

Pasi Palomäki

Projektin hinta-arvion määrittely käyttäen työmäärän arviointimallia

Opinnäytetyö

Kevät 2020

SeAMK Tekniikka

Konetekniikan tutkinto-ohjelma



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: SeAMK

Tutkinto-ohjelma: Konetekniikka

Suuntautumisvaihtoehto: Kone- ja tuotantotekniikka

Tekijä: Pasi Palomäki

Työn nimi: Projektin hinta-arvion määrittely käyttäen työmäärän arviointimallia

Ohjaaja: Heikki Heiskanen

Vuosi: 2020

Sivumäärä: 39

Liitteiden lukumäärä: -

Tämän opinnäytetyön tavoitteena oli tarkastella pienen insinööritoimiston asiakasprojektin hinta-arvion muodostumista käyttäen apuna rakentavan järjestelmätekniikan kustannus- eli COSYSMO-mallia. Työn teoriaosuudessa perehdyttiin COSYSMO-malliin ja lean-ohjelmistokehitykseen. Lean:lla pyrittiin estämään ja poistamaan turhia töitä.

Opinnäytetyö aloitettiin käymällä läpi käytössä oleva tarjouspohja. COSYSMO-mallin kokomuuttujien määrittelyä varten tehtiin käyttöohje. Käyttöohje lisättiin tarjouspohjan suunnittelu osaan. Käyttöohjeen mukaan täytetty tarjouspohja kokosi kokomuuttujat sisällysluetteloon. Kokomuuttujat täytettiin sisällysluettelosta valmiiseen Excel-malliin.

Opinnäytetyötä testattiin kahdella eri projektilla. Ensimmäinen projekti oli jo valmistunut projekti ja toinen keskeneräinen. Käyttöohjeet tarkennettiin ensimmäisen testikierroksen tulosten perusteella. Tarkentamisen jälkeen päästiin lähelle toteutunutta työmäärää. Opinnäytetyön toteutuksen käyttöönottoa suositeltiin yritykselle.

Avainsanat: COSYSMO, hinta-arvio, lean-ohjelmistokehitys

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Mechanical Engineering

Specialisation: Mechanical and Production Engineering

Author: Pasi Palomäki

Title of thesis: Defining a project cost estimate using the workload estimation model

Supervisor: Heikki Heiskanen

Year: 2020

Number of pages: 39

Number of appendices: -

The purpose of the thesis was to examine the cost estimation of a client project for a small engineering firm using the COSYSMO cost model of constructive system technology. The theoretical part of the thesis focused on the COSYSMO model and lean software development. Lean was used to prevent and eliminate unnecessary work.

The thesis was started by reviewing the current offer template. A manual was made to define the size variables for the COSYSMO model. The manual was added to the design section of the template. The template, completed according to the instruction manual, included size variables in the table of contents. The size variables were filled from the table of contents into a pre-built Excel template.

The thesis was tested in two different projects. The first project was already completed and the second one was unfinished. The operating instructions were refined based on the results of the first test round. After the refinement, the actual workload was approached. The company was recommended to implement the thesis.

Keywords: COSYSMO, price estimation, Lean software development

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	1
Thesis abstract.....	2
SISÄLTÖ.....	3
Kuvio- ja taulukkoluetelo.....	5
Käytetyt termit ja lyhenteet	6
1 JOHDANTO	7
1.1 Työn tausta	7
1.2 Työn tavoitteet.....	7
1.3 Työn rajaus	7
1.4 Työn rakenne	8
2 TEORIA	9
2.1 COSYSMO.....	9
2.1.1 COSYSMO-algoritmi.....	9
2.1.2 Projektin kokomuuttujat.....	10
2.1.3 Projektin hintamuuttujat.....	13
2.2 Lean-ohjelmistokehitys.....	18
2.2.1 Ajanhukan vähentäminen.....	18
2.2.2 Lisää oppimista	19
2.2.3 Viivästyä päätöksentekoa	19
2.2.4 Toimita niin nopeasti kuin mahdollista	20
2.2.5 Sitouta tiimi	21
2.2.6 Vastaavuus siihen, mitä on markkinoitu ja toimitettu.....	21
2.2.7 Optimoiki kokonaisuutta.....	22
3 YRITYS X:N TARJOUSPROSESSIN NYKYTILA.....	24
3.1 Tarjouspyynnön käsittely.....	24
3.2 Ongelmia.....	24
3.3 Vaatimukset toteutukselle	24
4 TUTKIMUSMENETELMÄT	25
4.1 Tutkimusmenetelmän rakenne.....	25
4.2 Projektin suunnittelupohja.....	25

4.3 COSYSMO-laskentamallin toteutus	26
4.4 Työohje	27
5 TUTKIMUSTULOKSET	30
5.1 Valmistuneen projektin aikavertailu.....	30
5.2 Keskenäisen projektin aikavertailu.....	31
5.3 Työohjeen parantaminen	33
6 JOHTOPÄÄTÖKSET JA SUOSITUKSET	35
7 YHTEENVETO.....	36
LÄHTEET	39

