

**VAASAN YLIOPISTO**  
**TEKNIIKAN JA INNOVAATIOJOHTAMISEN YKSIKKÖ**  
**TIETOTEKNIikka**

Jarkko Romppanen

**HYBRIDISOVELLUKSET SHAREPOINTISSA**

Suunnitelma laadunvarmistusprosesseista kohdeyrityksessä

Tietotekniikan  
pro gradu -tutkielma

**VAASA 2020**

<b>SISÄLLYSLUETTELO</b>		<b>sivu</b>
1	<b>JOHDANTO</b>	6
2	<b>KIRJALLISUUSKATSAUS</b>	8
	2.1 Yritysarkkitehtuuri	8
	2.2 Pilvilaskenta	12
	2.3 Prosessi- eli kontrolliteoria	15
	2.4 Prosessin hallinta sovelluksen elinkaaren tukena	16
	2.5 Kohderyhmät sovellusten laadunvarmistusprosesseissa	17
3	<b>TUTKIMUSMENETELMÄ</b>	19
	3.1 Tutkimusmenetelmien vertailua	19
	3.2 Tutkimusmetodien yhteenveto ja valinta	23
	3.3 Toimintatutkimusprosessi	25
	3.4 Suunnitelma tutkimuksen etenemisestä	27
	3.5 Tiedonkeruu tutkimuksen aikana	28
4	<b>KOHDEYRITYKSEN SHAREPOINT-SOVELLUSMALLIT</b>	30
	4.1 SharePoint	30
	4.2 SharePointin kehitys työkalusta liiketoimintasovellusalustaksi	33
	4.3 SharePoint sovellusmallit	35
	4.4 SharePoint Hybrid -sovellusmalli	39
	4.5 Kohdeyrityksen tietohallintorakenne relevantein osin	42
	4.6 SharePoint Server-sovellukset kohdeyrityksessä	45
	4.7 SharePoint Online-sovellukset kohdeyrityksessä	52
	4.8 SharePoint Hybrid-sovellukset kohdeyrityksessä	52
	4.9 Uusien prosessien tuomat riskit kohdeyrityksessä	54
5	<b>KOHDEYRITYKSEN PROSESSIEN MALLINNUS HAASTATTELUMENETELMIN</b>	56
	5.1 Ensimmäisen haastattelukierroksen olettamien pohdintaa	56
	5.2 Ensimmäisen haastattelukierroksen tulokset ja korjatut olettamat	63
	5.3 Toisen haastattelukierroksen tulokset	67

5.4	Optimoidut laadunvarmistusprosessit ja tunnistetut jatkokehityskohteet	71
6	JOHTOPÄÄTÖKSET	75
	LÄHTEET	78

## KUVALUETTELO

Kuva 1. Zachmanin viitekehys	9
Kuva 2. Objektit voivat olla toistensa aliobjekteja	12
Kuva 3. Pilvipalveluarkkitehtuuri	14
Kuva 4. Sovelluksen laadunvarmistusprosessin runko	17
Kuva 5. Reflektoivan toimintatutkimuksen eteneminen	20
Kuva 6. Suunnittelututkimuksen syklit	22
Kuva 7. Toiminnan suunnitelmatutkimuksen osa-alueet	23
Kuva 8. Soveltuvien tutkimusmetodien vertailua	24
Kuva 9. Toimintatutkimuksen tutkija osana organisaatiota	26
Kuva 10. Tutkimuksen syklien vaihemäärittelyt	27
Kuva 11. Vertailu erilaisten haastattelujen piirteistä	28
Kuva 12. Esimerkki tiedostokirjastossa SharePoint Onlinessa	31
Kuva 13. Esimerkki tietuelistasta SharePoint Onlinessa	32
Kuva 14. SharePoint Server-sovelluksien implementointi- ja muutosprosessit laadunvarmistuksen näkökulmasta	48
Kuva 15. SharePoint Server-sovellusten laadunvarmistusprosessien läpimenoajat	51
Kuva 16. Haastatteluiden sidosryhmät ja teemahaastattelun aihealueet	58
Kuva 17. SharePoint Server- ja SharePoint hybridisovelluksien suunniteltujen laadunvarmistusprosessien läpimenoaikojen vertailua	61
Kuva 18. Karkea kuvaus ehdotuksesta hybridimallisten sovellusten implementointi- ja muutosprosessiksi	62
Kuva 19. SharePoint Server- ja SharePoint hybridisovellusten laadunvarmistusprosessien läpimenoaikojen vertailua haastattelutuloksien pohjalta korjattuna	67
Kuva 20. Optimoidun implementointi- ja muutosprosessin läpimenoajat kohdeyrityksessä	70

Kuva 21. Hybridisovellusten implementointi- ja muutosprosessi, vastuukuvaus ja työnkulku

---

**VAASAN YLIOPISTO****Tekniikan ja innovaatiojohtamisen yksikkö****Tekijä:**

Jarkko Romppanen

**Tutkielman nimi:**Hybridisovellukset SharePointissa:  
Suunnitelma laadunvarmistusprosesseista  
kohdeyrityksessä**Ohjaaja:**

Tero Vartiainen

**Tutkinto:**

Kauppatieteiden maisteri

**Pääaine:**

Tietotekniikka

**Tutkielman valmistumisvuosi:**

2020

**Sivumäärä: 83**

---

**TIIVISTELMÄ:**

SharePoint-alustojen päälle rakennettujen liiketoimintasovellusten kenttä on muuttumassa radikaalisti. Muutoksen katalyyttinä on trendi pilviteknologioihin siirtymisessä. Liiketoimintasovellukset ovat suurissa yrityksissä perinteisesti rakennettu yritysten itse hallinnoimien servereiden päälle. Pilviteknologiat mahdollistavat liiketoimintasovellusten rakentamisen palvelutarjoajan infrastruktuurin päälle pilveen, joko kokonaan tai osittain. Ongelmaksi muodostuu ohjelmistokehitykselliset ja laadunvarmistukselliset prosessit, jotka heijastavat edelleen perinteistä omiin servereihin perustuvaa liiketoimintasovellusmallia.

Tutkimuksen tavoitteena oli luoda kohdeyritykselle uudistetut laadunvarmistusprosessit peilaamaan pilvipalveluteknologian mahdollistamia SharePoint-hybridisovelluksia. Tavoitteesta johdettiin tutkimukselle yleistettävämpi tavoite sovelluskehityksen implementointi- ja muutosprosessien suunnittelemiseksi laadunvarmistuksen näkökulmasta. Tutkimuskysymys muodostui seuraavaksi: Millaisin prosessein voimme viedä eteenpäin sovellusten implementointia ja muutoksia, niin että ne ovat linjassa yrityksen laadunvarmistuksen kanssa? Tutkimus toteutettiin iteratiivisena haastatteluihin perustuvana toimintatutkimuksena. Haastattelut olivat teemaperusteisia, ja ne käytiin kahdenkeskeisesti kirjoittajan ja kohdeyrityksestä tarkkaan valittujen sidosryhmien jäsenten välillä.

Tutkimuksen lopputuotoksena syntyi hybridisovellusten laadunvarmistusprosessit, eli implementointi- ja muutosprosessit, jotka otetaan käyttöön kohdeyrityksessä osana SharePoint-sovellusten hallinnan päivittäistä työtä. Prosessien optimointi on merkittävässä roolissa kustannusten laskemiseksi laadunvarmistuksellisissa toimissa. Lisäksi tutkimuksessa tunnistetaan jatkokehityskohteet myös tulevaisuuden optimointeja ja prosessien elinkaarien pituuksia ajatellen. Tutkija on poistunut organisaatiosta ennen prosessien käyttöönottoa. Tutkimuksen tuotokset ovat vahvasti sidoksissa kohdeyritykseen, eikä niiden soveltuvuutta voida yleistää kaikkeen laadunvarmistukselliseen prosessikehitykseen.

---

**AVAINSANAT:** Prosessit, laadunvarmistus, pilvipalvelut, SharePoint

## 1 JOHDANTO

Pilvipalvelut muuttavat yritysten infrastruktuuri- ja sovelluskenttää. Ne mahdollistavat omista tietokeskuksista luopumisen, ja niiden ylläpidosta johtuvista kustannuksista irtaantumisen (Li & Li 2013: 866-867, Webber-Cross 2014: 30). Pilvipalvelut voivat tuoda yrityksille merkittäviä säästöjä palveluiden nopean, helpon ja lähes rajattoman skaalautuvuuden (Li & Li 2013: 866-867), sekä korkean saatavuuden tuella (Webber-Cross 2013: 30). Toisaalta siirtyminen pilvipalveluihin suurissa organisaatioissa aiheuttaa myös muutostarpeen sovelluksia koskevissa prosesseissa (Webber-Cross 2014: 38). Prosessi-, eli kontrolliteorian avulla yritetään ymmärtää, kuinka organisaation jäsenet työskentelevät kohti yhteistä tavoitetta (Kirsch 1996: 1). Yritysarkkitehtuurin ymmärtäminen luo pohjan harkittujen muutosten toteuttamiselle strategisesti järkevällä tavalla (Harmon 2003: 3; Mamaghani, Madani & Sharifi 2011: 219).

Pilvipalvelut ovat tuoneet uusia mahdollisuuksia SharePoint-tekniikoilla rakennettuihin liiketoimintasovelluksiin. Tutkimuksen kohdeyrityksen laadunvarmistusprosessit eivät enää tukeneet ketterään pilvipalveluun kokonaan tai osittain pohjautuvia liiketoimintasovelluksia. Tutkimuskysymykseksi muodostui: *Millaisin prosessein voimme viedä eteenpäin sovellusten implementointia ja muutoksia, niin että ne ovat linjassa yrityksen laadunvarmistuksen kanssa?* Tutkimuksen tavoitteena oli tuottaa kohdeyritykselle uudet laadunvarmistusprosessit pilvipalveluita hyödyntäville SharePoint hybridisovelluksille, ja yrittää siten vastata tutkimuskysymykseen. Prosessikehityksen lähtökohtana olivat kohdeyrityksen nykyiset omaa infrastruktuuria hyödyntävien SharePoint-liiketoimintasovellusten laadunvarmistusprosessit.

Tutkimus toteutettiin iteratiivisena toimintatutkimuksena, jossa ongelman tunnistaminen ja toiminta vuorottelevat, kunnes hyväksyttävä ratkaisu löydetään (Tiainen et al. 2015: 8). Haastatteluissa hyödynnettiin kohdeyrityksen soveltuvien sidosryhmien asiantuntijoita ja ne toteutettiin teemaperusteisesti.

Tutkimuksen tuloksena saavutettiin tavoitteen mukaisesti yrityksessä käyttöönotettavat implementointi- ja muutosprosessit kohdeyrityksen nykyisille ja tuleville hybridisovelluksille. Tutkimus osoittaa toimintatutkimuksen soveltuvan sovellusprosessien suunnitteluun hyvin kohdeyrityksessä, mutta jättää sen soveltuvuuden yleistämiselle koko yrityskentässä vielä selvitettävää. Prosessit muodostuivat toimintatutkimuksen tuloksena, ja joiden optimointiin käytettiin vaiheittaisia kohderyhmähaastatteluita. Uusien prosessien optimoidut läpimenoajat lyhenivät merkittävästi aiemman sovellusmallin prosesseista, ja hybridisovellusmalli todettiin erinomaiseksi, sekä laadunvarmistuksellisten tehtävien läpimenoajan merkittävän laskun, että teknologisen mahdollisuuksien. Lisäksi voidaan todeta, että hybridimalliset sovellukset laskevat yritysten migraatiokustannuksia merkittävästi ja mahdollistaa Microsoftin ilmaisten päivitysten nopean käyttöönoton. Tutkimuksen viimeisessä osiossa kirjattiin johtopäätökset tutkimuksen tuloksista, tavoitteiden saavuttamisesta, sekä tutkimuksen hyödyllisyydestä sekä kohdeyritykselle, että tieteelliselle yhteisölle.



## 2 KIRJALLISUUSKATSAUS

Kirjallisuuskatsauksessa avataan tutkimuksen kannalta oleellisia aiheita, luodaan lukijalle yleiskuva tutkimuksen kentästä ja tunnistetaan sovellusten elinkaarille oleellisia ylätason käsitteitä. Tutkimuksen kannalta oleellista on tunnistaa uuden sovellusmallin tuomat uudet teknologiset mahdollisuudet, ja tunnistaa vaatimukset niiden ylläpidollisille toimenpiteille.

### 2.1 Yritysarkkitehtuuri

Sovelluksien laadunvarmistusprosessit, niiden tarve ja merkitys yritykselle, voidaan juurtaa yritysarkkitehtuurista. Yritysarkkitehtuuri (engl. Enterprise Architecture) on ylätason käsite ja työkalu, jonka avulla yritysjohto voi nähdä yrityksen kokonaisrakenteen ja auttaa sitä ymmärtämään sen (Harmon 2003: 3; Holm, Buschle, Lagerström & Ekstedt 2012: 825). Yritysarkkitehtuurin avulla yritysjohto hahmottaa yrityksen eri osa-alueet ja niiden väliset sidokset (Holm et al. 2012: 825; Mamaghani, Madani & Sharifi 2011: 219). Yritysarkkitehtuurin ymmärtäminen luo pohjan harkittujen muutosten toteuttamiselle strategisesti järkevällä tavalla (Harmon 2003: 3; Mamaghani et al. 2011: 219). Nogueira, Romero, Espadas & Molina (2012: 103) määrittelee yritysarkkitehtuurin olevan strukturoitu mallien kokonaisuus, joka kuvaa yrityksen muuttumattomia osakokonaisuuksia. Yritysarkkitehtuuri toimii pohjana yrityksen liiketoimintaa tukevien tietohallintojärjestelmien vaatimuksien tunnistamiselle (Nogueira et al. 2012: 103).

On olennaisen tärkeää, että yrityksen eri osa-alueet tukevat toisiansa. Yrityksen tietohallintopalveluiden muutokset tulee suunnitella niin, että ne tukevat mahdollisimman hyvin liiketoimintayksikköä. Toisaalta, jos muutos vaikuttaa useisiin yrityksen liiketoimintayksikköihin, tulee ne suunnitella tietohallintopalveluissa niin, että muutoksen optimaalinen hyöty on maksimissaan kaikki liiketoimintayksiköt huomioiden. Jonkun liiketoimintayksikön toivoma muutos voi olla kannattavaa kyseiselle liiketoimintayksikölle,

mutta se voi vaikuttaa negatiivisesti toisen liiketoimintayksikön toimintaan. Mikäli harkittu muutos on kannattavaa yrityksen kaikki osa-alueet huomioiden, kannattaa se luonnollisesti toteuttaa. Kokonaiskuvan ja muutosten vaikutuksien ymmärtäminen jokaiseen yrityksen osa-alueeseen on avainasemassa yritysarkkitehtuurin strategiselle hyödyntämiselle.

Yritysarkkitehtuuria tutkiessa törmää Zachmanin viitekehykseen (Harmon 2003: 1). Zachmanin viitekehystä on käytetty paljon varsinkin yritysten tietohallintomuutosten suunnittelun tukena (Harmon 2003: 4). Viitekehys on suunniteltu kattamaan suuren yritysarkkitehtuurin kaikista merkittävistä näkökulmista (Harmon 2003: 5, Sowa & Zachman 1992: 590). Viitekehyksen ei ole tarkoitus osoittaa mitä kaikkea yrityksestä tulee dokumentoida, vaan sen tarkoitus on auttaa hahmottamaan kaikkien matriisissa olevien solujen sidoksia toisiinsa (Harmon 2003: 5; Sowa et al. 1992: 590-591). Kuvassa 1 on esitetty Harmonin (2003: 5) ehdotelman mukaan Zachmanin viitekehyksestä ne osiot, jotka ovat tärkeitä yritysjohdolle liiketoiminnallisessa mielessä (rivit 1-2), ja mitkä taas ovat tärkeitä tietohallintomuutosten toteuttamisen kannalta (rivit 3-6).

	The Zachman Framework	DATA <i>Mitä</i>	FUNKTIO <i>Miten</i>	VERKKO <i>Missä</i>	IHMISET <i>Kuka</i>	AIKA <i>Milloin</i>	MOTIVAATIO <i>Miksi</i>
Liiketoimintajohtaja	SISÄLTÖ <i>Suunnittelija</i>	Lista liiketoiminnalle tärkeitä asioista	Lista prosesseista, joita yritys käyttää	Lista lokaatioista, joissa yritys toimii	Lista yrityksen liiketoiminnalle tärkeitä organisaatioista	Lista yritykselle merkittävistä tapahtumista	Lista yrityksen tavoitteista tai strategioista
	YRITYSMALLI <i>Omistaja</i>	Semanttinen malli	Liiketoimintaprosessi malli	Liiketoiminnan logistiikkamalli	Työnkulkumalli	Päämääräinen aikataulu	Liiketoimintasuunnitelma
Tietohallintojohtajat ja kehittäjät	JÄRJESTELMÄMALLI <i>Suunnittelija</i>	Looginen datamalli	Sovellusarkkitehtuuri	Hajautetun järjestelmän arkkitehtuuri	Käyttöliittymäarkkitehtuuri	Prosessointirakenne	Liiketoiminnan sääntömalli
	TEKNOLOGINEN MALLI <i>Rakentaja</i>	Fyysinen datamalli	Järjestelmäsuunnitelma	Teknologia-arkkitehtuuri	Esittämisarkkitehtuuri	Kontrollirakenne	Sääntösuunnitelma
	TÄSMÄLLISET ESITELMÄT <i>Alihankkija</i>	Datan määrittely	Ohjelma	Tietoverkkoarkkitehtuuri	Tietoturva-arkkitehtuuri	Ajoitusmäärittely	Sääntöomäärittely
	FUNCTIONING ENTERPRISE	Tosiasiallinen liiketoimintadata	Tosiasiallinen sovelluskoodi	Tosiasialliset fyysiset tietoverkot	Tosiasiallinen liiketoimintaorganisaatio	Tosiasiallinen liiketoiminnan aikataulu	Tosiasiallinen liiketoiminnan strategia

**Kuva 1.** Zachmanin viitekehys

Tämän tutkimuksen kannalta erityisen kiinnostava on Zachmanin viitekehyksen toinen sarake, joka kuvaa *funktiota*, ja joka vastaa kysymykseen *miten*. Viitekehystä on helppo tunnistaa tutkimukselle relevantteja termejä sisältäviä soluja, kuten *liiketoimintaprosessimalli*, *sovellusarkkitehtuuri*, *järjestelmäsuunnitelma*, *ohjelma* ja *tosiasiallinen sovelluskoodi*. Kaikki ovat tutkimuksen ytimessä ja vaikuttavat toisiinsa. Tutkimuksen tavoitteena on luoda kohdeyritykselle implementointi- ja muutosprosessit, jotka käyttöön otettuna voidaan lisätä *liiketoiminnassa käytettyjen prosessien listalle*.

Zachmanin (1999: 459) mukaan sovellusarkkitehtuuri ja järjestelmäsuunnitelma ovat määritelmällisesti lähellä toisiaan, mutta ne ovat luonteeltaan erilaisia, sillä niitä tarkastellaan eri näkökulmista. Zachmanin mukaan (1999: 459) sovellusarkkitehtuuri on sovellusarkkitehdin näkemys sovelluksesta, kun taas järjestelmäsuunnitelma on sen rakentajan näkemys sovelluksesta.

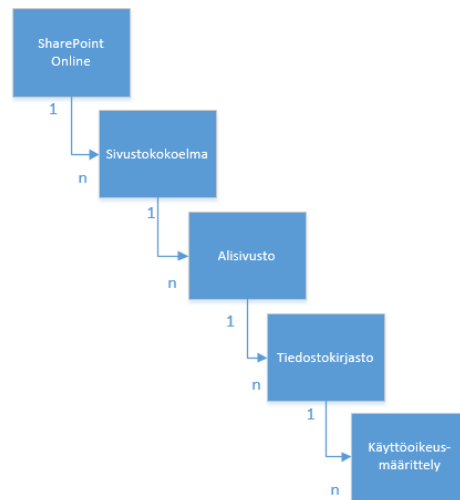
Boochin mukaan yritysarkkitehtuuri sekoitetaan helposti järjestelmäarkkitehtuurin kanssa, johtuen nimiensä yhtäläisyyksistä. Yritysarkkitehtuuri kuvaa teknologiaa käyttävän yrityksen liiketoiminnan arkkitehtuuria, kun taas järjestelmäarkkitehtuuri kuvaa tietojärjestelmäpohjaisia ratkaisuja, jotka tukevat liiketoimintaa. Toinen Boochin määritelmä näiden kahden erotukselle on se, että yritysarkkitehtuuri käsittelee sosio-tekniologista organisaatiota, kun taas järjestelmäarkkitehtuuri käsittelee tietojärjestelmän elementtejä. (Booch 2010: 96)

Zachmanin viitekehystä tarkastellessa huomaamme, että järjestelmäsuunnitelma on osana yritysarkkitehtuurin viitekehystä ja se kuuluu sovellusarkkitehtuurin piiriin. Se on osana erityisesti tutkimuksen näkökulmasta kiinnostavaa viitekehyksen funktio-osuutta, joten voitaneen yleistää, että sovelluksien järjestelmäsuunnitelma on osa yritysarkkitehtuuria. Myös Booch (2010: 95) toteaa, että näiden molempien tulee elää ja toimia yhdessä, sillä molempien tavoite on sama: tukea yrityksen liiketoiminnallisia ja teknologisia tavoitteita.

Tutkimuksen kannalta olennaista on ymmärtää mistä elementeistä uudentyyppiset tietojärjestelmäratkaisut, eli tutkimuksen tapauksessa SharePoint hybridiratkaisut, muodostuvat, jotta voimme tunnistaa kohdeyrityksen kyvykkyyden tukea niitä.

Sovellusarkkitehtuuri tarjoaa tarkentavan mallin sovelluksien kehittämiseksi niin, että ne tukevat liiketoiminnan vaatimuksia ja varmistaa tarvittavan laadun liiketoiminnan tarpeiden kattamiselle (Nogueira et al 2012: 103-104). Sovellusarkkitehtuuri kertoo yrityksen sovelluksien kokonaisuudesta. Termiä sovellus voidaan käsitellä monella tavalla, mutta tutkimuksessa sovelluksella tarkoitetaan sovellustyyppiä – esim. SharePoint, SAP ja M2 ovat sovelluksia.

Tämän tutkimuksen näkökulmasta järjestelmäsuunnitelma kertoo, minkälaisista osista, eli objekteista, sovellus muodostuu. Tämä määrittely on tutkimuksen osalta hyvin hankala, sillä objekti voidaan määrittellä usealla tasolla. Objekti voi olla sovellusmalli, kuten SharePoint Online tai SharePoint OnPremise. Toisaalta objekti voi olla SharePoint Onlinen sivustokokoelma, sen alisivusto, alisivustolla oleva tiedostokirjasto, tai yksi tiedostokirjaston käyttöoikeusmääritelmistä. Toisin sanoen järjestelmäsuunnitelman objektit voivat olla toistensa aliobjekteja. Tämä on esitetty esimerkin vuoksi kuvassa 2.



