillä tarkoitetaan.

Ensimmäiseen osioon määriteltäviksi termeiksi valittiin konfiguraation- ja versionhallinta (liite 5). Kummallakaan kohdeorganisaatiolla ei ollut ongelmia vastata konfiguraation- ja

versionhallintaa käsittelevään kysymykseen, joten toisessa osiossa päästiin tavoitteeseen esikyselylomakkeen kohdalla.

Kolmannen osion tueksi määriteltiin yhteensä viisi termiä. Termejä olivat revisio, variantti ja ohjelmisto-objekti, sekä muistutukseksi uudestaan konfiguraation- ja versionhallinta (liite 6). Vastausten analysoinnin jälkeen selvisi vastausten olevan yhtenäisiä, joten toisessa osiossa päästiin tavoitteeseen myös kolmannen osion aikana.

Yhteenvetona voisi todeta käsitteiden läpikäynnistä olleen suurta apua empiiriseen tutkimukseen. Vastaukset pysyivät yhtenäisinä kautta linjan, toisin sanoen kohdeorganisaatiot eivät vastanneet samaan kysymykseen eri aiheeseen liittyen ja kysymyksiin ei jätetty vastaamatta, koska kysymystä ei ymmärretty.

6.3 Konfiguraation- ja versionhallinnan ongelmat/ratkaisut

Tämän kohdan alakohdissa käsitellään empiirisen tutkimuksen osion 3 tulokset. Tulosten keräämiseen käytettiin liitteenä (liite 6) olevia viitekehyksiä ja tulosten analysointiin kuvion 15 mukaista ratkaisumallia. Alakohdissa esitellään, yksi kerrallaan, esitetyt kysymykset ja tieteellisiin lähteisiin pohjautuen, kartoitetaan mahdollisia ratkaisumalleja löydettyihin ongelmiin. Alakohdissa edetään viitekehyksien mukaisessa järjestyksessä. Ensin käsitellään konfiguraationhallinta teknisestä näkökulmasta (alakohta 6.3.1), tämän jälkeen siirrytään konfiguraationhallinnan hallinnolliseen näkökulmaan (alakohta 6.3.2) ja viimeiseksi versionhallintaan (alakohta 6.3.3).

Kohdeorganisaatiot A, C, D, E ja F suorittivat osion lomakekyselynä ja kohdeorganisaatio B teemahaastatteluna. Kustakin kohdeorganisaatiosta tutkimukseen osallistui yksi henkilö ja uusintakierroksia ei suoritettu. Koska C on vasta viimeaikoina siirtynyt mobiilipelien tuottamiseen, sovittiin kyseisen organisaation kanssa vastausmahdollisuudesta ”En osaa ottaa kantaa”. Kohdeorganisaation mukaan ottaminen oli perusteltua, sillä vastauksista voidaan hyvin päätellä, minkälainen tilanne on mobiilipelituotantoon vasta siirtyneellä organisaatiolla. Tieteellisistä lähteistä kartoitettiin kuhunkin viitekehykseen kahdeksan kysymystä, eli kysymyksiä oli yhteensä 24. Näistä 21 kappaleen ongelmat osoittautuivat, kohdeorganisaatioiden mukaan, oleelliseksi myös mobiilipelituotannossa. Kyseisiin

ongelmiin on otettu ratkaisumallit tutkielman tieteellisistä lähteistä. Lisäksi alakohdissa on analysoitu kohdeorganisaatioiden tämän hetkisiä ratkaisuja ongelmiin. Kuvion 15 mukaista ratkaisumallia käyttäen, vastaukset voidaan luokitella kolmeen eri luokkaan. Sekä luokittelumalli että vastausten luokan arviointi on tutkijan valmistamia ja suorittamia tätä tutkielmaa varten.

- Kohdeorganisaatioiden mukaan kysymyksen aihepiiriin liittyen ei esiinny ongelmia (tunnuksena tyhjä kenttä).
- Kohdeorganisaatioiden mukaan kysymyksen aihepiiriin liittyen on ongelmia, mutta ongelmiin löytyy oma ratkaisumalli (tunnuksena +).
- Kohdeorganisaatioiden mukaan kysymyksen aihepiiriin liittyen on ongelmia ja ongelmiin ei ole omaa ratkaisumallia (tunnuksena ++).

Analysoinnissa näihin luokkiin suhtaudutaan kuhunkin omalla tavalla. Ensimmäiseen vaihtoehtoon ei tehdä jatkotoimenpiteitä. + -vaihtoehtoon arvioidaan mobiilipelituottajan omaa ratkaisumallia verraten sitä tieteellisten lähteiden tarjoamaan ratkaisumalliin. Arvioinnit suorittaa aina tutkija, ilman kohdeorganisaatioiden konsultointia. ++ -vaihtoehdossa tutkitaan, miten ja minkälainen tieteellisten lähteiden ratkaisumalli soveltuu kyseiseen ongelmaan. Jos kohdeorganisaatioiden vastaukset kuuluvat eri luokkiin, analysoidaan vastaus alimman luokan mukaan. Toisin sanoen jos kohdeorganisaatio A:n vastaus kuuluu luokkaan + ja kohdeorganisaatio B:n vastaus luokkaan ++, niin tällöin vastaus analysoidaan luokan ++ mukaisesti. Luokittelumallin mukaisesti on alla olevaan taulukkoon (taulukko 3) luokiteltu kaikki kysymykset. Koska jokaisessa viitekehyksessä on kahdeksan kysymystä, käytetään kyseisessä taulukossa kysymysten listauksessa numerointia kysymysten aukikirjoittamisen sijaan. Kysymykset löytyvät numeroituina liitteestä 6.

Seuraavissa alakohdissa on käsitelty taulukon jokainen kysymys yksityiskohtaisesti. Alakohdista selviää, minkälaisia ongelmia kohdeorganisaatioilla on esiintynyt sekä kohdeorganisaatioiden väliset näkemyserot konfiguraation- ja versionhallinnan haasteisiin. Yksityiskohtaisempi tarkkailu on tärkeää, sillä kysymyksen luokituksen ollessa ++, ei

tarkoita sitä, ettei jollakin kohdeorganisaatiolla olisi omaa ratkaisumallia kyseiseen ongelmaan. Luokkiin + ja ++ kuuluvat vastaukset laskettiin ongelmien myöntämiseksi kyseisessä kysymyksen aihepiirissä. Kysymyksiä, joiden aihepiiriin liittyen vähintään 5/6 (yli 67 %) kohdeorganisaatioista myönsi ongelmia, oli neljä kappaletta. Kysymyksiä, joiden aihepiiriin liittyen 3–4/6 (50 – 67 %) kohdeorganisaatioista myönsi ongelmia, oli kolme kappaletta. Kysymyksiä, joiden aihepiiriin liittyen 1–2/6 (17 – 49 %) kohdeorganisaatioista myönsi ongelmia, oli 14 kappaletta. Kysymyksiä, joiden aihepiirissä ei esiintynyt ongelmia, oli kolme kappaletta. Kohdeorganisaatiokohtaiset luokitukset ja kysymyskohtaiset tilastot löytyvät jokaisen alakohdan johdannon jälkeisestä taulukosta.

TAULUKKO 3. Konfiguraation- ja versionhallinnan kysymysten luokittelu

Kysymyksen numero kyseisessä viitekehyksessä	**Konfiguraationhallinta Tekninen näkökulma**	**Konfiguraationhallinta Hallinnollinen näkökulma**	**Versionhallinta**
1	++	++	++
2	+	++	++
3	++	++	
4		++	++
5	+	++	++
6	++		++
7	++	++	++
8	++	++	++

6.3.1 Konfiguraationhallinta – tekninen näkökulma

Konfiguraationhallinnan tekniseltä kannalta + tai ++ luokituksen sai seitsemän kysymystä, esitetyistä kahdeksasta. Jokaisesta esitetystä kysymyksestä on analysoitu kohde-

organisaatioiden vastaukset. Tämän jälkeen on määritelty kysymyksen luokitus ja luokituksen vaatiessa käsitelty tieteellisten lähteiden ratkaisumalli. Tekniseltä kannalta kohdeorganisaatioille tarjottiin tiivistetty ratkaisumalli, joka on koostettu tieteellisistä lähteistä. Tämän ratkaisumallin soveltuvuutta mobiilipelituotantoon on arvioitu alakohdan lopussa. Taulukkoon 4 on koostettu luokitukset, kohdeorganisaatioittain, käsitellyistä kysymyksistä. Lisäksi taulukossa on tilastot, joiden avulla kohdeorganisaatioita ja kysymyksiä voi verrata keskenään. ∑+ koostaa kysymyksen/kohdeorganisaation + ja ++ arvot. Ei ongelmaa/ongelma sarake koostaa kohdeorganisaatioiden vastaukset.

Kohdeorganisaatioiden kokemat ongelmat jakautuivat kolmessa viimeisessä kysymyksessä yksittäisiksi tapauksiksi. Suurikokoisten mobiilipelien rakenteen hallinnassa (kysymys 1) kaikki vastanneet kohdeorganisaatiot kokivat ongelmia mobiilipelituotannossa. Yhtenäinen mielipide kohdeorganisaatioilla oli myös työkalujen tarjoaman tuen ja kehitys- ja ylläpito-prosessien laadukkuuden kohdalla (kysymykset 2 ja 4), sillä näitä ei koettu ongelmallisiksi mobiilipelituotannossa. Tosin kohdeorganisaatiot A ja B olivat aikaisemmin kokeneet ongelmia työkaluihin liittyen, mutta kyseisiin ongelmiin on molemmilla tällä hetkellä omat ratkaisumallit. Seuraavaksi käydään läpi yksityiskohtaisesti jokainen taulukossa (taulukko 4) mainittu kysymys numerojärjestyksessä.

1. Alalla on esiintynyt ongelmia suurikokoisten mobiilipelien rakenteen hallinnassa?

Teknisestä näkökulmasta konfiguraationhallinnassa tämä oli ainoa kysymys, jossa kaikkien vastanneiden kohdeorganisaatioiden vastaus kuului luokkaan ++. Suurimmaksi syyksi ongelmiin ilmeni laitteistoskaala. **Laitteistoskaalalla** tarkoitetaan tutkielmassa kohde-laitteiden, eli mobiililaitteiden kirjoa. Kun kyseessä on normaalia suurikokoisempi mobiili-peli ja kyseinen mobiilipeli pitää tehdä moniin eri matkapuhelinmalliin mahdollisimman vähällä koodimäärällä, aiheuttaa laitteistoskaala suuria ongelmia mobiilipelituotannossa. Laitteistoskaala osoittautui syyksi myös muutamalle muullekin ongelmalle. Kohde-organisaatio A mainitsi, että mobiilipelien tuottamisessa on esiintynyt hankaluuksia saada ”otetta” suurikokoisten mobiilipelien tuotannossa. Tosin A mainitsi, ettei ongelma johtunut siitä, että kyseessä on juuri mobiilipeli. C halusi jättää vastaamatta tähän kysymykseen.

TAULUKKO 4. Luokitukset kohdeorganisaatioittain tekniseltä kannalta

Konfiguraationhallinta – tekninen näkökulma							**Tilastot**		
Kysymys \ Kohdeorganisaatio	A	B	C	D	E	F	Ei ongelmaa	Ongelma	∑+
1. Alalla on esiintynyt ongelmia suurikokoisten mobiilipelien rakenteen hallinnassa?	++	++	ei vas-tausta	++	++	++	0	5	10
2. Tukevatko alalla käytössä olevat työkalut konfiguraation-hallinnan prosesseja?	+	+	ei vas-tausta				3	2	2
3. Toimivatko mobiilipelien kehittäjän/ylläpitäjän käyttämät ohjelmistot loogisena kokonaisuutena?			++	+	++	++	2	4	7
4. Onko mobiilipelien kehitys- ja ylläpitoprosessi systemaattinen/jäljitettävä?							6	0	0
5. Ohjelmisto-objektien tunnistamisessa, tallentamisessa ja käyttämisessä esiintyy ongelmia?		+					5	1	1
6. On huomattu ongelmia mobiilipelin tietyn konfiguraation jälkikäteen koostamisessa?						++	5	1	2
7. Mobiilipelien tuottamisessa on ilmennyt ongelmia tehtyjen/tehtävien muutosten vaikutuksien ymmärtämisessä?					++	++	4	2	4
8. Onko monimutkaisten mobiilipelien tuottamisessa esiintynyt ongelmia ryhmä-/yksilötyössä?	ei vas-tausta	+			++		3	2	3
∑+	3	5	2	3	8	8			29

Kysymyksen luokitus on ++. Ongelman aiheuttaa kaksi yhtäaikaisesti esiintyvää haastetta. Laitteistoskaalan takia konfiguraatioiden määrä kasvaa suureksi, lisäksi tuotannossa ollessa suurikokoinen mobiilipeli, kasvaa myös yksittäisen konfiguraation koko. Vain toisen edellä mainitun haasteen esiintyessä ei kohdeorganisaatioiden mukaan esiinny ongelmaa, mutta molempien haasteiden yhtäaikainen esiintyminen aiheuttaa suuriakin ongelmia mobiilipelituotantoon. Toisenkin haasteen minimoiminen voi siten selkeyttää mobiilipelituotantoa kyseisen ongelman kohdalla. Laitteistoskaala tuskin pienenee lähitulevaisuudessa ja useista laitevalmistajista johtuen laitteiden standardointikin on epätodennäköistä, ainakin riittävissä määrin. Koska laitteistoskaalaan mobiilipelituottajat eivät voi vaikuttaa, ratkaisu löytyy konfiguraationhallinnan tehostamisesta suurikokoisten konfiguraatioiden hallinnan osalta. Tieteellisten lähteiden perusteella suurikokoisten konfiguraatioiden hallinnassa isoimmat ongelmat ovat muutosten koordinoitu suorittaminen ja kehitettävän tuotteen pitäminen, koko elinkaaren ajan, määritellyssä tilassa (Babich 1986 mukaan, Conradi & Westfechtel 1998, 233; Estublier ym. 2005, 387). Tekniseltä kannalta ajateltuna, käytettäviä järjestelmiä pitäisi kehittää vastaamaan tehokkaammin Dartin (1991, 7) esittämistä kategorioista eritoten ohjelmisto-objekteihin, rakenteeseen ja prosessiin. Kyseisiä kategorioita tehostamalla saadaan tuki mobiilipelituottajalle ohjelmisto-objektien tunnistamiseen, käyttämiseen ja tallentamiseen sekä rakenteen hallintaan.

2. Tukevatko alalla käytössä olevat työkalut konfiguraationhallinnan prosesseja?

Kohdeorganisaatiot D, E ja F eivät kokeneet ongelmia tällä alueella. A ja B totesivat ongelmien olemassaolon, mutta molemmilla on ainakin osittain itse kehitetyt järjestelmät konfiguraationhallinnan prosessien tueksi. Toimialaa koskeva ongelma tähän liittyy, sillä molempien kohdeorganisaatioiden mukaan valmiita mobiilipelituotantoon sopivia työkaluja ei ole tällä hetkellä tarjolla. Toisin sanoen alalla aloittava organisaatio joutuu joko kehittämään omat järjestelmät tai muokkaamaan kaupallisen järjestelmän sopivaksi mobiilipelituotannon prosesseihin. C halusi jättää vastaamatta tähän kysymykseen.

Kysymyksen luokitus on +. Nykyiset konfiguraationhallintajärjestelmät ovat muun muassa käyttäjälle näkymättömiä ja riippumattomia ohjelmointikielistä (Introna & Edgar 1997, 35; Estublier ym. 2005, 391). Vaikka konfiguraationhallintajärjestelmistä on pyritty tekemään

sopeutuvaisia mihin tahansa ohjelmointikieleen, ei kyseiset konfiguraationhallinta-järjestelmät kohdeorganisaatioiden A ja B mukaan tällä hetkellä sovellu mobiilipeli-tuotantoon ilman erityistä räätälöintiä. Toisin sanoen konfiguraationhallintajärjestelmiä pitäisi laajentaa vastaamaan myös mobiilipelituotannon haasteisiin. Toinen vaihtoehto on tuottaa erityisesti mobiilipelituotantoon soveltuva konfiguraationhallintajärjestelmä. Tällä hetkellä mobiilipelituotannossa yritykset joutuvat sijoittamaan paljon resursseja omien järjestelmien kehittämiseen, kun vaihtoehtona markkinarakoa löytyisi konfiguraationhallintajärjestelmälle, joka on kehitetty mobiilipelituotantoon. Kohde-organisaatio D:n mukaan eri konfiguraatioiden rakentamiseen löytyy hyvin toimiva työkalu (J2MEPolish), mutta esimerkiksi dokumentaationhallinnan haasteisiin kyseisestä työkalusta ei ole apua.

3. Toimivatko mobiilipelien kehittäjän/ylläpitäjän käyttämät ohjelmistot loogisena kokonaisuutena?

Sekä A:n että B:n mukaan omat järjestelmät on kehitetty tarpeeksi hyvin, että ne toimivat loogisena kokonaisuutena läpi mobiilipelituotannon elinkaaren. Muut organisaatiot eivät olleet samaa mieltä. C:n mukaan kehitystyötä joudutaan tekemään useilla eri ohjelmistoilla ja heillä ei ainakaan ole vielä saumatonta integraatiota ohjelmistojen välillä. F:n mukaan varsinkin dokumentaationhallinnan puolelle tarjolla olevat järjestelmät eivät sovellu konfiguraationhallinnassa. Käytettävissä ohjelmistoissa korostuu selkeästi mobiilipeli-tuotantoon siirtyvän organisaation tilanne. Suoranaisesti mobiilipelituotantoon sopivia järjestelmiä ei ole ja oman järjestelmän räätälöiminen vaatii resursseja ja kokemusta mobiilipelituotannosta.

Kysymyksen luokitus on ++. Tieteellisten lähteiden ratkaisu liittyy kysymyksessä 2 esitettyyn ratkaisuun. Estublierin ym. (2005, 392) mukaan konfiguraationhallinta-järjestelmä antaa yhtenäisen työpöydän, jonka avulla voidaan tehdä tarvittavat toimenpiteet käyttäen joko konfiguraationhallintajärjestelmän tai ulkoisen työkalun ominaisuuksia. Koska tällä hetkellä mobiilipelituotantoon ei ole tarjolla täysin yhteensopivia konfiguraationhallintajärjestelmiä, ei mobiilipelituotannossa saada aikaan yllä mainittua Estublierin ym. (2005) määrittelemää järjestelmätukea ilman omien järjestelmien kehittä-

mistä tai kaupallisten järjestelmien räätälöintiä. Kohdeorganisaatio D on löytänyt useista valmiista työkaluista tarpeeksi hyvin toimivan kokonaisuuden, esimerkiksi mobiilipelien kehityksen puolella käytetään Estublierin ym. (2005) mainitsemia konfiguraationhallinta-järjestelmiä.

4. Onko mobiilipelien kehitys- ja ylläpitoprosessi systemaattinen/jäljitettävä?

Kaikki kohdeorganisaatiot vastasivat kysymykseen myönteisesti. Kohdeorganisaatio A:n vastaus perustelee hyvin, miksi kehitys- ja ylläpitoprosessit ovat systemaattisia ja jäljitettäviä: ”Voin tässä puhua vain omista prosesseistamme, jotka ovat hyvinkin systemaattisia. Olen kyllä siinä uskossa, että sama pätee kaikkiin firmoihin. Koska pelejä on yhtaikaa työn alla useita, uusia laitteita tulee jatkuvasti jne., niin prosessin on oltava ”tiukka”.” Mobiilipelien kehitys- ja ylläpitoprosessien ollessa systemaattisia ja jäljitettäviä, ei kysymykseen 4 liittyen rakenneta ratkaisumallia.

5. Ohjelmisto-objektien tunnistamisessa, tallentamisessa ja käyttämisessä esiintyy ongelmia?

Kohdeorganisaatio B myönsi ongelmia esiintyneen ohjelmisto-objekteihin liittyen, mutta kyseisiin ongelmiin reagoitiin järjestelmää muokkaamalla. Täten myös ohjelmisto-objektien käyttöön liittyviin ongelmiin ratkaisuksi osoittautuivat itse kehitetyt järjestelmät. Koska kohdeorganisaatioilla on omat järjestelmät, niin ne pystyvät välttämään ja tarvittaessa itse korjaamaan esiin tulevat ongelmat. Muiden kohdeorganisaatioiden mukaan ongelmia tällä alueella ei ole esiintynyt, vaikka käytössä ei olekaan itse kehitettyjä järjestelmiä. Ongelmattomuus tällä alueella saattaa johtua siitä, että kohdeorganisaatioilla ei ole vielä ollut monia suurikokoisia mobiilipeliprojekteja yhtäaikaisesti käynnissä. Tällöin organisaation sisäiset resurssit riittävät keskittymään intensiivisesti käynnissä oleviin projekteihin ja ongelmat eivät siten välttämättä ole tullut vielä esille.

Kysymyksen luokitus on +. Selkeästi nykyinen tapa ratkaista ohjelmisto-objektien käyttöön liittyvät ongelmat mobiilipelituotannossa on valmistaa tai muokata itse järjestelmä, joka kykenee hoitamaan tarvittavat toimenpiteet. Tästä syystä kokeneilla toimijoilla ohjelmisto-objektien käytössä ei esiinny ongelmia, mutta ratkaisumalli itsessään estää toimialan

kokonaisvaltaisen kehittymisen ja standardien leviämisen. Tieteellisissä lähteissä on tähän selkeä ratkaisu. Mobiilipelituotanto tarvitsee valmiita järjestelmiä, jotka vastaavat mobiilipelituotannon asettamiin haasteisiin teknillisesti Dartin (1991, 7) määrittelemien kategorioiden ja hallinnollisesti Abranin ym. (2004, 7 – 3) ja Estublierin ym. (2005, 392) määrittelemien aihe-alueiden osalta (kuvio 7).

