

Annika Pajula

KÄYTETTÄVYYDEN VAIKUTUS OHJELMISTOTUOTTEEN ELINKAAREN KUSTANNUKSIIN

Systemaattinen kirjallisuuskatsaus

Informaatioteknologian ja viestinnän tiedekunta

Kandidaatintyö

Kesäkuu 2020

TIIVISTELMÄ

Annika Pajula: Käytettävyyden vaikutus ohjelmistotuotteen elinkaaren kustannuksiin
Kandidaatintyö
Tampereen yliopisto
Tieto- ja sähkötekniikan tutkinto-ohjelma
Kesäkuu 2020

Digitalisoituneessa maailmassa ohjelmistojen käytettävyys on erityisen tärkeää. Ennen ohjelmistoja kehitettiin ammattilaiselta ammattilaiselle, mutta nyt ohjelmistot ja digitaaliset tuotteet ovat osa kaikenlaisten ihmisten työtä ja arkea. Silti käytettävyyteen saatetaan edelleen asennoitua kuin se olisi jotain ylimääräistä, jonka voi resurssien puuttuessa unohtaa. Tässä työssä tutkitaan käytettävyyden vaikutusta ohjelmistotuotteen kustannuksiin. Työn tavoite on selvittää, miten käytettävyyteen panostamalla voidaan säästää ohjelmistotuotteen elinkaaren kustannuksissa.

Tutkimusmenetelmänä on systemaattinen kirjallisuuskatsaus. Työssä perehdytään käytettävyyden teoriaan sekä sen mittaamiseen, suunnitteluun, mallinnukseen ja arviointiin, ja käsitellään tutkimusaineistoa käytettävyyden kustannusvaikutuksista. Aineistosta etsitään käytettävyyden hyötyjä, joiden voidaan arvioida edesauttavan säästöjen syntymistä tuotteen elinkaaren aikana, sekä malleja, joilla käytettävyyden tuomaa säästöä voidaan laskea.

Tutkimus osoittaa, että käytettävyydellä on paljon sellaisia hyötyjä, jotka voivat aikaansaada säästöä ohjelmistotuotteen elinkaaren aikana. Näitä hyötyjä ovat muun muassa käyttäjän aikaansaavuuden kasvaminen, käyttäjän tekemien virheiden väheneminen, käyttäjätuottavuuden kasvaminen, koulutustarpeen väheneminen, käyttäjätuen tarpeen väheneminen, henkilöstön poisolojen ja vaihtuvuuden väheneminen, korkeampi moraalit, pienemmät dokumentaatiokustannukset, oikeudenkäyntikuluilta välttyminen, järjestelmien sabotoinnilta välttyminen, luovan ongelmanratkaisun helpottuminen, myynnin kasvaminen, kilpailuetu, viime hetken muutoksilta ja yllättäviltä kuluilta välttyminen ohjelmistokehityksessä, palvelun laadun paraneminen sekä asiakastyytyväisyyden kasvaminen. Mainittuihin hyötyihin kohdistuu myös kritiikkiä. Hyödyt esitetään kirjallisuudessa usein helposti havaittavina, jolloin käytettävyyden tuloksiin voi kohdistua epärealistisia odotuksia. Todellisuudessa voi mennä kauankin, ennen kuin käytettävyyteen panostamisen hyödyt ovat nähtävissä, ja joissain tapauksissa voi olla vaikea arvioida mitä olisi tapahtunut, jos käytettävyyttä ei olisi tehty.

Työssä esitellään viisi aineistosta löytynyttä kustannus-hyötymallia, joilla käytettävyyden kustannuksia ja hyötyjä voidaan laskea. Eri tyyppisillä hyödyillä on vaihtelevan suuruinen merkitys eri tahoille, ja työssä käsitellään, kuinka laskelmassa voidaan valita tarkoitukseen sopivat hyötykategoriat. Mallit eivät eroa toisistaan merkittävästi: pohjimmiltaan kaikissa on ideana verrata käytettävyyden kustannuksia käytettävyyden aikaansaamiin hyötyihin.

Avainsanat: käytettävyys, ohjelmisto, kustannus-hyötyanalyysi, kustannus, hyöty

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

SISÄLLYSLUETTELO

1	Johdanto	1
2	Tausta	2
2.1	Käytettävyys	2
2.2	Käytettävyyden mittarit	4
2.3	Käytettävyystyön menetelmät	5
2.4	Ohjelmiston elinkaari	8
3	Tutkimusmenetelmät ja aineisto	9
3.1	Tutkimusprosessi	9
3.2	Tutkimusaineisto	11
4	Tulokset	13
4.1	Käytettävyyden hyödyt kustannusnäkökulmasta	13
4.1.1	Sisäinen näkökulma	14
4.1.2	Ulkoinen näkökulma	15
4.2	Käytettävyyden kustannushyödyn laskeminen	16
4.2.1	Mantein ja Teoreyn malli	17
4.2.2	Mayhew'n ja Tremainen malli	19
4.2.3	Donahuen malli	22
4.2.4	Sorflatenin malli	24
4.2.5	Rajperin malli	25
4.2.6	Esimerkkitapauksia	28
5	Yhteenveto	30
	Lähteet	33
	Liite A Tutkimusaineiston haku- ja sisäänottoprosessi tietokannoittain	35
	Liite B Kustannus-hyötylaskelmien vaiheiden yhteenveto	36

KUVALUETTELO

2.1 Käytettävyyden osa-alueet	3
2.2 Käytettävyyden mittarit	5
2.3 Ohjelmiston elinkaari	8
3.1 Tutkimusaineiston sisäänotto- ja poissulkuprosessi	10
3.2 Tutkimusaineisto julkaisuvuoden mukaan	12
4.1 Käytettävyyden kustannus-hyötyanalyysin hyötykategoriat	21
5.1 Käytettävyyden hyödyt kustannusnäkökulmasta	31
A.1 Tutkimusaineiston haku- ja sisäänottoprosessi tietokannoittain	35
B.1 Yhteenveto kustannus-hyötylaskelmien vaiheista.	36

TAULUKKOLUETTELO

2.1	Nielsenin heuristiikat	7
3.1	Katsaukseen valitut tutkimukset	11
4.1	Käyttäjätutkimuksen kustannusten laskeminen	17
4.2	Käytettävyystestauksen kustannusten laskeminen	20
4.3	Esimerkkiarvioita hyödyistä	22
4.4	Esimerkki hyödyn laskemisesta: käyttäjän aikaansaavuuden kasvaminen	22
4.5	Kustannuslaskelman parametrit Donahuen esimerkissä	23
4.6	Kustannuslaskelman parametrit Rajperin ym. mallissa	26
4.7	Potentiaalisten hyötyjen laskeminen Rajperin ym. mallissa	27

LYHENTEET JA MERKINNÄT

ACM	Association for Computing Machinery
COCOMO	Constructive Cost Model
IEEE	Institute of Electrical and Electronics Engineers
ISO	Kansainvälinen standardointiorganisaatio (engl. International Organization for Standardization)
QUIS	Questionnaire for User Interaction Satisfaction
SUMI	Software Usability Measurement Inventory
SUS	System Usability Scale
TAU	Tampereen yliopisto (engl. Tampere University)
TUNI	Tampereen korkeakouluyhteisö (engl. Tampere Universities)
UEM	Käytettävyystyön menetelmät (engl. Usability Engineering Methods)

1 JOHDANTO

Käytettävyyden merkitys tunnistetaan, mutta toisinaan siihen jätetään panostamatta, koska ajatellaan, että riittää, kunhan tuote toimii ja että sillä pystyy suorittamaan tietyt toiminnot. Joskus käytettävyyteen panostaminen nähdään ylimääräisenä vaivana ja kuluna (Marcus 2005). Tällöin ei oteta huomioon, mitä negatiivista käytettävyyden laiminlyömisestä voi seurata. Ohjelmistojen huono käytettävyys aiheuttaa piilokustannuksia esimerkiksi lisääntyneiden virheiden ja ongelmien ratkomiseen kuluvan ajan muodossa, puhumattakaan käyttäjän kokemasta haitasta.

Vaikeakäyttöisten ohjelmistojen oppiminen on hidasta ja työlästä. Yksityiskohdat unohtuvat, jos käyttökertojen välillä kuluu aikaa. Virheitäkin tulee paljon, ja käyttäjät turhautuvat. Käytettävyyttä parantamalla voidaan lisätä käyttäjätyytyväisyyttä ja edistää käytön oppimista. Se on tärkeää erityisesti käyttäjän kannalta, mutta hyvällä käytettävyydellä voi olla yllättävä vaikutus myös kustannuksiin, kun tarkastellaan kustannuksia ohjelmiston elinkaaren ajalta.

Tämän kandidaatintyön tutkimuskysymyksenä on, **miten käytettävyyteen panostamalla voidaan säästää ohjelmistotuotteen elinkaaren kustannuksissa**. Työssä etsitään malleja, joilla käytettävyyden tuomaa säästöä voidaan laskea, sekä myös käytettävyyden hyötyjä, joiden voidaan olettaa edesauttavan säästöjen syntymistä tuotteen elinkaaren aikana. Tutkimusmenetelmänä on systemaattinen kirjallisuuskatsaus.

Luvussa 2 perehdytään käytettävyyden ja ohjelmiston elinkaaren teoriaan. Luvussa 3 käydään läpi tutkimusmenetelmä ja -aineisto ja luvussa 4 esitellään tulokset. Tulosten käsittely on jaettu kahteen alalukuun: luvussa 4.1 käsitellään käytettävyyden hyötyjä kustannusnäkökulmasta ja luvussa 4.2 käydään läpi malleja käytettävyyden kustannushyödyn laskemiseen. Luvussa 5 on yhteenveto.

2 TAUSTA

Jotta voidaan arvioida käytettävyyden vaikutusta ohjelmistotuotteen elinkaaren kustannuksiin, on tunnettava keskeisiä käsitteitä. Niitä ovat käytettävyyden lisäksi esimerkiksi käyttäjäkokemus, käytettävyyden menetelmät ja käytettävyyden mittarit sekä ohjelmiston elinkaari. Tässä luvussa esitellään ensin peruskäsitteitä, kuten käytettävyys ja käyttäjäkokemus, ja niistä edetään käytettävyyden suunnitteluun, mallinnukseen, arviointiin ja mittaamiseen. Lopuksi esitellään vielä ohjelmiston elinkaari.

2.1 Käytettävyys

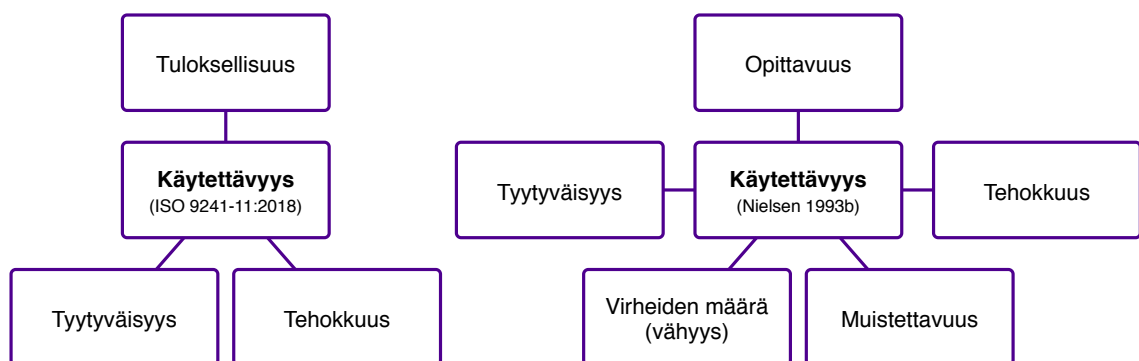
Käytettävyys (engl. usability) määritellään ISO-standardissa siten, missä määrin *tuloksellisuus*, *tehokkuus* ja *tyytyväisyys* toteutuvat, kun järjestelmää, tuotetta tai palvelua käytetään määriteltyjen tavoitteiden saavuttamiseen tietyssä käyttökontekstissa. Tuloksellisuus (engl. effectiveness) kattaa oikeastaan kaksi asiaa: missä määrin toteutunut lopputulos vastaa suunniteltua lopputulosta ja missä määrin käyttäjä kykenee saavuttamaan kaikki suunnitellut lopputulokset. Tehokkuus (engl. efficiency) tarkoittaa kaikkia niitä resursseja, joita tarvitaan lopputuloksen saavuttamiseksi. Resurssit voivat olla aikaa, vaivannäköä, rahaa tai materiaaleja. Tyytyväisyys (engl. satisfaction) on sitä, missä määrin järjestelmän, tuotteen tai palvelun käyttämisestä syntyvät fyysiset, kognitiiviset ja emotionaaliset vasteet kohtaavat käyttäjän tarpeet ja odotukset. (ISO 9241-11:2018)

Otetaan määritelmän tueksi esimerkki junalipun ostamisesta verkkokaupasta. Käyttäjä haluaa ostaa junalipun kaupungista A kaupunkiin B, ja istumapaikan lisäksi hän tarvitsee paikan myös polkupyörää varten. Käyttäjälle on tärkeää, että polkupyöräpaikka on samassa vaunussa kuin hänen istumapaikkansa, ja junassa hän istuu mieluiten käytävän puolella kasvot menosuuntaan päin. Käyttäjä törmää junalippua varatessa kahteen pulmaan: hän saa valittua tilaukseensa polkupyöräpaikan, mutta hän ei löydä, mistä saisi valittua polkupyöräpaikan juuri siitä vaunusta, jossa hän aikoo istua. Myös istumapaikan valinnassa on ongelma: käyttäjä saa valittua toivomansa käytäväpaikan, mutta hän ei tiedä, onko paikan suunta kasvot vai selkä menosuuntaa kohti. Käyttäjä vie tilauksen loppuun, mutta soittaa sitten asiakaspalveluun, koska ei onnistunut istumapaikan suunnan ja polkupyöräpaikan sijainnin valinnassa.

Pohditaan käytettävyyden toteutumista tuloksellisuuden, tehokkuuden ja tyytyväisyyden näkökulmasta. Tuloksellisuus toteutuu, jos toteutunut lopputulos vastaa suunniteltua ja jos kaikki suunnitellut lopputulokset saavutetaan: Käyttäjä sai junalipun ja polkupyöräpai-

kan ostettua, mutta pyöräpaikan sijainnista hänellä ei ole tietoa, sillä hän ei löytänyt, missä valinnan voi tehdä. Käyttäjä sai varattua käytäväpaikan, mutta ei toivotulla tavalla, sillä hän joutui arvaamaan, onko paikka suunnattu kasvot vai selkä menosuuntaa kohti. Tuoksellisuus toteutuu siis vain osittain: lippu saatiin ostettua sekä käyttäjälle itselleen että polkupyörälle ja istumapaikka saatiin varattua käytävän puolelta, mutta polkupyöräpaikan sijainti ja istumapaikan suunta jäivät valitsematta. Tehokkuus ei myöskään toteudu, sillä käyttäjä joutui käyttämään paljon aikaa ja vaivaa saadakseen varauksen tehtyä. Lopuksi käyttäjä päätyi soittamaan asiakaspalveluun varmistaakseen istumapaikan suunnan ja polkupyöräpaikan sijainnin. Asiakaspalveluun soittaminen maksaa, eli käyttäjä joutuu siis ajan ja vaivannäön lisäksi käyttämään myös rahaa. Tyytyväisyys ei myöskään toteudu täysin. Tilausta tehdessä käyttäjää harmitti, sillä hän halusi ostaa lipun nopeasti ja sujuvasti itse, mutta joutuikin käyttämään siihen enemmän aikaa ja pyytämään apua asiakaspalvelusta. Lopuksi käyttäjä kuitenkin on tyytyväinen, sillä hänellä on nyt tarvitsemansa lippu ja paikkavaraukset.

