



Vaasan yliopisto
UNIVERSITY OF VAASA

Jimi Jokinen

Projektijohtamisen menestystekijät ja haasteet ohjelmistoprojekteissa

Tekniikan ja innovaatiojohtamisen
akateeminen yksikkö
Kandidaatintutkielma
Tuotantotalous

Vaasa 2025

VAASAN YLIOPISTO**Tekniikan ja innovaatiojohtamisen akateeminen yksikkö**

Tekijä:	Jimi Jokinen		
Tutkielman nimi:	Kandidaatin tutkielma: Projektijohtamisen menestystekijät ja haasteet ohjelmistoprojekteissa		
Tutkinto:	Kauppätieteiden kandidaatti		
Koulutusohjelma:	tekniikan ja innovaatiojohtamisen yksikkö		
Opintosuunta:	Tuotantotalous		
Työn ohjaaja:	Tauno Kekäle		
Valmistumisvuosi:	2025	Sivumäärä:	37

TIIVISTELMÄ:

Tutkimus antaa kuvan siitä, miten ohjelmistoprojektit eroavat perinteisemmistä projekteista, mitä niiden johtamiseen liittyy ja millaiset tekijät johtavat onnistuneeseen projektiin. Ohjelmistoprojektien onnistunut toteutus projektijohtajan johdolla on ohjelmistoyritysten toiminnan kulmakivi ja projektien onnistumisella on suora vaikutus yrityksen kustannustehokkuuteen, kannattavuuteen ja kilpailukykyyn. Tutkimuksessa tarkastellaan projektijohtamisen menestystekijöitä ja niiden vaikutuksia ohjelmistoprojektien onnistumiseen. Menestystekijöiden tunnistaminen on tärkeää, jotta projektit voidaan viedä onnistuneesti loppuun. Tutkimuksessa pyritään tunnistamaan miten ketterät ja perinteiset johtamismenetelmät eroavat toisistaan ja mitä menestystekijöitä ne korostavat.

Toteutus on tehty kirjallisuuskatsauksena käyttämällä tieteellistä kirjallisuutta sekä vertaisarvioituja artikkeleita erilaisista tietokannoista. Tutkimus perustuu kriittisiin menestystekijöihin (CSF) ja projektijohtamisen menetelmiin.

Tutkimuksessa selvitetään, mitä ovat menestystekijät ja haasteet projektijohtamisessa ohjelmistoprojekteissa. Projektit voivat olla esimerkiksi ohjelmistopalveluihin (SaaS), ohjelmistokehitykseen tai järjestelmäintegraatioihin liittyviä. Digitalisaation ja teknologian kehityksen myötä projektijohtaminen on muuttunut paljon, joten johtamistavat ovat myös muuttuneet. Suuri osa ohjelmistoprojekteista on haasteellisia tai epäonnistuu ja tämä tulee yrityksille hyvin kalliiksi, minkä takia menestystekijöitä on tärkeää tutkia. Tutkimukset osoittavat, että tärkeimmät menestystekijät ovat pitkälti inhimillisiin eikä teknisiin tai organisatorisiin tekijöihin perustuvia.

AVAINSANAT: Project Management, Critical Success Factors (CSF), Agile, Software development, Software project

Sisällys

1	Johdanto	4
1.1	Tutkimuksen tausta	4
1.2	Tutkimuksen ongelmat ja tutkimuskysymykset	5
1.3	Tutkimuksen tavoitteet ja rakenne	5
2	Perusteet projektijohtamisesta ohjelmistoalalla	7
2.1	Projekti ja ohjelmistoprojekti	7
2.2	Projektijohtaminen	8
2.3	Ohjelmistoprojektin elinkaari ja vaiheet	11
3	Ohjelmistoprojektien haasteet	15
4	Kriittiset Menestystekijät	18
4.1	Tärkeimmät kriittiset menestystekijät ohjelmistoprojekteissa	18
4.2	Mittaaminen ja arviointi ohjelmistoprojekteissa	25
5	Yhteenveto	27
5.1	Tutkimuskysymykset ja johtopäätökset	27
5.2	Tulevaisuuden tutkimukset	29
	Lähteet	32

1 Johdanto

Tutkimuksessa selvitetään, mitä ovat menestystekijät ja haasteet projektijohtamisessa ohjelmistoprojekteissa. Projektit voivat olla esimerkiksi ohjelmistopalveluihin (SaaS), ohjelmistokehitykseen tai järjestelmäintegraatioihin liittyviä. Digitalisaation ja teknologian kehityksen myötä projektijohtaminen on muuttunut paljon, joten johtamistavat ovat myös muuttuneet.

1.1 Tutkimuksen tausta

Ohjelmistoteollisuus on kasvanut vertaansa vailla olevaa vauhtia viime vuosikymmeninä, ja nykypäivän ohjelmistojärjestelmien haasteet ovat erilaisia lisääntyneen monimutkaisuuden, koon ja kysynnän monimuotoisuuden suhteen kuin 50 vuotta sitten (Garousi ja muut, 2018, s. 429). Yritykset, joiden ohjelmistot eivät ole päivitettyjä ja ajan tasalla, saattavat menettää kilpailukykyään verrattuna hyvin johdettuihin ja uusia ohjelmistoja käyttäviin yrityksiin, joten kysyntä on suurta. Laadukas projektijohtaminen vaikuttaa yritysten menestykseen, maksimaalisen kilpailukyvyn saavuttamiseen sekä muuttuviin markkinoihin sopeutumiseen. Ymmärtämällä tärkeimmät kriittiset menestystekijät ja yhdistämällä ne ohjelmistoalalla jatkuvasti muuttuviin projektijohtamisen menetelmiin, voidaan keksiä parannuksia, jotka lisäävät ohjelmistotuotteiden arvoa ja laatua asiakkaille. Tutkimus tuo lisäarvoa projektijohtamiseen tunnistamalla keskeisimmät menestystekijät ja niiden hyödyntämisen ohjelmistoprojekteissa.

Jotta tutkimus olisi luotettava ja ajankohtainen, tutkimukseen etsittiin akateemisia, vertaisarvioituja artikkeleita sekä kirjoja projektijohtamisesta, kriittisistä menestystekijöistä, ohjelmistoprojekteista ja –johtamisesta sekä johtamismenetelmistä. Jotta tutkimus olisi ajankohtainen, tutkimuksessa pyrittiin käyttämään suurimmaksi osaksi uusia artikkeleita, jotta tieto olisi ajan tasalla ja arvokasta tällä hetkellä. Artikkeleita löydettiin eri tietokannoista kuten Google Scholarista, Researchgatesta, Academic Search Elitestä ja ACM Digital Library:stä.

1.2 Tutkimuksen ongelmat ja tutkimuskysymykset

Tutkimuksessa selvitetään, millä projektijohtamisen menestystekijöillä on suurimmat vaikutukset ohjelmistoprojektin onnistuneeseen lopputulokseen. Menestystekijöiden tunnistamisella ja käytännön hyödyntämisellä on suuri vaikutus ohjelmistoprojektiin, mutta niiden tunnistaminen voi olla haastavaa. Tutkimusongelmana on tutkia, mitä esteitä tiettyjen menestystekijöiden hyödyntämisessä ilmenee ja millaisia menetelmiä ohjelmistoprojekteissa on kannattavaa hyödyntää, sekä mikä vaikuttaa projektien epäonnistumiseen.

Pohjautuen tähän tutkimusongelmaan, tutkimuksessa käsitellään seuraavia kysymyksiä:

1. Mitkä ovat kriittiset menestystekijät onnistuneelle ohjelmistoprojektille?
2. Miten nämä menestystekijät vaikuttavat projektin lopputulokseen ja onnistumiseen?
3. Mitkä ovat suurimmat haasteet ohjelmistoprojekteissa ja menestystekijöiden noudattamisessa ja mikä johtaa epäonnistumiseen?

1.3 Tutkimuksen tavoitteet ja rakenne

Tutkimuksen päätavoite on tunnistaa ja analysoida menestystekijöitä, jotka vaikuttavat tehokkaasti projektijohtamisen kannalta ohjelmistoprojektien onnistumiseen. Tutkimus tähtää ymmärtämään keskeiset kriittiset menestystekijät, joiden hyödyntäminen johtaa onnistumiseen, kun taas laiminlyönti epäonnistumiseen.

Tutkimuksen rakenne on toteutettu seuraavasti: Kappaleessa 2 tarkastellaan ohjelmistoalan projektinhallinnan teoreettista perustaa ja hahmotellaan keskeisiä käsitteitä, kuten projektin ja projektijohtamisen määritelmä, ohjelmistoprojektien ominaisuudet ja projektinhallinnan perusteet.

Seuraavaksi tutkimus keskittyy kappaleessa 3 yleisiin haasteisiin ja esteisiin, jotka estävät kriittisten menestystekijöiden saavuttamisen ja vaikuttavat projektin epäonnistumiseen. Tässä osiossa vertaillaan myös eri projektinhallinnan malleja sen määrittämiseksi, mitkä lähestymistavat ovat osoittautuneet käytännössä tehokkaimmiksi.

4 kappaleessa tunnistamaan ja analysoimaan kriittisiä menestystekijöitä, jotka vaikuttavat ohjelmistoprojektien onnistumiseen. Näitä tekijöitä tarkastellaan teknisestä, inhimillisestä ja organisatorisesta näkökulmasta korostaen niiden keskinäisiä suhteita ja vaikutusta projektin kokonaissuorituskykyyn. Lisäksi käsitellään ohjelmistoprojektien menestystekijöiden mittaamista ja arviointia, esittelemällä keskeiset suorituskykyindikaattorit ja arviointimenetelmät, joita käytetään projektin tehokkuuden ja onnistumisen määrittämiseen.

Lopuksi 5 tutkimus päättyy yhteenvetoon ja synteesiin tärkeimmistä havainnoista, korostaen niiden vaikutuksia ohjelmistoprojektien hallintaan ja ehdottaen tulevaisuuden tutkimussuuntia.