6. On huomattu ongelmia mobiilipelin tietyn konfiguraation jälkikäteen koostamisessa?

Useimmat kohdeorganisaatiot eivät todenneet ongelmia konfiguraatioiden jälkikäteen koostamisessa. Kohdeorganisaatio B:ssä itse kehitetty järjestelmä kykenee suoriutumaan tarvittaessa vanhan konfiguraation rakentamisesta automaattisesti. Kohdeorganisaatio F:n mukaan konfiguraation rakentaminen jälkikäteen vaatii joskus suuriakin muutoksia lähdekoodiin.

Kysymyksen luokitus on ++. Kohdeorganisaatio F:n käyttämät järjestelmät eivät vastauksen mukaan kykene hoitamaan Dartin (1991, 7) määrittämää kategoriaa Rakentaminen. Konfiguraationhallintajärjestelmän avulla pitäisi kyetä rakentamaan ohjelmistotuote järjestelmän versioimista lähdetiedostoista. Lisäksi ohjelmisto-objektien vanhojen versioiden käyttö on oltava mahdollinen. F:n ongelma ilmenee nimenomaan vanhojen konfiguraatioiden rakentamisessa. Ratkaisu tähän on joko sekä konfiguraationhallintatietokannan että versiokirjaston rakentaminen tai pelkän versiokirjaston rakentaminen, jonka tueksi järjestelmään integroidaan konfiguraationhallintaa tukevia prosesseja. Konfiguraationhallintatietokanta on samantapainen kirjasto kuin versiokirjasto, erona vain se, että sitä käytetään konfiguraatioiden tallentamiseen. Usein versiokirjastoa kuitenkin käytetään sekä versioiden että konfiguraatioiden tallentamiseen, eikä erillistä konfiguraationhallintatietokantaa luoda. (Abran ym. 2004, 7 – 3; ITIL 2005, 124.)

7. Mobiilipelien tuottamisessa on ilmennyt ongelmia tehtyjen/tehtävien muutosten vaikutuksien ymmärtämisessä?

A:n mukaan tällaista tilannetta ei saisi missään tapauksessa olla, mutta laitteistoskaalan takia muutoksien vaikutuksia ei voida täysin ennakoida. Tämä johtuu A:n mukaan siitä,

että jokainen kohdelaite saattaa käyttäytyä eri tavalla tehtävään muutokseen. A, B, C ja D eivät koe ongelmia tehtyjen/tehtävien muutosten vaikutuksien ymmärtämisessä. Muun muassa B tutkii muutosten vaikutukset, ennen kuin nämä muutokset tehdään.

Kysymyksen luokitus on ++. Mobiilipelituotannossa yksittäisen muutoksen suunnitteleminen yhteen kohdelaitteeseen ei aiheuta ongelmia, vaan kokonaisvaltaisen arvioinnin suorittaminen suurelle laitteistoskaalalle on erittäin haasteellista. Ratkaisumallissa voidaan täten soveltaa Hallinta-kategoriaa (Dart 1991, 7). Kategorian määrityksen mukaan muutokset täytyy pystyä monistamaan hallitusti useisiin eri kohdelaitteisiin. Jos ongelmia ilmenee tällä alueella, tulisi kohdeorganisaation harkita muita vaihtoehtoisia konfiguraationhallintajärjestelmiä, jotka tarjoavat paremmat edellytykset muutosten vaikutuksien hallintaan.

8. Onko monimutkaisten mobiilipelien tuottamisessa esiintynyt ongelmia ryhmä-/yksilötyössä?

Tähän kysymykseen A jätti vastaamatta, tämä oikeus kohdeorganisaatioille annettiin. B ja E näkivät ongelmien olemassaolon myös mobiilipelituotannon puolella. Tarkennettuna ongelmat esiintyivät B:n mukaan tekstienhallinnassa, sillä tekstien käännöksissä jouduttiin käyttämään Microsoft Exceliä, joka ei taivu tarpeeksi hyvin versionhallinnan tarpeisiin. Kyseessä siis mobiilipelin tuottaminen monelle eri kielelle. Nykyään nämä ongelmat on kierretty B:ssä oman järjestelmän avulla. C:n organisaatiossa ongelmia ei ole ainakaan vielä esiintynyt, johtuen C:n kohdemarkkinoista, mutta kansainvälisille markkinoille siirryttäessä kohtaa C todennäköisesti B:n kuvaamia ongelmia mobiilipelien tuottamisessa. D ja F eivät, C:n tavoin, kokeneet ongelmia monimutkaisten mobiilipelien tuottamisessa ryhmä- tai yksilötyössä.

Kysymyksen luokitus on ++. Tieteellisissä lähteissä on käsitelty monimutkaisten ohjelmistotuotteiden tuottamiseen liittyviä ongelmia sekä Estublierin ym. (2005) että Korelin ym. (1991) artikkeleissa. Konfiguraationhallintajärjestelmillä haetaan ratkaisua muun muassa lähdetiedostoihin pääsynhallintaan, tallentamiseen, tunnistamiseen, lähde- ja kohdetiedostojen eri versioiden noutamiseen ja kohdetiedostojen uudelleen rakentamiseen, kun lähdetiedostot muuttuvat (Korel ym. 1991, 161). Lisäksi konfiguraationhallinta-

järjestelmä huolehtii siitä, kenellä on uusin versio jostakin tietystä ohjelmisto-objektista tai miten muokattavana ollut ohjelmisto-objekti palautetaan takaisin järjestelmään työn jälkeen (Estublier ym. 2005, 392). Mobiilipelituottajan tulisi huomioida edellä mainitut vaatimukset konfiguraationhallintajärjestelmältä organisaation tehdessä ostopäätöstä valmiista järjestelmästä, tai suunnitellessa räätälöivänsä itse oman järjestelmän, sillä konfiguraationhallintajärjestelmän tarjotessa tekniset ratkaisut kyseisiin vaatimuksiin, ei mobiilipelituottajalta mene turhaan arvokkaita työtunteja näiden tehtävien manuaaliseen hoitamiseen.

Teknisen näkökulman kysymysten jälkeen, kohdeorganisaatioille esiteltiin tieteellisten lähteiden tarjoama ratkaisumalli kysymyksissä käsiteltyihin ongelmiin. Ratkaisumalli on liitteessä 6, konfiguraationhallinnan teknisen näkökulman viitekehyksen lopussa. Tosin kohdeorganisaatioiden ajan säästämiseksi tekniseltä kannalta tarjottiin vain yksi tiivistetty ratkaisumalli, joka käsitteli koko konfiguraationhallinnan teknistä näkökulmaa. Esittämällä valmis ratkaisumalli, tarjottiin kohdeorganisaatioille mahdollisuus kommentoida alustavasti jo haastattelutilanteessa ratkaisumalleja. B:n mukaan kyseinen ratkaisumalli on hyvin lähellä sitä, minkälaisia tavoitteita asetettiin organisaation omalle järjestelmälle sitä suunniteltaessa. B:n toteuttaman ratkaisumallin avulla esimerkiksi artistien suorittamat muutokset (esimerkiksi uusien grafiikoiden lisäämiset) eivät vaikuta ohjelmoijien työskentelyyn. Tämäntyyppinen ratkaisu on suora toteutus tarjotun ratkaisumallin jälkimmäiseen osuuteen, jossa mainitaan muun muassa hajautetun työskentelyn ongelmia ohjelmisto-objektien käytön kannalta (liite 6).

6.3.2 Konfiguraationhallinta – hallinnollinen näkökulma

Seuraavaksi on vuorossa hallinnollisesta näkökulmasta konfiguraationhallinnan kahdeksan kysymystä. Kysymyksistä ++ luokituksen sai seitsemän kappaletta. Kohdeorganisaatioiden mukaan konfiguraatioista saadaan tarpeeksi tietoa ja konfiguraatiot ovat tarpeeksi määritellyssä tilassa, täten kysymys 6:n mukainen ongelma ei esiintynyt mobiilipelituotannossa. Alla olevaan taulukkoon (taulukko 5) on koostettu luokitukset kohdeorganisaatioittain hallinnollisen näkökulman kysymyksistä.

TAULUKKO 5. Luokitukset kohdeorganisaatioittain hallinnolliselta kannalta

Konfiguraationhallinta – hallinnollinen näkökulma							**Tilastot**		
Kysymys \ Kohdeorganisaatio	A	B	C	D	E	F	Ei ongelmaa	Ongelma	$\sum^{+}$
1. Alalla on esiintynyt ongelmia konfiguraatioiden tunnistamisessa tai ohjelmisto-objektien välisten suhteiden tunnistamisessa?		++				++	4	2	4
2. Onko mobiilipelien kehittäjillä/ylläpitäjillä konfiguraationhallinnan tarkkuustaso selvillä/yhtenäinen?	++	++	++	++		++	1	5	10
3. Onko mobiilipelien tuottaja-organisaatioissa esiintynyt ongelmia vastuualueissa, resursseissa, aikatauluissa, työkaluissa yms.?	++	++	++	++	++	++	0	6	12
4. Ovatko mobiilipeleihin tehdyt muutokset kontrolloituja?				++	++		4	2	4
5. Useat yhtäaikaiset mobiilipelien kehitys- ja ylläpito-projektit aiheuttavat ongelmia?	++		++				4	2	4
6. Esiintyykö mobiilipelien tuotannossa jokin seuraavista ongelmista? (kysymyksen tarkennus käsittelyn yhteydessä 7)							6	0	0
7. Ongelmia saavuttaa suunniteltu/määritelty mobiilipeli?	++	++		++	++		2	4	8
8. Ongelmia aikaisemmin määritetyn konfiguraation rakentamisessa (hallinnollinen näkökulma)?			++	++			4	2	4
$\sum+$	8	8	8	10	6	6			46

Kohdeorganisaatiokohtaiset profiilit eivät ole hallinnolliselta kannalta konfiguraationhallinnassa niin yhtenäiset kuin tekniseltä kannalta. Vastausprofiileja tarkasteltaessa kohdeorganisaatiot voidaan jakaa kahteen joukkoon, sillä kunkin joukon sisällä organisaatioiden kokemat ongelmat olivat keskenään yhtenäiset. Toiseen joukkoon kuuluvat organisaatiot B ja F ja toiseen organisaatiot A, C, D ja E. Hallinnollisen näkökulman käsittely erosi teknisestä ratkaisumallien osalta, sillä hallinnollisen näkökulman jokaiseen kysymykseen oli liitetty oma tiivistetty ratkaisumalli tieteellisistä lähteistä. Seuraavaksi käydään läpi yksityiskohtaisesti jokainen taulukossa 5 mainittu kysymys numerojärjestyksessä.

1. Alalla on esiintynyt ongelmia konfiguraatioiden tunnistamisessa tai ohjelmisto-objektien välisten suhteiden tunnistamisessa?

Kohdeorganisaatiot B ja F kokivat kysymyksen 1 sisältämät ongelmat oleelliseksi mobiilipelituotannossa. B tunnisti ongelmien ilmenevän joissain tapauksissa, tosin käytännön esimerkkiä haastateltavalle ei tullut mieleen haastattelun yhteydessä. Ongelmat ovat kuitenkin B:n mukaan lähinnä tietoliikenneteknisiä ongelmia.

Kysymyksen luokitus on ++. Koska empiirisen tutkimuksen aikana koettuihin ongelmiin ei saatu konkreettista esimerkkiä, käsitellään tässä yhteydessä vain tieteellisten lähteiden tarjoama ratkaisumalli. Konfiguraationhallinnassa on vakavia puutteita jos konfiguraatioiden tai ohjelmisto-objektien välisten suhteiden tunnistamisessa esiintyy ongelmia, sillä Abranin ym. (2004, 7 – (6–7)) mukaan konfiguraatioiden tunnistamisen alaiset prosessit tarjoavat perustan konfiguraationhallinnan toimivuudelle. Ensimmäinen askel on konfiguraationhallinnan alaisten ohjelmisto-objektien määrittäminen. Konfiguraationhallinnan alaisuuteen määritetään joko kaikki tai vain tarpeelliset ohjelmisto-objektit (Abran ym. 2004, 7 – 6; IEEE 1983 & IEEE 1988 mukaan, Conradi & Westfechtel 1998, 233). B:ssä konfiguraationhallintaan liittyvät määritelmät on tehty lokaalisti jokaiseen työpisteeseen ja näitä määritelmiä voidaan päivittää aina normaalien järjestelmäpäivitysten yhteydessä. Abranin ym. (2004, 7 – 6) ja Dartin (1991, 7) mukaan konfiguraationhallinnan täytyy taata konfiguraatioiden, näiden versioiden ja ohjelmisto-objektien välisten suhteiden tunnistamisen. Lokaalien määritysten avulla kyetään takaamaan nämä vaatimukset, mutta organi-

saation kokonaisvaltainen hyöty konfiguraatioista jää vajaaksi. Konfiguraationhallintatietokannan avulla koko organisaatio kykenee hyötymään määritetyistä konfiguraatioista tarpeen vaatiessa. Toki konfiguraationhallintatietokannan määrittämiseen ja rakentamiseen menee resursseja, mutta pidemmän päälle tietokanta voi tarjota oleellista historiatietoa entisten projektien konfiguraatioista yms.

2. Onko mobiilipelien kehittäjillä/ylläpitäjillä konfiguraationhallinnan tarkkuustaso selvillä/yhtenäinen?

Kohdeorganisaatioiden mukaan mobiilipelituotannossa konfiguraationhallinnan tarkkuustaso on selvillä, mutta yhtenäinen se ei ole. Sekä A että B korostivat eri operaattorien sekä julkaisijoiden vaatimusten vaikeuttavan mobiilipelien tuottamista. Erilaiset spesifikaatiot pakottavat luomaan joissakin tapauksissa täysin uuden projektin eri julkaisijoiden/operaattoreiden kohdalla. A:ta lainaten ”Pelistudio X voi tehdä alihankintana peliprojektin julkaisijalle Y, joka voi haluta vain yhden ns. referenssibuildin pelistä. Julkaisija Z taas voi vaatia sata.”. Kyseessä on siis ongelma, joka on lähtöisin mobiilipelituotannon ulkopuolelta, mutta vaikuttaa vahvasti mobiilipelituottajiin. B toteaakin, että organisaation sisäisen konfiguraationhallinnan tarkkuustaso varmistetaan koulutuksen ja tiedottamisen avulla. E oli ainoa kohdeorganisaatio, joka ei kokenut ongelmia konfiguraationhallinnan tarkkuustasoon liittyen.

Kysymyksen luokitus on ++. Esiintyneeseen ongelmaan rakennettu ratkaisumalli kuuluu prosessinhallinnan alaisuuteen, joka on yksi Abranin ym. (2004) määrittämistä aihealueista. Ongelma aiheutuu mobiilipelin tuottajan ja tilaajan välisestä suhteesta. Abranin ym. (2004, 7 – 4) ja ITIL:n (2005, 233) mukaan prosessinhallintaan kuuluu myös alihankkijoiden konfiguraationhallinnan kartoittaminen, joten ratkaisumallia voidaan soveltaa myös tämäntyyppiseen ongelmaan, ottaen ratkaisumallin soveltamisessa organisaatioiden rajat huomioon. Ratkaisumallin mukaan ensiksi on määriteltävä organisatorinen sisältö. Toisin sanoen mobiilipelin tuottajan ja tilaajan välisestä rajapinnasta pitäisi selvittää organisatoriset elementit ja niiden väliset suhteet, sillä konfiguraationhallinta on vuorovaikutuksessa useiden muiden avainprosessien kanssa. Määritysten avulla voidaan asettaa organisaatioille yhtenäiset konfiguraationhallinnan rajoitteet ja ohjeistukset. Tuotoksena

määrityksistä toimii konfiguraationhallinnan suunnitteludokumentti. Ratkaisumallin toimivuutta ja noudattamista tulee seurata aktiivisesti ja sitä kautta hankkia pitkän tähtäimen tietoa prosessin toimivuudesta. Ratkaisumallia oikeaoppisesti soveltaen mobiilipelituotannossa voitaisiin ainakin yhtenäistää tilaajien vaatimuksia ja siten helpottaa tämän hetkistä ongelmaa. (Abran ym. 2004, 7 – (2–5).)

3. Onko mobiilipelien tuottajaorganisaatioissa esiintynyt ongelmia vastuualueissa, resursseissa, aikatauluissa, työkaluissa yms.?

Kysymyksen aihepiireihin liittyvät ongelmat olivat erittäin oleellisia mobiilipelituotannossa, sillä kaikki kohdeorganisaatiot tunnistivat ongelmia aikatauluihin tai työkaluihin liittyen. Sekä A että B totesivat aikataulut suurimmaksi ongelmaksi. Ongelma on lähtöisin lisenssinhaltijan tai tilaajan asettamista tiukoista aikarajoista. B:n mukaan aikataulut ovat usein niin tarkasti määrättyjä, että mobiilipeleistä joudutaan useimmiten ennemminkin tiputtamaan joitakin osia pois, kuin joustamaan aikatauluissa. Lisäongelman aikatauluihin tuo se, että selkeää pelikonseptia ei ole usein olemassa, kun mobiilipelin kehitys aloitetaan. Pelikonseptiin kohdistuvat muutokset vaikuttavat usein aikatauluihin ja tätä kautta tulee B:n mukaan suurimmat ongelmat aikataulutuksiin. C korosti ongelman lähteeksi työkalut, sillä työkalut ja niiden käyttö ei ole täysin yhteismitallista ja määrämuotoista. Varsinkin alihankkijoita käytettäessä työkaluihin liittyvä kompleksisuus lisääntyy.

Kysymyksen luokitus on ++. Aikatauluihin liittyvä ongelma kuuluu prosessituen alaisuuteen ja työkaluihin liittyvä ongelma tuotetuen alaisuuteen (Estublier ym. 2005, 392–393; Bersoff ym. 1980 mukaan, Conradi & Westfectel 1998, 233). B:llä on aikataulutuksen apuna niin sanottu road-map, johon on koostettu organisaation tulevia projekteja loppuvuodelle. Aikataulut ovat kuitenkin niin tiukat, että kyseinen ”road-map” auttaa projektien aikatauluttamisessa, mutta ei tee aikatauluista joustavia. Alun perin konfiguraationhallintajärjestelmät oli suunniteltu vain muutostenhallintaan, mutta nykyään konfiguraationhallintajärjestelmät toimivat hyvin pitkälti myös kehitys- ja ylläpitoprosessien tukena. Tämä ei tarkoita sitä, että organisaatioiden tulisi hankkia uudet ja kalliit järjestelmät, vaan esimerkiksi CVS integroituna vanhempaan konfiguraationhallinta-

järjestelmään tarjoaa paljon hyödyllisiä ominaisuuksia skriptien muodossa nimenomaan prosessien tukemiseksi (Estublier ym. 2005, 393). Konfiguraationhallinnan suunnitteleminen sisältää organisaation omien työkalujen valinnan, alihankkijoiden konfiguraationhallinnan kartoittamisen (johon kuuluu käytänteiden lisäksi myös käytettävät työkalut) ja kausaalisuudenhallinnan (Abran ym. 2004, 7 – 4; ITIL 2005, 233). C:n ongelmat omien työkalujen yhtenäisyyden kanssa heijastuvat epäilemättä myös tämän kysymyksen aihepiireihin. Lisäksi on hyvin mahdollista, että käytettävillä alihankkijoillakin voi olla ongelmia omien työkalujen yhtenäisyyden kanssa. Kun tässä tilanteessa olevien kahden organisaation pitää toimia yhteistyössä, on konfiguraationhallinnan suunnitteleminen edellä mainittujen Abranin ym. ja ITIL:n mukaisten osa-alueiden osalta erityisen tärkeätä.

4. Ovatko mobiilipeleihin tehdyt muutokset kontrolloituja?

Kohdeorganisaatiot D ja E kokivat ongelmia muutosten kontrolloimisessa. B:n mukaan muutoksia lähdetään tekemään todella varovaisesti ja muutosten vaikutus muualle pyritään laskemaan tarkasti. A:n mukaan näiden muutosten vaikutus muualle kyetään arvioimaan kohtuullisen tarkasti, tosin ei aivan 100 % varmuudella.

Kysymyksen luokitus on ++. Muutospyyntöjen arvioinnin tuloksena muutosten vaikutukset pitäisi kyetä ennakoimaan ja kustannukset arvioimaan tehokkaasti – kyseiset tekijät vaikuttavat osaltaan mobiilipelituotannon tehokkuuteen (Abran ym. 2004, 7 – 7; ITIL 2005, 126). D:n tulisi keskittyä varsinkin edellä mainittuun muutoksenhallinnan prosessiin, sillä ongelma on hyvin todennäköisesti hallinnollinen D:n kieltäessä ongelmat teknisen näkökulman kysymyksessä 7 (ongelmia tehtyjen/tehtävien muutosten vaikutuksien ymmärtämisessä). E:llä ongelmat voivat olla lähtöisin myös tekniseltä näkökulmalta, sillä E myönsi kohdanneensa ongelmia muutosten arvioimisessa laajaan laitteistoskaalaan.