Nielsenin (1993b) mukaan käytettävyys koostuu seuraavista osa-alueista: opittavuus, tehokkuus, muistettavuus, virheiden määrä (vähyys) ja tyytyväisyys. Yhteisiä osa-alueita Nielsenin ja ISO-standardin määritelmillä ovat tehokkuus ja tyytyväisyys, joskin ne kuvailaan määritelmässä hieman eri tavoin. Käytettävyyden osa-alueet näiden kahden määritelmän mukaan on havainnollistettu kuvassa 2.1.



Kuva 2.1. Käytettävyyden osa-alueet, vasemmalla ISO-standardin mukaan (ISO 9241-11:2018) ja oikealla Nielsenin mukaan (Nielsen 1993b).

Opittavuus tarkoittaa sitä, että järjestelmän käyttämisen pitäisi olla helppoa oppia, jotta käyttäjä pääsee nopeasti käyttämään järjestelmää sen varsinaiseen tarkoitukseen. *Tehokkuus* taas kuvaa sitä, että kun käyttäjä on oppinut järjestelmän käytön, sitä on mahdollista käyttää tehokkaasti ja tuotteliaasti. *Muistettavuus* tarkoittaa sitä, että kun käyttäjä palaa takaisin järjestelmän pariin, käytön muistaminen on helppoa ja käyttöä ei tarvitse opetella alusta asti uudelleen. *Virheiden määrä (vähyys)* kuvaa sitä, että järjestelmää käyttäessä käyttäjän tekemiä virheitä on vain vähän tai ei ollenkaan. Virheiden määrään kuuluu myös se, että virhetilanteessa käyttäjä pystyy helposti palautumaan virheestä, eikä "katastrofaalisia" virheitä saa esiintyä. *Tyytyväisyys* tarkoittaa sitä, että käyttäjät kokevat järjestelmän käyttämisen miellyttävänä. (Nielsen 1993b)

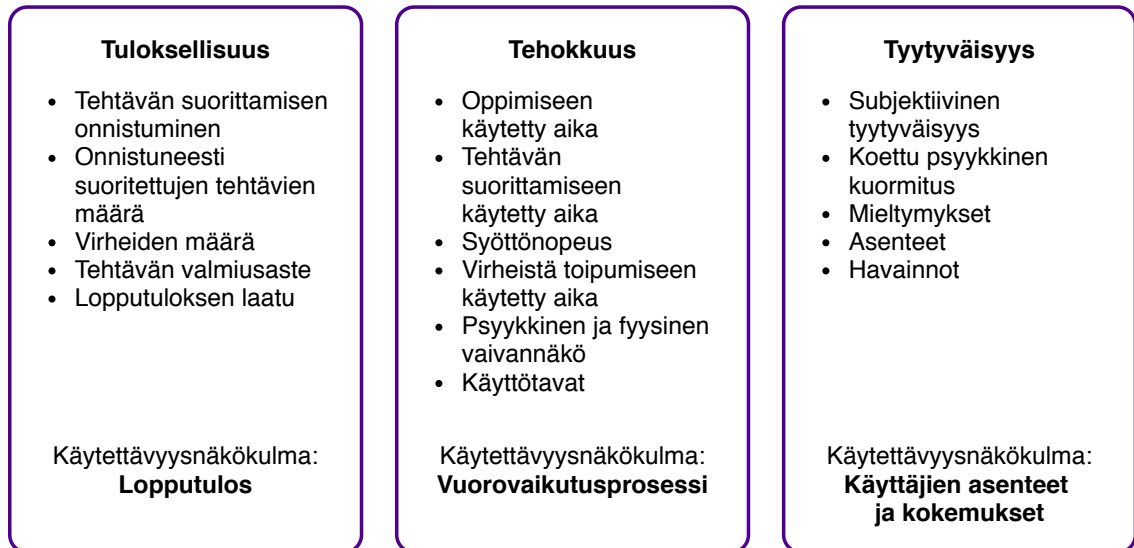
Tarkastellaan Nielsenin määritelmää junalippuesimerkin avulla. Käyttäjä onnistuu tilaamaan lipun, johon kuuluu yksi istumapaikka käytävän puolelta sekä varaus polkupyörälle. Käyttäjä onnistuu kaikessa helposti ja ilman apua lukuun ottamatta polkupyöräpaikan sijainnin valintaa ja istumapaikan suunnan valintaa. Kuitenkin muut ostotapahtuman osat alueet, kuten omien tietojen täyttäminen ja lipun maksaminen, sujuvat käyttäjältä kuin itsestään. Voidaan sanoa, että näiltä osin opittavuus toteutuu. Jotta voitaisiin kunnolla arvioida Nielsenin määritelmän mukaisen tehokkuuden toteutumista, pitäisi tietää, onko käyttäjä varannut junalippua aikaisemmin ja onko hän aikaisemmin varannut istumapaikan lisäksi myös polkupyöräpaikkaa. Oletetaan, että netissä varaaminen oli käyttäjälle uutta. Tehokkuutta voisi paremmin arvioida seuraavalla käyttökerralla – onko junalipun ostaminen mahdollista tehdä tehokkaasti ja tuottavasti, kun tietää, miten se tehdään. Muistettavuus nivoutuu myös tähän: kun käyttäjä tulee vaikkapa puolen vuoden kuluttua varaamaan seuraavaa junalippuaan, onnistuuko se helposti vai joutuuko käyttäjä opettelemaan samoja asioita uudestaan.

Käytettävyys on osa käyttäjäkokemusta. Käyttäjäkokemus (engl. user experience, UX) on käytettävyyttä laajempi käsite, joka kuvaa käyttäjän tunteita, uskomuksia, mieltymyksiä, aistimuksia, viihtyvyyttä, käyttäytymistä ja aikaansaannoksia, jotka voivat esiintyä ennen käyttöä, käytön aikana tai käytön jälkeen (ISO 9241-210:2019). Tässä työssä keskitytään käytettävyteen.

2.2 Käytettävyyden mittarit

Käytettävyyden mittariksi ei riitä, että epämääräisesti todetaan tuotteen olevan helppo käyttää. Hornbæk (2006) käy läpi 180 tutkimuksessa esiintyvät käytettävyyden mittarit, jotka hän luokittelee käytettävyyden määritelmään nojaten tuloksellisuuden, tehokkuuden ja tyytyväisyyden mukaan. Tuloksellisuuden mittareita ovat esimerkiksi tehtävän suorittamisen onnistuminen (joko onnistunut tai epäonnistunut), onnistuneesti suoritettujen tehtävien määrä, virheiden määrä, valmiusaste ja lopputuloksen laatu. Tehokkuuden mittareita taas ovat sekä oppimiseen että tehtävän suorittamiseen käytetty aika, syöttönopeus, virheistä toipumiseen käytetty aika, psyykkinen ja fyysinen vaivannäkö ja käyttötavat (engl. usage patterns). Tyytyväisyyden mittareita ovat esimerkiksi subjektiivinen tyytyväisyys, koettu psyykkinen kuormitus, mieltymykset, asenteet ja havainnot. Mittarit on havainnollistettu kuvassa 2.2.

Käytettävyyden osa-alueet eli tuloksellisuus, tehokkuus ja tyytyväisyys voidaan Hornbækin mukaan jaotella niin, että tuloksellisuus viittaa lopputulokseen, tehokkuus viittaa vuorovaikutusprosessiin ja tyytyväisyys viittaa käyttäjien asenteisiin ja kokemuksiin. Hän esittää, että käytettävyydelle on sekä objektiivisia että subjektiivisia mittareita. Objektiivisia mittareita ovat muun muassa asiantuntija-arvioinnit, aika, käyttötavat, opittavuus ja fyysinen käytettävyys, kun taas subjektiivisia mittareita ovat esimerkiksi käyttäjän havainnot lopputuloksesta, subjektiivisesti koettu tehtävään kulunut aika, psyykkinen kuormitus, tehtävän koettu hankaluus sekä validoidut kyselylomakkeet.



(Hornbæk 2006)

Kuva 2.2. Käytettävyyden mittareita. (Hornbæk 2006)

Hornbæk toteaa, että sopivien käytettävyyden mittareiden valinta on vaikeaa. Käytettävyytutkimuksen luotettavuus kärsii, jos mittareita tulkitaan väärin tai jos mittarit on valittu liian suppeasti ja tarkoitukseen sopimattomasti. Mittareita valitessa olisi hyvä varmistaa, että sekä objektiivisia että subjektiivisia mittareita otetaan mukaan. Niistä voidaan vetää erilaisia johtopäätöksiä tutkittavan kohteen käytettävyydestä.

2.3 Käytettävyyden menetelmät

Käytettävyysohjon on lukuisia eri menetelmiä (engl. Usability Engineering Methods, UEM), ja tässä luvussa paneudutaan niistä vain muutama. Ovaska, Aula ja Majaranta (2005) jaottelevat käytettävyytutkimuksen menetelmät suunnitteluun, mallinnukseen ja arviointiin tarkoitettuihin menetelmiin, mutta he lisäävät, että rajat pääluokkien välillä eivät ole tarkkoja, ja että jotkut menetelmät kuuluvat samanaikaisesti moneen vaiheeseen. Arviointimenetelmät voidaan vielä jakaa tarkistus- ja testausmenetelmiin sen mukaan, onko käyttäjä mukana arvioinnissa vai ei. Menetelmän valintaan vaikuttaa sen soveltuvuus esimerkiksi käyttökontekstiin, käyttäjäryhmään ja käytettävyystavoitteisiin. Myös käytettävissä olevat resurssit vaikuttavat. ISO-standardissa (ISO 9241-11:2018) esitellään viisi lähestymistapaa käytettävyyden arviointiin, ja tässä luvussa annetaan jokaisesta lähestymistavasta käytännön esimerkkejä:

1. arvioitavan kohteen vertailu määriteltyyn kriteeristöön
2. tutkitaan, mitä mahdollisia käytettävyyso ongelmia voi nousta esiin kun tehtävää yritetään suorittaa
3. käyttäjän käyttäytymisen havainnointi käytettävyyso ongelmien tunnistamiseksi testi ympäristössä tai aidossa käyttötilanteessa
4. käyttäjän suorituskyvyn mittaaminen: tuloksellisuus ja tehokkuus testi ympäristössä

tai aidossa käyttötilanteessa

5. käyttäjätyytyväisyyden mittaaminen testiympäristössä tai aidossa käyttötilanteessa

Haastattelu on tiedonkeruumenetelmä, jota voidaan käyttää kaikissa käytettävyyden vaiheissa. Se on erityisen tehokas yhdistettynä muihin menetelmiin, kuten tarkkailuun tai käytettävyydestaustaan. (Vuorela 2005) Haastattelu sopii etenkin alustavaan käyttäjiin ja käyttökontekstiin tutustumiseen. Tuloksena on laadullista aineistoa, jota voidaan käyttää suunnittelun tukena. (Ovaska, Aula ja Majaranta 2005) Haastattelun voi toteuttaa yksilö- tai ryhmähaastatteluna.

Kyselylomakkeet voivat käytettävyydetutkimuksessa täydentää muita menetelmiä, tai niitä voidaan käyttää ainoana menetelmänä tiedon keräämiseen (Vanhala 2005). Kysymysten tyypistä riippuen kyselylomakkeet voivat olla hyvinkin kustannustehokas menetelmä ja niiden avulla voidaan tavoittaa laaja joukko käyttäjäryhmän edustajia. Laadullisten kysymysten analysointiin tarvitaan enemmän resursseja kuin määrällisten kysymysten analysointiin. Kyselyn suunnittelussa on monia sudenkuoppia, ja onkin viisasta aloittaa kyselytutkimuksen suunnittelu selvittämällä, sopisiko jokin valmis käytettävyydetutkimus tarkoitukseen. Standardoitujen kyselylomakkeiden käyttäminen varmistaa myös tulosten vertailukelpoisuuden. Tällaisia valmiita kyselylomakkeita ovat esimerkiksi SUMI, QUIS ja SUS. (Vanhala 2005)

Heuristinen evaluointi on asiantuntija-arviointi, jossa kohdetta arvioidaan heuristiikkalistojen avulla. Tunnetuimpana heuristiikkalistana voidaan pitää Nielsenin heuristiikkoja (Nielsen 1993b), jotka on esitelty taulukossa 2.1. Menetelmä on toteutettavissa matalalla kynnyksellä ja arvioinnin suorittaminen on nopeaa. Huono puoli on se, että loppukäyttäjää ei osallisteta mukaan, ja onkin havaittu, että heuristisessa evaluoinnissa nousee esiin eri ongelmia kuin käytettävyydestaustassa. Evaluoinnin suorittaa yksi tai useampi asiantuntija. (Korvenranta 2005) Yksittäiseltä asiantuntijalta jää Nielsenin mukaan suurin osa ongelmista löytämättä, ja siksi heuristinen evaluointi olisikin parasta toteuttaa niin, että useampi asiantuntija on mukana. Optimaalisin tulos saavutetaan, kun arviota on tekemässä 3-5 asiantuntijaa. Tätä suurempi määrä ei enää oleellisesti paranna arvioinnin kattavuutta. (Nielsen 1993b) Toisaalta tätä lukumäärää on myös kritisoitu. Arvioijan kyvytkin vaikuttavat löytyvien ongelmien määrään. Muita heuristiikkalistoja ovat esimerkiksi Schneidermanin säännöt, erilaiset tyyliohjeistukset sekä standardit (Korvenranta 2005). Heuristiikkoja ei käytetä pelkästään arvioinnissa, vaan myös suunnittelun tukena.

Kognitiivinen läpikäynti on myös asiantuntija-arviointi. Siinä arvioija selvittää käyttöliittymän oppimisen helppoutta kysymysten avulla, joihin arvioija vastaa tehtävän jokaisessa vaiheessa. Kuten heuristisessa evaluoinnissakin, arvioinnin tarkkuus paranee, jos arvioijia on useampi. Kognitiivista läpikäyntiä varten käyttöliittymästä on oltava olemassa vähintään prototyyppi ja sen on oltava mahdollisimman todenmukainen, jotta tulokset olisivat luotettavia. Läpikäynnin suunnittelussa on selvitetään käyttäjäryhmät, suunnitellaan läpikäytävät tehtävät ja muodostetaan skenaario pohjustamaan tehtävien suoritusta. (Ranne 2005)

Taulukko 2.1. Nielsenin heuristiikat (Nielsen 1993b). Vasemmalla alkuperäinen heuristiikka ja oikealla sen tulkinta suomeksi (Korvenranta 2005).

