

Jarmo Rossi

**ENERGIANMITTAUKSEN TIETOJÄRJESTELMÄN
KONFIGURAATION HALLITTAVUUS**



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2018

TIIVISTELMÄ

Rossi, Jarmo

Energianmittauksen tietojärjestelmän konfiguraation hallittavuus

Jyväskylä: Jyväskylän yliopisto, 2018, 58 s.

Tietojärjestelmätiede, pro gradu -tutkielma

Ohjaajat: Seppänen, Ville ja Airaksinen Risto

Energiamittauksen tietojärjestelmään kuuluvan ohjelmistokokoonpanon hallinta on nousemassa tärkeäksi osaksi tutkimuksen kohdeyrityksen ohjelmistotuotteiden hallintaa. Tietojärjestelmien kehityksessä ja niiden elinkaareissa tarvitaan entistä tarkempaa ja luotettavampaa kokoonpanohallintaa. Myös energia-alan murros aiheuttaa tarkentuvia ja muuttuvia vaatimuksia mittausalan ohjelmistoille ja sen konfiguraationhallinnalle. Tutkimuksen tavoitteena on luoda alustava runko yrityksen ohjelmiston konfiguraationhallinnan menetelmälle ja luoda viitekehys yrityksen nykyisten, hajanaisten ohjelmiston konfiguraatiohallintamenetelmien yhdistämiselle. Tutkimus toteutettiin empiirisenä tapaustutkimuksena tekemällä havaintoja kohdeyrityksen tämänhetkisistä ohjelmiston konfiguraationhallintamenetelmistä analysoimalla yrityksessä käytössä olevia ohjelmiston konfiguraatioiden hallintatyökaluja sekä asiantuntijahaastattelulla. Empiirisen tapaustutkimuksen havaintoja verrataan yrityksessä käytössä olevaan CMMI-kypsyysmalliin. Tutkimuksen toinen osan toteutettiin asiantuntijoilta kyselylomakkeelta saadun yksityiskohtaisen tiedon pohjalta. Tämän tutkimuksen tärkein havainto oli CMMI-käytänteistä johdetun ohjelmiston konfiguraationhallinnan menetelmän rungon kelpoisuus energiamittauspalveluita tuottavan yrityksen käyttöön. Jatkokehityksen aiheeksi voidaan ottaa tässä tutkimuksessa esiin tuodut jatkokehityskohteet. Tutkimuksen tulokseksi muodostui selvitys ja runko ohjelmiston konfiguraationhallinnan menetelmästä, jonka perusteella ohjelmiston konfiguraationhallintaa voidaan edelleen kehittää.

Asiasanat: konfiguraationhallinta, energiamittaus, ohjelmiston evoluutio.

ABSTRACT

Rossi, Jarmo

Manageability of energy measurement system's configuration.

Jyväskylä: University of Jyväskylä, 2018, 58 p.

Information systems science, Master's Thesis.

Supervisors: Seppänen, Ville ja Airaksinen Risto

Software configuration management belonging to the energy measurement information system is emerging as an important part of the enterprise software product management. For information systems evolution and their life cycle, there is a need for more accurate and more reliable configuration management. Also, the transformation of energy in the industry causes sharpening and changing demands for measurement software and its configuration management. The aim of the study is to create a preliminary framework for the company's software configuration management method and to create a framework for combining the existing, fragmented configuration management methods of a company. The study was conducted as an empirical case study by observing the company's current software configuration management methods by analyzing the software configuration tools used in the company and by expert interview. The findings of the empirical case study are compared to the CMMI capability model used in the company. The second part of the tutorial was performed by a questionnaire from experts based on the detailed information obtained. The analysis of the configuration software for the configurations is carried out using a design science research method and semi-structured research interviews based on research results. The main observation of this study was the validity of the configuration management method of the software derived from CMMI practices for the use of energy measurement services. The study's artifact was formed by the software configuration management framework and the reference framework for further configuration of the software configuration management.

Keywords: configuration management, configuration item, energy measurement, software evolution.

KUVIOT

KUVIO 1 Ohjelmiston konfiguraationhallinnan kehitys	10
KUVIO 2 DSPR-tutkimusmalli (Design Science Research Process)	19
KUVIO 3 Vesiputousmalli.....	27
KUVIO 4 Energianmittausjärjestelmän kuvaus ja tutkimuksen rajaus	29
KUVIO 5 Tuotteen elinkaarimalli	30
KUVIO 6 Konfiguraationhallinta-alkioiden tunnistaminen (CM.SP 1.1)	34
KUVIO 7 Konfiguraationhallinnan menetelmän luominen (CM.SP 1.2)	35
KUVIO 8 Lähtötilanteen tunnistaminen (CM.SP 1.3)	35
KUVIO 9 Muutospyyntöjen seuranta (CM.SP 2.1)	36
KUVIO 10 Konfiguraationhallinta-alkioiden valvonta (CM.SP 2.2)	36
KUVIO 11 Perusta konfiguraationhallinnan aikakirja (Records) (CM.SP 3.1) ..	37
KUVIO 12 Suorita konfiguraationhallinnan lähtötilanteen auditointi (CM.SP 3.2)	37

TAULUKOT

TAULUKKO 1: CMMI-kypsyysmallin alueet	23
TAULUKKO 2: CMMI-kypsyysmalli ja ohjelmiston konfiguraationhallinnan käytännöt	24

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
KUVIOT	4
TAULUKOT	5
SISÄLLYS.....	6
1 JOHDANTO.....	8
1.1 Tutkimuksen kohdeyritys ja tehtävä	10
1.2 Tutkimuksen tavoitteet ja toteuttaminen	11
2 TIETOJÄRJESTELMÄN KONFIGURAATIONHALLINTA	12
2.1 Konfiguraationhallinta.....	12
2.2 Konfiguraationhallinta osana ohjelmistokehitystä	14
2.3 Konfiguraationhallinnan teemat	15
2.3.1 Teema 1: Muutossarjat ja niiden versiointi (change set)	15
2.3.2 Teema 2: Prosessituki.....	16
2.3.3 Teema 3: Algoritmit: Konfiguraatiotiedostojen vertailu ja yhdistäminen	16
2.3.4 Teema 4: Ohjelmiston konfiguraationhallinnan hajautettu- ja etäkehitys.....	16
2.3.5 Teemojen yhteenveto	17
3 ENERGIANMITTAUSALAN TAPAUSTUTKIMUS.....	18
3.1 Suunnittelutieteellinen tutkimus.....	18
3.2 Aineiston keruu.....	21
3.3 Aineiston analysointi.....	21
3.4 Intensiivinen tapaustutkimus	22
3.5 Kohdeyrityksen kuvaus.....	22
3.5.1 Organisaatio	23
3.5.2 Ohjelmiston konfiguraationhallinta ja CMMI-kypsyysmalli	23
3.5.3 Vesiputousmallinen mittalaitetuotanto	27
3.5.4 Scrum -mallinen ohjelmistotuotanto	28
3.5.5 Järjestelmä ja ohjelmisto	28
3.5.6 Organisaation konfiguraationhallintaprosessit	30
3.5.7 Ehdotettu menetelmärunko ja analyysi	37
4 DEMONSTRAATIO JA MENETELMÄRUNGON ARVIOINTI.....	42

4.1	Menetelmärungon demonstraatio ja versionhallinnan prosessin testaus.....	42
4.2	Arviointi.....	43
5	YHTEENVETO JA JOHTOPÄÄTÖKSET	46
	LÄHTEET	48
	LIITE 1 ASiantuntijoiden haastattelupohja	52
	LIITE 2 KYSELYLOMAKE	55
	LIITE 3 OHJELMISTON KONFIGURAATIOTIEDOSTOJEN HALLINTAPROSESSI	58

1 JOHDANTO

Kokonaisuutena konfiguraationhallinta on prosessi, joka tarjoaa yhteyden ohjelmistokehityksen vaiheiden, kuten suunnittelun, ohjelmoinnin, rakentamisen, testaamisen ja ohjelmiston jakelujen välille. Konfiguraationhallinnan vastuu voi kohdistua yksittäiselle ryhmälle tai johdon varmistamalle ja nimeämälle ryhmälle, jotta ohjelmistokehitykseen osallistuvat henkilöt saavat tiedon suunnitelluista, ohjelmoiduista, testatuista ja jaelluista ohjelmistoista. Lopulta näiden samat henkilöiden ja loppukäyttäjien pitää tietää mitä on rakennettu, testattu ja toimitettu, jotta ohjelmistoja käytetään tarkoituksenmukaisesti sekä tuetaan ja ylläpidetään loppukäyttäjän ympäristössä oikein. (Berlack, 2002)

Ohjelmiston tuotekehitys koostuu yleensä monimutkaisista, toisiinsa liittyvistä vaatimuksista, tuotekehitysmalleista, testitapauksista, lähdekoodista, skriptien laatimisesta ja muista tietoteknisistä rakennelmista. Nämä artefaktit rakennetaan kehittäjien ja testaajien välisen kanssakäymisen avulla, monesti käyttäen tiedonjakeluarkistoa. Huolimatta saaduista eduista, joita yleisesti ohjelmiston konfigurointihallinnan työkalut tarjoavat artefaktien versiointiin, nykyiset lähestymistavat eivät ota huomioon sen tarjoamia hyötyjä artefaktien suhteista toisiinsa. Tämä johtuu pääasiassa siitä, että nämä työkalut ovat edelleen rajoitettuja ja lähdekoodin kehitys ja kääntäminen ohjaavat niitä (Hoek ym., 1995). Edelleen Hoekin ym. mukaan ohjelmiston konfiguraationhallintajärjestelmän pitää tukea artefaktien suhteita, niiden eri versioiden välillä. Artefaktien elementtien tallennus- ja noutomenetelmään liittyvän sopivan arkkitehtuurin puute estää näiden elementtien vertaamista toisiinsa ja rajoittaa tietoja, tietämyksen edelleen käyttöä ja jakamista. Tähän arkkitehtuurin puutteeseen voidaan hakea ratkaisua tutkittavalla ohjelmiston konfiguraationhallinnalla.

Viime vuosina kasvava ymmärrys ohjelmistokehityksestä kokoelmana toisiinsa liittyviä prosesseja on vaikuttanut työskentelyyn konfiguraationhallinnan parissa. Tästä johtuen ohjelmiston konfiguraationhallintaa tarkastellaan myös prosessien näkökulmasta.

Ohjelmiston konfiguraationhallinta on tärkeässä roolissa nykyisten ohjelmiston elinkaarten ja evoluutioiden hallinnassa. Nykyisen ohjelmisto ovat monesti liitoksissa elektronisiin laitteisiin, mekaniikkaan ja palveluihin. Energianmittausalan

ohjelmiston elinkaaret ovat pitkiä ja elinkaaren aikaisen muutokset ovat hyvin todennäköisiä. Kohdeyrityksessä havaitut ohjelmistojen konfiguraationhallinnan menetelmän haasteiden kartoittamisen ja aiheen akateemisella tutkimisen uskotaan tuovan hyötyjä organisaatiolle.

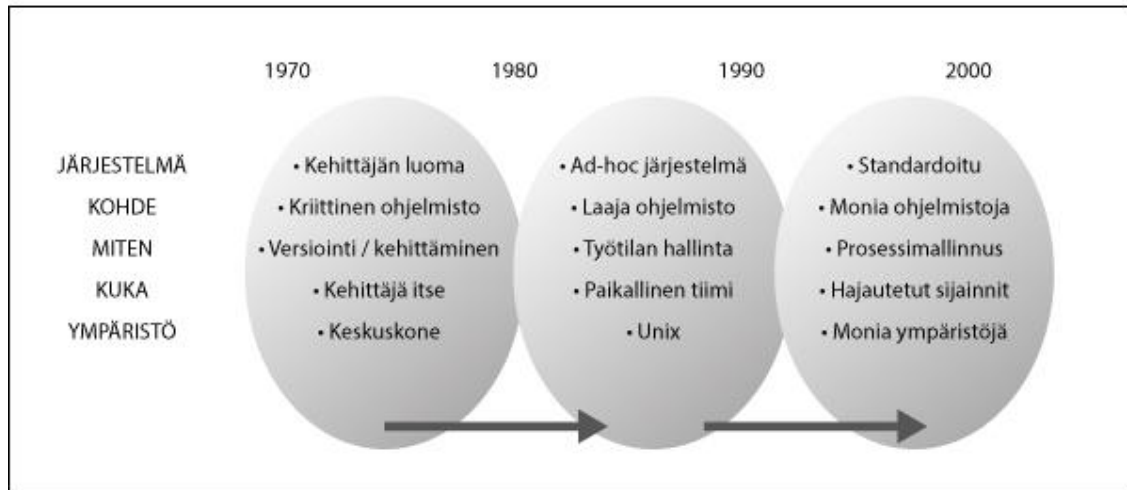
Aiemmin ohjelmistojen konfiguraationhallintatyökaluilla oli rajoitettu toiminnallisuus ja sovellettavuus, mutta modernit ohjelmistojen konfiguraationhallintajärjestelmät tarjoavat kehittyneitä toimintoja, jotka mahdollistavat tehokkaan hallinnan ohjelmistojen kehityksessä esiin nouseville kohteille, joita havaitaan monimiljoonaisen koodirivin järjestelmän kehityksessä. Tämä muutos pienistä, yksinkertaisista työkaluista koko ohjelmistojen konfiguraationhallinnaksi voidaan akateemisten ja teollisten tutkimusten mukaan määritellä vakaaksi tuotekehityksen tukitoiminnoksi. (Estublier, Leblang, Hoek & Conradi, 2005)

Ohjelmiston konfiguraationhallinnan tehtävänä on todentaa ja ylläpitää kaikki tuotteen tunnistetut ominaisuudet ja varmistaa niiden saatavuus tuotteen parissa toimiville osapuolille kaikissa elinkaaren vaiheissa (Laari, 2016).

Konfiguraationhallinta on myös arvoa lisäävää toimintaa, mutta se on lähes poikkeuksetta arvostukseltaan vähäisempää kuin esimerkiksi laadun- tai projektin hallinta. Käytännössä kuitenkin kumpaakaan näistä toimista ei ole mahdollista tehdä ilman konfiguraationhallintaa (Kidd & Burgess 2004).

Energia alan teknologinen murros (Tutkimushanke: Smart Energy Transition, 2016) aiheuttaa energian toimitukselle ja myös energia-alan ohjelmistoille lisääntyvää ja tarkentuvaa konfiguraationhallintaa laajentamalla ohjelmistojen tarvetta.

Kuten Kuviossa 1 on esitetty, ohjelmiston konfiguraationhallinnan järjestelmien kohde on muuttunut merkittävästi. Alussa niitä käytettiin yhden henkilön keskuskoneella hallinomiin kriittisiin ohjelmiin. Tämä vaati toimiakseen versiointia ja kehittämistukea, joka oli tyyppillisesti tuotettu jollain kotitekoisella järjestelmällä. Tämän jälkeen ohjelmiston konfiguraationhallintajärjestelmät muuttuivat pääasiassa tukemaan laaja-alaista kehitystä ja huoltamaan Unix-järjestelmän käyttäjryhmiä. Tämä johti työasemakohtaisen hallinnan tarpeeseen, johon vastattiin nopeasti uudenaikaisella ad hoc -tyylisellä ohjelmiston konfiguraationhallintajärjestelmällä. Tällä hetkellä ohjelmiston konfiguraationhallintajärjestelmät hallitsevat monien eri tyylisten ohjelmien evoluutioita monen eri ihmisen toimesta ja mahdollisesti hajautetuilla sijainneilla, monella eri tyyppisellä tietokoneella. Tämä edellyttää usein selkeitä prosessitukea. (Estublier ym., 2005).



KUVIO 1 - Ohjelmiston konfiguraationhallinnan kehitys (Estublier ym., 2005, 7)

Ohjelmiston konfiguraationhallinnan kirjallisuutta ja tutkimuksia kerättiin järjestelmällisillä Google Scholar ja Aisnet-tietokantahauilla. Hauissa käytettiin hakusanoja *cmmi process improvement, elinkaarenhallinta, energiateollisuus, ketterä ohjelmistokehitys, konfiguraationhallinta, ohjelmistojen konfiguraationhallinta, software configuration management, versionhallinta*. Kirjallisuuden avulla kerättiin perustietoa ohjelmiston konfiguraationhallinnasta.

1.1 Tutkimuksen kohdeyritys ja tehtävä

Energianmittausalalla toimivassa tutkimuksen kohdeyrityksessä ohjelmiston konfiguraationhallinnan tutkiminen on noussut tarpeelliseksi myös yrityksen johdon taholta. Yrityksen ohjelmiston konfiguraationhallinnalle ei tällä hetkellä olemassa kunnollista kirjoitettua menetelmää tai periaatteita, joten ns. lähtötilanteen kartoittaminen ja hallinnan menetelmän kirjaaminen ovat tärkeitä vaiheita. Keskusteluissa asiantuntijoiden ja johdon kanssa olen saanut suurta tukea menetelmän kirjaamiselle. Tutkielman tavoitteena on luoda yritykselle ohjelmiston konfiguraationhallinnan menetelmälle runko.

Energianmittausalan pitkien asiakassuhteiden ja tästä johtuvien ohjelmistojen pitkien evoluutioiden hallinta luo tarpeen ohjelmiston konfiguraationhallinnan tutkimiselle. Edellä mainittujen syiden ja yrityksessä havaitun tuen ja mahdollisen menetelmän tarpeen selvittämiseksi tutkimus keskittyi näihin tutkimuskysymyksiin:

- Kuinka aiemmin tutkittuja ohjelmiston konfiguraationhallinnan käytänteitä voidaan hyödyntää energianmittauksen ohjelmiston evoluutioiden hallinnassa.
- Miten kohdeyrityksen hyväksymät ja käytössä olevat CMMI käytänteet tukevat ohjelmiston konfiguraationhallinnan menetelmää?

Nykyään ohjelmiston ollessa kriittinen voimavara yhteiskunnassamme sen ollessa useiden modernien tuotteiden, prosessien tai palveluiden ydinosa. Tämä kasvava merkitys aiheuttaa uusia ja kovia haasteita ja paineita ohjelmistokehitysyrityksissä ja tiimeissä (Fugetta & Di Nitto, 2014). Kuitenkin hallittu ohjelmiston konfiguraatiohallinta saattaa helpottaa näiden haasteiden ja paineiden käsittelyä.

1.2 Tutkimuksen tavoitteet ja toteuttaminen

Tutkimuksen tavoitteena on luoda selvitys ja kirjoittaa menetelmärunko kohdeyrityksen ohjelmiston konfiguraationhallinnalle aiempien tutkimusten ja yrityksessä käytössä olevaa tuotekehityksen CMMI (Wibas, 2015) mallia seuraten. Tämä menetelmä luodaan yhdistämällä yrityksen hajanaiset ohjelmiston konfiguraationhallintaprosessit. Selvityksellä halutaan määrittää ratkaisutavoitteet (Kuvio 2) ja se toteutetaan yrityksen asiantuntijoiden teemahaastatteluilla. Lähestymistapana käytetään suunnittelutieteellistä prosessia.