2 Perusteet projektijohtamisesta ohjelmistoalalla

2.1 Projekti ja ohjelmistoprojekti

Projekti määritellään yksilölliseksi tai yhteiseksi hankkeeksi, johon saattaa liittyä tutkimusta tai suunnittelua, joka on yleensä projektitiimin huolellisesti suunnittelema tietyn tavoitteen saavuttamiseksi (Singh, n.d., s. 273). Tämä tarkoittaa, että projektille on määritelty alku ja loppu, se tuottaa jotain uutta ja se täytyy suorittaa tietyn ajan, budjetin ja laadun mukaisesti. Yleisesti projektit eroavat operationaalisesta työstä niiden ainutlaatuisuuden, epävarmuuden, useiden tehtävien ja sidosryhmien koordinoinnin vuoksi.

Yuge ja Badarch (2023, s. 2) artikkelissaan Research on Contemporary Software Development Life Cycle Models kertovat, että ohjelmistot jaetaan pääasiassa seitsemään luokkaan: järjestelmäohjelmistot, sovellusohjelmistot, tekniset/tieteelliset ohjelmistot, sulautetut ohjelmistot, tuotelinjaohjelmistot, verkkosovellukset ja tekoälyohjelmistot niiden käyttötarkoituksen mukaan. Ohjelmistoprojekti on prosessi, jossa luodaan tai kehitetään ohjelmisto tiettyyn liiketoiminnan tai käyttäjän tarpeeseen, jota tyypillisesti hallitaan muuttuvien vaatimusten ja kehittyvän teknologian alaisena.

Ohjelmistoprojektit, varsinkin dynaamisessa ympäristössä hyötyvät Agile-menetelmästä niiden sopeutumiskyvyn, sidosryhmien osallistumisen ja kehittyvien vaatimusten tarpeen takia (Das, 2025, s. 45). Hongbo ja muut (2024) tutkimuksessaan Software project scheduling under activity duration uncertainty kuvailevat, että ohjelmistoprojektit ovat ketterämpiä ja vaativat vähemmän fyysisiä resursseja. Ohjelmistoprojektien pääresurssit ovat erilaisilla taidoilla varustettuja henkilöstöresursseja, mikä tekee ohjelmistokehityksestä tyypillisen älykkyyttä vaativan prosessin ja siksi tehokas henkilöstöresurssien ohjaaminen on välttämätöntä ohjelmistoprojektien onnistumiselle (Hongbo ja muut, 2024).

2.2 Projektijohtaminen

The Project Management Institute määrittelee projektinjohtamisen tiedon, taitojen, työkalujen ja tekniikoiden soveltamiseksi projektitoimintoihin projektin vaatimusten täyttämiseksi. Yksinkertaisesti sanottuna se tarkoittaa projektin käsittelyn oikeaa suorittamista tiettyjen tavoitteiden saavuttamiseksi (Emsley & Altahtoo, 2013, s.366). Emsleyn ja Altahtoohin (2013, s. 367) mukaan projektijohtaminen on mahdollista nähdä sekä tieteenä, että taiteena ja se yhdistää analyyttisen ja systemaattisen osaamisen myös ihmisten johtamiseen, mikä vaatii inhimillistä vuorovaikutusta ja johtamistaitoja. Suunnittelu, toteutus ja valvonta ovat projektijohtamisen tieteen osa-alueita, kun taas johtajuus, tiiminrakennus ja realismi ovat projektijohtamisen taiteellisia osa-alueita (Emsley & Altahtoo, 2013, s.367).

Heagney (2016) korostaa teoksessaan Fundamentals of Project Management sitä, että projektijohtamisessa tärkeää on ihmisten johtaminen ja se, että esimiehen on saatava ihmiset suoriutumaan hyväksyttävän vähimmäissuoritus-tason yläpuolella. Joten ensimmäiset taidot, joita projektijohtaja tarvitsee, ovat ihmissuhdetaidot (Heagney, 2016). Heagney (2016) kertoo projektitiimin johtamisesta, että tiimejä on tarkoituksella rakennettava osallistumisen ja suunnittelun avulla. Hänen mukaansa tehokkaaseen tiiminrakennukseen kuuluu tavoitteiden, roolien ja vastuiden, menettelytapojen ja ihmissuhteiden käsittely järjestyksessä. Konfliktit johtuvat usein heikoista ihmissuhdetaidoista, joten kaikkia tiimin jäseniä tulisi kouluttaa tällä alueella. Johtamistyylin tulisi mukautua tiimin kehitysvaiheeseen: ohjaavaa muodostamisessa, vaikuttamista storming- vaiheessa, osallistuvaa normien asettamisessa ja delegeoivaa suorittamisessa (Heagney, 2016).

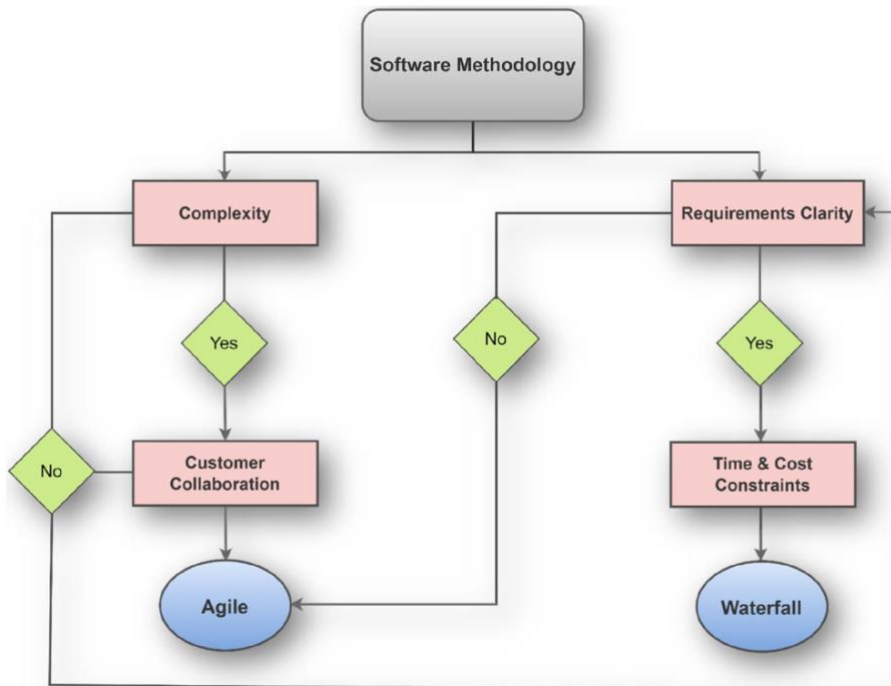
Kerznerin ja muiden (2022, s.64) teoksen Project Management Next Generation mukaan, jos projektijohtaminen määritellään vain seuraamalla kolmea päärajoitusta, jotka ovat aika, budjetti ja työn laajuus, suurimmassa tapauksista se ei auta voittamaan ongelmia, joita projekteissa kohdataan globaalilla tasolla. Epävarmuuden lisääntyessä

organisaatioista tulee yhä monimutkaisempia, ja ne edellyttävät muutosjohtamista (Kerzner ja muut, 2022, s.64).

Stan ja muut (2025) kertoo artikkelissaan ketteristä johtamismenetelmistä: Ketterä projektinhallinta (Agile), jolle on ominaista joustavuus, iteratiivinen kehitys ja keskittyminen asiakasyhteistyöhön, on tullut mullistavaksi tekniikaksi projektijohtamisen maailmassa. Ketterien tekniikoiden käytöllä on useita etuja, erityisesti nopean muutoksen ja epävarmuuden tilanteissa ja ketterät lähestymistavat parantavat joustavuutta ja sopeutumiskykyä, jolloin tiimit voivat reagoida muuttuviin vaatimuksiin ja sidosryhmien palautteeseen nopeammin (Stan ja muut, 2025, s. 2237). Noin 94 % IT-alan organisaatioista raportoi käyttävänsä ketteriä lähestymistapoja, mikä osoittaa merkittävän hyväksymisasteen IT-alan kyselyn mukaan ja sen laajamittaista käyttöönottoa vauhdittavat todistetut hyödyt, kuten nopeampi markkinoilletuloaika, parempi asiakastyytyväisyys ja tehokkaampi tiimityö (Lucos, 2025, s.1). Ohjelmistoprojektit hyötyvät Agile-menetelmästä niiden sopeutumiskyvyn, sidosryhmien osallistumisen ja kehittyvien vaatimusten tarpeen takia (Das, 2025, s. 45). Ketterä projektinhallinta sisältää myös erilaisia metodeita ja niistä ehkäpä suosituin on nimeltään Scrum. Reiffin ja Schlegelin (2021, s. 51) artikkelin Hybrid project management mukaan Scrum on yksi tunnetuimmista ja suosituimmista ketteristä menetelmistä. Monet Scrumin näkökohdat ovat yhteisiä ketterille menetelmille, kuten iteraatiot, inkrementaalinen kehitys, itseohjautuvat tiimit ja joustavuus muuttuvien vaatimusten edessä (Reiff & Schlegel, 2021, s.51).

Perinteinen menetelmä, jota kutsutaan myös vesiputousmalliksi (Waterfall) on vanhempi toimintatapa, jota käytetään edelleen. Se alkaa perusteellisella suunnittelulla ja analyysillä. Jokainen seuraava vaihe voi alkaa vasta edellisen valmistuttua. Mishran ja Alzoubin (2023) mukaan tämä menetelmä on melko jäykkä, eikä reagoi muutoksiin sekä yhteistyön puute asiakkaan kanssa on ongelma. Waterfall on kuitenkin uudelleenkäytettävä komponentti, jota voidaan käyttää projektin keston lyhentämiseen ja sen paremmassa järjestyksessä pitämiseen (Mishra & Alzoubi, 2023). Perinteistä

menetelmää kutsutaan vesiputousmalliksi, koska vaiheet virtaavat peräkkäin alaspäin eri vaiheiden läpi samalla tavalla kuin vesi virtaisi alas useita vaiheita (vaatimukset, suunnittelu, toteutus, testaus, käyttöönotto).



