



**TURUN
YLIOPISTO**
Kauppakorkeakoulu

Pilvinatiivin kehittämisen liiketoimintaedut organisaatioiden omalle tietojärjestelmäkehitykselle

Tietojärjestelmätieteen
kandidaatintutkielma

Laatija:
Miro Nissilä

Ohjaaja:
FT Kai Kimppa

10.12.2024
Turku

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -järjestelmällä.

Kandidaatintutkielma

Oppiaine: Tietojärjestelmätiede

Tekijä: Miro Nissilä

Otsikko: Pilvinatiivin kehittämisen liiketoimintaedut organisaatioiden omalle tietojärjestelmäkehitykselle

Ohjaaja: FT Kai Kimppa

Sivumäärä: 33 sivua

Päivämäärä: 10.12.2024

Pilvipalvelut ovat tuoneet organisaatiolle uuden tavan kehittää omia tietojärjestelmiään. Yhä suurempi osa organisaatioiden käsittelemistä ohjelmistokuormista, kuten laskentaresursseista ja datasta sijaitsee julkisessa pilvessä. Pilvinatiivin kehittämisen avulla organisaatiot pystyvät ottamaan täyden hyödyn irti pilvipalveluiden tarjoamasta käytön mukaan tapahtuvasta laskutuksesta ja joustavuudesta.

Tämän kandidaatintutkielman tarkoituksena on selvittää, mitä pilvinatiivi kehittäminen itsessään on ja mitä liiketoimintahyötyjä se tarjoaa organisaatiolle, jotka kehittävät itse omia tietojärjestelmiään. Tutkielmassa määritellään pilvinatiiviin kehittämiseen liittyvät keskeiset käsitteet ja menetelmät ja arvioidaan niiden liiketoimintaetuja perinteiseen IT-infrastruktuuriin verrattuna. Tutkielma on toteutettu kirjallisuuskatsauksena, jossa aineistona on hyödynnetty tieteellisiä artikkeleita ja muuta aiheeseen liittyvää kirjallisuutta.

Pilvinatiivia kehittämistä tarkastellaan niin siihen liittyvien teknologisten valintojen, kehittämismenetelmien, kuin järjestelmien ominaisuuksienkin kautta. Yleisesti pilvinatiivit järjestelmät hyödyntävät joko mikropalvelu- tai palvelimetonta arkkitehtuuria ja niitä kehitetään DevOps menetelmin. Kokonaisvaltainen lähestymistapa auttaa hyödyntämään pilvipalveluita maksimaalisesti.

Pilvinatiivi kehittäminen tuo organisaatioille liiketoimintaetua poistamalla alkuinvestointien tarpeen omaan laitteistoon ja mahdollistamalla käytön mukaan skaalautuvat kulut. Pilvinatiivit järjestelmät hyödyntävät pilven joustavuutta, minkä ansiosta resurssit mukautuvat dynaamisesti kysynnän vaihteluihin. Tämä parantaa resurssitehokkuutta ja vähentää IT-infrastruktuurin ylimääräistä kapasiteettia. Lisäksi pilvinatiivi kehittäminen nopeuttaa järjestelmien kehitystä ja markkinoille pääsyä hyödyntämällä valmiita pilvipalvelukomponentteja ja moderneja kehitysmenetelmiä.

Avainsanat: pilvinatiivi, julkinen pilvi, DevOps, mikropalvelut, palvelimeton arkkitehtuuri

SISÄLLYS

1	Johdanto	6
2	Julkinen pilvi ja pilvinatiivi kehittäminen	7
2.1	Julkinen pilvi ja pilvipalvelut	7
2.2	Pilvinatiivi kehittäminen	8
2.3	Pilvinatiivi arkkitehtuuri	9
2.4	Pilvinatiivit menetelmät	10
2.5	Haasteet pilvinatiivissa kehittämisessä	11
3	Organisaatioiden oma tietojärjestelmäkehitys	14
3.1	Kehittämisen valmiuden arviointi ja sen hyödyt	14
3.2	Kehittämisen riskit	16
4	Pilvinatiivin kehittämisen liiketoiminnalliset hyödyt	18
4.1	Kustannustehokkuus	18
4.2	Teknologiset tekijät	19
4.2.1	Skaalautuvuus ja elastisuus	19
4.2.2	Innovointi ja nopeampi markkinoille pääsy	20
4.2.3	Resilienssi, saatavuus ja tietoturva	21
5	Yhteenveto ja johtopäätökset	23
	Lähteet	28

KUVIOT

Kuva 1 Paikallinen palvelu ja pilvipalvelumallit (mukaillen Red Hat 2022)	8
Kuva 2 DevOps kehä (alkuperäinen kuvio Gunia, 2024)	11
Kuva 3 Pilvinatiivi kehittäminen	23

TAULUKOT

Taulukko 1 Päätöksentekoon vaikuttavat keskeiset tekijät	14
Taulukko 2 Pilvinatiivin kehittämisen liiketoimintahyödyt ja haasteet	25

1 Johdanto

Vuonna 2010 Armbrust ym. (2010) arvioivat pilvipalveluilla olevan potentiaalia muuttaa IT-alaa merkittävästi. Yritykset eivät enää tarvitsisi suuria alkuinvestointeja omille koneistoleilleen, vaan laskentatehoa ja tallennustilaa on saatavilla ketterästi palveluna internetin kautta. Julkisessa pilvessä palveluntarjoaja omistaa fyysisen infrastruktuurin ja pilven resurssit ovat avoinna kaikille organisaatioille käytön mukaan laskutettuna (Ali ym., 2015). Flexeran raportin mukaan vuonna 2024 noin puolet organisaatioiden hallinnoimista ohjelmistokuormista, resursseista ja datasta sijaitseekin julkisessa pilvessä (Flexera, 2024).

Vaikka julkisen pilven käyttö itsessään vapauttaa yrityksiä keskittymään ydinliiketoimintaansa sen sijaan, että niiden tarvitsisi perustaa ja hallinnoida omia datakeskuksiaan, on myös tärkeää, miten pilvipalveluja käytetään. Pilvinatiivit ohjelmistot hyödyntävät pilvipalveluntarjoajien mikropalveluita ja valmiita komponentteja sovellusarkkitehtuurissa jättäen näin organisaatioille enemmän resursseja varsinaiseen ohjelmistokehitykseen. (Gannon ym., 2017.) Kansainvälisesti tunnetut yritykset, kuten Netflix, eBay ja Uber ovat raportoineet pilvinatiivien teknologioiden käyttöönoton tuomista hyödyistä (Blinowski ym., 2022). Pilvinatiivi kehittäminen on huomioitu myös Gartnerin hypekäyrällä vuonna 2023 yhtenä tärkeimpänä viitekehityksenä, johon kohdistuu eniten odotuksia (Gartner, 2023).

Mutta mitä kaikkea pilvinatiivi kehittäminen pitää sisällään? Mitä hyötyä organisaatiot saavuttavat julkisen pilven käyttöönotolla ja järjestelmien kehittämisestä pilvinatiivisti? Tässä tutkielmassa perehdytään julkisilven sekä pilvinatiiviin kehittämiseen organisaatioiden oman tietojärjestelmäkehityksen liiketoimintaetua tuovana tekijänä.

Tutkielman tutkimuskysymykset ovat seuraavat:

1. Mitä on pilvinatiivi kehittäminen?
2. Mitä liiketoiminnallista hyötyä pilvinatiivista kehityksestä on organisaation omassa tietojärjestelmäkehityksessä?

Tutkielmassa keskitytään julkisilven ja pilvinatiivin kehittämisen tuomiin kilpailuetuihin omaa sovelluskehitystä harjoittavien yritysten näkökulmasta. Pilvipalveluista näkökulma rajataan tietojärjestelmäkehityksessä käytettäviin pilvipalveluihin.

2 Julkinen pilvi ja pilvinatiivi kehittäminen

2.1 Julkinen pilvi ja pilvipalvelut

Pilvipalveluissa jaetaan ja hyödynnetään digitaalisia resursseja internetin kautta. Sovellukset, laskentakapasiteetti ja tallennustilat ovat käytettävissä verkossa, eli pilvessä, ja käyttäjät voivat hyödyntää niitä internetin välityksellä. (Piccoli & Pigni, 2022.) Pilvipalveluilla tarkoitetaan siis sekä palveluna internetin kautta tarjottavia sovelluksia että palveluja tuottavien datakeskusten laitteistoja ja järjestelmäohjelmistoja. Julkisesta pilvestä on kyse, kun datakeskusten laitteiston ja ohjelmiston laskentakapasiteetti sekä tallennustila ovat yleisesti saatavilla suurelle asiakaskunnalle. (Armbrust ym., 2010.)

Keskeisimmät erot julkisen pilven ja paikallisten ratkaisuiden välillä liittyvät infrastruktuurin kustannusrakenteeseen sekä hallintaan ja ylläpitoon. Kun julkisessa pilvessä resurssit sijaitsevat pilvipalvelunpalveluntarjoajan datakeskuksissa ja ne ovat käytettävissä internetin kautta, paikallisissa ratkaisuissa resurssit sijaitsevat yrityksen omissa tiloissa ja niitä hallinnoidaan paikallisesti. (Ali ym., 2024.) Julkisessa pilvessä pilvipalveluntarjoaja vastaa myös datakeskusten infrastruktuurin ja järjestelmäsovellusten ylläpidosta sekä päivityksistä. Tällä on vaikutusta myös infrastruktuurin kustannusrakenteeseen, sillä hallinnoinnin ulkoistaminen vapauttaa organisaation sisäisten IT-tiimien resursseja. (Wease ym., 2018.) Julkisen pilven hyödyntäminen ei myöskään vaadi suuria ennakkokustannuksia omaan datakeskukseen ja laitteistoon. Tyypillistä julkisessa pilvessä toimimisessa on myös resurssien käytön mukaan tapahtuva laskutus. (Ali ym., 2024.)

Tällainen palvelumalli, jossa asiakas ostaa pilvipalveluntarjoajalta laitteiston toiminnallisuutta hankkimatta itse laitteistoa, on nimeltään IaaS (engl. Infrastructure as a Service). Palveluntarjoaja huolehtii IaaS-mallissa itse infrastruktuurin toimintojen suorittamisesta ja ylläpidosta ja asiakas vuokraa vain virtualisoidun laitteiston. (Piccoli & Pigni, 2022.) Virtualisoidussa laitteistossa fyysiset laitteistoresurssit, kuten palvelimet tallennuslaitteet sekä verkkoresurssit, muutetaan ohjelmiston avulla virtuaalisiksi versioiksi. Virtualisointiteknologialla fyysiset palvelimet voidaan jakaa useisiin virtuaalikoneisiin ja näin suorittaa useita käyttäjärjestelmiä ja sovelluksia yhtäaikaisesti. (Deka & Das, 2014.)