Kuvio- ja taulukkoluetelo

Kuvio 1. Projektisuunnitelman sisällysluettelo.....	28
Kuvio 2. Työohjeen integrointi suunnittelupohjaan.....	29
Kuvio 3. Päivitetty työohje	34
Taulukko 1. COSYSMO-Algoritmissa käytetyt suureet.	10
Taulukko 2. Järjestelmävaatimusten vaikeustason luokittelu.....	11
Taulukko 3. Järjestelmärajapintojen vaikeustason luokittelu.....	11
Taulukko 4. Järjestelmäkohtaisten algoritmien vaikeustason luokittelu.	12
Taulukko 5. Operatiivisten skenaarioiden vaikeustason luokittelu.	13
Taulukko 6. Järjestelmän kokomuuttujat yksinkertaiseen COSYSMO-laskentaan.	26
Taulukko 7. COSYSMO-tilauksen hintaparametrien valinta.	27
Taulukko 8. Valmistuneen projektin hintamuuttujat.....	30
Taulukko 9. Keskeneräisen projektin hintamuuttujat.....	32

Käytetyt termit ja lyhenteet

COSYSMO	Constructive Systems Engineering Cost Model antaa ar- vion henkilöstökuukausien lukumäärästä, joka kuuluu järjes- telmien suunnitteluun laitteisto- ja ohjelmistoprojekteissa.
Integroida	Yhdistää tai liittää osaksi jotakin.
Iteraatio	Iteraatiokierros on kuin pieni ohjelmistoprojekti, joka sisäl- tää kaikki uusien toimintojen julkaisemiseen tarvittavat teh- tävät. Iteraatiokierroksia toistetaan, kunnes saavutetaan haluttu lopputulos.
Kvantifioida	Ilmaista tai määrittää jokin määrä.
Optimoida	Parhaan vaihtoehdon etsimistä.
Refaktorointi	Yksinkertaisuuden, selkeyden ja ominaisuuksien vähim- mäismäärän säilyttämistä koodissa. Ohjelmistokoodin uu- delleen kirjoitusta kehityksessä edellä määriteltyjä pää- määriä tavoitellen.
Työkuukausi	Kuukauden sisältämä 152 tuntia työaika.
Validointiin	Jonkin asian luotettavuus tai paikkansapitävyys.

1 JOHDANTO

1.1 Työn tausta

Insinööritoimistossa X:ssä halutaan kehittää tarjouslaadintaprosessia. Tarjoukset uusista asiakasprojekteista laskee yksi asiantuntija kysyen tarkentavat lisäarviot projektin eri osien tekijöiltä. Aika arvioidaan kokemuksen perusteella, mihin lisätään riskikertoimet. Nykyisellä menetelmällä ei huomioida piilotyömääriä ja tämä aiheuttaa sekä aikataulullisia että taloudellisia ongelmia insinööritoimisto X:lle. Nykyinen menetelmä on nopea ja kevyt alle viiden hengen insinööritoimistolle.

Insinööritoimisto X:n asiakasprojektien hinta-arvion määrittelyyn ei ole mahdollista käyttää paljon aikaa, koska se vie aikaa asiantuntijoiden muilta töiltä. Asiantuntijat ovat tällä hetkellä ylityöllistettyjä.

Insinööritoimisto X:n asiakasprojektit jakaantuvat tai koostuvat pääosin kahdesta osiosta, joita ovat elektroniikka- ja ohjelmistokehitys. Insinööritoimistolle X:lle on tullut lisääntyvänä tarpeena mekaniikkasuunnittelu, joka on pääasiassa elektroniikalle tehtävää kotelosuunnittelua.

1.2 Työn tavoitteet

Opinnäytetyön tavoitteena on tutkia COSYSMO-mallin työmäärän laskentamenetelmän ja -mallin soveltuvuutta insinööritoimisto X:n asiakasprojektien hinta-arvion määrittelyyn. Toteutus sisältyisi osaksi nykyistä asiakasprojektin hinta-arvion määrittelyä.

1.3 Työn rajaus

Insinööritoimisto X:n pääasiallinen toimiala on IT-konsultointi, jossa yritys on erikoistunut sulautettujen järjestelmien kehittämiseen. Yrityksen projektit ovat pääasiallisesti ohjelmistokehitys- ja IT-projekteja. Tästä syystä projektimenetelmän ja kustan-

nusmallin on oltava ohjelmistokehitykseen ja IT-projektiin soveltuvia. Tätä mallia pyritään soveltamaan myös elektroniikka- ja mekaniikkasuunnittelun hinta-arvion määrittelyyn.

Insinööritoimisto X on alle viiden hengen yritys, joten asiakasprojektien hinta-arvion määrittelyyn on käytössä rajallinen aika. Toteutuksen tulee olla kevyt, helppokäyttöinen ja mahdollisimman tehokas.