Tutkimus toteutetaan konstruktiivisena tapaustutkimuksena (Lukka, 2001), jolla pyritään löytämään ja ratkaisemaan kohdeyrityksen ohjelmiston konfiguraatiohallinnan menetelmän puutteet. Tutkimuksessa tehtiin havaintoja yrityksen tämänhetkisistä konfiguraationhallintamenetelmistä analysoimalla yrityksessä käytössä olevia konfiguraatioiden hallintatyökaluja. Konstruktiivisen tapaustutkimuksen havaintoja verrattiin yrityksessä käytössä olevaan CMMI tuotekehityksen kypsyysmalliin (Wibas, 2015), joka sisältää mallin ohjelmiston konfiguraationhallintaan. Tutkimuksen toisessa osan syötetietona käytettiin edellä mainittuja havaintoja ja osa suoritettiin konstruktiivisesta tapaustutkimuksesta saadun yksityiskohtaisen aineiston perusteella luodulla asiantuntijoiden haastattelututkimuksella, tämän tutkimusvaiheen taksoituksena oli selvittää kohdeyrityksen ohjelmiston konfiguraatiohallinnan tila. Tapaustutkimuksen valinta tutkimustyyliseksi puoltaa myös pyrkimys saada monipuolisesi tietoja tutkittavasta kohteesta ja oppia ymmärtämään ilmiöitä sen ympärillä syvällisemmin (Metsämuuronen, 2001). Haastattelututkimukseen valittiin asiantuntijoita yrityksen neljältä eri osastolta (LIITE 1: Asiantuntijoiden haastattelu-pohja). Tutkimuksen kolmas osa toteutettiin aiempien tutkimusten, asiantuntijoiden haastatteluiden ja empiirisen tutkimisen perusteella yhteen vetämällä kyselyllä (LIITE 2).

Konfiguraatioiden hallintaohjelmistojen tutkiminen ja tiedonkeruu toteutetaan suunnittelutieteellisenä menetelmällä (design science research method) ja tutkimustulosten perusteella koostetulla puolistrukturoidulla tutkimushaastattelulla (Hirsjärvi & Hurme (2008).

2 TIETOJÄRJESTELMÄN KONFIGURAATIONHALLINTA

Yleisnimityksenä konfiguraationhallinnalla voidaan tarkoittaa mitä tahansa, jolla on määritelty rakenne tai joka koostuu joistakin ennalta määritetyistä maleista, kuten ohjelmisto, laitteisto, rakennukset, prosessitehdas, omaisuus ja jopa ihmiskeho kuuluvat konfiguraation laajaan määritelmään. Johdon näkökulmasta on usein parempi käyttää yleiskokoonpanon nimeä, koska se välttää ohjelmiston ja laitteiston aiheuttamat sekaannukset organisaatiossa (Kidd & Burgess, 2004).

Ohjelmiston konfiguraationhallintaa voidaan tarkastella monesta eri näkökulmasta, esimerkkeinä tieteellinen näkökulma ja käytännön tarpeet. Tuotteiden elinkaarten aikaisten muutosten ja niiden vaikutus ohjelmiston konfiguraationhallintaan tärkeässä asemassa sen lisätessä tuotteen arvoa.

2.1 Konfiguraationhallinta

Konfiguraationhallinta otettiin alun perin käyttöön Yhdysvalloissa 1960 luvun loppupuolella. 1970 luvulla Yhdysvaltojen hallitus kehitti monia sotilaallisia standardeja, joihin kuului konfiguraationhallinta. Myöhemmin, eritoten 1990 luvulla monet standardit ja julkaisut, joissa kerrottiin konfiguraation hallinnasta, lisääntyivät (Mette & Hass, 2002).

Tichyn (1988) mukaan ohjelmiston kokoonpanon hallinta on kurinalaisuuttasuurien ja monimutkaisten ohjelmistojärjestelmien kehityksessä ja ohjelmiston konfiguraationhallinta on tunnustettu laajalti, mikä näkyy erityisesti Ohjelmistoinstituutin kehittämässä CMM -mallissa (Humphrey, 1989), CMM määrittelee kypsyyden tasot, jotta voidaan arvioida ohjelmistojen kehitysprosesseja organisaatioissa. Tässä ohjelmiston konfiguraationhallinta nähdään yhtenä avaintekijänä siirryttäessä alkutilasta eli määrittelemättömästä prosessista toistettavissa ja käytössä oleviin projektinhallintaan, ohjelmiston konfiguraationhallintaan ja laadunvarmistukseen (Feiler, 1991).

Tutkimuksessaan Conradi ja Westfechtel (1997) toteavatkin ohjelmiston konfiguraationhallintamallin kehitysmahdollisuuden sellaiselle integroidulle hallintaversiolle, joka yhdistää laaja-alaisen ja intensiivisen versioinnin, liittää sen tila- ja muutos pohjaiseen hallintaan ja ohjelmiston korjauksiin ja muutoksiin, sekä huomio lähdekoodin ja jakelujen rakenteet ja yhtä lailla työtilojen ja pitkäaikaisten tapahtumien puitteet. Tutkimuksen mukaan tällaisen kehyksen ei odoteta tarjoavan mallia, vaan sen tulee olla muokattavissa sopivaksi tietyille ohjelmistolle.

Vuonna 2005 tutkijat Estublier, Leblang, Hoek ja Conradi loivat suuntaavivat konfiguraationhallinnalle, tutkijoiden näkemykset ja onnistuneista ja teolliseen käyttöön otetuista malleista, toimivat konfiguraationhallinnan tutkielman ohjaajana ja teemoina.

Konfiguraationhallinnan onnistumiskriteereitä käsitellään Ushmanin ja Alin julkaisussa (2012). Teoksessa tuodaan esiin 21 eri konfiguraationhallinnan onnistumiskriteeriä jaoteltuina suoritusjärjestykseen, päätöksentekoon, suorituskyvyn seuraamiseen, sopivaan resurssointiin, ympäristöjen tehokkuuteen, kommunikaatioon ja määritettyihin raja-arvoihin

Konfiguraationhallintaan liittyy läheisesti ohjelmiston ja sen julkaisujen muutosten hallintaan. Muutosten hallintaa on tutkittu ja ohjeistettu Aiellon ja Sachsin kirjassa Configuration Management, Best practices, practical methods that works in the real world (2010).

Nykyisiä konfiguraationhallintamenetelmiä kriittisesti kuvaavat Monte-Morin ja da Cunhan toteavat tutkimuksessaan GALO: Semantic Method for Software Configuration Management, etteivät nykyiset konfiguraationhallintamenetelmät huomio riittävästi komponenttien versiointia vaan pitävät päähuomion lähdekoodin tuotannossa ja ohjelmiston käänösprosessien vertailussa (Monte-Mor & da Cunham, 2014).

Ohjelmiston konfiguraationhallinta voidaan nähdä myös alajoukkona laajemmalle konfiguraationhallinnalle. Ohjelmiston konfiguraatioiden hallinnalla tarkoitetaan monimutkaisten järjestelmien kehityksen kurinalaista hallintaa (Estublier, 2007, 1). Laajemmin ohjelmiston konfiguraationhallinta on projektin toiminto, tavoitteenaan tehostaa teknisiä ja johdon aktiviteetteja (Brügge & Dutoit, 2004).

Aiemmin ohjelmiston konfiguraationhallinta työkaluilla oli rajoitettu toiminnallisuus ja sovellettavuus, mutta modernit ohjelmiston konfiguraationhallintajärjestelmät tarjoavat kehittyneitä toimintoja, jotka mahdollistavat tehokkaan hallinnan ohjelmiston kehityksessä esiin nouseville kohteille, joita havaitaan monimiljoonaisen koodirivin järjestelmän kehityksessä. Tämä muutos pienistä, yksikertaisista työkaluista koko ohjelmiston konfiguraationhallinnaksi kuvastaa akateemisten ja teollisten tutkimusten mukaan tulkita ohjelmiston konfiguraationhallinta -alan kehittymiseksi (Estublier ym., 2005).

Energianmittauksen järjestelmän ollessa osa asioiden internetiä (Kraijak, & Tuwanut, 2015.) ja energianmittauksen mittalaitteiden sisältäessä omat konfiguraationsa, niiden vaikutukset otetaan huomioon ainoastaan niiden vaikuttaessa ohjelmiston konfiguraationhallintaan.

2.2 Konfiguraationhallinta osana ohjelmistokehitystä

Ohjelmiston tuotekehitys koostuu yleensä monimutkaisia, toisiinsa liittyvistä artefakteista. Nämä artefaktien rakentuvat analyytikoiden, kehittäjien ja testaajien välisen kanssakäymisen avulla, artefaktien suhteita mietittäessä ei oteta huomioon konfiguraationhallinnan tarjoamia hyötyjä. Tämä johtuu pääasiassa siitä, että nämä työkalut ovat edelleen rajoitettuja ja lähdekoodin kehitys ja kääntäminen ohjaavat niitä (Hoek ym., 1995). Tähän arkkitehtuurin puutteeseen voidaan hakea ratkaisua tutkittavalla ohjelmiston konfiguraationhallinnalla.

Konfiguraationhallinta on prosessi, joka tarjoaa viestintäkanavan ohjelmoinnin vaiheiden erimielisyyksien selvityksiin suunnittelun, kehittämisen, julkaisun, testaamisen ja ohjelmistojen jakelun aikana. Konfiguraation hallinnan toimet voi olla ryhmän nimeämisen yksilön tai ryhmän vastuulla. Tällä tavoin varmistetaan, että kaikki ohjelmistokehitykseen osallistuvat henkilöt tietävät miten tuote on suunniteltu, kehitetty, julkaistu, testattu ja toimitettu. Lopulta näiden samojen henkilöiden ja lopullisten käyttäjien tulee tietää mitä ja minkälaisia ohjelmistoja on kasattu, testattu ja jaeltu, jotta ohjelmisto tuotetta käytetään niin kuin on tarkoitettu ja sitä voidaan tukea ja huoltaa loppukäyttäjän ympäristössä (Berlack, 1997).

Alla on prosessien näkökulmasta, määritellyt konfiguraationhallinnan aktiviteetit:

1. Komponentintunnistaminen vaiheessa tunnistetaan konfiguraatio -komponenttien metatieto. Metatiedon tunnistaminen on konfiguraationhallinnan kulmakivi, pitäen sisällään komponentin suhteen muihin komponentteihin ja rajapintoihin.
2. Komponentin tallennus vaiheen tarkoitus on säilyttää komponentti vahingoittumattomana. Tallennusvaiheessa on syytä kirjata ylös komponentin asennus ympäristön tiedot.
3. Komponentin muutoksenhallintavaihe on välttämätön osa ohjelmistokehitystä. Asiakkaan tarpeet muuttuvat ja ne aiheuttavat muutospyyntöjä koska asennusympäristöt kehittyvät ja muuttuvat. Ohjelmoijat kehittävät uusia ratkaisuja ja ne aiheuttavat usein muutoksia komponentteihin
4. Komponentin tilaraportointi tuottaa tietoa komponenttien tilasta konfiguraationhallinnan, tuotteen kehityksen ja ylläpidon tarpeisiin. (Mette & Hass, 2002).

Tuotteen elinkaarenaikainen muutosten- ja konfiguraationhallinta on vaativa tehtävä, sillä nykyaikainen toimintaympäristö luo monia haasteita kokonaisuuden hallintaan. Verkostoitunut, nopeasti muuttuva ja usein globaali toimintakenttä johtaa uusiin vaatimuksiin yrityksen sisäisten ja ulkoisten toimijoiden tekemiselle. Organisaation kyky hallita käytössään olevia tuotteita ja järjestelmiä on sen toiminnan ydin-osaamisalueita. Konfiguraationhallinnalla tuotteiden ja

järjestelmien kehitys, ajassa muuttuminen voidaan tehdä hallitusti ja tarkoituksen mukaisesti. Sen kautta organisaatio voi vaikuttaa oman toimintansa laatuun monilla tavoin. (Laari,2016). Käytännössä laadun- ja projektinhallinta vaativat tuekseen konfiguraationhallintaa. (Kidd & Burgess, 2004).

2.3 Konfiguraationhallinnan teemat

Tässä alaluvussa kuvataan aiempien tutkimusten teemat. Asiantuntijahaastattelussa esiin nousseista haasteista kerättiin tämän kappaleen teemat. Teemat liittyvät läheisesti Estublierin ym. (2005) tutkimustuloksiin, joista osa haluttiin esitellä tarkemmin tutkimuksessa. Teemat linkittyvät myös empiirisen tutkimuksen havaintoihin mahdollisina ohjelmiston konfiguraationhallinnan kehityskohteina. Tutkimuksesta rajattiin pois tutkimuksessa mainitut hylätyt ideat, tutkimukseni rajoituksessa uuden menetelmän luomiseen.

Nykyisillä ohjelmiston konfiguraationhallintaprosesseilla hallitaan monenlaisia ohjelmistoja, hajautetuista sijainneista useamman henkilön voimin ja eri tyyppisiä tietokoneita hyödyntäen. Tämä käytäntö vaatii usein tarkkaa organisaation prosessien tukea. Ohjelmiston konfiguraationhallinta perustuu tuotekehityksen sisäisten prosessien tukemiseen, sekä kohdeyrityksen sidosryhmien konfiguraationhallinnan prosessien ymmärtämiseen (Estublier ym., 2005).

2.3.1 Teema 1: Muutossarjat ja niiden versiointi (change set)

Ohjelmiston konfiguraatiokokonaisuuden luomisen perinteinen tapa on perustunut olemassa olevien hallinta-alkioiden korvaamiseen uusilla. Konfiguraationhallinnan puitteissa muutossarjat esiteltiin alun perin jo vuonna 1988 Aides de Campin toimesta (Estublier, 2005, s. 1). Muutossarjoja, jotka koskivat useampaa tiedostoa, ryhdyttiin kutsumaan loogisilla nimillä, kuten esimerkiksi "FixBug243". Myöhemmin konfiguraatio voitiin luoda lisäämällä tai poistamalla pohjakonfiguraatiosta muutossarjoja (kuten $C2 = C1 + \text{FixBug243}$ -laajennusosa2) jossa C1 tarkoittaa pohjakonfiguraatiota, C2 uusittua pohjakonfiguraatiota (Estublier ym., 2005).

Muutossarjoja on tutkittu, laajennettu ja muokattu akateemisissa tutkimuksissa paljon (Lie ym., 1989, Zeller & Snelting, 1997, Westfechtel ym., 2001.). Muuttuvat sarjat ovat vähitellen tulleet standardiksi ominaisuudeksi korkea tasoisessa ohjelmiston konfiguraationhallinnassa. Kuitenkin muutossarjojen hallinta on saanut paljon yksinkertaisemman muodon nimellä muutospaketti, joka yhdistää muutossarjat ja tehtävän käsitteen (Estublier ym., 2005). Tehtävän käsitteeseen liitettynä muutossarjojen kaksi ominaisuutta ovat:

1. Konfiguraationhallinnassa säilytettäviä mahdollisesti muuttuvia konfiguraatioita.

2. Tietyn asiakasjakelun konfiguraatioiden muutokset ja jakeluun liittyvien konfiguraatiomuutosten versiointi.

2.3.2 Teema 2: Prosessituki

Ohjelmiston konfiguraationhallinnan tehtävät kehittyivät tiedostojen hallinnasta ohjelmiston ylläpidosta ja kehityksestä vastaavien henkilöiden johtamiseksi, aluksi tämä vaatii muutosjohtamista, mutta viime aikoina ohjelmiston konfiguraationhallinta tukee yleisiä prosesseja. Yleensä tämä prosessituki oli liitetty työkaluihin kiinni ja tuen kehittäminen vaatii laajan kommentisarjan työkalun päälle. Nykyaikaiset, korkealaatuiset ohjelmiston konfiguraationhallintajärjestelmät lisäävät prosessitukea entisestään. Ne tukevat muutosten mm. muutostenhallintaa ja antava organisaatiolle mahdollisuuden suunnitella ja valvoa kehitysprosesseja koko yritystoiminnassa.

Ohjelmiston konfiguraationhallinnan menestyminen perustuu suurilta osin organisaation prosessitukeen eli vaatii kohdeorganisaation kypsyyden, yksinkertaisen prosessituen käsitteen ja rajapinnat.

2.3.3 Teema 3: Algoritmit: Konfiguraatiotiedostojen vertailu ja yhdistäminen

Alun perin ja edelleenkin ohjelmiston konfiguraationhallintajärjestelmät tukevat perinteiseen tekstirivien vertailuun ja yhdistämiseen (Hunt & McIlroy, 1967, Meyers, 1986).

1. ASCII -tiedostojen tekstirivien vertailu.
2. ASCII -tiedostojen tekstirivien yhdistäminen.

2.3.4 Teema 4: Ohjelmiston konfiguraationhallinnan hajautettu- ja etäkehitys

Estublierin ym. (2005) tutkimuksessa tämän teeman pääpainon sai hajautetun ja etäkehityksen alue, jonka ongelmanratkaisussa akateeminen tutkimus oli keskittynyt yksinkertaisen web -liittymän lisäämiseen ohjelmiston konfiguraationhallintaan, jotta saataisiin etäyhteys hallinta-alkioiden säilytyspaikkaan. Tämän jälkeen ensimmäinen läpimurto oli versionhallinnan (CVS) lisääminen asiakas/palvelin -protokollaan. Nykyään tämä melko yksinkertainen laajennus on laajassa käytössä ja toimii laajana tukiteknologiana. Mielenkiintoista on ohjelmiston konfiguraationhallinnan tekniikoiden menestyminen erityyppisellä jakelu ja hajautusmenetelmällä verrattuna esimerkiksi web-teknologioihin. Web teknologioista esimerkiksi voidaan ottaa web-sivujen hallinta, jossa käytetään nopeasti muuttu-

via resursseja, joita monet henkilöt hallinnoivat mahdollisesti monissa eri maantieteellisissä sijainneissa, mutta peruseriaatteet ja tekniikat ovat samat jotka suunniteltiin alun perin ohjelmiston konfiguraationhallinnalle.