Muita palvelumalleja pilvipalveluille ovat SaaS (engl. Software as a Service) ja PaaS (engl. Platform as a Service). Palvelumallit eroavat keskenään tarjottavien resurssien sekä asiakkaan vastuiden osalta. Kuvio 1 havainnollistaa asiakkaan ja palveluntarjoajan vastuut paikallisten palveluiden ja pilvipalvelumallien välillä. SaaS-mallissa palveluntarjoaja tarjoaa valmiin, täysin

pilvessä toimivan ohjelmiston. Palveluntarjoaja vastaa ohjelmiston infrastruktuurista sekä ylläpidosta. PaaS-mallissa pilvipalveluntarjoaja tarjoaa alustan, joka toimii kehitysympäristönä sovellusten rakentamisen, testaamiseen ja käyttöönottoon. Palveluntarjoaja vastaa siis taustalla olevasta infrastruktuurista ja asiakkaan vastuulla on itse ohjelmiston kehittäminen. (Piccoli & Pigni, 2022.)

Paikallinen palvelu	IaaS	PaaS	SaaS
Sovellus	Sovellus	Sovellus	Sovellus
Data	Data	Data	Data
Ajoaika	Ajoaika	Ajoaika	Ajoaika
Väliohjelmisto	Väliohjelmisto	Väliohjelmisto	Väliohjelmisto
Käyttöjärjestelmä	Käyttöjärjestelmä	Käyttöjärjestelmä	Käyttöjärjestelmä
Virtualisointi	Virtualisointi	Virtualisointi	Virtualisointi
Palvelimet	Palvelimet	Palvelimet	Palvelimet
Tallennus	Tallennus	Tallennus	Tallennus
Verkkoyhteys	Verkkoyhteys	Verkkoyhteys	Verkkoyhteys

■ Asiakas hallinnoi
■ Palveluntarjoaja hallinnoi

Kuva 1 Paikallinen palvelu ja pilvipalvelumallit (mukaillen Red Hat 2022)

Vaikka pilvipalvelumallien erottelu on tärkeää ovat ne kuitenkin vahvasti yhteydessä toisiinsa. Pandey ym. (2016) kuvaavat palvelumallien kerroksisuutta, joka ei näy loppukäyttäjälle. Loppukäyttäjä voi käyttää valmista SaaS-ohjelmistoa, jonka palveluntarjoaja on rakentanut toisen PaaS-palveluntarjoajan alustalle. PaaS-palveluntarjoaja puolestaan on voinut rakentaa alustan IaaS-palvelun päälle, jolloin palveluntarjoajien ketjuun linkittyy myös IaaS-palveluntarjoaja.

2.2 Pilvinatiivi kehittäminen

Terminä pilvinatiivia on käytetty laajasti pilvipalveluiden yleistyessä 2000-luvulla, mutta sittemmin sen esiintyminen on kadonnut lähes kokonaan. 2015 vuodesta eteenpäin pilvinatiivi käsitteen käyttö on kuitenkin kasvanut moninkertaisesti. Tieteellisessä kirjallisuudessa termille ”pilvinatiivi” ei ole

yksiselitteistä määritelmää, mutta tutkimuksessa korostuu silti yleinen samansuuntainen käsitys termin merkityksestä (Kratzke & Quint, 2017.)

Ymmärrystä pilvinatiivi termin merkityksestä voidaan hahmottaa tarkastelemalla pilvinatiiveille järjestelmille yhteisiä ominaisuuksia. Gannon ym. (2017) havainnoivat, että pilvinatiivien järjestelmien ominaispiirteisiin kuuluvat muun muassa globaali toiminta, jossa hyödynnetään paikallisia datakeskuksia latenssin minimoimiseksi. Järjestelmät skaalautuvat saumattomasti tuhansille samanaikaisille käyttäjille, ja ne on suunniteltu joustavaa infrastruktuuria varten siten, että virheiden esiintyminen on oletusarvoinen osa kehitystä. Lisäksi järjestelmät mahdollistavat jatkuvan tuotantoympäristön toiminnan ilman keskeytyksiä, vaikka niitä päivitetään ja testataan säännöllisesti. Tietoturvan huomioiminen on myös keskeinen osa sovellusarkkitehtuuria.

Pilvinatiivit järjestelmät on siis suunniteltu ja rakennettu hyödyntämään maksimaalisesti modernien pilvipalveluympäristöjen tarjoamia etuja (Lichtenthäler & Wirtz, 2024). Myös Cloud Native Computing Foundationin määritelmän mukaan pilvinatiivit käytännöt mahdollistavat organisaatioiden kehittämää, rakentaa sekä ottaa käyttöön työkuormia eri pilvipalveluympäristöissä skaalautuvasti ja toistettavasti (CNCF, 2024).

2.3 Pilvinatiivi arkkitehtuuri

Perinteisesti ohjelmistoja on kehitetty monoliittisen arkkitehtuurin mukaan, jossa ohjelmisto rakennetaan yhteen koodipohjaan, joka sisältää useita palveluita eli ohjelmiston toiminnallisia osia. Monoliittinen ohjelmisto toimii siis yhtenä prosessina ja esimerkiksi ohjelmiston uuden version käyttöönotto tapahtuu yhdellä kerralla. (Blinowski ym., 2022.) Vaikka tällaisen ohjelmiston voikin siirtää julkiseen pilveen monoliittisena virtuaalikoneena, voi se altistaa niitä saatavuusongelmille sekä kysyntäpiikeille, eikä se silloin mahdollista julkisen pilven täyttä hyödyntämistä. Siksi ohjelmistojen kehittäminen pilveen vaatiikin erilaista lähestymistapaa ohjelmistoarkkitehtuuriin ja ohjelmistokehitysprosesseihin. (Brunner ym., 2015.)

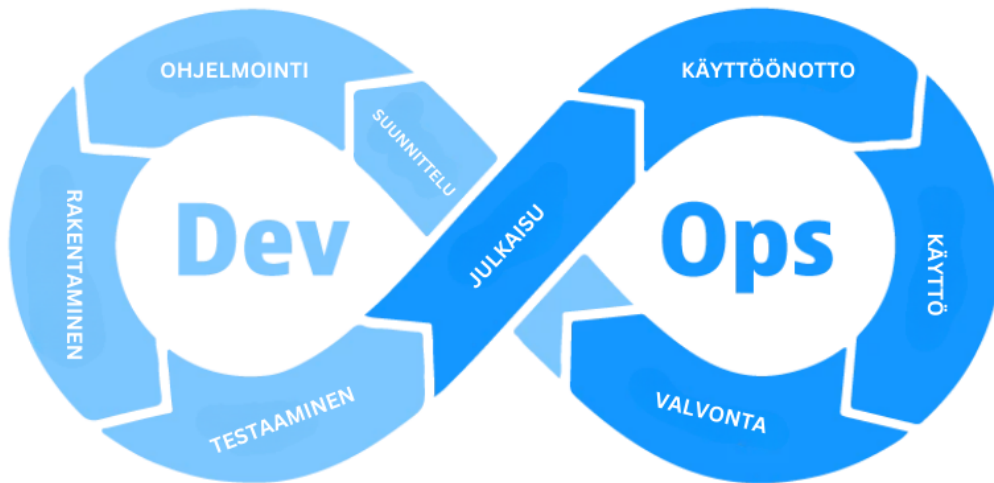
Mikropalveluarkkitehtuurin avulla voidaan kuitenkin hyödyntää laajemmin julkipilvialustojen tarjoamia hyötyjä. Se on myös yksi yleisimmistä tavoista muuntaa vanhoja monoliittisiä järjestelmiä pilvinatiiveiksi. (Hasan ym., 2023.) Mikropalveluarkkitehtuurissa sovellus jaetaan useisiin pieniin itsenäisiin palveluihin, joilla on rajattu vastuualue ja jotka toimivat erillään toisistaan. Jokainen mikropalvelu voidaan replikoida, skaalata, päivittää, ottaa käyttöön ja sitä voidaan hallinnoida itsenäisesti. (Gannon ym., 2017.) Mikropalveluarkkitehtuurissa, ohjelmisto koostuu siis ikään kuin useista pienemmistä sovelluksista, joilla on yksilölliset tehtävät. Täten esimerkiksi yhden

mikropalvelun päivittämisellä tai kaatumisella ei ole vaikutuksia muihin mikropalveluihin tai ohjelmistoihin. Mikropalveluiden tehokas hallinta ja käyttöönotto on mahdollista kontittamisella, joka mahdollistaa useiden sovellusten suorittamisen yhdellä virtuaalikoneella. (Hardikar ym., 2021.) Kontit mahdollistavat sovellusten suorittamisen eristetyissä ympäristöissä samalla käyttöjärjestelmällä. Koska kontit eivät vaadi ylimääräistä käyttöjärjestelmän kopiointia, ne käyttävät resursseja tehokkaammin kuin perinteiset virtuaalikoneet. Laittamalla ohjelma konttiin voidaan sitä siirtää tehokkaasti eri ympäristöjen välillä. (Martin ym., 2024.)

Pilvinatiivien sovellusarkkitehtuurien kehityksessä pyritään optimoimaan resurssien käyttäminen, joka on johtanut monoliittisista arkkitehtureista siirtymisen itsenäisesti käyttöönotettaviin mikropalveluihin. Palvelimeton arkkitehtuuri vie erikseen tarjottavien palveluiden hyödyntämisen ja arkkitehtuurin hajauttamisen vielä asteen pidemmälle. (Kratzke, 2018.) Toisin kuin mikropalveluarkkitehtuurissa, palvelimettomassa arkkitehtuurissa ei järjestelmäkehityksessä tarvitse varata laskentaresursseja palveluiden suorittamiseen, vaan määritellään tapahtumat, milloin koodi suoritetaan ja pilvipalveluntarjoaja huolehtii resurssien hallinnasta ja skaalaamisesta. (Gannon ym., 2017.) Palvelimettomassa arkkitehtuurissa infrastruktuurin operointi on siis piilotettuna kehittäjiltä, joka mahdollistaa keskittymisen sovelluksen liiketoiminnalliseen logiikkaan ja käyttäjäkokemukseen (Pelle ym., 2021).