1.4 Työn rakenne

Työn ensimmäinen osio on teoriaosuus, jossa käsitellään COSYSMO-mallia ja lean-ohjelmistokehitystä.

Työn toisessa osiossa kartoitetaan insinööritoimisto X:n nykyisiä määrittelymenetelmiä ja tarkastellaan nykyistä tilannetta. Osiossa luodaan myös uudelle menetelmälle vaatimusmäärittely, joka sopii työn rajaukseen.

Työn kolmannessa osuudessa kootaan vaatimusmäärittelyä vastaava toteutus.

Työn neljännessä osuudessa toteutusta testataan aikaisemmilla projekteilla sekä tarkastellaan saatuja tuloksia. Saaduista tuloksista pohditaan, onko toteutus toimiva.

Työn lopussa tehdään yhteenveto toteutuksen soveltuvuudesta sekä ilmenneistä puutteista.

2 TEORIA

2.1 COSYSMO

Rakentavan järjestelmätekniiikan kustannusmallilla autetaan ihmisiä pohtimaan järjestelmien suunnittelun taloudellisia vaikutuksia hankkeisiin. Malli kehitettiin Etelä-Kalifornian yliopistossa tutkimusprojektina BAE Systemsin, General Dynamicsin, Lockheed Martinin, Northrop Grummanin, Raytheonin ja SAIC:n avulla. COSYSMO-malli noudattaa parametrista mallinnusmenetelmää. Menetelmää käytetään arvioissa järjestelmätekniiikan työvoiman määrää henkilökuukausina, mikä tarvitaan suurten ohjelmisto- ja laitteistoprojektien suunnitteluun, testaamiseen ja käyttöönottoon. (Academic COSYSMO 2020.)

2.1.1 COSYSMO-algoritmi

Jokainen COSYSMO-algoritmin parametri on osa kustannusarvion suhdetta, jonka järjestelmän suunnittelijat määrittävät. Kaavassa 1 on esitetty, miten työ henkilökuukausina lasketaan. Taulukko 1 pitää sisällään suureet sekä niiden selitykset.

$$PM_{NS} = A \cdot \left(\sum_k (w_{e,k} \Phi_{e,k} + w_{n,k} \Phi_{n,k} + w_{d,k} \Phi_{d,k}) \right)^E \cdot \prod_{j=1}^{14} EM_j \quad (1)$$

Taulukko 1. COSYSMO-Algoritmissa käytetyt suureet.

Suure	Selitys
PM_{NS}	Työ henkilökuukausina
A	Historiallisista projektitiedoista johdettu kalibroitivakio
k	REQ, IF, ALG, SCN
w_x	Painoarvo helppo, normaalille tai vaikealle muuttujalle
Φ_x	k-muuttujan koko
E	Edustaa mittakaavaetuja
EM	Vaivanmuuttuja hintamuuttujaan

Järjestelmän koko on REQ-, IF-, ALG- ja SCN-parametrien painotettu summa ja se edustaa mallin additiivista osaa. EM-kerroin on puolestaan 14 kustannusmuuttujan tulos ja edustaa mallin kertovaa osaa. (Academic COSYSMO 2020.)

2.1.2 Projektin kokomuuttujat

Järjestelmävaatimusten lukumäärä edustaa kiinnostuksen kohteena olevaa järjestelmää koskevien vaatimusten lukumäärää tietyllä suunnittelutasolla. Kokomuuttujien määrään sisältyy vaatimukset, jotka liittyvät järjestelmän rajapintojen, järjestelmäkohtaisten algoritmien ja operatiivisten skenaarioiden suunnitteluun. (Academic COSYSMO 2020.)

Vaatimukset voivat olla erittelyyn käytetyn menetelmän mukaan luonteeltaan joko toiminnallisia, suorituskykyisiä, ominaisuus- tai palvelukeskeisiä. Myös asiakas tai urakoitsija voi määrittää vaatimukset. Jokaiseen vaatimukseen voi liittyä ylimääräistä työtä, kuten todentaminen, validointi, toiminnallinen hajoaminen tai toiminnallinen allokointi. Järjestelmävaatimukset voidaan tyypillisesti laskea järjestelmästä tai markkinointieritelmästä sovellettavien vaatimusten lukumääränä. Taulukossa 2 on

esitettyinä kriteerit järjestelmävaatimusten vaikeustason luokitteluun. (Academic COSYSMO 2020.)

Taulukko 2. Järjestelmävaatimusten vaikeustason luokittelu.

Helppo	Normaali	Vaikea
Helppo toteuttaa	Tuttua	Monimutkainen toteutettavaksi tai suunniteltavaksi
Pientä vaatimusten päällekkäisyyttä	Jonkin verran vaatimusten päällekkäisyyttä	Paljon vaatimusten päällekkäisyyttä

Järjestelmärajapintojen lukumäärä edustaa sisäisiä rajapintoja ja järjestelmän ulkopuolisia rajapintoja. Taulukossa 3 on esitettyinä kriteerit järjestelmärajapintojen vaikeustason luokitteluun. (Academic COSYSMO 2020.)