Kuva 1. Decision tree, joka auttaa valitsemaan onko Agile vai vesiputousmalli sopivampi projektille. (Mishra & Alzoubi, 2023)

Hybridi-malli on yhdistelmä Agile ja vesiputousmallista. Reiff ja Schlegel (2021, s. 45) kertoo artikkelissaan hybridimallista, että se on lähestymistapa, joka yhdistää perinteiset ja ketterät projektinhallintatekniikat ja sen tavoitteena on hyötyä kummankin lähestymistavan vahvuuksista ja samalla välttää heikkouksia. Ketterän mallin onnistumisprosentti on 40 %, kun taas vesiputousmallin projektien onnistumisprosentti on vain 15 % (Mishra & Alzoubi, 2023). Saadakseen parhaan molemmista maailmoista monet organisaatiot omaksuvat hybridimallin, jossa yhdistyvät ketterät menetelmät vesiputousmallin ja strukturoitujen ohjelmistokehitysmenetelmien kanssa (Mishra & Alzoubi, 2023).

2.3 Ohjelmistoprojektin elinkaari ja vaiheet

Yuge ja Badarch (2023, s. 3) esittelevät ohjelmistoprojektin elinkaaren (SDLC) seuraavasti; se on jäsennetty prosessi korkealaatuisen ja edullisen ohjelmiston rakentamiseen. Ohjelmiston elinkaarikehitys kuvataan seitsemässä vaiheessa, jotka sisältävät suunnittelun, vaatimusanalyysin, design, kehityksen, testauksen, toiminnan ja ylläpidon (Yuge & Badarch, 2023, s. 3). Kaikki ohjelmaprojektit eivät kuitenkaan noudata tätä elinkaarta ja projekti voikin kohdistua esimerkiksi vain testaukseen. Garousin ja muiden (2018, s. 443) tutkimuksessa tutkittiin sataa eri ohjelmistoprojektia ja he huomasivat, että 12 niistä eivät noudattaneet koko ohjelmistoprojektin elinkaarta. Heidän mukaansa useissa heidän meneillään olevissa teollisuuden ja akateemisen maailman yhteistyöprojekteissa projektien laajuus on ollut vain ohjelmistojen ylläpito tai vain testaus (Garousi ja muut, 2018, s. 443). Garousi ja muut (2018, s. 443) selvittivät projektien vaiheiden lukumäärän ja tyypit ja päätyi seuraavaan: heidän kyselyssä 88 projektia seurasi koko ohjelmistokehityksen elinkaaren, kun taas useimmat jäljellä olevista projekteista seurasivat vähintään kahta vaihetta tai toimintoa. Niistä projekteista, jotka koostuivat vain yhdestä vaiheesta tai toiminnosta, 5 oli ohjelmistovaatimusten/liiketoiminta-analyysien projekteja ja yksi oli ohjelmistojen testausprojekti (Garousi ja muut, 2018, s. 443). Kaikki ohjelmistoprojektit eivät siis koostu jokaisesta vaiheesta, mutta vähintään yhdestä tai kahdesta. Seuraavassa vaiheessa käydään läpi ohjelmistoprojektin eri vaiheet ja mitä niissä käytännössä tapahtuu.

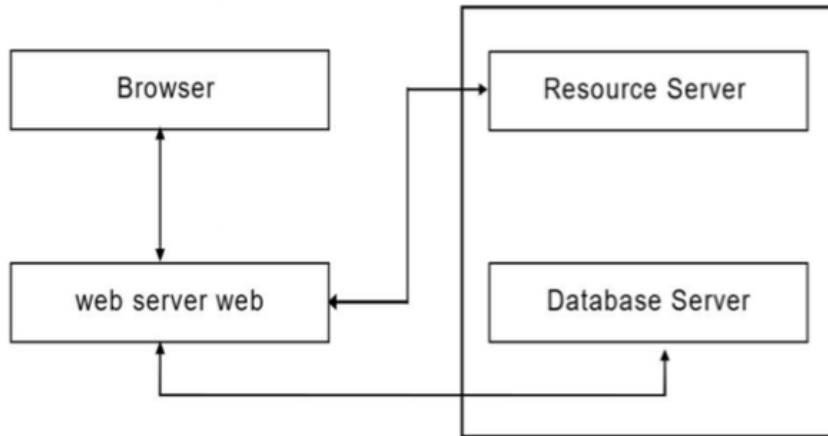
Suunnitteluvaihe: Yugen ja Badarchin (2023, s.3) mukaan ohjelmistoprojektin suunnitteluvaiheessa määritellään järjestelmän yleiset tavoitteet, vaatimukset ja tekniset rajoitukset samalla kun arvioidaan sen toteutettavuutta teknisestä, taloudellisesta ja sosiaalisesta näkökulmasta. Tässä vaiheessa analyytikot ja käyttäjät

tekevät yhteistyötä tunnistaakseen ongelmia, tutkiakseen mahdollisia ratkaisuja ja arvioidakseen tarvittavia resursseja, kuten laitteistoa, ohjelmistoja, henkilöstöä ja kustannuksia (Yuge & Badarch, 2023, s. 3). Tämän tuloksena on alustava projektin kehityssuunnitelma, joka sisältää tehtävien määritelmät, riski- ja kustannusanalyysit sekä yksityiskohtaisen projektin aikataulun.

Vaatimusanalyysi: Yugen ja Badarchin (2023, s.4) mukaan vaatimusanalyysivaihe on tärkein vaihe, ja sen tulos vaikuttaa koko ohjelmistokehitysprojektin onnistumiseen. Stiramin ja Bhuvanin (2011) artikkelissa *A Qualitative Study on Educational Software Requirements Analysis* he kuvailevat vaatimusanalyysiä seuraavanlaisesti; SDLC:n analyysivaiheessa tunnistetaan toiminnalliset ja ei-toiminnalliset vaatimukset, käyttäjävaatimukset, järjestelmävaatimukset ja rajapinnan määrittely. Toiminnallinen vaatimus määrittelee järjestelmän käyttäytymisen kaikissa olosuhteissa ja Ei-toiminnallinen vaatimus määrittelee palveluille asetetut rajoitukset (Sitram & Bhuvana, 2011) Sovellusalueen vaatimus määrittelee sovellusalueen ominaisuudet ja järjestelmän käyttäjävaatimukset on tunnistettava kaikkien tarvittavien käsitteellisten tietojen kera ja järjestelmävaatimukset määrittelevät käyttäjävaatimukset järjestelmäsuunnittelussa, sekä lisäksi rajapinnan määrittely kuvaa tarkan rajapinnan, joka sopii nykyiseen järjestelmään (Sitram & Bhuvana, 2011).

Design- vaiheesta Yuge ja Badarch (2023, s.4) kertovat seuraavasti: Designvaiheessa vaatimukset muutetaan konkreettiseksi ohjelmistoarkkitehtuuriksi määrittelemällä järjestelmän rakenne, moduulit, algoritmit ja datan organisointi. Se alkaa ohjelmistovaatimusten määrittelyllä ja sisältää sekä yleisen että yksityiskohtaisen suunnittelun, joka kattaa järjestelmäkehityksen, tietokannan ja rajapintojen määritelmät (Yuge & Badarch, 2023, s.4). Tässä vaiheessa tuotetaan myös suunnitteluasiakirjoja – kuten suunnittelumäärittelyksiä ja testaussuunnitelmia – ja korostetaan moduulitason yksityiskohtia, kuten funktioita, algoritmeja ja datarakenteita, selkeiden suhteiden ja tehokkaan järjestelmäintegraation varmistamiseksi (Yuge & Badarch, 2023, s.4).

Suunnitteluvaiheessa järjestelmän kokonaisrakenne voidaan kuvata tällä tavalla kaaviolausekkeen avulla (kuva 2).



Kuva 2. järjestelmän kokonaisrakenne kaaviolausekkeena (Yuge & Badarch, 2023, s.4)

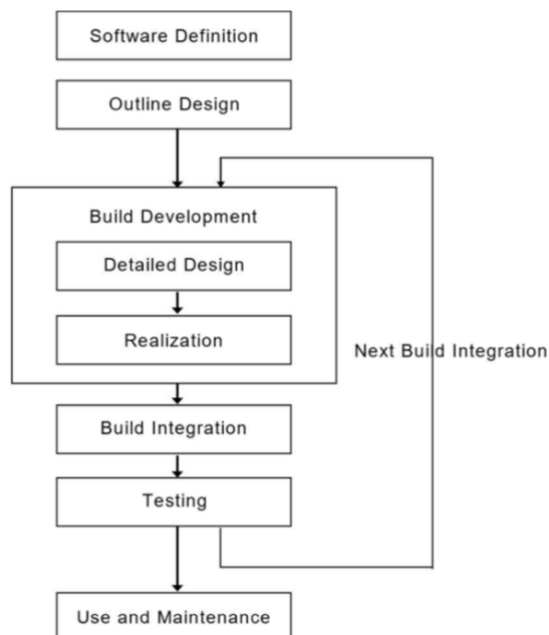
Ohjelmakoodin vaihe: Yugen ja Badarchin (2023, s.4–5) mukaan koodausvaiheessa ohjelmistosuunnittelu muunnetaan suoritettavaksi koodiksi varmistaen samalla, että noudatetaan koodausstandardeja, jotka edistävät luettavuutta, ylläpidettävyyttä ja tehokkuutta. Koodin laatua arvioidaan vakiintuneita virheprosenttirajoja vasten parannuskohteiden tunnistamiseksi. Ydinkehitysvaiheena se sisältää myös johdon valvontaa – tehtävien koordinoitua, edistymisen seuranta ja tiimin jäsenten välisen viestinnän ylläpitämistä koko kehitys- ja testausprosessin ajan (Yuge & Badarch, 2023, s.4).