2.4 Pilvinatiivit menetelmät

DevOps voidaan määritellä kokoelmaksi parhaita käytäntöjä, joiden tavoitteena on vähentää aikaa, joka kuluu järjestelmään tehtävien muutosten viemiseksi tuotantoympäristöön, pienentämällä kehitys- ja tuotantotoimintojen välistä kuilua. Vaikka DevOpsia voidaan käyttää myös monoliittisten järjestelmien kehittämisessä, mikropalvelut tekevät DevOpsin toteuttamisesta tehokkaampaa. (Balalaie ym., 2016.) DevOpsia hyödyntäessä kehitettävän palvelun tai ohjelmiston elinkaari ei katkea missään vaiheessa, vaan DevOps tiimi huolehtii niin ohjelmiston kehityksestä, testauksesta ja julkaisusta. Koska DevOps on laajennus ketteristä ohjelmistokehitysmenetelmistä, on automaatio myös keskeisessä roolissa julkaisujen viiveiden minimoimisessa. (Gokarna & Singh, 2021.) Kuva 2 havainnollistaa DevOpsin vaiheita.



Kuva 2 DevOps kehä (alkuperäinen kuvio Gunia, 2024)

Keskeisimpinä käytäntöinä DevOps:issa ovat jatkuva integraatio (engl. continuous integration, CI) ja jatkuva toimitus (engl. continuous delivery, CD) (Gokarna & Singh, 2021). Siinä ohjelmiston toimitusprosesseja, kuten testausta, integrointia ja käyttöönottoa automatisoidaan, mikä nopeuttaa uusien ominaisuuksien tuomista loppukäyttäjille. Tavoitteena on, että uudet ominaisuudet ja muutokset saadaan heti käyttäjille, ilman että niitä tarvitsee odottaa seuraavaan suunniteltuun julkaisupäivään. (Klotins ym., 2022.)

Keskeisessä roolissa DevOpsia hyödyntävissä organisaatioissa on myös infrastruktuuri koodina (engl. infrastructure as code, IaC). Se on lähestymistapa, jossa infrastruktuuri määritellään ja sitä hallitaan koodin avulla sen sijaan, että sitä provisioitaisiin manuaalisesti. Tämä mahdollistaa infrastruktuurin automaation, jolloin sovellusten käyttöönotto ja resurssien hallinta voidaan suorittaa luotettavasti ja toistettavasti. IaC avulla DevOps-tiimit pystyvät hallitsemaan ja skaalamaan pilvinatiiveja sovelluksia tehokkaasti, sillä infrastruktuurin muutokset voidaan tehdä nopeasti ja yhtenäisesti kaikissa ympäristöissä. (Quattrocchi & Tamburri, 2023.)

2.5 Haasteet pilvinatiivissa kehittämisessä

Tietoturva on yksi merkittävimmistä haasteista pilvipalveluissa. Perinteisessä IT-ympäristössä organisaatio säilyttää täyden hallinnan digitalisesta omaisuudestaan ja tietojen käsittely, siirto ja hallinta tapahtuu sen omassa valvonnassa. Pilvipalveluissa organisaatio ei pääse kuitenkaan suoraan

hallinnoimaan palveluita tai infrastruktuuria vaan joutuu siirtämään näitä pilvipalveluntarjoajan vastuulle. Pilvipalveluntarjoajan toteuttamat tietoturvatkaisut eivät myöskään yleensä ole organisaation nähtävissä, mikä voi lisätä epävarmuutta. (Ali ym., 2015.)

Mikropalvelut, kontittaminen ja DevOps metodien käyttö ovat myös tehneet pilvinatiiveista sovelluksista monimutkaisempia ja hankalammin hallittavia perinteisiin monoliittisiin sovelluksiin verrattuna. Kompleksisemmän rakenteensa takia pilvinatiivit sovellukset altistuvat yhä moninaisimmille hyökkäyksille, kuten automatisoiduille hyökkäyksille, laajoille palvelunestohyökkäyksille sekä suunnittelu-, kooditaso ja konfiguraatioheikkouksien hyväksikäytölle. Myös palvelimettomat arkkitehtuurit altistavat pilvinatiiveja sovelluksia kooditason haavoittuvuuksille, vaikka infrastruktuurin hallinta onkin pilvitoimittajan vastuulla. Palvelimettomien sovellusten monimutkaiset syötelähteet ja tapahtumakäynnistimet tekevät myös perinteisten uhkien tunnistamisesta ja hallinnasta vaikeampaa. Kustannustehokkuus voi myös kääntyä taloudelliseksi tappioksi, jos hyökkäyksellä aiheutetaan infrastruktuurin resurssien loppuminen. Sovelluskehittäjät joutuvatkin tasapainottelemaan yhä enemmän toimitusnopeuden ja tietoturvan parantamisen välillä pilvinatiiveja sovelluksia kehittäessä. (Chernyshev ym., 2021.)

Pilvinatiivien järjestelmien nopea kehityssykli voi johtaa teknisen velan syntymiseen. Teknisellä velalla tarkoitetaan ohjelmistokehityksen eri vaiheissa tehtyjä kompromisseja kehityksen nopeuttamiseksi, jotka johtavat järjestelmän kalliimpaan ylläpitoon ja kehittämiseen tulevaisuudessa. Pilvinatiiveissa järjestelmissä tekninen velka voi liittyä esimerkiksi arkkitehtuurin suunnitteluun, koodin laatuun, testauksen kattavuuteen tai tietoturvaan. Kiireiset aikataulut tai epäselvät vaatimukset voivat johtaa väliaikaisiin ratkaisuihin, joiden seurauksesta järjestelmään voi jäädä rakenteellisia ongelmia. Pilvinatiiveissa sovelluksissa teknisen velan vaikutukset korostuvat järjestelmän skaalautuvuudessa ja tietoturvassa. Huolimattomasti suunniteltu arkkitehtuuri voi estää järjestelmän tehokkaan skaalautumisen ja aiheuttaa merkittäviä lisäkustannuksia infrastruktuurin optimoinnissa. Huomiotta jääneet tietoturva haavoittuvuudet voivat myös aiheuttaa ongelmia nopeasti muuttuvissa ympäristöissä. Kehityksen nopeuttamiseksi tehdyt ratkaisut voivat loppupeleissä myös johtaa järjestelmän toimituksen viivästymiseen. Teknisen velan hallinta vaatii systemaattista lähestymistapaa, joka yhdistää kattavan testauksen, jatkuvan koodin laadun seurannan sekä ennakoivan arkkitehtuurisuunnittelun. (Ramač ym., 2022.)

Pilvinatiivit järjestelmät hyödyntävät monipuolisesti pilvipalveluntarjoajien tarjoamia palveluita ja ohjelmointirajapintoja. Skaalautuvuuden ja elastisuuden saavuttamiseksi käytetään usein esimerkiksi kuormantasaus- ja viestinvälityspalveluita. Useimmat näistä palveluista eivät

kuitenkaan noudata olemassa olevia pilvistandardeja. Koska palvelut ovat toimittajakohtaisia, eivätkä ne ole yhteensopivia muiden pilvialustojen tai -palveluiden kanssa syntyy toimittajaloukko (engl. vendor lock-in). (Kratzke & Peinl, 2016.) Toimittajaloukolla tarkoitetaan tilannetta, jossa asiakas on sidottu yhteen pilvipalveluntarjoajaan eikä voi vaivattomasti vaihtaa toiseen ilman merkittäviä kustannuksia tai teknisiä ongelmia. Toimittajaloukko tulee ongelmaksi, kun organisaatio haluaa vaihtaa pilvipalveluntarjoajaa tai yhdistää useiden tarjoajien palveluita, mutta ei kykene siirtämään sovelluksiaan tai dataansa eri pilvipalveluiden välillä. (Opara-Martins ym., 2016.)

Teknisten haasteiden lisäksi pilvipalveluihin liittyy myös oikeudellisia kysymyksiä, koska pilvipalveluntarjoajan resurssit voivat sijaita eri maantieteellisillä alueilla. Toisistaan poikkeavat lait voivat aiheuttaa ongelmia käyttäjille, jos dataa siirretään sijaintiin, jossa on erilaiset tietoturva vaatimukset. Data voi myös sijaita samanaikaisesti useammassa paikassa, jolloin haasteena voi olla määrittää minkä maan lainsäädäntöä sovelletaan. Myös tilanne, jossa pilvipalveluntarjoajan laitteisto takavarikoidaan tietyn asiakkaan tutkintaan liittyen voi aiheuttaa riskin muiden asiakkaiden tietosuojalle. (Ali ym., 2015.)

3 Organisaatioiden oma tietojärjestelmäkehitys

3.1 Kehittämisen valmiuden arviointi ja sen hyödyt

Tietojärjestelmät ovat kokonaisuuksia, jotka on suunniteltu keräämään, käsittelemään, tallentamaan ja jakamaan tietoa tietyn tehtävän tai tavoitteen saavuttamiseksi. Vastaanotetun tiedon perusteella tietojärjestelmä tuottaa siitä hyödyllistä tietoa käytettäväksi eri tilanteissa. (Hinton, 2006.)

Tietojärjestelmä koostuu siihen kytkeytyvistä ihmisistä, laitteistosta, ohjelmistosta ja sovelluksista, prosesseista sekä tietojärjestelmässä käsiteltävästä datasta (Aouhassi & Hanoune, 2019).

Hung & Low (2008) tutkivat suurten australialaisten organisaatioiden päätöksentekoa ohjelmistojen kehittämisen ja valmiiden ohjelmistojen hankkimisen välillä. Vaikka organisaatioiden välillä esiintyikin vaihtelua päätöksentekoprosessin taustalla vaikuttavista tekijöistä, voitiin silti tunnistaa yhteisiä tekijöitä. Hung & Low jakavat tekijät kolmeen ryhmään: keskeiset tekijät, jotka voivat ratkaisevasti vaikuttaa päätökseen; toissijaiset tekijät, jotka täydentävät päätöksentekoa, mutta eivät yksin ratkaise sitä; sekä lisätekijät, jotka ovat merkittäviä vain joillekin organisaatioille. Taulukossa 1 esitetään tutkimuksessa havaitut keskeiset päätöksentekijät.

Taulukko 1 Päätöksentekoon vaikuttavat keskeiset tekijät

Tekijä	Kuvaus
Strategia	IT-strategia ohjaa päätöksiä. Usein ydinjärjestelmät kehitetään, muut järjestelmät ostetaan.
Kilpailuetu	Usein alalla vakiintuneet järjestelmät ostetaan ja kilpailuetua tuovat kehitetään itse.
Kypsyys	Ohjelmistojen ja teknologioiden kypsyys vaikuttavat ostamisen houkuttelevuuteen.
Sisäinen kyvykkyys	Organisaation omat resurssit ja kyvykkyys kehittää tietojärjestelmiä vaikuttavat valintaan.
Vaatimusten soveltuvuus	Liiketoimintavaatimusten ensisijaisuus ja niiden täytyminen vaikuttavat valintaan.