Taulukko 3. Järjestelmärajapintojen vaikeustason luokittelu.

Helppo	Normaali	Vaikea
Yksinkertainen viesti	Kohtalainen monimutkaisuus	Monimutkaiset protokollat
Kytkemätön	Löysästi kytketty	Hyvin kytketty toisiinsa
Vahva yksimielisyys	Kohtalainen yksimielisyys	Matala yksimielisyys
Hyvin käyttäytyvä	Ennustettavissa oleva käyttäytyminen	Huonosti käyttäytynyt

Järjestelmäkohtaisten algoritmien lukumäärä edustaa uusien määriteltyjen tai merkittävästi muutettujen toimintojen lukumäärää. Nämä edellyttävät yksilöivien matemaattisten algoritmien johtamista järjestelmän suorituskykyvaatimusten saavuttamiseksi. Tähän voisi sisältyä esimerkiksi monimutkainen lentokoneiden jäljitysalgoritmi, kuten Kalman-suodatin. Suodatin johdetaan käyttämällä olemassa olevaa kokemusta perustana kaikkien näkökohtien hakutoiminnolle. Luku voidaan määrittää laskemalla yksilöivien algoritmien lukumäärä, joka tarvitaan järjestelmämäärittelyk-

sessä määriteltyjen vaatimusten toteuttamiseksi. Taulukossa 4 on esitettyinä kriteerit järjestelmäkohtaisten algoritmien vaikeustason luokitteluun. (Academic COSYSMO 2020.)

Taulukko 4. Järjestelmäkohtaisten algoritmien vaikeustason luokittelu.

Helppo	Normaali	Vaikea
Algebra	Suoraan eteenpäin laske- minen	Monimutkainen ja rajoit- tettu optimointi; kaavan- tunnistus
Suora rakenne	Sisäkkäinen rakenne päätöksentekologiikalla	Rekursiivinen rakenne hajautetulla ohjauksella
Yksinkertainen data	Relaatiotiedot	Turhaa ja huonosti määri- telyä tietoa.
Ajoitus ei ole ongelma	Ajoituksen rajoitus	Dynaaminen, ajoitus- ja epävarmuustekijöiden kanssa
Kirjastopohjaisen ratkai- sun mukauttaminen	Jonkin verran mallinnusta mukana	Mukana simulaatio ja mallintaminen

Operatiivisten skenaarioiden lukumäärä edustaa operatiivisten skenaarioiden luku-
määrää, jonka järjestelmän on täytettävä. Tällaisia skenaarioita ovat kaikki nimellis-
virran ulkopuoliset ketjut, jotka johtuvat huonosta tai puuttuvasta tiedosta. Skenaa-
rioiden lukumäärä voidaan tyypillisesti kvantifioida laskemalla järjestelmän testilan-
kapakettien tai ainutlaatuisten päästä päähän -testausten lukumäärä, jota käytetään
järjestelmän toimivuuden ja suorituskyvyn validointiin. Taulukossa 5 on esitettyinä
kriteerit järjestelmän operatiivisten skenaarioiden luokitteluun. (Academic
COSYSMO 2020.)

Taulukko 5. Operatiivisten skenaarioiden vaikeustason luokittelu.

Helppo	Normaali	Vaikea
Hyvin määritelty	Löysästi määritelty	Ei määritelty
Löysästi kytketty	Kohtalainen kytkeytyneisyys	Tiiviisti kytkettyjä tai monia riippuvuuksia / ristiriitaisia vaatimuksia
Aikataulut eivät ole ongelma	Aikataulu rajoituksia	Tiukka aikataulu skenaarioverkoston kautta
Harvoja ja yksinkertaisia säiettä, joilla ei ole nimellisarvoja.	Kohtuullinen lukumäärä tai monimutkaisia epänominaalista säiettä.	Monia tai erittäin monimutkaisia nimellisarvoista säiettä.

2.1.3 Projektin hintamuuttujat

Mallin kustannustekijät edustavat esitetyn mallin kerrannaisvaikutusta. Näitä muuttujia kutsutaan myös vaivan kertoimiksi, koska ne vaikuttavat koko järjestelmän suunnittelutehtävien laskentaan. Luokitusten antaminen näille muuttujille ei ole yhtä suoraviivaista kuin aiemmin mainittujen kokomuuttujien. Ero on siinä, että suurin osa kustannustekijöistä on luonteeltaan laadullisia ja ne vaativat subjektiivista arviointia, jotta niitä voidaan arvioida. (Academic COSYSMO 2020.)