**Kuva 2.** Objektit voivat olla toistensa aliobjekteja.

Ohjelmisto on objektien kokonaisuus. Objektit voivat olla riippuvaisia toisistaan. Sovelluskoodi on objektien kokonaisuuden lopputulos, itse fyysinen sovellus koodin muodossa. Objektien tekniset vaatimukset tunnistaaksemme, uppoudumme tutkimuksen aikana sovelluskoodi-tasolle. Tällä ei tarkoiteta sitä, että tutkisimme yksittäisen sovelluksen koodia, vaan tavoitteena on tunnistaa ne koodikielät, tai koodikielten ryhmät, eli ne teknologiset viitekehykset joita sovelluksessa on käytetty. Kaikissa tutkimuksen sovellusmalleissa siihen asti ei tarvitse mennä, sillä esimerkiksi yksinkertaisimmat SharePoint Online-sovellukset ovat konfiguroitavissa vain käyttöliittymästä, eikä sovelluskooditasolle silloin ole tarvetta uppoutua.

## 2.2 Pilvilaskenta

Pilvilaskennassa (engl. Cloud Computing) kuluttaja ostaa ja hyödyntää virtuaalikoneita, jotka ovat rakennettu pilvipalvelutarjoajan fyysisten serveri-infrastruktuurien päälle (Li & Li 2013: 867). Nämä palvelut tarjotaan internetin välityksellä (Webber-Cross 2014: 30).

IaaS, eli Infrastructure as a Service, on alimman tason palvelu, jossa kuluttajalle tarjotaan ainoastaan pilven kautta välitettävä serveri-infrastruktuuri. Se sisältää oikeuden käyttää osaa toimittajan servereistä ja tietokantaservereistä. Toimittaja on vastuussa itse fyysisistä servereistä ja tarjoaa niitä vuokrattavaksi asiakkailleen. Näiden servereiden päälle kuluttaja voi asentaa sovellusaloja ja sovelluksia. (Li & Li 2013: 867, Webber-Cross 2014: 30).

PaaS, eli Platform as a Service, on palvelu, jossa kuluttajalle tarjotaan valmis alusta applikaatioiden kehitystä, julkaisua ja käyttöä varten. Yleisesti alustan mukana annetaan kuluttajalle käyttöön tietty API (engl. Application Programming Interface), eli ohjelmointirajapinta ja siihen liittyvät työkalut (Li & Li 2013: 867). Myös PaaS toimitetaan internetin välityksellä.

SaaS, eli Software as a Service, on palvelu, jossa kuluttajalle tarjotaan valmiiksi tuotettu applikaatio, eli sovellus (Webber-Cross 2014: 30). Kaikki applikaation ylläpidolliset toimenpiteet, aina infrastruktuurista ja alustasta, applikaation huoltoon asti, ovat SaaS-applikaatioissa asiakkaan näkökulmasta toimittajan vastuulla. On kuitenkin mahdollista, että toimittaja käyttää IaaS ja/tai PaaS-palveluita kolmannen osapuolen kautta, jolloin toimittajan vastuulle jää itse applikaation ylläpito. SaaS-toimittajien tavoitteena on minimoida infrastruktuurikustannukset, minimoiden kuitenkin myös sen negatiiviset vaikutukset asiakkaille (Li & Li 2013: 867). Hyvä esimerkki SaaS-sovelluksesta on webmail-palvelut, jotka tarjotaan selaimen kautta ja jonka vuoksi kuluttajan ei tarvitse asentaa sähköpostisovellusta omalle päätteellensä (Webber-Cross 2014: 30).

Kolmen eri mallin päällekkäisyyksiä ja esimerkkejä tunnetuista kunkin mallin toteutuksista voi tarkastella Salas-Zárate & Colombo-Mendoza (2012: 49) ehdotuksen mukaan laaditusta kuvasta 3. Kuvasta on helppo tunnistaa, kuinka IaaS tarjoaa infrastruktuurin, PaaS sen päälle rakennettavan sovellusalustan ja SaaS tarjoaa niin sanotusti koko kakun - eli myös itse sovelluksen.

<b>SaaS</b> Software as a Service	<b>Sovellus</b> Liiketoimintasovellus, Web Servicet, Multimedia	Google Apps, Facebook, YouTube, Salesforces.com
<b>PaaS</b> Platform as a Service	<b>Alusta</b> Järjestelmäviitekehys (Java/Python/.NET..), Tietovarasto (tietokannat, tiedostot)	Microsoft Azure, Google AppEngine, Amazon SimpleDB/S3
<b>IaaS</b> Infrastructure as a Service	<b>Infrastruktuuri</b> Laskenta (VM), tietovarasto (lohko)	Amazon EC2, GoGrid, Flexiscale
	<b>Laitteisto</b> CPU, muisti, levy, siirtonopeus	Data Centerit

**Kuva 3.** Pilvipalveluarkkitehtuuri

Pilvipalveluiden käyttöönotto voi tuoda yrityksille merkittäviä säästöjä, mahdollistaa nopean käyttöönoton ja ennen kaikkea palveluiden nopean, helpon ja lähes rajattoman skaalattavuuden (Li & Li 2013: 866-867), sekä korkean tason saatavuuden (Webber-Cross 2014: 30). Pilvipalvelut käyttävät niin sanottua pay-as-you-go-mallia, jolloin käytöstä maksetaan käyttötason mukaan, ja se mahdollistaa kustannustehokkaamman resurssinhallinnan (Li & Li 2013: 866-867). Pilvipalvelut mahdollistavat omista tietokeskuksista (engl. data center) luopumisen, ja ennen kaikkea niiden ylläpidosta johtuvista kustannuksista irtaantumisen (Li & Li 2013: 866-867, Webber-Cross 2014: 30).

Microsoft Azure on yhteisnimitys Microsoftin pilvipalvelutarjonnalle (Webber-Cross 2014: 32). Microsoft tarjoaa pilvipalveluja Azuren nimen alla IaaS- ja PaaS-mallisina palveluina (Webber-Cross 2014: 32, Li & Li 2013: 867). Microsoft Azure tarjoaa pilvessä toimivan alustan esimerkiksi SharePoint-applikaatioille. SharePointin tapauksessa Azuren päälle tulee asentaa jokin SharePoint Server, jonka päälle voidaan rakentaa liiketoimintaa hyödyttäviä applikaatioita (IaaS-malli). Toisaalta Microsoft tarjoaa Azuren päällä olevan SharePoint Online-ratkaisun, joka toimii PaaS-mallisesti. Microsoft Azure on Webber-Crossin mukaan (2014: 33) itsestään selvä valinta, kun halutaan kehittää .NET-sovelluksia.

### 2.3 Prosessi- eli kontrolliteoria

Prosessi- eli kontrolliteorian tarkoitus on ymmärtää, kuinka organisaatiossa oleva henkilö tai ryhmä, voi varmistua siitä, että toinen organisaation henkilö tai ryhmä omaa ja työskentelee kohti yhteistä tavoitetta. (Kirsch 1996: 1)

Ouchi (1979: 833-837) määrittelee organisaatiolla olevan kolme tapaa kontrollointiin: 1) markkinamekanismi, jolla viitataan yksilön itsenäiseen toimintaan markkinaehdoin, 2) byrokraattimekanismi, jossa yksilön toiminnan tehokkuutta ja laatua kontrolloidaan esimiestoimin, ja 3) sosiaalinen mekanismi, tai klaanimekanismi, jonka perustana on luottamus yksilön itsenäiseen tekemiseen osana organisaatiota.

Voidaan määritellä, että prosessit perustuvat kontrolloitaviin, ja niitä ohjaaviin eli kontrollereihin. Kontrolloitava voi olla esimerkiksi työntekijä, ja kontrolleri voi olla esimies. Kontrollitapana voidaan käyttää byrokraattisia menetelmiä, kuten sääntöjä ja prosesseja, käyttäytymisperusteisia tai ts. sosiaalimekanismiperusteisia tapoja kuten luottamusta, tai markkinaperusteisia tapoja kuten yksilön palkitsemista.

Kirsch (1996: 1) toteaa informaatiojärjestelmäkehityksen (engl. ISD, Information Systems Development) olevan prosessi bisnestavoitteiden tunnistamiseksi osaksi tietojärjestelmää. Kirschin (1996: 1) mukaan prosessi sisältää seuraavat vaiheet: suunnittelu, rakentaminen, järjestelmätestaukset, implementointi, ja järjestelmän ylläpitotoimet. Tutkimuksessa keskitytään erityisesti järjestelmätestaukseen ja implementointiin, ja niiden prosesseihin. Suunnittelu, ylläpitotoimet, ja näistä käynnistyvät kehitystoimet ovat osana kehitettäviä prosesseja, mutta niiden kontrollointi on rajattu ulos tutkimuksen tavoitteiden mukaisista prosesseista. Tutkimuksessa keskitytään erityisesti järjestelmätestauksen ja implementoinnin kontrollointiin.



Kirsch (1996: 3) päättelee, että vaikka kontrollerilla olisi prosessi kontrolloitavan tekemisten seuraamiseksi, se ei ole sama asia, kuin että kontrolleri tekisi toimia kontrolloitavan käytöksen muuttamiseksi. Toisin sanoen kontrolleri ei prosessista huolimatta välttämättä ymmärrä kontrolloitavan prosessien toimien vaikutusta. Kirsch (1996: 14) ehdottaa, että informaatiojärjestelmäkehityksessä mitä suurempi kontrollerin ymmärrys on järjestelmäkehitysprosessista, sitä todennäköisemmin kontrolleri voi käyttää yksilön käyttäytymisperusteisia kontrolleja. Toisaalta Kirsch (1996: 15) myös toteaa, että mikäli järjestelmäkehitysprosessissa kontrolleja halutaan asettaa sellaisten henkilöiden toimesta, joilla on vähän tai ei lainkaan kokemusta kontrolloitavalta alueelta, käyttäytymisperusteinen kontrolli ei ole soveltuva vaihtoehto.

Tutkimuksen kontrolloitavat yksilöt rajoittuvat kohdeyrityksen sovellusomistajiin ja heidän kustantamiin ulkoistettuihin kehittäjiin. Nämä yksilöt ovat vastuussa pääasiassa suunnittelu- ja kehitystoimista.

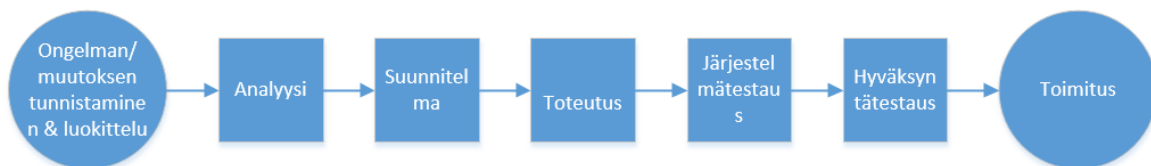
#### 2.4 Prosessin hallinta sovelluksen elinkaaren tukena

Clobridgen (2017: 63) mukaan vahvan hallintomallin rakentaminen on yksi tärkeimmistä huomioon otettavista asioista SharePoint-ratkaisuja suunnitellessa. Hallintomallilla voidaan tarkoittaa SharePointin kokonaisuuden hallintaa organisaatiossa – sitä, että uusia SharePoint-sovelluksia luodaan vain tarvittaessa, niitä luodaan vain organisaation tarpeisiin sopiksi, ja että yhteisiä ohjeistuksia sovelluksien ja sen objektien luomisessa ja ylläpidossa noudatetaan. Vahva hallintomalli voidaan ottaa vastaan organisaatiossa vastahakoisesti, mutta selvät ohjeistukset sovelluksien ja sen objektien hallinnasta ylätasolla tekee sovelluskentästä avoimemman ja vähentää väärinkäsityksiä (Clobridge 2017: 63).

Webber-Crossin mukaan (2014: 38) suurien organisaatioiden pilvipalvelujärjestelmät vaativat todennäköisesti uusia prosesseja niiden hallintaan. Tämä tukee tutkimuksen

lähtökohtaa: tarvitsemme uudet prosessit pilvipalvelujen tuoman muutoksen yhteydessä SharePoint-sovelluksien osalta.

Mamone (1994: 76) määrittelee sovelluksen laadunvarmistusprosessin koostuvan seuraavista vaiheista: ongelman/muutoksen tunnistaminen ja luokittelu, analyysi, suunnitelma, toteutus, järjestelmättestaus, hyväksyntättestaus, toimitus. Tämä korkean tason määritelmä vastaa kohdeyrityksen nykyisten SharePoint-sovellusten ylläpitomallia. Sovelluksen laadunvarmistusprosessirunkoa on kuvattu visuaalisesti kuvassa 4.



**Kuva 4.** Sovelluksen laadunvarmistusprosessin runko.

## 2.5 Kohderyhmät sovellusten laadunvarmistusprosesseissa

Mamone (1994: 75) määrittelee sovellusylläpidon olevan mitä tahansa sovellustuotteen käyttöönoton jälkeisiä muokkaustoimenpiteitä, jotka korjaavat virheitä (engl. Corrective), parantavat sen tehokkuutta tai ominaisuuksia (engl. Perfective), tai sopeutumista muuttuvaan toimintaympäristöön (engl. Adaptive). Koska nykypäivän bisnessovellukset ovat arkkitehtuuriltaan hyvin hajautettuja, ja ne voivat koostua useista eri sovelluksista (esim. SharePoint ja SAP), vaaditaan niiden ylläpitoon useita eri kohderyhmiä, joilla kaikilla on oma specialiteettinsa ja siten vastualueensa. Yksi tutkimuksen tavoitteen mahdollistavista lähtökohdista on tunnistaa objektien ylläpidon mahdollistavat tietotaidot. Implementointi- ja muutosprosessien luomiseksi tutkimuksessa täytyy myös tunnistaa kehitykseen ja hallintaan

osallistuvien kohderyhmien tietotaidot, jotta vastuu kunkin tehtävän osalta voidaan määritellä oikeille kohderyhmille. Jos, ja todennäköisesti kun, vastuita hajautetaan useille eri yrityksen kohderyhmille, oli ne sitten yrityksen sisäisiä tai ulkopuolisia tahoja, on ylläpidollisten laadunvarmistusprosessien keskitetty hallinta ja seuranta merkittävässä roolissa.

### 3 TUTKIMUSMENETELMÄ

Tämän kappaleen alussa käydään läpi tutkimukseen sopivia tutkimusmetodeja, joita vertaillaan keskenään ja niistä valitaan se, joka tukee tutkimuksen tavoitteita parhaiten.

#### 3.1 Tutkimusmenetelmien vertailua

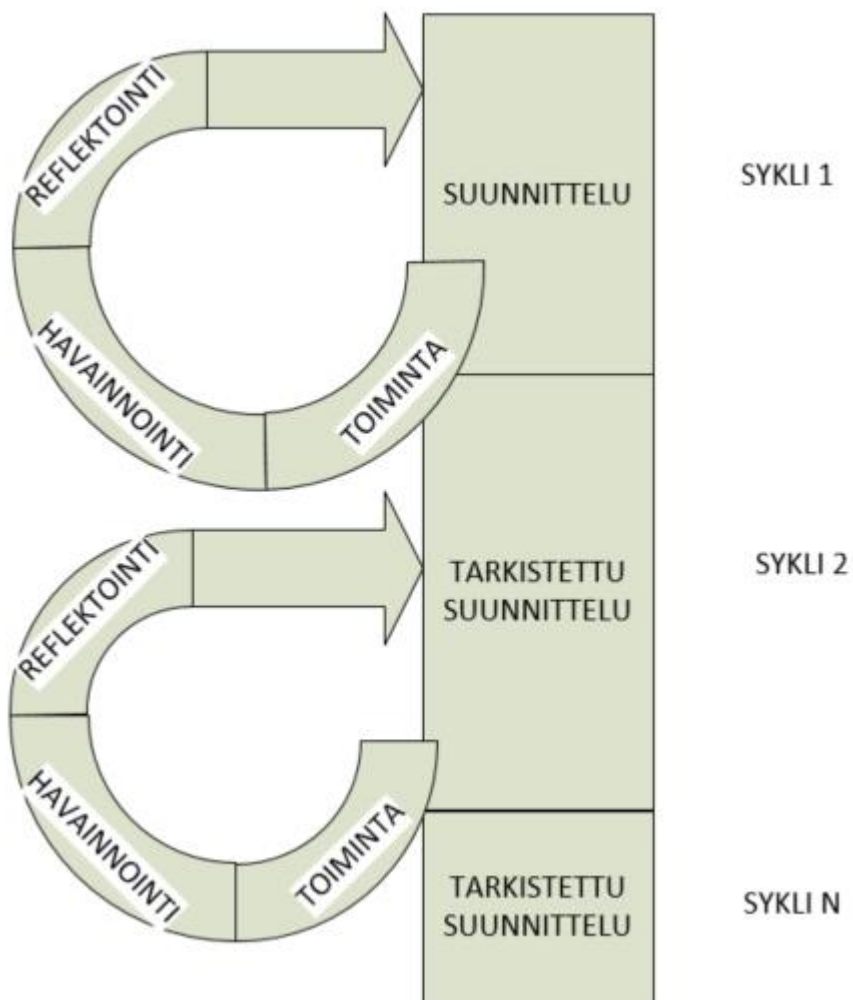
**Toimintatutkimus** (engl. Action Research) on tutkimusmetodi, jonka tavoitteena on luoda ratkaisu olemassa olevaan ongelmaan niin, että tutkija osallistuu itse aktiivisesti ratkaisun etsimiseen tai luomiseen, yhteistyössä kohdeyrityksen kanssa, ja käyttää siinä hyväkseen tutkimustietoa. Iversen, Mathiassen & Nielsen (2004: 424) mukaan toimintatutkimus kehittää ja muuttaa käytäntöjä tutkimustiedon avulla ja samalla tuottaa tietoa tutkimukseen käytännöstä. Toimintatutkimus on empiirinen tutkimus (Iversen et al. 2004: 408).

Toimintatutkimus sopii kaikille tieteen aloille, joissa on suorittajia ja tulosten tarvitsijoita eli asiakkaita, ja missä toiminnoissa voidaan saavuttaa yhteisymmärrys, tai missä tavoitteiden ja taustalla olevien arvostusten välille ei muodostu ristiriitaa (Anttila 2014: 7.4.5). Toimintatutkimuksen lähtökohtana on kohdeyrityksessä havaittu käytännön ongelma tai kehityskohde, joka halutaan ratkaista ja tuottaa samalla lisää tieteellistä tietämystä (Davidson, Martinsons & Kock 2004: 65; Sein, Henfridsson, Purao, Rossi & Lindgren 2011: 39). Tiaisen (2014) mukaan tutkimus on toimintatutkimus, mikäli tutkijat toimivat tiiviissä yhteistyössä kohdeyrityksen kanssa ja työn tulos on tarkoitus ottaa käyttöön kyseisessä yrityksessä. Toimintatutkimuksen relevanssi mitataan sen tuloksen hyödyllisyydestä organisaatiolle ja sen kehittäjille (Cole, Purao, Rossi & Sein 2005: 326). Prosessin luomiseen voidaan käyttää soveltuvaa metodologiaa, kuten toimintatutkimusta (Järvinen 2007: 51-53).

Toimintatutkimuksen taustalla on yrityksen, tai sen osan, tarve kehittää tai muuttaa toimintaa. Se etenee sykleissä, joissa tutkittavaan toimintaan tehdään pieniä muutoksia. Muutosten

vaikutusta arvioidaan. Toimintatutkimus on vahvasti sidoksissa työ- ja toimintatilanteeseen. (Tiainen, Aittoniemi, Haukijärvi & Yli-Karhu 2015: 2)

Linturin (2003) mukainen ehdotus reflektiivisestä toimintatutkimuksesta piirrettynä alla kuvassa 5. Kuva tuo esiin toimintatutkimuksen luonteen syklisenä tutkimusmetodina.



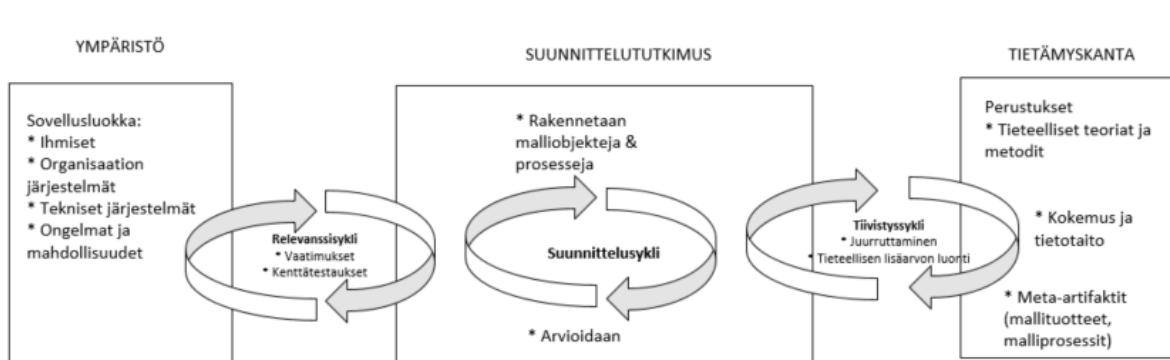
**Kuva 5.** Reflektioivan toimintatutkimuksen eteneminen.

Suunnittelututkimus (engl. Design Science Research) poikkeaa luonnontieteistä, vaikka niiden välillä on usein sidos. Suunnittelu ja luovuus ovat osana useita ammattiryhmiä, joita pääasiassa opetetaan luonnontieteiden kautta. Näitä ovat esimerkiksi lääketiede ja biologia, sekä finanssi ja matematiikka. Luonnontieteet tutkivat ja selittävät asioita niin kuin ne ovat. Toisaalta suunnittelun tarkoitus on selittää ja tutkia asioita niin kuin niiden tulisi olla. Molempia, sekä luonnontieteiden ymmärrystä, että luovuutta, tarvitaan osana kokonaisvaltaista koulutusta ja työelämän suunnitteluprosessia. Ironista lienee onkin, että tämä balanssin löytyminen itsessään on suunnitteluongelma. (Simon 1984: 129-133)

Suunnittelututkimuksen tarkoitus on maksimoida hyötyfunktion odotusarvo, ei maksimoida itse hyötyfunktioita (Simon 1984: 135). Suunnittelu oikeassa maailmassa ei ole vain vastauksien kasaamista palikoista, vaan soveltuvien kasausten löytämistä (Simon 1984: 143-144). Suunnittelussa ei ole soveltuvaa tutkia polkua, kunnes se onnistuu täysin, tai epäonnistuu varmasti, vaan etsiä useita vaihtoehtoisia polkuja tavoitteen saavuttamiseksi matkan varrella (Simon 1984: 144). Suunnittelututkimus voi koostua objektin luomisesta ja arvioinnista, sekä niihin liittyvien teorioiden aktiviteeteista (Cole et al. 2005: 326).