5. Useat yhtäaikaiset mobiilipelien kehitys- ja ylläpitoprojektit aiheuttavat ongelmia?

Kohdeorganisaatio A:n mukaan ongelmat aiheutuvat lähinnä tiimin koosta ja lisenssinhaltijan hyväksytyksistä useisiin yhtäaikaisiin projekteihin. B:n mukaan heillä on tarpeeksi hyvin määritelty organisaatiorakenne, jonka avulla tiedetään mitä kyetään tekemään rinnakkain, aiheuttamatta suuria ongelmia mobiilipelituotantoon. C:llä useat yhtäaikaiset

kehitys- ja ylläpitoprojektit aiheuttavat ongelmia resurssipuolella, joka on C:n mukaan täysin organisaation sisäinen haaste. D, E ja F eivät kokeneet useiden yhtäaikaisten mobiilipelien kehitys- ja ylläpitoprojektien aiheuttavan ongelmia.

Kysymyksen luokitus on ++. Muutoksenhallinnan sisältämillä ratkaisuilla pitäisi kyetä poistamaan A:n ja C:n kokemat ongelmat. Lisenssinhaltijoiden hyväksytykset ovat usein projektia hidastava tekijä. Lisenssinhaltijoille pitäisi kuitenkin muistaa tehdä selväksi organisaation käytänteet muutospyyntöjenhallinnan, muutospyyntöjen arvioinnin ja muutospyyntöjen hyväksynnän osilta. Tämän avulla saadaan lisenssihaltija ymmärtämään, että projekti ei välttämättä etene kaikilta osin ennen hyväksyntää ja lisenssinhaltija voi siten ehkä nostaa omien käytänteidensä tasoa muutostenhallintaan liittyen. Lisäksi mobiilipelien tuottajilla tulisi olla tehokkaasti toimivat käytänteet sovituista muutoksista tai kehityskohteista poikkeamiseen ja luopumiseen. Kun mobiilipelituottajalla on organisaatiossa tarkasti määritetyt keinot muutoksiin reagoimiseen, kyetään mobiilipelien tuottaminen pitämään tehokkaana, vaikka projektiin kohdistuisi ennakoimattomia muutoksia. (Abran ym. 2004, 7 – (7–8); ITIL 2005, 126.)

6. Esiintyykö mobiilipelien tuotannossa jokin seuraavista ongelmista:

    - konfiguraatioista ei saada tarpeeksi tietoa
    - konfiguraatiot eivät ole määritellyssä tilassa
    - konfiguraationhallinta ei ole tehokasta?

Kohdeorganisaatioiden mukaan konfiguraatioista saadaan tarpeeksi tietoa ja konfiguraatiot ovat tarpeeksi määritellyssä tilassa, tosin B mainitsi, että konfiguraationhallinta voisi olla tehokkaampaa. Kysymys 6 oli ainoa, jonka aihepiiriin liittyviä ongelmia yksikään kohdeorganisaatio ei myöntänyt esiintyvän mobiilipelituotannossa hallinnolliselta kannalta, joten tähän kysymykseen ei rakenneta ratkaisumallia.

7. Ongelmia saavuttaa suunniteltu/määritelty mobiilipeli?

Suunnitellun/määritellyn mobiilipelin saavuttamisessa ongelmia kokivat kaikki muut kohdeorganisaatiot paitsi C ja F. C:n mukaan heidän organisaationsa keskittyy projekteissa suurimmaksi osaksi pelien tuottamiseen, joten ongelmia kysymyksen aihepiiriin liittyen aiheudu. A:n mukaan ongelma ei ole yksistään mobiilipelien ongelma, vaan kokonaisuutena peleihin liittyvä ongelma. Pelien tekeminen ei ole A:n mukaan niin suoraviivaista, kuin voisi kuvitella. B yhtyy tähän mielipiteeseen toteamalla pelien tekemisen olevan välillä todella luovaa työtä. B:n mukaan suunnitellun/määritellyn mobiilipelin saavuttamisen suurin ongelma on, jo aikaisemminkin mainittu, mobiilipelin määrittelyn puuttuminen kehitysprojektin alkaessa. Tämä korostuu varsinkin silloin, kun mobiilipeli tehdään puhtaalta pöydältä. Vastakohtana lisensoidun pelin tekeminen on B:n mukaan huomattavasti helpompaa, koska silloin tiedetään heti projektin alkaessa, minkälainen mobiilipeli on tarkoitus tuottaa.

Kysymyksen luokitus on ++. Hallinnollisesta näkökulmasta konfiguraationhallintaan kuuluu myös laadunvalvonta. Laadunvalvonnan avulla varmistetaan, että ohjelmistotuotteet ja ohjelmistotuotantoprosessit (toisin sanoen mobiilipelit ja mobiilipelituotantoprosessit) ovat yhdenmukaisia määriteltyjen sääntöjen, standardien, rajoitusten, ohjeistuksien ja suunnitelmien kanssa (IEEE 1997, 25). Abranin ym. (2004, 7 – 9) tarjoaman ratkaisumallin mukaan laadunvalvonta tulisi jakaa kahteen osaan, fyysiseen ja funktionaaliseen konfiguraationhallintaan. Funktionaalisen konfiguraationhallinnan avulla pyritään varmistamaan tuotteen ja määritelmien yhdenmukaisuus, joten muiden paitsi C:n tulisi tehostaa organisaationsa laadunvalvonnan funktionaalista konfiguraationhallintaa. C:n tulisi keskittyä organisaatiossaan enemmän fyysiseen konfiguraationhallintaan, jonka tarkoituksena on valvoa ohjelmisto-objektien ja spesifikaatioiden yhdenmukaisuutta. Projekteissa, joissa on tiukka aikataulu, ei ole välttämättä aikaa keskittyä kaikkiin hallinnollisiin osa-alueisiin. Täten joudutaan valitsemaan sopiva keskitie, miten paljon resursseja sijoitetaan projektin funktionaaliseen konfiguraationhallintaan ja miten paljon fyysiseen konfiguraationhallintaan. Pääasia on, että kehityksen alainen tuote läpäisee sekä fyysiset että funktionaaliset vaatimukset hyväksyttävällä tasolla. (Abran 2004, 7 – 9.)

8. Ongelmia aikaisemmin määritetyn konfiguraation rakentamisessa (hallinnollinen näkökulma)?

Hallinnollisesta näkökulmastakaan kohdeorganisaatiot A, B ja E eivät myöntäneet aikaisemmin määritetyn konfiguraation rakentamisessa ongelmia mobiilipelituotannossa. Teknisestä näkökulmasta (konfiguraationhallinnan tekninen näkökulma kysymys 6) C:llä ja D:llä ei ollut ongelmaa konfiguraatioiden rakentamisessa, mutta hallinnolliselta kannalta ongelmia esiintyy. Syynä C:n mukaan on liiallinen keskittyminen pelien tuottamiseen, konfiguraationhallinnan sijaan. F:llä on ongelmia konfiguraatioiden rakentamiseen liittyen sekä tekniseltä että hallinnolliselta kannalta.

Kysymyksen luokitus on ++. Jakelunhallinnan peruspilarina toimiva konfiguraatioiden rakentaminen on organisaatiolle tärkeä prosessi, varsinkin toimitettaessa useita erilaisia konfiguraatioita tilaajille. Prosessilla rakentaminen ei tarkoiteta koodaamista, vaan konfiguraatiotietojen perusteella oikeanlaisen version rakentamista yhdistämällä ohjelmisto-objektit kokonaisuudeksi. Hallinnolliselta kannalta rakentamiseen kuuluu rakentamisohjeiden luominen ja valvonta, sekä rakentamisprosessin toimivuuden valvonta (esimerkiksi pystytään rakentamaan ohjelmistotuotteesta aikaisemmin julkaistu konfiguraatio täysin identtisenä kokonaisuutena). (Abran ym. 2004, 7 – 9.) Prosessi on tärkeä myös ylläpidon puolella, B mainitsee esimerkkinä tilanteen, jossa asiakkaalla on toimivuusongelmia mobiilipelin kanssa. Virheilmoituksen mukana saadaan informaatio asiakkaan käyttämästä laitteistosta ja mobiilipelin versiosta. Näiden tietojen avulla kyetään rakentamaan identtinen konfiguraatio mobiilipelistä ja tutkia mistä virhe on lähtöisin. Ongelmat konfiguraatioiden rakentamisessa voivat olla lähtöisin esimerkiksi konfiguraationhallintatietokannasta tai versiokirjastosta, tarkemmin näiden käyttämiseen määritellyistä prosesseista tai määriteltyjen prosessien noudattamisesta.

6.3.3 Versionhallinta

Versionhallinnan viitekehys käsitellään yhtenä kokonaisuutena. Kysymyksiä otettiin mukaan tutkimukseen kahdeksan kappaletta, joista ++ luokituksen sai seitsemän kappaletta. Täten versionhallinnassa ongelmia esiintyi yhtä monessa kysymyksen aihepiirissä

kuin konfiguraationhallinnan hallinnolliselta ja tekniseltä kannalta. Ainoastaan revisioiden aikaulottuvuuden seurannassa kohdeorganisaatioilla ei ollut ongelmia (kysymys 3). Alla olevaan taulukkoon (taulukko 6) on koostettu luokitukset versionhallinnan kysymyksiin.

Versionhallinnan ongelmat näyttäisivät olevan hyvin yrityskohtaisia, sillä kartoitettujen ongelmien esiintymät ovat hyvin hajanaiset yllä olevassa taulukossa. Toisaalta ongelmien esiintymättömyys on varsin yhtenäinen ja noudattaa suurin piirtein samaa kaavaa, lukuun ottamatta kohdeorganisaatio D:tä, jolla on vastauksien mukaan hyvin erilaiset ongelmat versionhallintaan liittyen. Seuraavaksi käydään läpi yksityiskohtaisesti jokainen taulukossa 6 mainittu kysymys numerojärjestyksessä.

1. Onko alalla keinoja pitää monista versioista (revisioista ja varianteista) koostuva mobiilipeli/ohjelmisto-objekti hyvin organisoituna?

Erityisesti variantit aiheuttavat ongelmia mobiilipelituotannossa. B:n mukaan projektinhallinta on suhteellisen helppoa siihen asti, kunnes joudutaan tekemään pieni variaatio jostain tietystä konfiguraatiosta. Useimmiten variantit johtuvat nimenomaan tilaajien vaatimuksista, koska useilla tilaajilla on eri spesifikaatiot mobiilipeleihin liittyen. Lainaten B:tä ”Mobiilipelituotanto menee niin kauan hyvin, kun vain päivitetään entisiä versioita ja tehdään uusia revisioita tuotteesta. Mutta sitten, kun joudutaan tekemään rinnakkain muutoksia, niin silloin se ei ole enää organisatorista ja usein eri variantit joudutaan hajottamaan eri projekteiksi”. A, C, D, E ja F kohdeorganisaatiot eivät ole kokeneet ongelmia revisioiden ja variantтien hallinnassa, tosin C totesi ”Meillä tilanne ei ole vielä ongelmallinen – tosin se saattaa sellaiseksi muodostua ilman riittävää suunnitelmallisuutta”. Tästä voi päätellä C:n kokeneen versionhallintaan liittyviä haasteita, tosin ei vielä ongelmallisella tasolla.

TAULUKKO 6. Luokitukset kohdeorganisaatioittain versionhallinnassa

	Versionhallinta						**Tilastot**		
Kysymys \ Kohdeorganisaatio	A	B	C	D	E	F	Ei ongelmaa	Ongelma	$\sum^{+}$
1. Onko alalla keinoja pitää monista versioista (revisioista ja varianteista) koostuva mobiilipeli/ohjelmisto-objekti hyvin organisoituna?		++					5	1	2
2. Mobiilipeleistä/ohjelmisto-objekteista kehittyy usein liian monta eri revisiota tai varianttia?		++		++		++	3	3	6
3. Onko esiintynyt ongelmia revisioiden aikaulottuvuuden kanssa?							6	0	0
4. Käytetäänkö päivitysten kokoluokissa mitään rajauksia?	++	++	++		++	++	1	5	10
5. Mobiilipelituotannossa on ongelmia versionhallinnassa ja versioiden säilytyksessä, kun versioiden määrä kasvaa?				++			5	1	2
6. Ovatko käytössä olevat versiokirjastot sekavia tai heikosti suunniteltuja?				++		++	4	2	4
7. Versioiden tarvitseman työmäärän seuraamisessa esiintyy ongelmia tai se ei ole mahdollista?				++	++		4	2	4
8. Mobiilipelien tuottajien (kehittäjien ja ylläpitäjien) suorittama versiointi on satunnaista, tehden version-hallinnan haastavammaksi?				++		++	4	2	4
$\sum+$	2	6	2	10	4	8			32

Kysymyksen luokitus on ++. Tutkitut tieteelliset lähteet eivät tarjoa ratkaisua variantteihin liittyviin ongelmiin, kun varianttien määrä kasvaa suureksi. ITIL:ssä (2005, 239) koros-

tetaan varianttien aiheuttamia ongelmia ohjelmistotuotantoon ja kyseisen lähteen mukaan ratkaisu on yksinkertaisesti varianttien määrän pitäminen mahdollisimman pienenä. Hallinnollista ongelmaa voidaan ehkäistä muun muassa revisiopuiden avulla, jolloin organisaatiolla säilyy monimutkaisissakin projekteissa hyvä kokonaiskuva revisioista ja varianteista. Revisiopuuhun voidaan lisätä tarvittaessa ominaisuuksia, jotka auttavat versioiden välisten suhteiden esittämisessä. Tällaisia ominaisuuksia voivat olla esimerkiksi riippuvuus ja yhteensopivuus, perinteisen pohjautuvuuden lisäksi (Sachweh & Schafer 1995, 25). Tosin Reichenbergerin (1989) mukaan myös revisiopuiden toimivuus kärsii, kun varianttien määrä kasvaa suureksi.

2. Mobiilipeleistä/ohjelmisto-objekteista kehittyy usein liian monta eri revisiota tai varianttia?

Ongelmalliseksi revisioiden ja varianttien liiallisen muodostumisen tunnistivat kohdeorganisaatiot B, D ja F. Ainoastaan E ei nähnyt revisioiden ja varianttien muodostumisessa ongelmaa, sillä myös A:n mukaan ongelma on olemassa. Tosin A näkee ongelman olevan hallittavissa, jos projektinhallintaan vain kiinnittää tarpeeksi huomiota. B toteaa ongelman esiintyvän enimmäkseen suurien hittipelien kohdalla, jolloin joudutaan tekemään normaalia enemmän variantteja mobiilipelistä. C ennakoi tilanteen muuttuvan omalla kohdallaan huonompaan suuntaan, kun organisaation jakelukanava kansainvälistyy.

Kysymyksen luokitus on ++. Ratkaisumalli tähän ongelmaan on samantapainen kuin versionhallinnan kysymyksessä 1. Ongelmien lähtökohtaisena erona on se, että kysymys 1:n ratkaisumalli käsitteli varianttien hallintaa ja tämä liiallista varianttien muodostumista. Esimerkiksi käytössä olevat revisiopuut eivät pysy hallinnassa, jos varianttien määrä kasvaa liian suureksi (Conradi & Westfechtel 1998). Lisäksi ITIL:n (2005, 239) mukaan versionhallinta säilyy huomattavasti luotettavampana, kun varianttien määrä kyetään pitämään pienenä. Ratkaisuksi mainitaan varianttien määrän kontrolloiminen. Mobiilipelituottajan pitäisi siis pyrkiä yhtenäistämään mobiilipelien tilaajien spesifikaatioita, jotta mobiilipeliprojektit kyettäisiin suorittamaan yhden projektin alaisena ja lopulta irrottamaan tästä projektista tarvittavat variantit ja revisiot eri tilaajille.

3. Onko esiintynyt ongelmia revisioiden aikaulottuvuuden kanssa?

Revisioiden hallinnassa kohdeorganisaatiot eivät kokeneet ongelmia, vaikkakin revisioiden aikaulottuvuuden seurannassa saattaa olla tehostettavia piirteitä. Lisäinvestointien jälkeen revisioiden hallinta voi kehittää organisatorista ymmärrystä ja sitä kautta tehostaa mobiilipelien tuottamista. Koska kohdeorganisaatiot eivät kokeneet ongelmia revisioiden aikaulottuvuuteen liittyen, ei tähän kysymykseen rakenneta ratkaisumallia.

4. Käytetäänkö päivitysten kokoluokissa mitään rajauksia?

Mobiilipeleissä ja peleihin liittyvissä ohjelmisto-objekteissa päivitysten kokoluokkien rajauksia käyttää vain D. Tämä ei tietenkään tarkoita, ettei muilla kohdeorganisaatioilla olisi revisionumerointia käytössä, mutta kokoluokkien rajauksia ei ole otettu käyttöön. Ellei mobiilipeliteollisuus kokonaisuudessaan kykene standardoimaan laitteistoja ja spesifikaatioita, aiheutuu tästä todennäköisesti ongelmia mobiilipelituotantoon. B totesi käyttävänsä organisaation omissa työkaluissa ITIL:n (2005, 205) mukaista revisioluokkien määrittelyä, mutta mobiilipeleihin liittyvää versionumeroinnin ohjeistusta ei ole yleisesti käytössä.

Kysymyksen luokitus on ++. Revisionumeroinnin käyttö edistää suuren versiomäärän hallintaa ja vähentää variantteihin liittyviä hallinnollisia ongelmia. Tärkeä revisio syrjäyttää edelliset revisiot ja aloittaa revisionumeroinnin pienten revisioiden kohdalta alusta. Esimerkiksi revisiot 2.0 ja 3.0 ovat tärkeitä revisioita ja revisiot 2.1 ja 3.1 ovat pieniä revisioita. Lisäksi on olemassa kolmas revisioluokka; pikakorjaus. Kyseisen luokan revisiot sisältävät usein vain pieniä korjauksia. Esimerkiksi revisiot 2.2.1 ja 2.2.2 kuuluvat pikakorjauksiin. (ITIL 2005, 205–207.) Revisionumeroinnin käytöllä saadaan esille myös paljon tärkeää tietoa kyseisistä versioista. Esimerkiksi tärkeät revisiot voivat tarkoittaa taaksepäin yhteensopimattomuutta. Lisäksi revisionumerointia voidaan käyttää hyväksi monille projekteille yhteisten ohjelmisto-objektien määrittämisessä, kuten tekoälyyn liittyvissä ohjelmisto-objekteissa.

5. Mobiilipelituotannossa on ongelmia versionhallinnassa ja versioiden säilytyksessä, kun versioiden määrä kasvaa?

Versioiden säilytyksessä koki ongelmia vain D, täten suurin osa kohdeorganisaatioista ei kokenut erillisen versiokirjaston tarkkaa suunnittelua oleelliseksi. Edullisten tallennusmedioiden vuoksi ongelmat eivät ole peräisin tallennustilan rajallisuudesta, vaan versiokirjastojen pitämisestä organisoituina.

Kysymyksen luokitus on ++. Versiokirjaston toiminnan takaa hyvin määritelty versiointimalli. Versiointimallin avulla jokaisesta tallennetusta versiosta saadaan nopeasti esille tarvittava informaatio, koskien esimerkiksi version kokeman muutoksen tyypin tunnistamista (Chapin ym. 2001). Lisäksi versiointimalli määrittää ohjelmisto-objektit, jotka kuuluvat versioinnin alaisuuteen, tunnistamistavan, rakenteen, operaatiot sille miten olemassa olevat versiot noudetaan ja uusien versioiden rakentamistavan (Conradi & Westfechtel 1998, 233). Tieteellisten lähteiden ratkaisumalli on käsitelty alakohdassa 3.3.2, ratkaisumalli perustuu suurimmalta osin Conradin ja Westfechtelin (1998) sekä Chapinin ym. (2001) artikkeleihin.

6. Ovatko käytössä olevat versiokirjastot sekavia tai heikosti suunniteltuja?

Kohdeorganisaatiot D ja F kokivat ongelmia versiokirjastojen toimivuudessa. D:llä versiokirjastoihin liittyvät ongelmat voivat olla lähtöisin myös versiointimalliin liittyvistä ongelmista (edellinen kysymys). Vaikka C ei myöntänyt ongelmaa tällä alueella, oli vastaukseen liitetty perustelu ”Kaikki on suhteellista. Pärjäämme toimimalla itse luomallamme mallilla”. Yleisesti ottaen versiokirjastojen toimivuuteen oltiin kohdeorganisaatioissa tyytyväisiä.

Kysymyksen luokitus on ++. Versiokirjaston toimivuus on erittäin tärkeää mille tahansa ohjelmistotuottajalle, sillä versiokirjasto on versionhallinnan pääasiallinen työkalu (IEEE 1990, 68). Projektien laajuudesta ja monimutkaisuudesta riippuen versiokirjastoja voidaan määrittää useisiin eri käyttötarkoituksiin. Erityyppisiä versiokirjastoja ovat muun muassa työskentely-, tuki- ja tuotekirjasto (Abran ym. 2004, 7 – 7; ITIL 2005, 203). Alakohtaan

3.3.4 on koottu listaksi versiokirjaston suunnittelun tärkeimmät huomioitavat kokonaisuudet.