Ensimmäinen hintamuuttuja on vaatimusten ymmärtäminen. Vaatimusten ymmärtämys kustannustekijänä arvioi kaikkien sidosryhmien ymmärtämistä järjestelmävaatimuksissa. Sidosryhmiksi luetaan järjestelmät, ohjelmistot, laitteistot, asiakkaat, ryhmän jäsenet ja käyttäjät. Kustannusmuuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea, erittäin korkea. Erittäin matala luokan kriteerinä on kesken projektin syntyvät vaatimukset tai uudet järjestelmät. Erittäin matala luokka sisältää suuren työmäärän. Erittäin korkea luokan kriteerinä on puolestaan vaatimusten ymmärtäminen ja entuudestaan tutut järjestelmät. Erittäin korkea luokka sisältää pienen työmäärän. (Academic COSYSMO 2020.)

Toinen hintamuuttuja on arkkitehtuurin ymmärtämys. Arkkitehtuurin ymmärtämys kustannustekijä arvioi järjestelmän arkkitehtuurin määrittämisen ja hallinnan suhteellisia vaikeuksia. Näitä ovat esimerkiksi alustat, standardit, komponentit ja niiden rajoitukset. Tähän sisältyy tehtäviä, kuten systeemianalyysi, kompromissianalyysi, mallinnus, simulointi sekä tapaustutkimukset. Hintamuuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea, erittäin korkea. Erittäin matalan luokan kriteerinä on huono ymmärtämys arkkitehtuurista ja uudesta järjestelmästä, mikä puolestaan kasvattaa työmäärää. Erittäin korkean luokan kriteerinä on arkkitehtuurin täydellinen tuntemus ja tuttu järjestelmä, mikä vähentää työmäärää. (Academic COSYSMO 2020.)

Kolmas hintamuuttuja on palvelun vaatimukset. Palvelun vaatimuksen kustannustekijä arvioi vaikeuksia, kriittisyyttä, turvallisuutta, vasteaikaa, yhteen toimivuutta ja ylläpitokykyä. Hintamuuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea ja erittäin korkea. Luokkaa määritettäessä otetaan huomioon vaikeustaso ja kriittisyysaste. Erittäin matalan luokan kriteereitä ovat yksinkertainen vaikeustaso ja kriittisyysaste pienissä haitoissa. Erittäin matala luokka on pieni työmääräinen, kun taas erittäin korkeassa luokassa vaikeus on hyvin monimutkainen ja kriittisyysaste sisältää riskin ihmishengelle. Näiden takia erittäin korkean luokan työmäärä on suuri. (Academic COSYSMO 2020.)

Neljäs hintamuuttuja on aikaisempien järjestelmien yhteensovittaminen projektiin. Tämä hintamuuttuja arvioi sitä, missä määrin vanha järjestelmä vaikuttaa siirtymän monimutkaisuuteen, jos sellaista on. Vanhat järjestelmäkomponentit, tietokannat, työnkulut ja ympäristöt voivat vaikuttaa uuden järjestelmän toteutukseen ja uuden tekniikan käyttöönoton sekä suunniteltujen päivitysten, lisääntyneen suorituskyvyn ja liiketoimintaprosessien uudelleensuunnitteluun. Muuttuja on jaettu neljään luokkaan, joita ovat normaali, korkea, erittäin korkea ja todella korkea. Luokkaa määritettäessä muuttujassa otetaan huomioon vanhan urakoitsijan ja vanhan järjestelmän vaikutus uuteen järjestelmään. Normaali luokan kriteereitä ovat vanha järjestelmä on dokumentoitu hyvin tai kaikki on uutta. Normaali luokka pitää sisällään pienen työmäärän. Todella korkean luokan kriteerinä on, että alkuperäinen urakoitsija ei

toimi enää, eikä dokumentaatiota ole saatavilla. Aiempia järjestelmiä kuuluu integraatioon, kehitykseen, arkkitehtuuriin ja suunnitteluun. Todella korkea luokka pitää sisällään suuren työmäärän. (Academic COSYSMO 2020.)

Viides hintamuuttuja on teknologiariskit. Teknologinen riski edustaa toteutettavan tekniikan kypsyyttä, valmiutta ja vanhenemista. Epäkypsät tai vanhentuneet teknikat vaativat enemmän järjestelmäsuunnittelua. Muuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea ja erittäin korkea. Muuttujassa otetaan huomioon luokkaa määrittäessä kypsyyden ja valmiuden puute sekä vanheneminen. Erittäin matala luokka pitää sisällään todistetusti toimivaa teknologiaa ja sitä käytetään laajasti koko teollisuudessa. Erittäin matala luokka pitää sisällään vain pienen teknologiariskin. Erittäin korkean riskin teknologia on vielä kehitysvaiheessa, koska konsepti on vasta määritelty tai teknologia on vanhentunut ja käyttöä tulisi välttää uudessa järjestelmässä. (Academic COSYSMO 2020.)