Testaus: Yuge ja Badarch (2023, s.5) kertovat testaamisesta, että Ohjelmistosuunnittelun valmistuttua kehitys etenee tiukan testausprosessin läpi suunnitteluongelmien tunnistamiseksi ja korjaamiseksi. Testaus jaetaan kolmeen vaiheeseen: yksikkötestaus, integraatiotestaus ja järjestelmätestaus, joissa käytetään sekä white box- että black box -menetelmiä tiukan testaus suunnitelman mukaisesti (Yuge & Badarch, 2023, s.5). Yksikkötestaus varmistaa yksittäisten komponenttien oikeellisuuden suunnittelumäärittämiä vasten dokumentaation tarkastelun, staattisen ja dynaamisen analyysin sekä koodin tarkastelun avulla. Integraatiotestaus varmistaa sitten, että

moduulit ja alijärjestelmät toimivat oikein yhdessä ja täyttävät suunnitteluvaatimukset, mikä auttaa vähentämään virheiden korjaamisen kustannuksia. Lopuksi järjestelmättestaus johon kuuluu toteutettavuus-, toimivuus- ja käytettävyydestit, kuten navigointi-, grafiikka-, sisältö- ja käyttöliittymättestaus validoi, että koko ohjelmisto täyttää kaikki määritetyt toiminnalliset ja suorituskykyvaatimukset (Yuge & Badarch, 2023, s.5).

Yugen ja Badarchin (2023, s.5) mukaan elinkaaren operaatio- ja ylläpitovaiheen aikana käyttäjät käyttävät viimeisteltyä ohjelmistoa, kun taas kehittäjät tarjoavat jatkuvaa ylläpitoa ja tukea, kunnes vastuu siirtyy ylläpito-organisaatiolle tai käyttäjille itselleen. Ohjelmistoa voidaan joutua muuttamaan virheiden korjaamiseksi, ympäristömuutoksiin sopeutumiseksi tai toiminnallisuuden parantamiseksi. Vaihe sisältää kolme alavaihetta: käyttö, ylläpito ja käytöstäpoisto. Yhdessä nämä alavaiheet määrittelevät ohjelmiston elinkaaren koko käyttövaiheen. (Yuge & Badarch, 2023, s.5)

Alla tutkijan tekemä kaavio, joka kuvaa ohjelmistoprojektin etenemistä vaiheittain.



Kuva 3. Tutkijan tekemä kaavio ohjelmistoprojektin kulusta vaiheittain (Yuge & Badarch, 2023, s.5)

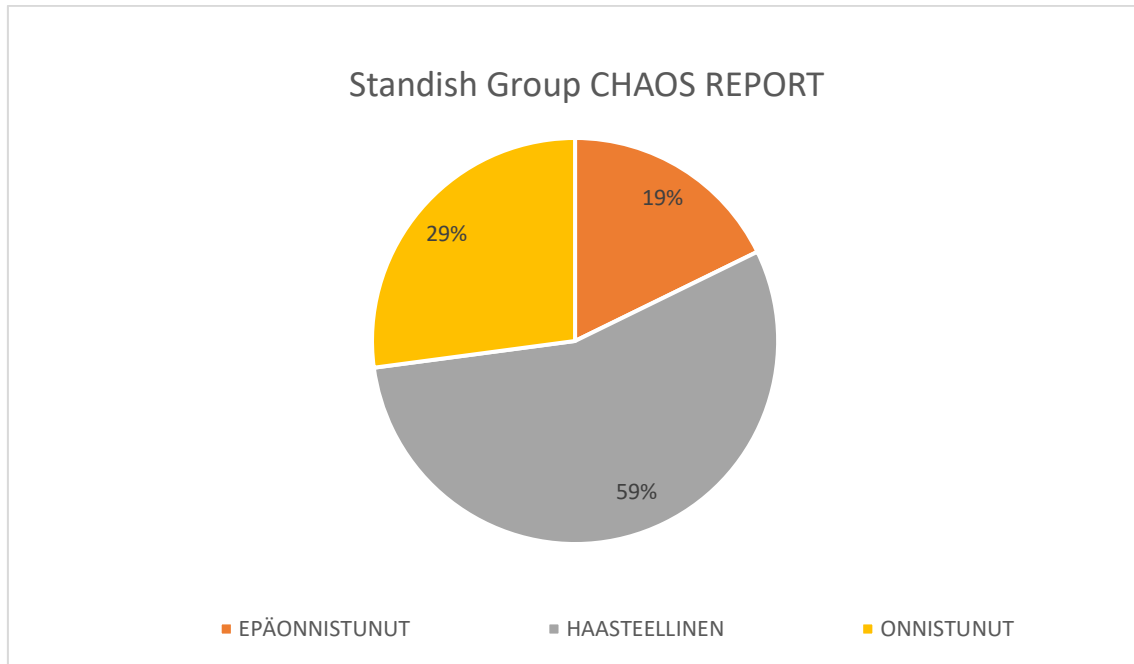
3 Ohjelmistoprojektien haasteet

Ohjelmistoprojekteihin liittyy useita haasteita ja ne ovat monimutkaisuuden ja jatkuvan vaihtuvuuden takia alttiita epäonnistumisille. Mircean ja Nguyenin (2023, s. 62) artikkeli, joka liittyy projektien virheiden määrien määrittämiseen, projektikustannusten ylittymiseen, budjettisuunnitteluvirheisiin ja aikataulujen noudattamisesta poikkeamiseen, he kertovat, että vaikka ohjelmistoprojektit ovatkin tärkeitä kaikille nykymaailman osa-alueille, niiden hallinta ei ole ihanteellinen prosessi. Yksi keskeisistä ongelmista, joita sekä ohjelmistokehittäjät että asiakkaat kohtaavat, on ohjelmistoprojektin käyttöönoton eri vaiheisiin liittyvän riskin aste. El-Deebin (2022, s. 15) artikkeli Major Challenges Currently Facing the Software Industry kertoo, että yhdysvaltalaisen yritysten epäonnistuneiden kehitysprojektien kokonaiskustannukset ovat arviolta 260 miljardia dollaria, kun taas huonolaatuisen ohjelmiston aiheuttamien operatiivisten häiriöiden kokonaiskustannukset ovat arviolta 1,56 biljoonaa dollaria.

Ohjelmistoprojektien epäonnistumisten syistä Tam ja Oliveira (2020) argumentoivat, että tekniset ongelmat olivat harvoin syynä ja että ongelmat voivat johtua ihmisistä, ja että tätä voidaan torjua asianmukaisella henkilöstöjohtamisella. Jeteendren ja muiden (2022) mukaan vaikeiden ja erilaisten prosessien vuoksi ohjelmistoprojektien hallintaan vaikuttavat usein monet ongelmat. Yleisimpiä ongelmia ovat huono projektinhallinta, budjettien ylittäminen, kyvyttömyys saada ohjelmistokehitys valmiiksi aikataulussa, ohjelmiston heikko laatu ja ohjelmistokehittäjien motivaation puute. Lisäksi sidosryhmien osallistumisen puute, tiimityön puute, kohtuuttomat odotukset ja heikko viestintätaso projektissa ovat yleisimpiä epäonnistumisen syitä ohjelmistoprojekteissa (Jeteendre ja muut, 2022, s. 21).

IT-projektien epäonnistumisiin voidaan lukea myös mukaan haasteelliset projektit, joissa esimerkiksi budjetti ylittyy huomattavasti. Altahtoohn ja Emsleyn (2013, s. 366) mukaan lähes kaikki heidän tutkimat tutkimukset osoittivat, että haasteelliset projektit ovat eräänlainen epäonnistunut projekti. Vanhemmassa tutkimuksessa CHAOS-raporttien mukaan epäonnistuneiden projektien keskimääräinen prosenttiosuus vuosien 1997-

2013 aikana on ollut 71 %, kun haasteelliset projektit on laskettu mukaan (24 % + 47 %) ja näistä 47 % on haasteellisia projekteja (Emsley & Altahtoo, 2013, s.366). Vaikka haasteelliset projektit eivät ole täysin epäonnistuneita projekteja, ne sijaitsevat harmaalla alueella. Tämä korkea projektien epäonnistumisaste aiheuttaa konflikteja projektipäälliköiden ja toimitusjohtajien sekä sidosryhmien välille. Emsley & Altahtoo (2013, s.366). Ohjelmistoprojektit ovat IT-projektien alalaji ja prosentit eivät ole täysin samanlaiset, mutta Emsleyn ja Altahtoon tutkimus antaa osviittaa siitä, kuinka paljon projekteja epäonnistuu. Kärklinan ja Pirtan (2021, s. 54) artikkelissa kerrotaan, että vuosi 2017 on kuitenkin ensimmäinen, jolloin onnistuneesti johdettujen projektien määrä on alkanut kasvaa, joten epäonnistumisen prosenttiluvut ovat pienemmät nykyään. Tärkeimmät syyt ovat teknisen ja johtamisen tiedon parantuminen sekä ketterät projektinhallintamenetelmät (Kärklina & Pirta (2021, s. 54). Standish Groupin vuonna 2023 tekemä tutkimus osoitti, että 19 % yli 50 000 ohjelmistoprojektista epäonnistuu eikä koskaan valmistu (Mircea & Nguyen 2023, s. 62). Nizam (2022) puolestaan kertoo artikkelissaan, että epäonnistuminen aiheuttaa taloudellisia ja moraalisia menetyksiä, vähentää tai tuhoaa odotettuja hyötyjä. Merkittävydestään ja käytetyistä resursseista huolimatta ohjelmistoprojektien epäonnistumisaste on pysynyt korkeana viimeisten kymmenen vuoden aikana ja Standish Groupin CHAOS-raportin mukaan onnistumisaste oli kokonaisuudessaan vain 29 prosenttia, kun taas haasteellisten projektien osuus oli 59 prosenttia ja peruutettujen projektien 19 prosenttia (Nizam, 2022).