7. Versioiden tarvitseman työmäärän seuraamisessa esiintyy ongelmia tai se ei ole mahdollista?

A, B, C ja F eivät kokeneet ongelmia versioiden tarvitseman työmäärän seuraamisessa, joten D ja E olivat ainoat ongelmat myöntäneet kohdeorganisaatiot. Versiohistoriaan kerättävien tietojen määrittäminen ja esittäminen, toimivalla tavalla, voi osoittautua ongelmaksi varsinkin ohjelmistotuotteissa, joissa versioiden määrä on suuri ja elinkaari pitkä.

Kysymyksen luokitus on ++. Versiohistoriasta on apua tulevien projektien suunnittelemisessa ja mobiilipelituotannon prosessien epäkohtien havaitsemisessa. Versiohistorian avulla yksittäisestä mobiilipelin tai ohjelmisto-objektin versiosta pitäisi saada koottua tietoa esimerkiksi kyseisen version kokemien muutosten määrästä (Eick ym. 2002). Tähän tietoon pohjautuen voidaan tarkkailla mobiilipelituotannon tehokkuutta. Lisäksi versiohistoriasta on apua mobiilipelin elinkaaren seurantaan. Eickin ym. (2002, 402–404) artikkeliin on koostettu kuviot, joista näkee ohjelmistotuotteen kokemien muutosten määrän versioittain sekä vuosittain. Koska mobiilipeleillä on yleensä lyhyempi elinkaari kuin Eickin ym. (2002) esimerkissä, voidaan tarkasteluväliä muokata vuoden sijasta kuukauden mittaiseksi. Tällaisen historiatiedon avulla on mahdollista ennakoida tulevien mobiilipeliprojektien vaatimaa työmäärää ja siten organisaation sisäisten resurssien käyttö tehostuu.

8. Mobiilipelien tuottajien (kehittäjien ja ylläpitäjien) suorittama versiointi on satunnaista, tehden versionhallinnan haastavammaksi?

Mobiilipelien tuottajien suorittama versiointi ei ole satunnaista mobiilipelituotannossa, lukuun ottamatta D ja F organisaatioita. Vaikka yhdessäkään kohdeorganisaatiossa ei revisioluokkia ollut määritelty, on muiden vastaajien mukaan versiointimalli tarpeeksi toimiva.

Kysymyksen luokitus on ++. Ensimmäinen askel versioinnin yhtenäistämisessä on versioitavien kohteiden määrittäminen. Versioinnin alaisuuteen voidaan määrittää kuuluvaksi kaikki ohjelmisto-objektit, tällöin kyseessä on alueriippumaton versiointimalli (Tichy 1985 mukaan, Conradi & Westfechtel 1998, 235). Vaihtoehtoisesti jos versioitavaksi halutaan vain tietyn tyyppiset ohjelmisto-objektit, on käyttöön otettava aluemääritelty versiointimalli (Conradi & Westfechtel 1998, 235). Usein valinta näistä kahdesta vaihtoehdosta perustuu siihen, kumpi aiheuttaa enemmän työtä organisaatiolle. Jos aluemääritellyn versiointimallin rakentaminen on työläs prosessi ja lopputuloksena saataisiin oletettavasti vain pieni joukko versioinnin ulkopuolelle jääviä ohjelmisto-objekteja, voidaan käyttöön ottaa alueriippumaton versiointimalli. Valinta perustuu B:n mukaan siihen, että ylimääräisten ohjelmisto-objektien versioinnin tarvitsema työmäärä on pitkälläkin tähtäimellä huomattavasti pienempi kuin aluemääritellyn versiointimallin rakentaminen. Kun versioinnin alaiset kohteet ovat määritetty, täytyy organisaation tutkia omien versiointikäytänteiden yhtenäistämistä. Tämä voidaan hoitaa esimerkiksi kouluttamisen tai tiedottamisen avulla. Conradin ja Westfechtelin (1998, 233) mukaan versionhallinnan täytyy tarkasti ja luotettavasti tallentaa versioidun ohjelmistotuotteen rakenne, kun ohjelmistotuote kehittyy moniksi eri revisioiksi ja varianteiksi. Satunnaisen versioinnin takia yllämainittu luotettavuuden vaatimus ei versionhallinnalta toteudu. Täten versionhallinnan avulla haettu tuki mobiilipelituotannolle ei välttämättä ole halutulla tasolla.

6.4 Yhteenveto

Empiirisessä tutkimuksessa kohteena olivat mobiilipelituotannon konfiguraation- ja versionhallinnan keskeisimmät ongelmat. Yhteenvetoon on koostettu pääpiirteittäin empiirisen tutkimuksen eteneminen ja oleellisimmat löydökset. Lisäksi yhteenvedossa käsitellään empiirisen tutkimuksen avulla löydetyt ongelmien aiheuttajat mobiilipelituotannossa, liittyen konfiguraation- ja versionhallintaan ja ratkaisumallien toteutus.

Tutkimusalueen varmentaminen suoritettiin vain kahdelle ensimmäiselle kohdeorganisaatiolle. Kyseisen osion toistamista ei koettu oleelliseksi muille, sillä vastaukset olivat tarpeeksi yhtenäisiä sekä keskenään että tieteellisten lähteiden kanssa, muutamaa

poikkeusta lukuun ottamatta. Mainitut poikkeukset koskivat organisaatioita, jotka toimivat useammalla kuin yhdellä tieteellisten lähteiden mainitsemalla kentällä yhtä aikaa. Esimerkiksi mobiilipelien tarjoaja ja verkon tarjoaja voi koostua samasta organisaatiosta. Vastausten perusteella mobiilipelituotannossa kehitys ja ylläpito on kyetty erottamaan toisistaan selkeiksi kokonaisuuksiksi. Kehityksen aikana keskitytään mobiilipelin kehitykseen valmistamalla vain muutama ”referenssibuildi” ja ylläpidossa rakennetaan valmiiseen mobiilipeliin pohjautuen versiot lopuille tarvittaville kohdelaitteistoille. Jotta ongelmien ja ratkaisumallien tarkasteleminen mobiilipelituotannossa oli mahdollista, täytyi tutkimusalueen varmentamisen avulla varmistaa, että mobiilipelituotannossa suoritetaan konfiguraation- ja versionhallintaprosesseja. Molemmat vastanneet kohdeorganisaatiot myönsivät toteuttavansa tieteellisten lähteiden määrittelyn mukaista konfiguraation- ja versionhallintaa omassa organisaatiossaan, joten empiirisen tutkimuksen jatkaminen oli mahdollista.

Käsitteiden suunniteltu läpikäyminen takasi empiirisen tutkimuksen vastauksien yhtenäisyyden vastaajien kesken. Sekä tutkimusalueen varmentamisesta (Osio 1) että konfiguraation- ja versionhallinnan ongelmista/ratkaisuista (Osio 3) saatiin selkeät vastaukset, eikä kysymyksien ymmärtämisessä ollut ongelmia. Täten käsitteiden läpikäynti (Osio 2) vaikutti positiivisella tavalla empiirisen tutkimuksen laatuun. Empiirisestä tutkimuksesta saatiin esille myös mobiilipelituotannon käyttämiä termejä, joita ei tieteellisissä lähteissä käytetty kuvaamaan samaa sisältöä. Esimerkiksi jakelunhallinnasta vastuu oli organisaatio B:ssä Submission-osastolla ja yleisesti alalla käytetty termi ”referenssibuildi” tarkoitti valittuun kohdelaitteeseen rakennettua konfiguraatiota mobiilipelistä.

Osio 3, eli konfiguraation- ja versionhallinnan ongelmat/ratkaisut oli tarkoitettu tutkimusongelmiin liittyvän tiedon keräämiseen. Tutkimuksen tuloksista voidaan päätellä tieteellisten lähteiden, koskien ohjelmistotuotantoa, soveltuvan myös mobiilipelituotantoon. Vastauksia varten rakennettiin erityinen luokittelu, jotta analysoinnissa saataisiin säilytettyä korkealaatuinen taso alusta loppuun. Konfiguraation- ja versionhallinnan keskeisimmät ongelmat jaettiin kolmeen viitekehykseen. Viitekehyksiä olivat konfiguraationhallinta teknisestä näkökulmasta, konfiguraationhallinta hallinnollisesta näkökulmasta ja versionhallinta. Jokaisessa viitekehyksessä oli yksi kysymys, jonka

sisältämä aihepiiri ei ollut yhdenkään kohdeorganisaation mukaan ongelmallinen mobiilipelituotannossa, mutta tunnistetuissa ongelmissa oli suuriakin eroavaisuuksia kysymysten tilastoja tarkasteltaessa. Konfiguraationhallintaan hallinnollisesta näkökulmasta liittyi eniten ongelmia, sillä ++ luokituksen vastauksia tältä kannalta oli yhteensä 23 kappaletta, verraten konfiguraationhallinnan teknisen näkökulman 12 kappaleeseen ja versionhallinnan 16 kappaleeseen. Vastausprofiileja analysoitaessa kohdeorganisaatioilla A, C ja E oli hyvin samantapaiset vastaukset, oli sitten kyseessä konfiguraationhallinta hallinnolliselta kannalta tai versionhallinta. Tekniseltä kannalta konfiguraationhallinnan vastausprofiilit kuitenkin kyseisillä organisaatioilla olivat hyvin erilaiset, pääryhmän muodostuessa organisaatioista A, B, ja D.

Tekniseltä kannalta kohdelaitteiden skaala ja tarjolla olevat konfiguraationhallinta-järjestelmät osoittautuivat suurimmiksi ongelmien lähteiksi. Kohdelaitteiden suuresta määrästä juontui useita ongelmia, sillä laitteistovalmistajia on paljon ja jokaisella on monta erilaista laitteistomallia. Näistä jokainen laite saattaa aiheuttaa yksilöllisiä vaatimuksia mobiilipelituotantoon. Mobiilipeliä tuottaessa tärkeä ominaisuus on koodimäärän pitäminen alhaisena. Kun tähän ongelmaan lisätään edellä mainittu laitteistoskaalan aiheuttama ongelma, syntyy konfiguraationhallinnan teknisille ratkaisuille, varsinkin rakenteen hallintaan liittyen, suuria haasteita. Laitteistoskaalan aiheuttamiin ongelmiin rakennetut ratkaisumallit koostuivat pääosin ohjelmisto-objekti, rakenne, prosessi ja hallinta -kategorioista (Dart 1991). Ratkaisumallit keskittyivät tähän ongelmaan liittyen lähinnä omien menetelmien ja toimintatapojen kehittämiseen.

Toinen selvitetty ongelmien lähde, konfiguraationhallinnan tekniseltä kannalta, oli tarjolla olevat valmiit konfiguraationhallintajärjestelmät. Kohdeorganisaatioiden mukaan valmiit konfiguraationhallintajärjestelmät eivät suoraan sovellu mobiilipelituotantoon. Tällä hetkellä mobiilipelituotantoa suoritetaan useilla eri työkaluilla, jotka eivät ainakaan ilman huomattavaa panostusta, toimi loogisena kokonaisuutena. Monet mobiilipelituottajat ovatkin joutuneet joko kehittämään omia konfiguraationhallinta- ja versionhallinta-järjestelmiä tai integroimaan valmiisiin järjestelmiin uusia ominaisuuksia, jotta käytettävät työkalut saataisiin paremmin toimimaan kokonaisuutena ja vastaamaan mobiilipelituotannon asettamiin haasteisiin. Tarjolla olevissa työkaluissa suurimmat puutteet olivat

dokumentaationhallintaan liittyvissä ominaisuuksissa. Mobiilipelituotantoon tarjolla olevien konfiguraationhallintajärjestelmien puutteisiin rakennetut ratkaisumallit käsittelivät osa-alueita, joita mobiilipelituottajan tulisi tehostaa konfiguraationhallintajärjestelmissään. Pääteemana ratkaisumalleissa oli konfiguraationhallintajärjestelmien tarjoama järjestelmätuki, jotta käytössä olevat työkalut saataisiin toimimaan yhtenäisenä ja selkeänä kokonaisuutena (Estublier ym. 2005).

Hallinnollisesta näkökulmasta suurimmat ongelmien aiheuttajat olivat spesifikaatioerot tilaajien välillä, työkalut, aikataulut ja laitteistoskaala. Spesifikaatioerojen takia mobiilipelituotannossa varianttien määrä kasvaa liian suureksi ja sitä kautta projektin hallinta muuttuu monimutkaisemmaksi. Mobiilipelituotannossa eri tilaajille voidaan joutua jopa tekemään omat projektit samaan tuotteeseen liittyen spesifikaatioerojen takia. Vakiintumattomat ja epäyhtenäiset työkalukokoonpanot vaikeuttivat sekä mobiilipelituottajan ja tilaajan että mobiilipelituottajan ja alihankkijoiden välistä yhteistoimintaa. Useiden kohdeorganisaatioiden mukaan yhteistoiminta alihankkijoiden ja tilaajien kanssa lisää mobiilipelituotannon kompleksisuutta liian paljon. Ratkaisumallit näihin ongelmiin käsittelivät pääosin tilaajien konfiguraationhallinnan kartoittamista (Abran ym. 2004, 7 – 4; ITIL 2005, 233). Kartoittamisen avulla organisaatioiden välinen toiminta voisi tehostua konfiguraationhallinnan yhtenäistyttyä.

Tiukat aikataulut ovat myös yleinen ongelma ohjelmistotuotannossa. Mobiilipelituotannon ollessa luovaa työtä, aiheuttavat tiukat aikataulut suuria ongelmia varsinkin projektin suunnittelemiseen. Lisäksi puutteita esiintyi määritelmissä, joiden avulla pyritään takaamaan suunniteltu ja määritelty mobiilipeli. Suurimmaksi osaksi puutteet esiintyivät fyysisessä konfiguraationhallinnassa, mutta myös funktionaalisessa konfiguraationhallinnassa oli ongelmia. Aikatauluihin liittyvien ongelmien ratkaisumalleissa keskityttiin konfiguraationhallintajärjestelmiin, sillä nykyisten konfiguraationhallintajärjestelmien pitäisi tukea ohjelmistotuotantoprosessin (mobiilipelituotantoprosessin) etenemistä ja sitä kautta ehkäistä aikatauluihin liittyviä ongelmia (Estublier ym. 2005).

Laitteistoskaala aiheutti hallinnolliselta kannalta konfiguraationhallintaan ongelmia varsinkin muutostenhallintaan. Muutosten arviointi ja ennakointi on työlästä johtuen

erilailla käyttäytyvistä kohdelaitteista, tästä johtuen myös resurssien käytön suunnittelu osoittautui hankalaksi. Ratkaisumalleissa muun muassa analysoitiin kohdeorganisaatioiden konfiguraationhallintaa ja tätä kautta selvitettiin minkä alueen heikkoudesta laitteistoskaalan ongelmat johtuivat. D:llä ongelmat olivat hallinnollisia ja E:llä todennäköisesti teknisiä.

Versionhallinnassa ei esiintynyt niin montaa ulkopuolista ongelman aiheuttajaa kuin konfiguraationhallinnassa. Suurimpana ulkopuolisena häiriötekijänä olivat spesifikaatio-erot, jotka aiheuttivat ongelmia varianttien määrän kasvamisen kautta. Ongelmat johtuvat siitä, että versionhallinnan tukena käytettävien graafisten apuvälineiden tarjoama hyöty heikkenee varianttien määrän kasvaessa, sillä tällöin esimerkiksi revisiopuut laajenevat hallitsemattomaksi. Ratkaisumalleissa keskityttiin revisiopuiden ominaisuuksiin, sillä näiden avulla revisiopuista voitaisiin saada toimivia, vaikka varianttien määrä olisikin suuri (Sachweh & Schafer 1995).

Enimmäkseen versionhallinnassa ongelmia aiheuttivat puutteet versionhallinnan toiminnallisissa ominaisuuksissa. Esimerkiksi versiointimallin, versiohistorian ja revisioluokituksen käytössä sekä määrityksissä esiintyi puutteita. Osassa organisaatioista versiohistoriaan ei kiinnitetty tarpeeksi huomiota ja tämän takia versiohistoriasta ei saatu kyseisissä organisaatioissa oleellisia historiatietoja esille aikaisemmista mobiilipeleistä tai ohjelmisto-objekteista. Versiohistorian ongelmiin liittyvässä ratkaisumallissa sovellettiin Eickin ym. (2002) ohjeistusta suurikokoisen ohjelmistotuotteen elinkaaren seurantaan.

Mobiilipelituotannossa terminologiaan liittyvän ongelman voi nähdä siinä, että pelintekijät eivät aina ole samoja henkilöitä kuin ne, jotka ylläpitävät peliä. Useilla mobiilipeli-tuottajilla on ylläpito siirretty halvemman tuotannon maihin. Versionhallintaa hankaloittivat myös versiointimallin toiminnalliset ongelmat sekä versioinnin satunnaisuus. Versiointimalliin liittyen, suurin puute mobiilipelituotannossa oli revisioluokituksen puuttuminen. Ainoastaan yksi kohdeorganisaatio totesi käyttävänsä tarkkaa revisio-luokitusta mobiilipelien versionhallinnassa, lisäksi toinen kohdeorganisaatio käytti kyseistä revisioluokitusta työkalujen versionhallinnassa. Näihin ongelmiin tehdyissä ratkaisu-

malleissa keskityttiin versiointimallin ominaisuuksiin, sillä useilla kohdeorganisaatioilla ilmeni puutteita nimenomaan versiointimalleihin liittyen.

7 YHTEENVETO JA JOHTOPÄÄTÖKSET

Yhteenveto ja johtopäätökset aloitetaan kertaamalla tutkielman tavoite ja tutkimusongelmat, kohdassa 7.1. Tämän jälkeen käsitellään yhteenvetona, kohdassa 7.2, tutkielman kirjallisuuskatsaus (luvut 2–4). Kohdassa keskitytään kirjallisuuskatsauksen toimivuuteen empiirisen tutkimuksen tukena. Kohdassa 7.3 on listattu perusteluin, empiirisen tutkimuksen tuloksista havaitut keskeisimmät ongelmat mobiilipelituotannon konfiguraation- ja versionhallintaan liittyen. Ongelmien kertaamisen jälkeisessä kohdassa esitetään tiivistetysti mobiilipelituotantoon tarjotut ratkaisumallit. Tämän jälkeen käsitellään tutkielman keskeiset tulokset ja empiirisen tutkimuksen luotettavuus omina kohtinaan. Viimeisessä kohdassa käsitellään jatkotutkimusaiheita. Kohdassa arvioidaan muun muassa, miten kirjallisuuskatsauksesta rakennettua kokonaisuutta voisi mahdollisesti hyödyntää muiden kohdealueiden, kuin mobiilipelituotannon tutkimiseen.

7.1 Tutkimusongelma ja tutkimusalue

Tavoitteena oli tutkia, esiintyykö mobiilipelituotannossa samankaltaisia ongelmia konfiguraation- ja versionhallintaan liittyen kuin perinteisessä ohjelmistotuotannossa. Tavoitteeksi asetettiin myös perinteisen ohjelmistotuotannon ratkaisumallien kartoitus mobiilipelituotannossa esiintyneisiin ongelmiin. Perinteisen ohjelmistotuotannon ongelmien ja ratkaisumallien kartoitus suunniteltiin suoritettavaksi aikaisemman tutkimuksen avulla ja mobiilipelituotannon ongelmien kartoitus empiirisen tutkimuksen avulla. Tutkielman tavoite muotoiltiin kahdeksi tutkimusongelmaksi seuraavanlaisesti:

- Esiintyykö mobiilipelituotannossa ohjelmistojen konfiguraation- ja versionhallinnan yleisimpiä ongelmia?
- Miltä osin ohjelmistojen konfiguraation- ja versionhallinnan ratkaisumallit soveltuvat mobiilipeliteollisuuteen?

Tutkimusongelmaa lähdettiin ratkaisemaan edeten perinteisesti kirjallisuuskatsauksesta empiirisen tutkimuksen tutkimuskehyksen rakentamiseen. Tutkimuskehyksen valmistuttua suoritettiin empiirinen tutkimus ja tämän jälkeen analysoitiin saadut vastaukset. Tieteel-

linen pohja tutkimukselle rakennettiin laajalla kirjallisuuskatsauksella. Kirjallisuuskatsauksessa keskityttiin sekä ohjelmistotuotantoon, tarkentuen konfiguraation- ja versionhallintaan että mobiilijärjestelmiin, tarkentuen mobiilipelituotantoon.

7.2 Kirjallisuuskatsaus

Luvussa 2 käsiteltiin ohjelmistotuottajan toimialaa. Luvun tarkoitus oli selventää perinteisen ohjelmistotuottajan organisaatiota, sillä tutkielmassa tarkasteltiin nimenomaan perinteisen ohjelmistotuotannon konfiguraation- ja versionhallinnan ongelmia ja ratkaisuja mobiilipelituotannossa. Luvussa tehtiin pintapuolinen vertailu perinteisen ohjelmistotuottajan ja mobiilipelituottajan välillä. Vertailun avulla todettiin molempien olevan ohjelmistotuottajakentän osajoukkoja. Luvun 2 monimutkaisen kokonaisuuden hahmottamiseksi tehtiin kuvio (kuvio 1), josta näkee selkeästi luvun rakenteen, etenemisen ja siirtymän seuraavaan lukuun. Ohjelmistotuottajien nykyistä tilannetta analysoitiin pääosin Forten (1997) ja Aprilin ym. (2005) artikkeleihin perustuen. Forten (1997, 120) ja Aprilin ym. (2005, 197) mukaan ohjelmistotuottajat ovat joutuneet kohtaamaan globalisaation tuoman kilpailun ja tästä johtuen on entistä tärkeämpää tarjota korkealaatuisia tuotteita kilpailukykyiseen hintaan.