2.3.5 Teemojen yhteenveto

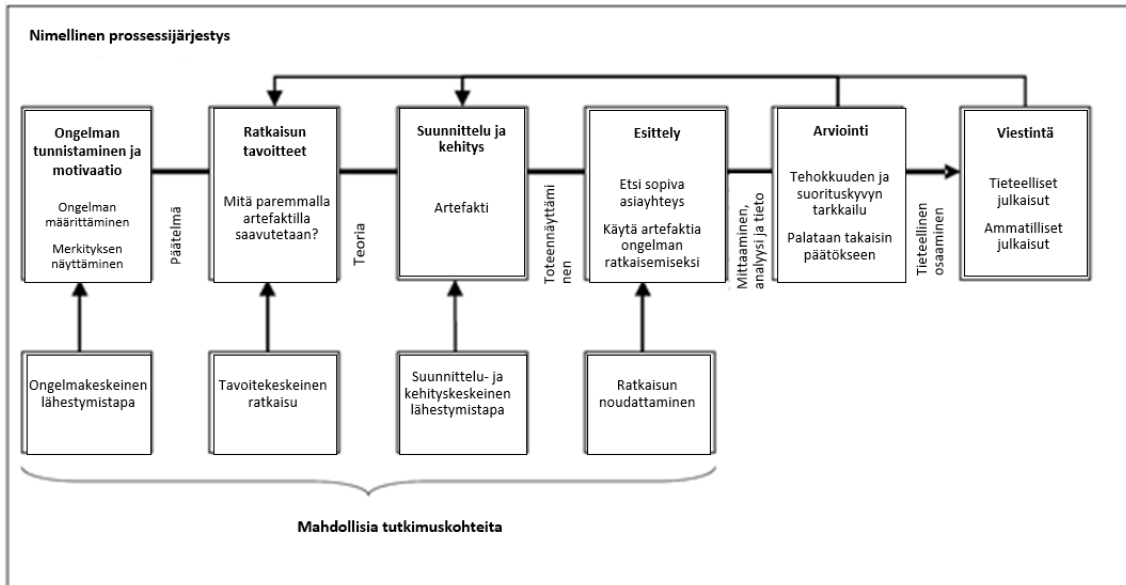
Tutkituista teemoista tarkasteltiin esimerkkejä alueista, joilla tutkimuksen mukaan olivat onnistuneet vaikutukset ohjelmiston konfiguraationhallinnan kehittämiseen. Tutkijoiden (Estublien ym. 2005) mukaan ohjelmistojen konfiguraationhallinnan onnistunut toteutus vaatii kolmen osatekijää: 1. Asiakkaalla tulee olla tietty ongelma osoitettavissa. 2. Ohjelmiston konfiguraationhallinnan tuottajien pitää olla kyvykkäitä tuottamaan tarvittu ominaisuus. 3. Asiakkaan tulee olla valmis hyväksymään mahdollinen uuden ominaisuuden käytön lisäämä ylimääräinen rasitus.

3 ENERGIANMITTAUSALAN TAPAUSTUTKIMUS

Tässä luvussa kuvataan kohdeyrityksen organisaatio, mittalaite- ja ohjelmistotuotantomallit, käytössä olevat ohjelmiston konfiguraationhallinnan prosessit ja tutkimuksen avulla luotu menetelmä. Luvussa esitellään myös yrityksen CMMI mallin linkittyminen ohjelmiston konfiguraationhallintaan.

3.1 Suunnittelutieteellinen tutkimus

Tämän tutkimuksen aloitustilanteeksi ja tavoitteisiin liittyväksi tutkimusmetodiksi valittiin Peffersin ym. (2007) DSRP-mallin (Design Science Research Process) (Kuvio 2). Malli koostuu kahdesta osasta: erilaisia alkutilanteita kuvaavasta osasta sekä varsinaisen tutkimuksen kulkua kuvaavasta osasta. Erilaisia aloitustilanteita mallissa on kuvattu neljä ja tutkimuksen prosessia kuvaavia vaiheita kuusi. Tutkimuksen alkutilanteesta riippuen tutkimus voidaan aloittaa eri tutkimusprosessin vaiheesta. Tutkimusprosessin vaiheet ovat ongelman tunnistaminen, tavoitteiden määrittelyt, artefaktin, eli selvityksen suunnittelu ja kehitys, demonstraatio, arviointi sekä kommunikaatio. Ongelma-keskeisessä alkutilanteessa tutkimus aloitetaan ensimmäisestä vaiheesta, tavoitekeskeisessä tilanteessa toisesta ja suunnittelu- ja kehityskeskeisessä lähestymistavassa kolmannesta vaiheesta. Neljännestä vaiheesta voidaan aloittaa, jos tutkimuksen kannalta olennainen ratkaisu on jo olemassa ja sitä halutaan tarkastella tarkemmin. Tämä tutkimus aloitettiin vaiheesta ongelman tunnistamisen -vaiheesta.



KUVIO 2 - DSPR-tutkimusmalli (Peffer ym., 2007)

Tutkimus koostettiin noudattamalla suunnittelutieteellisen tutkimuksen metodologiaa (Peffer ym., 2007), joka sisältää prosessia kuvaavan osan ja lähtötilanteen (Kuvio 2).

Seuraavaksi kuvataan suunnittelutieteellisen ¹tutkimusprosessin vaiheet (Peffer ym., 2007.).

1. Ongelman tunnistaminen ja motivointi: Tässä vaiheessa tunnistettiin tutkimuksen ongelma ja määriteltiin syyt ongelmanratkaisulle. Ratkaisun merkityksen selittäminen lisää motivaatiota tutkijalle sekä lukijalle. Tutkimusongelman ja ratkaisun merkitys määritellä tarkasti. Tutkimusongelma tulee määritellä tehokkaasti ja yksityiskohtaisesti jotta se on motivoiva.
2. Ratkaisun tavoitteiden määrittely: Ratkaisun tavoitteet määräytyvät määrittelystä ongelmasta. Tavoitteiden mahdollinen toteuttamiskelpoisuus tulee arvioida. Päämäärä voi olla laadullinen tai määrällinen; tärkeänä on pidettävä ratkaisun merkitystä ja sen eroavuutta olemassa olevin ratkaisuihin (Peffer ym., 2007). Tutkimuksessa määritellään menetelmä jonka vaikutukset yltyvät yrityksen ohjelmiston konfiguraationhallintaan tuotekehityksen alueella. Tutkimuksessa pyritään kehittämään menetelmä, jota voidaan hyödyntää yrityksen ohjelmiston konfiguraationhallinta -menetelmän kehityksessä.
3. Suunnittelu ja kehittäminen: Luodaan menetelmä jolla ratkaistaan tutkimusongelma. Ratkaisun suunnittelu ja kehittäminen vaativat empiiristä

tutkintaa ja kehittämistä, sen määrittäminen ja kuvaaminen tarvitsevat tietoutta suunnittelun ja kehittämisen tukemiseen (Peffer ym., 2007.).

Tutkimuksessa luotavan ohjelmiston konfiguraationhallinnan -menetelmän luominen vaatii aiempien tutkimusten läpikäynnin ja arvioinnin, aiemmista tutkimuksista löytyvän tietouden hyödyntäminen on välttämätöntä. Käytössä olevat prosessit ja menetelmät toimivat luotavan menetelmän luomisessa ohjaavina tekijöinä.

4. **Demonstraatio:** Tässä vaiheessa demonstroidaan ja näytetään toteen ratkaisun vahvuudet tutkimusongelmaan verrattuna. Ratkaisu esitellään määritellyn ongelman kontekstissa. Testaus voidaan suorittaa kokeen, simulaation, tapaustutkimuksen tai muun soveltuvan toiminnon avulla. Testaaminen vaatii tietämystä ratkaisun soveltamisesta (Peffer ym., 2007.).
5. **Arviointi:** Arvioidaan kuinka hyvin kehitetty menetelmä ratkaisee määritellyn ongelman. Arvioinnissa verrataan asetettuja tavoitteita saatuihin tuloksiin, määritellyn ongelman ratkaisu todennetaan yrityksen asiantuntijoille suunnatulla kyselyllä. Kysely liittyy tapaustutkimukseen ja sillä pyritään saamaan yksityiskohtaista tietoa yksittäisestä tapauksesta tai pienestä joukosta toisiinsa suhteessa olevia tapauksia (Hirsjärvi ym., 2008.).
6. Tutkimusprosessin iteratiivisuudesta johtuen voidaan palata kolmanteen vaiheeseen suunnittelun ja kehityksen parantamiseksi. Kehittämiskohde voidaan myös tunnistaa ja jatkaa kohtaan 6, ja esittää jatko-tutkimustarpeet (Peffer ym., 2007.)
7. **Viestintä:** Viimeisessä vaiheessa kommunikoidaan ja viestitään kohdeyleisön kanssa, kohdeyleisönä on tiedeyhteisö. Saavutetun tiedon jakaminen tutkimuksesta ja tutkimusongelmasta on tärkeää, jotta tutkimusta voidaan käyttää vertailuun ja tiedon lähteenä. Formaali raportointi on tärkeää, jotta tutkimuksesta saatua tietoa voidaan käyttää uskottavasti (Peffer ym., 2007.). Tutkielman toimiessa dokumenttina tehdystä työstä, tutkija hyödyntää tutkimusta menetelmän jatkokehityksessä.

Kohdeyrityksen käyttäessä monia eri menetelmiä, tämä menetelmän runko on luotu yrityksessä käytössä olevaa tuotekehityksen CMMI (Wibas, 2015) mallia seuraten, asiantuntijoiden teemahaastattelujen perusteella sekä aiempien tutkimuksiin tukeutuen.

3.2 Aineiston keruu

Selvitys toteutettiin yrityksen asiantuntijoiden puolistrukturoiduilla teemahaastatteluilla. Haastattelumetodina käytettiin puolistrukturoitu malli, jossa asiantuntijoita pyydettiin kertomaan ääneen näkemyksiä yrityksen ohjelmiston konfiguraationhallinnasta teemojen perusteella, jotta aiheesta saatiin kvantitatiivisempaa tietoa (Barcelos, 2003).

Haastattelun kohteiksi valikoitui yritysjohdon suosittelimina sen hetkisten ammattinimikkeiden ja kohdeyrityksen osastojen perusteella teknisiä kirjoittajia (technical writer), vanhempia testaussuunnittelijoita (senior testing designer), vanhempia ohjelmistosuunnittelijoita (senior software developer), ohjelmistoarkkitehteja (software architect), tuotepäällikkö (product manager) ja projektipäälliköitä (project manager).

Asiantuntijahaastatteluun osallistuvia haastateltavia oli 17 henkilöä ja vastaukset kirjattiin haastatteluraporttiin. Haastattelut suoritettiin kuutena eri ajankohtana (05.05.2017 – 05.06.2017). Haastattelujen aineiston kerääminen sekä analysointi tapahtuivat samanaikaisesti. Selvitys menetelmästä koostettiin tutkimuksen aikana ja selvityksen tueksi ja kehityskohteiden tunnistamiseksi luotiin kysely joka kohdistettiin 51 yrityksen asiantuntijalle ja johdolle, kyselyyn vastasi 9 henkilöä jolloin vastausprosentti on 17 %. Kyselyn vastaukset analysointiin teemoittelu -menetelmää käyttäen, kyselyn runko luotiin helpottamaan teemoittelua, teemoittelu jaoteltiin CMMI-mallin mukaisiin teemoihin.

Haastattelun jälkeen jatkettiin ohjelmiston konfiguraationhallinnan empiiristä tutkimista ja luotiin ensimmäinen versio yrityksen ohjelmiston konfiguraationhallintamenetelmän rungoksi, menetelmän rungon vaiheiden varmistamiseksi luotiin strukturoimaton kysely mallin toimivuudesta (LIITE 2). Kysely perustui empiiriseen tutkimukseen ja asiantuntijahaastattelujen avulla kerättyihin teemoihin ja niihin linkittyviin Estublierin ym. tutkimustuloksiin. Estublierin ym. (Estublier ym., 2005). Kysely lähetettiin 51 yrityksen johtajille ja asiantuntijoille, kyselyyn vastasi 9 henkilöä, vastausten perusteella menetelmärunko valmistui.

3.3 Aineiston analysointi

Haastattelujen ja kyselyn aineistoa analysointiin haastattelunaikaisesti, koska aineiston analyysitapojen määrittely ei ole Hirsjärven ym. (2007, 223) mukaan aina yksiselitteistä ja aineiston kerääminen sekä analysointi voivat tapahtua samanaikaisesti. Tässä opinnäytetyössä haastatteluja oli useita, joten analyysiä tehtiin jo haastattelujen aikana. Pääperiaatteen mukaan analyysi valitaan siten, että saadaan parhaiten vastaus tutkimusongelmaan tai tehtävään (Hirsjärvi ym., 2007, 224-225).

Käytetty lähestymistapa analyysin oli teoriaohjaava, kyse oli abduktiivisesta päättelystä, jossa tutkija tekee analyysiä aineistoa ja valmiita malleja yhdistelemällä (Tuomi & Sarajärvi 2009, 96-97). Alastalon ja Åkermanin (2010, 389-390)

mukaan asiantuntija-analyysin päätavoitteena on tuoda esiin faktoja, jotka tuotetaan yhdessä haastateltavan kanssa haastattelun tai tutkimusprosessin kuluessa. Tässä opinnäytetyössä olemassa olevia toimintatapoja myös tarkasteltiin kriittisesti. Haastatteluiden suurimpana tavoitteena oli tuoda esille haastateltavien mielipiteet siitä, minkälaisia haasteita he ovat kokeneet ohjelmiston konfiguraationhallinnan kanssa ja heidän kartoitta heidän jatkokehitysjatukset.

3.4 Intensiivinen tapaustutkimus

Tutkimus toteutettiin intensiivisenä tapaustutkimuksena tekemällä havaintoja yrityksen tämänhetkisistä konfiguraationhallintamenetelmistä analysoimalla yrityksessä käytössä olevia konfiguraatioiden hallintatyökaluja. Tapaustutkimus eteni Kuvion 2 mukaisesti, alla kuvattuna vaiheet:

1. Ongelma tunnistettiin tutkijan empiiristen kokemusten, yrityksen johdon ja asiantuntijahaastattelujen avulla. Tutkimusongelma määriteltiin tarkasti.
2. Ratkaisun tavoitteet määriteltiin tutkijan ja asiantuntijoiden kanssa yhteistyössä. Päämäärä oli kehittää uusi energianmittausalan uusi tietojärjestelmän konfiguraationhallintamalli.
3. Konfiguraationhallintamallin suunnittelu ja kehittäminen piti sisällään empiirisen tutkinnan ja asiantuntijahaastattelut. Vaiheessa käytiin läpi aiempia tutkimuksia ja yrityksessä käytössä olevaa CMMI kypsyysmallia. Yrityksen aiemmat käytössä olevat prosessimalleja käytettiin suunnittelun tukena. Tapauksen pohjalta luotiin strukturoimaton kysely mallin toimivuudesta (LIITE 2).
4. Demonstraatiovaiheessa esiteltiin luotu malli, määritellyn ongelman kontekstissa, yrityksen johdolle ja asiantuntijoille. Arviointivaiheessa yrityksen johdolle ja asiantuntijoille toimitettiin strukturoimaton kysely, jolla pyrittiin selvittämään yksityiskohtaista tietoa yksittäisestä tapauksesta tai pienestä joukosta toisiinsa suhteessa olevia tapauksia (Hirsjärvi ym., 2008.).
5. Tutkimuksen iteroinnissa palattiin takaisin suunnittelu ja kehitysvaiheeseen. Uutta menetelmää muokattiin johdon ja asiantuntijoiden kommenttien perusteelle.
6. Tutkimuksen päätteeksi lopulliset tutkimuspaperit toimitettiin asianomaisille osapuolille, kuten yrityksen asiantuntijoille ja johdolle.

3.5 Kohdeyrityksen kuvaus

Tapaustutkimuksen valinta tutkintasuunnaksi soveltui hyvin yrityksen ohjelmiston konfiguraationhallinnan tutkintaan. Tieteellisten paradigmojen ja käytännön tutkinnan yhdistäminen sopivat hyvin tutkimuksen ainutlaatuisuuden vuoksi intensiiviseksi tutkimukseksi (Eriksson & Kovalainen, 2014, 26)

Tapaustutkimuksen ajatellaan olevan usein perusluonteeltaan uutta löytävä lähestymistapa (exploratory case study). Tämän mukaisesti tavoitteena on tuottaa uusia teoreettisia ideoita, käsitteitä, propositioita tai hypoteeseja niistä syistä, olosuhteista, tai prosesseista, jotka tuottavat tutkittavia ilmiöitä. Ideoita, käsitteitä, propositioita ja hypoteeseja voidaan edelleen koetella joko samassa tai uusissa tutkimuksissa. (Eriksson & Koistinen, 2014, 22). Intensiivisessä tapaustutkimuksessa (intensive case study) on tavoitteena ainutlaatuisen, ja tästä syystä teoreettisesti mielenkiintoisen tapauksen tiheä kuvaus, tulkinta ja ymmärtäminen. (Eriksson & Koistinen, 2014, 26). Tapaustutkimus on toteutettu suunnittelu-tieteellisenä tutkimuksena.

Energia alan murroksessa (Tutkimushanke: Smart Energy Transition, 2016) aiheuttaa energian toimitukselle ja myös energia-alan ohjelmistoille lisääntyvää ja tarkentuvaa konfiguraationhallintaa. Yrityksen projektinhallintamenetelmä on ns. hybridi, sekoitus vesiputousmallista projektinhallintaa ja ketterää ohjelmistotuotantoa. Yrityksen laitekehityksessä vaatimukset (esim. aikataulut ja rajapinnat) ovat tarkkoja ja joustamattomia, kun taas käytössä oleva ketterä ohjelmistotuotantomalli on joustavampi.

3.5.1 Organisaatio

Energianmittaus alan yrityksen, Landis Gyr Oy:n Suomen osasto on yrityksen tuotekehitysosasto. Landis Gyr:in energianmittaus alkoi jo vuonna 1896 sähkön mittauksella. Monien vaiheiden jälkeen yritys osti vuonna 2007 suomalaisen Enermet Groupin (Landis Gyr Oy, History, 2017). Tästä osastosta on muodostunut yrityksen tuotekehitysosasto ja tällä hetkellä kansainvälinen tuotekehitys on hajautettu 30 maan alueelle (Landis Gyr, Global Organisation, 2017). Tutkimus on tuotekehitysonginisaatiolle ajankohtainen ja ohjelmiston konfiguraationhallintamenetelmän kirjaaminen tuottaa arvoa pelkällä olemassa olollaan.

3.5.2 Ohjelmiston konfiguraationhallinta ja CMMI-kypsyysmalli

CMMI- kypsyysmalli on esitelty viime aikoina voimavarana, jolla ohjelmistokehitys organisaatioiden käyttöön, jotta organisaatiot voisivat voittaa tai säilyttää asiakkaita. Organisaatio, joka voi osoittaa korkean tason kypsyyttä näillä käytännöillä pärjää ohjelmistotuotannon kilpailussa (Silva ym., 2015.).

Yrityksen käytössä on CMMI tuotekehityksen kypsyysmalli (Wibas, 2015), joka sisältää mallin ohjelmiston konfiguraationhallintaan. Mallista löytyvät palveluiden kypsyysalueet linkittyvät yrityksen ohjelmiston konfiguraationhallintaan (taulukko 1). Tässä tutkimuksessa keskityttiin tuotteen ja palvelun kehittämiseen (CMMI-DEV).

TAULUKKO 1 Yrityksen CMMI-kypsyysmallin alueet (Landis Gyr Oy, 2017)

Alue	Selite
Product and service development	Tuotteen ja palvelujen kehittämisen kypsyys (CMMI-DEV).
Service establishment, management and delivery	Palvelujen kypsyys (CMMI for Services).
Service Operation Product and service acquisition	Tuotteiden ja palvelujen hankinnan kypsyys (CMMI-ACQ).

CMMI käytänteistä löytyy konfiguraationhallinnan prosessin hallinnan vaiheet, nämä vaiheet soveltuvat myös ohjelmiston konfiguraationhallinnan prosessin vaiheiksi (taulukko 2).