Organisaation ydintoimintoihin liittyviä järjestelmiä kehitetään usein itse, sillä valmiit ohjelmistot eivät tarjoa välttämättä organisaatioiden tarvitsemaa joustavuutta ja ainutlaatuisuutta. Kehittämällä itse liiketoiminnalle kriittiset järjestelmät, voidaan varmistaa järjestelmän räätälöinti vastaamaan tarkasti liiketoimintaprosesseja. Myös organisaatiot, joilla on vahvaa IT-osaamista, kokeneita kehittäjiä ja pitkä historia tietojärjestelmien kehittämisestä pystyvät hyödyntämään resurssejaan tehokkaasti kehittämällä omia ratkaisujaan. Organisaatiot hyötyvät vahvasta IT-osaamisestaan erityisesti, jos markkinoilla ei ole tarjolla riittävän kypsiä ohjelmistopaketteja tai valmiiden

ratkaisujen mukauttaminen olisi liian kallista tai aikaa vievää. Lisäksi kustannusnäkökulmat ohjaavat kehittämispäätöstä. Vaikka valmisohjelmistot ovat yleisesti edullisempia hankkia, saattaa niihin liittyä esimerkiksi räätälöintiin liittyviä piilokustannuksia. Itse kehitetty tietojärjestelmä voi suuremman alkuinvestoinnin jälkeen olla pitkäaikaisesti edullisempi, etenkin jos se vähentää riippuvuutta ulkoisista toimittajista ja mahdollistaa nopean reagoinnin muutoksiin. (Hung & Low, 2008.)

Olsen & Sætre (2007) tarkastelevat kuinka pienten ja keskisuurten yritysten kannattaa harkita omien tietojärjestelmiensä kehittämistä, sen sijaan että ne ostaisivat valmiita kaupallisia ratkaisuja. Itse rakennettujen järjestelmien suurimpana etuna on niiden joustavuus ja kyky mukautua yrityksen yksilöllisiin tarpeisiin erityisesti dynaamisissa liiketoimintaympäristöissä. Itse kehitetyssä järjestelmässä yritys voi automatisoida ja optimoida kriittisiä prosesseja tavalla, joka ei ole mahdollista valmiilla ratkaisuilla. Lisäksi itse kehitetyt järjestelmät tarjoavat organisaatioille täyden kontrollin niiden ydintoiminnoista, mikä mahdollistaa niiden mukauttamisen nopeasti muuttuviin asiakasvaatimuksiin ja markkinoiden olosuhteisiin. Valmiit kaupalliset ratkaisut sen sijaan voivat rajoittaa yrityksen kykyä sopeutua muutoksiin, koska ne perustuvat standardoituihin liiketoimintamalleihin. Olsen & Sætre tuovat myös esiin, miten järjestelmän kehittämisen aikana toteutettava yrityksen prosessien kartoitus ja analyysi tuovat jo itsessään lisäarvoa. Kehittäessä omaa tietojärjestelmää organisaatiot voivat tunnistaa nykyisten prosessiensa puutteita ja suoraviivaistaa toimintaa.

Fang ym. (2020) tuovat tutkimuksessaan esille, miten itse kehitettyjen tietojärjestelmien tuotto kasvavaa ajan myötä. Tämä johtuu erityisesti siitä, että organisaatiot kerryttävät käyttökokemusta ja kehitysosaamista. Kerrytetyn osaamisen ansiosta järjestelmiä voidaan jatkokehittää ja laajentaa organisaation liiketoiminnan kasvun ja strategisten muutosten rinnalla. Itse kehitetyt tietojärjestelmät tukevat myös organisaatioiden innovointikykyä, sillä järjestelmiä voidaan hyödyntää uusien liiketoimintamallien ja palveluiden kehittämisessä. Organisaation itse kehittämät tietojärjestelmät pysyvät usein myös relevantteina ja tehokkaina pitkällä aikavälillä.

IT-osaamisen tehokas hyödyntäminen on keskeistä organisaation kyvyille sopeutua muuttuviin liiketoimintaympäristöihin ja markkinatrendeihin. Erityisesti absorptiokyky, eli organisaation kyky hankkia, jalostaa ja hyödyntää tietoa, vahvistaa organisaation ketteryyttä. Organisaation ketteryys voidaan jakaa markkinoita hyödyntävään ketteryyteen ja operatiivisen sopeutumisen ketteryyteen. Markkinoita hyödyntävällä ketteryydellä tarkoitetaan organisaation kykyä reagoida nopeasti markkinamuutoksiin ja hyödyntää uusia kaupallisia mahdollisuuksia. Operatiivisilla sopeutumisella

tarkoitetaan sisäisten prosessien nopeaa mukauttamista vastaamaan ulkoisia vaatimuksia. Vahva IT-osaaminen voi tukea organisaation kilpailukykyä pitkällä aikavälillä tarjoamalla joustavuutta ja kykyä reagoida muutoksiin nopeasti. (Mao ym., 2021.)

3.2 Kehittämisen riskit

Tietojärjestelmäprojektien ongelmien, kuten budjetin tai aikataulun ylitysten taustalla on usein riskien puutteellinen ymmärtäminen, tunnistaminen tai hallinta. Tietojärjestelmäkehitykseen liittyvät riskit voidaan jakaa sosiaalisiin ja teknisiin alajärjestelmän riskeihin. Koska tietojärjestelmäkehitys tapahtuu osana sosiaalista ympäristöä, joka saattaa olla epävakaa tai altis poliittisille ristiriidoille, voi kehitysprojektin vaatima sitoutuvuus tai resurssien saatavuus heikentyä. Myös tietojärjestelmän käyttäjät saattavat vastustaa muutosta tai osoittaa puutteellista sitoutumista. Sosiaaliset riskit voidaan jakaa edelleen organisaatioympäristöön liittyviin tekijöihin sekä käyttäjäryhmien asenteisiin ja toimintaan. Teknisen alajärjestelmän riski kuvaa tietojärjestelmäkehitykseen liittyvää haastetta, jossa monimutkainen tekninen kokonaisuus rakennetaan tiettyjen vaatimusten pohjalta. Jos vaatimukset ovat epäselviä tai alttiita muutoksille, kokonaisuuden toteuttaminen voi olla huomattavasti vaikeampaa. Lisäksi teknologian tuntemattomuus tai uuden teknologian hyödyntäminen voi kasvattaa projektin teknistä monimutkaisuutta. Teknisen alajärjestelmän riskejä voidaan tarkastella kahden riskidimension kautta: projektin vaatimusten selkeydestä ja projektin monimutkaisuudesta. Sosiaaliset ja tekniset riskit vaikuttavat myös toisiinsa, sillä esimerkiksi epävakaa toimintaympäristössä on haastavaa tuottaa selkeitä vaatimuksia tietojärjestelmälle. (Wallace ym., 2004.)

Teknisen alajärjestelmän riskit vaikuttavat erityisesti sisäisiin tietojärjestelmäprojekteihin, jossa niiden vaikutus järjestelmän suorituskykyyn korostuu ulkoistettuihin projekteihin verrattuna. Organisaatio joutuu sisäisissä projekteissa vastaamaan itse teknisten järjestelmien monimutkaisuuden hallinnasta ja sisäisten resurssien, kuten asiantuntemuksen ja työvoiman tehokkaasta käytöstä. Erityisesti strategisesti merkittävässä kehityshankkeissa tekniset haasteet voivat korostua projektin toimintaympäristön ja teknisten vaatimusten muuttuessa. (Liu & Wang, 2014.)

Projektin onnistumista arvioitaessa yksi keskeinen tekijä on, pystytäänkö se toteuttamaan suunnitellun budjetin puitteissa. Budjetin ylitykset voivat merkittävästi heikentää organisaation mahdollisuuksia saavuttaa taloudellista hyötyä projektista ja johtaa pahimmillaan organisaation konkurssiin. Budjetin hallinnan merkitys korostuu erityisesti suurissa IT-projekteissa, joissa kustannusten ylitykset voivat olla huomattavia ja vaikeasti ennustettavia. Tilastollisesti tarkasteltuna IT-projektien budjetin ylitykset eivät aina noudata normaalia jakaumaa, vaan saattavat noudattaa

potenssilakijakaumaa, jossa suuret ylitykset ovat odotettua yleisempiä. Jos projektinhallinnassa ei huomioida tätä mahdollisuutta, organisaatiot voivat aliarvioida suurten ylitysten todennäköisyyden ja altistua huomattaville taloudellisille riskeille. (Flyvbjerg ym., 2022.)

4 Pilvinatiivin kehittämisen liiketoiminnalliset hyödyt

4.1 Kustannustehokkuus

Kustannukset julkisessa pilvessä sekä pilvipalveluiden hinnoittelumallit ovat monimutkaisempia kuin toimittaessa perinteisessä konesali-infrastruktuurissa. Pilvipalveluntarjoajilla on käytössä yli 30 hinnoittelumallia. Yleisin hinnoittelumalli julkisessa pilvessä on käytön mukaan tapahtuva laskutus, jossa resurssit kuten laskentateho varataan asiakkaan käyttöön tietyn maksetun ajan ajaksi. (Ramesh ym., 2021.)

Kustannuksia julkisen pilven ja paikallisten palveluiden välillä voidaan vertailla pääomakulujen, juoksevien kulujen ja omistuksen kokonaiskustannusten avulla. Julkisen pilven käytössä pääomakustannukset pienenevät, sillä organisaatiot eivät omista enää fyysistä infrastruktuuria, vaan hyödyntävät pilvipalveluntarjoajan omistamia resursseja. (Kilcioglu ym., 2017.) Perinteisesti IT-investoinnit ovat vaatineet merkittäviä alkuinvestointeja laitteisiin ja infrastruktuuriin, mikä sitoo pääomaa ja vaatii pitkäaikaista budjetointia. Julkisessa pilvessä nämä alkuinvestoinnit eivät ole tarpeen, jolloin IT-kustannuksia voidaan siirtää pääomamenoista juokseviin kuluihin. Juoksevat kulut voidaan kohdistaa suoraan tuloja vasten, mikä parantaa kassavirtaa ja lisää taloudellista joustavuutta. (Senarathna ym., 2015.)