Tarkasteltavat konfiguraation- ja versionhallinta kuuluvat tehtäväalueista ohjelmistojen kehityksen ja ylläpidon alaisuuteen (April ym. 2005, 204; ISO 1995; Reichenberger 1989, 137). Tästä johtuen ohjelmistotuottajan nykyisen tilanteen selvityksen jälkeen keskityttiin ohjelmistojen kehityksen ja ylläpidon sekä näiden kanssa toimivien tehtäväalueiden kuvaamiseen. Kuvauksen yhteydessä rajattiin tutkielman ulkopuolelle laitteistovalmistajat, käyttötuki, infrastruktuuri ja tietokoneoperaatiot sekä asiakkaat ja käyttäjät (April ym. 2005, 199). Lisäksi yksikään näiden tehtäväalueiden rajapinnoista, jotka toimivat kehityksen ja ylläpidon kanssa yhteistoiminnassa, ei osoittautunut oleelliseksi tutkimusongelman kannalta.

Tehtäväalueiden kartoituksen jälkeen syvennyttiin, tutkielman kannalta oleellisiin, ohjelmistojen kehitykseen ja ylläpitoon. Ohjelmistojen kehityksen ja ylläpidon avainprosessit, jotka jaettiin molempien tehtäväalueiden osalta kolmeen liiketoimintaprosessiluokkaan,

lueteltiin yksityiskohtaisesti ja samalla suoritettiin rajaus tutkielmaan mukaan otettavista avainprosesseista. Liiketoimintaprosessiluokista tärkeimmäksi osoittautui tuki-prosessiluokka, jonka alaisina sekä kehityksen että ylläpidon puolella ovat konfiguraation- ja versionhallinta. (April ym. 2005, 204; ISO 1995; Conradi & Westfechtel 1998.) Mukaan tutkielmaan otettiin avainprosesseista myös dokumentaatio, sillä dokumentaationhallintaan kuuluu oleellisena osana konfiguraation- ja versionhallinta (Sachweh & Schafer 1995). Tutkittavaksi valittiin siis konfiguraationhallinta, versionhallinta ja dokumentaatio (konfiguraation- ja versionhallinnan ongelmien osalta). Abranin ym. (2004) mukaan konfiguraation- ja versionhallinta eivät eroa oleellisesti ohjelmistojen kehityksen ja ylläpidon välillä, joten kehitystä ja ylläpitoa ei käsitelty eri kokonaisuuksina. Dokumentaatiota tarkasteltiin osana konfiguraation- ja versionhallintaa.

Luvussa 3 tarkasteltiin valittuja avainprosesseja. Kohtaan 3.1 rakennettiin kuvio (kuvio 4), johon koostettiin kaikki oleelliset luvussa esiintyvät prosessit, termit, ongelmat ja ratkaisut. Kyseisen kuvion sisällöstä tuli niin laaja, että kuviossa esitettyjen kokonaisuuksien lähde-viittaukset ja tutkielman sisäiset viittaukset koostettiin taulukoksi (taulukko 1) kuvion perään. Kohta 3.2 sisälsi konfiguraationhallinnan ja kohta 3.3 versionhallinnan. Tiivistetysti konfiguraationhallinta on prosessi ohjelmiston konfiguraation tunnistamiseen tietyssä pisteessä ohjelmiston elinkaarta. Konfiguraationhallinnan tarkoituksena on systemaattisesti kontrolloida muutoksia konfiguraatioon ja ylläpitää konfiguraation eheys ja jäljitettävyys läpi ohjelmiston elinkaaren. (Abran ym. 2004.) Konfiguraationhallinta jaettiin kahteen Joerisin (1997) esittämään kokonaisuuteen, kokonaisuuksia olivat konfiguraationhallinnan tekninen ja hallinnollinen näkökulma. Tekniseen näkökulmaan rajattiin konfiguraation-hallinnan sisältämät mekanismit ja funktiot ohjelmistotuotteiden ja ohjelmisto-objektien hallintaan. Hallinnolliseen näkökulmaan taas rajattiin konfiguraationhallinnan organisatoriset ja hallinnolliset kokonaisuudet. (Joeris 1997.) Tekninen näkökulma koostui pääosin Estublierin ym. (2005) ja Dartin (1991) tekemistä konfiguraationhallintajärjestelmien määritelmistä. Suurin anti tutkielmalle tekniseltä kannalta oli Dartin (1991, 7) määrittelemät kategoriat koskien järjestelmävaatimuksia. Hallinnollinen näkökulma rakentui Abranin ym. (2004) ja Estublierin ym. (2005) artikkeleiden pohjalta tehdyn kuvion (kuvio 7) mukaiseksi. Kuvion avulla konfiguraationhallinta saatiin hallinnolliselta kannalta jaettua

selkeiksi aihealueiksi ja prosesseiksi. Aihe-alueita olivat prosessinhallinta, muutoksenhallinta, tilanvalvonta, laadunvalvonta, jakelunhallinta ja konfiguraatioiden tunnistaminen.

Perustuen Sachwehin ja Schaferin (1995), Conradin ja Westfechtelin (1998) ja Reichenbergerin (1989) määritelmiin, versionhallinnan keskeinen aspekti on aikaulottuvuuden seuranta ohjelmistotuotteiden tai ohjelmisto-objektien kehityksessä. Versionhallinnan avulla voidaan tarvittaessa palata tiettyyn ajalliseen pisteeseen ohjelmistotuotteen tai ohjelmisto-objektien elinkaaressa ja tarkastella kohteen sen hetkistä tilaa konfiguraationhallinnan avulla. Versionhallinta jaettiin useisiin alakohtiin, alakohtia olivat versiohistoria, versiointimalli, versiokirjasto, ohjelmistotuotteen elinkaari, versionhallinnan ratkaisut dokumentaation ongelmiin ja konfiguraationhallintajärjestelmä versionhallinnan tukena.

Versiohistorian tueksi esitettiin useita erilaisia revisiopuita (Conradi & Westfechtel 1998, 242; Reichenberger 1989, 137; Sachweh & Schafer 1995, 25). Lisäksi määriteltiin ITIL:n (2005, 205–207) mukainen revisioluokitus. Conradin ja Westfechtelin (1998, 235) mukaan versiointi, varsinkin laajoissa ohjelmistoissa, ei saa olla satunnaista, vaan sen pitää olla tarkasti annetun versiointimallin mukaisesti suoritettu. Versiointimallin määrittelyn lisäksi alakohdassa esiteltiin versiointimallin tukena toimiva ohjelmistojen kehityksen ja ylläpidon luokittelun päätöspuu (Chapin ym. 2001, 10).

Versiokirjastoissa, jotka toimivat versionhallinnan pääasiallisina työkaluina, säilytetään ohjelmisto-objektien ja ohjelmistotuotteiden eri versioita (IEEE 1990, 68). Versiokirjastojen määrittelyssä esiteltiin kolme yleistä versiokirjastotyyppiä. Näitä olivat työskentelykirjasto, tukikirjasto ja tuotekirjasto (Abran ym. 2004, 7 – 7; ITIL 2005, 203). Lisäksi esiteltiin useisiin tieteellisiin lähteisiin perustuen tekijät, jotka tulisi ottaa huomioon versiokirjastoa suunniteltaessa. Tekijöitä olivat fyysinen sijainti, nimeämiskäytännöt, tehtäväalueet, turvallisuusjärjestelyt, laajuus, säilytysaika, kapasiteettisuunnitelmat ja seurantamenetelmät (Conradi & Westfechtel 1998, 235; Dart 1991, 7; ITIL 2005, 209; Tichy 1985).

Ohjelmistotuotteen elinkaarta käsittelevässä alakohdassa määriteltiin ohjelmistotuotteen elinkaari ja käsiteltiin ne tiedot, jotka pitäisi pystyä selvittämään versiohistorian avulla ohjelmistotuotteen elinkaaresta. Näiden tietojen avulla organisaatio kykenee kokonais-

valtaisemmin arvioimaan ohjelmistotuotantoprosessia. Seuraavassa alakohdassa keskityttiin dokumentointiin liittyvään ongelmaan versionhallinnassa. Ongelmana olivat termien merkitysten eroavaisuudet eri standardeissa (Chapin ym. 2001, 6). Lisäksi alakohdassa esiteltiin dokumentaationhallinnan avuksi muutospuurakenne (Reichenberger 1989, 137). Viimeinen versionhallinnan alakohta koski versionhallinnan ja konfiguraationhallintajärjestelmän yhteistyötä. Yhteistyö rajattiin kahden Dartin (1991, 7) määrittelemän kategorian alaisuuteen, kategorioita olivat Seuranta ja Tarkastus. Version-hallinnan seurannan avuksi konfiguraationhallintajärjestelmän täytyy kyetä tarjoamaan edellytykset graafisten esityksien tekemiseen (Conradi & Westfechtel 1998, 234). Tarkastus kategoriaan ratkaisuksi esitettiin OID eli ohjelmisto-objektin tunnistin. Ohjelmisto-objektien tunnistimet jaettiin kahteen eri tyyppiin, ulkoisiin ja sisäisiin tunnistimiin (Conradi & Westfechtel 1998, 235). Jälkimmäinen on konfiguraationhallinta-järjestelmän automaattisesti asettama tunnistin.

Luvussa 4 käsiteltiin mobiilipelien tuottajan toimialan ominaispiirteitä. Luku koostui seuraavista kokonaisuuksista: mobiilijärjestelmät, mobiilipelit, mobiilipeliteollisuuden rakenne ja mobiilipelien tuottaja. Esimerkkinä mobiilijärjestelmistä käsiteltiin yleisesti tieteellisissä lähteissä hyväksytty mobiilijärjestelmän rakenne (kuvio 12) (Dunham & Helal 1995, 5). Kohdassa esiteltiin myös mobiilijärjestelmien nykyisiä käyttökohteita ja tutkiel-massa tarkasteltavien mobiililaitteiden (eli matkapuhelimien) alustat.

Mobiilipelejä käsittelevä kohta aloitettiin lyhyellä katsauksella tietokonepelien historiaan, jonka jälkeen siirryttiin mobiilipeleihin. Mobiilipelien suosion todettiin kasvaneen moninkertaiseksi 2000-luvun aikana. Tämän väitteen tukemiseksi kohdassa tutkittiin muun muassa mobiilipeliteollisuuden liikevaihdon kasvua 2000-luvulla, todeten liikevaihdon kasvaneen vuosien 2000–2006 välisenä aikana 0,59 miljardista eurosta vuodessa noin 5,15 miljardiin euroon vuodessa (Son & Tan 2008, 36).

Mobiilipeliteollisuuden rakenne koostettiin Akkawin ym. (2004) ja Funkin (2004) määri-telmiin perustuen kuvioksi (kuvio 13). Heidän mukaansa mobiilipeliteollisuus koostuu mobiilipelien tuottajasta, tarjoajasta, käyttäjistä, laitteistovalmistajasta, verkon tarjoajasta ja operationaalisen teknologian tarjoajasta (Akkawi ym. 2004, 78; Funk 2004, 77).

Mobiilipeliteollisuuden rakenteen koostaminen oli tärkeätä, sillä kyseistä rakennetta voitiin verrata mobiilipelituottajien näkemykseen mobiilipeliteollisuudesta. Tästä saatiin arvokasta tietoa tieteellisten lähteiden ja mobiilipelituotannon vastaavuuksista ja eroavaisuuksista.

Mobiilipelituottajilla todettiin olevan vain harvoin tarpeeksi laaja verkosto pelin tehokkaaseen markkinointiin ja jakamiseen, joten yhteistyö mobiilipelien tarjoajan kanssa on oleellista mobiilipelituottajalle. Mobiilipelituotannon odotetaan ohittavan sekä tietokone- että konsolipelituotannon markkina-arvossa vuoteen 2010 mennessä. (Son & Tan 2008, 37–38.) Aluksi mobiilipelejä tehtiin vain WAP tai c-HTML merkkauskielellä, jotka olivat ominaisuuksiltaan erittäin rajallisia, verrattuna nykyisin käytössä oleviin Java-pohjaiseen J2ME sovelluskehykseen ja C-kieleen perustuvaan BREW:in (Funk 2004; Son & Tan 2008). J2ME:stä esitettiin myös tarkempaa teknistä tietoa pohjautuen Williamsin ja Burgen (2004) artikkeliin.

7.3 Mobiilipelituotannon keskeisimmät ongelmat

Mobiilipelituotannossa, ainakin yhden kohdeorganisaation mukaan, määritellyistä 24 kysymyksestä + tai ++ luokituksen sai 21 kappaletta. Yhteenvetoon on koostettu yleisimmin esiintyneet ongelmat ja analysoitu empiirisen tutkimuksen tuloksia kokonaisuutena. Empiirisen tutkimuksen perusteella mobiilipelituotannossa koettiin suurimmat ongelmat seuraavien kysymysten aihepiiriin liittyen:

- Alalla on esiintynyt ongelmia suurikokoisten mobiilipelien rakenteen hallinnassa?
- Toimivatko mobiilipelien kehittäjän/ylläpitäjän käyttämät ohjelmistot loogisena kokonaisuutena?
- Onko mobiilipelien kehittäjillä/ylläpitäjillä konfiguraationhallinnan tarkkuustaso selvillä/yhtenäinen?
- Onko mobiilipelien tuottajaorganisaatioissa esiintynyt ongelmia vastuualueissa, resursseissa, aikatauluissa, työkaluissa yms.?

- Ongelmia saavuttaa suunniteltu/määritelty mobiilipeli?
- Mobiilipeleistä/ohjelmisto-objekteista kehittyy usein liian monta eri revisiota tai varianttia?
- Käytetäänkö päivitysten kokoluokissa mitään rajauksia?

Suurikokoisten mobiilipelien rakenteen hallinnassa kaikki vastanneet kohdeorganisaatiot kokivat ongelmia, kohdeorganisaatio C ei vastannut tähän kysymykseen. Suurimmaksi syyksi tähän ongelmaan ilmeni laitteistoskaala. Kun kyseessä on normaalia suurikokoisempi mobiilipeli ja kyseinen mobiilipeli pitää tehdä moniin eri matkapuhelinmalleihin mahdollisimman vähällä koodimäärällä, aiheuttaa laitteistoskaala suuria ongelmia mobiilipelituotannossa. Kohdeorganisaatio A mainitsi, että on esiintynyt hankaluuksia saada ”otetta” suurikokoisten mobiilipelien tuotannossa. Tosin A mainitsi, ettei ongelma johtunut siitä, että kyseessä on juuri mobiilipeli.

Valtaosa kohdeorganisaatioista oli sitä mieltä, etteivät kehittäjien ja ylläpitäjien käyttämät ohjelmistot toimi loogisena kokonaisuutena. Sekä A:n että B:n mukaan tosin omat järjestelmät on kehitetty tarpeeksi hyvin, että ne toimivat loogisena kokonaisuutena läpi mobiilipelituotannon elinkaaren. C:n mukaan kehitystyötä joudutaan tekemään useilla eri ohjelmistoilla ja heillä ei ainakaan vielä ole saumatonta integraatiota ohjelmistojen välillä. F:n mukaan tarjolla olevat järjestelmät eivät sovellu varsinkaan dokumentaationhallinnan puolelle. Mobiilipelituotannossa valmiit tarjolla olevat ohjelmistot eivät siis ole selkeä kokonaisuus ja dokumentaationhallinnan asettamiin haasteisiin nämä ohjelmistot eivät tarjoa ratkaisua. Käytettäviin ohjelmistoihin liittyvissä ongelmissa korostuu selkeästi mobiilipelituotantoon siirtyvän organisaation tilanne. Vaikka muun muassa Nokia tarjoaa valmiita työkalupaketteja mobiilipelituottajille, suoranaisesti mobiilipelituotantoon sopivia selkeänä kokonaisuutena toimivia järjestelmiä ei ole ja oman järjestelmän räätälöiminen vaatii resursseja ja kokemusta mobiilipelituotannosta (Son & Tan 2008, 37).

Viisi kuudesta kohdeorganisaatiosta oli sitä mieltä, että konfiguraationhallinnan tarkkuustaso ei ole selvillä tai yhtenäinen. E oli ainoa kohdeorganisaatio, joka ei kokenut ongelmia konfiguraationhallinnan tarkkuustasoon liittyen. A ja B korostivat konfiguraationhallinnan

tarkkuustason olevan omalta kohdaltaan selvillä, mutta yhtenäinen se ei heidän mukaansakaan ole. Ongelmaksi osoittautuivat tilaajien vaatimusten eroavaisuudet. Erilaiset spesifikaatiot pakottavat luomaan joissakin tapauksissa täysin oman projektin eri tilaajille. A perusteli vastausta seuraavanlaisesti: ”Pelistudio X voi tehdä alihankintana peliprojektin julkaisijalle Y, joka voi haluta vain yhden ns. referenssibuildin pelistä. Julkaisija Z taas voi vaatia sata.”. Ongelmat ovat aiheutuneet siis siitä, että mobiilipelituottajan ja mobiilipelin tilaajan välisiä spesifikaatioita ei ole standardoitu.

Kaikki kohdeorganisaatiot kokivat ongelmia mobiilipelituotannossa vastuualueisiin, resursseihin, aikatauluihin tai työkaluihin liittyen. Vastausten perusteluista selvisi, että suurimmat ongelmat aiheutuvat aikatauluista ja työkaluista. A ja B totesivat aikataulut suurimmaksi ongelmaksi, sillä lisenssinhaltijan tai tilaajan asettamat aikarajat ovat usein todella tiukat. B:n mukaan aikataulut ovat usein niin tarkasti määrättyjä, että mobiilipeleistä joudutaan useimmiten ennemminkin tiputtamaan joitakin osia pois, kuin joustamaan aikatauluissa. Tiukat aikataulut yhdessä sen ongelman kanssa, että selkeää pelikonseptia ei ole usein olemassa projektin alkuvaiheessa, tuovat haasteita mobiilipelituotantoon. Työkalut ongelmien lähteeksi tunnisti kohdeorganisaatio C, jonka mukaan työkalut ja niiden käyttö ei ole täysin yhteismitallista ja määrämuotoista. Varsinkin alihankkijoita käytettäessä työkaluihin liittyvä kompleksisuus lisääntyy. Toisin sanoen, jotta käytettävät työkalut saataisiin toimimaan selkeänä järjestelmänä sekä omassa organisaatiossa että oman organisaation ja alihankkijoiden välillä, tarvitaan investointeja käytettäviin työkaluihin.

Suunnitellun/määritellyn mobiilipelin saavuttaminen osoittautui yhteiseksi ongelmaksi kaikille muille kohdeorganisaatioille paitsi C:lle ja F:lle. C:n mukaan heidän organisaatiossa keskitytään projekteissa, ehkä liiankin paljon, pelkästään pelien tuottamiseen, joten ongelmia suunnitellun ja määritellyn mobiilipelin saavuttamisessa ei esiintynyt. Tosin C:n mukaan tämän on todettu kostautuvan muilla hallinnollisilla osa-alueilla. A:n mukaan suunnitellun/määritellyn mobiilipelin saavuttaminen ei ole yksistään mobiilipelien ongelma, vaan kokonaisuutena peleihin liittyvä ongelma. Tämä johtuu A:n ja B:n mukaan siitä, että pelien tuottaminen ei ole aina suoraviivaista, vaan välillä todella luovaa työtä. Kyseinen ongelma ei B:n mukaan esiinny lisensoiduissa peleissä niin suurena, sillä tällöin

ei tarvitse aloittaa mobiilipelin kehitystä täysin puhtaalta pöydältä, vaan konsepti on jo alusta asti selkeä.

Mobiilipeleistä/ohjelmisto-objekteista kehittyy liian monta revisiota tai varianttia kaikkien kohdeorganisaatioiden mukaan, lukuun ottamatta kohdeorganisaatio E:tä. Tosin oman organisaation toiminnassa ongelmia versioiden liiallinen muodostuminen aiheutti vain B:lle, D:lle ja F:lle. A mukaan heidän organisaatiossaan on kyseinen ongelma vältetty keskittämällä tarpeeksi voimavaroja mobiilipelituotannon hallinnollisiin osa-alueisiin. B totesi ongelman esiintyvän jos mobiilipelistä tulee suosittu, sillä tällöin joudutaan tekemään normaalia enemmän variantteja kyseisestä mobiilipelistä. C ei ole vielä kohdannut ongelmia omalla kohdallaan tällä alueella, mutta ennakoi tilanteen muuttuvan huonompaan suuntaan, kun organisaation jakelukanava kansainvälistyy.

Mobiilipelituotannossa peleihin ja pelien ohjelmisto-objekteihin liittyen, päivitysten kokoluokkien rajauksia käyttää vain kohdeorganisaatio D. Muilla kohdeorganisaatioilla voi olla versiointimalli määriteltynä ja revisionumerointi käytössä, mutta päivitysten kokoluokille ei ole tehty rajauksia. D:n lisäksi poikkeuksen teki kohdeorganisaatio B, joka totesi käyttävänsä organisaation omissa työkaluissa ITIL:n (2005, 205) mukaista revisioluokkien määrittelyä, mutta peleissä ja ohjelmisto-objekteissa päivitysten kokoluokkia ei ole erikseen määritelty.