Kuva 4. Ymypyräkaavio onnistuneista, haasteellisista ja epäonnistuneista ohjelmistoprojekteista Standish Groupin CHAOS-raportin mukaan.

Näiden haasteiden valossa on olennaista tunnistaa, mitkä tekijät erottavat onnistuneet ohjelmistoprojektit epäonnistuneista. Tämä onnistuu identifioimalla menestystekijät, jotka ehkäisevät haasteiden toteutumista.

4 Kriittiset Menestystekijät

Projektien menestyksen mittaustavat ja niiden saavuttamiskeinot ovat kehittyneet vähitellen vuosien varrella. Tamin ja Oliveiran (2020) mukaan aluksi kirjallisuudessa käytettiin rautakolmiota (Iron Triangle) projektien arviointiin, sitten kehitettiin kriittisten menestystekijöiden listat, ja sen jälkeen otettiin käyttöön ensimmäiset menestyskehukset. Nämä kehukset alkoivat korostaa asiakaskeksyyttä, ja tämä käytäntö on jatkunut 2000-luvulle asti (Tam & Oliveira, 2020).

Kriittiset menestystekijät on yleinen projektitoimintaan ja projektijohtamiseen liittyvä teoria, jossa tunnistetaan avainalueet ja edellytykset, jotka tulisi saavuttaa tai suorittaa hyvin, jotta projekti olisi onnistunut. Toisaalta, jos menestystekijöitä ei noudateta, projekti epäonnistuu. Aliaksen (2014) artikkelin Determining Critical Success Factors of Project Management Practice mukaan projektijohtamisen perspektiivistä menestystekijät ovat ominaisuuksia, olosuhteita tai muuttujia, joilla voi olla merkittävä vaikutus projektin onnistumiseen, kun niitä ylläpidetään, hoidetaan tai hallitaan asianmukaisesti. Jos projektipäälliköt eivät ole tietoisia lähtövaiheessa asetettuihin tavoitteisiin vaikuttavista kriteereistä, projekti ei onnistu (Alias, 2014). Tämän vuoksi on tärkeää tunnistaa, mitkä menestystekijät vaikuttavat eniten ohjelmistoprojektin onnistumiseen.

4.1 Tärkeimmät kriittiset menestystekijät ohjelmistoprojekteissa

Ohjelmistoprojekteissa kriittiset menestystekijät ovat erittäin tärkeitä, sillä ohjelmistoprojektit ovat usein erittäin monimutkaisia, teknisesti vaativia ja alttiita jatkuville muutoksille. Aikaisemmat tutkimukset, kuten Chow ja Cao (2008) perustelevat, että keskeisimmät ohjelmistoprojektien kriittiset menestystekijät voidaan jakaa kolmeen yläluokkaan: teknisiin, inhimillisiin ja organisatorisiin tekijöihin.

Tekniset tekijät ovat ohjelmistoprojektin teknologiaan liittyviä näkökohtia kuten työkalut, menetelmät ja tekninen asiantuntemus, jotka vaikuttavat projektin tavoitteiden

saavuttamisen tehokkuuteen. Tekniset tekijät voidaan jakaa Chow & Caon (2008) mukaan 8 eri tekijään. Tekijöitä ovat hyvin määritellyt koodausstandardit etukäteen, yksinkertaisen suunnittelun tavoittelu, perusteelliset refaktorointitoimet, oikea määrä dokumentaatiota, ohjelmiston säännöllinen toimitus, tärkeimpien ominaisuuksien toimittaminen ensin, oikea integraatiotestaus ja tiimin asianmukainen tekninen koulutus (Chow & Cao, 2008).

Inhimilliset tekijät liittyvät tiimin osaamiseen ja projektipäällikön kykyyn johtaa ja motivoida ihmisiä tehokkaasti. Chow & Caon (2008) mukaan tekijöitä ovat tiimin jäsenet, joilla on korkea osaaminen ja asiantuntemus, tiimin jäsenet, joilla on vahva motivaatio, esimiehet, joilla on kokemusta ketteristä prosesseista, esimiehet, joilla on kevyt tai mukautuva johtamistyyli, yhtenäinen, itseorganisoituva tiimityö ja hyvät asiakassuhteet. (Chow & Cao, 2008)

Organisatoriset tekijät projektissa ovat tekijöitä, jotka muokkaavat projektin jäsentämistä, hallintaa ja tukea laajemmin organisaatiossa. Chow & Cao (2008) kuvailee organisatoriset tekijät seuraavasti: vahva johdon tuki, sitoutunut esimies, yhteistyökykyinen organisaatiokulttuuri hierarkkisen sijaan, suullinen kulttuuri, jossa kasvokkain tapahtuva viestintä on tärkeää, organisaatiot, joissa ketterät menetelmät ovat yleisesti hyväksytyjä, Koko tiimin yhteissijoittaminen, tilat, joissa on asianmukainen ketterä työympäristö, ketteriin menetelmiin sopiva palkitsemisjärjestelmä. (Chow & Cao, 2008). Teknologian kehitys ja nykyinen työskentely ovat muokanneet osaa tekijöistä, kuten tilat ja suullinen kulttuuri, kun työskentely on siirtynyt yhä enemmän etätöihin. Nämä tekijät ovat edelleen tärkeitä, ja organisaatiot ovat pyrkineet ylläpitämään niitä hyvin jäseneltyjen etätöikäytäntöjen ja tehokkaiden digitaalisten viestintätyökalujen avulla, jotka mahdollistavat yhteistyön ja henkilökohtaisen vuorovaikutuksen myös ilman kasvokkain tapahtuvaa kontaktia. Etätö on myös lisännyt inhimillisen tekijän itseorganisoituvan tiimityön tärkeyttä.

Ohjelmistokehitysprojektien menestys on erittäin vaikeaa. Itse asiassa yksi suurimmista haasteista tällaisissa projekteissa on ymmärtää, miten ohjelmistokehitystä voidaan parantaa sen epäonnistumisen välttämiseksi. Bogopan ja Marnewickin (2022, s. 6) artikkelin *Critical Success Factors in Software Projects* mukaan, jos kysymys projektin onnistumisesta on "Toimitettiinko projekti ajallaan, budjetin puitteissa ja vaaditun laadun mukaisesti?" ja jos vastaus kaikkiin näihin kysymyksiin on "kyllä", se tarkoittaa, että projekti oli onnistunut projektipäällikön näkemyksen ja projektin onnistumisen määritelmän mukaan.

Kun ketterien ohjelmistokehitysprojektien kriittiset menestystekijät eivät ole loogisesti yhteydessä projektin tavoitteisiin ja niiden erityiseen ympäristöön, tämä johtaa ohjelmistoprojektien epäonnistumiseen aikataulussa pysymisessä ja ohjelmistoprojektille aiheutuviin ylimääräisiin kustannuksiin, jotka ovat korkeammat kuin budjettiin on suunniteltu (Chow & Cao, 2008). Standish Groupin vuosittain julkaisemien *CHAOS*-raporttien, sekä erilaisten tutkimusten (Wolff ja muut, 2024; Alias, 2014; Bogopa & Marnewick, 2022; Nasir & Sahibuddin, 2011; Sudhakar, 2012; Tam & Oliveira, 2020; Mircea & Nguyen, 2023) mukaan ohjelmistoprojekteissa yleisimmin tunnistetut kriittiset menestystekijät olivat tehokas johtajuus, selkeät tavoitteet ja vaatimukset, vahva tiimiosaaminen, asiakkaan osallistuminen, realistinen suunnittelu, tehokas viestintä ja kyky mukautua muutoksiin. Inhimilliset tekijät, kuten henkilöstöön ja johtamiseen liittyvät tekijät vaikuttavat projektin onnistumiseen vahvasti ja tämän lisäksi ohjelmistokehitykseen liittyy korkea monimutkaisuusaste jatkuvasti muuttuvassa ympäristössä, joten ohjelmistoprojektit vaativat tehokasta johtamista innovatiivisten lähestymistapojen avulla (Mircea & Nguyen, 2023).

Näiden tekijöiden voidaan katsoa olevan universaaleja riippumatta käytetystä menetelmästä, mutta niiden painoarvo vaihtelee projektin tyyppin ja laajuuden mukaan. Esimerkiksi ketterissä menetelmissä korostuvat jatkuva viestintä ja palautesyklit, kun taas perinteisissä vesiputousmallin projekteissa painotetaan suunnitelmallisuutta ja dokumentointia. Moni tutkimus on suoritettu haastatteleamalla Agile- menetelmällä johdettujen projektien johtajia. Tämä menetelmä on ominaista ohjelmistoprojekteille.