Kuvassa 12 esitellään Hevnerin (2007: 88) ehdotuksen mukainen kuvaus suunnittelututkimuksen osa-alueista. Kuvan keskiössä on suunnittelututkimus (engl. Design Science Research) ja se on sidoksissa sekä ympäristöön (engl. Environment), ja tietokantaan (engl. Knowledge Base). Ympäristön ja suunnittelututkimuksen välissä toimii relevanssisykli (engl. Relevance Cycle), joka kuvaa näiden kahden osa-alueen välistä sidosta. Relevanssisyklissä tutkija ja kohdeyritys määrittelevät suunniteltavan prosessin vaatimuksia, ja toisaalta prosessin valmistuttua sen testausta käytännössä. Toisella puolella suunnitelmatutkimusta ja tietokantaa yhdistää tiivistysykli (engl. Rigor Cycle), joka kuvaa

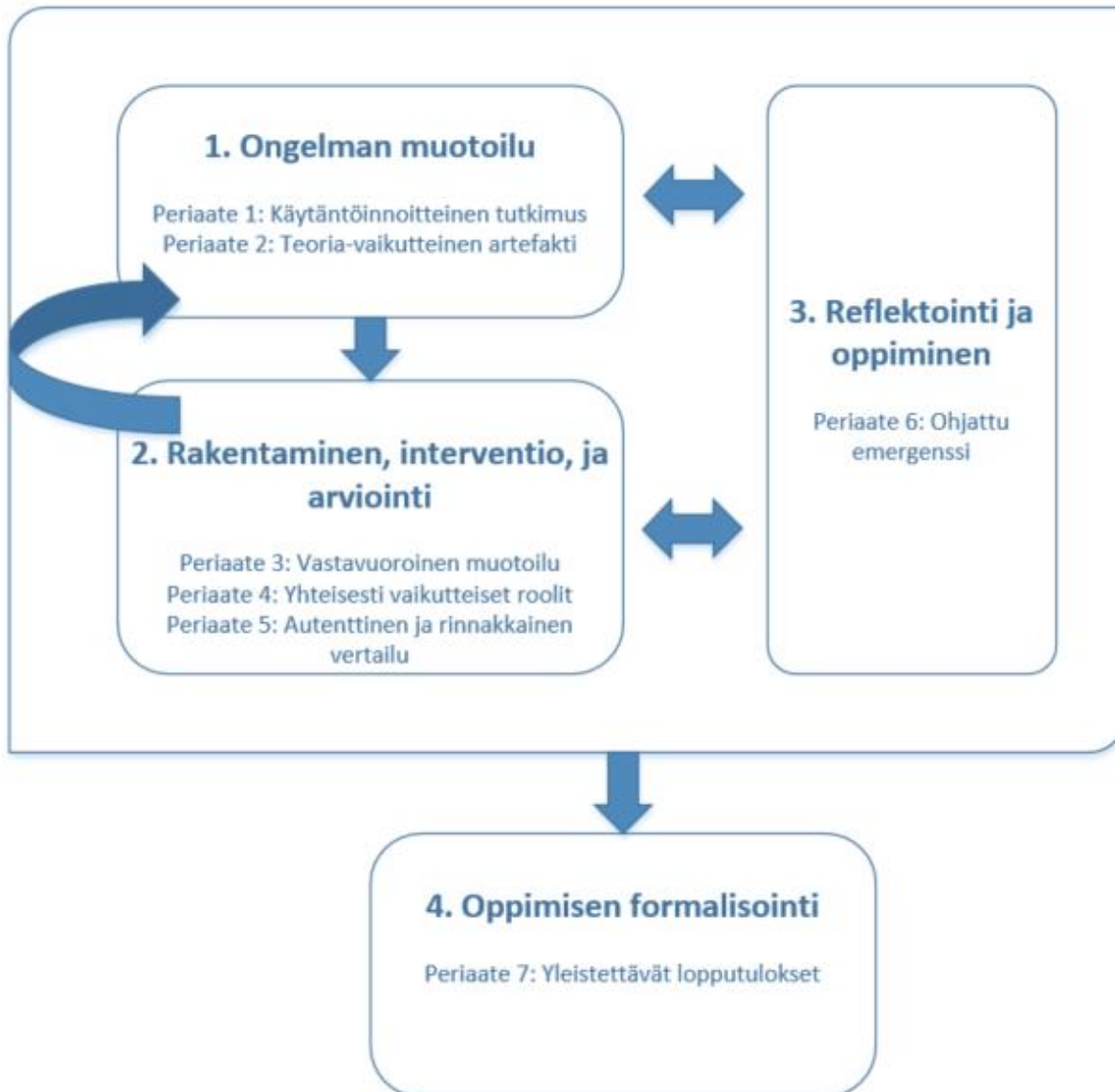
näiden kahden osa-alueen sidosta. Tässä syklissä tutkija tuottaa tieteellistä tietoa vertaistensa arvioitavaksi, ja toisaalta hakee lisätietoa prosessin kehityksen edistämiseksi.



**Kuva 6.** Suunnittelututkimuksen syklit.

Toiminnan suunnittelututkimus (engl. Action Design Research) on yhdistelmä toiminta- ja suunnittelututkimuksista. Toiminnan suunnittelututkimuksen keskiössä on organisaation toimintaan sopivan IT-artifaktin kehittämisessä ja arvioimisessa (Sein et al. 2011: 39-40). Metodi soveltuu erinomaisesti tuotteiden suunnittelua varten (Tiainen et al. 2015: 20), mutta sen soveltuvuus organisaation tietohallinnon toimintaprosessien suunnittelussa on kyseenalainen, koska se jättää organisaation käytäntöjen muutoksen sivuasiaksi (Tiainen et al. 2015: 24). Kuvassa 13 esitetään Sein et al. ehdotuksen mukainen (2011: 41) mallinnus toiminnan suunnittelututkimuksesta. Malli jakaantuu neljään osa-alueeseen (Sein et al. 2011: 40-45.):

1. Ongelman muotoilu
2. Rakentaminen, interventio, arviointi
3. Reflektio ja oppiminen
4. Oppimisen formalisointi



**Kuva 7.** Toiminnan suunnitelmatutkimuksen osa-alueet.

### 3.2 Tutkimusmetodien yhteenveto ja valinta

Toimintatutkimus on suunnittelututkimukseen verrattavissa oleva tutkimusmetodi. Suunnittelututkimuksessa tutkijan tehtävä on luoda ratkaisu organisaation ongelmaan, mutta hän on kohdeorganisaation ulkopuolinen kehittäjä, eikä tee tiivistä yhteistyötä



kohdeorganisaation henkilöstön kanssa. Toisaalta toimintatutkimuksen vertauskohtana voidaan pitää myös tapaustutkimusta (engl. Case Study), jossa tutkija on organisaation ulkopuolinen tarkkailija, joka kerää tietoa tutkimusta varten, mutta ei osallistu käytännön osalta tutkittavan ongelman ratkaisuun organisaatiossa. (Tiainen et al. 2015: 3)

Toiminnan suunnittelututkimus yhdistää osa-alueita sekä toimintatutkimuksesta, että suunnittelututkimuksesta. Se soveltuu hyvin tuotteiden suunnittelun tutkimusmetodiksi, mutta sen soveltuvuus organisaation tietohallintoprosessien tutkimusmetodiksi on kyseenalainen.

Alla olevassa kuvassa 8 kiteytetään tähän tutkimukseen harkittujen tutkimusmetodien vertailua tutkimustavoitteiden mukaisesti. Vihreällä pohjalla olevat tekstit kertovat mihin tutkimus soveltuu, kun taas punaisella pohjalla väritetyt kertovat mihin tutkimusmetodi ei sovellu.

Tapaustutkimus (engl. case study)	Suunnittelututkimus (engl. design research)	Toiminnan uunnittelututkimus (engl. action design research)	Toimintatutkimus (engl. action research)
- ratkaisu organisaation käytännön ongelmaan	+ ratkaisu organisaation käytännön ongelmaan	+ ratkaisu organisaation käytännön ongelmaan	+ ratkaisu organisaation käytännön ongelmaan
+ hyödyntää tutkimustietoa	+ hyödyntää tutkimustietoa	+ hyödyntää tutkimustietoa	+ hyödyntää tutkimustietoa
- yhteistyö kohdeorganisaation kanssa	- yhteistyö kohdeorganisaation kanssa	+ yhteistyö kohdeorganisaation kanssa	+ yhteistyö kohdeorganisaation kanssa
- soveltuu prosessikehitykseen	+ soveltuu prosessikehitykseen	- soveltuu prosessikehitykseen	+ soveltuu prosessikehitykseen

**Kuva 8.** Soveltuvien tutkimusmetodien vertailua.

Tutkimusmetodeista tähän tutkimukseen sopivimmaksi osoittautui toimintatutkimus. Se tukee tutkimusta kaikilta halutuilta osin: tuotetaan ratkaisu käytännön ongelmaan (a), se tehdään tiiviissä yhteistyössä kohdeyrityksen kanssa (b), tutkimustietoa hyödyntäen (c), ja se soveltuu prosessikehitykseen (d).

### 3.3 Toimintatutkimusprosessi

Toimintatutkimuksen lähtökohtana on usein käytännön ongelma, jonka kehittämistarve johtuu ympäristön jatkuvasta muutoksesta (Linturi 2003), ja olennainen osa sen ratkaisuun on yhteistyö kohdeyrityksen kanssa. Jotta kohdeyrityksen näkemykset suunnitellusta ratkaisusta saadaan mukaan, tulee tutkimus toteuttaa sykleissä. Toimintatutkimus on aina syklistä (Tiainen et al. 2015: 8). Iteratiivisessa toimintatutkimuksessa ongelman tunnistaminen ja toiminta vuorottelevat, kunnes hyväksyttävä ratkaisu löydetään (Tiainen et al. 2015: 8). Linturin (2003) mukaan toimintatutkimus on loputon syklinen prosessi.

Vaikkakaan tämä tutkimus ei ole muodoltaan kvalitatiivinen, Irvine & Gaffikin (2006: 115) ehdotuksen mukainen kvalitatiivinen tutkimusprosessi korkealla tasolla antaa hyvän yleiskuvan tutkimuksen toteuttamisesta organisaatiossa ja sen vaiheista. Se koostuu kolmesta vaiheesta: *pääsy organisaatioon, toiminta organisaatiossa, ja poistuminen organisaatiosta*. Tutkija on jo valmiiksi organisaatiossa, joten pääsy organisaatioon, joka Irvinen ja Gaffikin (2006: 121) mukaan on erityisen hankala vaihe, on jo selvitetty. Tämä tutkimus koostuu pitkälti tutkijan toiminnassa organisaatiossa – tutkija on muutoksen keskiössä ja keskustelelee oleellisten yrityksen sisäisten kohderyhmien kanssa. Tärkeää on myös tunnistaa missä vaiheessa tutkimusta ”poistutaan organisaatiosta”. Tutkija poistui organisaatiosta työn tulosten palautusten jälkeen, kun tutkimuksen tavoitteen mukaiset prosessit oli luotu ja hyväksytty kohdeorganisaatiossa. Irvinen & Gaffikinin määritelmien mukaiset vaiheet tämän tutkimuksen osalta on määritelty kuvassa 9.

Pääsy organisaatioon (getting in)	Toiminta organisaatiossa (getting on)	Poistuminen organisaatiosta (getting out)
Tutkija on ollut osana tutkimuskohteen organisaatiota 3½ vuotta	Tutkija käyttää hyväkseen tieteellistä tietoa tutkimusongelman ratkaisemiseksi	Tutkimusongelma on ratkaistu hyväksytysti -> Ylläpitoprosessi arvioitu ja käyttöön otettu

**Kuva 9.** Toimintatutkimuksen tutkija osana organisaatiota.

Tämän tutkimuksen päämääräinen tarkoitus on saada kohdeyritykselle käyttöönotettavat prosessimallit, vaikka voidaankin olettaa, että näitä prosesseja kehitetään jatkuvasti vielä tutkimuksen päätyttyäkin. Tutkimuksen toteutustapa tukee myös prosessien jatkokehitystä.

Tutkimus toteutetaan iteratiivisena toimintatutkimuksena, jonka ominaispiirteenä on syklinen ongelman ja toiminnan vuorottelu (Tiainen et al 2015: 10), kunnes hyväksytyt prosessit ovat saavutettu. Tutkimuksen syklien vaiheet toteutetaan Susman & Everedin (1978: 588) ehdotuksen mukaisesti seuraavasti:

1. Ongelman määrittely (engl. Diagnosing)
2. Toiminnan suunnittelu (engl. Action Planning)
3. Toiminnan toteutus (engl. Action Taking)
4. Arviointi (engl. Evaluating)
5. Opitun tunnistaminen (engl. Specifying Learning)

Tämä yleisluontoinen vaihemäärittely toimii parhaiten, mikäli tarvittavien syklien määrä ei ole tiedossa. Tutkimus on päätetty toteuttaa kahdessa syklissä, joten syklien vaiheiden määritelmät ovat hieman erilaiset molemmissa sykleissä, ja ne voidaan tunnistaa jo tässä

vaiheessa tutkimusta. Kuvassa 10 on tarkennettu molempien syklien vaiheet tämän tutkimuksen osalta.

SYKLI	VAIHE	Susman & Evered (1978) ehdotuksen mukainen vaihemäärittelmä	Tutkimuksen vaihemäärittely
1	1	Ongelman määrittely	Tutkimusongelman määrittely käytännön ongelman perusteella
1	2	Toiminnan suunnittelu	Tutkimuksen suunnittelu
1	3	Toiminnan toteutus	Olettamien muodostaminen prosessikehityksen tueksi
1	4	Arviointi	Olettamien arviointi haastattelumenettelyin
1	5	Opitun tunnistaminen	Tunnistetaan korjattavat oletamat haastattelun tuloksista
2	1	Ongelman määrittely	Määritellään korjausvaatimusten dilemmat prosessille
2	2	Toiminnan suunnittelu	Määritellään uudet oletamat
2	3	Toiminnan toteutus	Toteutetaan uudet prosessit ehdotukseksi
2	4	Arviointi	Prosessien arviointi kohdeyrityksen kohderyhmien avulla
2	5	Opitun tunnistaminen	Tunnistetaan jatkokehityskohteet, formalisoidaan oppiminen ja päätetään tutkimus

**Kuva 10.** Tutkimuksen syklien vaihemäärittelyt.

### 3.4 Suunnitelma tutkimuksen etenemisestä

Tässä vaiheessa tutkimusta olemme jo määritelleet ongelman ja suunnitelleet tutkimuksen etenemisen. Seuraavaksi tutkimuksessa tunnistetaan tiedonkeruumenetelmät tutkimuksen käytännön vaiheen osalta, tukemaan molempien syklien arviointivaihetta. Tämän jälkeen tunnistetaan ja kirjataan kohdeyrityksen hybridisovellusmalliin osallistuvat kohderyhmät, tunnistetaan soveltuvat teknologiat, luodaan ensimmäinen ehdotus hybridisovellusten implementointi- ja muutosprosesseista, jonka jälkeen toteutetaan ensimmäinen haastattelukierros. Ensimmäisen syklin arviointivaiheen jälkeen tunnistamme ensimmäisen ehdotuksen puutteet ja ongelmat, jotka siten voidaan tutkimuksen toisella syklillä määritellä, korjata, tai sisällyttää lopulliseen ehdotukseen prosesseista. Toisen haastattelukierroksen tavoite on arvioida haastattelumenetelmin prosessiehdotuksien toimivuutta käytännössä, ja kirjata jatkokehityskohteita prosessin kehityksen varmistamiseksi. Lopuksi kirjataan

johtopäätökset kohdeyrityksen, tieteellisen yhteisön ja tutkimuksen rajoitusten tunnistamiseksi.

### 3.5 Tiedonkeruu tutkimuksen aikana

Tutkimuksen tavoitteen mukainen ehdotelma prosesseista tulee arvioida. Tutkimusmenetelmän ja sen luonteen vuoksi prosessit arvioidaan haastattelumenetelmin. Ehdotettujen prosessien arvioinnissa käytetään hyväksi kohdeyrityksen tietohallintorakenteen eri sidosryhmiä, joilla on erilaisia näkökulmia prosesseihin – yksi sidosryhmä voi antaa kommentteja liiketoiminnan näkökulmasta, toinen taas mahdollisista teknisistä puutteista tai virheistä, kolmas prosessin hallinnan optimoimisesta. Tutkimuksen sidosryhmiä lisää kappaleessa neljä.

Haastattelussa voidaan käyttää kolmea lähestymistapaa: lomakehaastattelua, teemahaastattelua, tai avointa haastattelua (Tiainen: 2). Kuvassa 11 esitellään Tiaisen ehdotuksen mukaisia vertailuja erilaisten haastattelujen piirteistä.

	<b>Lomakehaastattelu</b>	<b>Teemahaastattelu</b>	<b>Avoin haastattelu</b>
<b>Kysymysten muotoilu</b>	Kiinteä, täsmälliset kysymykset	Suosituskysymykset, aihepiirit	Vapaa
<b>Kysymysalue</b>	Tiukasti määrätty	Teema määrätty	Vapaa
<b>Osallistujamäärä</b>	Suuri	Melko pieni	Pieni (muutama)
<b>Saatu tieto</b>	Pintapuolinen	Syvä	Syvä
<b>Työmäärä analyysivaiheessa</b>	Melko pieni	Suuri	Suuri

**Kuva 11.** Vertailu erilaisten haastattelujen piirteistä.

Tässä tutkimuksessa pintapuolinen haastattelu ei tuo prosessien optimointiin mitään lisäarvoa, joten lomakehaastattelu ei sovi tähän tutkimukseen. Valinta teemahaastattelun ja avoimen haastattelun väliltä osoittautui tosin mielenkiintoiseksi. Haastatteluun osallistutetaan melko pieni määrä ihmisiä, joilla jokaisella on syvä tietämys yhdestä tai useammasta osa-alueesta: nykyisistä prosesseista, niiden osista tai vaiheista, tai liiketoiminnallisesta merkityksestä. Koska haastatteluun osallistuvilla eri sidosryhmillä on kaikilla asiantuntemus eri osa-alueesta, ja siten selektiiviset havainnot, valittiin tutkimuksen haastattelumenetelmäksi teemahaastattelu. Tällä tavoin tutkija voi kohdentaa valittuun aiheeseen liittyvät kysymykset sopiville sidosryhmille.

Haastattelut toteutetaan mahdollisuuksien mukaan kasvotusten, tai vaihtoehtoisesti puhelinpalaveriin. Tiaisen mukaan (2) teemahaastattelussa haastattelijan ammattitaito, vuorovaikutusosaaminen ja asiantuntemus korostuvat, sillä haastattelijalla on vastuussa haastattelun etenemisestä ja aiheesta pysymisestä. Haastattelutilanteissa tutkija joutuu ottaa huomioon oman teknisen ymmärryksen rajallisuuden, jonka vuoksi teknisiin seikkoihin keskittyvät haastattelut ovat luonteeltaan avoimia keskusteluja, joissa kysymykset tarkentuvat saatujen vastausten perusteella.

Kuten aiemmin tutkimusmenetelmä kappaleessa mainittiin, tutkimus suoritetaan kahdessa syklissä. Täten myös haastattelut toteutetaan kahdessa kierroksessa. Haastatteluiden sisältö ja tavoite vaihtelevat kohderyhmittäin, mutta myös kierroksittain. Ensimmäisen haastattelukierroksen tavoite on varmistua tutkijan tekemän taustatutkimuksen paikkansapitävyydestä ja saada tarkennetut vaatimukset prosessien osa-alueiden osalta. Koska uuden sovellusmallin mukaisia prosesseja ei ole luotu aiemmin kohdeyrityksessä edes konsernitason tasolla, ensimmäisellä kierroksella selvitetään myös kohdeyrityksen vaatimuksia ja suunnitelmia tällä tasolla. Toisen kierroksen haastattelu toimii korjattujen olettamien tarkasteluna, vahvistamisena, sekä jatkokehityskohteiden tunnistamisen osana.

## 4 KOHDEYRITYKSEN SHAREPOINT-SOVELLUSMALLIT

Tässä kappaleessa käydään läpi SharePointia teknologisenä alustana ja kohdeyrityksen rakennetta SharePoint-ratkaisuihin liittyviltä osin. Lisäksi käydään läpi, miten erityyppisiä SharePoint-sovelluksia kohdeyrityksessä käytetään ja kuinka niiden implementointi- ja muutosprosessit toimivat tällä hetkellä. Nykyisistä sovelluksista tunnistetaan niiden käyttämät teknologiat ja mahdolliset rajoitteet hybridisovellusmalliin siirtymisen osalta.

### 4.1 SharePoint

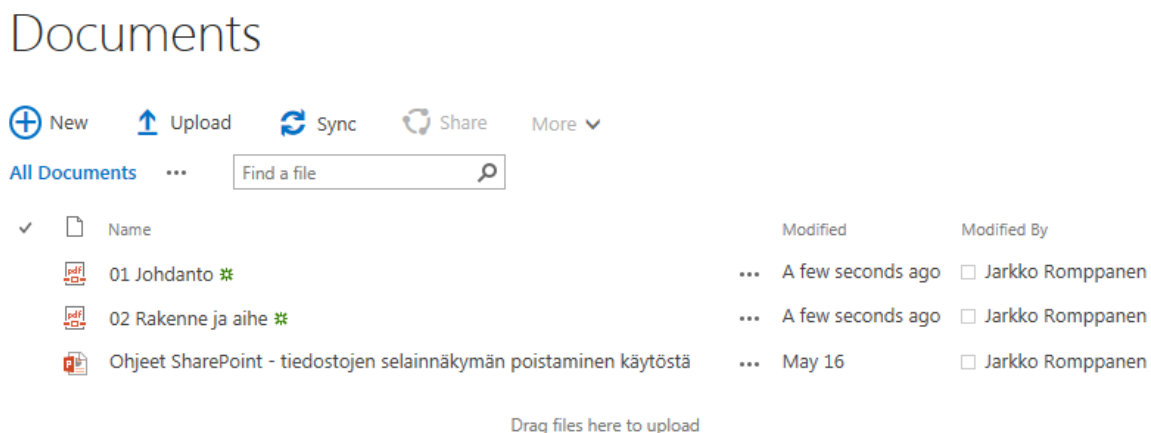
SharePoint on Microsoftin selaimessa käytettävä tiedonhallintatyökalu (engl. Content Management System). SharePoint tukee sekä sisällönhallintaa (engl. Content Management), että asiakirjahallintaa (engl. Document Management), ja on siten kattava tiedonhallintatyökalu.

SharePointilla on merkittävä markkina-asema tiedonhallintatyökalujen alueella. Sen suurimpia kilpailijoita ovat historiassa olleet mm. OnBase, Alfresco, IBM FileNet, Documentum, ja Liferay. Käytännössä suurin kilpailija on kuitenkin mikä tahansa verkkolevy, jota voidaan jakaa eri käyttäjille. SharePointilla on kuitenkin merkittävä etulyöntiasema kilpailijoihinsa nähden, sillä Microsoft on istuttanut SharePointin osaksi Office 365–tuoteperhepakettiaan tällä vuosikymmenellä, joka tarjoaa kilpailijoitaan paremmat mahdollisuudet mm. valmiisiin integraatioihin.

SharePointin avulla yritykset ja organisaatiot voivat luoda selaimella käytettäviä sivustoja. Sivustoja voidaan luoda useisiin eri käyttötarkoituksiin ja ne voidaan suunnata myös usealle eri yleisölle. Yleisimmin SharePointia käytetään tiimien sisäiseen kollaboratiiviseen työskentelyyn, eli tiedonjakoon, mutta sillä voidaan rakentaa myös kokonainen yrityksen sisäinen tiedonjakoverkosto eli Intranet. Toisaalta kustomoituna SharePointilla voidaan

tehdä jopa suurempia liiketoimintasovelluksellisia ratkaisuja. Tiimien sisäisen tiedonjaon käytötapauksessa sivustolle luodaan useimmiten tiedostokirjasto (engl. document library), jonne tiimi voi tallentaa yhteisiä tiedostojaan, ja jonka avulla kaikilla tiimin jäsenillä on pääsy


heille yhteisiin dokumentteihin. Toisaalta tiimi voi pitää myös kirjaa esim. asiakkaiden yhteyshenkilöistä tietuelistan (engl. custom list) avulla. Molemmat objektit voidaan luoda suoraan sovelluksen käyttöliittymästä oletustoiminnallisuuksina. Esimerkinäkymät tiedostokirjastosta SharePointissa kuvassa 12, ja tietuelistasta kuvassa 13.