TAULUKKO 2: CMMI-kypsyysmalli ja ohjelmiston konfiguraationhallinnan käytänteet (Wibas, 2015)

CMMI Käytänne	CMMI Käytänne kuvaus / vaihe
CM.SP 1.1	Konfiguraationhallinta-alkioiden tunnistaminen
CM.SP 1.2	Konfiguraationhallinnan menetelmän luominen
CM.SP 1.3	Lähtötilanteen tunnistaminen
CM.SP 2.1	Muutospyyntöjen seuranta
CM.SP 2.2	Konfiguraationhallinta-alkioiden valvonta
CM.SP 3.1	Perusta konfiguraationhallinnan aikakirja (records)
CM.SP 3.2	Suorita konfiguraationhallinnan lähtötilanteen auditointi

Ohjelmiston konfigurointihallinnan tarkoituksena on luoda ja ylläpitää tuotteiden eheyttä konfiguraation tunnistamisen, konfigurointihallinnan, konfigurointitilojen kirjanpidon ja konfigurointitarkastusten avulla (CMMI, 2010). Alla kuvattuna tarkat CMMI käytänteet. Käytänteet olivat tukemassa asiantuntijahaastatteluja.

CM.SP 1.1 Konfiguraationhallinta-alkioiden tunnistaminen

Konfiguraationhallintaan talletettavien komponenttien, hallinta-alkioiden ja niihin liittyvien työtuotteiden tunnistaminen.

Ali-käytänteet:

1. Konfiguraationhallinnan kohteiden valinta ja tuotteet, jotka muodostuvat dokumentoitujen kriteereiden perusteella.
2. Yksilöivien tunnisteiden luominen konfiguraationhallinnan hallinta-alkioille.
3. Konfiguraationhallinta-alkioiden ominaisuuksien määrittely (esim. tiedoston laatija, tiedoston tyyppi, ohjelmointikieli, tiedoston tarkoitus).
4. Konfiguraationhallinta-alkion konfiguraationhallintaan lisäämisaika.
5. Konfiguraationhallinta-alkion omistaja ja vastuu.

6. Konfiguraationhallinta-alkion suhteet muihin hallinta-alkioihin ja konfiguraationhallinnan rakenteeseen.

CM.SP 1.2 Konfiguraationhallinnan menetelmän luominen

Konfiguraationhallintamenetelmä sisältää tallennusvälineet, menettelyn ja työkalut järjestelmän käyttämiseen. Menetelmä voi koostua useista osajärjestelmistä, joilla on erilaiset toteutukset, jotka sopivat kullekin konfiguraationhallintaympäristölle.

Ali-käytänteet:

1. Monitasoisen mekanismin luominen konfiguraationhallinnalle. Mekanismin tasot valitaan tyypillisesti projektin tavoitteiden, riskien ja resurssien perusteella. Tasot voivat vaihdella hankkeen elinkaaren, kehitteillä olevan järjestelmän tyyppin ja erityisten projektivaatimusten mukaan.
2. Konfiguraationhallintajärjestelmän käyttöoikeuksien hallinta.
3. Konfiguroinnin hallinta-alkioiden noutaminen ja tallennus konfiguraationhallintajärjestelmästä/ään.
4. Konfiguroinnin hallinta-alkioiden jakaminen eri tasoille.
5. Konfiguroinnin hallinta-alkioiden tallennus ja palautus.
6. Konfiguroinnin hallinta-alkioiden aikakirjojen tallennus, päivitys, noutaminen.
7. Konfiguraationhallinnan raporttien luominen (konfiguraationhallinnasta).
8. Konfiguraationhallinnan sisällön talletus.
9. Konfiguraationhallinnan rakenteen tarkistus tarvittaessa.

CM.SP 1.3 Lähtötilanteen tunnistaminen

Lähtötilanteen tunnistaminen ja sisäisen kehityksen ja asiakkaalle toimituksen lähtötilannepisteen luominen. Monia eri lähtötilanteita voi olla käytössä esim. tuotekehitykselle ja testaukselle. Yksi yhteinen viitekehys sisältää järjestelmän vaatimukset ja tuotteen määritykset tuotteen alkuvaiheessa.

Ali-käytänteet:

1. Konfiguraationhallinnan lähtötilanteen luomiselle oikeutuksen saaminen yrityksen johdolta.
2. Julkaiseminen tai lähtötilanteet luominen vain konfiguraationhallinnasta löytyvillä hallinta-alkioilla.
3. Lähtötilanteiden sisältämine hallinta-alkioiden kuvaaminen.
4. Lähtötilanteiden saataville asettaminen.

CM.SP 2.1 Muutospyyntöjen seuranta

Vaatimuksista johdetut, virhekorjauksista ja tuotteen korjauksiin liittyvien muutosten seuraaminen.

Ali-käytänteet:

1. Muutospyyntöjen luominen ja kirjaaminen muutospyyntöjen tietokantaan.
2. Muutospyynnöissä esitettyjen muutosten ja korjausten vaikutusten arviointi.
3. Muutospyyntöjen luokittelu ja priorisointi.
4. Seuraavan lähtötilanteen muutospyyntöjen katselmointi olennaisten sidosryhmien kanssa.
5. Muutospyyntöjen seuranta niiden sulkemiseen asti.

CM.SP 2.2 Konfiguraationhallinta-alkioiden valvonta

Hallinta-alkioiden seuranta pitää sisällään kaikkien hallinta-alkioiden seurannan, muutosten hyväksynät ja lähtötilanteen päivittämisen.

Ali-käytänteet:

1. Konfiguraation hallinta-alkioiden elinaikainen hallinta
2. Asianmukaisen valtuutuksen hankinta ennen muuttuneiden hallinta-alkioiden lisäämistä konfiguraationhallinta järjestelmään.
3. Konfiguraation hallinta-alkioiden oikeellisuuden ja eheyden valvonta, hallinta-alkioita käytettäessä.
4. Konfiguraation hallinta-alkioiden katselmoinnit ja varmistukset, jottei muutokset vaaranna niiden turvallisuutta.
5. Konfiguraation hallinta-alkioiden muutosten ja niiden syiden kirjaaminen.

CM.SP 3.1 Perusta konfiguraationhallinnan aikakirja (Records)

Konfiguraationhallinta-alkioiden aikakirjan perustaminen ja ylläpito.

Ali-käytänteet:

1. Konfiguraationhallinnan toimien tallentaminen riittävän yksityiskohtaisesti, jotta kunkin hallinta-alkion sisältä ja tila tunnetaan ja aiempien versioiden palauttaminen on mahdollista.
2. Asianomaisten sidosryhmien seuranta-oikeuksien varmistaminen.
3. Uusien lähtötilanteiden määrittäminen:
 - Käyttöoikeuksien tuottaminen valtuutetuille sidosryhmien loppukäyttäjille.
 - Hallinta-alkioiden lähtökohtaiset kopioiden saataville tarjoaminen
 - Automaattisten hälytysten tuottaminen sidosryhmille, kun hallinta-alkioiden tarkistuksista, poistoista tai muutoksista (muutos ehdotusten seuranta).
4. Tietyn lähtötilanteen muodostavien hallinta-alkioiden tunnistaminen.
5. Onnistuneiden lähtökohtien erojen kuvaaminen.

6. Konfiguraation hallinta-alkioiden tilan ja historian tarkastaminen tarpeen mukaan.

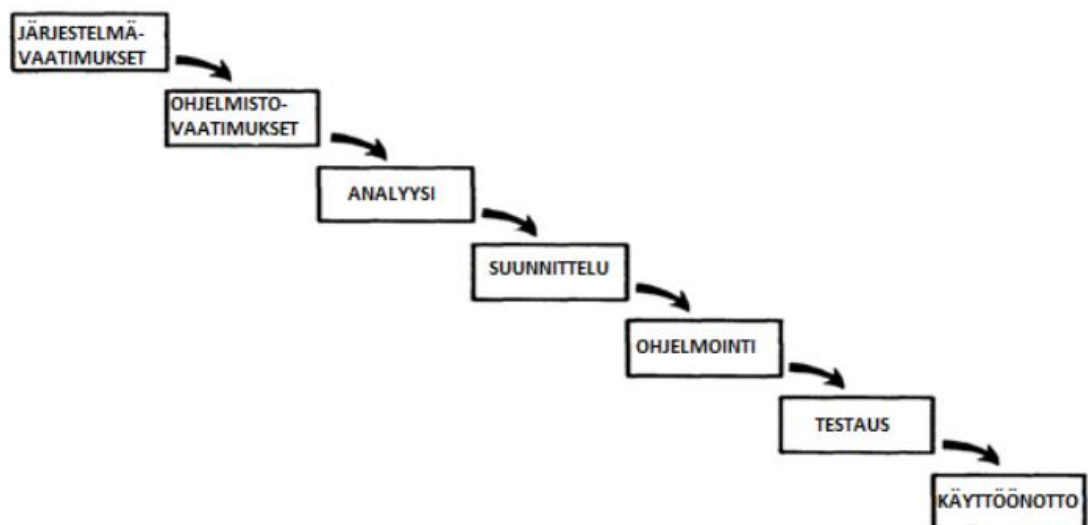
CM.SP 3.2 Suorita konfiguraationhallinnan lähtötilanteen auditointi

Ali-käytänteet:

1. Konfiguraationhallinta-alkioiden yhtenäisyyden ja riippuvuuksien arviointi.
2. Varmista, että konfiguraationhallinta aikakirja tunnistaa oikein hallinta-alkiot.
3. Varmista konfiguraationhallinta-alkioiden rakenne ha eheys.
4. Ohjelmiston konfiguraationhallinnan kohteiden täydellisyyden, oikeellisuuden ja johdonmukaisuuden varmistaminen. Täydellisyys, oikeellisuus ja johdonmukaisuus perustuvat vaatimuksista johdettuihin suunnitelmiin ja hyväksytyjen muutospyyntöjen toimittamiseen.
5. Sovellettavien kokoonpanonhallintastandardien ja -menettelyjen noudattamisen varmistaminen.
6. Ohjelmiston konfiguraationhallinta-alkioiden seuranta auditoinnista sulkemiseen.

3.5.3 Vesiputousmallinen mittalaitetuotanto

Yrityksen projektinhallinnan käytössä oleva vesiputousmalli toimii ohjelmisto- ja tuotekehityksen mallien pohjana. Sen historia ulottuu 1970 luvulla, kun Royce jaotteli ohjelmistoprojektin analysointi ja ohjelmointiosioon, ja vielä tarkemmin kuvassa esiintyvään seitsemään eri vaiheeseen: järjestelmävaatimukset, ohjelmistovaatimukset, analyysi, suunnittelu, ohjelmointi, testaus ja käyttöönotto.



KUVIO 3 - Vesiputousmalli (Royce, 1970).

Energianmittausjärjestelmän ja tuotteen kehitys etenee tuotteen Kuvion 5 elinkaarimallin mukaisesti, tuotteen elinkaarimallin on jaoteltu vaiheisiin ja välitavoitteisiin ja ohjelmistokehityksen scrum -aikataulut ovat sidottu sen päivämääriin.

3.5.4 Scrum -mallinen ohjelmistotuotanto

Yrityksen energiamittauksen tietojärjestelmä koostuu useasta ohjelma komponentista, eli ohjelmistosta. Ohjelmistokehityksen projektinhallinnan mallina käytetään scrum viitekehystä ja tätä viitekehystä pyritään noudattamaan yrityksessä. Scrum -mallisen ohjelmistokehityksen vaiheet ovat:

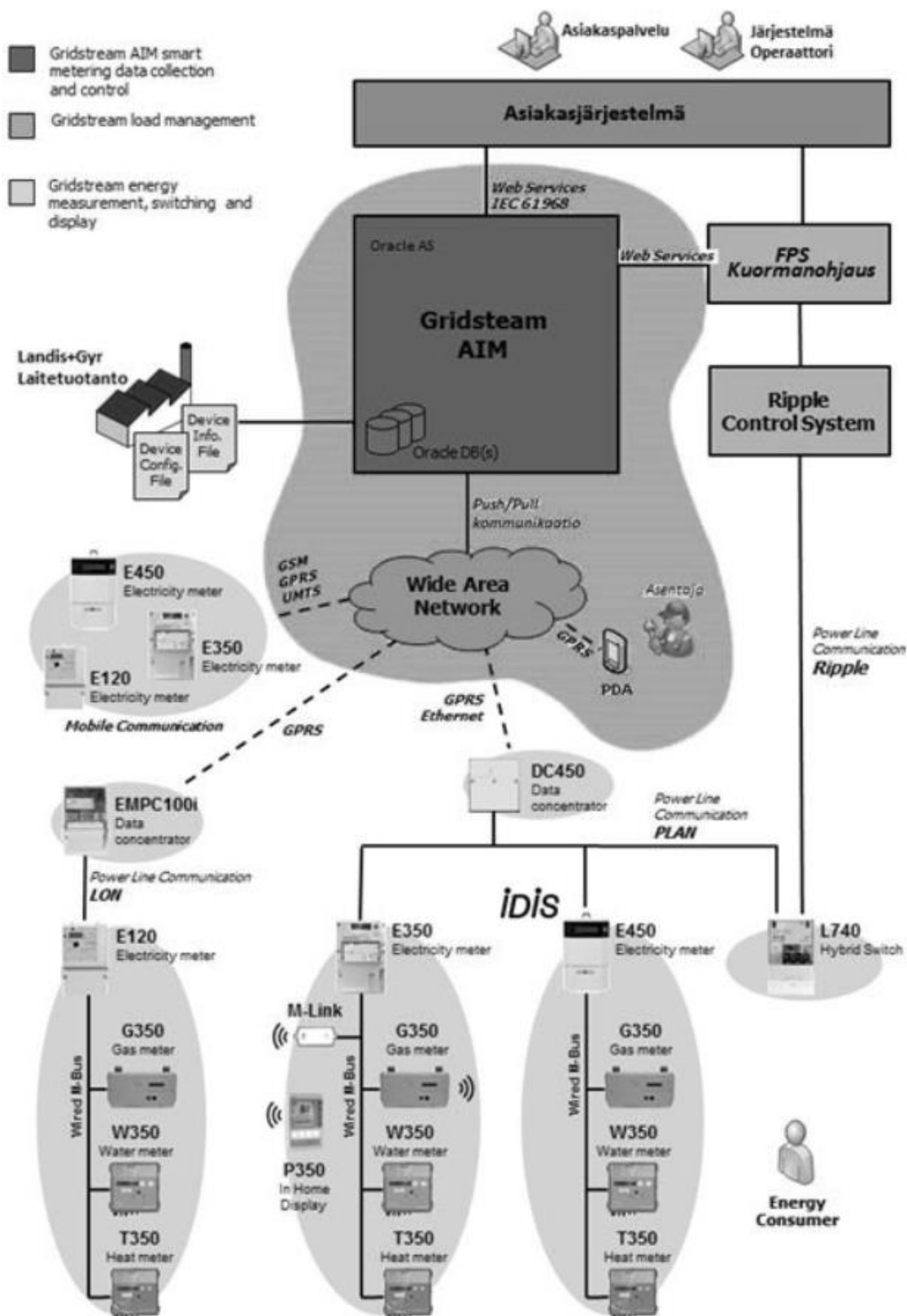
1. Vaatimusten määrittely (käyttäjätarinat, julkaisujen suunnittelu),
 2. Järjestelmien ja ohjelmistojen kehittäminen,
 3. Täytäntöönpano ja yksikkötestaus,
 4. Integraatio ja järjestelmän testaus (jatkuva integraatio ja hyväksyntätestaus),
 5. Ohjelmiston asennus sisäiselle asiakkaalle.
- (Huo, Verner, Zhu & Ali Babar, 2004, 5).

Scrum on kehys monimutkaisten tuotteiden kehittämiseksi ja ylläpidolle ja se tarjoaa puitteet, joissa ihmiset voivat käsitellä monimutkaisia sopeutumiso ongelmia ja tuottaa tuottavasti ja luovasti tuotteita, jotka ovat mahdollisimman hyviä. Scrum-kehys koostuu Scrum-tiimeistä ja niihin liittyvistä tehtävistä, tapahtumista, artefakteista ja sääntöistä. Scrum-tiimi koostuu tuotteen omistajasta, tuotekehitys tiimistä ja Scrum Masterista. Scrum-tiimit ovat itseorganisoituvia ja riskiin toimivia ja niiden itsejärjestäytyneet ryhmät valitsevat, miten ne voivat parhaiten suorittaa työnsä, sen sijaan, että heitä ohjataan muiden ulkopuolelta. (Schäfer & Sutherland, 2013, 4).

Kohdeyrityksen ohjelmistokehityksen scrum-mallissa tuotekehitys alueet jaotellaan tuotekehitystiimien kesken Scrum viitekehityksen mukaisesti. Kehitystiimit kertovat sprinttikatselmuksissaan sidosryhmilleen tuotteen kehityksestä ja tiimin saavutuksistaan, muokkaavat kehitysjonoaan saadun palautteen mukaisesti. Kohdeyrityksen ohjelmistokehitysmalli toimii joustavasti scrum-mallia noudattaen. Haasteita yrityksen scrum-mallin joustavuudelle aiheuttavat projektinhallinnan käytössä oleva vesiputousmallinen ohjelmisto- ja tuotekehitysmalli ja siihen sidotut kiinteät julkaisupäivämäärät.

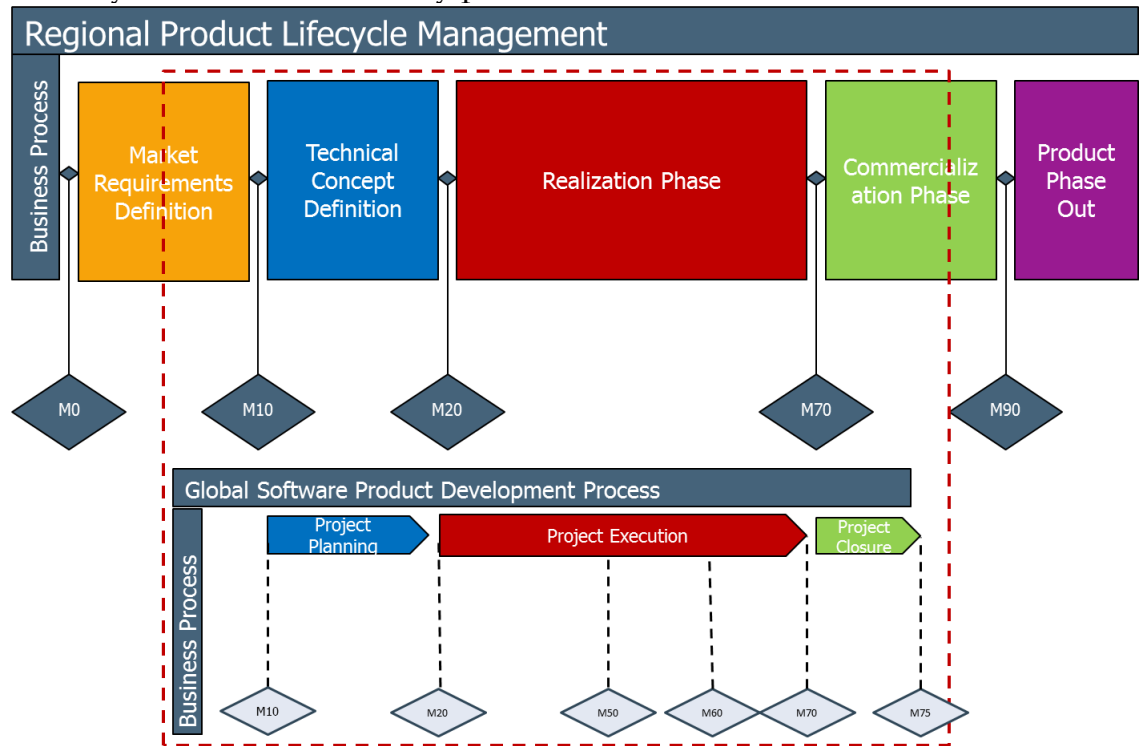
3.5.5 Järjestelmä ja ohjelmisto

Energianmittaus ja palvelut pitävät sisällään monimutkaisen kokonaisuuden; Itse energianmittauksen, tiedonsiirron, verkonhallinnan, ohjelmistot, kuluttajien energianhallintatyökalut ja palvelut). Tässä tutkielmassa keskityttiin energianmittauksen ohjelmistojen konfiguraatioiden hallinnan prosessien ja menetelmien tutkimiseen ja mittaustiedon tallennusohjelmistojen ja hallinnoinnin konfiguraatioihin. Tutkimusalue on rajattu tumman harmaalla alla Kuviossa 4.