Nielsenin heuristiikka	Suomenkielinen tulkinta
Visibility of system status	Palvelun tilan näkyvyys
Match between the system and the real world	Palvelun ja tosielämän vastaavuus
User control and freedom	Käyttäjän kontrolli ja vapaus
Consistency and standards	Yhteneväisyys ja standardit
Error prevention	Virheiden estäminen
Recognition rather than recall	Tunnistaminen mieluummin kuin muistaminen
Flexibility and efficiency of use	Käytön joustavuus ja tehokkuus
Aesthetic and minimalist design	Esteettinen ja minimalistinen suunnittelu
Helping users recognise, diagnose and recover from errors	Virhetilanteiden tunnistaminen, ilmoittaminen ja korjaaminen
Help and documentation	Opastus ja ohjeistus

Prototyypien avulla voidaan mallintaa tuotesuunnitelma konkreettiseen muotoon. Prototyyppi voi olla karkea (engl. low-fidelity) tai täsmällinen (engl. high-fidelity). (Ovaska, Aula ja Majaranta 2005) Karkeassa prototyypissä voi olla muutama luonnos tärkeimmistä näytöistä. Suunnitteluvaiheessa karkeat prototyypit ovat hyödyllisiä, sillä niiden tekeminen on nopeinta, ja ne auttavat palautteen saamisessa ja idean kommunikoinnissa. Karkeasta prototyypistä pidemmälle viety prototyyppi on paperiprototyyppi, jonka avulla käyttäjä voi testata vuorovaikutusta tuotteen kanssa. (usability.gov 2020) Täsmällinen prototyyppi taas on prototyyppiohjelmistolla kehitetty, klikattava prototyyppi, joka muistuttaa jo huomattavasti enemmän lopullista tuotetta, ja sopii hyvin käytettävyydestäukseen ja idean esittelyyn.

Käytettävyydestä on menetelmä, jossa käyttäjät testaavat tuotetta tai prototyyppiä tehtävien avulla. Tehtävät simuloivat aitoa käyttötilannetta ja testikäyttäjiksi valitaan ihmisiä, jotka edustavat tuotteen käyttäjäryhmää tai -ryhmiä. Testauksessa käytetään äänenajattelutekniikkaa: käyttäjää pyydetään ajattelemaan ääneen kun hän suorittaa tehtävää. Moderaattori tarkkailee testitilanteessa käyttäjän toimintaa ja ohjaa testin läpivientiä, ja lisäksi tilannetta seuraa vielä yksi tai useampi tarkkailija. Perinteisesti käytettävyydestäus toteutetaan siihen suunnitellussa laboratoriossa, jossa testitila ja tarkkailutila ovat erilliset, ja jossa on hyvät puitteet äänen ja kuvan tallennukseen. Huoneiden välissä on peililasi, josta tarkkailijat näkevät sisään testitilaan, mutta testitilasta ei nää tarkkailutilaan. Testitilanne tallennetaan, jotta käyttäjän käyttäytymistä voidaan analysoida. Laboratorion käyttämiseen ei aina ole mahdollisuutta, ja käytettävyydestäusta voi tehdä ilman laboratoriotakin, vaikka testaukseen suunniteltu tila tarjoaakin etuja. Toisaalta on myös olemassa käyttötilanteita, joita ei ole mahdollista simuloida laboratoriossa. (Koskinen 2005)

Menetelmä on raskas, mutta antaa erittäin paljon tietoa tuotteen käytettävyydestä ver-

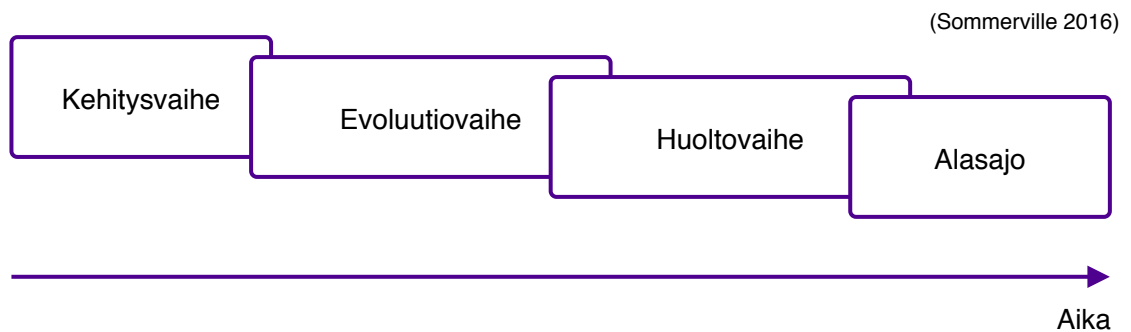
rattuna esimerkiksi pelkkään asiantuntija-arviointiin (Koskinen 2005). Käytettävyydestä voidaan suorittaa myös haastattelu, kyselylomakkeen täyttäminen tai molemmat. Käytettävyydestä ei tarjota pelkästään laadullista tietoa, vaan menetelmällä saadaan tietoa myös siitä, kuinka kauan tehtävien suorittamiseen kuluu aikaa, ja kuinka suuri prosentti tehtävistä onnistuu.

Muita menetelmiä ovat esimerkiksi fokusryhmät ja työpajat sekä havainnointi laboratorio-oloissa tai kentällä.

2.4 Ohjelmiston elinkaari

Ohjelmistot tuotetaan tyypillisesti projekteina, joissa on osapuolina vähintään toimittaja ja asiakas. Asiakas voi olla yrityksen sisäinen asiakas tai ulkoinen asiakas, ja niin ikään myös toimittaja voi olla yrityksen sisäinen tai ulkoinen. Ohjelmisto toteutetaan ratkaisemaan jokin asiakkaan ongelma, jonka asiakas kommunikoi toimittajalle. (Haikala ja Mikkonen 2011)

Sommerville (2016) kuvaa ohjelmistoja tyypillisesti pitkäikäisiksi. Esimerkiksi liiketoiminnassa käytettävät ohjelmat pysyvät käytössä usein yli 10 vuotta, ja armeijan järjestelmät voivat olla käytössä jopa 30 vuotta tai kauemmin. Jotta ohjelmistot pysyisivät relevanttina ja jatkaisivat asiakkaan tarpeisiin vastaamista, niihin on tehtävä muutoksia niiden elinkaaren aikana. Ohjelmiston elinkaaren vaiheet ovat 1) kehitysvaihe, 2) evoluutiovaihe, 3) huoltovaihe sekä 4) alasajovaihe (engl. retirement). Vaiheet on esitetty ajan suhteen kuvassa 2.3.



Kuva 2.3. Ohjelmiston elinkaari. Mukailtu lähteestä Sommerville 2016.

Ohjelmistokehitysvaiheessa ohjelmistoa rakennetaan. Tähän kuuluu mm. esitutkimus, määrittely ja suunnittelu. Evoluutiovaiheessa ohjelmisto on otettu käyttöön ja sen arkkitehtuuriin ja toiminnallisuuteen tehdään merkittäviäkin muutoksia. Sen sijaan huoltovaiheessa tehtävät muutokset ovat suhteellisen pieniä, mutta niitä on tehtävä, jotta järjestelmä pysyisi käyttökelpoisena. Evoluutio- ja huoltovaiheen raja on häilyvä, ja ne ovatkin keskenään osittain päällekkäiset. Huoltovaiheessa yritys todennäköisesti miettii jo, miten järjestelmä korvataan. Elinkaaren viimeisessä vaiheessa eli ohjelmiston alasajovaiheessa ohjelmisto voi edelleen pysyä käytössä, mutta muutoksia ei välttämättä tehdä enää ja rinnalla otetaan käyttöön ohjelmiston seuraaja. (Sommerville 2016)

3 TUTKIMUSMENETELMÄT JA AINEISTO

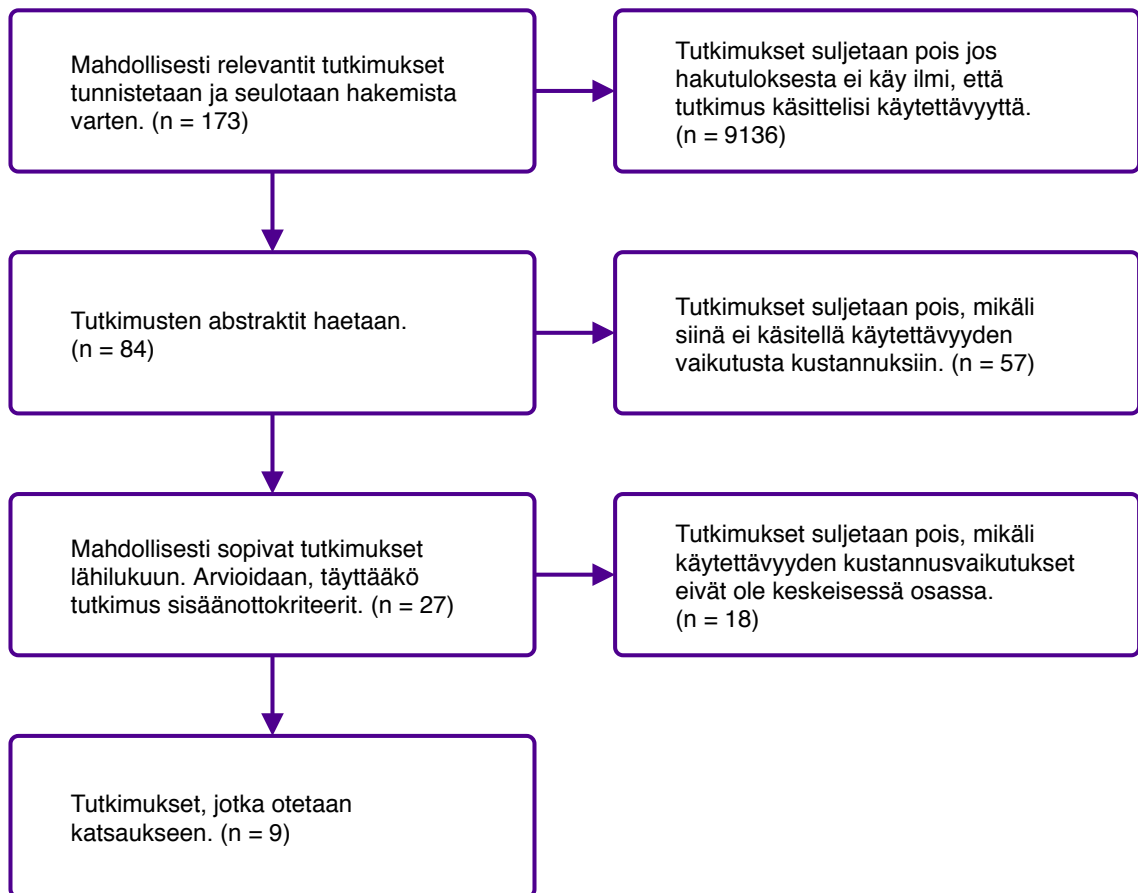
Tutkimusmenetelmänä on systemaattinen kirjallisuuskatsaus. Systemaattinen kirjallisuuskatsaus on järjestelmällisesti toteutettu, objektiivisuuteen pyrkivä ja kattava katsaus alan tutkimukseen, jota voidaan käyttää työkaluna, kun halutaan etsiä vastauksia tutkimuskysymykseen (Petticrew ja Roberts 2008). Perinteisen kirjallisuuskatsauksen sijaan menetelmäksi valittiin systemaattinen katsaus, sillä se auttaa vähentämään vahvistusharhaa (engl. confirmation bias) ja systemaattisella etenemistavalla saadaan aikaan laajempi läpileikkaus aiheesta tehtyyn tutkimukseen.

3.1 Tutkimusprosessi

Työ etenee ennalta määriteltyjen vaiheiden mukaan. Tässä työssä noudatetaan Petticrewn ja Robertsin (2008) kuvailemia systemaattisen katsauksen toteutusvaiheita. Systemaattisen kirjallisuuskatsauksen toteutusvaiheet ovat

1. Tutkimuskysymyksen määrittäminen
2. Katsaukseen valittavien tutkimusten rajaaminen
3. Kirjallisuushaku valittuihin tietokantoihin
4. Kirjallisuushakujen tuloksena saatujen tutkimusten vertailu sisäänottokriteereihin
5. Sisäänotettujen tutkimusten kriittinen arviointi
6. Synteesin tekeminen valituista tutkimuksista
7. Katsauksen tulosten jakaminen.

Katsaukseen valittavien tutkimusten rajaukseen vaikuttaa kandidaatintyöhön käytettävissä oleva aika ja tutkimusten saatavuus. Kaikkia sisäänottokriteerit täyttäviä tutkimuksia ei siis mitenkään ole mahdollista ottaa mukaan, vaan on priorisoitava. Tutkimuksen on oltava kokonaisuudessaan saatavilla Tampereen yliopiston lisenssillä ja hakuja tehdään vain elektronisista tietokannoista. Koehakujen perusteella alan keskeisistä tietokannoista käytettävyyssaiheisia tutkimuksia löytyy muutamia poikkeuksia lukuunottamatta vain englanniksi, joten sisäänotettavien tutkimusten kieleksi rajataan englanti. Sisäänottokriteerinä on, että tutkimuksessa käsitellään käytettävyyden kustannusvaikutuksia siten, että kustannusvaikutusten käsittely on keskeisessä osassa, eikä pelkästään lyhyenä mainintana. Aineiston sisäänotto- ja poissulkuprosessi on havainnollistettu kuvassa 3.1.



Kuva 3.1. Tutkimusaineiston sisäänto- ja poissulkuprosessi

Hakulausekkeena käytetään *usability AND ("cost-benefit" OR "cost-benefit analysis" OR "cost-justification" OR "cost-justifying" OR "return on investment")*. Koska kaikkien tietokantojen hakutoiminnot eivät tue esimerkiksi sulkujen käyttöä, toteutetaan varmuuden vuoksi viisi erilaista hakua:

- *usability AND "cost-benefit"*
- *usability AND "cost-benefit analysis"*
- *usability AND "cost-justification"*
- *usability AND "cost-justifying"*
- *usability AND "return on investment"*

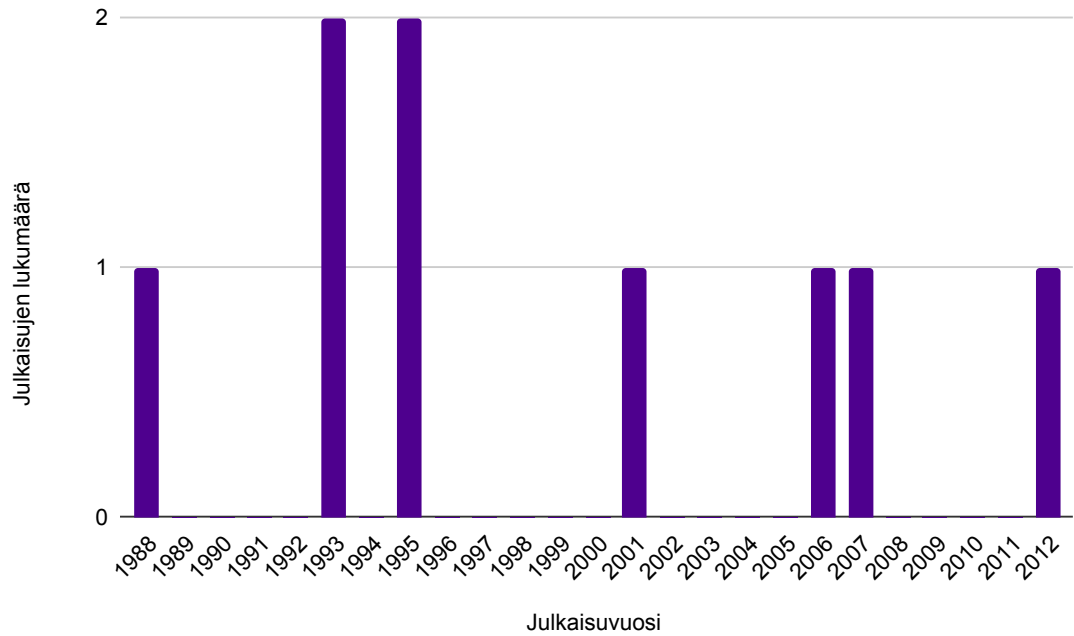
Tietokantojen valinnassa käytettiin apuna Tampereen yliopiston kirjaston (2020) opasta, jossa kerrotaan tietotekniikan alan keskeisimmät tietokannat. Oppaan mukaan keskeiset tietokannat ovat ACM Digital Library, Computer Science Database (ProQuest), IEEE, ScienceDirect (Elsevier), Scopus (Elsevier) sekä SpringerLink. Koehakuja tehdessä Scopusista ei juurikaan löytynyt relevantteja tuloksia, joten se jätetään pois. Aineiston hakuprosessi on esitetty tietokantakohtaisesti liitteessä A.