**Kuva 12.** Esimerkki tiedostokirjastossa SharePoint Onlinessa (Romppanen 2017: 19).



# Yhteystiedot



 **new item** or edit this list

All Items

...

Find an item



✓	Nimi	Puhelinnumero	Osoite
	Essi Esimerkki 	... 0401234567	Yliopistonranta 1, 65200 Vaasa
	Eetu Esimerkki 	... 0501234567	Yliopistonranta 5, 65200 Vaasa

**Kuva 13.** Esimerkki tietuelistasta SharePoint Onlinessa (Romppanen 2017: 20).

SharePoint-sivustoille sallittuja käyttäjiä voidaan hallita SharePointin käyttöoikeushallintatyökaluilla. Käyttöoikeudet perustuvat pääasiassa yrityksen tai organisaation Windows AD-tunnuksiin, joiden avulla voidaan määritellä ja auktorisoida sivustolle halutut käyttäjät. Käyttöoikeuksia voidaan hallita sivustolle luotavien käyttöoikeusryhmien perusteella. Käyttöoikeusryhmiin voi lisätä myös Windows AD-ryhmiä, jolloin henkilöitä voidaan lisätä sivustolle SharePointista riippumattomasti. Kun halutut henkilöt ovat lisätty oikeisiin ryhmiin, annetaan tälle ryhmälle sivustolle tarvittavan tasoiset oikeudet. Yleisimmät oikeustasot ovat omistajille tarkoitetut rajoittamattomat käyttöoikeudet (Owners – Full Control), sisällöntuottajille tarkoitetut käyttöoikeudet (Members – Contribute) ja vierailijoille tarkoitetut käyttöoikeudet (Visitors – Read). Käyttöoikeusryhmille voi tarvittaessa antaa sivustolla myös eri osioihin, kuten tiedostokirjastoihin, eritasoisia oikeuksia.

## 4.2 SharePointin kehitys työkalusta liiketoimintasovellusalustaksi

Microsoft julkaisi SharePointin vuonna 2001 ja ensimmäistä versiota kutsutaan nykypäivänä julkaisuvuoden mukaisesti: SharePoint 2001. SharePointin ensimmäinen versio vastaanotettiin kaksijakoisin mielipitein. Se mahdollisti yksinkertaisemman ja vähemmän työllistävän tavan jakaa tietoa, mutta ohjelmisto oli hyvin virheherkkä, kömpelö käyttää, ja sitä oli lähes mahdoton kustomoida organisaation tarpeiden mukaiseksi. Lisäksi käyttöliittymä koettiin epämiellyttävänä. (Roumani 2015)

Vuonna 2003, Microsoft julkaisi uuden version SharePointista osana Office 2003–tuotepakettia. Microsoft julkaisi kyseisen version uudelleen brändättyinä nimellä Windows SharePoint Services, mutta versio tunnetaan yleisemmin julkaisuvuoden mukaista nimeä kantaen SharePoint 2003:na. Uusi versio oli merkittävästi edeltäjänsä käyttökelpoisempi. Se sisälsi paljon uusia mahdollisuuksia, kuten paremman käyttöliittymän, paremmat hakutoiminnot, ja antoi mahdollisuuden personoida sivustoja. Lisäksi kehittäjät pystyivät käyttämään samoja työkaluja uuden SharePointin hallintaan, kuin mihin he olivat tottuneet muiden Office–tuoteperheen tuotteiden kanssa. (Roumani 2015)

Merkittävin tähän asti tapahtunut muutos tapahtui vuonna 2006, kun Microsoft julkaisi SharePoint 2007–version. Se oli mahdollista saada sekä Office-paketin mukana, että erillisenä ohjelmistona. SharePoint 2007 vastasi useisiin edeltäjänsä piileviin kriittisiin ongelmiin ja se mahdollisti organisaatioille huomattavasti paremmat ja laajennettavat alustan keskeiset toiminnot. Uusi versio toi mukanaan myös merkittävän kehitysaskelen InfoPath-lomakkeiden muodossa. (Roumani 2015)

SharePoint 2007:a pidetään yleisesti ensimmäisenä yritysten realistisesti hyödynnettävissä olevana versiona siihenastisista SharePoint-tuotteista. Tämä näkemys on kuitenkin katsottuna tämän päivän kokemusten näkökulmasta, eikä kirjoittajalla ole tarkempaa omakohtaista kokemusta aiemmista SharePointeista, varsinkaan sen julkaisuajankohtana,

jolloin asianmukainen vertailu olisi ollut tehtävissä muihin silloisiin kilpaileviin sovelluksiin nähden.

SharePoint 2010 versio toi mukanaan integraatiot Office-tuoteperheen tuotteiden kanssa (Roumani 2015). Se mahdollisti myös käyttäjäprofiilit sosiaalisessa muodossa, toi huomattavia parannuksia käyttöliittymään ja käyttäjäkokemukseen (Roumani 2015). Omasta mielestäni, sivustojen kehittäjänä, tärkein lisäys 2010 versiossa oli työnkulkujen mahdollistaminen. Työnkulut mahdollistavat helposti kehitettävän automaatiologiikan luomisen sivustoille. Näillä automaatioilla voidaan esimerkiksi muuttaa tietuelistojen tietoja, tai lähettää automaattisia muistutusviestejä sähköpostitse, kun määritellyt ehdot täyttyvät. Automaatiot voitiin luoda lisätyökalun käyttöliittymästä, eikä niiden luomiseksi tarvittu koodausosaamista. SharePoint 2010 mahdollisti myös ylivoimaisesti helpommat mahdollisuudet luoda yhteyksiä muissa järjestelmissä olevan liiketoiminnalle kriittisen datan, kuten ERP-järjestelmädatan, ja SharePointin välille. SharePoint 2010 versio mahdollisti siten ensimmäisenä versiona kompleksisempien liiketoimintasovelluksien kehittämisen.

SharePoint 2013, vaikkakin hyödyllinen päivitys aiemmasta versiosta, ei sinänsä tuonut mukanaan mitään merkittäviä uusia mahdollisuuksia edeltäjänsä verrattuna. Uusi versio oli enemmänkin Microsoftin tapa luoda uusi mielikuva ohjelmistosta, paremman visuaalisen käyttäjäkokemuksen muodossa. Muutos oli siltä osin merkittävä – SharePoint 2013 toi mukanaan muissa Office-paketin tuotteissa nähdyn visuaalisen ilmeen, joten ohjelmiston käyttökin tuntui loppukäyttäjälle helpommalta, vaikka teknisesti siinä ei ollut tapahtunutkaan suuria muutoksia. Vaikka kyseinen versio ei tuonut pöytään mitään teknisesti maata mullistavaa, ei se tarkoita, etteikö Microsoft olisi tuon version yhteydessä luonut jotain merkittävää. Ennen SharePoint 2013-versiota, kaikki aiemmat versiot olivat kytköksissä organisaatioiden omilla servereillä toteutettuihin alustoihin. SharePoint Server 2013 ei itsessään eronnut siitä, mutta Microsoft julkaisi yhdessä SharePoint Server 2013-version

kanssa uuden mallin rakentaa SharePointin päälle: SharePoint Onlinen. Se oli visuaaliselta ilmeeltään SharePoint 2013:a vastaava, mutta se oli rakennettu Microsoftin pilvipalvelun päälle. Tämä mahdollisti organisaatioille huomattavasti kustannustehokkaamman tavan luoda sivustoja, sillä sen käyttäminen ei vaatinut kalliita investointeja servereihin, tai niiden hallintaan liittyviin henkilöstökuluihin.

SharePoint 2016 toi paljon kehittäjien toivomia uudistuksia ja keskittyikin pitkälti teknisiin ominaisuuksiin ja niiden parannuksiin. Tietokantojen kokoot viisinkertaistettiin ja sivustokokoelmien maksimäärää nostettiin kaksikymmenkertaiseksi (Dagar 2017). Myös loppukäyttäjien ongelmia ratkottiin nostamalla yksittäisen tiedoston koon maksimirajaksi 10GB edellisen version 2GB rajoituksen sijaan (Dagar 2017). Siltikin versio oli uudistuksena enemmän tai vähemmän vain muodollisuus. Microsoftin fokus SharePointin osalta oli selvästi siirtynyt räjähdysmäisen suosion saaneen pilvituotteen kehitykseen.

Syksyllä 2017 Microsoft ilmoitti Ignite-tapahtumassa julkaisevansa SharePointista vielä Server 2019 version (Mackie 2017). Microsoft on tehnyt strategisen päätöksen tulevaisuuteen sijoittamalla SharePointin käyttötarkoituksen nykyisestä julkaisu- ja tiedonjakokanavan roolista, kohti tuoteperheen keskiötä, jossa se toimii enemmän ja enemmän yritykselle suunnatun tuoteperheen tietovarastona. Käytännössä Microsoftin Teams -tuote on ottamassa SharePointin nykyistä käyttäjälle näkyvää roolia, mutta jossa tiedontallennus tapahtuu kuitenkin SharePointiin. On tärkeää huomioida, että tämä näkemys on suoranaisesti kytköksissä vain Office 365 paketin SharePointiin, eli SharePoint Onlineen.

#### 4.3 SharePoint sovellusmallit

**SharePoint Server**, joka tunnetaan yleisesti myös nimellä SharePoint OnPremise, on SharePoint-tuote, joka asennetaan ja jota hallitaan yrityksen omalla serverillä. Kaikki SharePoint-tuotteet ennen pilvipalvelussa toimivan SharePoint Onlinen julkaisua olivat

malliltaan tällaisia. SharePoint Server antaa yritykselle mahdollisuuden kustomoida SharePointia omiin tarpeisiinsa. SharePoint itsessään tarjoaa melko kattavat OOTB-toiminnallisuudet (out-of-the-box, eli oletustoiminnallisuudet), mutta ne harvemmin kattavat suurempien yritysten kaikkia tarpeita. Yleensä kustomoinnit ovat luonteeltaan sellaisia, että ne mahdollistavat uusia toimintoja, joita voidaan käyttää SharePointin ohessa. Ne voivat olla yksinkertaisia kustomointeja, kuten yrityksen brändi-ilmeen mukaiset sivustot, tai kompleksisemmat integraatiot muihin yrityksen käyttämiin järjestelmiin, kuten esimerkiksi toiminnanohjausjärjestelmiin (engl. ERP System, tai Enterprise Resource Planning System). Tällainen integraatio mahdollistaa tiedon siirtämisen SharePointista suoraan toiminnanohjausjärjestelmään, tai vaihtoehtoisesti toiminnanohjausjärjestelmästä voidaan esittää tietoa myös SharePointissa.

Toisaalta SharePoint Server-sovellukset tuottavat yrityksille myös päänvaivaa, varsinkin jos tämä malli on ollut yrityksellä käytössä kauemman aikaa, ja kun sovelluksia on alkanut servereille kertymään useampia – kymmeniä tai satoja. Yksittäisten sovellusten kanssa harvoin on suurempia ongelmia, ainakaan sen vuoksi että ne olisivat samassa SharePoint Server-farmissa. Farmi on servereiden linkitetty kokonaisuus, jossa serverit käyttävät yhteisiä komponentteja toiminnan tehostamiseksi, ja joiden tehojen jakoa hallitaan ohjelmallisesti.

Ongelmaksi muodostuu kaikkien serverillä olevien sovellusten kokonaisuus. Tällä hetkellä dominoivassa mallissa, aina kun SharePoint Server-sovelluksessa halutaan tehdä suurempia muutoksia, eli kun sovelluksesta asennetaan uusi versio asennuspaketein, vaatii se koko serverin väliaikaista alasajoa. Tämä aiheuttaa useita ongelmia. Osa servereillä olevista sovelluksista voivat olla yrityksille liiketoiminnallisesti merkittäviä – ja niiden tulisi olla aina saatavilla. Mikäli yhteen serverille asennettuun sovellukseen halutaan tehdä muutos, vaatii se myös muiden servereillä olevien sovellusten hetkellisen alasajon päivityksen asennushetkellä. Lisäksi tämä toimintamalli vaatii hyvin tarkkaa valvontaa sovellusten

asennuspakettien ja sovelluskoodin osalta. Yhdessä sovelluksessa oleva virhe saattaa aiheuttaa virheitä koko serverille – ja siten kaikkiin muihinkin sinne asennettuihin sovelluksiin. Suuressa yrityksessä sovelluksen laadunvarmistus ja sen varmistavat riskinhallintatehtävät aiheuttavat lisäksi merkittäviä viiveitä valmiinkin sovelluksen käyttöönotossa. Sovelluksien käyttöönotto voi vaatia esimerkiksi arkkitehtuuri- ja koodikatselmuksia, sekä useita eri asennuksia testiympäristöihin, ennen kuin lopullinen käyttöönotto, eli tuotantoympäristöasennus voidaan tehdä. Mikäli SharePoint Serverillä on kymmeniä applikaatioita, voi asennuspäivät ja asennusajatkin olla hyvin tarkkaan määriteltyjä, joka tekee käyttöönotosta sitäkin jäykempää.

SharePointin versiopäivitykset, eli siirtymiset SharePoint Server versiosta toiseen (esim. 2010 -> 2013), aiheuttaa yrityksille suurta päänvaivaa. Versiopäivitykset itsessään tuovat aina uusia hyödynnettävissä olevia toiminnallisuuksia, ja yritykset siten lähes aina haluavat toteuttaa versiopäivityksen mahdollisimman nopeasti. Päänvaivan aiheuttaa melko ironisestikin se, että yrityksen omat SharePoint Server-sovelluksien kustomoinnit käytännössä estävät versiopäivityksen vaivattoman toteuttamisen. Pelkkiä OOTB-ratkaisuja sisältävän SharePoint Serverin versiopäivitys on melko mutkatonta, mutta kustomoinnit eivät päivity vaivattomasti. Tämä johtaa siihen, että varsinkin suuret yritykset joutuvat versiopäivityksen yhteydessä tekemään massiivisen, helposti kuukausiakin kestävän migraatioprojektin, jonka yhteydessä kaikki toteutetut kustomoinnit joudutaan käytännössä kirjoittamaan uudelleen uuteen versioon sopiviksi. Nämä migraatioprojektit tuovat yritykselle merkittäviä kustannuksia. Lisäksi vaihtoehtoiskustannuksena migraatioprojektit viivyttävät versiopäivityksen tuomien ilmaisten toiminnallisuuksien käyttöönottoa.

Voidaankin yleistää, että SharePoint Server sovellusten kustomointien teettäminen voi olla edullista ja kannattavaa, mutta yleensä niiden ylläpidosta ja versiopäivityksistä aiheutuvat kustannukset saattavat tehdä kustomoinneista kannattamattomia. Näitä sovelluksen pidemmän elinkaaren mahdollistavia toimenpiteitä ei usein osata huomioida tarpeeksi hyvin

kustomoinnin toteutusvaiheessa, tai vähintäänkin niiden tarkka monetaarinen arviointi on hyvin hankalaa. Mikäli yrityksellä on kokemuksia versiopäivityksien aiheuttamista ongelmista ja kustannuksista, saattaa se empiä uusien kustomointien toteuttamisen osalta, varsinkin mikäli uusi versiopäivitys näkyy horisontissa jo muutaman vuoden sisällä. Ongelman luo kuitenkin se, että tekemättömän kustomoinnin tuomat edut voisivat muuten olla taloudellisesti kannattavia, ja siten yritys menettää niiden tuoman potentiaalisen tuoton.

**SharePoint Online**, eli Office 365–tuotepakettiin kuuluva, pilvessä toimiva SharePoint, on vahvasti integroitu Microsoftin Office 365-tuotepaketin eri tuotteiden kanssa. Tällaisia tuotteita ovat massalle tutut sisällöntuottoon ja esittämiseen suunnatut tuotteet, kuten Excel, Word ja PowerPoint, mutta myös hieman tuntemattomammat tuotteet kuten yrityksen sisäiseksi mediaksi luokiteltu Yammer, ja muistiinpanovälineenä käytettävä OneNote. SharePoint Online sivustot voidaan yleistää ns. perussivustoiksi. SharePoint Online sivustojen kustomointimahdollisuudet ovat melko rajatut, sillä ne sijaitsevat fyysisesti Microsoftin omilla servereillä, eikä yrityksellä siten ole mahdollisuutta tehdä kustomointeja serveritasolla. Tämän vuoksi SharePoint Online sivustot ovat myös käytännössä aina saatavilla - katkoitta. Microsoft lupaakin yleisessä palvelulupauksessaan 99,9 % saatavuuden SharePoint Online sivustoille (Jha 2013). SharePoint Online sivustoja voidaan konfiguroida asetusmuutoksin, CSOM-koodilla (engl. Client Side Object Model), joka sisältää esimerkiksi yleisesti käytetyt, tyylimuutokset mahdollistavan CSS-koodin, tai pientä automaatiota mahdollistavan JavaScript-koodin.

Kirjoitushetkellä useat suuret yritykset eivät ole vielä uskaltaneet, taikka edes pystyneet, siirtymään täysin SharePoint Online–versioon. Syynä tähän on se, ettei SharePoint Online tarjoa tietoturvaluokitukseltaan korkeimpien dokumenttien vaatimaa suojaustasoa. Yritykset minimoivat riskejä pitämällä tällaisen dokumentaation vielä tiukasti yrityksen oman verkon sisällä, SharePoint Onlinen kaltaisen pilvipalvelun sijaan. Toinen syy on myös se, että SharePoint Onlinessa ja sen kustomoinnissa on paljon rajoituksia. On tärkeää huomioida, että

suurempien kustomointien tarve SharePointissa ei ole katoamassa mihinkään, joten voidaan todeta, että SharePoint Online kaikkine hyötyineenkään, ei vastaa suurempien yritysten kaikkia SharePoint-tarpeita sellaisenaan. Kun huomioidaan sekä SharePoint Onlinen rajoitteet, sekä SharePoint Server sovelluksien kustomointien aiheuttamat rajoitteet versiopäivityksien osalta, on helppo ymmärtää, minkä vuoksi Microsoft on tehnyt aloitteen uudenmallisten sovelluksien mahdollistamiseksi ns. hybridisovellusten muodossa.

#### 4.4 SharePoint Hybrid -sovellusmalli

SharePoint Hybrid Applications on nimitys uudenmallisista sovelluksista SharePointin kentässä (jatkossa hybridisovellukset). Hybridisovellukset voidaan jakaa kahteen jo vakiintuneeseen termiin erityyppisistä SharePoint hybridisovellusmalleista: SharePoint-hosted app ja Cloud-hosted app, joista jälkimmäinen voidaan jakaa edelleen kahteen teknisesti hieman erityyppiseen: provider-hosted Appeihin ja autohosted appeihin (Johnson 2013). Hybridinimitys kuvaa hyvin uutta sovellusmallia kokonaisuutena, sillä se käyttää kahden aikaisemmin käytössä olleen sovellusmallin parhaita puolia: SharePoint Serverin kustomointimahdollisuuksia ja SharePoint Onlinen jatkuvaa saatavuutta ja saumatonta kehitystä.

SharePoint-hosted app-mallissa kaikki bisneslogiikka sijoitetaan JavaScripteihin, jotka ajetaan sivukohtaisesti. Mallissa voidaan käyttää myös SharePointin objektimallin JavaScript-versiota (JSOM), joka mahdollistaa SharePoint datan CRUD-operaatiot: Create, Read, Update, ja Delete. (Microsoft 2017a)

SharePoint-hosted app-mallia on vaikea nähdä hybridisovellusmallina, sillä se ei käytä mitään SharePointista eroavaa ulkoista komponenttia bisneslogiikan ajamiseen. Täten suljemme tämän tutkimuksen osalta SharePoint-hosted app-mallin hybridisovelluksien joukosta.



Provider-hosted app:n ja Autohosted app:n ero on yksinkertainen: ensin mainitussa bisneslogiikan sijoituspaikka määritellään ja luodaan itse, ja jälkimmäisessä sovelluksen sisään on rakennettu logiikka, joka luo automaattisesti tarvittavat bisneslogiikan komponentit. Yleisesti ottaen nämä bisneslogiikan komponentit luodaan aina pilvipalveluun. Periaatteessa ero nähdään myös siinä, mistä sovellus ladataan: Provider-hosted app-mallin sovellukset ovat yleensä organisaation sisäisiä ja autohosted app-mallin sovellukset löytyvät yleensä julkisesta sovelluskaupasta. Tämän tutkimuksen tarkoitus on perehtyä kohdeyrityksen hybridisovelluksiin, joten suljemme pois autohosted app-mallin sovellukset, sillä niiden käyttäminen kohdeyrityksen liiketoimintaprosesseissa on hyvin epätodennäköistä.

Hybridisovelluksissa, toisin kuin SharePoint Server-sovelluksissa, unohdetaan täysin ns. full trust-ratkaisut, jotka toteutetaan serverille asennettavina asennuspaketteina. Hybridisovellus voidaan asentaa pilveen, omalle serverille, tai jopa SharePoint Serverille, joka on samassa farmissa kuin yrityksen SharePoint On Premise, jolloin se ei kuitenkaan vaikuta koko serverin toimintaan (Johnson 2013). Tällä tarkoitetaan sitä, että itse toiminnallisuus, tai bisneslogiikka, voidaan suorittaa täysin SharePointin ulkopuolella ja siitä riippumattomasti. Se onkin hybridisovellusten tärkein ominaisuus, koska sillä voidaan varmistaa, että yksikään hybridisovellus ei vaikuta 1) SharePoint-alustaan, ja siitä johdettuna 2) toisiin sovelluksiin (Juvonen 2013). Täten hybridisovellukset mahdollistavat ketterän kehityksen ja ketterän muutosprosessin, koska muutokset ja niiden vaikutukset ovat sidottuna vain käsitteillä olevaan sovellukseen. Toisaalta vältytään myös katkoista johtuvilta negatiivisilta bisnesimplikaatioilta, kuten aiemmassa kappaleessa SharePoint Server sovellusten osalta kirjoitettiin. Koska hybridisovellusten bisneslogiikan koodi ajetaan SharePointin ulkopuolella, sovelluksien kehittäjän ei tarvitse välttämättä ymmärtää SharePointia lainkaan, joskaan se ei päde kaikissa tapauksissa, eikä tätä kannata yleistää.

Hybridisovellukset voivat näkyä SharePointissa kolmella tavalla. Ne voivat näkyä koko sivun sovelluksena (engl. Full Page), sivun app-osana (engl. App Part), tai käyttöliittymän kustomoituina toimintoina (engl. User Interface Custom Actions). Koko sivun sovelluksen ainoana vaatimuksena SharePointin puolelta on se, että sivuston ylälaidassa tulee olla 'Takaisin SharePointiin' tyyppinen toiminto. App-osa on SharePointin sivulla oleva iFrame-objekti, joka peilaa sovelluksen osaa SharePointiin. iFrame on web-tekniikka, jonka avulla sivun sisään saadaan tuotua toinen sivu – samasta tai toisesta järjestelmästä. Käyttöliittymän kustomoidut toiminnot ovat sivustolle lisättäviä toiminnallisuuspainikkeita, joilla voidaan käynnistää sovellukseen tehtyjä kustomoituja toiminnallisuuksia.