Kuudes hintamuuttuja on dokumentaatio. Dokumentaatio vastaa elinkaaritarkoituksia ja edustaa muodollisesti toimitettavan dokumentaation muodollisuutta sekä yksityiskohtaisuutta järjestelmän elinkaaritarkoitusten perusteella. Muuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea ja erittäin korkea. Muuttujassa otetaan huomioon luokkaa määrittäessä muodollisuus ja yksityiskohtaisuus. Erittäin matala luokka sisältää vain yleiset tavoitteet, minimimäärän tai ei määritetyt dokumentointi- ja tarkistusvaatimuksia suhteessa elinkaaritarkoituksiin. Erittäin matala luokka on työmäärältään pieni. Erittäin korkea luokka noudattaa puolestaan tiukkoja standardeja ja vaatimuksia sekä sisältää laajat dokumentointi- ja tarkistusvaatimukset suhteessa elinkaaritarkoituksiin. Erittäin korkea luokka pitää sisällään suuren työmäärän. (Academic COSYSMO 2020.)

Seitsemäs hintamuuttuja on asennusten tai alustojen lukumäärä ja monimuotoisuus. Toimintaympäristössä, kuten avaruudessa, meressä, maassa tai mobiilissa tulee ottaa huomioon vastauksen painottaminen. Muuttuja on jaettu neljään luokkaan, joita ovat erittäin normaali, korkea, erittäin korkea ja todella korkea. Normaali kuvaa pientä työmäärää, kun taas todella korkea kuvaa todella suurta työmäärää. (Academic COSYSMO 2020.)

Kahdeksas hintamuuttuja on suunniteltujen rekursiivisten tasojen lukumäärä. Muuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea ja erittäin korkea. Muuttujassa otetaan huomioon luokkaa määrittäessä järjestelmä suunnittelun tasojen lukumäärä. Tässä muuttujassa erittäin matala kuvaa pientä työmäärää ja erittäin korkea taas suurta työmäärää. (Academic COSYSMO 2020.)

Yhdeksäs hintamuuttuja on sidosryhmien yhteenkuuluvuus. Sidosryhmäjohtajuus edustaa moniominaisuusparametria, joka sisältää johtajuuden, sidosryhmien monimuotoisuuden, hyväksyntäjaksot, ryhmädynamiikan ja vastuunmuutoksen määrän. Muuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea ja erittäin korkea. Muuttujassa otetaan huomioon luokkaa määrittäessä kulttuuri, yhteensopivuus ja perehtyneisyys. Erittäin matala kuvaa huonoa yhteenkuuluvuutta ja erittäin korkea hyvää yhteenkuuluvuutta. (Academic COSYSMO 2020.)

Kymmenes hintamuuttuja on henkilöstön tai ryhmän kyky analysoida monimutkaisia ongelmia ja luoda ratkaisuja. Muuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea ja erittäin korkea. Erittäin matalan luokan kriteerinä on viisitoista prosenttinen kyky analysoida monimutkaisia ongelmia ja luoda ratkaisuja. Erittäin matala luokka sisältää suuren työmäärän, kun puolestaan erittäin korkea luokka sisältää pienen työmäärän. Erittäin korkean luokan kriteerinä on yhdeksänkymmentä prosenttinen kyky analysoida monimutkaisia ongelmia ja luoda ratkaisuja. (Academic COSYSMO 2020.)

Yhdestoista hintamuuttuja on henkilöstön kokemus sekä vaihtuvuus projektin aikana. Muuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea ja erittäin korkea. Muuttujan luokittelussa otetaan huomioon kokemus ja vuosittainen liikevaihto. Erittäin matalan luokan kriteerinä on, että henkilöstön työkokemus on alle kaksi kuukautta ja projektin osuus vuosiliikevaihdosta on 48 prosenttia. Erittäin matala luokka sisältää suuren työmäärän, kun puolestaan erittäin korkea luokka pienen työmäärän. Erittäin korkean luokan kriteerinä on kymmenen vuoden kokemus ja projektin osuus vuosiliikevaihdosta on kolme prosenttia. (Academic COSYSMO 2020.)

Kahdestoista hintamuuttuja on prosessin kyky. Tämä muuttuja perustuu projektitiimin johdonmukaisuuteen ja tehokkuuteen suoritettaessa järjestelmäsuunnitteluprosesseja. Muuttuja on jaettu kuuteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea, erittäin korkea ja todella korkea. Luokkaa määritettäessä muuttujassa otetaan huomioon arviointi, projektitiimin käyttäytymisominaisuudet sekä järjestelmäsuunnittelun hallintasuunnitelman hienostuneisuus. Erittäin matala luokka kuvastaa huonoa prosessi kykyä. Erittäin matala luokka sisältää suuren työmäärän ja erittäin korkea luokka sisältää vain pienen työmäärän. Todella korkea luokka kuvastaa hyvää ymmärrystä. (Academic COSYSMO 2020.)