3.2 Tutkimusaineisto

Tutkimukseen valittiin yhdeksän tutkimusta aikaväliltä 1988-2012. Mukaan otetut tutkimukset on listattu taulukossa 3.1 ja tutkimusaineiston asettumista 22 vuoden ajalle on havainnollistettu kuvassa 3.2, jossa esitetään julkaisujen määrä julkaisuvuoden mukaan. Haut kohdistuivat julkaistuihin tutkimuksiin ja artikkeleihin, mutta aineistoon otetaan mukaan myös Biasin ja Mayhew'n toimittamassa Cost-Justifying Usability -kirjassa esitettyä materiaalia, sillä kirja on saavuttanut perusteoksen maineen (Wixon 1995; Sutton 2007). Yhtä kattavaa opasta käytettävyyden kustannusvaikutuksista ei ole toistaiseksi julkaistu. Myös Rajanen ja Iivari (2007) kuvailevat sitä parhaaksi lähteeksi erilaisille käytettävyyden kustannushyötymalleille, vaikka ensimmäinen painos julkaistiin jo vuonna 1994.

Taulukko 3.1. Katsaukseen valitut tutkimukset järjestettynä julkaisuvuoden mukaan vanhimmasta uusimpaan.

Tekijä(t)	Otsikko	Julkaisija	Julkaisu- vuosi
Mantei, Marilyn M.; Teorey, Toby J.	Cost/Benefit analysis for incorporating human factors in the software lifecycle	ACM	1988
Nielsen, Jakob	Is usability engineering really worth it?	IEEE	1993
Karat, Claire M.	Usability engineering in dollars and cents	IEEE	1993
Montaniz, Frank; Kissel, George V.	Reversing the Charges	ACM	1995
Dray, Susan	The Importance of Designing Usable Systems	ACM	1995
Donahue, George M.	Usability and the bottom line	IEEE	2001
Sorflaten, John	Making the Fuzzy Parts of ROI Clear	ACM	2006
Rajanen, Mikko; Iivari, Netta	Usability cost-benefit analysis: How usability became a curse word?	Springer	2007
Rajper, Samina; Shaikh, Abdul W.; Shaikh, Zubair A.; Amin, Imran	Usability cost benefit analysis using a mathematical equation	Springer	2012



Kuva 3.2. Tutkimusaineisto julkaisu vuoden mukaan.

Käytettävyyden kustannushyödyn laskemiseen esitetään malleja Mantein ja Teoreyn (1988), Donahuen (2001), Sorflatenin (2006) sekä Rajperin ym. (2012) tutkimuksissa. Lisäksi työhön otetaan mukaan Mayhew'n ja Tremainen (2005) malli. Muissa tutkimuksissa on esitelty käytettävyyden hyötyjä, joilla voidaan arvioida olevan vaikutus ohjelmiston elinkaaren kustannuksiin. Aineistoa käsitellään tarkemmin luvussa 4.

4 TULOKSET

Kirjallisuuskatsauksen tulokset voidaan jakaa kahteen pääkategoriaan: *käytettävyyden hyödyt kustannusnäkökulmasta* ja *käytettävyyden kustannushyötyjen laskeminen*. Käytettävyyden hyödyt kustannusnäkökulmasta tarkoittavat sellaisia käytettävyyden tuomia hyötyjä, joiden voidaan ajatella alentavan kustannuksia ohjelmiston elinkaaren aikana. Käytettävyyden kustannushyötyjen laskeminen tarkoittaa konkreettisia keinoja käytettävyyden kustannusten laskemiseen: mitä säästöjä ehdotetut toimenpiteet voivat tuoda ja mitä toimenpiteet maksavat.

4.1 Käytettävyyden hyödyt kustannusnäkökulmasta

Käytettävyyden kustannushyötyjä voidaan tarkastella *sisäisestä näkökulmasta* eli yrityksen käytössä olevien ohjelmistojen näkökulmasta sekä *ulkoisesta näkökulmasta* eli ohjelmistoja tuottavan yrityksen näkökulmasta. Erottelu on olennaista siksi, että hyödyillä on vaihtelevan suuruinen merkitys eri osapuolille. Esimerkiksi yritys, jossa tehdään tiettyä ohjelmistoa käyttäen ei suoraan hyödy kustannusmielessä siitä, jos ohjelmiston toimittajayrityksen myynti kasvaa käytettävyyden ansiosta. Sen sijaan yritys hyöttyy suoraan siitä, että heidän käytössään olevilla ohjelmistoilla voidaan tehdä töitä tehokkaasti ja mahdollisimman virheettä niin, että ohjelmisto ei haittaa ja hidasta työntekoa.

Toimittajayritys taas ei välttämättä suoraan hyödy esimerkiksi siitä, että paremman käytettävyyden myötä ohjelmistoa käytettäessä tehdyt virheet vähenevät. Silläkin on merkitystä, tekeekö toimittaja ohjelmistoja kuluttajille, yrityksille vai molemmille. Kuluttajilla on joustavampi mahdollisuus vaihtaa käyttämiään ohjelmistoja jos vaihtoehto on tarjolla, ja tällöin käytettävyys tuo merkittävää kilpailuetua. Yritykset taas ostavat tiettyyn tarpeeseen räätälöityjä ja näin ollen kalliita ohjelmistoja, joiden vaihtaminen toiseen ei ole yksinkertaista. Käytettävyyden kustannushyödyt ovat tapauskohtaisia, usein epäsuoria, eivätkä aina yksinkertaisia jäljittää.

On kuitenkin viisasta panostaa käytettävyyteen jo ohjelmiston suunnitteluvaiheessa. Pressmanin (1992) mukaan ongelman korjaaminen kehitysvaiheessa maksaa kymmenen kertaa sen verran mitä se maksaisi suunnitteluvaiheessa. Kun järjestelmä on julkaistu, ongelman korjaaminen maksaa satakertaisesti sen verran mitä se maksaisi suunnitteluvaiheessa. Samassa tutkimuksessa havaittiin myös, että 80% ohjelmiston elinkaaren kuluista ilmenee ylläpitovaiheessa. Niistä moni liittyy käyttäjän vaatimukseen ja muihin ongelmiin, jotka voidaan välttää käytettävyyden avulla. (katso Donahue 2001) On kuitenkin

huomioitava, että Pressmanin tutkimus on vuodelta 1992, ja 28 vuodessa mm. internet on huomattavasti muuttanut sitä, miten sovelluksia myydään ja käytetään. On siis mahdollista, että tänä päivänä ongelman korjaaminen ohjelmiston julkaisun jälkeen ei välttämättä ole satakertaisesti kalliimpaa kuin suunnitteluvaiheessa. Tutkimus antaa kuitenkin näkökulmaa siihen, että suunnitteluvaiheella on merkittävä rooli siinä, kuinka kalliiksi ohjelmisto voi tulla elinkaarensa aikana.

4.1.1 Sisäinen näkökulma

Käyttäjän aikaansaavuus kasvaa paremman käytettävyyden myötä (Donahue 2001; Nielsen 1993a; Karat 1993; Dray 1995). Käyttäjän aikaansaavuuden kasvamisen kustannushyöty perustuu siihen, että järjestelmää käyttäessään käyttäjä suoriutuu tehtävistä ja etenkin toistuvista tehtävistä nopeammin. Syntyy ajansäästöä, jonka ansiosta syntyy kustannussäästöä.

Käyttäjän tekemät virheet vähenevät (Mantei ja Teorey 1988; Mayhew ja Tremaine 2005; Dray 1995). Käyttäjän tekemät virheet ovat aika- ja vaivasyöppöjä, sillä virheestä palautumiseen kuluu aikaa, joka on pois tuottavasta tekemisestä. Virhe voi myös lisätä käyttäjätuen tarvetta, jos käyttäjä ei pääse eteenpäin omin avuin. Virheet voivat pahimmillaan tulla erittäin kalliiksi. Esimerkkinä vaikkapa tehtaiden tuotannossa käytetyt järjestelmät: käytettävyysongelmilla voi pahimmillaan olla hintavat seuraukset, jos niiden takia lopputuotteeseen tulee virheitä. On järjestelmiä, joissa huonon käytettävyyden hintana voi olla ihmishenkiä. Silloin käytettävyysongelmien aiheuttamille virheille ei yksinkertaisesti ole sijaa.

Käyttäjätuetyytyväisyys kasvaa (Nielsen 1993a; Karat 1993). Käyttäjätuetyytyväisyyden kasvu on yksi ilmeisimmistä käytettävyyden hyödyistä. Sen liittäminen kustannushyötyihin ei kuitenkaan ole välttämättä suoraviivaista (Nielsen 1993a). Nielsen kuitenkin päättelee käyttäjätuetyytyväisyyden kasvun vähentävän poissaoloja ja henkilöstön vaihtuvuutta, kun on kyse yrityksen sisäisessä käytössä olevista ohjelmistoista. Voidaan arvella, että käyttäjätuetyytyväisyydellä on yhteys työttyytytytytytyteen ja työhyvinvointiin.

Koulutustarve vähenee (Donahue 2001; Nielsen 1993a; Dray 1995). Käytettävyyttä edistää käytön oppimista ja käytön muistamista, jolloin järjestelmän käyttökoulutusta tarvitaan vähemmän, jos lainkaan. Järjestelmien käytön oppiminen on myös nopeampaa. Käytön muistaminen on tärkeää etenkin yrityksissä, joissa on käytössä monia järjestelmiä, joita yksittäinen työntekijä tarvitsee vain silloin tällöin. Jos käyttökertojen välissä ehtii unohdtaa, kuinka järjestelmää käytetään, kuluu turhaan aikaa asioiden uudelleenopetteluun ja virhealttius kasvaa.

Käyttäjätuen tarve vähenee (Dray 1995). Käyttäjätuen kustannukset voivat olla suuret: Montaniz ja Kissel (1995) selvittivät muutaman yleisimmän, huonosti suunnitellun virheilmoituksen vaikutusta käyttäjätuen kustannuksiin toimittajayrityksessä, ja yleisimpien virhetilanteiden aiheuttamat käyttäjätukikustannukset olivat puolessa vuodessa 411 918 dollaria. Perinteisen teknisen tuen lisäksi suuri piilokulujen aiheuttaja on ns. vertaiskäyt-

täjätuki, eli kun käyttäjä pyytää apua työkaverilta, jonka tietää osaavan auttaa. Tällöin kokeneemman työkaverin aikaansaavuus kärsii. (Rajanan ja livari 2007) Tämän kaltaisten piilossa pysyvien vertaiskäyttäjätukikulujen on arvioitu maksavan 6000-15 000 dollaria vuodessa yhtä tietokonetta kohti (Bulkeley 1992, Rajanan ja livari 2007 mukaan). Arviot ovat tosin vuodelta 1992, joten niitä ei välttämättä voida sellaisenaan siirtää nykyhetkeen.

Muita mainittuja kustannuksiin yhteydessä olevia hyötyjä ovat henkilöstön poissaolojen ja vaihtuvuuden väheneminen (Dray 1995), korkeampi moraali (Dray 1995), pienemmät dokumentaatiokustannukset (Donahue 2001), oikeudenkäyntikuluilta välttyminen (Donahue 2001), järjestelmien sabotoinnilta välttyminen (Mantei ja Teorey 1988) ja luovan ongelmanratkaisun helpottuminen (Mantei ja Teorey 1988). Hyvä käytettävyys vähentää riskiä järjestelmän hylkäämiseen (Mantei ja Teorey 1988).

4.1.2 Ulkoinen näkökulma

Myynti kasvaa (Nielsen 1993a). Käytettävyyden roolia myynnin kasvamisessa on kuitenkin vaikea laskea, sillä käytettävyys ei ole ainoa tekijä tuotteen valinnassa. Nielsenin mukaan tuote, jonka käytettävyys on huono, voi silti olla suosittu, jos sen muut ominaisuudet menevät asiakkaan tärkeysjärjestyksessä käytettävyyden ohi. Toisaalta on myös vaikeaa laskea kuinka paljon myyntiä on menetetty sen vuoksi, että asiakas on valinnut jonkin toisen tuotteen paremman käytettävyyden takia. Käytettävyys parantaa palvelun laatua ja kasvattaa asiakastyytyväisyyttä (Dray 1995), ja sillä saattaa olla yhteys myynnin kasvamiseen. Rajanan ja livari (2007) selvittivät tapaustutkimuksessaan, että käytettävyyttä voidaan hyödyntää myynnissä ja markkinoinnissa asiakkaan vakuuttamiseksi.

Yritys saa kilpailuetua (Donahue 2001). Donahue kuvailee käyttäjien arvostavan sellaisia ohjelmistoja ja web-sivuja, joiden parissa ei mene aikaa hukkaan, ja jotka eivät koettele käyttäjien kärsivällisyyttä. Käytettävyyden huomioiminen viestittää käyttäjille, että yritys arvostaa käyttäjiensä aikaa, eikä pidä heitä itsestäänselvyytenä.

Viime hetken muutokset ohjelmistokehityksessä vähenevät (Mantei ja Teorey 1988; Mayhew ja Tremaine 2005). Käytettävyyden menetelmät auttavat selvittämään käyttäjätarpeet ja käyttökontekstin perusteellisesti heti suunnitteluvaiheessa, jolloin ohjelmiston suunnittelua päästään toteuttamaan niiden pohjalta sen sijaan, että toimittaisiin arvioiden varassa. Mitä aikaisemmassa vaiheessa muutokset tehdään, sitä halvemmaksi se tulee; myöhemmässä vaiheessa suurien muutosten teko saattaa olla liki mahdotonta tai niin kallista, ettei siihen haluta ryhtyä. Kun vaatimusten kokoaminen ja ensimmäiset prototyypit iteraatioineen on toteutettu tiiviissä yhteistyössä asiakkaan ja käytettävyydestaajien (eli loppukäyttäjryhmien edustajien) kanssa, saadaan ennakoitua sellaisia käyttäjätarpeita, jotka eivät olisi muuten tulleet mieleen niin asiakkaalle kuin toimittajallekaan.