7.4 Ratkaisumallit

Tieteellisistä lähteistä määriteltiin useita teorioita, jotka osoittautuivat oleelliseksi myös mobiilipelituotantoon. Ensimmäiseksi käsitellään kuitenkin kirjallisuuskatsaukseen kartoitetut teoriat, joista ei ollut odotettua hyötyä mobiilipelituotannon kokemiin ongelmiin. Kysymykset, joiden aihepiirissä kohdeorganisaatiot eivät kokeneet ongelmia, olivat:

- Onko mobiilipelien kehitys- ja ylläpitoprosessi systemaattinen/jäljitettävä?
- Esiintyykö mobiilipelien tuotannossa jokin seuraavista ongelmista, konfiguraatioista ei saada tarpeeksi tietoa, konfiguraatiot eivät ole määritellyssä tilassa tai konfiguraationhallinta ei ole tehokasta?

- Onko esiintynyt ongelmia revisioiden aikaulottuvuuden kanssa?

Jokainen esitelty kysymys oli johdettu tieteellisistä lähteistä ja jokaiseen kysymykseen oli kartoitettu alustava ratkaisumalli. Tästä syystä edellä mainittujen kolmen kysymyksen osalta määritellyt ratkaisumallit jäivät osittain tulosten analysoinnin ulkopuolelle, näitä ratkaisumalleja olivat konfiguraationhallinnan keskeinen tehtävä, konfiguraationhallinnan tilanvalvonta ja versionhallinnan revisiopuumallit. Tosin versionhallinnan revisiopuumalleja sovellettiin muiden versionhallinnan ongelmien ratkaisuissa.

Seuraavaksi käsitellään mobiilipelituotannon konfiguraation- ja versionhallinnan ongelmiin esiteltyjä ratkaisumalleja. Ratkaisumallit kysymyksittäin esiteltiin yksityiskohtaisesti kohdassa 6.3, joten tässä kohdassa on tehty yhteenveto oleellisimmista ratkaisumalleista. Alla olevaan taulukkoon (taulukko 7) on koostettu ratkaisumallien keskeisimmät lähteet ja kysymykset, joissa näitä ratkaisumalleja sovellettiin. Kohteiden sarakkeessa on käytetty lyhenteitä. TN tarkoittaa teknistä näkökulmaa konfiguraationhallinnassa, HN hallinnollista näkökulmaa konfiguraationhallinnassa ja VH versionhallintaa.

Jokaisessa ratkaisumallissa käytettiin aikaisempaan tutkimukseen perustuvaa tietoa. Konfiguraationhallinnan teknisessä näkökulmassa ratkaisumallit perustuivat pääosin Dartin (1991) ja Estublierin ym. (2005) artikkeleihin. Konfiguraationhallinnan hallinnollisessa näkökulmassa ratkaisumallit koostuivat suurimmilta osin Abranin ym. (2004) sekä ITIL:in (2005) teorioista. Versionhallinnan ratkaisumalleissa yleisin lähde oli Conradi ja Westfechtel (1998), lisäksi tukena käytettiin useita yksittäisiä lähteitä. Mobiilipelituotannon suurimmat ongelmat koettiin kysymysten TN 1, TN 3, HN 2, HN 3, HN 7, VH2 ja VH 4 aihepiiriin liittyen. Nämä ongelmat on käsitelty edellisessä kohdassa ja empiirisen tutkimuksen tulosten yhteydessä.

TAULUKKO 7. Lähteiden käyttö ratkaisumalleissa

Lähde	**Sivu # lähteessä**	**Ratkaisumalli**	**Kohde**
Abran ym. 2004	7 – 3	Konfiguraationhallintatietokanta	TN 6
	7 – (6–7)	Konfiguraatioiden tunnistaminen	HN 1
	7 – (2–5)	Prosessinhallinta	HN 2 & HN 3
	7 – (7–8)	Muutoksenhallinta	HN 4 & HN 5
	7 – 9	Laadunvalvonta	HN 7
	7 – 9	Jakelunhallinta	HN 8
	7 – 7	Versiokirjastot	VH 6
Bersoff ym. 1980 mukaan, Conradi & Westfectel 1998	233	Prosessituki	HN 3
Chapin ym.	10	Versiointimalli	VH 5
Conradi & Westfechtel 1998	–	Revisiopuut	VH 2
	233, 235	Versiointimalli	VH 5 & VH 8
Dart 1991	7	Kategoriat	TN 1, TN 5, TN 6 & TN 7
Eick ym. 2002	402–404	Versiohistoria	VH 7

(jatkuu)

TAULUKKO 7. (jatkuu)

Estublier ym. 2005	391	Näkymättömyys	TN 2
	392	Järjestelmätuki	TN 3 & TN 8
	392–393	Tuotetuki & prosessituki	TN 5 & HN 3
IEEE 1983 & IEEE 1988 mukaan, Conradi & Westfechtel 1998	233	Konfiguraatioiden tunnistaminen	HN 1
IEEE 1990	68	Versiokirjasto	VH 6
IEEE 1997	25	Laadunvalvonta	HN 7
Introna & Edgar 1997	35	Näkymättömyys	TN 2
ITIL 2005	124	Konfiguraationhallintatietokanta	TN 6
	233	Prosessinhallinta	HN 2 & HN 3
	126	Muutoksenhallinta	HN 4 & HN 5
	205–207	Revisioluokat	VH 4
	203	Versiokirjastot	VH 6
Korel ym. 1991	161	Järjestelmätuki	TN 8
Reichenberger 1989	–	Revisiopuut	VH 1
Sachweh & Schafer 1995	25	Revisiopuut	VH 1
Tichy 1985 mukaan, Conradi & Westfechtel 1998	235	Versiointimalli	VH 8

Teknisen näkökulman kysymys 1 käsitteli suurikokoisten mobiilipelien rakenteen hallintaa. Kysymys käsiteltiin luokan ++ mukaisesti, kyseessä oli siis ongelma, johon kohdeorganisaatioilla ei ollut omaa ratkaisumallia. Ratkaisumallissa keskityttiin suurikokoisten mobiilipelien hallintaan, sillä ratkaisumallin yhteydessä todettiin laitteistoskaalan suuruuden johtuvan mobiilipelituottajien ulkopuolisista tekijöistä. Ratkaisumallissa todettiin ensin, tieteellisten lähteiden avulla, suurikokoisten konfiguraatioiden hallinnassa esiintyvän ongelmia myös perinteisessä ohjelmistotuotannossa. Tämän jälkeen esiteltiin tekniseen näkökulmaan soveltuva ratkaisu kyseiseen ongelmaan.

Mobiilipelituottajien käyttämät ohjelmistot eivät toimineet loogisena kokonaisuutena, eli tekniseltä näkökulman kysymyksen 3 aihepiirissä esiintyi ongelmia mobiilipelituotannossa. Kysymys käsiteltiin luokan ++ mukaisesti, sillä neljästä ongelmat tunnistaneesta vain yhdellä kohdeorganisaatiolla oli oma ratkaisumalli. Ratkaisumallissa korostettiin järjestelmätuen merkitystä ohjelmistotuotantoon samalla kritisoiden mobiilipelituotantoon tarjolla olevien järjestelmien tilannetta.

Mobiilipelituotannossa esiintyi ongelmia konfiguraationhallinnan yhtenäisyyden kanssa viidessä kuudesta kohdeorganisaatiosta (hallinnollisen näkökulman kysymys 2). Ongelmaan ei ollut yhdelläkään kohdeorganisaatiolla omaa ratkaisumallia, joten kysymys käsiteltiin luokan ++ mukaisesti. Ratkaisumallissa keskityttiin prosessinhallintaan, joka on yksi Abranin ym. (2004) määrittelemistä aihealueista. Ratkaisumallin avulla pyrittiin korjaamaan mobiilipelin tuottajan ja tilaajien välille muodostuneita ongelmia.

Hallinnollisen näkökulman kysymyksen 3 aihepiiriin liittyen esiintyi ongelmia kaikissa kohdeorganisaatioissa, joten kysymys käsiteltiin luokan ++ mukaisesti. Ratkaisumallissa keskityttiin aikatauluihin ja käytettäviin työkaluihin liittyviin ongelmiin. Keskeisenä osana ratkaisumallia toimi prosessituen ja tuotetuen sisäiset määritelmät.

Suunnitellun ja määritellyn mobiilipelin saavuttamisessa esiintyi ongelmia neljällä kuudesta kohdeorganisaatiosta. Kysymys käsiteltiin luokan ++ mukaisesti, sillä yhdelläkään kohdeorganisaatiolla ei ollut omaa ratkaisumallia ongelmiin. Kysymyksessä käsiteltiin laadunvalvontaan liittyviä ongelmia. Ratkaisumalli perustui pääosin Abranin ym. (2004) määritelmään konfiguraationhallinnan laadunvalvonnasta.

Versionhallinnassa toiseksi yleisin ongelma oli varianttien liian suuri määrä. Varianttien määrän kasvamiseen ei ollut kohdeorganisaatioilla omaa ratkaisumallia, joten kysymyksen käsittelyssä käytettiin ++ luokkaa. Varianttien liian suureen määrään liittyvät ongelmat ovat keskeinen asia mobiilipelituotannolle, sillä siitä muun muassa aiheutuu ongelmia revisiopuiden käytölle ja lisäksi versionhallinnan luotettavuus kärsii (Conradi & Westfechtel 1998; ITIL 2005, 239). Tieteellisistä lähteistä koostettu ratkaisumalli ei sinällään ottanut kantaa varianttien määrän kasvamisen estämiseen, vaan ratkaisumallissa keskityttiin tekijöihin, jotka aiheuttivat ylimääräisten varianttien muodostumisen.

Versionhallinnan apuna mobiilipeleissä ja mobiilipelien ohjelmisto-objekteissa käytti päivitysten kokoluokissa ennalta määriteltyjä rajauksia vain yksi kohdeorganisaatio. Kyseessä oli siis versionhallinnan yleisin ongelma, joten kysymys käsiteltiin luokan ++ mukaisesti. Ratkaisumallissa perusteltiin päivitysten rajaamisen tärkeyttä tuomalla esille sen tarjoamat edut versionhallinnalle. Tämän jälkeen keskityttiin ITIL:n (2005) pohjalta rakennettuun ratkaisumalliin, jossa kerrattiin kirjallisuuskatsauksessakin esitelty revisioluokitus.

7.5 Keskeiset tulokset

Tarkkaa mobiilipelituottajaorganisaatioiden määrää ei ole tiedossa, mutta useiden foorumien, yritysluetteloiden ja yritysten omien kotisivujen kartoituksen perusteella Suomessa olisi 12 aktiivista mobiilipelituottajaa. Empiirisen tutkimuksen tulokset edustavat tämän tiedon perusteella noin 50 % suomalaisista mobiilipelituottajaorganisaatioista, joten empiirisen tutkimuksen tuloksia voidaan pitää kohtuullisen kattavana otoksena toimialasta. Kattavuuden lisäksi empiirisen tutkimuksen vastaajiksi onnistuttiin saamaan asiantuntevat henkilöt, sillä jokainen vastaajista oli organisatorisista ja teknisistä kokonaisuuksista vastaava henkilö omassa organisaatiossaan. Vastaajia olivat muun muassa tekniset päälliköt, kehityspäälliköt ja toimitusjohtajat.

Empiirisen tutkimuksen avulla haettiin vahvistusta tutkimusongelmaan, esiintyykö mobiilipelituotannossa ohjelmistojen konfiguraation- ja versionhallinnan yleisimpiä ongelmia. Tulosten perusteella konfiguraation- ja versionhallinnan ongelmat olivat

haasteellisia myös mobiilipelituotannossa, joten ensimmäiseen tutkimusongelmaan saatiin vastaus. Esitetyissä ratkaisumalleissa kyettiin käyttämään useita tieteellisiä lähteitä, vaikka keskeisenä osana ratkaisuja olivat Abranin ym. (2004), Dartin (1991) ja Estublierin ym. (2005) artikkelit. Monipuolisten lähteiden avulla ratkaisumallit eivät keskittyneet vain yhden lähteen näkökantaan ongelmien ratkaisuissa. Ratkaisumallien kartoitus ja soveltaminen mobiilipelituotantoon kuului osana toiseen tutkimusongelmaan. Tutkimusongelma oli: Miltä osin ohjelmistojen konfiguraation- ja versionhallinnan ratkaisumallit soveltuvat mobiilipeliteollisuuteen? Ratkaisumallien toimivuutta mobiilipelituotannossa ei pyritty todentamaan käytännössä, vaan analysoimalla kohdeorganisaation vastauksia kyettiin rakentamaan perinteiseen ohjelmistotuotantoon suunnatuista lähteistä mobiilipelituotannon ongelmiin sopivat ratkaisumallit. Kohdeorganisaatioiden kokemat ongelmat tosin olivat mobiilipelituotannolle spesifejä, joten perinteisen ohjelmistotuotannon ratkaisumalleja jouduttiin osin muokkaamaan kyseisiin vaatimuksiin sopiviksi. Suurin osa ratkaisumalleista tarjosi tietämystä siitä, millaisia ominaisuuksia ja tukea konfiguraation- ja versionhallinnan tulisi tarjota. Ratkaisumalleissa ei siis keskitytty tarkkoihin teknisiin ratkaisuihin, sillä mobiilipelituottajien käyttämät järjestelmät eivät olleet kirjallisuuskatsausta suoritettaessa tiedossa. Koska ratkaisumallit kyettiin koostamaan tieteellisiin lähteisiin perustuen, saatiin toiseenkin tutkimusongelmaan vastattua. Eli ohjelmistojen konfiguraation- ja versionhallinnan ratkaisumallit soveltuvat suurimmalta osin mobiilipelituotantoon.

Tutkielman haasteellisimmaksi osaksi tiedostettiin konfiguraation- ja versionhallinnan koostaminen tiiviiksi ja selkeäksi kokonaisuudeksi tieteellisistä lähteistä. Kirjallisuuskatsaukseen jouduttiin tästä syystä sijoittamaan suurin osa aikataulullisista resursseista. Kirjallisuuskatsauksesta keskeisimpinä tuloksina olivat:

- Konfiguraation- ja versionhallinnan jakaminen omiksi kokonaisuuksiksi.
- Mobiilipeliteollisuuden rakenteen kuvaaminen.

Empiirisessä tutkimuksessa käytetty menetelmä (kuvio 14) osoittautui varsin toimivaksi, ensin kartoitettiin toimialan rakenne ja varmistettiin organisaatiossa toteutettavan tarkkailtavia avainprosesseja. Tästä saatiin tärkeää tietoa toimialasta, toimialalla käytettävästä

terminologiasta ja organisaation sisäisestä toiminnasta. Tiedon avulla helpottui varsinaisen empiirisen tutkimuksen (Osio 3) suorittaminen ja tulosten analysointi, sillä alun kartoituksen (Osio 1) avulla esille tuli esimerkiksi useita terminologisia eroavaisuuksia. Eroavaisuudet olisivat voineet vaikeuttaa tulosten analysointia, elleivät ne olisi tulleet perusteluin esille aikaisemmin. Lisäksi käytetyn menetelmän mukaan tärkeimmät termit määriteltiin sekä osiossa 1 että osiossa 3, jotta terminologian aiheuttamat virheet (taulukko 2) empiirisessä tutkimuksessa saataisiin minimoitua.

Konfiguraation- ja versionhallinnan ongelmia mobiilipelituotannossa ei ole ennen tätä tutkielmaa kartoitettu tieteellisissä lähteissä, joten tästä tutkielmasta saadut tulokset ovat ainutlaatuisia ja niitä ei vielä voida verrata muihin tutkimuksiin. Tutkielman alussa jo asetettiin hypoteesiksi laitteistoskaalan aiheuttamat erikoisvaatimukset mobiilipelituotantoon, hypoteesi osoittautui todeksi empiirisen tutkimusten tuloksien perusteella. Ennen tutkielman aloittamista ei osattu ennakoida tarjolla olevien konfiguraationhallintajärjestelmien ja työkalujen yhteensopimattomuutta mobiilipelituotantoon sekä mobiilipelin tuottajan ja tilaajan välisessä suhteessa esiintyviä ongelmia. Kuitenkin ratkaisumallien avulla kyettiin esittämään ratkaisut, myös kyseisten tekijöiden aiheuttamiin ongelmiin.

Tutkielmasta saatiin keskeisinä tuloksina muutakin, kuin vastaukset tutkimusongelmiin. Tutkielman avulla osoitettiin, että aikaisempia tutkimuksia ohjelmistotuotannossa voidaan soveltaa myös erikoisemmille toimialoille, vaikka osa käytetyistä lähteistä on varhaisemmalta ajalta kuin mobiilipelituotanto. Ratkaisumallien soveltaminen mobiilipelituotantoon tosin vaati huomattavan paljon työtä. Kohdeorganisaatioille keskeisimpinä tuloksina toimivat organisaatiokohtaiset ongelmat ja ratkaisumallit, sillä näiden perusteella vastaajat voivat halutessaan vähentää omassa organisaatiossaan esiintyviä puutteita.

7.6 Empiirisen tutkimuksen luotettavuus

Tässä kohdassa arvioidaan empiirisen tutkimuksen luotettavuutta ja mahdollisia virhelähteitä. Virheet voivat olla lähtöisin puutteellisesta kirjallisuuskatsauksesta, empiirisen tutkimuksen tutkimuskehyksestä, vastaajista tai vastausten analysoinnista. Tutkielman

luotettavuutta on pyritty pitämään yllä, ottamalla ennalta huomioon edellä mainitut ongelmien aiheuttajat ja reagoimalla näihin ongelmien aiheuttajiin.

Siinä tapauksessa, jos kirjallisuuskatsaus on ollut puutteellinen, kaikkia oleellisimpia konfiguraation- ja versionhallinnan ongelmia ei ole välttämättä osattu tiedustella kohdeorganisaatioilta. Tämä ei varsinaisesti aiheuta virheellisiä tuloksia, vaan joitakin oleellisia ongelmia on saattanut jäädä tiedustelematta. Empiirisessä tutkimuksessa käytetyssä tutkimuskehyksessä on pyritty eliminoimaan virheitä aiheuttavat tekijät. Muun muassa terminologian aiheuttamien ongelmien ennaltaehkäisemiseen kiinnitettiin paljon huomiota. Tutkimuksen luotettavuuteen vaikuttavia tekijöitä voi tästä huolimatta esiintyä. Vastauksia analysoitaessa esimerkiksi huomattiin konfiguraationhallinnan kysymyksen 3 olevan aihepiiriltään turhan laaja. Lisäksi vastaajien osaamistasossa on voinut olla eroavaisuuksia ja osassa vastauksia on saattanut täten olla virheitä. Vastauksia on kuitenkin pyritty analysoimaan kokonaisuutena, joten mahdolliset virheet vastauksissa ovat todennäköisimmin tulleet esille vastausten ristiinvertaamisen yhteydessä. Vastaajien analysoinnissa voi aina esiintyä virheitä. Analysoinnin luotettavuuden parantamiseksi otettiin vastauksiin mukaan selkeät Kyllä/Ei vastaukset tekstikenttien lisäksi. Tämän avulla tekstikenttään tehdyn perustelun kautta ei vastausta voida analysoida vääräntyyppiseksi. Toisaalta mainitun konfiguraationhallinnan kysymyksen 3 laajuuden aiheuttamat ongelmat ehkäistiin tekstikenttien avulla, sillä Kyllä/Ei vastauksen lisäksi, kohdeorganisaatioilta saatiin tarkennukset millä osa-alueilla he kokivat ongelmia.

7.7 Jatkotutkimukset

Tutkielmassa jouduttiin keskittymään konfiguraation- ja versionhallinnan lisäksi mobiilipelituotantoon. Tästä johtuen konfiguraation- ja versionhallinnasta rakennetut kokonaisuudet eivät olleet täysin kattavia. Jatkotutkimuksena voitaisiin keskittyä rakentamaan selkeä viitekehys joko molemmista tai vain toisesta avainprosessista.

Kirjallisuuskatsauksen pohjalta rakennettua tutkimuskehystä voidaan pienellä muokkauksella hyödyntää myös muiden toimialojen tutkimiseen. Tällaisia kohteita voivat olla mitkä tahansa ohjelmistotuottajat, joiden oletetaan kokevan ongelmia konfiguraation- ja version-

hallintaan liittyen. Jotta rakennettua tutkimuskehystä voitaisiin soveltaa samoin menetelmin, täytyy kohteena olevasta toimialasta kyetä rakentamaan selkeä kokonaisuus käytössä olevien tieteellisten lähteiden avulla.

Empiirisen tutkimuksen pohjalta rakennettuja ratkaisumalleja voitaisiin soveltaa käytännössä mobiilipelituotantoon. Tällöin tehtäisiin tapaustutkimus yhteen kohdeorganisaatioon ja tutkimuksessa käytettäisiin osallistuvaa havainnointia (Nunamaker ym. 1991 mukaan, Mathiassen 1998, 69) tutkimusmenetelmänä. Empiirisessä tutkimuksessa keskityttiin myös vain Suomessa toimiviin mobiilipelituottajiin. Mobiilipeliteollisuuden rakenne ja tarkasteltujen avainprosessien sisältämät ongelmat voivat olla erilaisia esimerkiksi Ruotsissa tai Yhdysvalloissa. Empiirisen tutkimuksen toteuttaminen haastatteluna ei tällöin olisi todennäköisesti mahdollista, mutta Korppiin rakennettu kysely toimisi sellaisenaan, kohdekieleen kääntämisen jälkeen.