Hybridisovelluksien SharePoint-käyttöliittymässä käytettävien iFrame-objektien negatiivinen puoli on niiden hitaus – ne ovat oletukseltaan hitaampia kuin esimerkiksi Script Editor Web Partit, joilla voidaan tuoda SharePoint-sivustolle CSOM-koodia, kuten JavaScriptiä. iFrame voi ladata jopa muutaman sekunnin, kun taas Script Editor saattaa viedä esimerkiksi 100 millisekuntia. Syy tähän on siinä, että iFrame-objekti joutuu tarkastamaan käyttöoikeuspyynnön, sekä lataamaan SharePointin datan, JavaScript-kirjastot jne. Toisaalta iFrame-objekti on täten myös turvallisempi kuin Script Editor Web Part, johtuen siitä, että sivun muut kontrollit eivät pääse siihen käsiksi. Se on myös turvallisempi loppukäyttäjälle – iFramen kontroleilla ei ole pääsyä käyttäjän Office 365 profiiliin. (Microsoft 2017b)

Nimitys app-mallista voitaneen johtaa siitä, kuinka toiminnallisuudet otetaan SharePointissa käyttöön. Sovellukset otetaan käyttöön kuten mikä tahansa mobiiliapplikaatio useimmiten otettaisiin käyttöön puhelimessa, eli ne ladataan sovelluskaupasta. Koska suurin osa yritysten sovelluksista ovat sellaisia, joita yritys haluaa käyttää vain itse, eikä julkaista niitä yleisesti käytettäväksi, on Microsoft kuitenkin jakanut sovelluskauppa-metodin ns. kahteen osaan: organisaation sisäiseen ja julkiseen. Tämä käyttöönototapa mahdollistaa sovellusten helpon uudelleenkäytettävyyden. Mikäli sovellus on luonteeltaan yleispätevä, ja se on hyvin toteutettu, sovelluksen kehittäjän ei tule välttämättä edes ymmärtää missä sivustoilla se on

käytössä. Koska hybridisovelluksia voidaan käyttää myös SharePoint Server-sivustoissa, ratkaisee se osittain ongelman tiedostojen tietoturvaluokittelusta. Hybridisovellus ei sinänsä ratkaise ongelmaa sivuston serverin ylläpidosta aiheutuvista kustannuksista, joskin vähentää niitä, koska hybridisovellusten laadunvarmistustehtävistä johtuvat kustannukset saadaan teoriassa kutakuinkin nollattua.

Hybridisovellusten ja SharePointin välinen kommunikaatio tapahtuu REST API- ja CSOM-tekniologioiden avulla (Johnson 2013). Lisäksi sovellukset käyttävät tarvittaessa OAuth-tekniologiaa saadakseen oikeudet SharePoint-sivuston osiin (Johnson 2013). Itse sovelluksen ja käyttöliittymän välinen yhteys määritellään kuitenkin XML-tiedostoon, jossa määritellään myös mm. perusasetukset (basic properties), missä sovellusta ajetaan, mitä SharePointin tulee tehdä, kun sovellus käynnistetään, mitä kieliä sovellus tukee, mitä SharePoint-palveluita sovellus käyttää, sekä mitkä oikeudet sovellus tarvitsee Host Webiin (Microsoft 2017a).

Provider-hosted app-mallin sovelluskehityksessä on käytettävissä käytännössä rajaton määrä teknologioiden kirjo. Ainut rajoittava tekijä sovelluskehitykselle on pilvipalvelualan tukemat teknologiat. Microsoft ei ole rajannut oman tuotteen (SharePoint) kustomointeja vain heidän tarjoamaan pilvipalveluunsa (Azure) ja sen tarjoamiin sovelluskehityksellisiin viitekehyyksiin, kuten .NET viitekehyyseen. (Microsoft 2017a)

#### 4.5 Kohdeyrityksen tietohallintorakenne relevantein osin

Kohdeyrityksen tietohallintorakenne Sharepointiin keskittyvin osin voidaan jakaa kahdeksaan erityyppiseen kohderyhmään. Kohdeyritys on suuri kansainvälinen konserniyritys, joka toimii useissa kymmenissä maissa. Tässä tutkimuksessa keskitytään maatason sovellushallintaan, mutta prosesseissa käytetään hyväksi myös globaalin tietohallintomallin rakennetta ja sen tuomia synergiaetuja.

Sovelluksen **loppukäyttäjä** (engl. End User) on henkilö, joka käyttää sovellusta osana työtehtäviään. Loppukäyttäjä voi työssään olla täysin riippuvainen sovelluksen toiminnasta, mutta toisaalta hän saattaa käyttää sovellusta vain satunnaisestikin. Loppukäyttäjä raportoi mahdollisia sovellukseen liittyviä ongelmia tai muutostarpeita sovellusomistajalle. Loppukäyttäjä kuuluu useimmiten kohdeyritykseen, mutta toisaalta hän voi olla myös yrityksen ulkopuolinen sovelluksen käyttäjä.

**Sovellusomistaja** (engl. Application Owner) on henkilö, joka toimii kohdeyrityksessä jossakin liiketoimintayksikössä, ja joka on yhden tai useamman sovelluksen nimetty omistaja. Sovellusomistajan vastuulla on sovelluksen elinkaaren hallinta. Sovellusomistaja vastaa liiketoimintayksiköllensä ja on sovelluksen loppukäyttäjien ensisijainen yhteyshenkilö. Sovellusomistaja on vastuussa sovelluksen muutostarpeista, niiden suunnittelusta, ja muutosten realisoituessa vastaa muutoksen toteuttajan kilpailutuksesta ja muutosprojektin hallinnasta. Sovellusomistaja on myös toteutettujen muutosten ensisijainen hyväksyntätestaaaja. Sovellusomistaja on vastuussa sovelluksen yleisluontoisista ylläpitotoimenpiteistä, kuten sovelluksen käyttöoikeushallinnasta.

Muutokset sovellukseen tekee useimmiten ulkoinen **toimittaja** (engl. Vendor). Kohdeyritys käyttää useita eri toimittajia. Toimittaja on ensisijaisesti vastuussa sovitun sovelluksen teknisestä suunnittelusta ja toteutuksesta. Uuden sovelluksen kehityksen yhteydessä toimittajan vastuulla on luoda sovelluksen arkkitehtuurikuvaus, tekninen kuvaus, sekä itse sovellus. Muutosten yhteydessä toimittajan vastuulla on tehdä sovitut muutokset sovellukseen omassa kehitysympäristössään. Mikäli sovellus on ns. kustomoitu sovellus, on toimittaja vastuussa myös asennettavan sovelluspaketin luomisesta, sen lataamisesta sovittuun kohdeyrityksen hallitsemaan tietojärjestelmään, sekä lähdekoodin viemisestä kohdeyrityksen lähdekoodivarastoon, joka sijaitsee pilvipalvelussa, Microsoftin Team Foundation Serverillä, mutta on täysin kohdeyrityksen hallitsema.

Tutkija toimi tutkimuksen aikana kohdeyrityksessä roolissa **SharePoint muutoskoordinaattori**. Tehtävän toimiin kuuluu ennen kaikkea kustomoitujen, eli SharePoint On Premise–sovelluksien muutoshallinta. Muutoskoordinaattorin tehtävä on hallita uusien sovelluksien implementointiprosessia, sekä sovellusmuutosten yhteydessä muutosprosessin kokonaisuutta. Sovellusmuutoksen aikana muutoskoordinaattori vastaanottaa tiedon tulevasta muutoksesta, pyytää tarkemmat tiedot asennuspaketin sekä lähdekoodin sijainneista toimittajalta tai sovellusomistajalta, ja pyytää asennuspaketille koodikatselmuksen, testiympäristöasennuksen ja tuotantoympäristöasennuksen SharePoint Operations Team:lta. Muutoskoordinaattorin tärkein tehtävä on toimia siltana kaikkien implementointi- ja muutosprosesseihin kuuluvien kohderyhmien välillä, ja ajaa muutoksia eteenpäin saumattomasti.

SharePoint muutoskoordinaattorin esimiehenä toimii **SharePoint sovelluspäällikkö**, joka on vastuussa SharePoint-palvelusta maatasolla. SharePoint sovelluspäällikkö vastaa täten myös tutkimuksen tavoitteena olevan prosessin vaatimuksista ja on oleellisena osana myös prosessien arviointivaiheessa.

**SharePoint Operations Team** on vastuussa kohdeyrityksen SharePoint-ympäristöistä. Yksikkö on vastuussa sekä ympäristöjen asennuksista, hallinnasta ja korjauksista, mutta myös ympäristössä sijaitsevien sovelluksien muutosten laadunvarmistuksesta, eli arkkitehtuurikatselmuksista, koodikatselmuksista, testiympäristöasennuksista ja tuotantoasennuksista. Yksikkö on vastuussa SharePoint testi- ja tuotantoympäristöistä globaalisti koko yritykselle. Kyseisen yksikön osaaminen korostuu SharePoint-ympäristöjen ylläpitotehtävissä, mutta yksikön jäsenillä on myös valtavasti kokemusta SharePoint-kehityksestä. Yksikköön kuuluu myös SharePoint-arkkitehdit, jotka ovat vastuussa ennen kaikkea sovelluksien arkkitehtuuri- ja koodikatselmuksista. SharePoint Operations Team voidaan mieltää yrityksen sisäiseksi PaaS-tarjoajaksi.

**Cloud Hosting Team** on vastuussa kohdeyrityksen pilvipalveluiden tarjonnasta liiketoimintayksiköille globaalisti konsernitasolla. Yksiköllä on omat prosessinsa, jotka liittyvät olennaisesti pilvipalveluiden käyttöönottoon. Käyttöönottoaiheiksi voidaan laskea suunnitellun sovelluksen tarkastelu pilvipalvelumallin soveltuvuudesta, jossa sovelluksen sopivuus pilvipalvelusovellukseksi tarkastellaan turvallisuuden ja sopivuuden kannalta, sekä pilvipalvelualustojen hankkimisesta liiketoimintayksiköiden tarpeisiin. Cloud Hosting Team voidaan mieltää yrityksen sisäiseksi pilvipalveluiden IaaS-tarjoajaksi.

**AMS-palvelu** (engl. Application Maintenance Service) on yrityksen elin, joka on ottanut vastuulleen määriteltyjen kohdeyrityksen maaton SharePoint-sovellusten ylläpidon. Ylläpidon pääasiallinen tehtävä on tuotantokäytössä olevien sovellusten ongelmatilanteiden selvittäminen ja korjaaminen, mutta AMS-palvelu tekee myös pieniä lisäominaisuuksia ja muutoksia sovelluksiin, vaikka ne eivät nimellisesti olisikaan rikki. Toistaiseksi palvelu kattaa pääasiassa vain kustomoidut SharePoint OnPremise-sovellukset, joskin myös jokunen SharePoint Online-sovellus on otettu mukaan palvelun piiriin niiden liiketoiminnallisen merkittävyyden vuoksi. AMS-palvelu on myös vastuussa kohdeyrityksen maaton SharePoint kehitysympäristön hallinnasta. Suunniteltavien hybridisovellusten implementointi- ja muutosprosessien osalta AMS-kumppani voidaan mieltää toimittajaksi.

#### 4.6 SharePoint Server-sovellukset kohdeyrityksessä

SharePoint Server-ratkaisuja on käytetty kohdeyrityksessä SharePoint Server 2007 versiosta lähtien. Yritys hankki maatasolla omat serverit, jolle SharePoint-alusta asennettiin. SharePoint-servereitä kohdeyrityksessä ylläpitää SharePoint Operations Team, joka on vastuussa servereistä, niiden turvallisuudesta, SharePoint-alustoista ja niiden ylläpidosta, sekä tarvittavien päivitysten ja paikkausten implementoinnista. Kirjoitushetkellä kohdeyrityksessä on Server-mallisista SharePoint-alustoista käytössä SharePoint Server 2010 ja SharePoint Server 2013 versiot. Lisäksi SharePoint Server 2016 versio on

käyttönottovaiheessa. Serverit ovat sittemmin siirretty Microsoft Azure-pilvipalveluun käyttäen IaaS-mallia, mutta näillä servereillä hostattuja ratkaisuja pidetään edelleen täysin On Premise-mallisina, sillä SharePoint-alustoja ylläpidetään edelleen kohdeorganisaation edellä mainitun kohderyhmän toimesta. SharePoint Operations Team hankkii Cloud Hosting Teamilta pilvipalveluserverit IaaS-palveluna ja tarjoaa ne globaalisti kohdeyrityksen maatasojen tietohallinnoille PaaS-palveluina, eli valmiina SharePoint-alustoina.

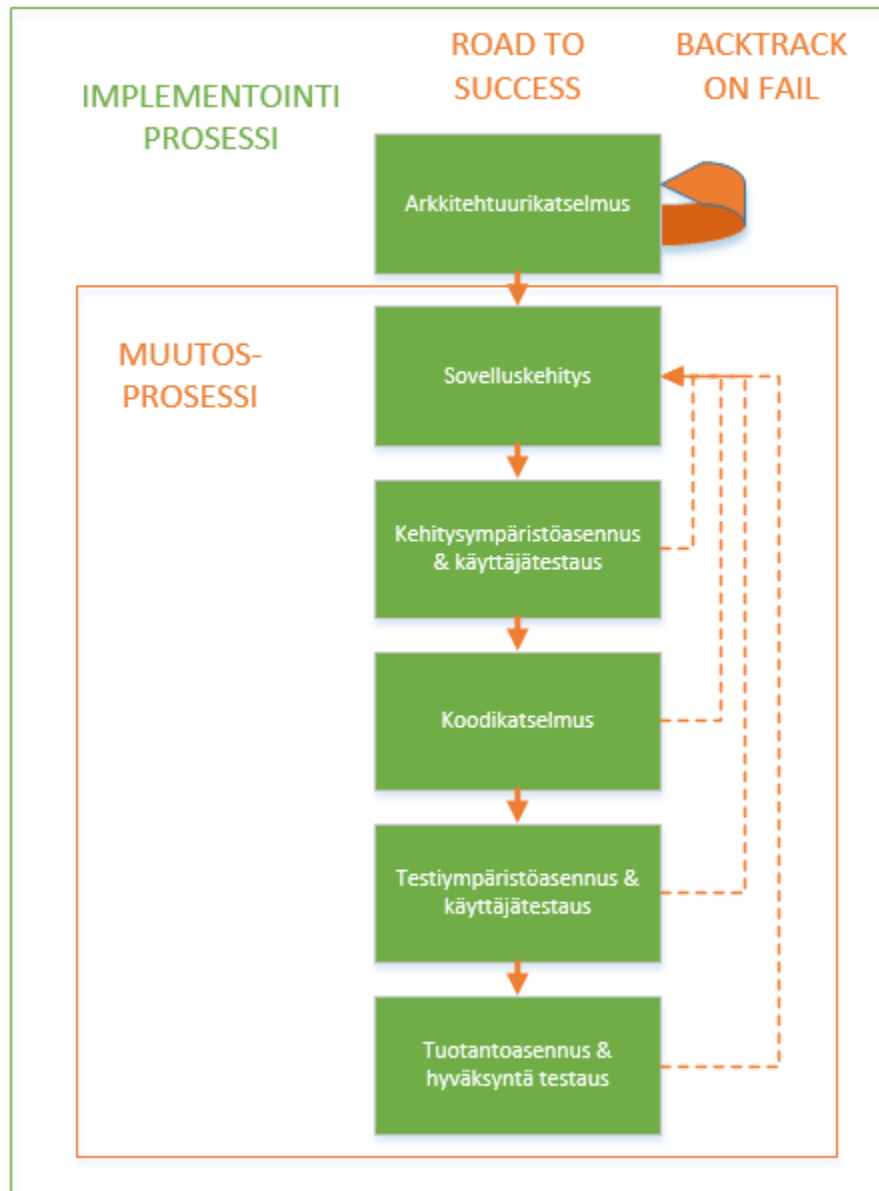
Kohdeyrityksessä SharePoint Server-ratkaisuja on luotu sellaisissa käyttötarkoituksissa, joissa SharePoint Online-ratkaisut eivät ole kattaneet kaikkia sovelluksien tarpeita. Lisäksi yrityksessä on joitakin ratkaisuja, jotka voisivat nykyään toteuttaa myös SharePoint Online-ratkaisuina, mutta jotka kehityshetkellä pystyttiin tekemään ainoastaan Server-mallisena, ja jotka ovat jääneet kyseiseen malliin kustannussyistä. SharePoint Onlinen rajoituksia ovat mm. tiedon turvallisuusluokitus tai kustomointien rajoitteet, joita SharePoint Server-ratkaisuissa on huomattavasti vähemmän. Kohdeyrityksen SharePoint Server-ratkaisuissa on useimmiten integraatioita organisaation muihin tietohallintojärjestelmiin, kuten ERP-järjestelmään. Tällaisia integraatioita ei SharePoint Online sellaisenaan tue. Integraatioiden, eli ns. siltojen SharePoint-ratkaisun ja sillan toisessa päässä olevan sovelluksen ylläpito toteutetaan yhteistyössä integrointikohteen sovelluksen tai järjestelmän ylläpitäjän kanssa. Toisinaanottuna esim. ERP-järjestelmän ylläpito ei kuulu SharePoint-ylläpidon piiriin, vaikka SharePoint-sovellus olisi siihen integroitu. SharePointin ja toisen järjestelmän välinen integraatio on molempien osapuolien yhteisvastuulla. Integraatioita on myös muihin kolmannen osapuolen tarjoamiin komponentteihin, joiden käyttötarkoitukset vaihtelevat merkittävästi. Näiden komponenttien ylläpito on usein myös sen tuottaneen toimittajan vastuulla, joskaan se ei ole ikinä riittävää, sillä ne pyörivät pääasiassa kohdeyrityksen servereillä, jonne kolmannen osapuolen toimittajalla ei ole pääsyä.

SharePoint Server-ratkaisuja voidaan kehittää Microsoftin .NET-viitekehityksen tukemilla teknologioilla. Voidaan määritellä, että .NET-viitekehitys teknologiaobjekti ja siihen kuuluvat

koodikielien ovat sen aliobjekteja. Koska tämä viitekehys on jokseenkin rajallinen, ja koska kaikki tarvittavat tietotaidot voidaan kiteyttää koodikielten sijasta tämän teknologiaviitekehysten tietotaitovaatimuksiin, ei alkuperäisen suunnitelman mukainen tarkemman sovellusanalyysin tekeminen yrityksen nykyisille SharePoint Server-ratkaisuille ole tarpeen.

Kohdeyrityksellä on vakiintuneet implementointi- ja muutosprosessit SharePoint Server-sovelluksille. Hybridimallisten sovellusten tukemiseksi luotavat prosessit tullaan luomaan näiden prosessien pohjalta. Tämä on looginen lähtökohta, sillä kohdeyrityksen kohderyhmät tuntevat SharePoint Server-sovellusten prosessien eri vaiheet, erityisesti laadunvarmistuksen osalta. SharePoint Server-sovellusten implementointi- ja muutosprosessit ovat hyvin toistensa kaltaisia. Muutosprosessi on käytännössä uuden sovelluksen implementointiprosessin osa. Kuvassa 14 on esitetty implementointi- ja muutosprosessien vaiheet. Mikäli jokin vaihe sovelluskehityksen jälkeen epäonnistuu, palataan aina sovelluskehitysvaiheeseen, jossa tehdään tarvittavat korjaukset ja prosessi aloitetaan siitä kohdin uudelleen.





**Kuva 14.** SharePoint Server-sovelluksien implementointi- ja muutosprosessit laadunvarmistuksen näkökulmasta.

**Arkkitehtuurikatselmuksen** vaatimuksena on sovelluksen arkkitehtuurikuvaus. Toimittaja on vastuussa kuvauksen toteuttamisesta, sovellusomistaja sen eteenpäin toimittamisesta muutoskoordinaattorille, joka taas on vastuussa katselmuksen pyytamisestä SharePoint Operations Teamin arkkitehdiltä kohdeyrityksen muutoksenhallintatyökalussa. Arkkitehti

tarkastelee arkkitehtuurikuvauksen ennen kaikkea laadunvarmistuksen näkökulmasta, mutta tarkastelee myös sovelluksen ehdotetun arkkitehtuurin optimaalisuutta. Jokaisen arkkitehtuurikatselmuksen määritelty läpimenoaika on viisi työpäivää. Mikäli arkkitehtuurikatselmuks on suoritettu hyväksytysti, annetaan toimittajalle lupa aloittaa **sovelluksen kehitys**. Sovelluskehitykselle ei ole määritelty läpimenoaikoja, sillä jokainen sovellus on yksilöllinen, ja jokaisen kehityksestä ja läpimenoajasta, sovitaan toimittajan kanssa erikseen.

Kun sovelluskehitys on toimittajan ja sovellusomistajan näkökulmasta valmis, pyydetään AMS-palvelulta **kehitysympäristöasennus** (engl. Development Environment Installation). Kun asennus on suoritettu onnistuneesti kehitysympäristöön, sovellusomistaja, tai hänen nimeämänsä ryhmä, hoitaa sovelluksen testauksen kyseisessä ympäristössä. Tämän testauksen tavoitteena on antaa hyväksyntä sovelluksen toiminnalliselle valmiudelle SharePoint-toteutuksen osalta. Kehitysympäristö on kuitenkin kahdesta muusta ympäristöstä (testausympäristö, tuotantoympäristö) poikkeava, sillä sieltä ei ole tarkoitus luoda integraatiota muihin kohdeyrityksen järjestelmiin. Jokaisen kehitysympäristöasennuksen läpimenoajaksi on määritelty kaksi työpäivää. Kehitysympäristöasennukset tapahtuvat melko ketterästi, jonka vuoksi kehitysympäristöasennuksia tehdään yleensä useita ennen kuin sovellus on täysin valmis.

Jos käyttäjättestaus kehitysympäristössä on suoritettu hyväksytysti, oletetaan että sovellus on valmis ja siirrytään implementointiprosessissa läpivientivaiheeseen. Läpivientivaiheen ensimmäinen osuus on **koodikatselmuks**, jonka suorittamiseksi vaaditaan asennuspaketti, asennusohjeet. Lisäksi lähdekoodi viedään toimittajan toimesta kohdeyrityksen määrittelemään ja hallitsemaan lähdekoodivarastoon, jonne myös SharePoint Operations Teamilla on pääsy. Muutoskoordinaattori avaa koodikatselmukspyynnön kohdeyrityksen muutoksenhallintatyökalussa ja koodikatselmuks suorittaa SharePoint Operations Teamin arkkitehti. Koodikatselmuksessa tarkastellaan mm. koodauksen ohjeiston noudattamista ja

koodin laatua, mutta ennen kaikkea koodin mahdollisia haavoittuvuuksia ja vaikutuksia kohdeympäristöön. Kohdeympäristössä on useita kymmeniä muitakin sovelluksia, ja mikäli sovellus aiheuttaa ympäristölle ylimääräistä kuormaa, vaikuttaa se siten potentiaalisesti kaikkiin muihinkin sovelluksiin negatiivisesti.