KUVIO 4 - Energianmittausjärjestelmän kuvaus ja tutkimuksen rajaus tumman harmaalla (Landis Gyr Oy, 2017).

Ohjelmistotuotanto voidaan nähdä myös osana yrityksessä käytössä olevaa ohjelmistotuotteiden elinkaarenhallintamallia, malli on kuvattu Kuviossa 5. Kuviossa näkyy paikallinen tuotteen elinkaarimalli ja sen linkittyminen kansainväliseen tuotantoprosessiin on osoitettu katkoviivoin. Tutkielmassa käsiteltävä ohjelmistokehityksen vaihe on merkitty punaisella värillä.



Kuvio 5 - Tuotteen elinkaarimalli (Landis Gyr Oy, 2017).

3.5.6 Organisaation konfiguraationhallintaprosessit

Yrityksen eri osastoilla on käytössä monta eri ohjelmiston konfiguraationhallintaprosessia, tässä alaluvussa esitellään CMMI-malliin tukeutuvien ja tutkittujen prosessien käytänteet ja niihin liittyvät osastokohtaiset käytänteet, haasteet ja muut huomiot. Luvun lopussa ovat kuviot prosessien tilasta.

1. Tuotehallinnan prosessit

Kuten alla olevista Kuvioista 6-12 ilmenee tuotehallinta käyttää osittaista prosessia konfiguraationhallinnan tunnistamiseksi ja konfiguraationhallinnan menetelmän luomiseksi. Muutospyyntöjen seuranta ja alkioiden valvonta ovat vaivallisesti hallinnassa. Lähtötilanteen tunnistaminen ja konfiguraationhallinta-alkioiden valvonta puuttuvat kokonaan.

Käytänteet:

- Ohjelmiston Konfiguraationhallinta perustuu pitkälti dokumentaatioon, joka tuotetaan yhteistyössä tuotekehityksen kanssa.
- Asiakas saa vain tarvitsemansa dokumentit.

Haasteet:

- Ohjelmiston Konfiguraationhallintamenetelmä puute aiheuttaa ongelmia etenkin hajautettujen ympäristöjen hallinnassa.

Huomiot:

- Kehitystyön kohteena parhaillaan julkaisun mukana toimitettavat sovellusten 'OnLine Help':it.

2. Ohjelmistohallinnan prosessit

Kuten alla olevista kuvioista 6-12 ilmenee ohjelmistohallinta käyttää osittaista prosessi konfiguraationhallinnan tunnistamiseksi. Muutospyyntöjen seuranta ja alkioden valvonta ovat vaillinaisesti hallinnassa. Lähtötilanteen tunnistaminen ja konfiguraationhallinta-alkioden valvonta puuttuvat kokonaan.

Käytänteet:

- Rajapintamuutokset pyritään pitämään vähissä ja se auttaa ohjelmiston konfiguraationhallintaa osaltaan.

Haasteet:

- Ohjelmiston konfiguraationhallinnalle kokeiltu eri menetelmiä.

Huomiot:

- Wiki sivut ovat korvaamassa dokumentaation.

3. Ohjelmistokehityksen prosessit

Kuten alla olevista kuvioista 6-12 ilmenee tuotehallinta käyttää osittaista prosessia konfiguraationhallinnan tunnistamiseksi. Vaillinainen käytäntö on käytössä konfiguraationhallinnan menetelmän luomiseksi, lähtötilanteen tunnistamisessa, muutospyyntöjen seuraamisessa ja konfiguraationhallinta-alkioden valvonnassa. Konfiguraationhallinnan valvonta puuttuu kokonaan. ohjelmistokehitykseltä

Käytänteet:

- Kaikki kehitettävän ohjelmiston konfiguraatiot ovat kunkin kehittäjän räätälöimiä ja ohjelmiston konfiguraation hallinta-alkioiden rakenne on asennuksen jälkeen räätälöity ohjelmiston asentajan toimesta.

Haasteet:

- Kunkin kehittäjä kädenjälki näkyy siis ohjelmiston konfiguraationhallinta-alkioiden yhteen liitoksissa.

Huomiot:

- Korjaukset konfiguraatioissa ja niiden paremetroinnissa korjaavat aina vain tietyn ongelman.

4. Ohjelmistoarkkitehtuurin prosessit

Kuten alla olevista kuvioista 6-12 ilmenee tuotehallinta käyttää osittaista prosessia konfiguraatiohallinnan tunnistamiseksi. Vaillinainen käytäntö on käytössä konfiguraatiohallinnan menetelmän luomiseksi, lähtötilanteen tunnistamisessa. Muutospyyntöjen seuraamisessa, konfiguraatiohallinta-alkioiden valvonnassa ja konfiguraatiohallinnan aikakirjan perustamiseksi puuttuvat kokonaan ohjelmistoarkkitehtuurin prosesseista.

Käytänteet:

- Osastolla ei ole omaa ohjelmiston konfiguraatiohallintaprosessia.

Haasteet:

- Ohjelmiston konfiguraatiohallintaa on tutkittu ja ohjelmiston konfiguraatiohallinnan jatkokehitys kohdistetaan avoimen lähdekoodin SaltStackiin (SaltStack, 2017.).

Huomiot:

- Konfiguraatiot olisi hyvä saada tietokantaan.

5. Jatkuvan integroinnin ja toimituksen prosessit

Kuten alla olevista kuvioista 6-12 ilmenee jatkuva integrointi ja toimituksen prosesseissa käytössä olevan toimivat prosessit konfiguraatiohallinnan tunnistamiseksi ja menetelmän luomiseksi. Vaillinainen käytäntö on käytössä, lähtötilanteen tunnistamisessa. Muutospyyntöjen seuraaminen, konfiguraatiohallinta-alkioiden valvonta ja konfiguraatiohallinnan aikakirja puuttuvat kokonaan jatkuvan integroinnin ja toimituksen prosesseilta.

Käytänteet:

- Osastolla on käytössään ohjelmiston asennusparametrien tallentamiseen .xml -skriptejä. Skriptit suoritetaan Jenkins (Jenkins, 2017) automaatiopalvelimella. Ohjelmiston hallinta alkioiden tallennus suoritetaan ennen asennusta.
- Edellisten skriptien avulla hallittujen asennusparametrien ylläpito ja ohjelmistojulkaisujen jakelun on todettu osaston rajapintojen ja sisäisen auditoinnin puolesta ja yrityksen johdon mukaan toimivan hyvin integraatiotestauksessa (smoke-testaus).

Haasteet:

- Tiimin käyttämät asennusympäristöt ovat staattisia.

Huomiot:

- -

6. Ohjelmiston integroinnin prosessit

Kuten alla olevista kuvioista 6-12 ilmenee jatkuva ohjelmiston integroinnin prosesseissa käytössä olevan toimivat prosessit konfiguraationhallinta-alkioiden tunnistamiseksi, menetelmän luomiseksi, lähtötilanteen tunnistamiseksi ja muutospyyntöjen seuraamiseksi. Vaillinainen käytäntö on käytössä alkioiden valvonnassa. Prosessi konfiguraationhallinnan aikakirjaa varten puuttuu kokonaan.

Käytänteet:

- Ohjelmiston konfiguraationhallintaa hoitamaan on luotu prosessi (LIITE 3). Empiirisen tutkimuksen ja asiantuntijahaastattelujen perusteella ja yrityksen johdon mukaan prosessi toimii hyvin ohjelmistojulkaisujen integraatiotestauksessa (smokki -testaus).
- Manuaalit ja ohjeet ovat tärkeässä asemassa.

Haasteet:

- Mittalaitteiden konfiguraationhallinta kaipaa omaa menetelmää.

Huomiot:

- -

7. Asiakaspalvelun ja tuen prosessit

Käytänteet:

- Sovelluksen päivitys: Asiakasympäristöjen ohjelmiston konfiguraationhallinta hoidetaan kopioimalla ohjelmiston konfiguraatio-tiedosto-kansiot eri sijaintiin ennen sovelluksen päivitystä.
- Sovelluksen uusi asennus: Konfiguraationhallinta hoidetaan asennusmanuaalien ja kunkin asentajan hiljaisen tiedon ja räätälöinnin avulla.
- Erikoisympäristön asennus: Ohjelmiston konfiguraationhallinta hoidetaan asentajan räätälöimillä hallinta-alkioilla ja osittain tuotekehityksen tuella.
- Huoltopaketin asennus: Asennus hoidetaan huoltopaketin mukana toimitettavalla asennusdokumentaation avulla.
- Kiireellisen korjaus -paketin asennus: Asennus hoidetaan osin puutteellisten asennusohjeiden mukaisesti.

Haasteet:

- Koko asennuskansioiden kopioiminen raskas operaatio.
- Asiakaskohtaiset konfiguraatiotiedostojen räätälöinnit ovat harmaata aluetta.
- Kiireellisten korjausten asennusohjeet ovat osittain puutteellisia.

Huomiot:

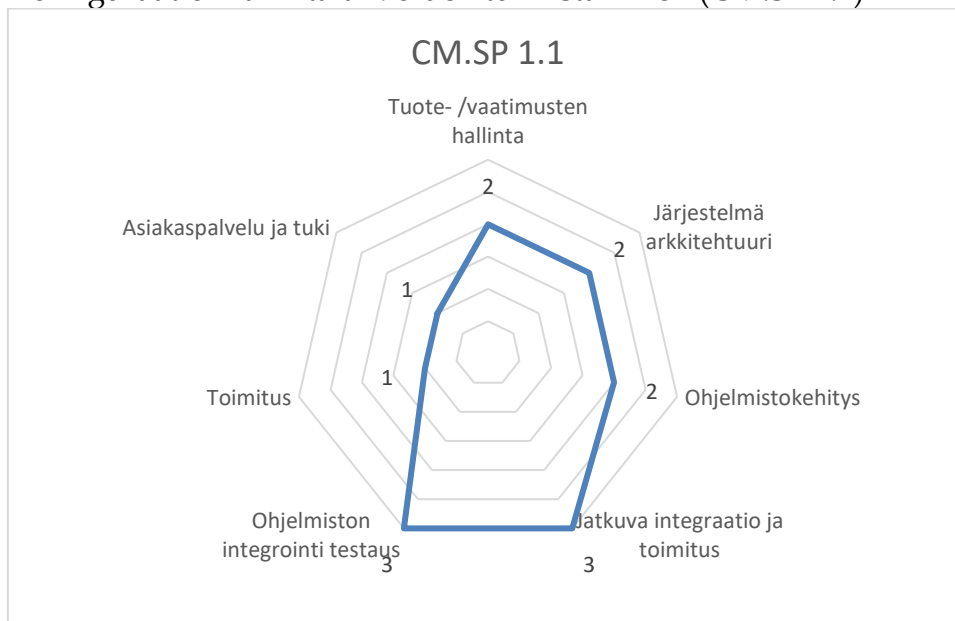
- Konfiguraationhallinta osittain hiljaisen tiedon varassa.

Osastojen prosesseja analysoitiin teemoittelun avulla, teemoittelussa tutkimusongelma tyypiteltiin CMMI käytänteiden mukaisiksi aihepiireiksi. Tyypittelyn avulla haastatteluista saatiin koostettua kohdeyrityksen asiantuntijoiden eri prosessien nykytila. Tyypittelyä jatkettiin haastattelujen aikana pilkkomalla CMMI-

käytänteitä alikäytänteisiin. Haastattelun tulokset luokiteltiin ja teemoitettiin jo haastattelu vaiheessa CMMI käytänteiden mukaisesti. Haastattelujen jälkeen tulokset tarkistettiin ja korjattiin luokittelujen mukaisesti. Kohdeyrityksen ohjelmistojen- konfiguraationhallintaprosesseja vertailtiin CMMI käytänteiden viitemalleihin. Viitemallit valikoituivat vertailukohtaksi koska CMMI on viitekehys, joka auttaa organisaatioita järjestelmäkehityksen parhaiden käytäntöjen käyttöönotossa (Chrissis ym., 2011, s. 18.) ja CMMI määrittelee mitä tulee tehdä, eikä niinkään, kuinka asiat tehdään. Mallin prosessit ovat hyvin määriteltyjä ja dokumentoituja toimintoja. (O'Regan, 2011, s. 46.). Viitemallien vertailujen ja sen jälkeisessä kohdeyrityksen ohjelmiston konfiguraationhallinnan prosessien tulkinnaassa osastojen prosesseille annettiin arvot 0-4 (0=Puuttuva käytänne, 1= Vaillinaisen käytänne, 2=Käytänne osittain käytössä, 3= Prosessi käytössä tai valmistumassa 4=Toimiva prosessi käytänteen hoitamiseksi) (Kuviot 6 - 12). Osastojen prosessien tarkempi kuvaus on jätetty tutkimuksen ulkopuolelle.

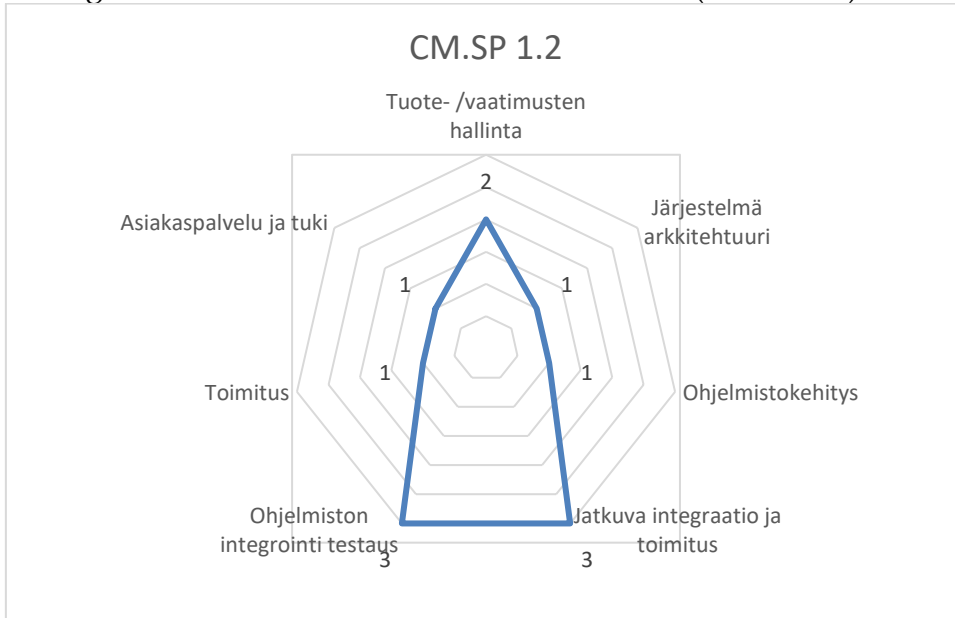
Kuvioissa alla on näkyvissä eri osastojen prosessien tulokset verrattuna CMMI käytänteisiin.

Konfiguraationhallinta-alkioiden tunnistaminen (CM.SP 1.1)



KUVIO 6 - Konfiguraationhallinta-alkioiden tunnistaminen CM.SP 1.1

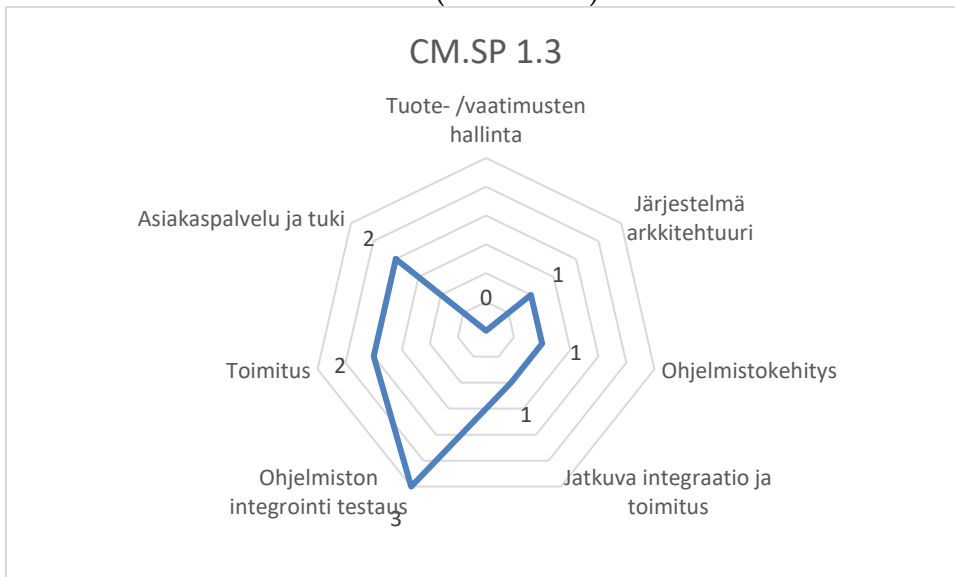
Konfiguraationhallinnan menetelmän luominen (CM.SP 1.2)



KUVIO 7 - Konfiguraationhallinnan menetelmän luominen (CM.SP 1.2)

Kuten Kuvio 7 osoittaa, konfiguraationhallinnan menetelmä on vaillinaisesti käytössä tuote- ja vaatimustenhallinnalla, käytössä ohjelmistokehityksen loppuvaiheissa integrointi testauksessa, sekä jatkuvassa integroinnissa ja toimituksessa. Tästä voidaan päätellä ohjelmiston konfiguraationhallinnan menetelmän kehitystyön olevan yrityksessä alkuvaiheessa.