Koska julkista pilveä käyttäessä infrastruktuuri ei ole omassa hallinnassa, on juoksevien kulujen kustannusrakenne myös erilainen verrattuna paikallisiin palveluihin. Pilvipalveluissa juoksevat kulut perustuvat yleisesti käyttöaikaan, jolloin organisaatio maksaa esimerkiksi laskentatehosta, tallennustilasta ja datan siirrosta tarpeen mukaan. Paikallisissa palveluissa juokseviin kuluihin kuuluvat puolestaan infrastruktuurin ylläpitoon liittyvät kustannukset. Näitä ovat esimerkiksi sähkö, jäähdytys, laitteiston ylläpito sekä mahdolliset lisenssimaksut. Kun pilvipalveluiden hinnoittelu mahdollistaa kustannusten skaalautuvuuden ja joustavuuden, on taas paikallisten palveluiden kulurakenne usein vakaampi, sillä laitteisto on jo hankittu ja ylläpitokustannuksia pystytään ennustamaan suhteellisen helposti. (Rajamohan & Settlege, 2020.) Erityisesti pienten ja keskisuurten yritysten kohdalla juoksevat kulut ovat usein pienemmät käytettäessä julkista pilveä verrattuna oman datakeskuksen ylläpitoon. Julkisen pilven hyödyntäminen poistaa organisaatioilta datakeskusten ylläpitoon ja hallinnointiin liittyvät kulut. Pilvipalveluiden skaalautuvuus mahdollistaa myös resurssien tehokkaamman käytön, kun ylimääräistä kapasiteettia ei tarvitse varata huippukuormituksia varten. (Abdulle ym., 2022.)

Juoksevien kulujen kustannussäästöjen määrä riippuu kuitenkin käyttötapauksista ja työkuormien luonteesta. Erityisesti joustavuutta vaativissa ja vaihtelevissa työkuormissa pilvipalvelut tarjoavat usein kustannustehokkaamman vaihtoehdon. Sen sijaan ennakoitavissa oleville ja tasaisille käyttökuormille paikalliset ratkaisut voivat olla kustannustehokkaampia. Julkisessa pilvessä datan siirtokustannukset ja integrointikustannukset voivat nostaa kokonaiskuluja etenkin pitkäkestoisissa projekteissa tai suurissa datavolyymeissa. (Rajamohan & Settlege, 2020.)

Tutkimukset (ks. esim. Johnson ym., 2019; Chahal ym., 2021; Juhasz, 2021; Abdulle ym., 2022; Abdulmohson ym., 2022) osoittavat, että julkisen pilven hyödyntäminen mahdollistaa alhaisemmat omistamisen kokonaiskustannukset etenkin lyhyellä ja keskipitkällä aikavälillä. Paikallisessa infrastruktuurissa kustannustehokkuutta saavutetaan, jos sen käyttöaste pysyy jatkuvasti korkeana. Myös arkkitehtuurivalinnoilla on vaikutusta omistamisen kokonaiskustannuksiin. Palvelimettomilla arkkitehtuureilla voidaan saavuttaa kustannussäästöjä, kun työkuormat ovat piikkimäisiä ja tapahtumakohtaisia. (Chahal ym., 2021.)

Pilvipalveluiden joustavuus ja käytön mukaan tapahtuva laskutus voi kuitenkin johtaa kustannusten nopeaan nousuun, jos resursseja ei hallita tehokkaasti. FinOps (engl. financial operations) on viitekehys, joka keskittyy pilvikustannusten ja resurssien hallintaan. FinOps painottaa aktiivisia hallintastrategioita, kuten resurssien käytön jatkuvaa seuraamista ja tarkkaa ennustamista. FinOps:in avulla organisaatiot voivat tasapainottaa kustannusten minimoinnin ja suorituskyvyn ylläpidon muuttuvissa pilviympäristöissä. (Smendowski & Nawrocki, 2024.)

4.2 Teknologiset tekijät

4.2.1 Skaalautuvuus ja elastisuus

Duboc ym. (2007) määrittelevät skaalautuvuuden tietojärjestelmäkehityksessä ohjelmistojärjestelmän laaduksi, joka kuvastaa miten hyvin järjestelmä pystyy sopeutumaan ympäristönsä ja rakenteensa muuttumiseen säilyttäen samalla toivotut suorituskykyominaisuudet. Pilvipalveluiden kohdalla skaalautumisella viitataan järjestelmän kykyyn ylläpitää resurssien käyttö tasapainoisena ja tehokkaana riippumatta käsiteltävän datan määrästä (Rana ym., 2019). Paikallisissa palveluissa skaalaaminen ylöspäin vaatii merkittäviä investointeja laitteistoon sekä sen ylläpitoon (Man & Huh, 2013). Yllättävät kysyntäpiikit voivat johtaa järjestelmän kaatumiseen tai aiheuttaa pullonkauloja sen tehokkuudessa. Kysyntäpiikkeihin voidaan varautua paikallisissa palveluissa ylimitoittamalla IT-infrastruktuurin resursseja. Tämä voi kuitenkin johtaa resurssien

tehottomaan käyttöön ja korkeampiin operatiivisiin kustannuksiin, kun kysyntä ei ole yhtä suurta. (Wease ym., 2018.)

Joustavuudella tarkoitetaan pilvipalveluiden kykyä dynaamisesti lisätä tai vähentää infrastruktuurin resursseja kysynnän vaihteluiden mukaisesti. Joustavuus mahdollistaa pilvipalveluiden resurssitehokkaan hyödyntämisen ja vähentää yli- ja alikapasiteetin kustannuksia. Pilven joustavuuden ansiosta pilvinatiivit sovellukset pystyvät minimoimaan kysynnän ja tarjonnan välisen eron, jolloin käytettävissä oleva kapasiteetti vastaa tarkoin todellista kuormitusta. Automatisoitujen mekanismien avulla resursseja voidaan provisoida ja vapauttaa dynaamisesti. (Barnawi ym., 2020.)

Pilvessä skaalautuminen voidaan toteuttaa vertikaalisesti ja horisontaalisesti. Vertikaalisessa skaalauksessa resursseja kuten prosessointitehoa tai muistia lisätään tai vähennetään yksittäiselle virtuaalikoneelle. Tämä mahdollistaa korkeamman suorituskyvyn työkuormille, jotka hyötyvät yksittäisen palvelimen suuremmasta laskentatehosta. Horisontaalisessa skaalauksessa puolestaan infrastruktuuriin lisätään uusia virtuaalikoneita, mikä mahdollistaa työkuorman jakamisen useampien palvelimien kesken. Horisontaalisen ja vertikaalisen skaalautuvuuden käyttö yhdessä voi tarjota merkittäviä kustannussäästöjä verrattuna pelkästään yhden skaalautuvuustyypin käyttöön. (Gandhi ym., 2018.)

4.2.2 Innovointi ja nopeampi markkinoille pääsy

Pilvinatiivi kehittäminen mahdollistaa liiketoiminnan ketteryyden parantamisen ja nopeamman markkinoille pääsyn. Tietojärjestelmien kehittämisessä voidaan hyödyntää pilvipalveluntarjoajien valmiiksi käyttöön otettavia pilvipohjaisia resursseja, jotka ovat käytettävissä tarpeen mukaan ja tukevat muuttuvia liiketoimintatarpeita nopeammin kuin perinteisten IT-resurssien hankinta, asennus ja ylläpito. (Sahandi ym., 2012.) Pilvipalveluntarjoajat investoivat myös jatkuvasti uusien teknologioiden tutkimukseen ja kehittämiseen (Thakur & Jha, 2023). Lisäksi ne mahdollistavat myös kolmannen osapuolen kehittäjille ja ohjelmistojen tarjoajille palveluiden myynnin alustoillaan (Shang & Kauffman, 2020). Pilvinatiivin kehittämisen myötä organisaatioilla on mahdollisuus hyödyntää nousevia teknologioita, kuten tekoälyä, koneoppimista tai massa data-analytiikka matalammalla kynnyksellä ja aikaisessa vaiheessa (Thakur & Jha, 2023).

Myös DevOpsin käyttöönotto tukee pilvinatiivien arkkitehtuurien tarjoamia hyötyjä nopeammasta markkinoille pääsystä ja liiketoiminnan ketteryyden parantamisesta. DevOpsin avulla organisaatiot voivat lyhentää järjestelmien uusien ominaisuuksien ja ohjelmistomuutosten julkaisemiseen kuluvaan aikaa sekä reagoida nopeasti asiakastarpeisiin ja markkinoiden muutoksiin. Pilvipalveluiden ja

DevOpsin yhteiskäyttö tarjoaa yrityksille mahdollisuuden kehittää ketterästi uusia tuotteita ja palveluita sekä testata ja ottaa käyttöön innovaatioita, jotka parantavat kilpailukykyä ja liiketoiminnan suorituskykyä. (Mukange & Langerman, 2022.)

Abbasin & Walian (2024) toteuttamassa tutkimuksessa havaittiin, että pilvipalveluiden ja DevOpsin käytön yhdistämisellä saatiin IoT-ratkaisujen käyttöönottoaikaa vähennettyä 71,67 % perinteisiin ohjelmistokehitysmenetelmiin verrattuna. CI/CD menetelmien avulla koodimuutokset voitiin yhdistää jatkuvasti tietovarastoon ja testata automaattisesti ennen tuotantoon viemistä. Tämä vapautti kehittäjät keskittymään tehokkaammin uusien ominaisuuksien luomiseen ja virheiden korjaamiseen. Lisäksi pilvipalveluiden ja DevOps-menetelmien yhdistäminen mahdollisti reaaliaikaisen data-analytiikan hyödyntämisen, mikä tehosti IoT-järjestelmien reagoitukykyä. Reaaliaikaisen analyysin ansiosta reagointiajat lyhenivät 91,67 %, jolloin järjestelmät pystyivät vastaamaan muutoksiin ja ongelmiin nopeasti ilman pitkiä viiveitä.

4.2.3 Resilienssi, saatavuus ja tietoturva

Pilvinatiivit järjestelmät on suunniteltu sietämään pilviympäristön vaihtelevia olosuhteita ja ajoittaisia resurssihäiriöitä. Järjestelmät osaavat käsitellä odottamattomia suorituskyvyn heilahteluja ja resurssikatkoja ilman, että ne heikentävät palvelun laatua tai saatavuutta. Sovelluksen komponentteja voidaan jakaa eri vikasietoisuusalueille, kuten saatavuusvyöhykkeille tai datakeskuksiin, mikä mahdollistaa komponenttien nopean uudelleenkäynnistyksen vikatilanteissa. (Toffetti ym., 2017.) Myös automaatiota voidaan hyödyntää sekä vikojen havaitsemiseen, että niiden korjaamiseen. Valvonta- ja metriikkajärjestelmien avulla järjestelmän tilaa voidaan analysoida jatkuvasti reaaliaikaisen datan avulla. Ongelmien ilmetessä automaatiojärjestelmät voivat käynnistää ennakkoon määritellyt toimenpiteet, kuten resurssien lisäämisen tai viallisen palvelun uudelleenkäynnistämisen. Automaation avulla voidaan vähentää inhimillisen puuttumisen tarvetta ja nopeuttaa ongelmien ratkaisemista. (Deng ym., 2024.)