Mikäli koodikatselmus on suoritettu hyväksytysti, pyytää muutoskoordinaattori **testiympäristöasennuksen**. Asennus pyydetään suoraan kohdeyrityksen muutoshallintatyökalussa. Asennuspyynnön liitteeksi vaaditaan linkki hyväksytyyn koodikatselmukseen, joka sisältää kaikki tarvittavat tiedot asennuksen suorittamiseksi. Asennuksen suorittaa SharePoint Operations Teamin administraattori, joka asentaa sovelluksen testiympäristöön sovittuna ajankohtana. Asennuksen jälkeen muutoskoordinaattori informoi sovellusomistajaa onnistuneesta asennuksesta ja pyytää häntä suorittamaan **käyttäjätestauksen**. Testiympäristössä testataan myös mahdolliset integraatiotoiminnallisuudet ja niiden toimivuus.

Mikäli käyttäjätestaus testiympäristössä on suoritettu onnistuneesti, pyytää sovellusomistaja **tuotantoympäristöasennuksen** muutoskoordinaattorilta. Muutoskoordinaattori avaa asennuspyynnön kohdeyrityksen muutoshallintatyökalussa. Asennuspyynnön liitteeksi tarvitaan sama linkki koodikatselmukseen, kuin mitä testiympäristöasennuksessa on käytetty. SharePoint Operations Teamin administraattori tekee asennuksen sovittuna ajankohtana. Muutoskoordinaattori informoi sovellusomistajaa asennuksen jälkeen ja pyytää häntä suorittamaan **hyväksyntätestauksen**. Mikäli hyväksyntätestaus on suoritettu onnistuneesti implementointiprosessissa, voidaan sovellus ottaa tuotantokäyttöön. Mikäli muutosprosessin jälkeinen hyväksyntätestaus on epäonnistunut, pyydetään ns. takaisinrullaus SharePoint Operations Teamilta. Takaisinrullauksessa sovellus asetetaan siihen tilaan, jossa se oli ennen viimeisintä tuotantoasennusta.

Laadunvarmistuksen vaihe	Uuden SharePoint Server-sovelluksen implementointiprosessissa (arkipäivää)	SharePoint Server-sovelluksen muutosprosessissa (arkipäivää)
Arkkitehtuurikatselmus	5	N/A
Sovelluskehitys	N/A	N/A
Kehitysympäristöasennus	2	2
<i>Käyttäjättestaus</i>	N/A	N/A
Koodikatselmus	5	5
Testiympäristöasennus	5	5
<i>Käyttäjättestaus</i>	N/A	N/A
Tuotantoympäristöasennus	10	10
<i>Hyväksyntättestaus</i>	N/A	N/A
<b>Läpimenoaika yhteensä (arkipäivää)</b>	<b>27</b>	<b>22</b>

**Kuva 15.** SharePoint Server-sovellusten laadunvarmistusprosessien läpimenoajat.

Kuvassa 15 on esitetty SharePoint Server-toteutusten implementointi- ja muutosprosessien läpimenoajat laadunvarmistuksen näkökulmasta. Esitettyjen läpimenoaikojen edellytyksenä on se, että yhdestäkään vaiheesta ei löydetä ongelmia, joita joudutaan korjaamaan, ja jonka vuoksi prosessi joudutaan aloittamaan joko kokonaan tai osittain uudelleen. Todellisuudessa läpimenoajat ovat huomattavasti korkeammat, sillä on naiivia odottaa, että tietojärjestelmä olisi ongelmaton. Lisäksi on tärkeää huomioida, että nämä läpimenoajat eivät ota kantaa sovelluskehitykseen, tai käyttäjä-, sekä hyväksyntätestauksiin käytettävään aikaan, koska kyseisten muuttujien läpimenoajat vaihtelevat huomattavan paljon toisistaan.

#### 4.7 SharePoint Online–sovellukset kohdeyrityksessä

SharePoint Online–sivustoja voidaan kustomoida ainoastaan CSOM-koodilla (Client side object model), eli koodilla, jota ei ajeta serveritasolla. Tämän vuoksi yrityksessä ei seurata kyseisiä kustomointeja sisältäviä ratkaisuja, eikä niiden kustomoinneista pidetä rekisteriä. Syy on yksinkertainen, sivuston kustomoinnit eivät voi vaikuttaa muihin sivustoihin tai sovelluksiin, joten niiden tarkempi seurauksena olisi epäloogista, se toisi turhaa byrokratiaa, joka puolestaan tekisi sivustojen kehityksestä jäykempää. Täten kustomointien dokumentointi ja muutoksien tarkastelu laadunvarmistusnäkökulmasta on täysin sivuston omistajan vastuulla.

Yrityksessä on maatasolla muutama niin kompleksinen SharePoint Online-sivusto, että ne ovat sisällytetty yrityksen SharePoint AMS-palveluun, jonka vuoksi niiden kustomoinnit ovat dokumentoitu tarkalla tasolla. Näitä sovelluksia ei kuitenkaan saa sekoittaa hybridisovelluksiin, sillä niiden koodit ovat täysin CSOM-mallisia, eikä niissä ole toisessa järjestelmässä ajettavaa bisneslogiikkaa. SharePoint Online-sivustoja ei voi tehdä kustomoituja integraatioyhteyksiä toisiin järjestelmiin, joten niitäkään ei ole tarpeen huomioida. Tähän voidaan kuitenkin pitää poikkeuksena niitä Microsoftin tarjoamia ulkoisia järjestelmiä, jotka sisältyvät Office-pakettiin. Näiden integraatioiden vastuu on täysin Microsoftilla, eikä kohdeyrityksellä ole mahdollisuutta tai tarvetta niihin vaikuttaa.

#### 4.8 SharePoint Hybrid–sovellukset kohdeyrityksessä

Hybridisovellusten käyttöliittymänä toimii aina joko SharePoint Online, tai SharePoint OnPremise–sivusto. Tästä johtuen hybridisovellusten loppukäyttäjien käyttöoikeushallinta toimii täysin näiden ympäristöjen standardien mukaisesti, eli käytännössä SharePointin käyttöoikeushallintatyökaluja käyttämällä. Useimmiten käyttöoikeushallinta käyttää hyväkseen yrityksen AD-federaatiopalveluita. Koska nämä palvelut ovat aina käytettävään ympäristöön liittyviä, hallitsee niitä aina ympäristön admin, eli kohdeyrityksen tapauksessa

SharePoint Operations Team:n jäsen. Täten käyttöoikeusinfrastruktuurin hallinta ei ole hallittavissa maatasolla, eikä käyttöoikeusinfrastruktuurin hallinta kuulu tämän tutkimuksen piiriin. Toisaalta sivusto- tai sovelluskohtainen käyttöoikeushallinta kuuluu sovellusomistajan vastuisiin.

SharePoint Hybrid-sovelluksissa käytettävä teknologioiden kirjo on käytännössä rajaton. Tämän vuoksi tutkimuksessa päädyttiin tekemään yksi ylimääräinen haastattelu AMS-palvelutoimittajan SharePoint arkkitehdin kanssa, jotta saatiin kattava kuva teknologioista, niiden rajoitteista ja mahdollisuuksista nykyisissä ja tulevista sovelluksissa.

Haastattelusta ja sovellusanalyyseistä voidaan todeta, että kaikki nyt käytössä olevat, ja suunnitellut hybridisovellukset, käyttävät ainoastaan .NET-viitekehysten teknologioita. Kaikkien sovelluksien bisneslogiikka sijoitetaan Microsoft Azuren pilvipalveluun, josta ne ovat linkitettyinä REST-tekniikan avulla SharePoint Onlineen, joka toimii sovelluksien käyttöliittymänä. Osa sovelluksista on yhteydessä muihin järjestelmiin, ja niiden integraationa toimii Azureen rakennettu Web Service, jonka avulla saadaan haettua ja vietyä tietoa muihin ulkoisiin järjestelmiin, kuten ERP-järjestelmä SAP:iin. Lähes kaikki sovellukset ovat suuressa kaavassa arkkitehtuuriltaan identtisiä integraatioiden teknologioiden osalta. Kaikki Azuresta muihin kuin SharePoint Onlineen tehdyt integraatiot perustuvat Web Service –tekniikkaan.

Web Servicet ovatkin erityisen tärkeä osa sellaisia hybridisovelluksia, joista muodostetaan yhteys muihinkin järjestelmiin, kuin SharePoint Onlineen. Web Servicet toimivat siltana hybridisovelluksen ja ulkoisen järjestelmän välillä, ja ne vaativat vastakappaleen myös ulkoisen järjestelmän osalta. Voidaan yleistää, että Web Service vaatii toisen Web Servicen toimiakseen.

Voidaan myös todeta, että vaikka hybridisovelluksia voidaan rakentaa suuremmalla teknologioiden kirjolla, on suuren organisaation suositeltavaa pitää sovellukset yhden viitekehysten teknologioiden sisällä. Tällä tavoin sovellusten ylläpito vaatii vähemmän tietotaitoa ja siten resursseja. Se helpottaa myös AMS-kumppanina käytetyn toimittajan kilpailutusta, sillä se laajentaa kykenevien toimittajien kenttää. Lisäksi voidaan yleistää, että loogisin käytettävä viitekehys SharePoint hybridisovelluksille on Microsoftin .NET-viitekehys, sillä SharePoint itsessäänkin on Microsoftin tuote, ja voidaan olettaa, että .NET-viitekehys tulee aina tukemaan SharePoint kehitystä. On hyvä muistaa, että tämä logiikka on täysin teoreettinen, sillä nykyinen hybridisovellusmalli tukee muitakin viitekehysiä, ja koska sovellusten bisneslogiikkaa ei toteuteta SharePoint-servereille, eikä niitä tarvitse toteuttaa välttämättä edes Microsoftin pilvipalvelu Azureen.

Kohdeyrityksen tapauksessa päädyttiin .NET-viitekehysten teknologioihin, sillä se 1) kattaa kaikki yrityksen hybridisovellusmallin kehityksen tarpeet, ja koska 2) yrityksen hybridisovelluskentän kohderyhmällä on merkittävä osaaminen kyseisen viitekehysten teknologioissa.

#### 4.9 Uusien prosessien tuomat riskit kohdeyrityksessä

Uudet prosessit tuovat mukanaan riskejä. Muutosvastarinta koetaan lähes aina riskinä mille tahansa muutokselle yrityksen prosesseissa, mutta tutkimuksen tapauksessa riski on melko pieni. Syy sille on yksinkertainen: nykyiset prosessit eivät kata uusia hybridisovelluksia sellaisenaan millään tavalla, joten liiketoimintayksiköt ottavat uudet prosessit vastaan ilomielin, mikäli niiden odotetaan tuovat edes jotain hyötyä. Kuitenkin voidaan odottaa, että uudet prosessit tuovat esiin paljon huolenaiheita liiketoimintayksiköistä, koska prosessien täydellinen optimointi lähtötilanteessa on hyvin vaikeaa, ja siten sillä on vaikeaa vastata liiketoimintayksiköiden kaikkiin vaatimuksiin.

Toinen tunnistettu riski on se, että kaikkia hybridisovelluskehityksen teknologioita ei tässä tutkimuksessa onnistuta tunnistamaan. Todennäköisyys tälle tapahtumalle on hyvin korkea – teknologioiden määrä on lähes rajaton ja tutkimuksen käytössä oleva materiaali, eli nykyisten sovellusten analyysi ja kirjallisuuskatsaus eivät kata kaikkia. Yhtenä riskinä voidaan myös tunnistaa mahdollinen yrityksestä puuttuva tietotaito jonkin teknologian ylläpitämiseksi. Mikäli tämä riski realisoituu, jokin teknologia joudutaan joko pudottamaan prosessien ulkopuolelle, jonka vuoksi mahdollisesti myös koko sovellus, tai sen osa, joudutaan poistamaan yleisen sovellusten laadunvarmistusprosessin piiristä. Tätä riskiä on vaikea lieventää, sillä sen poistamiseen vaadittaisiin joko kyseisen tietotaidon hankkiminen, tai nykyisen kohderyhmien laajentamisen uudella toimittajalla, mikä ei ole taloudellisista syistä toivottavaa.



## 5 KOHDEYRITYKSEN PROSESSIEN MALLINNUS HAASTATTELUMENETELMIN

Tämän tutkimuksen tavoite on luoda sovellusylläpidolliset implementointi- ja muutosprosessit yhdelle määritellylle kohdeyritykselle, joten lukijan on tärkeää huomioida, ettei tässä kappaleessa esitettyä mallia voi välttämättä yleistää tai kopioida suoraan käytettäväksi toisaalla. Tutkimuksen tietoja voi kuitenkin hyödyntää vastaavien prosessien luomiseksi. Tuotetun tiedon avulla ja yrityksen tietohallintorakenteen, ja sen eri sidosryhmien tietotaidot ymmärtäessä, voi kukin luoda omat optimoidut prosessinsa.

### 5.1 Ensimmäisen haastattelukierroksen olettamien pohdintaa

Ensimmäinen ehdotus hybridisovellusten implementointi- ja muutosprosesseiksi perustuu tutkijan havaintoihin ja oletuksiin, jotka tutkija on saanut kirjallisuuskatsauksen ja SharePoint hybridisovellusmallin tutkimisen aikana. On tärkeää huomioida, että kyseinen ehdotus ei vielä sisällä kohdeyrityksen antamia rajoituksia, vaan ainoastaan kohdeyrityksessä toimivan tutkijan roolissa tehtyjä havaintoja, ja siten olettamia joistakin rajoituksista. Olettamien hyväksyttävyyden vahvistaminen, ja virheellisten oletuksien havainnoiminen, on ensimmäisen syklin haastatteluiden tärkein tavoite.

Kuten aiemmin todettu, hybridimallisten sovellusten kehittämiseksi ei ole teknologiaviitekehysellisiä rajoitteita. Sovelluksia voidaan toteuttaa jo nyt SharePoint Server-sovelluksissa käytetyillä Microsoftin .NET-viitekehysten teknologioilla, yhdessä Microsoft Azure pilvialustan kanssa, mutta niitä voidaan luoda myös useilla muilla eri viitekehysillä ja pilvialustoilla. Pilvialustojen osalta tutkija olettaa, että hybridisovellukset toteutetaan kohdeyrityksessä Microsoft Azure-pilvipalvelualustalle, koska se on jo yleisesti kohdeyrityksessä käytössä. Ensimmäiselle haastattelukierrokselle muodostuu kysymys siitä, kannattaako kohdeyrityksen hybridimallisten sovellusten laadunvarmistusprosesseihin

sisällyttää rajaton määrä teknologioita, jolloin sovellusmassan ylläpitäminen on vähintäänkin haastavaa. Tiedetään, että kohdeyrityksen sidosryhmät palvelu pystyvät tukemaan .NET-viitekehityksen teknologioita käyttäviä sovelluksia, sillä se tukee niitä jo SharePoint Server-sovelluksien osalta. Lisäksi tämä vahvistettiin myös ylimääräisellä alustavalla haastattelulla arkkitehdin kanssa. Voidaan olettaa, että kohdeyritys ei pystyisi nykyisellä kokoonpanollaan kuitenkaan kattamaan kaikkia hybridisovellusmallin tukemia teknologioita. Todennäköisesti kohdeyritys joutuisi varaamaan nykyistä enemmän resursseja eri teknologiaviitekehysten ylläpidon kattamiseksi, jolla olisi suoraan negatiivinen vaikutus ylläpitokustannuksiin. Toisaalta useammat teknologiaviitekehitykset saattavat mahdollistaa useampien sovellusten toteuttamisen hybridimallisina sovelluksina, jolla taas laadunvarmistusprosessikustannusten vähenemisen vuoksi olisi taloudellisesti positiivinen vaikutus.

Haastatteluissa tutkija yrittää myös varmistua jo aiemmin ylimääräisen haastattelun perusteella tehdystä havainnosta siitä, että .NET-viitekehityksen teknologiat kattavat kaikki kohdeyrityksen sovelluskehitykselliset tarpeet SharePointin kentässä. Kuvassa 16 on esiteltyä haastatteluiden sidosryhmät ja teemahaastatteluiden aihealueet.

Sidosryhmän sijainti organisaatiossa	Sidosryhmä / tehtävänimike	Tehtävä haastattelussa	Teemahaastattelun aiheet
Maatason IT-hallinto	SharePoint muutoskoordinaattori	Haastattelijä	N/A
Maatason IT-hallinto	SharePoint sovelluspäällikkö, IS Business Engagement Manager & SharePoint Service Manager	Haastateltava	Laadunvarmistusprosessi, liiketoiminnan vaatimukset
Maatason liiketoimintayksikkö	Sovellusomistaja, Application Owner	Haastateltava	Sovelluksien liiketoiminnalliset vaatimukset
Ulkoinen AMS-toimittaja	Toimittaja/sovelluskehitys & AMS-palvelun ylläpitotoimittaja, Senior SharePoint Consultant	Haastateltava	Sovelluskehitykselliset vaatimukset, viitekehitykselliset toiveet
Globaali IT-hallinto	SharePoint Operations Team Admin, Senior SharePoint Administrator	Haastateltava	Sovellusasennukset, viitekehitykselliset ohjeistukset
Globaali IT-hallinto	SharePoint Operations Team, Senior SharePoint Architect	Haastateltava	Arkkitehtuurikatselmus, viitekehitykselliset vaatimukset
Globaali IT-hallinto	Cloud Hosting Team Architect	Haastateltava	Pilvipalvelualustojen provisiointiprosessi

**Kuva 16.** Haastatteluiden sidosryhmät ja teemahaastattelun aihealueet.

Ensimmäinen ehdotusolettama on, että teknologiaviitekehyksiä ei suoranaisesti rajoiteta, jotta kohdeyrityksen liiketoimintayksiköt voivat kehittää sovelluksia haluamallaan tavalla, mutta maatason hybridisovellusten laadunvarmistusprosesseihin liitetään vain ne sovellukset, jotka noudattavat .NET-viitekehyksen teknologioita ja käyttävät Microsoft Azure-pilvipalvelualustaa. Tällöin liiketoimintayksiköt sopivat itsenäisesti muiden hybridisovellusten ylläpitotoimenpiteiden järjestelyistä. Tällä ei ole suoraa vaikutusta implementointi- ja muutosprosesseihin – ainoastaan potentiaaliin ylläpidollisiin haasteisiin,

jotka eri teknologioiden osalta saattaisi nousta esiin. Lisäksi tällaiset poikkeusjärjestelyt tulee sopia erikseen Cloud Hosting Teamin kanssa.

*Olettama 1: Prosesseihin sisällytetään vain ne sovellukset, jotka käyttävät .NET viitekehysten teknologioita ja Microsoft Azure –alustaa. Muita viitekehysten käyttävien sovelluksien ylläpito on liiketoimintayksikön vastuulla.*

Koska hybridisovellukset sijaitsevat omalla alustallaan, eivätkä vaikuta toisiin sovelluksiin, voidaan olettaa, että laadunvarmistusprosessit saadaan minimoitua sekä läpimenoajallisesta, että taloudellisesta näkökulmasta. Tutkija olettaa, että mikään SharePoint Server-sovellusmallin laadunvarmistusprosessin vaihe ei ole tarpeellinen sovelluskohtaisten pilvipalvelualueilla olevien sovelluksien osalta, koska vaikutusta muihin sovelluksiin ei ole.

*Olettama 2: Mikään nykyisistä SharePoint Server-sovellusmallin laadunvarmistusprosessien vaiheista ei ole tarpeellinen hybridisovellusten implementointi- tai muutosprosessien osalta.*

Tämän vuoksi tutkija myös olettaa, että AMS-palvelulle voidaan antaa täydet valtuudet hallita sovellusta pilvipalvelualueilla. AMS-palvelu voi tehdä korjaukset ja muutokset omassa ympäristössään, ja viedä hyväksytyt muutokset ns. tuotantoympäristöön ketterästi sovellusomistajan hyväksymänä aikana. Siten teoreettinen laadunvarmistuksellinen läpimenoaika muutosprosessin osalta on 0 arkipäivää. Tämä ei sisällä AMS-palvelun asennuksiin käytettävää aikaa, eikä sovellusomistajan käyttäjä- tai hyväksyntätestauksiin käytettävää aikaa. AMS-palvelu voi tehdä asennuksia ketterästi ilman palvelusopimukseen määriteltävää aikamäärettä.

*Olettama 3: AMS-palvelu saa täydet oikeudet sovelluskohtaisiin testi- ja tuotantoympäristöihin hybridisovellusten osalta. Määritelty teoreettinen laadunvarmistuksen läpimenoaika muutosprosessin osalta on täten 0 arkipäivää. Todellinen*

*laadunvarmistuksen läpimenoaika sisältää myös AMS-palvelun asennuksiin käytettävän ajan, sekä sovellusomistajan käyttäjä-, että hyväksyntätestaukset.*

Tutkija on tietoinen siitä, että kohdeyritys vaatii pilvialustalle asennettavalta uudelta sovellukselta ns. pilvipalveluvalmiustarkastelun (engl. Cloud Assessment). Tutkija ei ole tietoinen kyseisen tarkastelun prosessista, sen vaatimuksista tai läpimenoajasta. Ensimmäisen ehdotuksen oletuksena on, että tarkastelun läpimenoaika on kutakuinkin SharePoint Server-sovelluksien arkkitehtuurikatselmuksen kaltainen, eli viisi arkipäivää.

*Olettama 4: Pilvipalveluvalmiustarkastelun kesto, ja siten koko laadunvarmistusprosessin kesto uusien sovelluksien osalta, on viisi arkipäivää.*

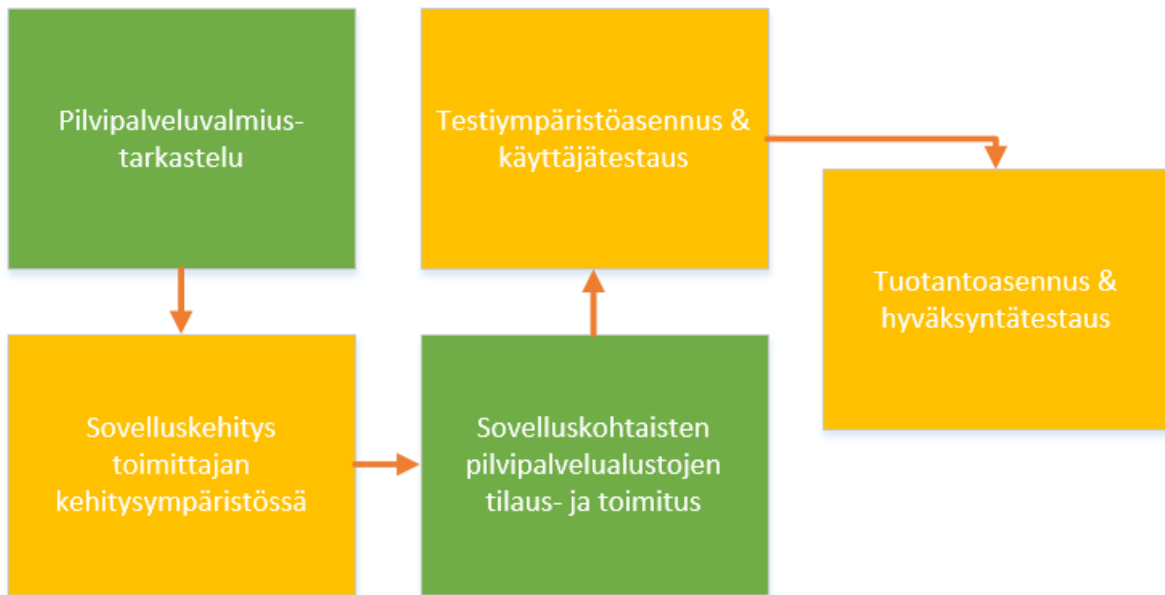
Kuvassa 17 vertaillaan SharePoint Server-sovellusten nykyisiä, ja hybridisovellusten osalta tutkijan oletamiin perustuvia laadunvarmistusprosessien läpimenoaikoja. Ero on merkittävä. Mikäli oletamat 2-4 vahvistetaan ensimmäisen haastattelukierroksella sellaisenaan, voidaan todeta, että hybridisovellusten muutosprosessin laadunvarmistuksellinen läpimenoaika (0 arkipäivää) mahdollistaa todellisesti sovellusten ketterän kehityksen. Myös implementointiprosessin läpimenoaika (viisi arkipäivää) on huomattavasti nykyistä SharePoint Server-sovellusmallia vastaavaa prosessia nopeampi (27 arkipäivää). Lisäksi, nämä luvut eivät ota huomioon epäonnistuneita laadunvarmistusprosessin läpikäyntejä, jolloin ketterämpi hybridisovellusmalli on verrattain vieläkin nopeampi implementointiprosessin osalta. Esimerkkinä voidaan antaa tilanne, jolloin uuden SharePoint Server-sovelluksen laadunvarmistuksellinen läpimeno-prosessi pysähtyy ensimmäisellä kerralla kehitysympäristöasennukseen, toisella koodikatselmuksen, ja kolmannella testiympäristöasennuksen jälkeiseen käyttäjätestaukseen. Tällöin kaikki edeltäneet vaiheet joudutaan toistamaan uudelleen, pois lukien arkkitehtuurikatselmuksen, jos oletetaan että sovelluksen arkkitehtuuri ei muutu korjausten seurauksena. Tällöin pelkästään laadunvarmistuksellisen prosessin läpimenoaika pidentyisi jopa 9 arkipäivällä. Tässä ei vielä

edes huomioida sovelluskehitykseen, eli itse korjauksiin, tai testaustoimenpiteisiin kulunutta aikaa, joka on tutkijan kokemuksen perusteella vähintään aiheutuneen laadunvarmistusprosessin läpimenoajan pidentymistä vastaava, eli esimerkkitapauksessa toiset 9 arkipäivää. Tämä tilanne, vaikkakin eri vaiheiden yhdistelmänä, realisoituu tutkijan kokemuksen mukaan hyvinkin usein, ja toimii siten hyvänä vertauskuvana.