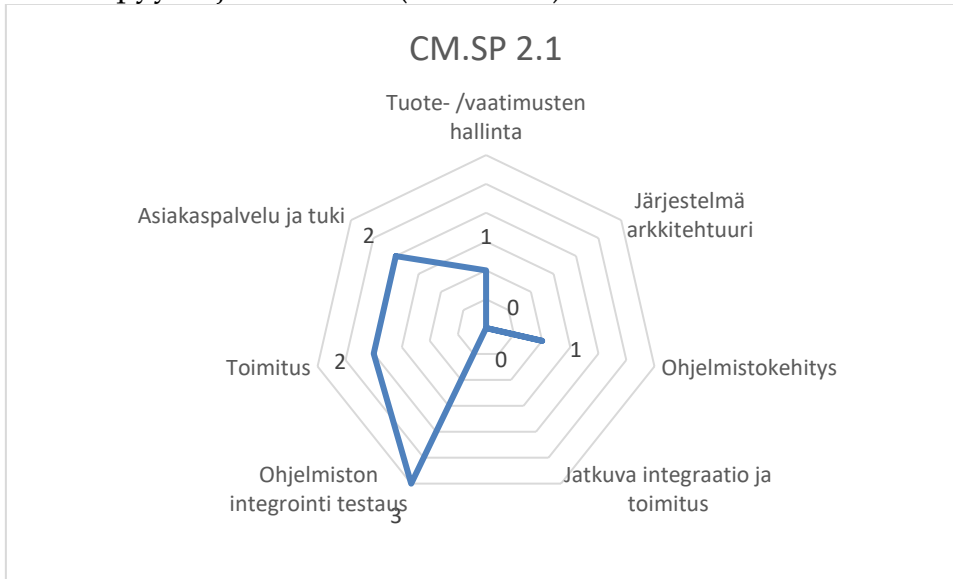
Lähtötilanteen tunnistaminen (CM.SP 1.3)



KUVIO 8 - Lähtötilanteen tunnistaminen (CM.SP 1.3)

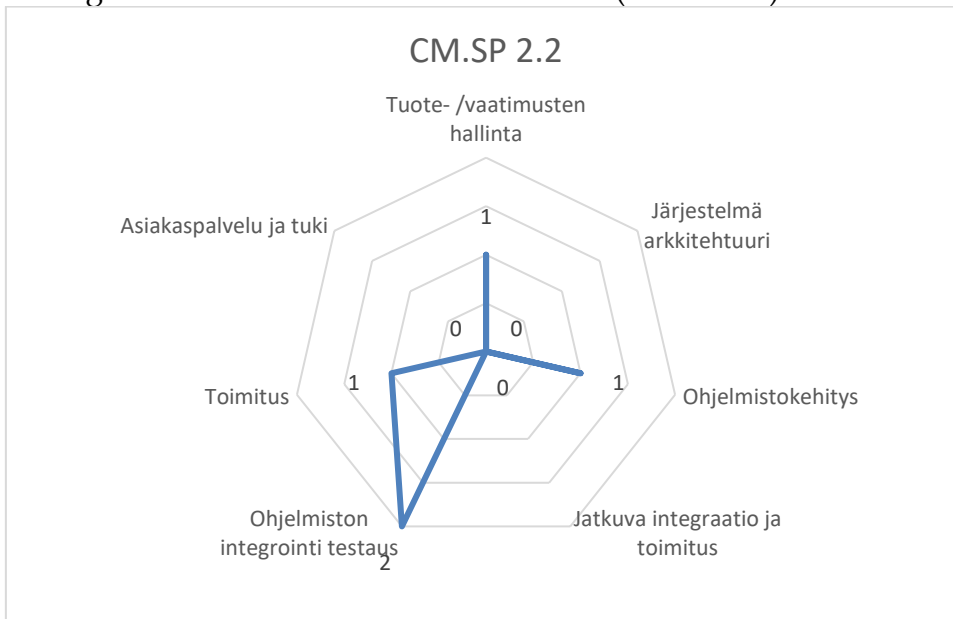
Lähtötilanteen tunnistaminen, eli hallinta-alkioiden määrittäminen ja hallinta ovat vähintään osittain käytössä asiakaspalvelussa ja tuessa, sekä tuotetoimituksissa. Ohjelmiston integrointi testauksessa lähtötilanne tunnistetaan ympäristökohtaisesti. Lähtötilanteen tunnistaminen on yrityksen johdon mukaan tärkeä vaihe.

Muutospyyntöjen seuranta (CM.SP 2.1)



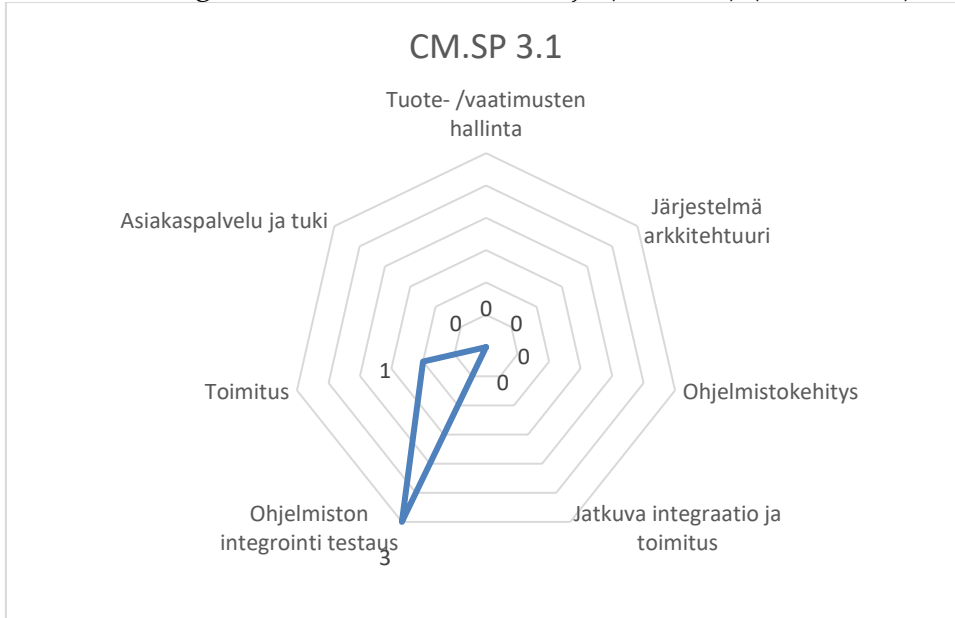
KUVIO 9 - Muutospyyntöjen seuranta (CM.SP 2.1)

Konfiguraationhallinta-alkioiden valvonta (CM.SP 2.2)



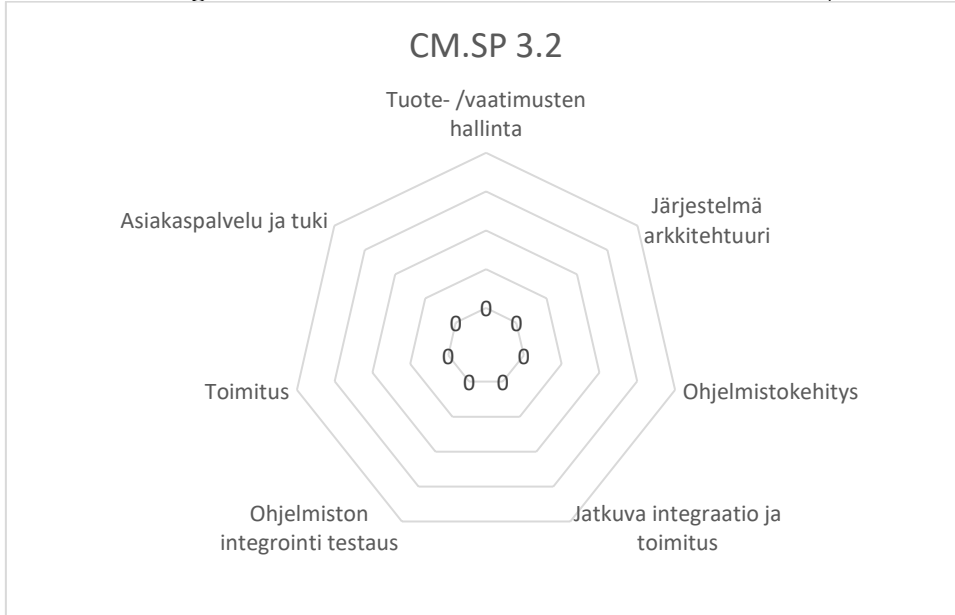
Kuvio 10 - Konfiguraationhallinta-alkioiden valvonta (CM.SP 2.2)

Perusta konfiguraationhallinnan aikakirja (Records) (CM.SP 3.1)



KUVIO 11 - Perusta konfiguraationhallinnan aikakirja (Records) (CM.SP 3.1)

Suorita konfiguraationhallinnan lähtötilanteen auditointi (CM.SP 3.2)



KUVIO 12 - Suorita konfiguraationhallinnan lähtötilanteen auditointi (CM.SP 3.2)

3.5.7 Ehdotettu menetelmärunko ja analyysi

Alaluvun 3.4.6 tulosten perusteella kehitettiin uusi konfiguraationhallinnan selvitys. Tässä alaluvussa esitellään selvitys ohjelmiston konfiguraationhallinnan

tutkimuksesta, ohjelmiston konfiguraationhallinnan menetelmärunko ja asiantuntijahaastatteluista kerätyt jatkokehityskohteet. Tämä menetelmän runko luotiin yhdistämällä yrityksen hajanaiset ohjelmiston konfiguraationhallintaprosessit, aiemmin tutkitut teemat ja empiirisen tutkinnan avulla kerätty tieto CMMI käytänteisiin (Wibas, 2015).

Tutkimuksen tavoitteena oli löytää vastaus siihen, miten kohdeyrityksen ohjelmiston konfiguraationhallintaa tulisi kehittää. Tarkoituksena oli hajanaisten toimintatapojen kartoittaminen ja niiden tallentaminen. Empiirisen tutkinnan ja asiantuntijahaastattelujen perusteella luotiin ensimmäinen runko yrityksen konfiguraationhallinnan menetelmälle. Lisäksi menetelmän rungon vaiheiden varmistamiseksi luotiin strukturoimaton kysely jollakerättiin lisätietoja kohdeyrityksen ohjelmiston konfiguraationhallinnan kehityskohteista. Kyselyn vastauksen analysoinnin jälkeen valmistui ensimmäinen ehdotettu selvitys menetelmästä. Ehdotuksessa vaiheisiin liittyvät oleellisesti kappaleen 3.4.2 CM.SP käytänteen ja alikäytänteet. Alla kuvattuna luotu selvitys.

Ohjelmiston konfiguraationhallinnan menetelmärunko vaiheineen ja asiantuntijahaastatteluiden ja strukturoimattoman kyselyn perusteella kerätyt jatkokehityskohteet:

Vaihe 1: CM.SP 1.1 Konfiguraationhallinta-alkioiden tunnistaminen

Ohjelmiston konfiguraationhallinta-alkioiden tunnistaminen tehdään yhteistyössä yrityksen tiimien kanssa, muuttuneet tai lisätyt konfiguraationhallinta-alkiot ilmoitetaan aina konfiguraationhallinnasta vastaavalle.

Tunnistamisessa on otettava huomioon seuraavat ominaisuudet, seikat ja tavat:

- Hallinta-alkioiden ollessa käytössä kahdella tai useammalla ryhmällä vaatii hallinta-alkioiden tarkkaa versionhallintaa ja valvontaa.
- Hallinta-alkioiden joiden odotetaan muuttuvan jatkossa virheiden korjausten tai muuttuvien vaatimusten johdosta.
- Toisistaan riippuvaiset hallinta-alkiot (yhden alkion muutoksen vaikuttaessa muihin alkioihin)
- Ohjelmistolle kriittisten hallinta-alkioiden tunnistaminen

Vaihe 2: CM.SP 1.2 Konfiguraationhallinnan menetelmän luominen

Ali-käytänteistä osa on toteutettu kohdeyrityksen eri osastoilla. Lähinnä tämän käytänteen toteutus hoidetaan versionhallinnalla, konfiguraatiodokumenttien versionhallinnan käytössä ovat käyttäjäoikeuksien hallinta, ympäristökohtainen hallinta-alkioiden tallennus, noutaminen ja mahdollisuus palautukseen. Täysin ulkopuolelle ovat jäänyt yrityksen laajuinen kokonais-konfiguraationhallinta ja suuri osa alikäytänteistä. Menetelmän luonnissa otettava huomioon alikäytänteet ja seuraavat seikat:

- Loppukehityksen käytössä olevaa osittaista konfiguraationhallintamenetelmää ylläpidetään ja liitetään yhteen tuotehallinnan ja ohjelmistohallinnan sekä muutoksenhallintajärjestelmien kanssa, jotta voidaan parantaa projektien hallinta-alkioita ja itse tuotteen konfiguraatioita.
- Ohjelmiston konfiguraationhallintaan luotava yksi ohjaava menetelmä ja yhtenäiset prosessit. Eri osastojen tarpeet huomioitava.

Vaihe 3: CM.SP 1.3 Lähtötilanteen tunnistaminen

Kohdeyritys tukeutuu kehitystiimien ja lähtötilanteisiin, sisältäen järjestelmän vaatimusten noudattamisen ja määritysten kirjaamisen dokumentaatioon. Ohjelmiston konfiguraationhallinta tukeutuu näihin samoihin vaatimukseen ja dokumentaatioon. Alla seikkoja ja tapoja konfiguraationhallinnan jatkokehitykselle.

- Sisäisen lähtökohdan julkistaminen pitää sisällään ohjelmiston konfiguraationhallinta-alkioiden tunnistamisen eri tarpeeseen. (esim. tuotekehitys ja testaus).
- Ulkoiset lähtökohdat tunnistetaan asiakaskohtaisesti.
- Yhteinen perusviitekehys sisältää järjestelmätason vaatimukset, järjestelmän elementtien tason suunnittelumääritykset ja tuotemäärityksen tuotekehitys vaiheen lopussa tai asiakasjakelujen alussa.
- Ohjelmiston konfiguraationhallinnan kuvauksen luonti ja konfiguraationhallinnan vastuuhenkilöiden ja tehtävien määrittely.
- Konfiguraationhallinnan standardointi yrityksen sisäiseksi ja eri prosessien standardiksi parantaa tuotteen laatua.
- Konfiguraationhallinta ulotettava koko yrityksen parhaimpien menetelmien työkalupakkiin.
- Standardoidun menetelmän lisäksi tulee tapauksia jotka hoidettava muulla tavoin.
- Tämänhetkiset konfiguraationhallinnan käytännön toimet otettava mukaan standardointiin, kuten millaiset hallinta-alkioiden muutokset ovat sallittuja.

Vaihe 4: CM.SP 2.1 Muutospyyntöjen seuranta

Ohjelmiston konfiguraationhallinnan kannalta, muutoksia havaitaan hallinta-alkioiden muutoksissa.

- Muutospyynnöt koskevat paitsi uusia tai muuttuneita vaatimuksia, myös julkaisujen vikoja ja puutteita. Muutospyynnöt analysoidaan sen määrittämiseksi, miten muutos vaikuttaa julkaisuihin, siihen liittyviin työsuorituksiin, budjettiin ja aikatauluun
- Muutospyyntöjen ennalta testaus suoritettava ennen muutosten julkaisua.

- Muutospyyntöjen aiheuttamien muutosten tiedottamista ja oheistamista tehostettava.
- Asiakaskohtaisten muutosten vaikutus kokonaisjulkaisuun ja muiden asiakkaiden tarpeisiin.
- Ohjelmiston ja sen konfiguraatioiden toiminta turvattava järjestelmäpäivitysten jälkeen. Olemassa oleva toiminnallisuus turvattava.
- Asiakaskohtaiset vaatimukset ja niistä johtuvat ohjelmistojen räätälöinnit lisäävät ylläpidon kuormaa.
- Muutosten hallinnalle perustettava ryhmä, joka vastaa muutospyyntöjen ristiriitaisuuksien ratkaisusta. Ryhmä päättää hyväksytyistä /suositelluista, kielletyistä konfiguraatioista ja mahdollisista tuotemuutoksista.

Vaihe 5: CM.SP 2.2 Konfiguraationhallinta-alkioiden valvonta

Kohdeyrityksen käytössä ei ole hyvää hallinta-alkioiden valvontatyökalua, jolla otettaisiin huomioon alikäytänteet ja tässä tutkimuksessa selvinneet kehityskohteet:

- Ohjelmistokehityksen aikana määritettävä raamit ohjelmiston konfiguraatioiden rajoitetulle muokkautuvuudelle.
- Asiakkaan tekemille ohjelmiston konfiguraatiomuutoksille määriteltävä rajat, jotta välttyään asiakkaan tekemien konfiguraatiomuutosten yhteensopivuusongelmilta.
- Muutosten läpikäynti ja hyväksyntä vaativat arkkitehti ja tuotteenomistaja tasoisiin jäseniin. Yhteys ylläpidon kanssa säilytettävä, asiakkaiden tarpeiden kartoittamiseksi.
- Hallinta-alkioiden valvonta perustuu tuotteiden lähtökohtien ja peruskokoonpanon mukaisesti. Valvontaan kuuluu uusien hallinta-alkioiden kokoonpanojen hyväksymiset ja tarvittaessa lähtökohtien päivittämiset.

Valvonnan työkalut ovat

- i. Ohjelmiston konfiguraationhallinta-alkioiden muutoshistoria tallentaminen.
- ii. Lähtötilanteiden arkistot.

Vaihe 6: CM.SP 3.1 Perusta konfiguraationhallinnan aikakirja (records)

Kohdeyritykseltä puuttuu ohjelmiston konfiguraationhallinnan aikakirja. Alla käytänteisiin lisättyinä tutkitut ja ehdotetut tavat ohjelmiston konfiguraationhallinnan aikakirjan luomiseksi:

Ohjelmiston konfiguraationhallinnan aikakirjan perustaminen ja sen ylläpito:

Aikakirjan perustamisen työkalut ovat:

- i. Hallinta-alkioiden muutoshistoria
 - ii. Muutos loki
 - iii. Muutospyyntöjen kopiot
 - iv. Hallinta-alkioiden tila
 - v. Erot lähtökohtien välillä
- Ohjelmiston asiakaskohtaiset konfiguraatiot kerättävä keskitettyyn paikkaan.
 - Tärkeässä roolissa on ohjelmiston konfiguraatioiden säilyttäminen käytänteet järjestelmäpäivitysten yhteydessä.
 - Ohjelmiston konfiguraatioiden muutosten identifioiminen mahdollistettava ja edellisen ohjelmiston konfiguraation palauttaminen mahdollistettava.

Vaihe 7: CM.SP 3.2 Suorita konfiguraationhallinnan lähtötilanteen auditointi
Kohdeyritykseltä puuttuu ohjelmiston konfiguraationhallinnan auditointiprosessi. Alla haastattelusta poimittu seikka:

- Ohjelmiston konfiguraationhallinnan lähtötilanteen auditointi varmistaa lähtökohtien eheyden. Lähtökohtien konfiguraatioiden tarkastukset tulisi ulottaa koskemaan myös muita yrityksen tietolähteitä, jotta varmistetaan konfiguraationhallinnan tarkkuus ja yhdenmukaisuus.