Kolmastoista hintamuuttuja on projektikoordinaatio. Muuttuja on jaettu kuuteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea, erittäin korkea ja todella korkea. Luokkaa määritettäessä muuttujassa otetaan huomioon sijainti, viestintä ja yritysten yhteistyön esteet. Erittäin matalan luokan ensimmäinen kriteeri on kansainväliset sijainnit, joihin vaikuttavat vakavasti aikavyöhyke-erot. Toisena kriteerinä on, että viestinnässä käytetään vähän puhelinta ja sähköpostia. Kolmas kriteeri on vakavat viesti- ja turvallisuusrajoitukset. Erittäin matala luokka sisältää suuren työmäärän, kun puolestaan erittäin korkea luokka sisältää vain pienen työmäärän. Todella korkean luokan vaatimukset ovat, että kaikki ovat samassa sijainnissa, viestinnässä käytetään interaktiivista viestintää eikä viesti- ja turvallisuusrajoituksia ei ole. (Academic COSYSMO 2020.)

Neljästoista hintamuuttuja on tuki työkalulle, joka kuvaa työkalujen kattavuutta, integrointia ja kypsyyttä. Muuttuja on jaettu viiteen luokkaan, joita ovat erittäin matala, matala, normaali, korkea ja erittäin korkea. Erittäin matalan luokan kriteerinä on, että käytettävät työkalut eivät ole järjestelmäsuunnittelutyökaluja. Erittäin matala luokka sisältää suuren työmäärän, kun puolestaan erittäin korkea luokka sisältää vain pienen työmäärän. Erittäin korkean luokan kriteerinä on prosessien ja mallipohjaisten järjestelmien suunnittelu sekä hallintajärjestelmien vahva ja osaava käyttö. (Academic COSYSMO 2020.)

2.2 Lean-ohjelmistokehitys

Lean-ohjelmistokehitys on käänös lean-ajattelun periaatteista ja käytännöistä ohjelmistokehitykseen. Lean on jatkuvan oppimisen ja kehityksen ajattelumalli. Ensimmäisen kerran lean-termistä puhuttiin ”Lean Software Development: An Agile Toolkit”-nimisessä kirjassa, jonka ovat kirjoittanut Mary ja Tom Poppendieck vuonna 2003. Lean-ohjelmistokehityksen 7 periaatetta ovat konseptina hyvin lähellä leanin valmistusperiaatteita.

2.2.1 Ajanhukan vähentäminen

Lean-filosofia pitää hukkana kaikkea, joka ei tuo lisäarvoa asiakkaalle:

1. Väärän tuotteen tai ominaisuuden tekeminen
2. Työkasauman väärin hallinta
3. Uudelleen tehtävä työ
4. Tarpeettoman monimutkaiset ratkaisut
5. Ylimääräisen ajattelun kuorma
6. Psykologinen stressi
7. Odottelu ja monet samanaikaiset tehtävät
8. Tiedonmenetykset
9. Tehoton viestintä.

Jotta ajanhukat voidaan poistaa, ne täytyy osata tunnistaa. Mikäli toiminta voidaan ohittaa tai tulos voidaan saavuttaa ilman sitä, silloin puhutaan hukasta:

- Osittain tehty koodaus, josta luovutaan kehitysprosessin aikana
- Lisäominaisuudet, joita asiakkaat eivät usein käytä
- Ihmisten vaihtaminen tehtävien välillä

- Muiden toimintojen, ryhmien ja prosessien odottaminen
- Työn suorittamiseen vaadittava uudelleenarviointi
- Viat ja huonompi laatu
- Johtohenkilöt, jotka eivät tuota todellista arvoa. (Poppendieck & Poppendieck 2003, 15-26.)

2.2.2 Lisää oppimista

Ohjelmistokehitys on jatkuva oppimisprosessi, joka perustuu iteraatioihin koodin kirjoittamisessa. Ohjelmistosuunnittelu on ongelmanratkaisuprosessi, jossa kehittäjät kirjoittavat koodin sekä sen, mitä ovat oppineet. Ohjelmiston arvo mitataan siitä, onko ohjelmisto käyttökelpoinen. (Poppendieck & Poppendieck 2003, 27-55.)

Vähennä dokumentointia tai yksityiskohtaista suunnittelua ja korvaa niitä kirjoittamalla toteutus. Käyttäjävaatimusten keräämisprosessia voidaan yksinkertaistaa esittämällä näkymät loppukäyttäjille ja pyytämällä heiltä palautetta. Vikojen kertyminen estetään suorittamalla koodin testaus heti, kun koodi on kirjoitettu. (Poppendieck & Poppendieck 2003, 27-55.)

Oppimisprosessia nopeutetaan käyttämällä lyhyitä iteraatiosyklejä yhdessä refaktoroinnin ja integraatiotestauksen kanssa. Lyhyet palauteistunnot asiakkaan kanssa voivat auttaa määrittämään nykyisen kehitysvaiheen ja sopeuttamaan ponnisteluja tulevia parannuksia varten. Näiden lyhyiden istuntojen aikana sekä asiakasedustajat että kehitysryhmä oppivat lisää pääongelmasta ja keksivät mahdollisia ratkaisuja jatkokehitystä varten. Näin myös asiakkaat ymmärtävät paremmin tarpeitaan olemassa olevan kehitystyön tulosten perusteella ja kehittäjät oppivat paremmin tyydyttämään tarpeet. (Poppendieck & Poppendieck 2003, 27-55.)