Ohjelmistokehityksessä voidaan välttyä yllättäviltä kuluilta (Karat 1993). Yllättävien kulujen välttäminen liittyy viime hetken muutosten vähenemiseen. Kun käyttäjärühmät ja käyttökonteksti on selvitetty perusteellisesti ja tuotteen prototyyppiä on varhaisista versioista saakka iteroitu käytettävyydesteistä saatujen tietojen pohjalta, on todennäköistä,

että tuotteesta on tullut sellainen, jonka asiakas haluaa, ja joka täyttää tehtävänsä hyvin. Perusteellisella käytettävyydellä voidaan siis välttyä ikäviltä, kehitysaikaa viivästyttäviltä yllätyksiltä. Karat (1993) raportoi, että eräässä projektissa onnistuttiin käytettävyyden avulla välttämään kuluja, joita ei oltu osattu odottaa. Uusi järjestelmä myös valmistui ajallaan ilman suuria ongelmia, ja käyttäjätyytyväisyys oli korkea. Rajanen ja livari (2007) havaitsivat tapaustutkimuksessaan, että ohjelmistoyritys voi käyttää käytettävyyttä myös keinona pitää asiakkaat poissa ohjelmistokehitysvaiheesta, jolloin vältetään kasvavilta ohjelmistokehityskuluilta. Tässä kontekstissa asiakkaat ovat yrityksiä. Havainto ohjelmistokehityskulujen pienentämisestä ei kuitenkaan perustu siihen, että vaatimukset saadaan perusteellisesti selville ohjelmistokehityksen alussa, vaan siihen, että käytettävyyttä käytetään asiakkaan vakuuttamiseen ja asiakas hyväksyy määrittelydokumentin. Määrittelydokumentin hyväksymisen jälkeiset muutokset laskutetaan erikseen, jolloin ohjelmistokehityskulut eivät kasva.

Kirjallisuudessa esiintyviin väitteisiin käytettävyyden kustannushyödyistä kohdistuu myös kritiikkiä. Rajanen ja livari (2007) toteuttivat tutkimuksen, jossa verrataan kirjallisuudessa esitettyjä käytettävyyden kustannushyötyjä empiirisestä tapaustutkimuksesta saatuihin tuloksiin. Tutkimus toteutettiin keskisuudessa ohjelmistoyrityksessä, joka tuottaa laajan skaalan informaatiojärjestelmiä yrityksille, sekä kansainvälisille markkinoille suunnattuja ohjelmistointensiivisiä (engl. software-intensive) tuotteita. Ennen tapaustutkimusta yrityksessä ei juurikaan oltu tehty käytettävyytyötä, ja tutkimuksen aikana kokeiltiin mm. asiakasvierailuja, käytettävyyksvaatimusten määrittelytyöpajoja, paperiprototyyppejä, käytettävyytestausta ja käyttöliittymän tyylioppaan kehittämistä.

Käytettävyytyötä ei kuitenkaan haluttu jatkaa yrityksessä sen tuomien lisäkustannusten vuoksi, eikä tieto mahdollisista tulevaisuudessa ilmenevistä hyödyistä riittänyt vakuuttamaan yritystä käytettävyyksaktiiviteettien jatkamisesta. Käytettävyytyö näyttäytyi yrityksen johdolle liian kalliina ja aikaavievänä. Rajanen ja livari huomauttavat, että käytettävyyden aikaansaamille hyödyille ei annettu tarpeeksi aikaa tulla esiin, ja että käyttöliittymäkulut realisoituvat jossain muodossa joka tapauksessa, sillä yrityksen kaikissa tuotteissa kuitenkin on käyttöliittymä. Donahue (2001) on samoilla linjoilla käytettävyytyön kuluista: jos tuotteessa on käyttöliittymä, sen kehittämiseen menee aina aikaa ja rahaa, vaikka ei tietoisesti käytettäisikään käytettävyytyön menetelmiä.

4.2 Käytettävyyden kustannushyödyn laskeminen

Tässä luvussa esitellään aineistossa esiintyvät mallit käytettävyyden kustannushyödyn laskemiseen. Esiteltävät mallit ovat Mantein ja Teoreyn malli (1988), Donahuen malli (2001), Mayhew'n ja Tremainen malli (2005), Sorflatenin malli (2006) ja Rajperin ym. malli (2012). Lisäksi esitellään, miten Karat (1993) sekä Montaniz ja Kissel (1995) ovat tutkimuksissaan laskeneet käytettävyyden kustannushyötyjä.

4.2.1 Mantein ja Teoreyn malli

Mantei ja Teorey (1988) käsittelevät artikkelissaan käyttöliittymien parannusmetodeja, ihmisten tekijöiden tuomaa elementtiä ohjelmistoprojektissa sekä kustannus-hyötynäkökulmia. Mantei ja Teorey ovat pyrkineet rinnastamaan kustannus-hyötylaskelmansa COCOMO-malliin (Constructive Cost Model), joka on ohjelmistotuotannon kustannusmalli. He ehdottavat, että ihmisten tekijöiden huomioiminen ohjelmistoprojektissa lisää siihen seuraavat kustannukset. Mantei ja Teorey antavat esimerkkilaskelman kaikista mainittujen käytettävyyssaktiviteettien kustannuksista, ja tässä työssä esitetään niistä esimerkkinä käyttäjätutkimuksen kustannuslaskelma taulukossa 4.1.

1. Fokusryhmien vetämisen kustannukset.
2. Tuotemallien (engl. mock-up) luomisen kustannukset.
3. Ensimmäisen prototyypin luomisen kustannukset.
4. Prototyypin muokkaamiskustannukset.
5. Prototyypiohjelmiston kustannukset.
6. Käyttäjätutkimusten kustannukset.
7. Käytettävyyss tutkimusympäristön (laboratorion) perustamisen kustannukset.
8. Käyttäjäkyselyn toteuttamisen kustannukset.

Taulukko 4.1. Käyttäjätutkimuksen kustannusten laskeminen Mantein ja Teoreyn esimerkissä. Mukailtu lähteestä Mantei ja Teorey 1988.

Kustannus	Määrä
Kyselylomakkeen kehittäminen	1600 \$
Kyselylomakkeen pilottitesti	1600 \$
Kyselyn jakaminen ja kerääminen	300 \$
Tietojen koodaaminen ja syöttäminen	300 \$
Tulosten analysointi	1600 \$
Kyselylomakkeen täyttämiseen käytetty aika	1600 \$
Tietokoneaika	100 \$
Tarvikkeet ja päällekkäiset kulut	100 \$
Yhteensä	7200 \$

Mantei ja Teorey arvioivat ehdottamiensa käytettävyysspanostusten tuovan seuraavat konkreettiset hyödyt: käyttäjän oppimiseen kuluvan ajan väheneminen, käyttäjän tekemien virheiden väheneminen ja järjestelmän ylläpidon kustannusten pieneneminen. Näille hyödyille on esitelty laskukaavat ja -esimerkit. Säästöt koulutuskustannuksista voidaan las-

kea kaavalla

$$\text{säästöt per vuosi} = (\text{henkilöstön vaihtuvuus}) \cdot (\text{säästetty koulutusaika}) \cdot (\text{palkka}), \quad (4.1)$$

jossa *henkilöstön vaihtuvuus* on henkilöstön vaihtuvuusluku vuodessa, *säästetty koulutusaika* on työntekijän koulutusaika tunteina ja *palkka* on työntekijän tuntipalkka. Toinen konkreettinen hyöty eli käyttäjän tekemien virheiden väheneminen voidaan laskea selvittämällä vuodessa sattuvien virheiden määrä ja laskea siitä virheiden kustannukset:

$$\text{virheet per vuosi} = (\text{työntekijöiden määrä}) \cdot (P[\text{virhe}]) \cdot (\text{skenaariot per tunti}) \cdot (\text{tunnit per vuosi}), \quad (4.2)$$

jossa *työntekijöiden määrä* on niiden työntekijöiden määrä, jotka suorittavat skenaariota, $P[\text{virhe}]$ on todennäköisyys sille että käyttäjä tekee virheen, *skenaariot per tunti* on suoritettujen skenaarioiden määrä tunnissa ja *tunnit per vuosi* on kuinka monta tuntia vuodessa skenaariota toistetaan. Virheiden kustannukset vuotta kohden voidaan laskea kaavalla

$$\text{kustannukset per vuosi} = (\text{virheet per vuosi}) \cdot (\text{virheestä palautuminen}) \cdot (\text{tuntipalkka}), \quad (4.3)$$

jossa *virheet per vuosi* saadaan kaavalla 4.2, *virheestä palautuminen* on virheistä palautumiseen käytetty aika tunteina ja *tuntipalkka* on työntekijän tuntipalkka. Kolmas hyöty eli järjestelmän ylläpidon kustannusten pieneminen perustuu siihen, että muutokset on edullisinta tehdä aikaisessa vaiheessa. Mantein ja Teoreyn mukaan prototyyppivaiheessa tehtyjen suunnittelumuutosten voidaan arvioida maksavan neljäsosan siitä, mitä muutokset maksaisivat käyttöönotetussa järjestelmässä. Toisin sanoen siis myöhäisessä vaiheessa tehdyt muutokset maksavat nelinkertaisesti sen verran mitä ne maksaisivat prototyyppivaiheessa. Varhaisessa vaiheessa tehtyjen muutosten aikaansaamat säästöt voidaan laskea kaavalla

$$\text{varhainen kustannus} = (\text{tunnit per muutos}) \cdot (\text{muutosten määrä}) \cdot (\text{tuntipalkka}), \quad (4.4)$$

jossa *tunnit per muutos* on muutosten tekemiseen kuluva aika tunteina, *muutosten määrä* on tarvittavien muutosten lukumäärä ja *tuntipalkka* on sen työntekijän palkka, joka toteuttaa muutokset. Kokonaisuudessaan myöhäisten suunnitelmamuutos-

ten välttämisestä tulevat säästöt voidaan laskea kaavalla

$$\begin{aligned} & \text{säästöt suunnitelmamuutoksista} \\ & = (\text{myöhäinen kustannus}) - (\text{varhainen kustannus}) \\ & = 4 \cdot (\text{varhainen kustannus}) - (\text{varhainen kustannus}), \end{aligned} \tag{4.5}$$

jossa *myöhäinen kustannus* on nelinkertaisesti varhaisen kustannuksen suuruinen ja jossa *varhainen kustannus* saadaan kaavalla 4.4.

On huomioitava, että Mantein ja Teoreyn tutkimus on julkaistu vuonna 1988 ja monet artikkelissa esitetyt luvut eivät sellaisenaan ole relevantteja enää vuonna 2020. Esimerkiksi prototyyppisovellusten hintojen kerrotaan sijoittuvan 2500 dollarin ja 15 000 dollariin välille, ja että tyyppillisesti kaikki tarpeelliset ominaisuudet sisältävä ohjelmisto maksaa n. 10 000 dollaria. Vertailun vuoksi vuonna 2020 prototyyppiohjelmistoja ovat esimerkiksi Adobe XD, InVision ja Figma. Hinnoittelu on porrastettua ja kaikista edellä mainituista ohjelmista on olemassa ilmainen starter-versio yksityishenkilöille. Yrityskäyttöön hinnat ovat kalleimmillaan seuraavat: Adobe XD maksaa 79,99 dollaria per käyttäjä per kuukausi, InVision maksaa 99 dollaria per (max.) viisi käyttäjää per kuukausi ja Figma maksaa 45 dollaria per organisaatio per kuukausi (Adobe 2020; InVision 2020; Figma 2020). On siis huomattava, että sovellusten hinnoittelu on muuttunut suurista kertamaksuista jatkuviksi kuukausittaisiksi pienemmiksi maksuiksi. Sopiva paketti ostetaan tarpeen mukaan ja hinta määräytyy paketin laajuuden mukaan.

4.2.2 Mayhew'n ja Tremainen malli

Mayhew ja Tremaine (2005) esittelevät perusrungon käytettävyyden kustannusten perusteluun Cost-Justifying Usability -kirjassa. Vuonna 2005 ilmestynyt versio on kirjan toinen painos, ja ensimmäinen painos ilmestyi vuonna 1994, eli kustannus-hyötymalli on alunperin julkaistu jo silloin. He kuvailevat käytettävyyden kustannusten laskemisen varsin helpoksi, kun käytettävät menetelmät ovat tiedossa, mutta huomauttavat hyötyjen laskemisen olevan hankalampaa. Lopputulos on yksinkertainen ainakin teoriassa: verrataan käytettävyyden kustannuksia sen aikaansaamiin hyötyihin, jotta voidaan pohtia, missä määrin hyödyt ylittävät kustannukset. Mayhew'n ja Tremainen mallin perusvaiheet ovat:

1. Muodosta käytettävyydesuunnitelma.
2. Vahvista analyysiin tarvittavat parametrit.
3. Laske jokaisen käytettävyydesuunnitelmassa olevan tehtävän kustannukset.
4. Valitse sopivat hyötykategoriat.
5. Arvioi hyödyt.
6. Vertaa kustannuksia hyötyihin.

Mallin ensimmäinen vaihe on käytettävyydesuunnitelman muodostaminen, jotta pystytään laskemaan kustannukset jokaiselle vaiheelle. Käytettävyydesuunnitelmaan valitta-

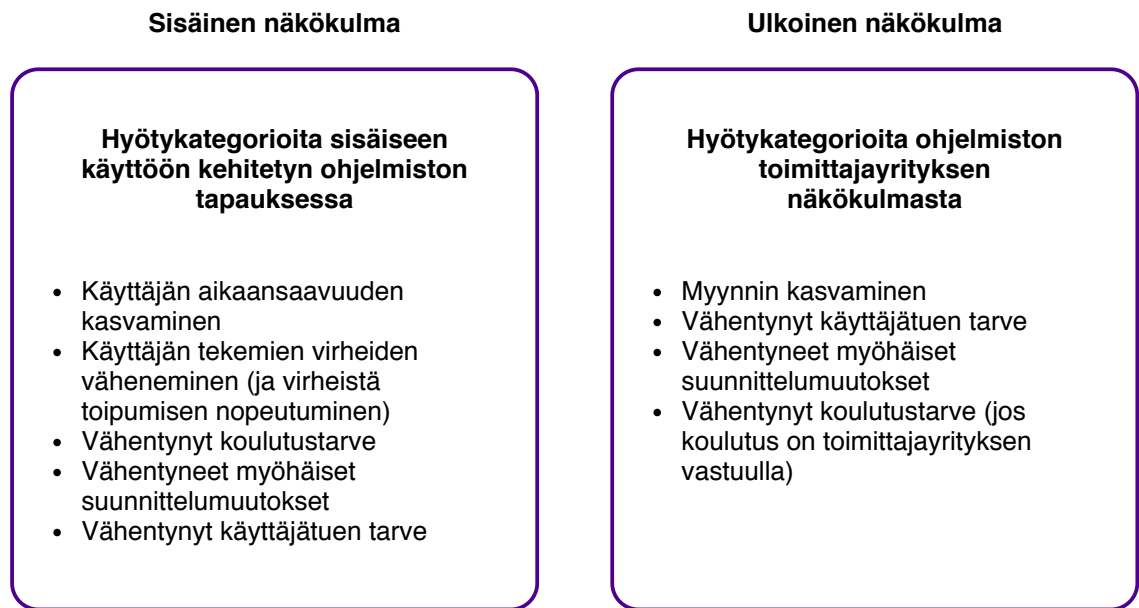
vat menetelmät vaihtelevat aina budjetin, aikataulun ja projektin monimutkaisuuden mukaan. Toinen vaihe on analyysiin tarvittavien parametrien selvittäminen. Tarvittavat parametrit riippuvat siitä, ollaanko kiinnostuneita sisäisestä vai ulkoisesta näkökulmasta. Mayhew'n ja Tremainen mukaan sisäisestä näkökulmasta muutamia tärkeimpiä parametreja ovat esimerkiksi käyttäjien määrä, tapahtumien määrä ja käyttäjän ladattu tuntipalkka. Ladattu tuntipalkka tarkoittaa sitä, että työntekijälle maksettavan palkan lisäksi lukuun sisällytetään kaikki kustannukset, mitä työntekijästä on yritykselle. Kolmas vaihe on käytettävyyssuunnitelmassa olevien tehtävien kustannusten laskeminen. Mayhew'n ja Tremainen esimerkki käytettävyydestauksen kustannusten laskemisesta on taulukossa 4.2.