Tässä tutkimuksessa kuvatut kohdeyrityksessä käytössä olevat ohjelmiston konfiguraationhallinnan osaprosessit osoittavat ohjelmiston konfiguraationhallinnan olevan osittain käytössä ohjelmistokehityksen loppuvaiheen testauksissa ja kehityksessä ja osan prosesseista toimivan yrityksen johdon mielestä hyvin (Kuviot 6-12 alaluvussa 3.4.6 Organisaation konfiguraationhallintaprosessit.), varsinaista hallittua konfiguraationhallintaa ei ole nähtävissä. Ohjelmistokehityksen aiemmissa vaiheissa ohjelmiston konfiguraationhallintaa varten on kehitetty eri menetelmiä ja todettu kehittäjien omien räätälöintien toimivan parhaiten.

Tutkimuksen lopulliseksi esitykseksi muodostui CMMI käytänteiden, ohjelmistokehityksen loppuvaiheessa hyviksi todettujen vaiheiden, haastatteluista ja kyselystä saatujen arvokkaiden tietojen sekä empiirisen tutkinnan avulla kirjoitettu ohjelmiston konfiguraationhallinnan menetelmärunko kohdeyritykselle.

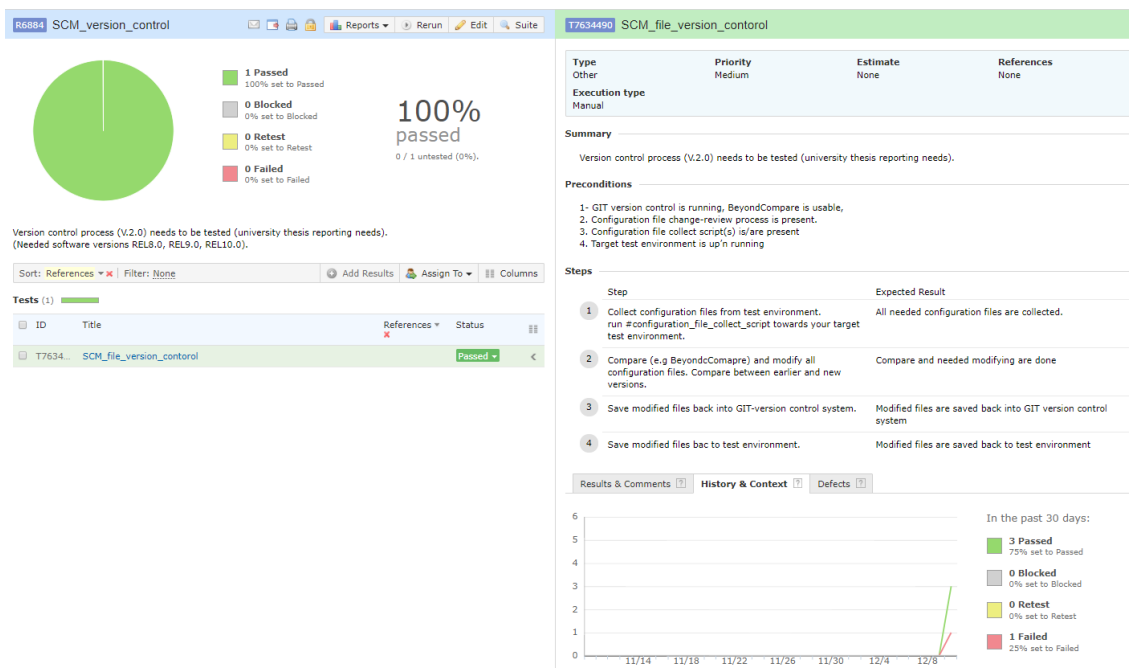
4 DEMONSTRAATIO JA MENETELMÄRUNGON ARVIOINTI

Tässä luvussa esitetään kirjallisuuskatsauksen, asiantuntijoiden haastatteluiden, empiirisen tutkimisen ja strukturoimattoman kyselyn avulla tuotettu ohjelmiston konfiguraationhallinnan menetelmärunko, arvioidaan sen toimivuutta sekä analysoidaan sen kehityskohteita tutkielmalle asetettujen tutkimusongelmien näkökulmista. Tässä tutkimuksessa kirjoitettu selvitys ohjelmiston konfiguraationhallinnasta esittelee CMMI käytänteiden mahdollisuuden toimia ohjelmiston konfiguraationhallintamenetelmän kehittämiskokona.

4.1 Menetelmärungon demonstraatio ja versionhallinnan prosessin testaus

Ohjelmiston konfiguraationhallinnan menetelmärungon testaus oli käytännössä mahdotonta kohdeyrityksen monien ohjelmiston konfiguraationhallinnan osaprosessien kehityskohteiden vuoksi. Tutkimuksen aikana hyväksi todettua loppukehityksen hallinta-alkioiden versionhallinnan prosessia testattiin ohjelmiston sisäisten jakelujen osalta, käytössä olevalla prosessilla (liite 3), tässä tutkimuksessa havaitut kehityskohteet jätettiin testauksen ulkopuolelle.

Kohdeyrityksen ohjelmiston konfiguraationhallintaan liittyvät versionhallinnan prosessit toimivat hyvin suljetuissa testiympäristössä ohjelmistojakelun kehittäjien ja suunnittelijoiden ollessa läheisessä yhteistyössä jakelun asennuksen ja ohjelmiston konfiguraationhallinnan tehtävien kanssa. Prosessin testaus kohdistettiin kolmeen jo julkaistuun ohjelmistojakeluun, niiden eri vaiheissa oleviin testiympäristöihin. Alla kuvattuna (kuvio 13) yhden ohjelmistojakelun versionhallinnan testiraportti.



KUVIO 13 - Versionhallintaprosessin testiraportti.

Versionhallinta prosessin testausvaiheen esi-ehtoihin kuuluivat: GIT versionhallinta-ohjelmiston käyttöönotto ja käyttöoikeuksien varmistaminen, Muutottenhallinnan prosessin tutustuminen. Konfigurointitiedostojen kopiointi skriptien muokkaaminen testiympäristölle sopivaksi. Testausvaihees suoritus:

1. Testiympäristön hallinta-alkioiden kopioiminen ja tallettaminen ulkoiseen sijaintiin.
2. Hallinta-alkioiden vertailu aiempiin versioihin ja muokkaaminen.
3. Talletettujen (ja muokattujen) hallinta-alkioiden talletus GIT versiohallintaan.
4. Muokattujen hallinta-alkioiden talletus testiympäristöön.

4.2 Arviointi

Tutkimuksen lähtökohtana oli energia-alan yrityksen ohjelmiston konfiguraationhallinnan tilan selvittäminen ja tarve luoda ohjelmiston konfiguraationhallinta menetelmälle runko. Yrityksen tämänhetkinen ohjelmiston konfiguraationhallinnan selvitys toi esille yrityksen moninaiset eri tavat hoitaa ohjelmiston konfiguraatioita. Menetelmien hajanaisuus ja moninaisuus lisäsivät tarvetta ohjelmiston konfiguraationhallintamenetelmän rungon kehittämiseksi. Tässä tutkimuksessa esitelty menetelmärunko kohdistettiin kohdeyrityksen tarpeeseen.

Tutkimus toteutettiin DSRM (Peffer's ym., 2007) menetelmää noudattaen. Kohdeyrityksen ohjelmiston konfiguraationhallinnan tutkimiseen DSRM:n vaiheistus sopi hyvin. Tutkimuksen lopulliseksi selvitykseksi muodostunut *'ohjelmiston konfiguraationhallinta menetelmä ja sen jatkokehityskohteet'*, selvitys tuotettiin vaiheittain kolmella iteraatiokierroksella.

Ongelman tunnistus ja motivointi vaihe paljasti tarpeen ohjelmiston konfiguraationhallinnan tutkimuksen jatkamiselle ja vaihe lomittui kiinteästi toisen ratkaisun tavoitteiden määrittelyvaiheen kanssa. Selvityksen suunnittelu ja kehittämisvaihe piti sisällään empiiristä tutkintaa, asiantuntijahaastatteluja, aiempien tutkimusten läpikäyntiä ja arviointia, tämän vaiheen päätökseksi tuotettiin ensimmäinen ehdotelma ohjelmiston konfiguraationhallinta selvityksestä. Luotua selvitystä demonstroitiin kohdeyrityksessä määriteltyjen tutkimuskysymysten kontekstissa. Toimivuus testattiin kohdeyrityksen ohjelmiston sisäisten jakelujen ja yrityksessä aiemmin tuotetun versionhallintaprosessin osalta, tutkimusta varten koostettiin versionhallinnalle hallinta-alkioiden osajoukko ja versionhallintaprosessia suoritettiin (LIITE 3). Esitetyn selvityksen ja ratkaisuehdotuksen demonstraatioissa esiin tulleet seikat otettiin huomioon selvityksen rakenteessa, muutosten jälkeen ehdotettu selvitys otettiin hyvin vastaan yrityksessä.

Arviointivaiheessa yrityksen johdolle ja asiantuntijoille toimitettiin strukturoimaton kysely, jolla pyrittiin selvittämään yksityiskohtaista tietoa yksittäisestä tapauksesta tai pienestä joukosta toisiinsa suhteessa olevia tapauksia. Arviointivaiheen jälkeen järjestettiin yrityksen johdolle suunnatulla demonstraatio, johdon mukaan ehdotettuun menetelmään kirjatut kehityskohteet voidaan hyväksyä, kerätystä aineistosta johdetut menetelmän kehityskohteet näkyvät taulukossa 3.

TAULUKKO 3: Ehdotetun menetelmän kehityskohteet

CMMI Käytänne	Tämänhetkinen toteutus	Kehityskohteet
CM.SP 1.1 Konfiguraationhallinta-alkioiden tunnistaminen	Osittain kunnossa tuotekehityksen loppuvaiheissa	Hallinta-alkioiden tunnistamista kehitettävä. Alkioiden priorisointia kehitettävä Hallinta-alkioiden linkityksen ja vaikutukset muihin alkioihin tunnistettava. Tuotekehityksen vaatimusten vaikutu alkuihin tunnistettava.
CM.SP 1.2 Konfiguraationhallinnan menetelmän luominen	Menetelmää ei ole, tutkittu muutamia vaihtoehtoja.	Yritykselle luotava menetelmä ja prosessi(t). Menetelmän rungoksi voidaan ottaa jo tutkittu

		SaltStackkonfigurointi-ohjelmisto.
CM.SP 1.3 Lähtötilanteen tunnistaminen	Lähtötilanne osittain luotu tuotteen integrointitestauksessa.	Lähtökohdan tunnistamisprosessi luotava.
CM.SP 2.1 Muutospyyntöjen seuranta	Muutospyyntöjen hallinta toimii asiakkaan suuntaan, muutospyyntöjen vaikutukset eivät välity kaikille tarvitseville.	Muutospyyntöjen aiheuttamat muutoksista tiedottamista tarkennettava.
CM.SP 2.2 Konfiguraationhallinta-alkioiden valvonta	Alkioita valvotaan versiohallinnan avulla tuotekehityksen loppuvaiheessa.	Alkioiden valvonnalle luotava prosessi, joka ulotettava koko tuotekehityksen käytätän.
CM.SP 3.1 Perusta konfiguraationhallinnan aikakirja (records)	Aikakirjaa ei ole	Aikakirja luotava
CM.SP 3.2 Suorita konfiguraationhallinnan lähtötilanteen auditointi	Auditointeja ei suoriteta	Lähtötilanteen auditointiprosess luotava.

Arviointi vaiheen jälkeen tutkimuksen tulokset kommunikointiin kohde yritykselle ja tiedeyhteisölle.

5 YHTEENVETO JA JOHTOPÄÄTÖKSET

Tässä tutkimuksessa tarkasteltiin energiamittauksen ohjelmiston konfiguraationhallintaa ja sen kehityskohteita. Tutkimuksen tuotoksena tuotettiin selvitys ja kirjoitettiin menetelmärunko kohdeyrityksen ohjelmiston konfiguraationhallinnalle aiempien tutkimusten, empiirisen tutkinnan, asiantuntijahaastatteluiden ja survey-tutkimuksen avulla. Aiempiin tutkimuksiin yhdistettiin empiirinen tutkimus. Teemojen ja kohdeyrityksessä käytössä olevan CMMI mallin mukaan koottiin asiantuntijahaastattelu.

Tässä tutkimuksessa tutkittiin, kuinka aiemmin tutkittuja ohjelmiston konfiguraationhallinnan käytänteitä voidaan hyödyntää energiamittauksen ohjelmiston evoluutioiden hallinnassa ja CMMI-käytänteiden sisältämien ohjelmiston konfiguraationhallinnan käytänteiden soveltuvuutta kohdeyrityksen ohjelmiston konfiguraationhallinnan menetelmän rungoksi. Tutkimusongelmaa selvitettiin seuraavien tutkintakysymysten avulla:

- Kuinka aiemmin tutkittuja ohjelmiston konfiguraationhallinnan käytänteitä voidaan hyödyntää energiamittauksen ohjelmiston evoluutioiden hallinnassa.
- Soveltuvatko kohdeyrityksen hyväksymät ja käytössä olevat CMMI käytänteet ohjelmiston konfiguraationhallinnan menetelmän rungoksi?

Ensimmäiseen kysymykseen vastattiin kirjallisuuteen pohjautuvassa luvussa kaksi, luvussa esiteltiin neljä ohjelmiston konfiguraationhallinnan teemaa ja todettiin valittujen teemojen haasteellisuus. Valituista teemoista johdetut kaksi haastetta liitettiin selvitykseen ja menetelmärunkoon, haasteita aiheuttavia teemoja olivat: Teema 1: Muutossarjat ja niiden versiointi (change set) ja tähän versiointiin liittyvä Teema 3: Algoritmit: Konfiguraatio tiedostojen vertailu. Teemojen hyödyntäminen todettiin selvityksen ja menetelmärungon avulla. Kohdeyrityksen hyväksymät ja ohjelmiston konfiguraationhallinnan kannalta vähäisessä roolissa olleet CMMI käytänteet soveltuvat ohjelmiston konfiguraationhallinnan rungoksi, tämä todettiin johdolle demonstraation palautteesta ja empiirisen tutkinnan avulla.

Seuraavassa tutkimuksen onnistuminen ja sen reliabiliteettia ja validiteettia arvioidaan. Tutkimuksen asiantuntijahaastattelujen otos oli laajuudeltaan riittävä ja se edusti kohdeyrityksen perusjoukkoa hyvin. Otos valittiin kohdeyrityksen johdon suositusten ja empiirisen tutkinnan avulla. Osa haastatteluiden aineistosta piti sulkea tutkimuksen ulkopuolelle, aineistojen aiheeseen liittymättömyyden vuoksi. Haastattelujen vastauksia pidän luotettavina, haastattelukysymysten rakenteen ja haastateltavien konfiguraationhallinnan osaamiseen ja kokemuksiin viitaten. Haastattelujen tuloksia voitaisiin parantaa haastattelulomaketta tarkentamalla ja haastateltavien määrän kasvattamisella. Haastattelun, selvityksen ja ensimmäisen menetelmärungon valmistumisen jälkeen luotiin kyselytutkimus varmistamaan menetelmän runkoa. Kyselylomakkeen kysymykset luotiin kirjallisuuden ja CMI-käytänteiden pohjalta, joten kyselyn katsotaan mitaavan haluttuja asioita ohjelmiston konfiguraationhallinnasta. Kyselyn vastaukset olivat subjektiivisia ja kohdistuivat osittain annetun aiheen ulkopuolelle, tällaiset aiheen ulkopuoliset vastaukset jätettiin huomiomatta tutkimusaineisossa. Tutkimuksen reliabiliteetin ja validiteettiä katsotaan onnistuneen riittävä hyvin. Tutkimus on toistettavissa ja se tuottaa vastaavat tulokset uudelleen tutkittaessa. Tutkimustuloksilla katsotaan olevan varsinkin käytännöllistä merkitystä kohdeyrityksen ohjelmiston konfiguraationhallinnan jatkokehityksille, myös tieteellinen merkitys korostuu energiamittausalan ohjelmiston konfiguraationhallinnan ollessa energia-alalla ja myös kohdeyrityksessä ajankohtainen aihe ja sen vaikutukset ulottuvat läpi koko yrityksen organisaation. Ohjelmiston konfiguraationhallintaa on tutkittu jo pitkää ja sen käyttöönotto energiamittauksen ohjelmistolle on tarpeellista. Tämän tutkimuksen avulla yrityksen organisaatio voi paremmin ymmärtää, mitä etuja hyvällä ohjelmiston konfiguraationhallinnalla voidaan saavuttaa.

Energia-alan ohjelmistojen konfiguraationhallinta on koko ajan muuttuva kokonaisuus ja mielenkiintoisia jatkotutkimuksen aiheita olisivat ohjelmiston konfiguraationhallinnan hajauttaminen ja etäkehitys, konfiguraationhallinnan tehokkuuden mittaaminen ja konfiguraationhallinnan standardointi sekä energia-alaa läheisesti koskettavan asioiden internetin (IoT) tuomat haasteet ohjelmiston konfiguraationhallinnalle.

Yrityksen Konfiguraationhallinnan menetelmärungon toteutuksen lähtökohdaksi valikoitui CMMI vaihe 2 'CM.SP 1.2 Konfiguraationhallinnan menetelmän luominen'. Yrityksessä jo tutkittu SaltStack (2017) mahdollistaa yrityksen hajanaisten ohjelmiston konfiguraationhallintaprosessien yhteen koostamisen ja hallinnan. Jään innolla seuraamaan ohjelmiston konfiguraationhallinnan prosessien jatkokehitystä ja niiden hallintaa.

LÄHTEET

- Aiello, B. & Sachs, L. (2010). Configuration Management, Best practices: Practical methods that works in the real world. Pearson Education, 9.
- Ali, U. & Kidd, C. (2013). Critical success factors for configuration management im-plementation. *Industrial Management & Data Systems*, Vol. 113 No. 2. Management of Projects, School of Mechanical Aerospace and Civil Engineering, The University of Manchester, Manchester, UK
- Barcelos AMF. (2003) Researching Beliefs About SLA: A Critical Review. © 2008 Kluwer Academic Publishers.
- Berlack, H. R. (2002). Software Configuration Management. Encyclopedia of Software Engineering. Copyright © 1999-2017 John Wiley & Sons, Inc, 9.
- Brügge, B., Dutoit, A. (2004). Software Configuration Management, University of Zurich, Department of Information. © 2004 B. Brügge, A. Dutoit, Pearson Education, 5.
- Chef (2017) *Avoimen lähdekoodin konfigurointiohjelmisto*. Haettu 10.6.2017 osoitteesta <https://www.chef.io/chef/>.
- Chrissis, M., Konrad, M. & Shrum, S. (2011). CMMI for Development: Guide-lines for process integration and product improvement (2. painos). Massachusetts: Westford
- CMMI, 2010, CMMI for serv CMMI, (2010). CMMI for Development, Version 1.3, Software Engineering Institute, Carnegie Millon University, Pittsburgh, PA.
- Conradi, R., Westfechtel, B., (1997). *Version Models for Software Configuration Management*. ACM Computing Surveys (CSUR)
- Eriksson, P., Kovalainen, A. (2014) *Qualitative Methods in Business Research*. 2nd edition. Sage Publications Ltd. Los Angeles.
- Eriksson, P., Koistinen, K. (2014) *Monenlainen tapaustutkimus. Kuluttajatutkimuskeskuksen tutkimuksia ja selvityksiä 11*. Helsinki.
- Estublier, J. (2005) Software Configuration Management. A Road Map. Da-saultSystemes/LSR, Grenoble University, Actimart, Allee de Roumanie 38610 Gieres France, 2.
- Estublier, J., Leblang D., Hoek A., Conradi R. (2005) Impact of Software Engi-neering Research on the Practice of Software Configuration Management. Jour-nal ACM Transactions on Software Engineering and Methodology (TOSEM), 6–7, 32, 35.
- Feiler, P. H. 1991. Configuration management models in commercial environ-ments. Tech. Rep. CMU/SEI-91-TR-7 (March), Software Engineering Institute, Carnegie-Mellon, University, Pittsburgh, PA.