Laadunvarmistuksen vaihe	Uuden SharePoint Server-sovelluksen implementointiprosessissa (arkipäivää)	SharePoint Server-sovelluksen muutosprosessissa (arkipäivää)	SharePoint Hybridisovelluksen implementointiprosessissa (arkipäivää)	SharePoint Hybridisovelluksen muutosprosessissa (arkipäivää)
Arkkitehtuurikatselmus	5	N/A	5	N/A
Sovelluskehitys	N/A	N/A	N/A	N/A
Kehitysympäristöasennus	2	2	N/A	N/A
<i>Käyttäjättestaus</i>	N/A	N/A	N/A	N/A
Koodikatselmus	5	5	N/A	N/A
Testiympäristöasennus	5	5	0	0
<i>Käyttäjättestaus</i>	N/A	N/A	N/A	N/A
Tuotantoympäristöasennus	10	10	0	0
<i>Hyväksyntättestaus</i>	N/A	N/A	N/A	N/A
<b>Läpimenoaika yhteensä (arkipäivää)</b>	<b>27</b>	<b>22</b>	<b>5</b>	<b>0</b>

**Kuva 17.** SharePoint Server- ja SharePoint hybridisovelluksien suunniteltujen laadunvarmistusprosessien läpimenoaikojen vertailua.

Kuvan 17 olettamilla SharePoint hybridimallisten sovellusten implementointi- ja muutosprosessit kohdeyrityksessä voidaan toteuttaa, kuten kuvan 18 mukaisessa ehdotuksessa on hahmotettu. Ensimmäisen vaiheen ehdotus on tehty hyvin korkealla tasolla, sillä ehdotus perustuu suureen määrään tutkijan olettamia, eikä tarkemman tason prosessikuvaus vastuurooleineen ja tarkennuksineen olisi ajankäytöllisesti kannattavaa.



**Kuva 18.** Karkea kuvaus ehdotuksesta hybridimallisten sovellusten implementointi- ja muutosprosessiksi.

*Olettama 5: Pilvipalveluvalmiustarkastelun suorittaa kohdeyrityksen kohderyhmä Cloud Hosting Team, joka on vastuussa myös sovelluskohtaisten pilvipalvelualustojen toimituksesta.*

Nykyisessä SharePoint Server-mallin sovelluksien implementointiprosessissa sovelluskehitys on tapahtunut toimittajan omassa kehitysympäristössä. Malli on todettu hyväksi, eikä mallin jatkamisessa nähdä mitään esteitä ja oletetaan, että samaa mallia kehityksen osalta käytetään myös hybridisovelluskehityksen osalta.

*Olettama 6: Sovelluskehitys tapahtuu toimittajan (mukaan lukien AMS-palvelu) omassa kehitysympäristössä. Sovellusta ei SharePoint Server-sovellusmallin tapaan ole tarpeen testata kohdeyrityksen kehitysympäristössä.*

Pilvipalvelualustojen tilaus on uusi ulottuvuus implementointiprosessiin. Koska alusta tilataan Cloud Hosting Teamilta ja se on osana implementointiprosessia, voitaneen pitää loogisena, että tilaus suoritetaan SharePoint muutoskoordinaattorin kautta.

*Olettama 7: Sovelluskohtaisten pilvipalvelualustojen tilaus tapahtuu maaston SharePoint muutoskoordinaattorin kautta.*

## 5.2 Ensimmäisen haastattelukierroksen tulokset ja korjatut oletamat

*Olettama 1: Prosesseihin sisällytetään vain ne sovellukset, jotka käyttävät .NET viitekehysten teknologioita ja Microsoft Azure –alustaa.*

Ensimmäisen syklin ehdotuksen olettamana oli se, että kaikki prosesseihin sisällytettävät sovellukset toteutetaan .NET viitekehysten teknologioilla ja Microsoft Azure-alustaa käyttäen. Tämä oletama johti kysymykseen ”voidaanko kaikki yrityksen sovelluskehitykselliset tarpeet täyttää näillä teknologioilla?”. Haastatteluiden avulla päästiin konsensukseen siitä, että .NET teknologiat yhdessä Microsoft Azuren kanssa kattavat kaikki tarpeet, mutta siihen ehdotettiin lisättäväksi NodeJS-teknologia, joka on tyyliltään hyvin .NET viitekehysten sisältämän JavaScriptin kaltainen. Pystyttiin myös toteamaan, että nykyinen AMS-palvelu pystyy tukemaan NodeJS-teknologiaa ilman muutoksia toimittajan kokoonpanoon ja siten palvelun hintaan.

*Korjattu oletama 1: Prosesseihin sisällytetään vain ne sovellukset, jotka käyttävät .NET viitekehysten teknologioita ja/tai NodeJS-teknologiaa, sekä Microsoft Azure –alustaa.*

*Olettama 2: Mikään nykyisistä SharePoint Server-sovellusmallin laadunvarmistusprosessien vaiheista ei ole tarpeellinen hybridisovellusten implementointi- tai muutosprosessien osalta.*



Haastattelujen perusteella päästiin siihen tulokseen, että nykyiset laadunvarmistusprosessien vaiheet ovat toistaiseksi tarpeellisia, mutta niitä voidaan nopeuttaa huomattavasti hybridisovellusten osalta. Yrityksen tavoitteena on kuitenkin siirtyä kohti ketterämpää mallia, jossa laadunvarmistusprosessin vaiheet voidaan poistaa. Syynä niiden nykyiselle tarpeellisuudelle ei ole itse laadunvarmistus, vaan muutoksenhallinnan työkalun käytännön vaatimukset. Työkaluun tulee tehdä merkittäviä muutoksia hybridisovellusten laadunvarmistusprosessien optimoimiseksi.

*Korjattu oletama 2: Nykyiset SharePoint Server-sovellusmallin laadunvarmistusprosessit tulevat osaksi myös hybridisovellusten implementointi- ja muutosprosesseja. Laadunvarmistusprosessien vaiheet ovat tarpeellisia ainoastaan muutoksenhallintatyökalun vaatimien vaiheiden vuoksi, ja vaihekohtaisia läpimenoaikoja voidaan optimoida ja lyhentää huomattavasti.*

*Olettama 3: AMS-palvelu saa täydet oikeudet sovelluskohtaisiin testi- ja tuotantoympäristöihin. Määritelty teoreettinen laadunvarmistuksen läpimenoaika muutosprosessin osalta on täten 0 arkipäivää. Todellinen laadunvarmistuksen läpimenoaika sisältää myös AMS-palvelun asennuksiin käytettävän ajan, sekä sovellusomistajan käyttäjä- ja hyväksyntätestaukset.*

Haastatteluiden perusteella ulkoistetulle AMS-palvelulle ei toistaiseksi voida tietoturvasyistä antaa täysiä hallintaoikeuksia testi- ja tuotantoympäristöihin, vaikkakin tätä toimintamallia yritetään viedä eteenpäin tulevaisuudessa kohdeyrityksessä. Toimintatapojen muutos suuressa kansainvälisessä yrityksessä on erityisen hidasta tämän tyyppisen aiheen ympärillä. Nykyiset SharePoint Server-sovelluksien laadunvarmistusprosessit ovat täten tarpeellisia, joskaan prosessin vaiheisiin käytettävä aika ei ole sidottu aiempaan toimintamalliin. Tämän vuoksi lopulliseen prosessiin jouduttiin lisäämään yksi (1) arkipäivä per

laadunvarmistuksellinen vaihe. Olettama 3 lisättiin kohdeyrityksen jatkokehityskohteiden listalle.

*Olettama 4: Pilvipalveluvalmiustarkastelun kesto, ja siten koko laadunvarmistusprosessin kesto uusien sovelluksien osalta, on 5 arkipäivää.*

Olettamaa 4 ei voitu vahvistaa ensimmäisen haastattelukierroksen perusteella. Pilvipalveluvalmiustarkastelulle ei olla annettu palvelukestomääritelmää. Olettamaan neljä palataan toisen haastattelukierroksen yhteydessä.

*Olettama 5: Pilvipalveluvalmiustarkastelun suorittaa kohdeyrityksen kohderyhmä Cloud Hosting Team, joka on vastuussa myös sovelluskohtaisten pilvipalveluvalmiustarkastelujen toimituksesta.*

Ensimmäisellä haastattelukierroksella todettiin, että pilvipalveluvalmiustarkastelun suorittaa Cloud Hosting Team, mutta todettiin myös, että se ei kaikkien sovelluksien osalta ole tarpeellinen. Täten pilvipalveluvalmiustarkastelu ehdotetaan toisella kierroksella jaettavan kahteen osaan: tarkastelun tarpeellisuuden määrittelyyn ja päätökseen, jonka suorittaa SharePoint Operations Team, ja kyseisen päätöksen perusteella itse pilvipalveluvalmiustarkasteluun, jonka suorittaa alkuperäisen oletaman mukaisesti Cloud Hosting Team. Toimenpidekokonaisuuden tilaa keskitetysti maatasoon SharePoint muutoskoordinaattori sovellusomistajan pyynnöstä.

*Korjattu olettama 5: Päätöksen pilvipalveluvalmiustarkastelun sovelluskohtaisesta tarpeellisuudesta tekee SharePoint Operations Team. Mikäli pilvipalveluvalmiustarkastelu on tarpeen, sen suorittaa Cloud Hosting Team.*

*Olettama 6: Sovelluskehitys tapahtuu toimittajan (mukaan lukien AMS-palvelu) omassa kehitysympäristössä. Sovellusta ei SharePoint Server-sovellusmallin tapaan ole tarpeen testata kohdeyrityksen kehitysympäristössä.*

Tämä olettama vahvistettiin sellaisenaan ensimmäisen haastattelukierroksen yhteydessä.

*Olettama 7: Sovelluskohtaisten pilvipalvelualustojen tilaus tapahtuu maaston SharePoint muutoskoordinaattorin kautta.*

Ensimmäisen haastattelukierroksen aikana todettiin, että SharePoint muutoskoordinaattorin tulee tilata pilvipalvelualusta, mutta se tulee tilata SharePoint Operations Teamin kautta, eikä suoraan Cloud Hosting Teamilta.

*Korjattu olettama 7: Sovelluskohtaisten pilvipalvelualustojen tilaus tapahtuu maaston SharePoint muutoskoordinaattorin toimesta, SharePoint Operations Teamilta, joka tekee lopullisen tilauksen Cloud Hosting Teamilta.*

Kuvassa 19 ensimmäisen haastattelukierroksen tuloksia heijastaen korjatut implementointi- ja muutosprosessien läpimenoajat.

Laadunvarmistuksen vaihe	Uuden SharePoint Server-sovelluksen implementointiprosessissa (arkipäivää)	SharePoint Server-sovelluksen muutosprosessissa (arkipäivää)	SharePoint Hybridisovelluksen implementointiprosessissa (arkipäivää)	SharePoint Hybridisovelluksen muutosprosessissa (arkipäivää)
Arkkitehtuurikatselmus	5	N/A	5	N/A
Sovelluskehitys	N/A	N/A	N/A	N/A
Kehitysympäristöasennus	2	2	N/A	N/A
<i>Käyttäjättestaus</i>	N/A	N/A	N/A	N/A
Koodikatselmus	5	5	1	1
Testiympäristöasennus	5	5	1	1
<i>Käyttäjättestaus</i>	N/A	N/A	N/A	N/A
Tuotantoympäristöasennus	10	10	1	1
<i>Hyväksyntättestaus</i>	N/A	N/A	N/A	N/A
<b>Läpimenoaika yhteensä (arkipäivää)</b>	<b>27</b>	<b>22</b>	<b>8</b>	<b>3</b>

**Kuva 19.** SharePoint Server- ja SharePoint hybridisovellusten laadunvarmistusprosessien läpimenoaikojen vertailua haastattelutuloksien pohjalta korjattuna.

### 5.3 Toisen haastattelukierroksen tulokset

*Olettama 1 (korjattu): Prosesseihin sisällytetään vain ne sovellukset, jotka käyttävät .NET viitekehityksen teknologioita ja/tai NodeJS-teknologiaa, sekä Microsoft Azure –alustaa.*

Korjattu olettama 1 vahvistettiin toisella haastattelukierroksella myös muiden kohderyhmien puolesta. Kaikki kohderyhmät eivät osanneet ottaa kantaa, mutta kaikki hyväksyivät määritellyn tavan toimia.

*Olettama 2 (korjattu): Nykyiset SharePoint Server-sovellusmallin laadunvarmistusprosessit tulevat osaksi myös hybridisovellusten implementointi- ja muutosprosesseja. Laadunvarmistusprosessien vaiheet ovat tarpeellisia ainoastaan muutoksenhallintatyökalun vaatimien vaiheiden vuoksi, ja vaihekohtaisia läpimenoaikoja voidaan optimoida ja lyhentää huomattavasti.*

Korjattu olettama 2 hyväksyttiin toisella haastattelukierroksella sellaisenaan. Muutoksia muutoksenhallintatyökaluun luvattiin ottaa suunnitteluun jo lähiaikoina. Lopullinen ratkaisu

prosessin läpimenoajan optimoimiseksi tämän osalta jää kuitenkin odottamaan luvattua muutosta. Muutoksenhallintatyökalun muutokset ovat yrityksen jatkokehityslistalla.

*Olettama 3: AMS-palvelu saa täydet oikeudet sovelluskohtaisiin testi- ja tuotantoympäristöihin. Määritelty teoreettinen laadunvarmistuksen läpimenoaika muutosprosessin osalta on täten 0 arkipäivää. Todellinen laadunvarmistuksen läpimenoaika sisältää myös AMS-palvelun asennuksiin käytettävän ajan, sekä sovellusomistajan käyttäjä- ja hyväksyntätestaukset.*

Olettama 3 poistettiin ensimmäisen haastattelukierroksen yhteydessä tämän tutkimuksen yhteydessä tuotettavista prosesseista, mutta lisättiin yrityksen jatkokehityslistalle.

*Olettama 4: Pilvipalveluvalmiustarkastelun kesto, ja siten koko laadunvarmistusprosessin kesto uusien sovelluksien osalta, on 5 arkipäivää.*

Olettaman 4 viiden päivän arvion määriteltiin toisen haastattelukierroksen perusteella olevan konservatiivisen realistinen arvio osuuden todellisesta läpimenoajasta. Pilvipalveluvalmiustarkastelulle ei kuitenkaan tutkimuksen aikana saatu lopullista virallista prosessia. Prosessin kehityksestä on vastuussa Cloud Hosting Team, ja tutkimuksen lopullista prosessia tullaan muokkaamaan edellä mainitun valmistuessa. Pilvipalveluvalmiustarkastelun osuus lisättiin yrityksen jatkokehityslistalle.

Koko laadunvarmistusprosessiin lisättiin kuitenkin ensimmäisen haastattelukierroksen perusteella yhteensä kolme päivää, joten lopullinen läpimenoaika määräytyi kahdeksaksi (8) päiväksi uusien sovellusten osalta.

*Olettama 5 (korjattu): Päätöksen pilvipalveluvalmiustarkastelun sovelluskohtaisesta tarpeellisuudesta tekee SharePoint Operations Team. Mikäli pilvipalveluvalmiustarkastelu on tarpeen, sen suorittaa Cloud Hosting Team.*

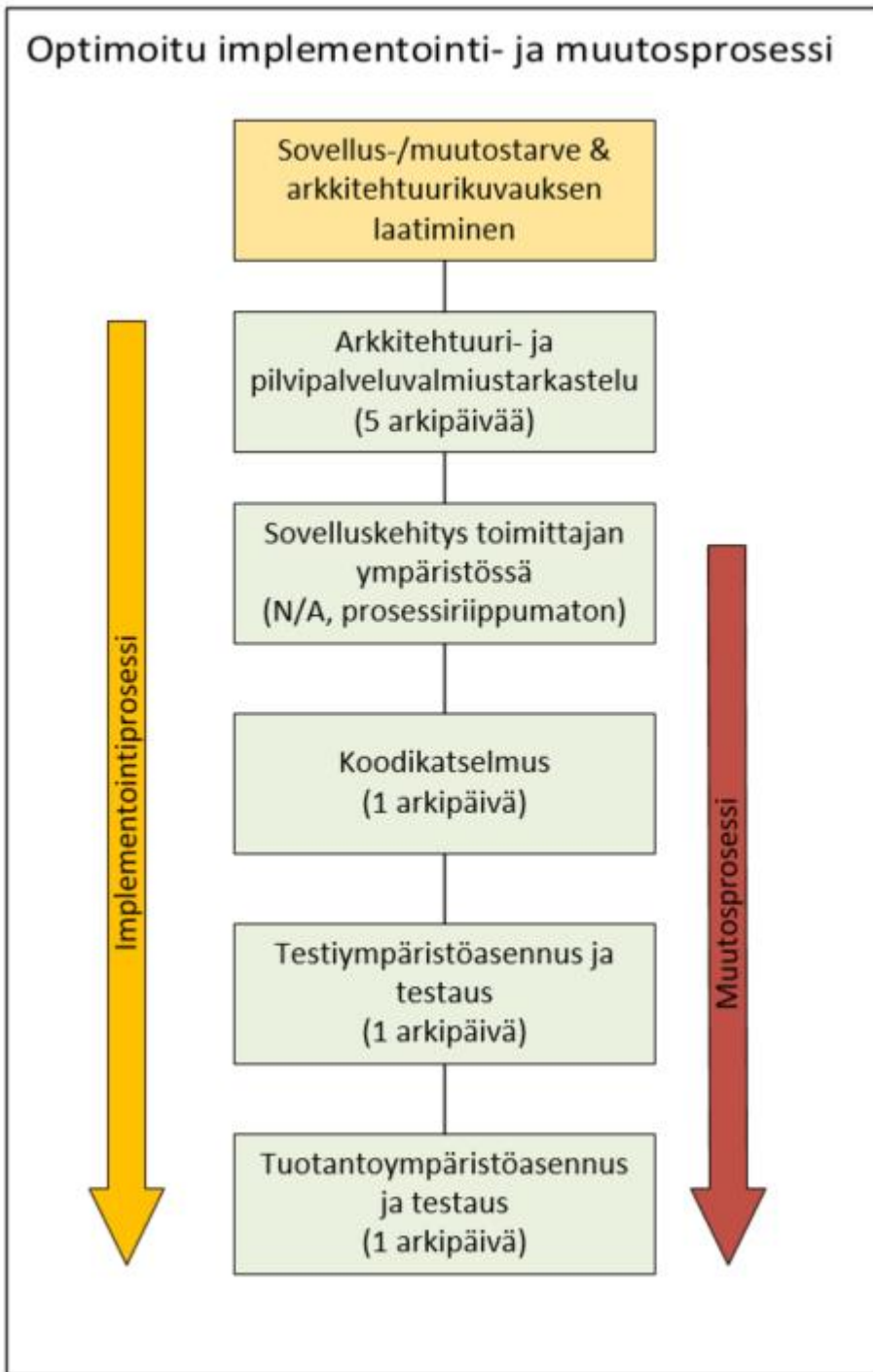
Korjattu olettama 5 vahvistettiin toisen haastattelukierroksen osalta sellaisenaan.

*Olettama 6: Sovelluskehitys tapahtuu toimittajan (mukaan lukien AMS-palvelu) omassa kehitysympäristössä. Sovellusta ei SharePoint Server-sovellusmallin tapaan ole tarpeen testata kohdeyrityksen kehitysympäristössä.*

Olettama 6 vahvistettiin jo ensimmäisen haastattelukierroksen yhteydessä.

*Korjattu olettama 7: Sovelluskohtaisten pilvipalveluvalmiustarkastelujen tilaus tapahtuu maataason SharePoint muutoskoordinaattorin toimesta, SharePoint Operations Teamilta, joka tekee lopullisen tilauksen Cloud Hosting Teamilta.*

Korjattu olettama 7 vahvistettiin toisen haastattelukierroksen osalta sellaisenaan.



**Kuva 20.** Optimoidun implementointi- ja muutosprosessin läpimenoajat kohdeyrityksessä.

#### 5.4 Optimoidut laadunvarmistusprosessit ja tunnistetut jatkokehityskohteet

Haastatteluiden tulosten perusteella määriteltiin yrityksen lopulliset implementointi- ja muutosprosessit. Täten vastattiin myös kolmanteen tutkimuskysymykseen ”Millaisin prosessein voimme viedä eteenpäin sovellusten implementointia ja muutoksia, niin että ne ovat linjassa yrityksen laadunvarmistuksen kanssa?”. Prosessit ja niiden osittain yhteiset vaiheet ovat kuvattuna tässä kappaleessa.

**Arkkitehtuurikatselmus** on myös hybridisovellusten implementointiprosessissa vaadittu tehtävä. Hybridisovelluksen toimittaja vastaa arkkitehtuuridokumentaation toimittamisesta sovellusomistajalle. Arkkitehtuuridokumentaatio tulee luoda kohdeyrityksen ohjeistuksien (engl. Architecture Guideline) mukaisesti. Operations Teamin SharePoint arkkitehti tulee tarkentamaan hybridisovelluksia koskevia tarkennuksia ohjeistukseen, mutta se on sellaisenaan jo pätevä arkkitehtuurikuvauksen luomiseksi. Sovellusomistaja pyytää muutoskoordinaattoria avaamaan arkkitehtuurikatselmuksen muutoksenhallintatyökalussa. Arkkitehtuurikatselmuksessa SharePoint Operations Teamin arkkitehti tarkastelee sovelluksen sopivuutta pilvipalvelualustalle, ennen kaikkea tietoturvallisuuden näkökulmasta. Arkkitehti tekee päätöksen pilvipalveluvalmiustarkastelun tarpeellisuudesta, ja on vastuussa tarkastelun suorittamisesta yhdessä Cloud Hosting Teamin kanssa. Arkkitehti tekee myös pyynnön sovelluksen pilvipalvelukomponenttien toimittamisesta arkkitehtuurikuvauksen perusteella.