Taulukko 4.2. Käytettävyydestauksen kustannusten laskeminen Mayhew'n ja Tremainen (2005) esimerkissä. Mukailtu lähteestä Mayhew ja Tremaine 2005.

Askel	Käytettävyyssinsinöörin työtunnit	Kehittäjän työtunnit	Päällikön työtunnit	Käyttäjän työtunnit	
Suunnittele testimateriaalit	32				
Suunnittele ja kokoa testiympäristö	4	32			
Suorita pilottitesti ja päivitä materiaalit	10	8		6	
Suorita testi ja kerää data (2 käytettävyyssinsinööriä)	32			16	
Järjestele data	16				
Analysoi data ja suunnittele parannusehdotukset	24				
Dokumentoi ja esittele johtopäätökset	24				
Tunnit yhteensä	142	40	0	22	
Tuntihinta	175 \$	175 \$	200 \$	25 \$	
Hinta yhteensä	24 850 \$	7000 \$	0 \$	550 \$	32 400 \$

Neljäs vaihe on hyötykategorioiden valitseminen. Mallia sovelletaan eri käyttötarkoituksiin valitsemalla kuhunkin tapaukseen sopivat hyötykategoriat. Esimerkiksi sisäiseen käyttöön kehitetyssä ohjelmistossa (sisäinen näkökulma) sopivia hyötykategorioita ovat käyt-

täjän aikaansaavuuden kasvaminen, käyttäjän tekemien virheiden väheneminen (ja virheistä toipumisen nopeutuminen), vähentynyt koulutustarve, vähentyneet myöhäiset suunnittelumuutokset ja vähentynyt käyttäjätuen tarve. Ohjelmiston toimittajajärityksen näkökulmasta (ulkoinen näkökulma) sopivia hyötykategorioita ovat myynnin kasvaminen, vähentynyt käyttäjätuen tarve, vähentynyt koulutustarve (jos koulutus on toimittajajärityksen vastuulla) ja vähentyneet myöhäiset suunnittelumuutokset. Hyötykategoriat on koottu kuvaan 4.1. Käytännössä hyötykategorioita voi tilanteesta riippuen olla muitakin.



(Mayhew ja Tremaine 2005)

Kuva 4.1. Käytettävyyden kustannus-hyötyanalyysin hyötykategoriat sisäisestä ja ulkoisesta näkökulmasta. (Mayhew ja Tremaine 2005)

Kun hyötykategoriat on valittu, viides vaihe on hyötyjen arvioiminen jokaisessa kategoriassa. Jos sisäiseen käyttöön suunnitellun sovelluksen hyötykategorioiksi on valittu käyttäjän aikaansaavuuden kasvaminen, käyttäjän tekemien virheiden väheneminen, vähentynyt koulutustarve ja vähentyneet myöhäiset muutokset ohjelmistokehitysvaiheessa, hyödyt voisivat olla esimerkiksi sellaiset kuten esitetty taulukossa 4.3. Esimerkki käyttäjän aikaansaavuuden kasvamisen tuoman hyödyn laskemisesta on taulukossa 4.4.

Taulukko 4.3. Esimerkkiarvioita hyödyistä sisäiseen käyttöön suunnitellussa (ja yrityksen sisällä kehitetyssä) sovelluksessa. Mukailtu lähteestä Mayhew ja Tremaine 2005.

Hyötykategoria	Arvioitu hyöty
Käyttäjän aikaansaavuuden kasvaminen	Vähentynyt aika per tapahtuma (5 s = 0.001389 h)
Käyttäjän tekemien virheiden väheneminen	1 virhe per päivä
Vähentynyt koulutustarve	10 tuntia vähemmän
Vähentyneet myöhäiset muutokset ohjelmistokehityksessä	20 muutosta tehty aikaisessa vaiheessa

Taulukko 4.4. Hyödyn laskeminen Mayhew'n ja Tremainein mallissa, esimerkkinä käyttäjän aikaansaavuuden kasvaminen. Parametrit kerrotaan keskenään, jolloin saadaan lopputulokseksi hyödyn suuruus vuodessa. Mukailtu lähteestä Mayhew ja Tremaine 2005.

Analyysiparametri	Arvo
Käyttäjien määrä	250
Päivien määrä	230
Tapahtumien määrä	100
Säästettyjen tuntien määrä per tapahtuma	0.001389
Tuntipalkka	25 \$
Yhteensä	199 652,78 \$

Kuudes ja viimeinen vaihe on kustannusten vertaaminen hyötyihin. Se tarkoittaa sitä, että vähennetään kokonaiskustannukset kokonaisyödyistä, jolloin saadaan tulokseksi nettohyöty. Seuraavat askeleet riippuvat nettohyödyn suuruudesta. Jos nettohyöty on suuri, käytettävyyssuunnitelmaa kannattaa alkaa toteuttaa. Jos nettohyöty jää pieneksi tai jos kustannukset ovat suuremmat kuin hyödyt, Mayhew ja Tremaine ohjeistavat tarkastelemaan käytettävyyssuunnitelmaa uudelleen ja pohtimaan, voisiko joitakin vaiheita tehdä oikoen tai yksinkertaisemmin.

4.2.3 Donahuen malli

Donahue (2001) esittelee kustannus-hyötyanalyysimallin, joka koostuu neljästä vaiheesta. Hän kuvailee perusidean olevan se, että arvioidaan kustannukset ja hyödyt käytettävyyssuunnitelma-aktiiviteeteista (esim. prototyyppien luominen, käytettävyyssuunnitelma, heuristinen arviointi) ja verrataan niitä kustannuksiin, jotka ilmenisivät, jos käytettävyyssuunnitelma-aktiiviteetit jätettäisiin tekemättä. Donahuen malli perustuu Mayhew'n ja Tremainein malliin. Vaiheet

Donahuen mallissa ovat:

1. Käytettävyyssmenetelmän valinta.
2. Sopivan mittayksikön valitseminen.
3. Hyödyn laajuuden arvioiminen.
4. Odotettavissa olevan hyödyn muuttaminen rahasummaksi.

Donahue esittelee mallin kuvitteellisen esimerkin avulla, jossa järjestelmänä on yrityksen sisällä kehitetty henkilöstöhallintojärjestelmä. Esimerkissä käytettävyyssasiantuntija haastattelee henkilöstöosaston johtajaa ja järjestelmän käyttäjiä, ja saa heiltä tiedon, että järjestelmä on liian monimutkainen. Erityisesti hakemuksen käsittely, jota toistetaan usein, vie aikaa ja on hankalaa. Taustatietojen selvityksestä käy siis ilmi, että järjestelmän käyttäjien aikaansaavuus voisi kasvaa, jos hakemuksen käsittelyn käytettävyyteen panostettaisiin. Kehittäjiltä varmistetaan olisiko uudenlainen, käytettävämpi toteutustapa teknisesti mahdollinen, ja kuinka paljon siihen tulisi varata työtunteja. Kun tarpeellinen uudistus on selvillä, lasketaan kustannukset. Kustannusten ja hyötyjen laskemista varten on selvitettävä muutama parametri, jotka on esitetty taulukossa 4.5.

Taulukko 4.5. *Kustannusten ja hyötyjen laskemiseen tarvittavat parametrit Donahuen (2001) esimerkissä.*

Analyysiparametri	Arvo
Hakemuksen käsittelyyn kuluva aika nykyisessä järjestelmässä keskimäärin	4 h
Hakemuksen käsittelyyn kuluva aika muutoksen jälkeen	3 h
Dataa käsittelevän työntekijän (ladattu) tuntipalkka	25 \$
Hakemuksen käsittelyn kustannukset per hakemus	$4 \cdot 25 \$ = 100 \$$
Hakemusten määrä vuodessa	1000
Hakemuksen käsittelyn kustannukset per vuosi	$1000 \cdot 100 \$ = 100\,000 \$$
Muutoksen toteuttamiseen kuluva aika	40 h
Muutoksen toteuttavan työntekijän (ladattu) tuntipalkka	60 \$

Donahue ilmoittaa esimerkissä muutoksen toteuttamisen kustannuksiksi 2400 dollaria, joka on muutoksen toteuttamiseen kuluvan ajan ja muutoksen toteuttavan työntekijän tuntipalkan tulo. Mainittavaa on, että tässä esimerkissä muutoksen toteuttamisen kustannuksissa ei huomioitu käytettävyyssyön kustannuksia, vaikka Donahuen esimerkissä tehtiin haastatteluja ja testausta paperiprototyypin avulla.

Kustannusten jälkeen selvitetään hyödyt. Esimerkissä havaitaan, että suunniteltu muutos hakemusten käsittelyyn voisi puolittaa käsittelyajan. Halutaan kuitenkin pelata varman päälle, ja oletetaan, että muutos pienentäisi käsittelyaikaa 25 prosenttia. Käsittelyaika olisi muutoksen jälkeen kolme tuntia, ja näin ollen yksittäisen hakemuksen käsittelyn kustannukset olisivat 75 dollaria.

Kun kaikki tarvittavat luvut ovat selvillä, voidaan arvioida, paljonko säästöä muutos toisi vuodessa. Kerrotaan hakemuksen käsittelyn kustannukset vuosittaisella hakemusten määrällä, jolloin saadaan tulokseksi 75 000 dollaria. Muutoksen avulla voitaisiin siis säästää 25 000 dollaria. Tästä vähennetään muutoksen toteuttamisen kulut, jolloin saadaan arvio ensimmäisen vuoden aikana saavutetusta säästöstä: 22 600 dollaria. Järjestelmän arvioidaan kuitenkin olevan käytössä ainakin kolme vuotta, jolloin voidaan laskea säästö koko ajalta. Järjestelmän elinkaaren aikana voitaisiin säästää 72 600 dollaria. Kustannus-hyötysuhteeksi tulee tällöin 1:30,25.

4.2.4 Sorflatenin malli

Sorflaten (2006) lähestyy käytettävyyden sijoitetun pääoman tuottoa käyttäjän aikaansaavuuden kasvamisen näkökulmasta, jolloin käytettävyyden avulla voidaan säästää työvoimakustannuksissa. Sorflaten tuo esiin, että käytettävyydelle laskettu sijoitetun pääoman tuotto on usein liian epätarkka, koska laskelmissa käytetään käytettävyydestä saatuja keskimääräisiä arvoja. Hän esittää parannusehdotuksena virhemarginaalin käyttämistä keskiarvoille. Sorflatenin laskelman vaiheet ovat:

1. Tunnista säästöjen laajuus.
2. Laske säästöaste.
3. Tunnista säästöjen lähde.
4. Laske vuosittaiset kokonaissäästöt.
5. Tunnista parannusten kustannukset.
6. Laske lopullinen sijoitetun pääoman tuotto.

Ensimmäinen vaihe on säästöjen laajuuden tunnistaminen. Sorflatenin esimerkissä lasketaan sijoitetun pääoman tuotto asiakaspalveluohjelmistolle. Kyseessä on järjestelmä, jota käyttää 5000 asiakaspalvelijaa. Toinen vaihe on säästöasteen laskeminen, johon tarvitaan työntekijöiden keskimääräinen ladattu tuntipalkka. Esimerkissä asiakaspalvelijan keskimääräinen ladattu tuntipalkka vuodessa on 80 000 dollaria, ja jos vuodessa on 2000 työtuntia, saadaan ladatuksi tuntipalkaksi 40 dollaria. Jos tunnissa tulee keskimäärin kymmenen puhelua asiakaspalvelijalle, 40 dollarin tuntipalkka tarkoittaa sitä, että yksi puhelu maksaa 4 dollaria, eli 0,0111 dollaria per sekunti.

Kolmas vaihe on säästöjen lähteen tunnistaminen. Esimerkissä arvioidaan, että uudelleensuunnittelulla voitaisiin lyhentää tietyn työvaiheen kestoja 20 sekuntia. Tällöin asiakaspalvelijan aikaa säästyisi 20 sekuntia per puhelu: 0,0111 dollaria per sekunti kerrottuna 20 sekunnilla on 22 senttiä per puhelu. Tunnissa tulee keskimäärin kymmenen puhelua, joten tunnissa säästyisi 2,22 dollaria. Vuosittaiset säästöt yhtä asiakaspalvelijaa kohden saadaan, kun kerrotaan 2,22 dollaria 2000 tunnilla, eli 4 440 dollaria per vuosi.

Neljäs vaihe on vuosittaisten kokonaissäästöjen laskeminen. Kokonaissäästöt saadaan kertomalla yhteen asiakaspalvelijaan kohdistuvat säästöt asiakaspalvelijoiden määrällä,

siis 4 440 dollaria kerrottuna 5000:lla. Säästöä tulisi siis vuodessa 22,2 miljoonaa dollaria. Viides vaihe on parannusten kustannusten tunnistaminen. Projektin kustannuksia ovat esimerkiksi ohjelmointi ja dokumentointi, käytettävyytyö sekä henkilöstön uudelleen kouluttaminen. Niiden arvioidaan maksavan yhteensä 2,7 miljoonaa dollaria. Kuudes vaihe on lopullisen sijoitetun pääoman tuoton laskeminen. Kun vähennetään arvioidut kustannukset arvioiduista säästöistä, saadaan tulokseksi, että säästöä tulee 19,5 miljoonaa dollaria. Sijoitetun pääoman tuotto prosentti saadaan, kun kerrotaan voiton ja sijoitetun pääoman osamäärä 100:lla. Sijoitetun pääoman tuotto prosentti on siis 722 prosenttia.

Sorflaten kommentoi, että arviot ja keskiarvot tuovat laskelmaan epätarkkuutta, eikä käytettävyytsteissä saatu arvio ajansäästöä ole yleistettävissä sellaisenaan. Sorflaten nostaa esiin tärkeän kysymyksen: mistä voidaan tietää, että muutos todella saa aikaan 20 sekunnin säästön? Jos käytettävyytsteihin osallistuu muutama henkilö ja testeissä saadut keskiarvot tehtävien suoritusajasta yleistetään sadoille henkilöille, virhemarginaalin laskeminen on tarpeellista. Valitaan siis luottamustaso, esimerkiksi 95 prosenttia, mikä tarkoittaa sitä, että 95 prosentissa tapauksista ajansäästö on arvioidulla välillä. Näin voidaan selvittää, ovatko käytettävyytsteissä saadut luvut luotettavia, vai tarvitaanko lisää testausta.