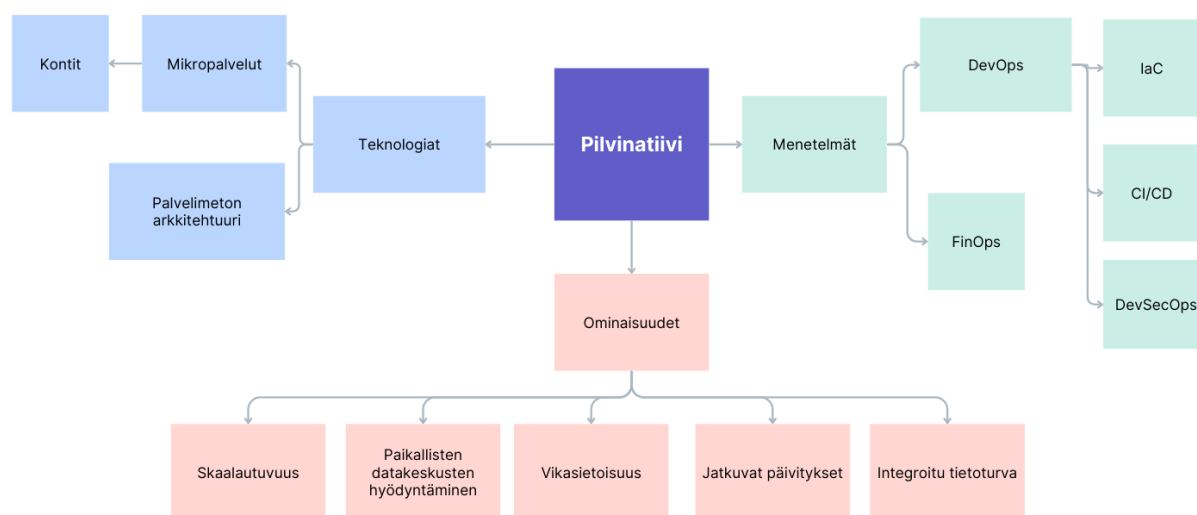
Automaatiota voidaan hyödyntää myös tietoturvan parantamiseen pilvinatiiveissa järjestelmissä. DevSecOpsissa turvallisuuskäytännöt ja -kontrollit yhdistetään DevOpsin jatkuvaan toimitus- ja kehitysprosessiin. Automatisoitujen tietoturvatarkistusten lisäämisen CI/CD-putken vaiheisiin mahdollistaa jatkuvan turvallisuusanalyysin, jolloin voidaan vähentää turvallisuusongelmien esiintymistä myöhemmissä kehitysvaiheissa. (Caniglia ym., 2025.) Staattinen sovelluksen tietoturvatestausta (engl. static application security testing, SAST) ja dynaaminen sovelluksen tietoturvatestausta (engl. dynamic application security testing, DAST) ovat keskeisiä osia DevSecOpsin turvallisuusprosessissa. SAST analysoi koodin haavoittuvuuksia kehitysvaiheessa ja

se voidaan integroida CI/CD-putkeen. DAST testaa sovelluksen ajonaikaisia haavoittuvuuksia. Sen täysi automatisointi on haastavampaa, sillä ajonaikaisessa ympäristössä on tarpeen simuloida realistisia hyökkäysvektoreita, joita on hankala ennakoida automaatiotyökaluilla. (Naidoo & Möller, 2022.)

5 Yhteenveto ja johtopäätökset

Tässä tutkielmassa perehdyttiin pilvinatiiviin kehittämiseen ja sen organisaatioille tuomiin liiketoimintaetuihin. Pilvipalveluita ja niiden teknologiaa on tutkittu tietotekniikan alan kirjallisuudessa runsaasti. Kauppatieteiden alan kirjallisuudessa niiden tutkimus on kuitenkin melko rajallista. Tässä tutkielmassa pyritään täyttämään tätä tutkimusaukkoa ja ennen kaikkea tuomaan pilvipalveluita lähestyttävämmäksi niin kauppatieteiden opiskelijoille kuin sen parissa työskenteleville.

Tutkielman ensimmäinen tutkimuskysymys on mitä on pilvinatiivi kehittäminen, johon vastattiin ensimmäisessä luvussa. Alan kirjallisuudessa ei termille ole yksiselitteistä määritelmää, mutta tutkimuksessa korostuu yleinen kutakuinkin yhtenäinen käsitys sen merkityksestä. Yleisesti pilvinatiivi termillä tarkoitetaan modernien pilvipalveluympäristöjen tarjoamien etujen maksimaalista hyödyntämistä. Tämä ei kuitenkaan tarjoa kattavaa kuvaa pilvinatiivista, joten tässä tutkielmassa pyrittiin tarkastelemaan pilvinatiivia kehittämistä siihen liittyvien teknologisten ratkaisujen, viitekehysten ja pilvinatiiveille järjestelmille yhteisten ominaisuuksien kautta. Näitä havainnollistetaan kuvassa 3.



Kuva 3 Pilvinatiivi kehittäminen

Pilvinatiivit järjestelmät hyödyntävät yleisesti moderneja sovellusarkkitehtuureja, kuten mikropalveluarkkitehtuuria tai palvelimeton arkkitehtuuria. Keskeisenä erona perinteisiin monoliittisiin sovelluksiin on, että ohjelmistoja ei rakenneta enää yhteen koodipohjaan vaan niissä

hyödynnetään laajasti itsenäisesti käyttöön otettavia palveluita. Kehittämisessä hyödynnetään myös moderneja paradigmoja kuten DevOpsia. Kehittämisessä pyritään myös hyödyntämään automatisaatiota julkaisujen viiveiden minimoimiseksi ja resurssitehokkuuden parantamiseksi. Infrastrukturi koodina (engl. infrastructure as code, IaC) lähestymistavalla infrastruktuurin määrittelyä hallitaan koodina ja resurssien hallintaa pystytään automatisoimaan. Jatkuvan integraation (engl. continuous integration, CI) ja jatkuvan toimituksen (engl. continuous delivery, CD) tarkoituksena on automatisoida ohjelmiston toimitusprosesseja, kuten testausta, integrointia ja käyttöönottoa. DevSecOps puolestaan on viitekehys, jossa tietoturva otetaan jatkuvaksi osaksi ohjelmistokehitysprosessia. Pilvinatiivi kehittäminen voidaankin täten käsittää kokoelmaksi moderneja teknologioita, sovellusarkkitehtuurisia ratkaisuja sekä parhaita käytäntöjä pilvestä saatavien hyötyjen maksimoimiseksi.

Tutkielman toinen tutkimuskysymys oli: Mitä liiketoiminnallista hyötyä pilvinatiivista kehityksestä on organisaation omassa tietojärjestelmäkehityksessä? Oman tietojärjestelmäkehityksen hyötyjä ja riskejä käsitellään tutkielman kolmannessa luvussa ja samalla alustetaan viitekehystä pilvinatiivin kehittämisen hyötyjen tarkastelulle. Itse tutkimuskysymykseen vastattiin neljännessä luvussa.

Tutkielmassa havaittiin, että organisaatiot, joilla on vahvaa IT-osaamista kehittävät usein liiketoiminnalleen kriittisiä järjestelmiä itse. Tällä pyritään saavuttamaan joustavuutta, parempaa mukautuvuutta organisaation prosesseihin ja kilpailuetua erityisesti silloin, kun markkinoilta ei löydy riittävän kypsiä tai sopivia vaihtoehtoja. Itse kehitetyt järjestelmät tarjoavat pitkän aikavälin kustannushyötyjä ja tukevat innovointia. Päätökset valmiin ohjelmiston ostamisen ja kehittämisen välillä perustuvat strategisiin, teknologisiin ja taloudellisiin tekijöihin.

Tietojärjestelmien kehittämiseen itse liittyy myös riskejä, joita voidaan tarkastella sosiaalisten ja teknisten riskikategorioiden kautta. Sosiaaliset riskit liittyvät organisaation ja käyttäjien sitoutumiseen, kun taas tekniset riskit johtuvat esimerkiksi kehitettävän järjestelmän monimutkaisuudesta, vaatimusten epäselvyydestä tai uuden teknologian hyödyntämisestä. Näiden riskien hallinta on organisaatioille tärkeää, koska budjetin ja aikataulun ylitykset voivat vaarantaa projektin onnistumisen ja organisaation taloudellisen aseman. Riskien tunnistaminen ja hallinta onkin keskeistä tietojärjestelmäprojektien menestykselle.

Kun organisaatio päättää lähteä kehittämään omaa tietojärjestelmää, tarjoavat pilvipalvelut kiinnostavan vaihtoehdon paikallisten palveluiden rinnalle. Tutkielmassa havaittiin, että näiden järjestelmien kehittäminen pilvinatiivein menetelmin tuo mukanaan useita liiketoiminnallisia etuja,

mutta pilvinatiiviin kehittämiseen liittyy myös haasteita, joita organisaatiot joutuvat punnitsemaan. Pilvinatiivin kehittämisen liiketoimintahyötyjä ja haasteita havainnollistetaan taulukossa 2.

Taulukko 2 Pilvinatiivin kehittämisen liiketoimintahyödyt ja haasteet

Liiketoimintahyödyt	Haasteet
Kustannustehokkuus: Ei tarvita alkuinvestointeja laitteistoon ja sen ylläpitoon. Kustannukset myös skaalautuvat käytön mukaan.	Kustannusten hallinta: Käytön mukaan laskutus voi johtaa myös haasteisiin kustannusten hallinnassa.
Skaalautuvuus ja elastisuus: Pilvinatiivit järjestelmät mukautuvat nopeasti kysynnän vaihteluihin ja mahdollistavat resurssien optimoinnin.	Kompleksisuus: Mikropalvelut ja palvelimettomat arkkitehtuurit tekevät järjestelmistä monimutkaisempia hallita.
Nopeampi markkinoille pääsy: Pilvinatiivi kehittäminen nopeuttaa kehityssyklejä, mikä mahdollistaa nopeamman reagoinnin markkinoihin.	Tekninen velka: Nopeat kehityssykliit voivat johtaa teknisen velan syntymiseen, joka nostaa ylläpitokustannuksia ja aiheuttaa haasteita tulevaisuudessa.
Innovointi: Uusien teknologioiden, kuten tekoälyn ja koneoppimisen käyttöönotto on mahdollista matalalla kynnyksellä.	Toimittajaloukko: Palveluiden riippuvuus yksittäisistä pilvitarjoajista voi rajoittaa siirtymistä muihin alustoihin.
Resilienssi: Järjestelmät on suunniteltu sietämään häiriöitä ja takaamaan jatkuva toiminta.	Tietoturva: Pilvipalveluiden tietoturvakäytännöt eivät ole aina täysin organisaation hallinnassa.