LÄHTEET

Abran A., Bourque P., Dupuis R. & Moore J.W. 2004. Guide to the Software Engineering Body of Knowledge: 2004 Version – SWEBOK. New Jersey: IEEE Computer Society Press

Akkawi A., Schaller S., Wellnitz O. & Wolf L. 2004. A Mobile Gaming Platform for the IMS. Feng W. (Toim.) Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games Portland, Oregon, August 30. New York: ACM Press, 77–84.

Ambriola V., Bendix L. & Ciancarini P. 1990. The Evolution of Configuration Management and Version Control. Software Engineering Journal 5(6), 303–310.

April A., Hayes J.M., Abran A. & Dumke R. 2005. Software Maintenance Maturity Model (SM^{mm}): The Software Maintenance Process Model. Journal of Software Maintenance and Evolution: Research and Practice 17(3), 197–223.

ATK-sanakirja 4.0. 2003. Tietotekniikan Liiton Sanastotoimikunta. ISBN: 951-762-831-5.

Babich W.A. 1986. Software Configuration Management. Massachusetts: Addison-Wesley publishing company

Bersoff E.H., Henderson V.D. & Siegel S.G. 1980. Software Configuration Management: An Investment in Product Integrity. New Jersey: Prentice-Hall.

Chapin N., Hale J.E., Khan K., Ramil J.F. & Tan W-G. 2001. Types of Software Evolution and Software Maintenance. Journal of Software Maintenance and Evolution: Research and Practice 13(1), 3–30.

Chen L. & Skelton G.W. 2005. Mobile Commerce Application Development. Hershey: CyberTech Publishing.

Conradi R. & Westfechtel B. 1998. Version Models for Software Configuration Management. ACM Computing Surveys 30(2), 232–282.

Damsgaard J. & Marchegiani L. 2004. Like Rome, a Mobile Operator's Empire Wasn't Built in a Day!: a Journey Through the Rise and Fall of Mobile Network Operators. Teoksessa Janssen M., Sol H.G. & Wagenaar R.W. (Toim.) Proceedings of the 6th International Conference on Electronic Commerce, Delft, The Netherlands, October 25–27. New York: ACM Press, 639–648.

Dart S. 1991. Spectrum of Functionality in Configuration Management Systems. Software Engineering Institute. Carnegie Mellon University. Pittsburgh, Pennsylvania 15213. [Viitattu 11.12.2007]. Saatavilla www-muodossa <ftp://ftp.sei.cmu.edu/pub/case-env/config_mgt/tech_rep/cm_spectrum_of_func_TR11_90.pdf>

Dunham M.H. & Helal A. 1995. Mobile Computing and Databases: Anything New? ACM SIGMOD Record 24(4), 5–9.

Eick S.G., Graves T.L., Karr A.F., Mockus A. & Schuster P. 2002. Visualizing Software Changes. IEEE Transactions on Software Engineering 28(4), 396–412.

Electronic Arts 2003. Jamdat Bowling 2. Saatavilla www-muodossa <http://www.eamobile.com/Web/Catalog/US/en/game/mobile/ProductDetailOverviewView/genre-1082666698714/product-23104>

Electronic Arts 2007. The Simpsons: Minutes to Meltdown. Saatavilla www-muodossa <http://www.eamobile.com/Web/Catalog/US/en/game/mobile/ProductDetailOverviewView/product-26222>

Estublier J., Clemm G., Leblang D., Tichy W., Van Der Hoek A., Conradi R. & Wiborg-Weber D. 2005. Impact of Software Engineering Research on the Practice of Software Configuration Management. ACM Transactions on Software Engineering and Methodology 14(4), 383–430.

Forte G. 1997. Managing Change for Rapid Development. IEEE Software 24(6), 120–122.

Fraternali P. & Paolini P. 2000. Model-driven Development of Web Applications: The AutoWeb System. Transactions on Information Systems 18(4), 323–382.

Fritsch T., Ritter H. & Schiller J. 2005. Can Mobile Gaming Be Improved? Teoksessa Cheok A.D. & Ishibashi Y. (Toim.) Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games, Singapore, October 30–31. New York: ACM Press, 1–4.

Funk J.L. 2004. Mobile Disruption: The Technologies and Applications Driving the Mobile Internet. New Jersey: John Wiley & Sons Inc.

Hall L., Gordon A., James R. & Newall L. 2004. A Lightweight Rule-based Al Engine for Mobile Games. Teoksessa Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, Singapore, June 3–5. New York: ACM Press, 284–289.

Harjani D-R. & Queille J-P. 1992. A Process Model for the Maintenance of Large Space Systems Software. Teoksessa Conference on Software Maintenance, Orlando, Florida, United States, November 9–12. Los Alamitos: IEEE Computer Society Press, 127–136.

Hirsjärvi S. & Hurme H. 2000. Tutkimushaastattelu: Teemahaastattelun Teoria ja Käytäntö. Helsinki: Yliopistopaino.

Huebscher M., Price N. & Dulay N. 2006. Issues in Developing Ubicomp Applications on Symbian Phones. Teoksessa International Workshop on System Support for Future Mobile Computing Applications, Orange County, California, United States, September 17. Los Alamitos: IEEE Computer Society Press, 51–56.

Humphrey W. 1989. Managing the Software Process. Massachusetts: Addison-Wesley publishing company.

IEEE 1983. IEEE Standard for Software Configuration Management Plans. ANSI/IEEE Standard 828. New York: IEEE.

IEEE 1988. IEEE Guide to Software Configuration Management. ANSI/IEEE Standard 1042. New York: IEEE.

IEEE 1990. IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard 610.12. New York: IEEE.

IEEE 1997. IEEE Standard for Software Reviews. ANSI/IEEE Standard 1028. New York: IEEE.

IEEE 1998. IEEE Standard for Software Maintenance. ANSI/IEEE Standard 1219. New York: IEEE.

Introna L.D. & Edgar A.W. 1997. Against Method-ism: Exploring the Limits of Method. Information Technology & People 10(1), 31–45.

ISO 1995. SABS ISO/IEC 12207:1995. Software Life Cycle Processes.

ISO 1999. Software Engineering – Software Maintenance. ISO/IEC 14764:1999. Geneva: International Standards Organization.

ITIL 2005. Service Support (CCTA): Part 15. Norwich: Central Computer & Telecommunications Agency.

Joeris G. 1997. Change Management Needs Integrated Process and Configuration Management. Teoksessa Jazayeri M. & Schauer H. (Toim.) Proceedings of the 6th European Conference Held Jointly With the 5th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Zurich, Switzerland, September 22–25. SIGSOFT: ACM Special Interest Group on Software Engineering, 125–141.

Kitchenham B.A., Travassos G.H., Mayrhauser A., Niessink F., Scheidewind N.F., Singer J., Takada S., Vehvilainen R. & Yang H. 1999. Towards an Ontology of Software Maintenance. Journal of Software Maintenance: Research and Practice. 11(6), 365–389.

Korel B, Wedde H, Nagaraj S., Nawaz K., Dayana V., Santhanam B. & Xu M. 1991. Version Management in Distributed Network Environment. Teoksessa Feiler P.H. (Toim.) Proceedings of the 3rd International Workshop on Software Configuration Management, Trondheim, Norway, June 12–14. SIGSOFT: ACM Special Interest Group on Software Engineering, 161–166.

Mathiassen L. 1998. Reflective Systems Development. Scandinavian Journal of Information Systems. 10(1–2), 67–117.

May P. 2001. Mobile Commerce: Opportunities, Applications, and Technologies of Wireless Business. Cambridge, United Kingdom: Cambridge university press.

McNurlin B.C. & Sprague R.H. 2002. 5. uusittu painos. Information Systems Management in Practice. New Jersey: Pearson Education Inc.

Nunamaker J.F. Jr., Chen J.M. & Purdin T.D.M. 1991. Systems Development in Information Systems Research. Journal of Management Information Systems, 7(3), 631–640.

Reichenberger C. 1989. Orthogonal Version Management. Teoksessa Richard N.T. (Toim.) Proceedings of the 2nd International Workshop on Software Configuration Management, Princeton, New Jersey, United States, October 24–27. SIGSOFT: ACM Special Interest Group on Software Engineering, 137–140.

Sachweh S. & Schafer W. 1995. Version Management for Tightly Integrated Software Engineering Environments. Teoksessa Verrall M.S.(Toim.) Proceedings of the 1995 Software Engineering Environment Conferences, Noordwijkerhout, Holland, April 5–7. Washington: IEEE Computer Society Press, 21–31.

Seaman C.B. 1993. Organization and Process Together. Teoksessa Gawman A., Kidd E. & Larson P-Å. (Toim.) Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative Research, Toronto, Canada, October 24–28. Toronto: IBM Press 314–324.

Sirén F. 2004. THQ Wireless “Mobile Distribution Publishers View”. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos.

Son J.O.B & Tan B.C.Y. 2008. Mobile Gaming. Communications of the ACM, 51(3), 35–39.

Swanson E.B. 1976. The Dimensions of Maintenance. Teoksessa Yeh R.T. & Ramamoorthy C.V. (Toim.) Proceedings of the 2nd International Conference on Software Engineering, San Francisco, California, United States, October 13–15. Los Alamitos: IEEE Computer Society Press, 492–497.

Tichy W.F. 1982. Design, Implementation, and Evaluation of a Revision Control System. Teoksessa Ohno Y., Basili V., Enomoto H., Kobayashi K. & Yeh R.T. (Toim.) Proceedings of the 6th International Conference on Software Engineering, Tokyo, Japan, September 13–16. Los Alamitos: IEEE Computer Society Press, 58–67.

Tichy W.F. 1985. RCS – A System for Version Control. Software – Practice & Experience 15(7), 637–654.

Tichy W.F. 1988. Tools for Software Configuration Management. Teoksessa Winkler J.F.H. & Verlag E.T. (Toim.) Proceedings of the International Workshop on Software Version and Configuration Control, Grassau, Germany, January 27–29. Grassau: Teubner Verlag 1–20.

Tuohey W.G. 2002. Benefits and Effective Application of Software Engineering Standards. Software Quality Control 10(1), 47–68.

Westfechtel B., Bjorn P. & Conradi R. 2001. A Layered Architecture for Uniform Version Management. IEEE Transactions on Software Engineering 27(12), 1111–1133.

Williams C. & Burge M. 2004. MIDP 2.0 Changing the Face of J2ME Gaming. Teoksessa Yoo S-M. & Etzkorn L.H. (Toim.) Proceedings of the 42nd Annual Southeast Regional Conference, Huntsville, Alabama, April 2–3. New York: ACM Press, 37–41.

LIITTEET

Liite 1. Haastattelupyyntö

Haastattelupyyntö

Hei

Teen gradua Jyväskylän yliopistossa informaatioteknologian tiedekunnassa. Graduni aihe on Ohjelmistojen konfiguraation- ja versionhallinnan ongelmat ja ratkaisut mobiilipelituotannossa.

Haluaisin haastatella henkilöä, joka tietää yrityksenne organisaatiorakenteen ja käytettävät tukiprosessit, näistä eritoten konfiguraation- ja versionhallinnan. Haastattelu koostuu tässä yhteydenottoviestissä mukana tulleesta esikyselylomakkeesta ja haastatteluosiosta. Esikyselylomakkeeseen vastaaminen vie n. 2–8 minuuttia ja haastattelun suorittaminen noin tunnin. Vaihtoehtoisesti haastattelu voidaan suorittaa myös lomakekyselynä Jyväskylän yliopiston Korppijärjestelmässä SSL-suojatulla yhteydellä, ellei teiltä löydy haastatteluun vapaata aikaa.

Miksi olen valinnut juuri mobiilipelituotannon? Peliteollisuus toimialana kiinnostaa, kohdelaitteistojen nopea kehitys ja jatkuvasti laajenevat/monimutkaistuvat pelit tuottavat yhdessä mielenkiintoisen tutkimuskohteen.

Haastattelun tavoitteet:

- Saada hyvä kokonaiskuva mobiilipeliteollisuudesta pelien tuottajan näkökulmasta esikyselylomakkeen avulla.
- Haastattelulla kartoittaa konfiguraation- ja versionhallintaa koskevat haasteet toimialallanne.
- Tarjota perinteiseen ohjelmistotuotantoon suunniteltuja ratkaisumalleja mobiilipelituotannon tueksi, tutkielman valmistuttua.

Haastattelun edut:

Tehokkaan konfiguraation- ja versionhallinnan avulla myös yrityksenne mobiilipelituotanto voi huomattavasti tehostua.

- Haastattelu vie teiltä aikaa vain tunnin, ei maksa mitään ja tutkimusta tekee aiheeseen omistautunut henkilö.
- Tutkimuksen valmistuttua voimme halutessanne sopia uuden tapaamisen, jossa käymme läpi tutkimuksen tulokset perusteellisesti.
- Yritystänne koskevat tiedot säilyvät anonyymeinä ja vastaukset käsitellään luottamuksellisesti.

Gradun valmiusaste:

Kirjallisuuskatsaus on suoritettu ja vuorossa on suunnittelemani viitekehyksen soveltaminen mobiilipelituotantoon, joten haastatteluiden pohjalta saadaan tulokset analysoitua nopeasti. Gradu on tarkoitus saada valmiiksi huhtikuun loppuun mennessä.

Aikataulu:

Esikyselylomakkeen kysymyksiin tarvitsen vastaukset viimeistään 3.3.2008 mennessä. Haastattelu voidaan suorittaa minä tahansa päivänä 10.3–28.3.2008 välisenä aikana tai lomakekyselynä, jolloin vastaukset viimeistään 28.3.2008 mennessä.

Mitä Teidän tarvitsee tehdä:

Vastata liitetiedoston kysymyksiin ja lähettää se vastauksineen takaisin minulle (vie aikaa alle 10 minuuttia). Mikäli kuitenkin haluatte kieltäytyä haastattelusta, toivoisin että lähetätte kieltäytymisen sähköpostitse.

Terveisin,

Matti Kanervo

p. 050 - 3315 112

Email: maolkane@cc.jyu.fi

Informaatioteknologian tiedekunta

Jyväskylä Yliopisto

Liite 2. Ohjelmistojen kehityksen ensisijaiset prosessit

TAULUKKO 8. Ohjelmistojen kehityksen ensisijaiset prosessit (ISO 1995, 8)

Hankintaprosessit	**Toimitusprosessit**	**Kehittämis-prosessit**	**Operaatio-prosessit**	**Ylläpitämis-prosessit**
Eivät oleellisia ohjelmistojen kehittämisen kannalta	Eivät oleellisia ohjelmistojen kehittämisen kannalta	Prosessin toimeenpano Ohjelmiston vaatimusten analyysi Ohjelmiston arkkitehtuurin suunnittelu Ohjelmiston yksityiskohtainen suunnittelu Ohjelmiston koodaaminen ja testaaminen Ohjelmiston integrointi Ohjelmiston kvalifikaation testaus Ohjelmiston integraatio järjestelmään Ohjelmiston asennus Ohjelmiston vastaanoton tukeminen	Eivät oleellisia ohjelmistojen kehittämisen kannalta	Eivät oleellisia ohjelmistojen kehittämisen kannalta

Liite 3. Ohjelmistojen kehityksen tukiprosessit

Lista ohjelmistojen kehityksen tukiprosesseista (Conradi & Westfechtel 1998; ISO 1995, 9)

- Dokumentaatioprosessi
- Konfiguraationhallintaprosessi
- Laadunvarmistusprosessi
- Verifikaatioprosessi
- Validointiprosessi
- Katselmusprosessi
- Auditointiprosessi
- Ongelmanhallintaprosessi
- Versionhallintaprosessi

Liite 4. Ohjelmistojen kehityksen organisatoriset prosessit

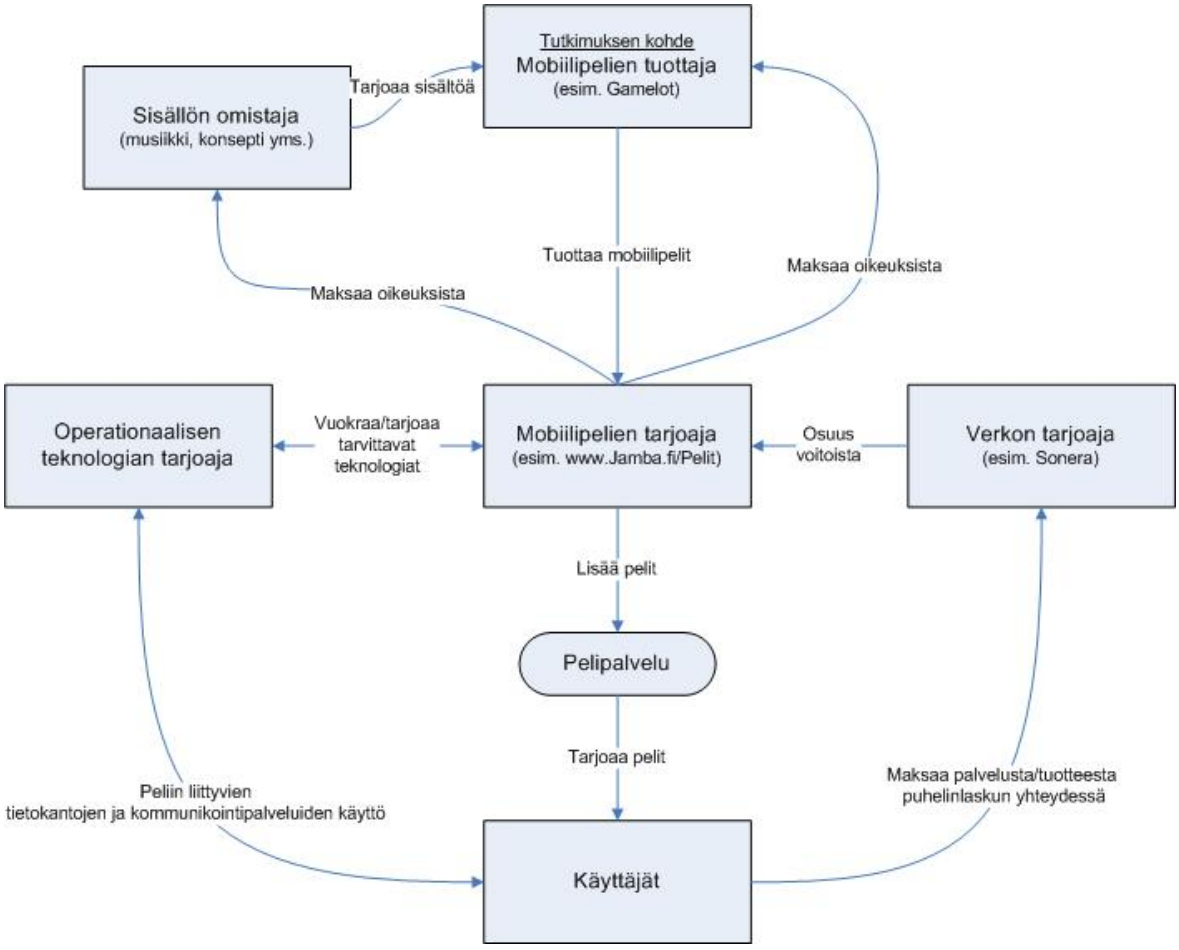
Lista ohjelmistojen kehityksen tukiprosesseista (ISO 1995, 10)

- Hallinnollisprosessi
- Infrastruktuurin ylläpitämisprosessi
- Prosessien kehittämisprosessi
- Koulutusprosessi

Liite 5. Empiirisen tutkimuksen osion 1 ja 2 lomakekysely

Kysymyksiin voitte vastata joko tämän lomakkeen vastauksiin rajatuille alueille tai suoraan sähköpostivastaukseen. Esikyselylomakkeeseen kuuluu 4 kysymystä ja lopussa olevat konfiguraation- ja versionhallinta termien määritykset.

1. Näetkö mobiilipeliteollisuuden rakenteessa eroavaisuuksia verrattuna alla olevaan kirjallisuuden määrittelemään rakenteeseen?



Mobiilipeliteollisuuden rakenne (Perustuen: A Mobile Gaming Platform for the IMS ja Mobile Disruption: The Technologies and Applications Driving the Mobile Internet)

Vastaus:

2. Erotetaan mobiilipelituotannossa mobiilipelien kehitys ja ylläpito toisistaan? Jos erotetaan niin millä tasolla, eri henkilöt/tiimit, erilailla määritetyt prosessit?

Vastaus:

3. Suoritetaanko mobiilipelituotannossa konfiguraation[1]- ja/tai versionhallintaa[2] mobiilipelien kehityksen tai ylläpidon tukena?

Vastaus:

4. Teemahaastattelun ajankohta, aikahaarukasta 10.3–28.3 ja tarkempi tapaamispaikka tai vaihtoehtoisesti lomakekyselynä suojatun yhteyden avulla?