Kun arkkitehtuurikatselmus on hyväksytty, saa toimittaja luvan aloittaa **sovelluskehityksen**. Sovelluskehitykselle ei ole määritelty aikarajoja laadunvarmistusprosessien osalta, mutta muuten sovelluskehitysaikataulu sovitaan tapauskohtaisesti sovellusomistajan ja toimittajan välisesti. Toimittaja kehittää sovelluksen omassa kehitysympäristössään kohdeyrityksen antamia ohjeistuksia (coding guideline, deployment guide) noudattaen. Kun sovelluskehitys on valmis, toimittaja lähettää kaiken tarpeellisen materiaalin sovellusomistajalle



testiympäristöasennusta varten, kohdeyhteyksien ohjeistuksien mukaisesti. Sovellusomistaja siirtää tiedon muutoskoordinaattorille, ja pyytää asennustoimenpiteitä.

**Koodikatselmus** vaaditaan toistaiseksi myös hybridisovelluksilta, sillä se on osa muutoksenhallintatyökalun prosessia, eikä implementointi- tai muutosprosessin läpivieminen muutoksenhallintatyökalussa onnistu toistaiseksi ilman koodikatselmustehtävää. SharePoint Operations Team pyrkii lähitulevaisuudessa poistamaan koodikatselmuksen pakollisuuden hybridisovellusten osalta sovelluksien riippumattomuuden vuoksi. Muutoskoordinaattori tekee pyynnön koodikatselmuksesta muutoksenhallintatyökalussa saatuaan materiaalit. Koodikatselmus suoritetaan ns. automaattisena hyväksyntänä, ellei sovellusomistaja erityisesti ole pyytänyt koodikatselmuksen suorittamista myös laadunvarmistuksellisesta näkökulmasta. Koodikatselmuksen suorittaa SharePoint Operations Teamin SharePoint arkkitehti.

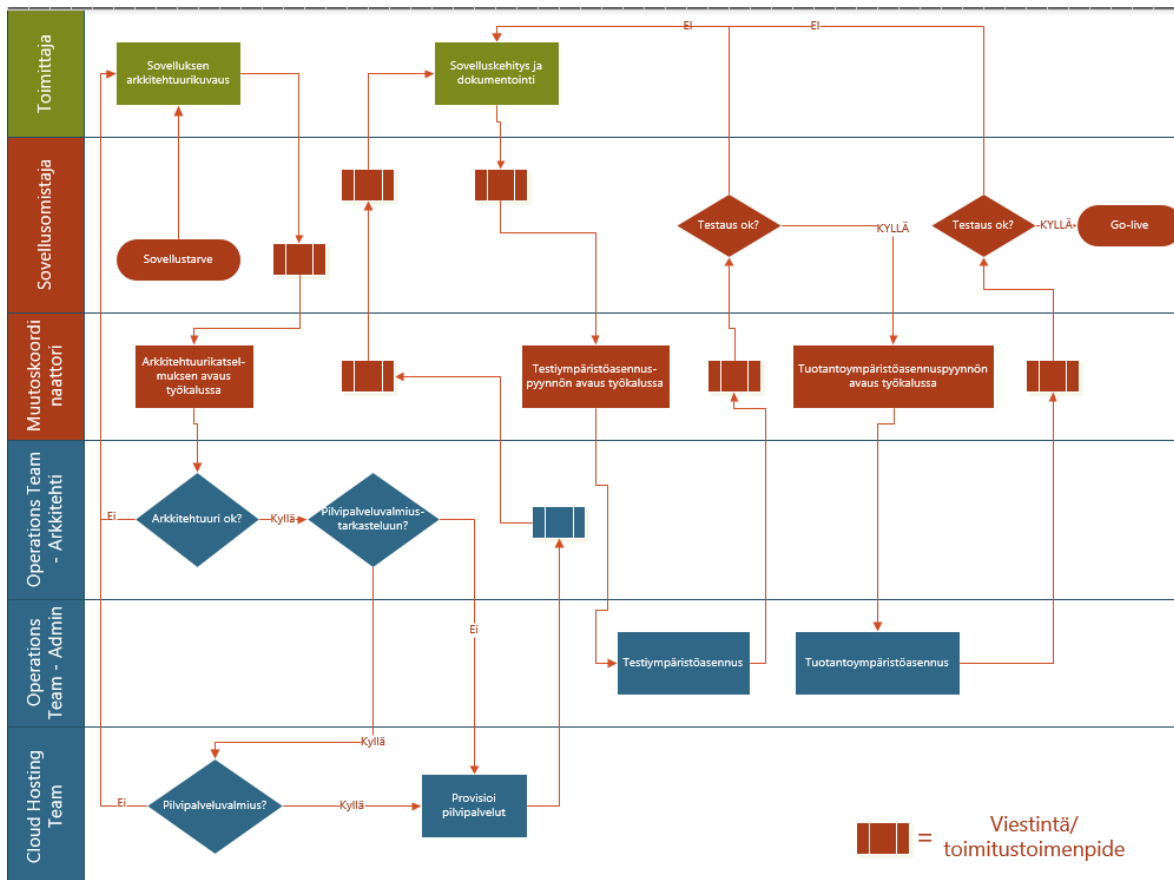
**Testiympäristöasennus** pyydetään muutoksenhallintatyökalusta muutoskoordinaattorin toimesta heti koodikatselmuksen valmistuttua. Pyyntöön ei lisätä erillistä materiaalia, vaan se linkitetään suoritettuun koodikatselmuksipyyntöön, josta löytyy asennusmateriaalit ohjeineen. Operations Teamin administraattori suorittaa asennuksen sovitussa aikataulussa ja merkitsee tehtävän suoritetuksi muutoksenhallintatyökalussa. Muutoskoordinaattori saa tästä tiedon, ja ilmoittaa tehtävän valmistumisesta sovellusomistajalle, joka on vastuussa sovelluksen käyttötestauksesta testiympäristössä. Jos testaukset ovat onnistuneet ja sovellus toimii toivotulla tavalla, pyytää sovellusomistaja tuotantoon vientiä muutoskoordinaattorilta. Muutoin muutoskoordinaattori ilmoittaa sovellusomistajalle epäonnistuneesta asennuksesta, ja pyytää sovellusomistajaa tekemään mahdolliset korjaukset toimittajan kanssa, ja palaamaan sen jälkeen asiaan uuden koodikatselmuksipyynnön muodossa.

**Tuotantoympäristöasennus** pyydetään muutoksenhallintatyökalusta muutoskoordinaattorin toimesta. Pyyntö linkitetään onnistuneeseen

testiympäristöasennuspyyntöön muutoksenhallintatyökalussa, josta löytyy edelleen tarvittavat asennusmateriaalit ohjeineen. Tällä varmistutaan siitä, että sovelluksesta löytyy aina sama versio sekä testi-, että tuotantoympäristössä. Operations Teamin administraattori suorittaa asennuksen sovitussa aikataulussa ja merkkää tehtävän suoritetuksi muutoksenhallintatyökalussa. Muutoskoordinaattori saa tästä tiedon, ja pyytää sovellusomistajaa tekemään lopullisen hyväksyntätestauksen ja käyttöönoton. Epäonnistuneen asennuksen tapauksessa muutoskoordinaattori pyytää jälleen sovellusomistajaa tekemään mahdolliset korjaukset toimittajan kanssa, ja palaamaan sen jälkeen asiaan uuden koodikatselmuspyynnön muodossa.

Yrityksen jatkokehityskohteiden listalle muodostettiin seuraavat asiakohdat:

- 1) Pilvipalvelutarkastelun osuudelle määritelty prosessi (Cloud Hosting Team)
- 2) Toimintatapoja yritetään muuttaa niin, että AMS-palvelu saa täydet oikeudet sovelluskohtaisiin testi- ja tuotantoympäristöihin, jotta laadunvarmistuksen läpimenoaikaa voidaan lyhentää ylläpidollisten toimenpiteiden aiheuttamien muutoksien osalta.
- 3) Muutoksenhallintatyökalun muutokset, joilla saadaan poistettua ns. automaattisesti hyväksyttävät toimenpiteet, joilla saadaan lyhennettyä molempien prosessien läpimenoaikaa.



**Kuva 21.** Hybridisovellusten implementointi- ja muutosprosessi, vastuukuvaus ja työnkulku.

## 6 JOHTOPÄÄTÖKSET

Tutkimus antaa lukijalle viitteitä sovellusten kehityksen ja laadunvarmistusprosessien suunnitteluun, siihen sovellettaviin tutkimusmenetelmiin. Soveltuvan tutkimusmetodin löytyminen oli edellytys tutkimuksen tavoitteiden saavuttamiseksi. Kohdeyrityksen hybridisovellusten laadunvarmistusprosessien osalta toimintatutkimus todettiin tunnistettuja verrokkejaan paremmaksi valinnaksi, pääasiassa koska siinä 1) korostuu yhteistyö kohdeorganisaation kanssa, ja koska se 2) soveltuu prosessikehitykseen erinomaisesti.

Tutkimus antaa lukijalle käsityksen SharePoint hybridisovelluksista, ja osviittaa niiden kehityksessä käytettävistä teknologioista. Tutkimus antaa kohdeyritykselle kattavan kuvan uudesta sovellusmallista, joka epäilemättä tulee kasvattamaan jatkuvasti suosiotaan yrityksen SharePoint-sovellusten portfolioissa. Tämä voitiin vahvistaa haastattelemalla useita eri sidosryhmiä, joista kaikki toisistaan tietämättä muodostivat konsensuksen – hybridisovellukset ovat SharePointin tulevaisuus. Hybridisovellukset laajentavat sovelluskehityksellisiä mahdollisuuksia laajemman teknologiakirjon avulla, vähentävät laadunvarmistuksesta johtuvia kustannuksia, sekä parantavat sovellusten saatavuutta ja elinikää.

Tutkimuksen edetessä tunnistettujen haasteiden määrä kasvoi merkittävästi. Suurimmaksi ongelmaksi tutkimuksen kanssa muodostui hybridisovelluskehitykseen käytettävän teknologiakirjon laajuus. Koska SharePoint hybridisovelluksia voidaan rakentaa käytännössä mihin vain pilvialustaan, lähes millä tahansa teknologisella viitekehityksellä ja millä tahansa koodikielellä, on jokaisen tällaisen objektin tunnistaminen ja analysointi mahdotonta. Tämän vuoksi alkuperäisestä tutkimussuunnitelmasta jouduttiin siirtymään yksittäisten teknologioiden tasolta viitekehitykselle, ja valitun viitekehityksen tarkempaan analyysiin. SharePoint hybridisovelluskehitykseen soveltuvien ja suosittavien teknologioiden löytyminen oli tutkimuskysymyksen 2 keskiössä, ja vaikka sopivia yksittäisiä teknologioita ei saatukaan tutkimuksen tuloksena kirjattua, oltiin viitekehityksen määrittelyyn

kohdeyrityksessä varsin tyytyväisiä. On tärkeää huomioida, että yhden viitekehyksen valinta itsessään ei ole välttämättömyys, mutta sen rajaaminen sovellusylläpidollisesti on todennäköisesti kannattavaa. Tutkimuksen yhdeksi tärkeimmäksi hyödyksi kohdeyritykselle muodostui sopivan viitekehyksen rajaaminen yrityksen AMS-palveluun, sekä implementointi- ja muutosprosesseihin hyväksyttäviin hybridisovelluksiin. Tällä rajauksella minimoitiin ylläpidolliset riskit, sillä käytettävät teknologiat ovat sellaisia, joita AMS-palvelu pystyy tukemaan nykyisellä kokoonpanollaan, ja sillä parannetaan kyseisen palvelun kilpailutusta tulevaisuudessa. Täten voidaan olettaa, että kohdeyritys saa laajemman kilpailutuksen myötä myös taloudellista hyötyä säästöjen muodossa.

Lukija saa ennen kaikkea läpikatsovan kuvan sovellusten laadunvarmistusprosessikehityksen etenemisestä suuressa kansainvälisessä organisaatiossa. Tutkimuskysymykseen *”Millaisin prosessein voimme viedä eteenpäin sovellusten implementointia ja muutoksia, niin että ne ovat linjassa yrityksen laadunvarmistuksen kanssa?”* vastattiin haastattelumenetelmin korjattujen ja validoitujen olettamien perusteella kehitetyin prosessein. Erityisesti iteratiivinen teemahaastatteluperusteinen prosessikehitys todettiin toimivaksi. Käytetystä prosessikehitysmallista voitaneen tehdä vertauskuva sovelluskehityksen ketterän kehityksen malliin. Haastattelut toimivat erinomaisena laadunvarmistuksen työkaluna, sillä menetelmä varmisti kaikkien oleellisten kohderyhmien osallistumisen prosessikehitykseen pirstaloituneessa kohdeyrityksessä. Tutkimuksen lopullisena tuotoksena kohdeyritykselle toimitettiin hybridisovellusmallille uudet implementointi- ja muutosprosessit, jotka ovat aiempia laadunvarmistusprosesseja huomattavasti ketterämpiä varsinkin läpimenoajallisesti.

Tutkimus on kirjoitettu pääosin kohdeyrityksen sovellus- ja järjestelmäkenttää ajatellen. Lukijan on tärkeää huomioida, että tutkimuksen kirjoittaja ei ole sovellusinsinööri, eikä tässä tutkimuksessa ole keskitytty eri teknologioiden teknisiin piirteisiin syvällisemmin. Sen sijaan se antaa kattavan kuvan SharePoint hybridisovellusten teknologisesta ja liiketoiminnallisesta kentästä kokonaisuutena. Toisaalta voidaan myös todeta, että vaikka toimintatutkimus

tutkimusmetodina soveltui erinomaisesti kohdeyrityksen laadunvarmistusprosessien kehitykseen – jää akateemikoille ja praktikoille selvitettäväksi, voidaanko sen soveltuvuutta vastaavaan käyttötarkoitukseen yleistää koko yritys kentässä.

## LÄHTEET

- Anttila, Pirkko (2014). *Tutkimisen taito ja tiedon hankinta* [online]. [19.7.2017]. Saatavilla: <https://metodix.fi/2014/05/17/anttila-pirkko-tutkimisen-taito-ja-tiedon-hankinta/>.
- Booch, Grady (2010). Enterprise Architecture and Technical Architecture. *IEEE Software* [online]. Vol 27: 2, s. 95-96. [27.10.2017].
- Clobridge, Abby (2017). *Opening Up SharePoint* [online]. Online Searcher May/June 2017. s. 63-65. Saatavilla: <http://search.proquest.com.proxy.uwasa.fi/docview/1907273701>.
- Cole, R., S. Purao, M. Rossi & M. Sein (2005). Being Proactive: Where Action Research Meets Design Research. *ICIS Proceedings* [online]. Vol 27, s. 325-336. [6.4.2019].
- Dagar, Rahul (2017). *Differences Between SharePoint 2013 vs. 2016* [online]. [28.3.2019] Saatavilla: <https://www.c-sharpcorner.com/article/features-difference-between-sharepoint-2013-vs-2016>.
- Davidson, R., M. G. Martinsons, & N. Kock (2004). Principles of Canonical Action Research. *Information Systems Journal* [online]. Vol 14: 1, s. 65-86. [20.7.2017].
- Finto. Tiedonhallinta [online]. [17.4.2018] Saatavilla: <http://finto.fi/tt/fi/page/t25?clang=en>
- Harmon, Paul (2003). *Developing an Enterprise Architecture*. Business Process Trends. [online]. [26.10.2017]. Saatavilla: <http://www.bptrends.com/publicationfiles/Enterprise%20Architecture%20Whitepaper-1-23-03.pdf>.

- Hevner, Alan R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems* [online]. Vol 19: 2, s. 87-92. [8.4.2019].
- Holm, H., M. Buschle, R. Lagerström & M. Ekstedt (2012). Automatic Data collection for enterprise architecture models. *Software & Systems Modeling* [online]. Vol 13: 2, s. 825-841. [28.10.2017].
- Irvine, Helen & Michael Gaffikin (2006). Getting in, getting on and getting out: reflections on a qualitative research project. *Accounting, Auditing & Accountability Journal*. [online]. Vol 19: 1, s. 115-145. [7.11.2017].
- Iversen, J. H., L. Mathiassen & P. A. Nielsen (2004). Managing Risk in Software Process Improvement: An Action Research Approach. *MIS Quarterly* [online]. Vol 28: 3, s. 395-433. [17.7.2017].
- Jha, Rajesh (2013). *Cloud services you can trust: Office 365 availability*. Microsoft Office blog [online]. [6.1.2018]. Saatavilla: <https://blogs.office.com/en-us/2013/08/08/cloud-services-you-can-trust-office-365-availability/?eu=true>.
- Johnson, Bruce (2013). *Introduction to SharePoint 2013 App Model* [online]. [7.1.2018] Saatavilla: <https://www.youtube.com/watch?v=-5EkgFeTR8Y>
- Juvonen, Vesa (2013). *SharePoint 2013 App Model – is it really the future of SharePoint* [online]. [7.1.2018] Saatavilla: <https://www.youtube.com/watch?v=7Sf0-g0EbZw>.
- Järvinen, Pertti (2007). Action Research is Similar to Design Science. *Quality & Quantity* [online]. Vol 41: 1, s. 37-54. [8.9.2019].



- Kirsch, Laurie (1996). The Management of Complex Tasks in Organizations: Controlling the Systems Development Process. *Organization Science* [online]. Vol 7: 1, s. 1-21. [12.4.2020].
- Li, Chulin & Layuan Li (2013). Efficient Resource Allocation for Optimizing Objectives of Cloud Users, IaaS Provider and SaaS Provider in Cloud Environment. *The Journal of Supercomputing* [online]. Vol 65: 2, s. 866-885. [29.10.2017].
- Linturi, Hannu (2003). *Toimintatutkimus*. [online]. [7.11.2017]. Saatavilla: [http://nexusdelfix.internetix.fi/sv/sisalto/materiaalit/2\\_metodit/5\\_actix?C:D=61566&C:selres=61566](http://nexusdelfix.internetix.fi/sv/sisalto/materiaalit/2_metodit/5_actix?C:D=61566&C:selres=61566).
- Mackie, Kurt (2017). *Microsoft Unveils SharePoint Server 2019 and Other Upcoming Improvements* [online]. [29.10.2017] Saatavilla: <https://redmondmag.com/articles/2017/09/27/microsoft-unveils-sharepoint-server-2019.aspx>.
- Mamaghani, N. D., F. M. Madani & A. Sharifi (2011). Customer Oriented Enterprise IT Architecture Framework. *Telematics and Informatics* [online]. Vol 29: 2, s. 219-232. [28.10.2017].
- Mamone, Salvatore (1994). The IEEE Standard for Software Maintenance. *ACM SIFSOFT Software Engineering Notes* [online]. Vol 19: 1, s. 75-76 [6.12.2017].
- Microsoft (2017a). SharePoint Add-ins [online]. [17.4.2018] Saatavilla: <https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/sharepoint-add-ins>

- Microsoft (2017b). SharePoint Framework Overview [online]. [17.4.2018] Saatavilla: <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/sharepoint-framework-overview>
- Nogueira, J. M., Romero, D., Espadas, J. & A. Molina (2012). Leveraging the Zachman Framework Implementation Using Action-research Methodology – a Case Study: Alining the Enterprise Architecture and the Business Goals. *Enterprise Information Systems*. [online]. Vol 7: 1, s. 100-132. [11.11.2017].
- Ouchi, William (1979). A Conceptual Framework for the Design of Organizational Control Mechanisms. *Management Science* [online]. Vol 25: 9, s. 833-847. [12.4.2020].
- Romppanen, Jarkko (2017). Dokumentoinnin hyvät käytännöt – katsaus SharePoint-ratkaisuihin. Tietotekniikan kandidaatintutkielma. Vaasan Yliopisto. S. 19-20. [19.11.2017].
- Roumani, Karim (2015). *History of SharePoint: The Past, Present and Future* [online]. [29.10.2017] Saatavilla: <https://www.portalfronthosting.com/blog/history-of-sharepoint-the-past-present-and-future>.
- Salas-Zárate, M. & L. Colombo-Mendoza (2012). Cloud Computing: A Review of PaaS, SaaS Services and Providers. *Lámpsakos*. [online]. Vol 2012: 7, s. 47-57. [29.10.2017].
- Sein, M. K., O. Henfridsson, S. Purao, M. Rossi & R. Lindgren (2011). Action Design Research. *MIS Quarterly* [online]. Vol 35: 1, s. 37-56. [20.7.2017].
- Simon, Herbert A. (1984). *The Sciences of the Artificial*. 2. painos. Cambridge etc.: The Massachusetts Institute of Technology. 229 s. ISBN 0-262-19193-8.

- Sowa, J. F. & J. A. Zachman (1992). Extending and formalizing the framework for information systems architecture. *IBM Systems Journal* [online]. Vol 31: 3, s. 590-616. [19.11.2017].
- Susman, G. I. & R. D. Evered (1978). An Assessment of the Scientific Merits of Action Research. *Administrative Science Quarterly* [online]. Vol 23: 4, s. 582-603. [8.4.2019].
- Tiainen, T., J. Aittoniemi, I. Haukijärvi & T. Yli-Karhu (2015). *Toimintatutkimus tietojenkäsittelytieteen tutkimuksessa* [online]. Tampere: Tampereen yliopiston informaatiotieteiden yksikkö. [9.7.2017]. Saatavilla: [http://tampub.uta.fi/bitstream/handle/10024/97044/toimintatutkimus\\_tietojenkäsittely\\_2015.pdf](http://tampub.uta.fi/bitstream/handle/10024/97044/toimintatutkimus_tietojenkäsittely_2015.pdf). ISBN 978-951-44-9814-5.
- Tiainen, Tarja. *Haastattelu tietojenkäsittelytieteen tutkimuksessa – haastattelun suunnittelu, analyysi ja raportointi*. [29.10.2017]. Saatavilla: [https://moodle.uwasa.fi/pluginfile.php/101941/mod\\_resource/content/1/Haastattelu-TT-14.pdf](https://moodle.uwasa.fi/pluginfile.php/101941/mod_resource/content/1/Haastattelu-TT-14.pdf).
- Tiainen, Tarja (2014). Laadullinen tutkimus – menetelmät ja prosessi – kurssin luennot. [28.3.2019]. Tampereen yliopisto, informaatiotieteiden yksikkö, syksy 2014.
- Webber-Cross, Geoff (2014). *Learning Microsoft Azure* [online]. Kustannuspaikka: Packt Publishing, [7.11.2017]. Saatavilla: <https://ebookcentral-proquest-com.proxy.uwasa.fi/lib/tritonia-ebooks/reader.action?docID=1818072>. ISBN: 978-178-21733-8-0.
- Zachman, J. A. (1999). A Framework for Information Systems Architecture. *IBM Systems Journal* [online]. Vol 38: 2-3, s. 454-470 [2.12.2017].