Etelä-Afrikkalainen Chiyangwan (2017) tutkimus *Modelling the critical success factors of agile software development projects in South Africa*, jossa tutkittiin erilaisten kriittisten menestystekijöiden vaikutusta projektin lopputulokseen, tultiin seuraaviin johtopäätöksiin: Tulokset osoittavat, että yksilöiden taipumusta soveltaa ketteriä ohjelmistokehitysprojekteja määrittävät pääasiassa organisatoriset tekijät, jotka liittyvät suoritusodotuksiin. Ihmistekijöillä havaittiin olevan merkittävä positiivinen vaikutus suoritusodotuksiin, mikä puolestaan edistää ketterien ohjelmistoprojektien onnistumista. Lisäksi organisatoristen tekijöiden todettiin olevan tilastollisesti merkitseviä ketterän ohjelmistokehityksen kannalta, mutta suoritusodotusten havaittiin olevan kaikkein tärkein tekijä, joka tulisi ottaa huomioon ohjelmistoprojektien edistämiseksi Etelä-Afrikassa. (Chiyangwa, 2017)

Bogopan ja Marnewickin (2022) tutkimuksen mukaan taas näyttää siltä, että onnistuneilla projekteilla on vahvoja ei-teknisiä tekijöitä, kuten johdon tuki ja käyttäjien osallistuminen, jotka voivat johtaa selkeästi määriteltyihin vaatimuksiin ja projektin tavoitteisiin. Joten teknologialla, työkaluilla ja menetelmillä on tärkeä, mutta vähemmän vaikuttava rooli. Ohjelmistokehittäjät ja järjestelmäanalytikot ovat tunnustaneet, että käyttäjien osallistuminen on yksi ohjelmistoprojektien onnistumisen ratkaisevista tekijöistä (Bogopa & Marnewick, 2022). Tämän tutkimuksen mukaan siis inhimilliset tekijät ovat tärkeämpiä kuin Chiyangwan (2017) tutkimuksen organisatorisilla tekijöillä. Tämä voi johtua projektien luonteesta tai siitä, että Bogopan ja Marnewickin (2022) tutkimus on 5 vuotta uudempi, ja ohjelmistoalalla viidessä vuodessa teknologia ja projektien luonteet ehtivät muuttua paljon. Tutkimukset kuitenkin osoittavat johdonmukaisesti, että menestys riippuu muustakin kuin vain teknisestä huippuosaamisesta. Ajallaan, budjetilla ja laadulla saavutettujen tavoitteiden saavuttaminen määrittelee menestyksen, mutta tutkimukset korostavat, että inhimilliset ja organisaatioon liittyvät tekijät kuten johtajuus, selkeät tavoitteet, tehokas viestintä, käyttäjien osallistuminen ja sopeutumiskyky ovat tärkeämpiä kuin pelkät työkalut tai menetelmät. Ketterät ja perinteiset projektit painottavat erilaisia käytäntöjä, mutta

keskeiset menestystekijät ovat universaaleja. Kaiken kaikkiaan vahva ihmiskeskeinen johtaminen ja projektin tavoitteiden, odotusten ja kontekstin välinen yhdenmukaisuus ovat olennaisia tulosten parantamiseksi ja epäonnistumisten vähentämiseksi ohjelmistokehityksessä.

Seuraavaksi seitsemästä tutkimuksesta on löydetty 3 tärkeintä menestystekijää ohjelmistoprojekteissa ja ne on laitettu tärkeysjärjestykseen. Tutkimukset ovat laajalta ajalta ja ne antavat mallia siitä, mitkä menestystekijät nousevat useimmin esiin.

Garousi ja muut (2018)

1. tarka seuranta ja valvonta
2. Hyvä suunnittelu
3. Tiimin kyvykkyys

(Lin & Chang, 2018)

1. Johdon tuki ja johtajan osaaminen
2. kokemus
3. tehokas viestintä

(Sahibuddin, 2011)

1. selkeät vaatimukset ja tavoitteet
2. hyvä suunnittelu
3. johdon tuki ja johtajan osaaminen

(Bogopa & Marnewick, 2022)

1. tiimin kyvykkyys
2. asiakkaiden osallistaminen
3. selkeät vaatimukset ja tavoitteet

(Tam & Oliveira 2020)

1. tiimin kyvykkyys
2. asiakkaiden osallistaminen
3. johdon tuki ja johtajan osaaminen

(Setiawan, 2025)

1. Johdon tuki ja johtajan osaaminen
2. tiimin kyvykkyys
3. asiakkaiden osallistaminen

(Gheni, 2025)

1. tiimin kyvykkyys
2. tehokas viestintä
3. selkeät vaatimukset ja tavoitteet

Menestystekijä	esiintynyt tutkimuksissa top-3 (%)	Tutkimukset, joissa esiintyy	Sija tärkeysjärjestyksessä (1-7)
tehokas viestintä	28,6 %	Lin & Chang (2018), Gheni (2025)	5
Asiakkaiden osallistaminen (UI ja UP)	42,9 %	Bogopa & Marnewick (2022), Tam & Oliveira (2020), Setiawan (2025)	3
Selkeät vaatimukset ja tavoitteet	42,9 %	Sahibuddin (2011), Bogopa & Marnewick (2022), Gheni (2025)	4
johdon tuki ja projektijohtajan osaaminen	57,1 %	Lin & Chang (2018), Sahibuddin (2011), Tam & Oliveira (2020), Setiawan (2025)	2
tiimin kyvykkyys	71,4 %	Garousi ja muut (2018), Bogopa & Marnewick (2022), Tam & Oliveira (2020),	1

		Setiawan (2025), Gheni (2025)	
hyvä suunnittelu	28,6 %	Garousi ja muut (2018), Sahibuddin (2011)	6
tarkka seuranta ja valvonta	14,3 %	Garousi ja muut (2018)	7

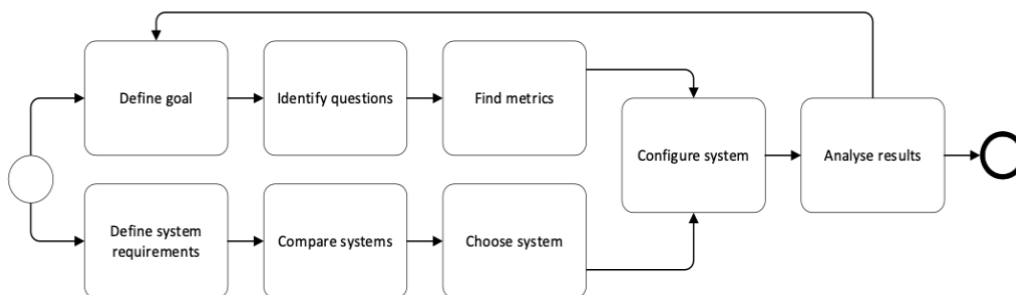
Taulukko 1. Yhteenveto tärkeimmistä menestystekijöistä ohjelmistoprojektissa

Tutkimusten perusteella 4 tärkeintä menestystekijää ovat tiimin kyvykkyys, johdon tuki ja projektijohtajan osaaminen, asiakkaiden osallistaminen ja selkeät vaatimukset ja tavoitteet.

Tiimin kyvykkyys: Jos tiimi on aiemmin johtanut vastaavia projekteja, sillä on parempi ymmärrys kyseisestä toimialasta ja siten se todennäköisesti tekee vähemmän vakavia virheitä. Tiimitasolla, jos tiimi on jo johtanut useita projekteja yhdessä, se ymmärtää paremmin projektin toteutusprosessin ja pystyy siten paremmin toteuttamaan tulevia projekteja menestyksekkäästi (Lin ja muut, 2016). Johdon tuki ja projektijohtajan osaaminen: Linin ja muiden (2016) mukaan projektijohtajan on kyettävä ymmärtämään ja hallitsemaan koko tilannetta ja valitsemaan sopivat toimenpiteet onnistumisen varmistamiseksi. Se, tukeeko ylin johto projektia ja kuinka paljon he tukevat sitä, vaikuttaa tiimin pääsyyn resursseihin ja niiden soveltamiseen (Lin ja muut, 2016). Asiakkaiden osallistaminen: Projektin sidosryhmät yrittävät usein puuttua projektiin omien etujensa mukaisesti ja yrittää muuttaa prosessia tai sisältöä, ja tällä on merkittävä vaikutus lopputulokseen (Lin ja muut, 2016). Selkeät vaatimukset ja tavoitteet: vaikuttaa projektin aikataulun ja budjetin tehokkaaseen noudattamiseen. Jotta projekti on onnistunut, se vaatii selkeät vaatimukset ja tavoitteet johdolta sekä asiakkaalta (Nasir & Sahibuddin, 2011).

4.2 Mittaaminen ja arviointi ohjelmistoprojekteissa

Perinteisessä näkemyksessä projektin onnistumisesta lähtökohtaisesti parannuksia arvioitiin rautakolmion, jota kutsutaan kolmoisrajoitukseksi, kautta: laajuus, määräaika ja kustannukset. Siksi menestyksen mittaaminen ja arviointi ei ole helppoa ja se riippuu sidosryhmän näkökulmasta, projektin tyypistä, ajallisesta näkökulmasta ja organisaatiosta (Besteiro & Pinto, 2015). Nykyaikaisissa jatkuvaa muutosta vaativissa ohjelmistoprojekteissa Kärklinan ja Pirtan (2021, s. 54) mukaan tulisi suunnitella ja käyttää omia mittareita vastauksena tunnistettuihin tarpeisiin ennalta määriteltyjen mittareiden sijaan. Hyvän mittarin tulisi olla helposti mitattavissa, sidottu liiketoimintatavoitteisiin; sen tulisi ennustaa tulevaa liiketoiminnan suorituskykyä ja olla erillään eri tekijöistä. (Kärklina & Pirta, 2021, s. 54)



Kuva 5. Kärklinan ja Pirtan luoma Flow-kartta projektin mittaamisesta ja arvioinnista (Kärklina & Pirta, 2021, s. 57)

Projektin onnistumista voi olla vaikea mitata, ja onnistumisen merkit voivatkin näkyä vasta vuosien päästä. Business Management Dynamics Besteiron ja Pinton (2015) teoksen Business Management Dynamics mukaan projektin onnistumista voidaan tarkastella ajallisesti eri näkökulmista, sillä projektinhallinnan onnistuminen voidaan arvioida jo projektin päättyessä, kun taas itse projektin todellinen menestys saattaa ilmetä vasta kuukausien tai jopa vuosien kuluttua projektin valmistumisesta.