- Fuggetta A., Di Nitto E., (2014) Software Process. Politecnico di Milano and CE-FRIEL, Via Fucini, 220133 Milano – Italy. Politecnico di Milano, Piazza Leonar-do da Vinci, 3220133 Milano – Italy, 10.
- Hirsjärvi, S., Remes, P. & Sajavaara, P. (2007). Tutki ja kirjoita. 13-14. painos. Helsinki: Tammi, 130.
- Hoek, A., Heimbigner, D., Wolf, A. Does configuration management research have a future? In: ESTUBLIER, J. (Ed.). SCM. [S.l.]: Springer, 1995.
- Humphrey, W. S. 1989. Managing the Software Process. SEI Series in Software Engineering, Addison-Wesley, Reading, MA.
- Hunt, J., McIlroy, M. (1976). An efficient algorithm for differential file comparison. Tech. Rep. 41. Bell Labs (June).
- ISO (2003) Quality management systems: Guidelines for Configuration Management (ISO 10007:2003). International Organization for Standardization, Ge-neva.
- Jenkins (2017), Jenkins:in tuotesivu. Haettu 08.07.2017 osoitteesta: <https://jenkins.io/>.
- Kidd, C. & Burgess, T. (2004). Managing configurations and data for effective project management, in Morris, P.W.G. & Pinto, J.K. (Eds), The Wiley Guide to Managing Pro-jects, Wiley, Hoboken. New Jersey, 498.
- Kidd, C., Burgess, T. (2007) Project, Technology, Supply Chain & Procurement Management. Copyright 2007 by John Wiley & Sons, Inc. All rights reserved, 101, 108.
- Kraijak, S., Tuwanut, P. (2015) “A survey on IoT architectures, protocols, applications, security, privacy, real-world implementation and future trends”. Teo-ksessa 11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015).
- Laari, J. (2016) Elinkaaren aikaisen konfiguraationhallinan toteutus. Lappeen-rannan teknillinen yliopisto LUT, 9.
- Landis Gyr Oy (2017) Historia. Haettu 10.07.2017 osoitteesta <http://www.landisgyr.fi/about/history/>
- Landis Gyr, Global Organisation, (2017) Organisaatio. Haettu 10.07.2017 osoit-teesta: <http://www.landisgyr.fi/about/global-organization/>
- Lie, A., Didriksen, T., Reidar C., Karlsson, EA. (1998) Division of Compute Systems and Telematics, Norwegian Institute of Technol-ogy, Trondheim, Norway.
- Lukka, K. 2001. Konstruktiivinen tutkimusote. [www-lähde] Metodix. [Viitattu 26.2.2016]
- Metsämuuronen, J. (2001). Laadullisen tutkimuksen perusteet. Helsinki: Inter-national Methelp

- Mette, A., Hass, J. (2002) Configuration Management Principles and Practice. Copyright © 2003 by Pearson Education, Inc., 32.
- Meyers, E. (1986). An O(ND) difference algorithm and its variations. *Algorithmica* 1, Springer Verlag New York Inc.
- Milton, N., Clarke, D., Shadbolt, N. (2006) Knowledge engineering and psychology: Towards a close relationship. Epistemics, Strelley Hall, Nottingham, NG8 6PE, UK. School of Psychology, University of Nottingham, University Park, Nottingham, NG7 2RD, UK. Department of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK, 1218.
- Monte.Mor, J., Cunha, AM. (2014) GALO: A Semantic Method for Software Configuration Management. Federal University of Itajuba - UNIFEI Itabira/MG, Brazil. Brazilian Aeronautics Institute of Technology - ITA, Sao Jose dos Campos/SP, Brazil, 1.
- O'Regan, G. (2011). Introduction to software process improvement. London: Springer London. Haettu 8.11.2017 osoitteesta <http://dx.doi.org.ezproxy.jyu.fi/10.1007/978-0-85729-172-1>
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3), 45-77.
- Puppet (2017) Avoimen lähdekoodin konfigurointiohjelmisto. Haettu 10.6.2017 osoitteesta <https://puppetlabs.com/>.
- SaltStack (2017) Avoimen lähdekoodin konfigurointiohjelmisto. Haettu 10.6.2017 osoitteesta <http://saltstack.com/>.
- Schwaber K., Sutherland, J. 2013 The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. ©2014 Scrum.Org and ScrumInc, 3.
- Silva, F., Furtado Soares F., Peres A, de Azevedo, I. Vasconcelos, A., Kamei F., de Lemos Meira, S. (2015) Using CMMI together with agile software development: A systematic review. Center of Informatics, Federal University of Pernambuco (UFPE), Recife, PE, Brazil, Computer Science Course (FACET), Mato Grosso State University (UNEMAT), Barra do Bugres, MT, Brazil, Recife Center of Advanced Studies and Systems (C.E.S.A.R), Recife, PE, Brazil, Cesmac University Center, Maceió, AL, Brazil, Federal Institute of Alagoas (IFAL), Ara-piraca, AL, Brazil, 2.
- Technology Dictionary 2017: Haettu 10.11.2017 osoitteesta: <https://www.techopedia.com/definition/24822/configuration-controlconfiguration-management-cm>
- Tichy, W. F., 1988. Tools for software configuration management. In Proceedings of the International Workshop on Software Version and

Configuration Control (Grassau, Germany), J. F. H. Winkler, Ed., Teubner Verlag, 1-20.

Tutkimushanke: Smart Energy Transition, 2016. Haettu osoitteesta 12.7.2017
<http://smartenergytransition.fi/fi/teknologiamurros/>

Westfechtel, B., Munch, B.P., Conradi, R. (2001) A layered architecture for uniform version management. Published in: IEEE Transactions on Software Engineering (Volume: 27, Issue: 12, Dec 2001).

Wibas. (2015) Version 1.7.1 © 2015 wibas GmbH. Haettu 2.3.2017 osoitteesta
<https://www.wibas.com/cmmi/>

Zeller, A., Snelting, G. (1997) Unified Versioning through Feature Logic. Institut für Programmiersprachen und Informationssysteme Abteilung Softwaretechnologie Technische Universität Braunschweig Bültenweg 88 D-38092 Braunschweig/Germany.

Liite 1 Asiantuntijoiden haastattelupohja

Alla olevat CMMI-käytänteet toimivat haastattelun runkona, haastateltavia pyydettiin kertomaan omat näkemykset CMMI-käytänteiden toteutumisesta kohdeyrityksessä.

CMMI-käytänteet:

<https://www.wibas.com/cmmi/configuration-management-cm-cmmi-dev>

CM.SP 1.1 Konfiguraationhallinta-alkioiden tunnistaminen

Konfiguraationhallinta-alkioiden, komponenttien ja tuotteiden tunnistaminen

Ali-käytänteet:

1. Konfiguraationhallinnan kohteiden valinta ja tuotteet, jotka muodostuvat dokumentoitujen kriteereiden perusteella.
2. Yksilöivien tunnisteiden luominen konfiguraationhallinnan hallinta-alkioille.
3. Konfiguraationhallinta-alkioiden ominaisuuksien määrittely (esim. tiedoston laatija, tiedoston tyyppi, ohjelmointikieli, tiedoston tarkoitus).
4. Konfiguraationhallinta-alkion konfiguraationhallintaan lisäämisaika.
5. Konfiguraationhallinta-alkion omistaja ja vastuu.
6. Konfiguraationhallinta-alkion suhteet muihin hallinta-alkioihin ja konfiguraationhallinnan rakenteeseen.

CM.SP 1.2 Konfiguraationhallinnan menetelmän luominen

Konfiguraationhallinta menetelmä sisältää tallennusvälineet, menettelyn ja työkalut järjestelmän käyttämiseen. Menetelmä voi koostua useista osajärjestelmistä, joilla on erilaiset toteutukset, jotka sopivat kullekin konfiguraationhallintaympäristölle.

Ali-käytänteet:

1. Monitasoisen mekanismin luominen konfiguraationhallinnalle. Mekanismin tasot valitaan tyypillisesti projektin tavoitteiden, riskien ja resurssien perusteella. Tasot voivat vaihdella hankkeen elinkaaren, kehitteillä olevan järjestelmän tyyppin ja erityisten projektivaatimusten mukaan.
2. Konfiguraationhallintajärjestelmän käyttöoikeuksien hallinta.
3. Konfiguroinnin hallinta-alkioiden noutaminen ja tallennus konfiguraationhallintajärjestelmästä/ään.
4. Konfiguroinnin hallinta-alkioiden jakaminen eri tasoille.
5. Konfiguroinnin hallinta-alkioiden tallennus ja palautus.
6. Konfiguroinnin hallinta-alkioiden aikakirjojen tallennus, päivitys, noutaminen.

7. Konfiguraationhallinnan raporttien luominen (konfiguraationhallinnasta).
8. Konfiguraationhallinnan sisällön talletus.
9. Konfiguraationhallinnan rakenteen tarkistus tarvittaessa.

CM.SP 1.3 Lähtötilanteen tunnistaminen

Lähtötilanteen tunnistaminen ja sisäisen kehityksen ja asiakkaalle toimituksen lähtötilannepisteen luominen. Monia eri lähtötilanteita voi olla käytössä esim. tuotekehitykselle ja testaukselle. Yksi yhteinen viitekehys sisältää järjestelmän vaatimukset ja tuotteen määritykset tuotteen alkuvaiheessa.

Ali-käytänteet:

1. Konfiguraationhallinnan lähtötilanteen luomiselle oikeutuksen saaminen yrityksen johdolta.
2. Julkaiseminen tai lähtötilanteet luominen vain konfiguraationhallinnasta löytyvillä hallinta-alkioilla.
3. Lähtötilanteiden sisältäminen hallinta-alkioiden kuvaamiseen.
4. Lähtötilanteiden saataville asettaminen.

CM.SP 2.1 Muutospyyntöjen seuranta

Vaatimuksista johdetut, virhekorjauksista ja tuotteen korjauksiin liittyvien muutosten seuraaminen.

Ali-käytänteet:

1. Muutospyyntöjen luominen ja kirjaaminen muutospyyntöjen tietokantaan.
2. Muutospyynnöissä esitettyjen muutosten ja korjausten vaikutusten arviointi.
3. Muutospyyntöjen luokittelu ja priorisointi.
4. Seuraavan lähtötilanteen muutospyyntöjen katselmointi olennaisten sidosryhmien kanssa.
5. Muutospyyntöjen seuranta niiden sulkemiseen asti.

CM.SP 2.2 Konfiguraationhallinta-alkioiden valvonta

Hallinta-alkioiden seuranta pitää sisällään kaikkien hallinta-alkioiden seurannan, muutosten hyväksynnät ja lähtötilanteen päivittämisen.

Ali-käytänteet:

1. Konfiguraation hallinta-alkioiden elinaikainen hallinta
2. Asianmukaisen valtuutuksen hankinta ennen muuttuneiden hallinta-alkioiden lisäämistä konfiguraationhallinta järjestelmään.

3. Konfiguraation hallinta-alkioiden oikeellisuuden ja eheyden valvonta, hallinta-alkioita käytettäessä.
4. Konfiguraation hallinta-alkioiden katselmoinnit ja varmistukset, jottei muutokset vaaranna niiden turvallisuutta.
5. Konfiguraation hallinta-alkioiden muutosten ja syiden kirjaaminen.

CM.SP 3.1 Perusta konfiguraationhallinnan aikakirja (Records) Konfiguraationhallinta-alkioiden aikakirjan perustaminen ja ylläpito.

Ali-käytänteet:

1. Konfiguraationhallinnan toimien tallentaminen riittävän yksityiskohtaisesti, jotta kunkin hallinta-alkion sisältä ja tila tunnetaan ja aiempien versioiden palauttaminen on mahdollista.
2. Asianomaisten sidosryhmien seuranta-oikeuksien varmistaminen.
3. Uusien lähtötilanteiden määrittäminen:
 - Käyttöoikeuksien tuottaminen valtuutetuille sidosryhmien loppukäyttäjille.
 - Hallinta-alkioiden lähtökohtaiset kopioiden saataville tarjoaminen
 - Automaattisten hälytysten tuottaminen sidosryhmille, kun hallinta-alkioiden tarkistuksista, poistoista tai muutoksista (muutos ehdotusten seuranta).
4. Tietyn lähtötilanteen muodostavien hallinta-alkioiden tunnistaminen.
5. Onnistuneiden lähtökohtien erojen kuvaaminen.
6. Konfiguraation hallinta-alkioiden tilan ja historian tarkastaminen tarpeen mukaan.

CM.SP 3.2 Suorita konfiguraationhallinnan lähtötilanteen auditointi

Ali-käytänteet:

1. Konfiguraationhallinta-alkioiden yhtenäisyyden ja riippuvuuksien arviointi.
2. Varmista, että konfiguraationhallinta aikakirja tunnistaa oikein hallinta-alkiot.
3. Varmista konfiguraationhallinta-alkioiden rakenne ha eheys.
4. Ohjelmiston konfiguraationhallinnan kohteiden täydellisuuden, oikeellisuuden ja johdonmukaisuuden varmistaminen. Täydellisyys, oikeellisuus ja johdonmukaisuus perustuvat vaatimuksista johdettuihin suunnitelmiin ja hyväksytyjen muutospyyntöjen toimittamiseen.
5. Sovellettavien kokoonpanonhallintastandardien ja -menettelyjen noudattamisen varmistaminen.
6. Ohjelmiston konfiguraationhallinta-alkioiden seuranta auditoinnista sulkemiseen.

Liite 2 Kyselylomake

Kysely ohjelmiston konfiguraationhallinnasta

Hei,

Teen Jyväskylän yliopistolle pro-gradu tutkielmaa yrityksemme ohjelmiston konfiguraationhallinnasta. Siihen liittyen, tämän kyselyn tavoitteena on selvittää, millaisena tutkittu ohjelmiston konfiguraationhallinnan menetelmä nähdään yrityksen asiantuntijoiden parissa. Kyselyn kaikki kysymykset liittyvät yrityksen ohjelmiston konfiguraationhallintaan

Kyselyssä on yhdeksän kysymystä ja vastaaminen kestää 15 – 20 minuuttia. Vastaamalla autat pro-gradu työni tekemistä ja ohjelmiston konfiguraationhallintaan liittyvien kehityskohteiden löytämistä. Vastauslomake käsittelee vastaajan tiedot anonyymina.

Vastauksia otetaan vastaan keskiviikkoon 6.9. saakka.

Suuri kiitos ajastasi!

T: Jarmo

1. Mitä ymmärrät ohjelmiston konfiguraationhallinnalla, mitä se tarkoittaa sinun tehtävissäsi?
2. Kenen mielestäsi pitää määritellä hallittavat ohjelmiston konfiguraatiot? (esim. projektipäälliköt, asiakkaat, käyttäjät vai joku muu.)
3. Tuotteen elinkaaren aikaiset muutokset aiheuttavat haasteen ohjelmiston konfiguraationhallinnalle. Mitkä mielestäsi ovat suurimmat haasteet elinkaaren aikaiselle konfiguraationhallinnalle?

4. Ohjelmiston konfiguraationhallinnan yksi iso aihe on konfiguraatioiden muutostenhallinta. Kerro alle näkemyksesi aktiviteeteista:
 - a) Muutospyynnöt (asiakkaiden ja sisäiset muutokset)?

 - b) Ristiriitaisuudet (esim. asiakkaiden eri tyyppiset vaatimukset)?

 - c) Muutostenhallinta ryhmän toimiminen (muutosten läpikäynti ja hyväksyntä)?

5. Ohjelmiston konfiguraatioiden versionhallinnan samanaikaiset muutokset vaativat versionhallinnalle hallitun rakenteen ja työkalut. Näetkö versionhallinnan tärkeänä osana konfiguraationhallintaa? esim. versiointi puurakenteeseen tms., samanaikaiset muutokset, työkalut toteutukselle.)

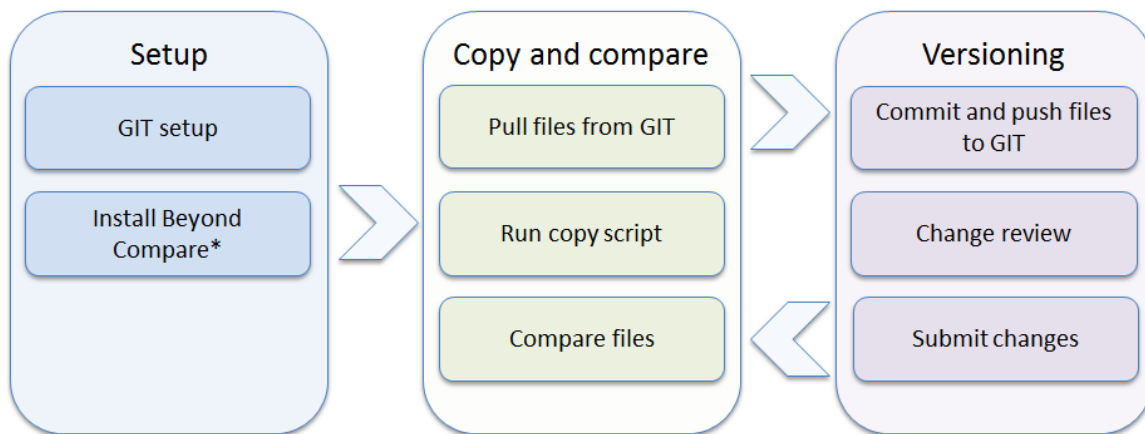
6. Ohjelmiston konfiguraationhallinnan mittaaminen tuo tärkeätä tietoa menetelmän toiminnasta. Mm muutoksien määrä, muutospyynnöt, virheiden määrä ja prioriteetti. Voidaanko ohjelmiston konfiguraationhallintaa käyttää tuotteen suorituskyvyn mittarina? Perustele vastauksesi.

7. Ohjelmiston konfiguraationhallinnalle voidaan luoda yrityksen sisäinen standardi, näetkö sellaisen tarpeellisena? Perustele vastauksesi.

Liite 3 Ohjelmiston konfiguraatiodostojen hallintaprosessi

This document contains information how to implement Solution Configuration File Management. Solution has several configuration files which are edited during installation or update or when a new feature is taken into use. Solution configuration files from test environments are saved to GIT version control to improve configuration file management.

User Guide: Below image instructs how to start using GIT version control and relating tools, how to collect configuration files from test environments, how to compare the files of different versions and how to save the files into GIT.



Process: Solution Configuration File Management is intended to be used during test environment solution installation, update or configuration change. It helps to identify changes in configuration files after these actions and gives automated tool to inform other team members about the changes.