4.2.5 Rajperin malli

Rajper ym. (2012) ovat kehittäneet Mayhew'n ja Tremainen mallin pohjalta matemaattisen yhtälön käytettävyyden kustannushyödyn laskemiseen. Yhtälöön on tiivistetty mallin perusidea: hyötyjen ja kustannusten erotus.

$$UCJ = TPB - TUC, \quad (4.6)$$

jossa UCJ on käytettävyykustannusten perustelu (engl. usability cost justification), TPB on potentiaalinen kokonaishyöty (engl. total potential benefit) ja TUC on käytettävyyden kokonaiskustannus (engl. total usability cost). Rajperin ym. mallissa laskelma aloitetaan käytettävyyden kustannusten laskemisesta, jota varten on oltava tiedossa suunnitellut käytettävyytyön menetelmät ja analyysin kannalta oleelliset parametrit, joita voivat olla esimerkiksi käyttäjien määrä, palkka, ohjelmistokehittäjien palkka ja niin edelleen. Pohjatyönä selvitetään siis ensin parametrit, jotta pystytään laskemaan TUC eli käytettävyyden kokonaiskustannus.

Taulukossa 4.6 esitellään yleistetty muoto analyysiparametreista, jota voi soveltaa eri näkökulmiin. Todellisuudessa parametreja voi siis olla enemmän, ja ne ovat tapauskohtaisia. Ladattu tuntipalkka tarkoittaa sitä, että työntekijälle maksettavan summan lisäksi huomioidaan myös kaikki muut kulut mitä työntekijästä on yritykselle. Mayhew'n ja Tremainen (2005) mukaan laskelmassa voi käyttää tuntipalkkaa kerrottuna kahdella, mikäli ei ole mahdollisuutta selvittää yksityiskohtaisesti työntekijän kuluja.

Käytettävyyden kokonaiskustannusten TUC laskemisen vaiheet ovat:

Taulukko 4.6. Yleistetty muoto kustannuslaskelman analysointiparametreista Rajperin ym. (2012) mallissa. Mukailtu lähteestä Rajper et al. 2012.

Analyysiparametri	Arvo
Käyttäjien määrä	n
Käyttäjien työpäivien määrä per vuosi	n
Käyttäjien tuntipalkka (ladattu)	n
Ohjelmistokehittäjän tuntipalkka (ladattu)	n
Käytettävyyssinsinöörin tuntipalkka (ladattu)	n
Päällikön tuntipalkka (ladattu)	n
n	n

1. Käyttäjän kustannusten laskeminen. Kaavalla lasketaan kaikkien käytettävyyssyöissä mukana olevien henkilöiden kustannukset, ei siis pelkästään käyttäjien. Paremmiin kuvaava muuttujan nimi olisi esimerkiksi *henkilö*, mutta malli esitellään tässä työssä sellaisena kuin Rajper ym. ovat sen julkaisseet. Käytettävyyssyöhön osallistuvan henkilön kustannukset lasketaan kaavalla

$$käyttäjä = n \cdot tunnit \cdot palkka, \quad (4.7)$$

jossa n on henkilöiden lukumäärä, joilla on saman suurinen tuntipalkka, ja jotka käyttävät saman verran työtunteja käytettävyyssyöhön.

2. Tehtävän kustannusten laskeminen. Yhden tehtävän suorittamiseen tarvitaan usein useamman henkilön työpanosta. Oletetaan siis, että käyttäjien (henkilöiden) määrä on u . Yksittäisen tehtävän kustannukset saadaan laskemalla yhteen kaikkien osallistuvien henkilöiden kustannukset, eli

$$\begin{aligned} tehtävä &= \sum_{u=1}^m käyttäjä_u \\ &= käyttäjä_1 + käyttäjä_2 + \dots + käyttäjä_m, \end{aligned} \quad (4.8)$$

jossa *käyttäjä* lasketaan kaavalla 4.7.

3. Työvaiheen kustannusten laskeminen. Yhden työvaiheen suorittaminen koostuu useammasta tehtävästä. Näin ollen työvaiheen kustannukset saadaan laskettua tehtävien kustannusten summana, eli

$$\begin{aligned} vaihe &= \sum_{s=1}^o tehtävä_s \\ &= tehtävä_1 + tehtävä_2 + \dots + tehtävä_o, \end{aligned} \quad (4.9)$$

jossa *tehtävä* lasketaan kaavalla 4.8.

4. Käytettävyyden kokonaiskustannuksen laskeminen. Käytettävyyssyön kokonaiskustannukset lasketaan kaikkien työvaiheiden summana. Olkoon vaiheiden lukumäärä k . Tällöin

käytettävyyden kokonaiskustannus on

$$\begin{aligned}
 TUC &= \text{käytettävyyden kokonaiskustannus} \\
 &= \sum_{i=1}^k \text{kustannus}_i \\
 &= \text{vaihe}_1 + \text{vaihe}_2 + \dots + \text{vaihe}_k
 \end{aligned}
 \tag{4.10}$$

jossa *vaihe* lasketaan kaavalla 4.9.

Potentiaalisen kokonaishyödyn *TPB* laskemista varten tarvitaan hyötykategoriat, jotka valitaan ja lasketaan kuten Mayhew'n ja Tremainen mallissa. Rajper ym. ovat poimineet esimerkiksi hyötykategorioiksi käyttäjän aikaansaavuuden kasvamisen ja käyttäjän tekemien virheiden vähenemisen, jotka soveltuvat etenkin sisäisessä käytössä olevan ohjelmiston käytettävyyden hyötyjen laskemiseen. Hyötykategoriat ja hyötyjen laskuesimerkit on esitetty taulukossa 4.7. Potentiaalisia hyötykategorioita voi olla useampia eivätkä ne ole ennalta määrättyjä: yleistä tapausta kuvastaa muuttuja z .

Taulukko 4.7. Rajperin ym. (2012) esimerkki potentiaalisten hyötyjen laskemisesta, alunperin mukailtu Mayhew'n ja Tremainen (2005) mallista.

Potentiaaliset hyötykategoriat	Laskelma	Yhteensä
Käyttäjän aikaansaavuuden kasvaminen	$\text{päivien määrä} \cdot \text{käyttäjien määrä} \cdot \text{tapahtumien määrä} \cdot \text{säästettyjen tuntien määrä} \cdot \text{tuntipalkka}$ Esimerkki: $250 \cdot 230 \cdot 100 \cdot 0.001389 \cdot 25\$$	199 652.78 \$
Käyttäjän tekemien virheiden väheneminen	$\text{käyttäjien määrä} \cdot \text{päivien määrä} \cdot \text{eliminoitujen virheiden määrä} \cdot \text{tunnit per säästetty virhe} \cdot \text{tuntipalkka}$ Esimerkki: $250 \cdot 230 \cdot 1.0 \cdot 0.03333 \cdot 25\$$	47 916.67 \$
z	$v_1 \cdot v_2 \cdot \dots \cdot v_p$	xyz \$

Kuten taulukon 4.7 esimerkistä huomataan, potentiaalinen hyöty on muuttujien tulo. Potentiaalisen hyödyn laskemisen kaava on

$$\begin{aligned}
 PB &= \prod_{a=1}^p V_a \\
 &= v_1 \cdot v_2 \cdot \dots \cdot v_p,
 \end{aligned}
 \tag{4.11}$$

jossa v on potentiaaliseen hyötyyn liittyvä muuttuja. Muuttujia voivat olla esimerkiksi käyttäjien määrä tai eliminoitujen virheiden määrä. Potentiaalinen kokonaishyöty *TPB* laske-

taan kaavalla

$$\begin{aligned}
 TPB &= \text{potentiaalinen kokonaishyöty} \\
 &= \sum_{jk=1}^z PB_{jk} \\
 &= PB_1 + PB_2 + \dots + PB_z,
 \end{aligned}
 \tag{4.12}$$

jossa summataan kaavalla 4.11 lasketut potentiaaliset hyödyt yhteen. Kun käytettävyyden kokonaiskustannukset TUC ja potentiaalinen kokonaishyöty TPB on selvitetty, voidaan käytettävyyuskustannusten perustelu UCJ laskea kaavalla 4.6.

4.2.6 Esimerkkitapauksia

Montaniz ja Kissel (1995) lähestyvät aihetta epäonnistumisen kustannusten kautta. He tutkivat yrityksen lippulaivatuotteen käyttäjätukeen saapuneita puheluita, ja he törmäsivät listaan ongelmallisista virheviesteistä, joita sovellus antaa. Käyttäjätukitiimi oli yrittänyt vakuuttaa kehitystiimiä virheviestien parantelusta, koska niihin liittyviä yhteydenottoja tuli paljon. Montaniz ja Kissel selvittivät, mitkä ovat kustannukset, jos virheviesteille ei tehdä mitään. He toteuttivat kaksi kyselyä käyttäjätukitiimille, joiden avulla selvitettiin, mitkä virheviestit olivat yleisimpiä, ja kuinka usein käyttäjätukeen tuli niistä yhteydenottoja.

Kululaskelmaan otettiin käytettävyyistyön kulut ja käyttäjätuen kulut. Virheviestien aiheuttaman työn hinnaksi kuuden kuukauden ajalta saatiin 411 918 dollaria. Laskelman tekemisen hinta oli 533 dollaria. Montanizin ja Kisselin mukaan kehitysjohtaja oli todennut heidän artikkelinsa luettuaan, että jos virheviestien aiheuttamat yhteydenotot vähenisivät edes 25%, niin siitä vapautuvalla rahalla voitaisiin kattaa kolmen uuden tukihenkilön palkkaaminen.

Karat (1993) esittelee tekemänsä tapaustutkimukset kahdesta eri projektissa, joissa toteutettiin käytettävyyistyötä sisäisessä käytössä olevien ohjelmistojen parissa. Ensimmäisessä projektissa liiketoiminnan tukijärjestelmä korvattiin uudella ja siihen lisättiin uusi toiminto. Seitsemän kuukauden aikana toteutettiin kolme iteraatiota prototyyppien ja testiversioiden käytettävyydestejä (sekä kentällä että käytettävyydlaboratoriossa) ja lisäksi tehtäväänalyysiä, käyttöliittymien suunnittelua, tulosten raportointia ja uudelleensuunnittelusuosituksia. Käytettävyyden kustannushyöty laskettiin käyttäjän aikaansaavuuden kasvamisen näkökulmasta, ja lopputulos oli, että käyttäjän aikaansaavuuden kasvaminen sai aikaan 41 700 dollarin säästön, kun lukuja verrattiin ensimmäisen prototyypin ja lopullisen version välillä. Kustannus-hyötysuhde tässä tapauksessa oli 1:2. Ohjelmistokehittäjien mukaan he olisivat toteuttaneet ensimmäisen prototyypin mukaisen käyttöliittymän, mikäli sitä ei olisi käytettävyyistyön myötä iteroitu.

Toisessa projektissa käytettiin samoja menetelmiä kuin edellisessä. Nyt käytettävyyden kustannushyöty mitattiin käyttäjän aikaansaavuuden kasvamisena yhtä liiketapahtumaa kohti järjestelmän ensimmäisen käyttövuoden aikana. Tässäkin tapauksessa verrattiin

ensimmäisen prototyypin lukuja lopullisen version lukuihin. Käyttäjän aikaansaavuuden kasvamisen avulla saatiin 6 800 000 dollarin säästöt, ja kustannus-hyötysuhteeksi tuli 1:100.

5 YHTEENVETO

Tämän kandidaatintyön tutkimuskysymyksenä oli, **miten käytettävyyteen panostamalla voidaan säästää ohjelmistotuotteen elinkaaren kustannuksissa**. Työ toteutettiin systemaattisena kirjallisuuskatsauksena. Aineistosta etsittiin käytettävyyden hyötyjä, joilla voidaan arvioida olevan yhteys säästöjen syntymiseen tuotteen elinkaaren aikana, sekä malleja, joilla käytettävyyden tuomaa säästöä voidaan laskea. Lisäksi työssä tutustuttiin käytettävyyden teoriaan, mittareihin ja suunnittelu-, mallinnus- ja arviointimenetelmiin, sekä ohjelmiston elinkaareen.

Tutkimusaineisto haettiin tietotekniikan alan keskeisimmistä elektronisista tietokannoista: ACM Digital Library, Computer Science Database (ProQuest), IEEE, ScienceDirect (Elsevier), Scopus (Elsevier) sekä SpringerLink. Katsaukseen valittiin ennalta määrättyjen sisäänottokriteerien mukaan 9 tutkimusta ja artikkelia vuosilta 1988-2012. Lisäksi aineistoon otettiin mukaan materiaalia Cost-Justifying Usability -kirjasta. Tärkein sisäänottokriteeri oli, että käytettävyyden kustannusvaikutukset ovat keskeisessä osassa.

Kirjallisuuskatsaukseen valitut hakusanat toivat paljon hakutuloksia. Vielä haku suodattimien asettamisen jälkeenkin hakutuloksia oli melkein 5000. Sisäänottokriteereihin sopivien tutkimusten valitseminen tehtiin huolella, mutta hakutulosten määrän vuoksi alkuvaiheen karsinnassa oli edettävä nopeasti. Myös lähiluvussa oli priorisoitava, jotta aika riittäisi, ja siksi aineistoon otettiin mukaan vain tutkimuksia, joissa käytettävyyden kustannusvaikutukset olivat erittäin keskeisessä osassa. On mahdollista, että relevantteja tutkimuksia on jäänyt aineiston ulkopuolelle. Tutkimus ei aikarajoitteen vuoksi ole niin laaja kuin voisi olla.

Aineistossa yllättävää oli se, että vuosia sitten julkaistut tutkimukset olivat tärkeimmiltä osin edelleen relevantteja. Laskelmissa esitetyt luvut eivät välttämättä pidä paikkaansa enää tänä päivänä, mutta aineistossa esiintyviä käytettävyyden menetelmiä käytetään edelleen. Toinen yllättävä asia oli, että konkreettisia laskelmia käytettävyyden kustannushyödyistä ei juurikaan löytynyt, vaan lähes kaikki laskelmat ja mallit olivat esimerkinomaisia. Syynä saattaa olla se, että käytettävyyden kustannushyödyn laskeminen voi tapauksesta riippuen olla monimutkaista. Kustannus-hyötyanalyysia ei välttämättä tehdä, ellei ole erityistä tarvetta vakuuttaa jotakuta käytettävyyden tarpeellisuudesta.

Tutkimuksessa havaittiin, että käytettävyydellä on paljon hyötyjä, jotka voivat auttaa aikaansaamaan säästöä ohjelmistotuotteen elinkaaren aikana, ja että käytettävyyden kustannushyödyn laskemiseen on olemassa valmiita malleja. Aineistossa esitetyt käytettä-

vyyden kustannushyödyt on koottu kuvaan 5.1. Hyötyjä voidaan tarkastella sisäisestä tai ulkoisesta näkökulmasta. Jaottelu on tarpeen, sillä hyödyillä on vaihtelevan suuruinen merkitys eri osapuolille. Sisäinen näkökulma tarkoittaa käytettävyyden kustannushyötyjä yrityksen käytössä olevien ohjelmistojen näkökulmasta, ja ulkoinen näkökulma tarkoittaa käytettävyyden kustannushyötyjä ohjelmistojen tuottavan yrityksen näkökulmasta.