Orozco-Garcesin (2022) artikkelissa Metrics Model to Complement the Evaluation of DevOps in Software Companies näytetään malli mittareista DevOps-ympäristössä

ohjelmistoalan yrityksille. Artikkelin mukaan ohjelmistoalalla Monet organisaatiot ovat viime aikoina keskittyneet parantamaan operatiivisia käytäntöjä ITIL, COBIT ja ISO/IEC-standardien kaltaisten viitekehysten avulla varmistamaan luotettavuuden ja vakauden tuotantoympäristöissä (Orozco & Garces, 2022). DevOps käsite yhdistää kehitys- (Dev) ja operatiiviset (Ops) käytännöt parantaakseen yhteistyötä, automaatiota ja tehokkuutta koko ohjelmiston elinkaaren ajan (Orozco-Garces, 2022). Se korostaa jatkuvaa integrointia, automatisoitua testausta, käyttöönottoa ja ylläpitoa tuottavuuden, laadun ja kilpailukyvyn parantamiseksi. DevOps-tutkimus tarjoaa esimerkin siitä, miten strukturoitu mittaaminen voi parantaa sekä kehitys- että operatiivisia käytäntöjä, ja artikkelissa esitetäänkin metriikkamalli, joka on suunniteltu systemaattisen ja organisoidun prosessin avulla tavoite-kysymys-metriikka (GQM) lähestymistapaa noudattaen (Orozco-Garces, 2022). Tämä malli siis mahdollistaa yrityksille DevOps toteutuksen laajuuden arvioinnin neljästä näkökulmasta: ihmiset, kulttuuri, teknologia ja prosessit. Yhdenmukaistamalla kirjallisuuden systemaattisen kartoituksen avulla tunnistettuja metodologisia ratkaisuja ja työkaluja malli tarjoaa selkeän ja kattavan kehyksen DevOps-käytäntöjen arvioinnille (Orozco-Garces, 2022).

5 Yhteenveto

Kun ohjelmistoprojektien kannalta tärkeimmät kriittiset menestystekijät on identifioitu, voidaan analysoida, miten menestystekijöiden oikea toteutus vaikuttaa suoraan projektin onnistumiseen varmistamalla tavoitteiden saavuttamisen ja asiakastyytyvyyden. Nämä tekijät täydentävät toisiaan ja korostavat teknisten, organisatoristen ja sosiaalisten taitojen integroinnin merkitystä optimaalisten tulosten saavuttamiseksi.

5.1 Tutkimuskysymykset ja johtopäätökset

Tässä kappaleessa vastataan kolmeen tutkimuskysymykseen. Vastauksiin on päädytty tutkimalla tieteellistä kirjallisuutta, kuten useita akateemisesti arvioituja artikkeleita.

1. Mitkä ovat tärkeimmät kriittiset menestystekijät onnistuneelle ohjelmistoprojektille?

Tutkimusten mukaan ohjelmistoprojektien tärkeimmät menestystekijät liittyvät vahvasti ihmillisiin tekijöihin, mutta myös teknisillä ja organisatorisilla tekijöillä on tärkeä vaikutus projektin onnistumiseen. Kappaleessa 4.1 tehdyn taulukon (Taulukko 2) mukaan tärkeimmäksi menestystekijäksi osoittautui tiimin kyvykkyys. Muita tärkeimpiä menestystekijöitä olivat johdon tuki ja osaaminen, asiakkaiden osallistaminen (UI ja UP), Selkeät vaatimukset ja tavoitteet, tehokas viestintä, hyvä suunnittelu ja tarkka seuranta ja valvonta. Lisäksi tutkimukset osoittavat, että tehokas tiimin vuorovaikutus ja selkeä viestintä sidosryhmien kanssa ovat olennaisia riskien ja viivästysten minimoimiseksi. Huolellinen suunnittelu ja jatkuva seuranta mahdollistavat ongelmien oikea-aikaisen tunnistamisen ja varmistavat projektin tavoitteiden saavuttamisen. Samoin johdon tuki ja aktiivinen asiakkaan osallistuminen edistävät odotusten ja lopputulosten tyydyttämisen yhdenmukaistamista. Nämä yhdistetyt tekijät korostavat teknisten,

organisatoristen ja pehmeiden taitojen integroinnin merkitystä ohjelmistoprojektien onnistumisen kannalta.

2. Miten nämä menestystekijät vaikuttavat projektin lopputulokseen ja onnistumiseen?

Kun tärkeimmät kriittiset menestystekijät ohjelmistoprojektien kannalta on tunnistettu, voidaan tutkia niiden vaikutuksia projektin lopputulokseen ja onnistumiseen. Kun muutamat kriittiset menestystekijät ovat kunnossa projektin onnistuminen on todennäköisempää (Lin ja muut, 2016). Projektin onnistuminen vaatii muun muassa odotusten ja vaatimusten täyttymistä, asiakkaan tyytyväisyyttä, johdon tukea ja projektijohtajan osaamista sekä tiimin kyvykkyyttä. Näihin tavoitteisiin päästään noudattamalla kriittisiä menestystekijöitä. Tunnistetut kriittiset menestystekijät eivät ainoastaan lisää projektin onnistumisen todennäköisyyttä, vaan myös parantaa lopputuotteen laatua ja asiakastyytyväisyyttä. Kun tiimillä on oikeat taidot ja vahva johdon tuki, ongelmanratkaisu ja tehokas päätöksenteko helpottuvat. Aktiivinen asiakasosallistuminen varmistaa, että vaatimukset ovat selkeät ja odotukset vastaavat tuotoksia. Huolellinen suunnittelu ja projektin jatkuva seuranta ovat myös tärkeää ja ne mahdollistavat riskien tunnistamisen aikaisin ja projektin mukauttamisen nopeasti, mikä parantaa merkittävästi onnistumisen mahdollisuuksia.

3. Mitkä ovat suurimmat haasteet ohjelmistoprojekteissa ja menestystekijöiden noudattamisessa ja mikä johtaa epäonnistumiseen?

Löydettyjen menestystekijöiden toteuttaminen on vaikeaa kuten Standish Groupin CHAOS-raportissa todettiin, ohjelmistoprojektien epäonnistumisten määrä on korkea (19%) vuonna 2023. Mircean ja Nguyenin (2023) mukaan ohjelmistoprojektien toteuttamiseen liittyy paljon ongelmia, kuten budjettien ylittyminen, suunnittelu ja aikatauluvirheet ja asiakastyytyväisyyden epäonnistuminen. Ohjelmistoprojektien hallintaprosessit ovat haastavia ja erityisen haastavaa on riskien hallinta eri

projektivaiheissa, mikä vaikuttaa kehittäjiin sekä asiakkaisiin (Mircea & Ngyuyen, 2023, s. 63). Erilaisten tutkittujen artikkelien mukaan epäonnistumiset harvoin johtuvat teknisistä tekijöistä vaan usein inhimillisiin tekijöihin liittyvistä tekijöistä. Tämä viittaa siihen, että onnistunut projektijohtaminen ihmisten johtamisen kannalta on tärkeää, mikä vaatii hyvää tiimityötä, asiakkaiden osallistamista ja tehokasta viestintää. Lisäksi selkeät tavoitteet ja vaatimukset, hyvä suunnittelu ja oikeanlainen metodi projektinhallintaan auttavat estämään haasteiden ilmenemistä. Suuremmassa kuvassa IT-projektien epäonnistumiset korreloivat myös ohjelmistoprojektien haasteisiin. Kärklinan ja Pirtan (2021, s. 54) artikkelin mukaan 2017 vuoden jälkeen onnistuneiden projektien osuus on kasvanut ketterien menetelmien, johtamisosaamisen ja teknisen osaamisen kehityksen takia, mutta Standish Groupin laajan CHAOS-raportin mukaan edelleen yli 50 000 ohjelmistoprojektista noin 19% epäonnistuu eikä valmistu (Mircea & Nguyen, 2023, s. 62).

Näiden havaintojen avulla voidaan perustella, että ohjelmistoprojektien epäonnistumiset johtuvat usein muista kuin teknisistä syistä, eli inhimillisistä ja organisatorisista syistä. Onnistuneiden ja epäonnistuneiden ohjelmistoprojektien erot syntyvät siitä, kuinka hyvin tärkeimmät menestystekijät toteutuvat käytännössä. Näitä ovat johdon tuki ja projektijohtajan osaaminen, selkeä viestintä, tiimityöskentely ja tiimin kyvykkyys, sidosryhmien osallistuminen ja realistiset tavoitteet ja suunnittelu.

5.2 Tulevaisuuden tutkimukset

Tulevaisuuden tutkimuksissa olisi tärkeää syventää osaamista ohjelmistoprojektien johtamisessa erilaisissa organisaatioissa ja toimintaympäristöissä. Empiirinen tutkimus olisi tarpeellista ja se voisi tarkastella esimerkiksi inhimillisten ja organisatoristen tekijöiden kuten tiimityöskentelyn, johtamismenetelmien, tehokkaan viestinnän ja sidosryhmien osallistamisen vaikutusta erilaisten kehitysmallien yhteydessä. Walkerin ja Lloyd-Walkerin (2019) artikkelissa *The future of the management of projects in the 2030s* todetaan, että teknologian kehityksen haittapuolena on se, että ne vaativat

jatkuvaa oppimista ja koulutusta niiden käytössä ja soveltamisessa. Tämä tarkoittaa, että nykyinen jatkuvan muutoksen ja ohjelmistopäivitysten ja kehityksen sykli todennäköisesti kiihtyy. Ohjelmistoprojektien johtaminen on siis entistä hankalampaa ja riskit epäonnistumisille voi lisääntyä. Tulevaisuuden tutkimuksessa olisi tärkeää tutkia, miten erilaisia johtamismenetelmiä pystyy kehittämään ja yhdistelemään parempien lopputulosten saavuttamiseksi. Lisäksi tutkimusta tulisi syventää tutkimalla syy-seuraus suhteita ohjelmistoprojektien onnistumisessa kysely- ja tapaustutkimusten avulla. Tekoälyn ja automatisaation lisääntyvä rooli teknologiassa ja projektinhallinnassa avaa uusia suuntia tutkimuksille, sillä ne voivat tukea monia menestystekijöiden osa-alueita projekteissa.