Tutkimuksessa pilvinatiivin kehittämisen keskeiseksi hyödyksi havaittiin pilvipalveluiden palvelumallista johtuvat alhaiset käyttöönottokustannukset. Organisaatioiden ei tarvitse investoida omaan laitteistoon ja sen ylläpitoon, vaan pilvipalvelut voidaan ottaa käyttöön ilman suuria alkuinvestointeja. Pilvipalvelut mahdollistavat täten pääomakulujen siirtämisen juokseviin kuluihin, mikä tarjoaa taloudellista joustavuutta organisaatioille. Kustannusrakenne eroaa myös pilvipalveluita hyödynnettäessä verraten paikallisiin palveluihin. Kun paikallisissa palveluissa organisaatiot maksavat omasta laitteistostaan maksimikapasiteetin mukaan, vaikka todellinen kulutus olisikin vain pieni osa tästä, mahdollistavat pilvipalvelut laskutuksen käytön mukaan.

Järjestelmien kehittäminen pilvinatiivisti mahdollistaa myös alhaisemmat omistamisen kokonaiskustannukset. Tämä on kuitenkin vahvasti sidoksissa järjestelmän luonteeseen ja siihen kohdistuvaan työkuormaan. Pilvipalvelut tarjoavat usein kustannustehokkaamman vaihtoehdon vaihtelevien ja piikkimäisten kuormien käsittelyyn, kun taas tasaisten ja ennustettavien työkuormien suorittaminen voi olla kustannustehokkaampaa paikallisessa infrastruktuurissa. Kustannusten hallinta voi muodostua myös haasteeksi pilvipalveluiden monimutkaisemman rakenteen takia. FinOps viitekehys tarjoaa organisaatiolle työkaluja resurssien optimoinnin ja kustannusten tasapainottamisen avuksi. Kustannusten näkökulmasta pilvinatiivi kehittäminen vapauttaakin

organisaatiot infrastruktuurin hankintaan ja ylläpitoon liittyvistä kuluista, mutta pitkän aikavälin kustannussäästöt vaativat tarkempaa suunnittelua ja vahvaa osaamista.

Pilven joustavuuden hyödyntäminen tarjoaa organisaatiolle resurssitehokkaan tavan varautua kysynnän vaihteluihin. Kysyntäpiikkeihin ei tarvitse varautua ylimitoittamalla resursseja, vaan pilvinatiivit järjestelmät lisäävät ja vähentävät resursseja dynaamisesti. Organisaatioiden ei tarvitse enää investoida uuteen laitteistoon kysynnän kasvaessa, sillä esimerkiksi IaC:n avulla resursseja voidaan provisoida automaattisesti. Joustavuus tuo myös kilpailuetua kysynnän laskiessa merkittävästi, kun IT-kuluja saadaan karsittua nopeasti skaalaamalla resursseja alaspäin.

Pilvinatiivit järjestelmät on myös suunniteltu kestämään vaihtelevia olosuhteita ja resurssihäiriöitä, joka voi näkyä parempana asiakastyytyväisyytenä, kun järjestelmät eivät kaadu riittämättömien resurssien vuoksi. Vikojen havaitsemista, korjaamista ja tietoturvaa voidaan myös automatisoida, jolloin ongelmiin pystytään puuttumaan nopeammin ja vähentämään järjestelmän käyttökatkoja.

Pilvinatiivi kehittäminen mahdollistaa myös nopeammat kehityssykliä ja markkinoille pääsyn organisaatioille. Kehittämisessä voidaan hyödyntää pilvipalveluntarjoajien valmiita palveluita, jolloin järjestelmää ei tarvitse rakentaa alusta alkaen itse. Myös uusien teknologioiden, kuten koneoppimisen tai massa data-analytiikan implementointi on mahdollista matalammalla kynnyksellä. DevOpsin avulla organisaatiot voivat myös automatisoida kehitysprosessin vaiheita, kuten testaamista ja integrointia, jolloin muutokset järjestelmään saadaan tuotantoon nopeammin. Nopeat kehityssykliä voivat kuitenkin johtaa teknisen velan syntymiseen, joka voi aiheuttaa ylläpito- ja kehityskustannusten kasvua. Kehityksen nopeuttamiseksi tehdyt kompromissit voivat loppupeleissä viivästyttää järjestelmän toimitusta. Myös valmiiden palveluiden hyödyntäminen voi aiheuttaa toimittajaloukon, jossa organisaatiot eivät voi halutessaan vaihtaa toiseen pilvipalveluntarjoajaan ilman merkittäviä kustannuksia ja teknisiä ongelmia.

Tietoturva tuo myös omat haasteensa pilvinatiiveissa järjestelmissä. Pilvipalveluiden luonteen takia organisaatiot luovuttavat osan vastuusta pilvipalveluntarjoajille. Pilvinatiivien järjestelmien kompleksisuus altistaa ne myös moninaisille tietoturvaohuille. Toisaalta pilvinatiivi kehittäminen tarjoaa myös organisaatiolle uusia työkaluja tietoturvasta huolehtimiseen. Tietoturvapäivityksiä voidaan automatisoida ja DevSecOpsin avulla tietoturva voidaan ottaa kiinteäksi osaksi kehitysprosessia.

Tutkimuksessa havaitut liiketoimintaedut edesauttavat organisaatioiden taloudellista ja teknistä ketteryyttä. Pilvinatiivi kehittäminen ei ole kuitenkaan vain teknologiavalinta, vaan vaatii myös organisaation prosessien ja toimintatapojen mukauttamista. Strateginen lähestymistapa on tarpeen,

jotta pilvinaltiivin kehittämisen hyödyt saadaan tukemaan organisaation liiketoimintastrategiaa. Myös riskien hallinta ja niiden toteutumisen todennäköisyyden arviointi auttaa organisaatiota hyödyntämään pilvipalveluita parhaalla mahdollisella tavalla.

Lähteet

- Abbas, S. I., & Walia, R. (2024). *Analysis of Revolutionizing IoT Solutions through Cloud Computing Integration*. 39–45. <https://doi.org/10.1109/ICSADL61749.2024.00012>
- Abdulle, A. A., Ali, A. F., & Abdullah, R. H. (2022). Cost-Benefit Analysis of Public Cloud Versus In-House Computing. *International Journal of Engineering Trends and Technology*, 70(6), 300–307. <https://doi.org/10.14445/22315381/IJETT-V70I6P231>
- Abdulmohson, A., Kadhim, M. F., Hussein Anssari, O. M., & Al-Jobouri, A. A. (2022). Cost analysis of on-premise versus cloud-based implementation of Moodle in Kufa University during the pandemic. *Indonesian Journal of Electrical Engineering and Computer Science*, 25(3), 1787–1794. <https://doi.org/10.11591/ijeecs.v25.i3.pp1787-1794>
- Ali, A., Laghari, A. A., Kandhro, I. A., Kumar, K., & Younus, S. (2024). Systematic analysis of on-premise and cloud services. *International Journal of Cloud Computing*, 13(3), 214–242. <https://doi.org/10.1504/IJCC.2024.139604>
- Ali, M., Khan, S. U., & Vasilakos, A. V. (2015). Security in cloud computing: Opportunities and challenges. *Information Sciences*, 305, 357–383. <https://doi.org/10.1016/j.ins.2015.01.025>
- Aouhassi, S., & Hanoune, M. (2019). Specific Qualification for Information System Components from Managers and Technical Staff Perspective. *Teoksessa Studies in Big Data (Vsk. 53, ss. 40–49)*. https://doi.org/10.1007/978-3-030-12048-1_6
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. <https://doi.org/10.1145/1721654.1721672>
- Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture. *IEEE Software*, 33(3), 42–52. <https://doi.org/10.1109/MS.2016.64>
- Barnawi, A., Sakr, S., Xiao, W., & Al-Barakati, A. (2020). The views, measurements and challenges of elasticity in the cloud: A review. *Computer Communications*, 154, 111–117. <https://doi.org/10.1016/j.comcom.2020.02.010>
- Blinowski, G., Ojdowska, A., & Przybyłek, A. (2022). Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation. *IEEE Access*, 10, 20357–20374. <https://doi.org/10.1109/ACCESS.2022.3152803>
- Brunner, S., Blöchlinger, M., Toffetti, G., Spillner, J., & Bohnert, T. M. (2015). Experimental Evaluation of the Cloud-Native Application Design. *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, 488–493. <https://doi.org/10.1109/UCC.2015.87>

- Caniglia, A., Dentamaro, V., Galantucci, S., & Impedovo, D. (2025). FOBICS: Assessing project security level through a metrics framework that evaluates DevSecOps performance. *Information and Software Technology*, 178. <https://doi.org/10.1016/j.infsof.2024.107605>
- Chahal, D., Mishra, M., Palepu, S., & Singhal, R. (2021). *Performance and cost comparison of cloud services for deep learning workload*. 49–55. <https://doi.org/10.1145/3447545.3451184>
- Chernyshev, M., Baig, Z., & Zeadally, S. (2021). Cloud-Native Application Security: Risks, Opportunities, and Challenges in Securing the Evolving Attack Surface. *Computer*, 54(11), 47–57. <https://doi.org/10.1109/MC.2021.3076537>
- CNCF (2024) FAQ. <<https://www.cncf.io/about/faq/>> haettu 18. syyskuuta 2024.
- Deka, G. C., & Das, P. K. (2014). An overview on the virtualization technology. Teoksessa *Handbook of Research on Cloud Infrastructures for Big Data Analytics* (ss. 289–321). <https://doi.org/10.4018/978-1-4666-5864-6.ch012>
- Deng, S., Zhao, H., Huang, B., Zhang, C., Chen, F., Deng, Y., Yin, J., Dustdar, S., & Zomaya, A. Y. (2024). Cloud-Native Computing: A Survey From the Perspective of Services. *Proceedings of the IEEE*, 112(1), 12–46. <https://doi.org/10.1109/JPROC.2024.3353855>
- Duboc, L., Rosenblum, D., & Wicks, T. (2007). *A framework for characterization and analysis of software system scalability*. 375–384. <https://doi.org/10.1145/1287624.1287679>
- Fang, J., Liu, X., & Qu, W. G. (2020). The changing returns on IT investment. *Industrial Management and Data Systems*, 120(11), 2025–2039. <https://doi.org/10.1108/IMDS-04-2020-0234>
- Flexera (2024) State of the cloud report. <https://resources.flexera.com/web/pdf/Flexera-State-of-the-Cloud-Report-2024.pdf?elqTrackId=7adb640823d641a8bb962034503b3e20&elqaid=7675&elqat=2&elqak=8AF52A3AEA6F01ED04922E8C44DC905D36C8C3ECC4F95CBC0871736C365AB00E7C39&_gl=1*_ljss0o*_gcl_au*MjAxMTY2NzYyNS4xNzI2MzE2NjM1> haettu 14. syyskuuta 2024.
- Flyvbjerg, B., Budzier, A., Lee, J. S., Keil, M., Lunn, D., & Bester, D. W. (2022). The Empirical Reality of IT Project Cost Overruns: Discovering A Power-Law Distribution. *Journal of Management Information Systems*, 39(3), 607–639. <https://doi.org/10.1080/07421222.2022.2096544>
- Gandhi, A., Dube, P., Karve, A., Kochut, A., & Zhang, L. (2018). Model-driven optimal resource scaling in cloud. *Software and Systems Modeling*, 17(2), 509–526. <https://doi.org/10.1007/s10270-017-0584-y>
- Gannon, D., Barga, R., & Sundaresan, N. (2017). Cloud-Native Applications. *IEEE Cloud Computing*, 4(5), 16–21. IEEE Cloud Computing. <https://doi.org/10.1109/MCC.2017.4250939>