Kuva 5.1. Käytettävyyden hyödyt kustannusnäkökulmasta.

Käytettävyyden kustannusten ja hyötyjen laskemiseen löytyy useita tapoja, joista tässä työssä käsiteltiin Mantein ja Teoreyn malli (1988), Donahuen malli (2001), Mayhew'n ja Tremainen malli (2005), Sorflatenin malli (2006) ja Rajperin ym. malli (2012). Kattavimpana mallina voidaan pitää Mayhew'n ja Tremainen mallia, joka alun perin julkaistiin vuonna 1994. Siihen pohjautuvat esimerkiksi Donahuen ja Rajperin ym. mallit, ja myös Sorflaten käyttää lähteenään Mayhew'n ja Tremainen materiaalia. Mayhew'n ja Tremainen mallista on annettu esimerkit sekä sisäisestä että ulkoisesta näkökulmasta. Näiden lisäksi on annettu myös verkkokaupoille ja tuotetietosivustoille räätälöidyt esimerkit.

Mantein ja Teoreyn malli on ennemminkin laskelma, jossa selvitetään, mitä kustannuksia käytettävyyden menetelmät tuovat ohjelmistoprojektiin. Muutamille käytettävyyden aikaansaamille hyödyille on annettu laskukaavat Mantein ja Teoreyn artikkelissa. Mantein ja Teoreyn julkaisu on vanhin käytettävyyden kustannusvaikutuksista julkaistu tutkimus, joka osui tämän työn hakutermeihin ja sisäänottokriteereihin.

Donahuen mallin esimerkissä käytettävyyden tuomaa säästöä lähestytään sisäisestä nä-

kökulmasta ja hyötykategoriana on käyttäjän aikaansaavuuden kasvaminen. Donahuen mallia ei esitelty artikkelissa kovin kattavasti, eikä esimerkissä huomioitu käytettävyyssyön kustannuksia, kun selvitettiin kustannus-hyötysuhdetta. Kustannus-hyötysuhde väärin, jos käytettävyyssyön kustannuksia ei oteta huomioon.

Myös Sorflatenin esimerkissä käytettävyyden kustannusvaikutuksia tarkastellaan sisäisestä näkökulmasta, ja hyötykategoriana on käyttäjän aikaansaavuuden kasvaminen. Sorflatenin tärkein viesti on virhemarginaalin käyttäminen keskiarvoille, jotta voidaan varmistua siitä, ovatko esimerkiksi käytettävyyssyöissä saadut aika-arviot luotettavia.

Rajperin ym. malli muuttaa Mayhew'n ja Tremainen mallin matemaattisen yhtälön muotoon. Matemaattisesta yhtälöstä mallin perusidea voi olla helpompi ymmärtää, mutta on silti tunnettava Mayhew'n ja Tremainen malli, jotta yhtälöä voisi käyttää tarkoituksenmukaisesti. Myös Rajperin ym. mallissa esimerkki on esitetty sisäisestä näkökulmasta. Perusidea kaikissa malleissa on loppujen lopuksi aina sama: verrataan käytettävyyssyön kustannuksia sen aikaansaamiin hyötyihin. Kaikkien tässä työssä esiteltyjen laskelmien ja mallien vaiheet on koottu liitteeseen B.

Jatkotutkimusaiheena voisi olla kustannus-hyötymallien parametrien ja esimerkkien päivittäminen, sillä aineisto painottuu aikaan ennen 2000-lukua. Muita jatkotutkimusaiheita voisivat olla käytettävyyden kustannusten ja hyötyjen merkityksen tutkiminen kuluttajille myytävissä tuotteissa, ja kustannus-hyötymallien laajentaminen käyttäjäkokemukseen ja käyttäjäkeskeiseen suunnitteluun. Aineistossa lähes kaikki laskelmat olivat esimerkinomaisia, joten yksi jatkotutkimusaihe voisi olla tapaustutkimus käytettävyyden kustannuksista ja hyödyistä, ja niiden vertailu kirjallisuuteen.

Tutkimus osoittaa, että käytettävyydellä on paljon hyötyjä, jotka voivat aikaansaada säästöä tuotteen elinkaaren aikana. Käytettävyyssyö ei siis suinkaan ole pelkkä lisäkustannus. Kirjallisuudessa hyödyt esiintyvät usein hyvin selkeinä, mutta todellisuudessa hyötyjen ilmenemiseen voi kulua kauankin. Joissain tapauksissa on vaikea arvioida mitä olisi tapahtunut, jos käytettävyyssyö olisi jätetty tekemättä. On kuitenkin tarpeellista pohtia sitä, että esimerkiksi käyttöliittymäkulut realisoituvat aina jossain muodossa, vaikka käyttöliittymän toteutuksessa ei tietoisesti hyödynnettäisikään käytettävyyssyön menetelmiä.

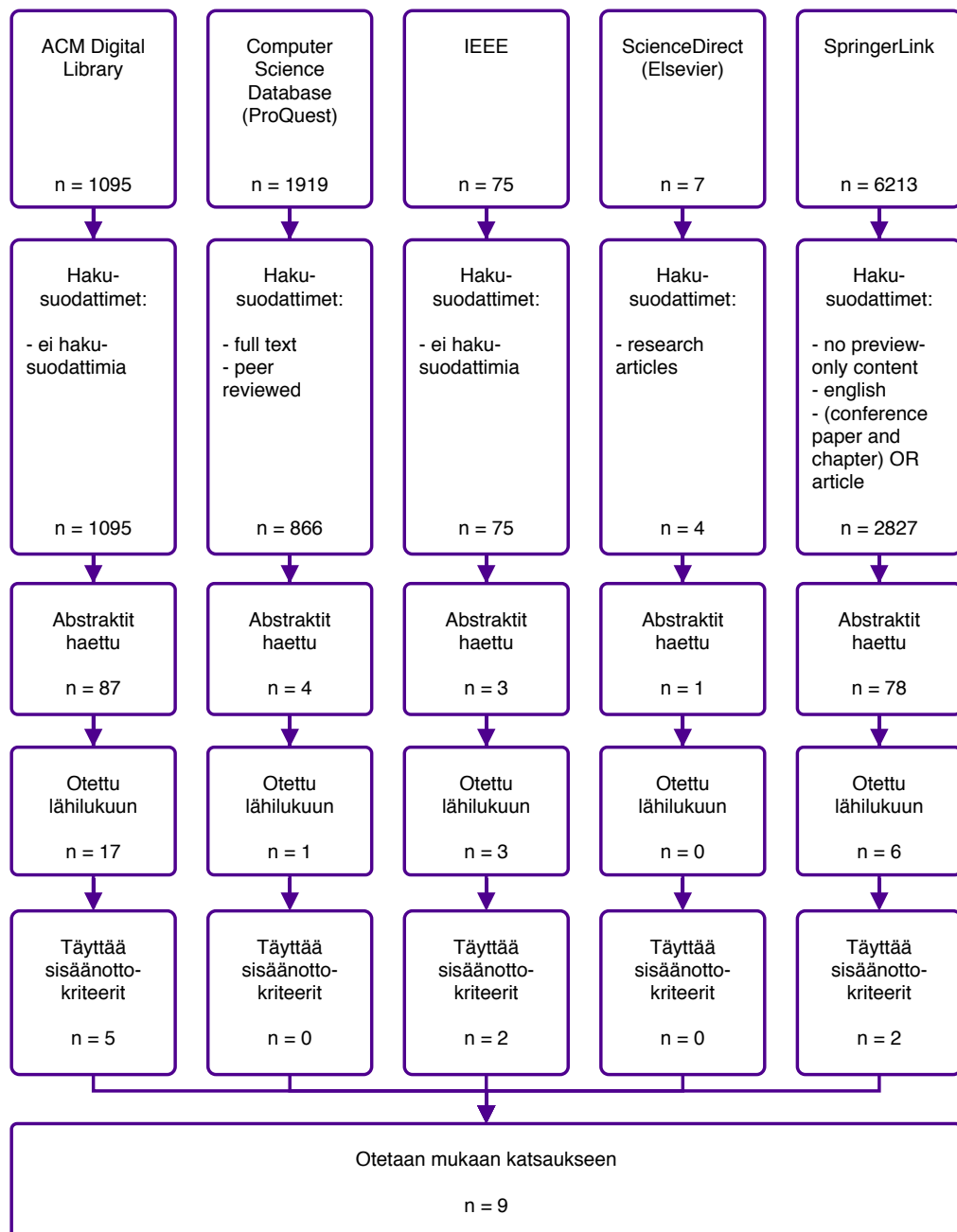
Ohjelmistoprojekteissa on hyvä muistaa, että ohjelmistot ovat tyypillisesti käytössä vuosia, ja muutosten tekeminen on edullisinta suunnitteluvaiheessa. Jos käytettävyyssyö jätetään tekemättä, saatetaan tuotteen elinkaaren ajan maksaa suuria summia piilokustannusten muodossa, vaikka se olisi ollut vältettävissä. Ohjelmiston elinkaaren aikana huonon käytettävyyden seuraukset ehtivät tulla kalliiksi.

LÄHTEET

- Adobe (2020). *Adobe XD - Compare Plans (Business)*. Luettu 21.3.2020. URL: <https://www.adobe.com/products/xd/compare-plans.html>.
- Donahue, G. M. (2001). Usability and the bottom line. *IEEE software* 18.1, pp. 31–37.
- Dray, S. (1995). The importance of designing usable systems. *interactions* 2.1, pp. 17–20.
- Figma (2020). *Figma Pricing*. Luettu 21.3.2020. URL: <https://www.figma.com/pricing>.
- Haikala, I. ja Mikkonen, T. (2011). *Ohjelmistotuotannon käytännöt*. Talentum Media Oy, pp. 23–26.
- Hornbæk, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. *International journal of human-computer studies* 64.2, pp. 79–102.
- InVision (2020). *InVision Plans*. Luettu 21.3.2020. URL: <https://www.invisionapp.com/plans>.
- ISO 9241-11:2018 (2018). *ISO 9241-11:2018 Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts*. Standardi. International Organization for Standardization.
- ISO 9241-210:2019 (2019). *ISO 9241-210:2019 Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems*. Standardi. International Organization for Standardization.
- Karat, C.-M. (1993). Usability engineering in dollars and cents. *IEEE software* 10.3, pp. 88–89.
- Korvenranta, H. (2005). Asiantuntija-arvioinnit. *Käytettävyytutkimuksen menetelmät*. Toim. S. Ovaska, A. Aula ja P. Majaranta. Tampereen yliopisto, tietojenkäsittelytieteiden laitos. B-2005-1., pp. 111–124.
- Koskinen, J. (2005). Käytettävyytestaus. *Käytettävyytutkimuksen menetelmät*. Toim. S. Ovaska, A. Aula ja P. Majaranta. Tampereen yliopisto, tietojenkäsittelytieteiden laitos. B-2005-1., pp. 187–208.
- Mantei, M. M. ja Teorey, T. J. (1988). Costbenefit analysis for incorporating human factors in the software lifecycle. *Communications of the ACM* 31.4, pp. 428–439.
- Marcus, A. (2005). User Interface Design's Return on Investment: Examples and Statistics. *Cost-justifying usability: An update for the Internet age*. Toim. R. G. Bias ja D. J. Mayhew. 2. painos. Morgan Kaufmann, pp. 17–36.
- Mayhew, D. J. ja Tremaine, M. M. (2005). A Basic Framework. *Cost-justifying usability: An update for the Internet age*. Toim. R. G. Bias ja D. J. Mayhew. 2. painos. Morgan Kaufmann, pp. 41–101.
- Montaniz, F. ja Kissel, G. V. (1995). Reversing the charges. *interactions* 2.3, pp. 29–33.
- Nielsen, J. (1993a). Is usability engineering really worth it? *IEEE software* 10.6, pp. 90–92.

- Nielsen, J. (1993b). *Usability Engineering*. Academic Press, Inc, pp. 23–26.
- Ovaska, S., Aula, A. ja Majaranta, P. (2005). Johdatus käytettävyytutkimukseen. *Käytettävyytutkimuksen menetelmät*. Toim. S. Ovaska, A. Aula ja P. Majaranta. Tampereen yliopisto, tietojenkäsittelytieteiden laitos. B-2005-1., pp. 1–6.
- Petticrew, M. ja Roberts, H. (2008). *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons.
- Rajanen, M. ja Iivari, N. (2007). Usability cost-benefit analysis: How usability became a curse word?: *Human-Computer Interaction – INTERACT 2007*. Toim. C. Baranauskas, P. Palanque, J. Abascal, S. Diniz ja J. Barbosa. IFIP International Federation for Information Processing. Springer, pp. 511–524.
- Rajper, S., Shaikh, A. W., Shaikh, Z. A. ja Amin, I. (2012). Usability cost benefit analysis using a mathematical equation. *International Multi Topic Conference 2012*. Toim. Chowdhry, B. S. et al. Springer, pp. 349–360.
- Ranne, S. (2005). Kognitiivinen läpikäynti. *Käytettävyytutkimuksen menetelmät*. Toim. S. Ovaska, A. Aula ja P. Majaranta. Tampereen yliopisto, tietojenkäsittelytieteiden laitos. B-2005-1., pp. 125–140.
- Sommerville, I. (2016). *Software Engineering*. Pearson Education Limited.
- Sorflaten, J. (2006). Making the fuzzy parts of ROI clear. *interactions* 13.6, pp. 38–41.
- Sutton, S. (2007). Review of "Cost-Justifying Usability: An Update for the Internet Age" (2nd Ed.) by Randolph Bias and Deborah Mayhew, Editors. *Interactions* 14.5, pp. 48–50. ISSN: 1072-5520.
- Tampereen yliopiston kirjasto (2020). *Tietojenkäsittely ja tietotekniikka: Artikkelit ja tietokannat*. Luettu 5.1.2020. URL: <https://libguides.tuni.fi/tietojenkäsittely>.
- usability.gov (2020). *Prototyping*. Luettu 24.5.2020. URL: <https://www.usability.gov/how-to-and-tools/methods/prototyping.html>.
- Vanhala, T. (2005). Kyselylomakkeet käytettävyytutkimuksessa. *Käytettävyytutkimuksen menetelmät*. Toim. S. Ovaska, A. Aula ja P. Majaranta. Tampereen yliopisto, tietojenkäsittelytieteiden laitos. B-2005-1., pp. 17–36.
- Vuorela, S. (2005). Haastattelumenetelmät. *Käytettävyytutkimuksen menetelmät*. Toim. S. Ovaska, A. Aula ja P. Majaranta. Tampereen yliopisto, tietojenkäsittelytieteiden laitos. B-2005-1., pp. 37–52.
- Wixon, D. (1995). Cost Justifying Usability Book Review by Dennis Wixon. *SIGCHI Bulletin* 27.2, pp. 94–97.

A TUTKIMUSAINEISTON HAKU- JA SISÄÄNOTTOPROSESSI TIETOKANNOITTAIN



Kuva A.1. Tutkimusaineiston haku- ja sisäänottoprosessi tietokannoittain.

B KUSTANNUS-HYÖTYLASKELMIEN VAIHEIDEN YHTEENVETO



Kuva B.1. Yhteenveto kustannus-hyötylaskelmien vaiheista.