Vastaus:

[1] Konfiguraatio esittää tiettyyn ympäristöön tai käyttöön tarkoitetun ohjelmiston osat ja niistä koostuvan rakenteen. Konfiguraationhallinta taas on prosessi ohjelmiston konfiguraation tunnistamiseen tietyssä pisteessä ohjelmiston elinkaarta. Konfiguraationhallinnan tarkoituksena on systemaattisesti kontrolloida muutoksia konfiguraatioon ja ylläpitää konfiguraation eheys ja jäljitettävyys läpi ohjelmiston elinkaaren. (Guide to the Software Engineering Body of Knowledge: 2004 Version – SWEBOK)

[2] Elinkaarensa aikana monimutkaiseen ohjelmistotuotteeseen liittyvät ohjelmisto-objektit, kuten vaatimusten määritykset, suunnittelumallit, ohjelmiston lähdekoodi, dokumentaatiot, ja testidata kehittyvät moniksi versioiksi. Perustuen Sachwehin & Schaferin (1995), Conradin & Westfechtelin (1998) ja Reichenbergerin (1989) tekemiin määritelmiin, versionhallinnan keskeinen aspekti on aikaulottuvuuden seuranta ohjelmistotuotteen tai ohjelmisto-objektin kehityksessä. Tämä tarkoittaa sitä, että tarvittaessa voidaan palata tiettyyn ajalliseen pisteeseen ja tarkastella sen hetkistä ohjelmistotuotteen tilaa konfiguraationhallinnan avulla.

Liite 6. Empiirisen tutkimuksen osion 3 lomakekysely

Kyselyiden hallinta -sivulle

Kyselyn esikatselu

Konfiguraation- ja versionhallinnan ongelmat ja ratkaisut mobiilipeliteollisuudessa sivu (1/5)

Johdanto

Tutkielman avulla pyritään kartoittamaan, mitkä perinteisessä ohjelmistotuotannossa yleisimmin esiintyvistä ongelmista esiintyy myös mobiilipelituotannossa. Lisäksi tutkielman tarkoituksena on selvittää valmiiden ratkaisumallien sopivuutta teoriatasolla mobiilipelituotantoon. Aiheesta johtuen voi kysymysten kokonaiskuva antaa liian negatiivisen vaikutuksen alastanne, tämä ei ole kuitenkaan tarkoitus eikä tavoite. Kysely koostuu kolmesta osiosta. Ensimmäinen käsittelee konfiguraationhallinnan teknistä puolta ja toinen konfiguraationhallinnan hallinnollista puolta. Lopuksi kolmas osio käsittelee versionhallintaa.
Tämän kyselyn kysymyksiin voitte vastata joko kaikkiin yhdellä kertaa, mikä on toivottavaa, tai osa kysymyksistä kerrallaan. Jos ette ehdi vastata kaikkiin kysymyksiin yhdellä kertaa, niin painakaa session lopetusvaiheessa tallennusnappia sivun alalaidassa. Seuraavalla vastauskerralla entiset vastauskentät ovat tyhjiä, mutta vastaukset ovat kuitenkin tallessa. Tästä johtuen sellaiset kysymykset joihin olette jo vastanneet, voitte jättää väliin.
Kysymyksiin vastataan Kyllä ja Ei vaihtoehdoilla. Kaikkiin kysymyksiin on kuitenkin lisätty myös tekstipohjaiset vastauskentät, joihin olisi toivottavaa saada Kyllä ja Ei vastauksiin perusteluita tai muita huomion arvoisia kommentteja. Suurimpiin ongelma-alueisiin olen lisännyt tieteellisten lähteiden ratkaisumallin, tosin tilan ja teidän lukuajan säästämiseksi nämä ratkaisumallit ovat tiivistetyssä muodossa. Tutkielman valmistuttua voin toimittaa teille täydet kuvaukset näistä ratkaisumalleista.

Kyselyn kannalta keskeisimpiä termejä

Revisiolla tarkoitetaan tässä yhteydessä versiota, jonka tarkoitus on korvata edeltäjänsä.

Variantilla tarkoitetaan tässä yhteydessä versiota, jonka tarkoitus on toimia rinnakkain, ei syrjäyttää edeltäjäänsä.

Ohjelmisto-objektilla tarkoitetaan tässä yhteydessä mitä tahansa ohjelmistoon liittyvää tekijää, joka voidaan laittaa versionhallinnan alaisuuteen. Täten ohjelmisto-objekti voi sisältää ohjelmiston lähdekoodia, dokumentaatiota tai esimerkiksi testidataa.

Konfiguraatio ja konfiguraationhallinta. Konfiguraatio esittää tiettyyn ympäristöön tai käyttöön tarkoitetun ohjelmiston osat ja niistä koostuvan rakenteen. Konfiguraationhallinta taas on prosessi ohjelmiston konfiguraation tunnistamiseen tietyssä pisteessä ohjelmiston elinkaarta. Konfiguraationhallinnan tarkoituksena on systemaattisesti kontrolloida muutoksia konfiguraatioon ja ylläpitää konfiguraation eheys ja jäljitettävyys läpi ohjelmiston elinkaaren.

Versionhallinta. Elinkaarensa aikana monimutkaiseen ohjelmistotuotteeseen liittyvät ohjelmisto-objektit, kuten vaatimusten määritykset, suunnittelumallit, ohjelmiston lähdekoodi, dokumentaatiot, ja testidata kehittyvät moniksi versioiksi. Versionhallinnan keskeinen aspekti on aikaulottuvuuden seuranta ohjelmistotuotteen tai ohjelmisto-objektin kehityksessä. Tämä tarkoittaa sitä, että tarvittaessa voidaan palata tiettyyn ajalliseen pisteeseen ja tarkastella sen hetkistä ohjelmistotuotteen tilaa konfiguraationhallinnan avulla

Näytä seuraava sivu

Kyselyiden hallinta -sivulle

Kyselyn esikatselu

Konfiguraation- ja versionhallinnan ongelmat ja ratkaisut mobiilipeliteollisuudessa sivu (2/5)

Konfiguraationhallinta, tekninen puoli

Tekninen puoli pitää sisällään mekanismit ja funktiot, joiden avulla hallitaan ohjelmisto-objektien konfiguraatioita. Näin saadaan rakennettua polveutuvat versiot ja yhtenäiset konfiguraatiot. Loppuun on lisätty yksi tieteellisten lähteiden tarjoamista ratkaisumalleista konfiguraationhallinnan teknisen puolen ongelmiin. Tämän ratkaisumallin sopivuutta mobiilipeliteollisuuteen voitte kommentoida ratkaisumallin jälkeiseen tekstikenttään.

1. Alalla on esiintynyt ongelmia suurikokoisten mobiilipelien rakenteen hallinnassa?

	Kyllä	Ei
Vastaus	☐	☐

Perustelua tai kommentteja

2. Tukevatko alalla käytössä olevat työkalut konfiguraationhallinnan prosesseja?

Näitä prosesseja ovat prosessinhallinta, muutoksenhallinta, tilanvalvonta, laadunvalvonta, konfiguraatioiden tunnistaminen ja jakelunhallinta.

	Kyllä	Ei
Vastaus	☐	☐

Perustelua tai kommentteja

3. Toimivatko mobiilipelien kehittäjän/ylläpitäjän käyttämät ohjelmistot loogisena kokonaisuutena?

	Kyllä	Ei
Vastaus	☐	☐

Perustelua tai kommentteja

4. Onko mobiilipelien kehitys- ja ylläpitoprosessi systemaattinen/jäljitettävä?

	Kyllä	Ei
Vastaus		

Perustelua tai kommentteja

5. Ohjelmisto-objektien tunnistamisessa, tallentamisessa ja käyttämisessä esiintyy ongelmia?

Kyllä Ei

Vastaus

Perustelua tai kommentteja

6. On huomattu ongelmia mobiilipelin tietyn konfiguraation jälkikäteen koostamisessa?

Kyllä Ei

Vastaus

Perustelua tai kommentteja

7. Mobiilipelien tuottamisessa on ilmennyt ongelmia tehtyjen/tehtävien muutosten vaikutuksien ymmärtämisessä?

Kyllä Ei

Vastaus

Perustelua tai kommentteja

8. Onko monimutkaisten mobiilipelien tuottamisessa esiintynyt ongelmia ryhmä-/yksilötyössä?

Mm. konfiguraationhallinta muodostuu tällöin liian raskaaksi

Kyllä Ei

Vastaus

Perustelua tai kommentteja

Tieteellisten lähteiden tarjoama ratkaisu yllä mainittuihin ongelma-alueisiin:

Järjestelmätuki konfiguraationhallintajärjestelmän avulla. Järjestelmätuki tarkoittaa sitä, että konfiguraationhallintajärjestelmä antaa yhtenäisen työpöydän, jossa kehittäjä/ylläpitäjä voi tehdä tarvittavat tehtävät käyttäen joko konfiguraationhallintajärjestelmän ominaisuuksia tai ulkoisia työkaluja. Lisäksi esimerkiksi hajautetuissa järjestelmissä ei tarvitse keskittyä siihen, kenellä on uusin versio jostakin tietystä ohjelmisto-objektista tai miten muokattavana ollut ohjelmiston-objekti palautetaan takaisin konfiguraationhallintajärjestelmään työn jälkeen, vaan järjestelmä hoitaa sen viemättä liikaa ohjelmistotuotantoon varattua aikaa.

Kommentit

Näytä edellinen sivu | Näytä seuraava sivu

Kyselyiden hallinta -sivulle

Kyselyn esikatselu

Konfiguraation- ja versionhallinnan ongelmat ja ratkaisut mobiilipeliteollisuudessa sivu (3/5)

Konfiguraationhallinta, hallinnollinen puoli

Hallinnollinen puoli taas keskittyy konfiguraationhallinnan organisatorisiin ja hallinnollisiin näkökulmiin. IEEE standardien määritelmä konfiguraationhallinnan hallinnollisesta puolesta sisältää ohjelmisto-objektien ja niiden versioiden tunnistamisen, muutoksenhallinnan, tilanvalvonnan ja laadunvalvonnan.

Tässä osiossa jokaisen kysymyksen perään on lisätty tieteellisten lähteiden esittämät ratkaisut kyseisiin ongelmiin ohjelmistotuotannossa. Kyseisten ratkaisumallien sopivuutta mobiilipelituotantoon voitte kommentoida ratkaisumallin jälkeiseen tekstikenttään.

1. Alalla on esiintynyt ongelmia konfiguraatioiden tunnistamisessa tai ohjelmisto-objektien välisten suhteiden tunnistamisessa?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Konfiguraatioiden tunnistamiseen kuuluu konfiguraationhallinnan alaisten ohjelmisto-objektien määrittäminen ja konfiguraationhallintatietokannan käyttö. Konfiguraatioiden tunnistamisen apuna toimii peruslinja. Peruslinjalla tarkoitetaan ohjelmistotuotteesta tietyssä pisteessä otettua konfiguraatiota, joka esittää sekä konfiguraation rakenteen että tarvittavat yksityiskohdat.

Kommentit

2. Onko mobiilipelien kehittäjillä/ylläpitäjillä konfiguraationhallinnan tarkkuustaso selvillä/yhtenäinen?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Konfiguraationhallinnan rajoitteet ja ohjeistukset määrittelevät mm. sen millä tarkkuustasolla konfiguraationhallintaa suoritetaan.

Kommentit

3. Onko mobiilipelien tuottajaorganisaatioissa esiintynyt ongelmia vastuualueissa, resursseissa, aikatauluissa, työkaluissa yms.?

Kyllä Ei

Vastaus ☐ ☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Konfiguraationhallinnan suunnitteleminen sisältää vastuualueiden määrittämisen, resurssien ja aikataulujen kartoittamisen ja työkalujen valinnan. Tämän tuotoksena toimii konfiguraationhallinnan suunnitteludokumentti.

Kommentit

4. Ovatko mobiilipeleihin tehdyt muutokset kontrolloituja?

Toisin sanoen muutosten vaikutukset ja kustannukset pystytään ennakoimaan?

Kyllä Ei

Vastaus ☐ ☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Muutospyyntöjenhallinta, -arviointi ja –hyväksyntä. Lisäksi muutoksista luopumisen- ja poikkeamisenhallinta. Laadukkaan konfiguraationhallinnan tuloksena muutosten vaikutukset pystytään ennakoimaan ja kustannukset arvioimaan tehokkaasti, kyseiset tekijät vaikuttavat osaltaan ohjelmistotuotannon tehokkuuteen.

Kommentit

5. Useat yhtäaikaiset mobiilipelien kehitys- ja ylläpitoprojektit aiheuttavat ongelmia?

Kyllä Ei

Vastaus ☐ ☐

Perusteluja

tai kommentteja

Tieteelliset lähteet: Laadukkaan konfiguraationhallinnan avulla muutokset saadaan toteutettua oikeisiin versioihin ja peruslinjoihin tehokkaasti.

Kommentit

6. Esiintyykö mobiilipelien tuotannossa jokin seuraavista ongelmista?

Konfiguraatioista ei saada tarpeeksi tietoa, konfiguraatiot eivät ole määritellyssä tilassa tai konfiguraationhallinta ei ole tehokasta.

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Tilanvalvonta, vastuullaan tietojen kerääminen ja raportointi. Tietojen kerääminen hoidetaan yleensä konfiguraationhallintajärjestelmän avulla automaattisesti, joten kyseinen prosessi säilyy näkymättömänä ohjelmistojen kehitykselle ja ylläpidolle. Raportoinnista on hyötyä useille eri organisaation tehtäväalueille, esimerkiksi projektin johdolle ja ohjelmistotuotantotiimeille.. Raporteista projektin johto voi valvoa mm. ohjelmisto-objektien kokemien muutosten määrää ja muutoksiin tarvittua aikaa.

Kommentit

7. Ongelmia saavuttaa suunniteltu/määritelty mobiilipeli?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Konfiguraationhallinnassa laadunvalvonnan tarkoitus on valvoa fyysisten ja funktionaalisten ominaisuuksien toteutumista ennalta määritellyllä tasolla. Ensiksi mainitussa huomio kohdistuu suunnittelu- ja viitedokumentaation yhdenmukaisuuteen ohjelmistotuotteen kanssa. Jälkimmäisessä valvotaan ohjelmisto-objektien ja spesifikaatioiden yhdenmukaisuutta. Esimerkiksi peruslinjojen tulisi läpäistä sekä fyysinen että funktionaalinen konfiguraation valvonta.

Kommentit

8. Ongelmia aikaisemmin määritetyn konfiguraation rakentamisessa (hallinnollinen näkökulma)?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Kyseisen prosessin tekninen toteutus kuuluu konfiguraationhallinnan tekniseen puoleen, hallinnolliseen puoleen kuuluu puolestaan rakentamisohjeiden luominen ja valvonta, sekä rakentamisprosessin toimivuuden valvonta (esimerkiksi pystytään rakentamaan ohjelmistotuotteesta aikaisemmin julkaistu konfiguraatio täysin identtisenä kokonaisuutena).

Kommentit

Näytä edellinen sivu | Näytä seuraava sivu

Kyselyiden hallinta -sivulle

Kyselyn esikatselu

Konfiguraation- ja versionhallinnan ongelmat ja ratkaisut mobiilipeliteollisuudessa sivu (4/5)

Versionhallinta

Tässä osiossa tarkkaillaan versionhallintaan liittyviä osa-alueita. Suurin osa kysymyksistä on hallinnolliseen näkökulmaan painottuvia. Kysymykset koskevat mm. versiointityyppejä, revisiopuuta, revisioluokkia ja versiokirjaston käyttöä/suunnittelua.

Jokaisen kysymyksen perään on lisätty tieteellisten lähteiden esittämät ratkaisut kyseisiin ongelmiin ohjelmistotuotannossa. Kyseisten ratkaisumallien sopivuutta mobiilipelituotantoon voitte kommentoida ratkaisumallin jälkeiseen tekstikenttään.

1. Onko alalla keinoja pitää monista versioista (revisioista ja varianteista) koostuva mobiilipeli/ohjelmisto-objekti hyvin organisoituna?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

2. Mobiilipeleistä/ohjelmisto-objekteista kehittyy usein liian monta eri revisiota tai varianttia?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Oikeanlaisen versioinnin (esim. kokonaisvaltaisen, tuote- tai komponenttipohjaisen versioinnin) valitseminen. Näiden avulla voidaan keskittyä oleelliseen versiointitasoon riippuen tuotettavasta ohjelmistotuotteesta. Ts. onko oleellista keskittyä ohjelmisto-objektien versiointiin vai kokonaisten ohjelmistotuotteiden versiointiin.

Kommentit

3. Onko esiintynyt ongelmia revisioiden aikaulottuvuuden kanssa?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai

kommentteja

Tieteelliset lähteet: Hyvä tapa esittää ohjelmistotuotteen useat revisiot ja variantit graafisessa muodossa on revisiopuu. Revisiopuun (eli versiopuun) avulla saadaan hyvin esille revisioiden aikaulottuvuus.

Kommentit

4. Käytetäänkö päivitysten kokoluokissa mitään rajauksia?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Reviosioluokkien käyttöönotto. Tärkeä (major) revisio usein syrjäyttää edelliset revisiot ja aloittaa revisionumeroinnin pienten (minor) revisioiden kohdalta alusta. Esimerkiksi revisiot 2.0 ja 3.0 ovat tärkeitä revisioita ja revisiot 2.1 ja 3.1 ovat pieniä revisioita. Lisäksi on olemassa kolmas, pikakorjaus (fix), revisiointiluokka. Kyseisen luokan revisiot sisältävät usein vain pieniä korjauksia. Esimerkiksi revisiot 2.2.1 ja 2.2.2 kuuluvat pikakorjauksiin. Varianttien määrä kannattaa pitää mahdollisimman pienenä, sillä se tekee versionhallinnasta huomattavasti helpompaa ja luotettavampaa.

Kommentit

5. Mobiilipelituotannossa on ongelmia versionhallinnassa ja versioiden säilytyksessä, kun versioiden määrä kasvaa?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Versiokirjastoissa (software library) säilytetään ohjelmisto-objektien ja ohjelmistotuotteiden eri versioita.Versiokirjastoja voi olla useita erityyppisiä (esim. työskentelykirjasto, tukikirjasto ja tuotekirjasto).

Kommentit

6. Ovatko käytössä olevat versiokirjastot sekavia tai heikosti suunniteltuja?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

Tieteelliset lähteet: Versiokirjastojen luokittelu ja 8 huomioitavaa seikkaa versiokirjastoa suunniteltaessa. Fyysinen sijainti, nimeämiskäytännöt, tehtäväalueet, turvallisuusjärjestelyt, versiokirjaston laajuus, vanhojen versioiden säilytysaika,kapasiteettisuunnitelmat ja seurantamenetelmät.

Kommentit

7. Versioiden tarvitseman työmäärän seuraamisessa esiintyy ongelmia tai se ei ole mahdollista?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja

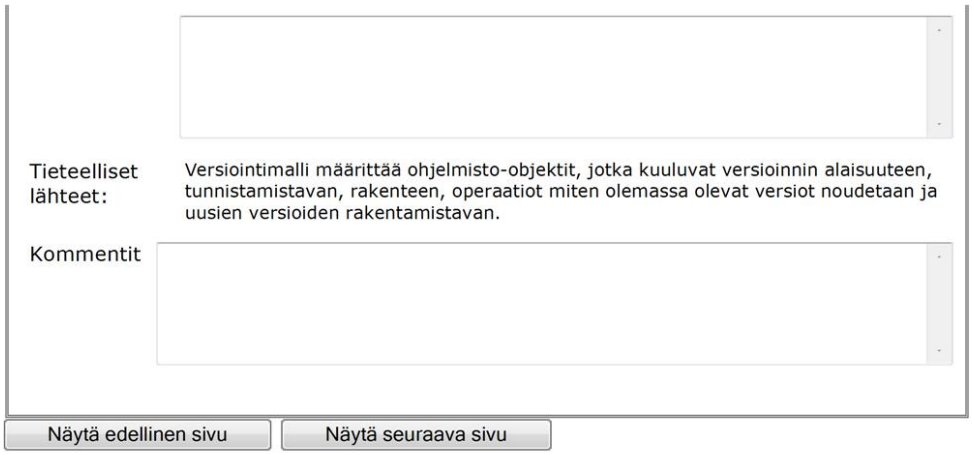
Tieteelliset lähteet: Versioiden kokemien muutosten määrän seuranta. Esim. Eick ym. kuvio ohjelmistotuotteen elinkaaren seurantaan: Jokaisen version kohdalla on kaksi palkkia, ensimmäinen esittää lisättyjen ja toinen poistettujen koodirivien määrän.

Kommentit

8. Mobiilipelien tuottajien (kehittäjien ja ylläpitäjien) suorittama versiointi on satunnaista, tehden versionhallinnan haastavammaksi?

	Kyllä	Ei
Vastaus	☐	☐

Perusteluja tai kommentteja


Tieteelliset lähteet:
Versiointimalli määrittää ohjelmisto-objektit, jotka kuuluvat versioinnin alaisuuteen, tunnistamistavan, rakenteen, operaatiot miten olemassa olevat versiot noudetaan ja uusien versioiden rakentamistavan.
Kommentit
Näytä edellinen sivu
Näytä seuraava sivu

Kyselyiden hallinta -sivulle

Kyselyn esikatselu

Konfiguraation- ja versionhallinnan ongelmat ja ratkaisut mobiilipeliteollisuudessa sivu (5/5)

Kiitokset

Lämpimimmät kiitokset osallistumisestanne tähän tutkielmaan. Halutessanne voin lähettää mm. Gradun empiirisen tutkimuksen yhteenvedon teille sähköpostitse, kun tutkielma valmistuu. Ja vielä viimeinen kysymys:

Haluaisin sähköpostitse
- Koko Pro Gradun
- Tarkennukset vain niiltä ongelma-alueita, jotka näin aiheelliseksi
- Ei mitään kiitos

Näytä edellinen sivu