- Gartner (2023) *4 Exciting New Trends in the Gartner Emerging Technologies Hype Cycle*. <<https://www.gartner.com/en/articles/what-s-new-in-the-2023-gartner-hype-cycle-for-emerging-technologies>>, Haettu 28. marraskuuta 2024.
- Gokarna, M., & Singh, R. (2021). *DevOps: A Historical Review and Future Works*. 366–371. <https://doi.org/10.1109/ICCCIS51004.2021.9397235>
- Gunja, Saif (2024) What is DevOps? Unpacking the purpose and importance of an IT cultural revolution. <<https://www.dynatrace.com/news/blog/what-is-devops/>> Haettu 28. marraskuuta 2024.
- Hardikar, S., Ahirwar, P., & Rajan, S. (2021). *Containerization: Cloud Computing based Inspiration Technology for Adoption through Docker and Kubernetes*. 1996–2003. <https://doi.org/10.1109/ICESC51422.2021.9532917>
- Hasan, M. H., Osman, M. H., Admodisastro, N. I., & Muhammad, M. S. (2023). Legacy systems to cloud migration: A review from the architectural perspective. *Journal of Systems and Software*, 202, 111702. <https://doi.org/10.1016/j.jss.2023.111702>
- Hinton, M. (2006). Generic types of information systems. Teoksessa *Introducing Information Management: The Business Approach* (ss. 91–97). <https://doi.org/10.4324/9780080458397-15>
- Hung, P., & Low, G. C. (2008). *Factors affecting the buy vs build decision in large Australian organisations*. 23(2), 118–131. <https://doi.org/10.1057/palgrave.jit.2000098>
- Johnson, L., Callaghan, C., Balasubramanian, M., Haq, H., & Spallek, H. (2019). Cost Comparison of an On-Premise IT Solution with a Cloud-Based Solution for Electronic Health Records in a Dental School Clinic. *Journal of Dental Education*, 83(8), 895–903. <https://doi.org/10.21815/JDE.019.089>
- Juhasz, Z. (2021). Quantitative cost comparison of on-premise and cloud infrastructure based EEG data processing. *Cluster Computing*, 24(2), 625–641. <https://doi.org/10.1007/s10586-020-03141-y>
- Kilcioglu, C., Rao, J. M., Kannan, A., & McAfee, R. P. (2017). *Usage patterns and the economics of the public cloud*. 83–91. <https://doi.org/10.1145/3038912.3052707>
- Klotins, E., Gorschek, T., Sundelin, K., & Falk, E. (2022). Towards cost-benefit evaluation for continuous software engineering activities. *Empirical Software Engineering*, 27(6). <https://doi.org/10.1007/s10664-022-10191-w>
- Kratzke, N. (2018). A brief history of cloud application architectures. *Applied Sciences (Switzerland)*, 8(8). <https://doi.org/10.3390/app8081368>
- Kratzke, N., & Peinl, R. (2016). *ClouNS-a Cloud-Native Application Reference Model for Enterprise Architects*. 2016-September, 198–207. <https://doi.org/10.1109/EDOCW.2016.7584353>

- Kratzke, N., & Quint, P.-C. (2017). Understanding cloud-native applications after 10 years of cloud computing—A systematic mapping study. *Journal of Systems and Software*, 126, 1–16. <https://doi.org/10.1016/j.jss.2017.01.001>
- Kumar Pandey, N., Chaudhary, S., & Joshi, N. K. (2016). Resource allocation strategies used in cloud computing: A critical analysis. *2016 2nd International Conference on Communication Control and Intelligent Systems (CCIS)*, 213–216. <https://doi.org/10.1109/CCIntelS.2016.7878233>
- Lichtenthäler, R., & Wirtz, G. (2024). Formulating a quality model for cloud-native software architectures: Conceptual and methodological considerations. *Cluster Computing*, 27(4), 4077–4093. <https://doi.org/10.1007/s10586-024-04343-4>
- Liu, S., & Wang, L. (2014). Understanding the impact of risks on performance in internal and outsourced information technology projects: The role of strategic importance. *International Journal of Project Management*, 32(8), 1494–1510. <https://doi.org/10.1016/j.ijproman.2014.01.012>
- Man, N. D., & Huh, E.-N. (2013). *Cost and efficiency-based scheduling on a general framework combining between cloud computing and local thick clients*. 258–263. <https://doi.org/10.1109/ComManTel.2013.6482401>
- Mao, H., Liu, S., Zhang, J., Zhang, Y., & Gong, Y. (2021). Information technology competency and organizational agility: Roles of absorptive capacity and information intensity. *Information Technology and People*, 34(1), 421–451. <https://doi.org/10.1108/ITP-12-2018-0560>
- Martin, R., Vidal, I., & Valera, F. (2024). A software-defined connectivity service for multi-cluster cloud native applications. *Computer Networks*, 248. <https://doi.org/10.1016/j.comnet.2024.110479>
- Mukange, T., & Langerman, J. (2022). *The Alignment between DevOps and Business Outcomes*. 8th International Conference on Engineering and Emerging Technologies, ICEET 2022. <https://doi.org/10.1109/ICEET56468.2022.10007221>
- Naidoo, R., & Möller, N. (2022). *Building Software Applications Securely With DevSecOps: A Socio-Technical Perspective*. 2022-June, 198–205. <https://doi.org/10.34190/eccws.21.1.295>
- Olsen, K. A., & Sætre, P. (2007). ERP for SMEs—Is proprietary software an alternative? *Business Process Management Journal*, 13(3), 379–389. <https://doi.org/10.1108/14637150710752290>
- Opara-Martins, J., Sahandi, R., & Tian, F. (2016). Critical analysis of vendor lock-in and its impact on cloud computing migration: A business perspective. *Journal of Cloud Computing*, 5(1). <https://doi.org/10.1186/s13677-016-0054-z>
- Pelle, I., Czentye, J., Doka, J., Kern, A., Gero, B. P., & Sonkoly, B. (2021). Operating Latency Sensitive Applications on Public Serverless Edge Cloud Platforms. *IEEE Internet of Things Journal*, 8(10), 7954–7972. <https://doi.org/10.1109/JIOT.2020.3042428>

- Piccoli, G., & Pigni, F. (2022). *Information systems for managers in the digital age: With cases* (Edition 5.0.). Prospect Press.
- Quattrocchi, G., & Tamburri, D. A. (2023). Infrastructure as Code. *IEEE Software*, 40(1), 37–40. <https://doi.org/10.1109/MS.2022.3212034>
- Rajamohan, S., & Settlage, R. E. (2020). *Informing the On/Off-prem Cloud Discussion in Higher Education*. 64–71. <https://doi.org/10.1145/3311790.3396627>
- Ramač, R., Mandić, V., Taušan, N., Rios, N., Freire, S., Pérez, B., Castellanos, C., Correal, D., Pacheco, A., Lopez, G., Izurieta, C., Seaman, C., & Spinola, R. (2022). Prevalence, common causes and effects of technical debt: Results from a family of surveys with the IT industry. *Journal of Systems and Software*, 184. <https://doi.org/10.1016/j.jss.2021.111114>
- Ramesh, A., Pradhan, V., & Lamkuche, H. (2021). *Understanding and Analysing Resource Utilization, Costing Strategies and Pricing Models in Cloud Computing*. 1964(4). <https://doi.org/10.1088/1742-6596/1964/4/042049>
- Rana, M. E., Farooq, U., & Rahman, W. N. W. A. B. (2019). Scalability enhancement for cloud-based applications using software oriented methods. *International Journal of Engineering and Advanced Technology*, 8(6), 4208–4213. <https://doi.org/10.35940/ijeat.F8869.088619>
- Red Hat (2022) IaaS vs. PaaS vs. SaaS. <<https://www.redhat.com/en/topics/cloud-computing/iaas-vs-paas-vs-saas>> Haettu 28. marraskuuta 2024.
- Sahandi, R., Alkhalil, A., & Opara-Martins, J. (2012). *SMEs' perception of cloud computing: Potential and security*. 380 *AICT*, 186–195. https://doi.org/10.1007/978-3-642-32775-9_19
- Senarathna, I., Warren, M., Yeoh, W., & Salzman, S. (2015). A conceptual model for cloud computing adoption by SMEs in Australia. Teoksessa *Web-Based Services: Concepts, Methodologies, Tools, and Applications* (ss. 307–334). <https://doi.org/10.4018/978-1-4666-9466-8.ch015>
- Shang, D., & Kauffman, R. J. (2020). A 2020 perspective on “Client risk informedness in brokered cloud services: An experimental pricing study”. *Electronic Commerce Research and Applications*, 41. <https://doi.org/10.1016/j.elerap.2020.100948>
- Smendowski, M., & Nawrocki, P. (2024). Optimizing multi-time series forecasting for enhanced cloud resource utilization based on machine learning. *Knowledge-Based Systems*, 304. <https://doi.org/10.1016/j.knosys.2024.112489>
- Thakur, S., & Jha, S. K. (2023). *Cloud Computing and its Emerging Trends on Big Data Analytics*. 1159–1164. <https://doi.org/10.1109/ICESC57686.2023.10193144>
- Toffetti, G., Brunner, S., Blöchlinger, M., Spillner, J., & Bohnert, T. M. (2017). Self-managing cloud-native applications: Design, implementation, and experience. *Future Generation Computer Systems*, 72, 165–179. <https://doi.org/10.1016/j.future.2016.09.002>

- Wallace, L., Keil, M., & Rai, A. (2004). How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model. *Decision Sciences*, 35(2), 289–321. <https://doi.org/10.1111/j.00117315.2004.02059.x>
- Wease, G., Boateng, K., Yu, C.-J., Chan, L., & Barham, H. (2018). Technology Assessment: Cloud Service Adoption Decision. Teoksessa *Innovation, Technology and Knowledge Management* (ss. 447–471). https://doi.org/10.1007/978-3-319-68987-6_16