Tutkimukset perustuvat pitkälti kansainvälisiin aineistoihin ja yrityksiin, joten olisi hyödyllistä toteuttaa tutkimusta myös kansallisella tasolla tutkimalla esimerkiksi Suomessa toimivien organisaatioiden toimintatapoja ja kulttuurillisia tekijöitä. Tämä lisäisi ymmärrystä ohjelmistoprojektien onnistumisen parantamisesta kansallisella tasolla.

Kuvat

Kuva 1. Decision tree, joka auttaa valitsemaan onko Agile vai vesiputousmalli sopivampi projektille. (Mishra& Alzoubi, 2023).....	10
Kuva 2. järjestelmän kokonaisrakenne kaaviolausekkeena (Yuge & Badarch, 2023, s.4)	13
Kuva 3. Tutkijan tekemä kaavio ohjelmistoprojektin kulusta vaiheittain (Yuge & Badarch, 2023).....	14
Kuva 4. Ymypyraakaavio onnistuneista, haasteellisista ja epäonnistuneista ohjelmistoprojekteista Standish Groupin CHAOS-raportin mukaan.	17
Kuva 5. Kärklinan ja Pirtan luoma Flow-kartta projektin mittaamisesta ja arvioinnista (Kärklina & Pirta, 2021).....	25

Taulukot

Taulukko 1. Yhteenvedo tärkeimmistä menestystekijöistä ohjelmistoprojektissa.....	24
---	----

Lähteet

Singh (n.d.). International Journal of Latest Trends in Engineering and Technology, 273 noudettu 14.10. osoitteesta <https://www.ijltet.org/journal/150694929045.%201938.pdf?utm>

Emsley ja Altahtoo (2013) Issues in information Systems, 366–367 Noudettu 14.10 osoitteesta https://iacis.org/iis/2013/229_iis_2013_366-375.pdf?utm

Garousi ja muut (2018) Software Quality Journal, 429–443, Noudettu 14.10 osoitteesta <https://link.springer.com/article/10.1007/s11219-018-9419-5>

Das (2025) A Comparative Analysis of Agile vs. Traditional Project Management Approaches, 45 Noudettu 14.10. osoitteesta <https://siddhainternationalpublication.org/index.php/jhcsi/article/view/27>

Mishra ja Alzoubi (2023) Structured software development versus agile software development: a comparative analysis noudettu 22.10. osoitteesta <https://link.springer.com/article/10.1007/s13198-023-01958-5>

Stan ja muut (2025) Agile Project Management in Large Organizations: Challenges and Solutions, 2237, Noudettu 22.10. osoitteesta <https://sciendocom/article/10.2478/picbe-2025-0172>

Lucos (2025) The Impact of Leadership Styles on Agile Project Management in Global IT Sector, 1, Noudettu 22.10. osoitteesta <https://pmworldlibrary.net/wp-content/uploads/2025/09/pmwi156-Sep2025-Lucos-impact-of-leadership-styles-on-agile-project-management.pdf>

Jeteendre ja muut (2022) Project Failure in Software Project Management, 21, Noudettu 14.10. osoitteesta

<https://www.iosrjournals.org/iosr-jce/papers/Vol24-issue6/Ser-1/D2406012125.pdf>

Hongbo ja muut (2024,) Software project scheduling under activity duration uncertainty noudettu 14.10. osoitteesta

<https://research-ebsco-com.proxy.uwasa.fi/c/slwlh3/viewer/html/3jhm7cs6qj?auth-callid=fea8369d-62bb-437a-8640-3729d9962dd3>

Yuge ja Badarch (2023) Research on Contemporary Software Development Life Cycle Models, 1-5, Noudettu 15.10. osoitteesta

<https://www.sciencepublishinggroup.com/article/10.11648/j.ajcst.20230601.11>

Sitram ja Bhuvana (2011) A Qualitative Study on Educational Software Requirements Analysis– The Analysis Phase of SDLC Noudettu 15.10 osoitteesta

<https://research.ijcaonline.org/rtmc/number6/rtmc1042.pdf>

Heagney (2016) Fundamentals of project management Noudettu 25.10 osoitteesta

[https://books.google.fi/books?hl=fi&lr=&id=Vy58DAAAQBAJ&oi=fnd&pg=PR2&dq=Heagney+\(2016\)+Fundamentals+of+project+management+&ots=7qjbOWYI06&sig=i5Lm89e_weeHxzz6KhLtJvqzSfl&redir_esc=y#v=onepage&q=Heagney%20\(2016\)%20Fundamentals%20of%20project%20management&f=false](https://books.google.fi/books?hl=fi&lr=&id=Vy58DAAAQBAJ&oi=fnd&pg=PR2&dq=Heagney+(2016)+Fundamentals+of+project+management+&ots=7qjbOWYI06&sig=i5Lm89e_weeHxzz6KhLtJvqzSfl&redir_esc=y#v=onepage&q=Heagney%20(2016)%20Fundamentals%20of%20project%20management&f=false)

Tam ja Oliveira (2020) The factors influencing the success of on-going agile software development projects noudettu 23.10 osoitteesta

https://www.sciencedirect.com/science/article/pii/S0263786320300089?casa_token=jfQ6j9oW1xcAAAAA:2thb9i8CnluZd65eEMZs3hKP5tzouQ7GBswmtWFG-fp8tww5t3q3Gt9nb9nDi7-jtVMKt3apCHhF

Chow ja Cao (2008) A survey study of critical success factors in agile software projects

noudettu 23.10. osoitteesta

<https://www.sciencedirect.com/science/article/abs/pii/S0164121207002208>

Alias (2014) Determining Critical Success Factors of Project Management Practice: A

conceptual framework noudettu 22.10. osoitteesta

<https://pdf.sciencedirectassets.com/277811/1-s2.0-S1877042814X00480/1-s2.0-S1877042814054834/main.pdf>

Wolff ja muut (2024) A comparison of soft factors in the implementation and adoption of digitalization projects Noudettu osoitteesta

<https://aisel.aisnet.org/ijispm/vol12/iss2/5/>

Bogopa & Marnewick (2022) Critical Success Factors in Software Development

Projects) Noudettu osoitteesta

https://www.researchgate.net/publication/362193172_Critical_success_factors_in_Software_Development_Projects

Nasir & Sahibuddin (2011) Critical Success Factors for Software Projects: A

Comparative Study Noudettu osoitteesta

https://www.researchgate.net/publication/283749993_Critical_success_factors_for_software_projects_A_comparative_study

Sudhakar (2012) A Model of Critical Success Factors for Software Projects Noudettu

osoitteesta

<https://www.emerald.com/jeim/article/25/6/537/199121/A-model-of-critical-success-factors-for-software>

Reiff & Schlegel (2021) Hybrid project management, 45–51 Noudettu 24.10. osoitteesta

<https://revistas.uminho.pt/index.php/ijispm/article/view/4084/4295>

Chiyangwa (2017) Modelling the critical success factors of agile software development projects in South Africa Noudettu 24.10. osoitteesta

<https://research-ebsco-com.proxy.uwasa.fi/c/slwlh3/search/details/vh4w6mzh5?db=afh&limiters=RV%3AY&q=Critical%20success%20factors%20AND%20agile&searchMode=boolean>

Kārklīņa & Pirta (2018) Quality Metrics in Agile Software Development Projects, 54–57, Noudettu 29.10. osoitteesta

<https://archive-journals.rtu.lv/itms/article/view/itms-2018-0008/pdf>

Mircea & Nguyen (2023) Expert system research and usage in IT project management, 62–63, Noudettu 1.11. osoitteesta

https://mts.intechopen.com/storage/books/11260/authors_book/authors_book.pdf

Orozco-Garces (2022) Model to Complement the Evaluation of DevOps in Software Companies Noudettu 7.11. osoitteesta

<https://www.redalyc.org/journal/4139/413974254002/movil>

Setiawan (2025) Critical Success Factors and Challenges in Applying Agile Project Management Noudettu 10.11. osoitteesta

<https://ijsoc.goacademica.com/index.php/ijsoc/article/view/1379/1183>

Lin ja muut (2016) Analysis of the CSFs affecting project execution noudettu 11.11 osoitteesta

<https://www.tandfonline.com/doi/full/10.1080/14783363.2016.1245581#d1e199>

Gheni (2025) Perceived Critical Success Factors for Implementing Building Information Modelling in Construction Small- and Medium-Sized Enterprises Noudettu 9.11 osoitteesta

<https://www.mdpi.com/2673-4109/6/1/5>

Besteiro & Pinto (2015) Success factors in project management Noudettu 29.10. osoitteesta

<https://research.ebsco.com/c/slwlh3/viewer/pdf/dvcgbge6cf>

El-Deeb (2022) Major Challenges Currently Facing the Software Industry, 15, Noudettu 13.11 osoitteesta

<https://dl.acm.org/doi/pdf/10.1145/3539814.3539818>

Nizam (2022) Software Project Failure Process Definition Noudettu 13.11 osoitteesta

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9743906>

Walker & Lloyd-Walker (2019) The future of the management of projects in the 2030s Noudettu 13.11 osoitteesta

<https://www.proquest.com/abicomplete/docview/2239133522/C931233A13D84FFE9Q/12?accountid=14797&parentSessionId=IhMHdkG%2Fnh9CZbFQrnqArLVPsEVVe9879TfynHbge1A%3D&sourcetype=Scholarly%20Journals>

Kerzner ja muut (2022) Project Management Next Generation, 64, Noudettu 10.11
osoitteesta

[https://www.google.fi/books/edition/Project_Management_Next_Generation/
5050EAAAQBAJ?hl=fi&gbpv=0](https://www.google.fi/books/edition/Project_Management_Next_Generation/5050EAAAQBAJ?hl=fi&gbpv=0)