2.2.3 Viivästyä päätöksentekoa

Ohjelmistokehitykseen liittyy aina jonkin verran epävarmuutta. Parempia tuloksia saavutetaan joukko- tai vaihtoehtopohjaisella lähestymistavalla viivästyttämällä

päätöksiä, kunnes ne voidaan tehdä tosiasioiden perusteella eikä oletusten tai ennusteiden perusteella. Mitä monimutkaisempi järjestelmä on, sitä enemmän siihen tulisi rakentaa muutoskapasiteettia, joka mahdollistaa tärkeiden ja ratkaisevien sitoumusten viivästymisen. Toistuva lähestymistapa edistää tätä periaatetta eli kykyä mukautua muutokseen ja korjata virheitä, jotka saattavat olla erittäin kalliita, jos ne löydetään järjestelmän julkaisun jälkeen. (Poppendieck & Poppendieck 2003, 56-73.)

Ketterä lähestymistapa ohjelmistokehityksessä voi siirtää vaihtoehtojen rakentamista aikaisemmin asiakkaille ja lykätä siten tiettyjä tärkeitä päätöksiä, kunnes asiakkaat ovat ymmärtäneet tarpeet paremmin. Tämä mahdollistaa myös myöhemmän sopeutumisen muutokseen ja kalliimpien aikaisempien teknologiarajoitteisten päätösten estämisen. Tämä ei tarkoita, että suunnitteluun ei tulisi osallistua vaan päinvastoin. Suunnittelutoiminnassa tulisi keskittyä erilaisiin vaihtoehtoihin ja mukautua nykyiseen tilanteeseen, eikä selkeyttää hämmentäviä tilanteita luomalla nopean toiminnan malleja. Eri vaihtoehtojen arviointi on tehokasta heti, kun huomataan, että ne eivät ole ilmaisia, mutta tarjoavat tarvittavan joustavuuden myöhäiseen päätöksentekoon. (Poppendieck & Poppendieck 2003, 56-73.)

2.2.4 Toimita niin nopeasti kuin mahdollista

Mitä nopeammin lopputuote toimitetaan ilman suuria vikoja, sen nopeammin palaute voidaan vastaanottaa ja sisällyttää seuraavaan iteraatioon. Mitä lyhyempi iteraatioaika, sitä parempi on ryhmässä oppiminen ja viestintä. Asiakkaat arvostavat laadukkaan tuotteen nopeaa toimitusta. (Poppendieck & Poppendieck 2003, 74-96.)

Juuri ajoissa tapahtuvaa tuotantoideologiaa sovelletaan ohjelmistojen kehittämisessä tunnistamalla sen erityisvaatimukset ja ympäristö. Esittämällä saavutetaan tarvittava tulos ja antamalla tiimille aikaa organisoida itsensä ja jakaa tehtävät tarvittavan tuloksen suorittamiseksi tietylle iteraatiolle. Alussa asiakas antaa tarvittavat tiedot. Asiakkaan antama tieto voidaan esittää pieninä kortteina. Kehittäjät arvioivat kunkin kortin toteuttamiseen tarvittavan ajan. Näin työn organisointi muuttuu itseve-

toiseksi järjestelmäksi, kun joka aamu kokouksen aikana jokainen ryhmän jäsen tarkastelee eilen tehtyä työtä ja määrittää seuraavat työtehtävät. (Poppendieck & Poppendieck 2003, 74-96.)

2.2.5 Sitouta tiimi

Yrityksissä on ollut perinteinen usko organisaation päätöksentekoon. Johtajat kertovat työntekijöille, kuinka heidän tulee tehdä oma työnsä. "Work-Out-tekniikassa" roolit käännetään, jolloin esimiehet opetetaan kuuntelemaan kehittäjiä, jotta he selettävät paremmin toimenpiteet, joita voidaan toteuttaa ja antavat parannusehdotuksia. Kevyt lähestymistapa noudattaa ketterää periaatetta, jossa löydetään ammattitaitoisia työntekijöitä ja annetaan heidän tehdä omaa työtänsä. Samalla rohkaistaan kehittymään, tarttumaan virheisiin ja poistamaan esteitä. (Poppendieck & Poppendieck 2003, 97-121.)

Toinen virheellinen käsitys on ihmisten pitäminen resursseina. Ihmiset saattavat olla resursseja tilastollisen tietolomakkeen kannalta, mutta ohjelmistokehityksessä, samoin kuin kaikessa organisaatioliiketoiminnassa ihmiset tarvitsevat muutakin kuin vain tehtäväluettelon ja varmuuden siitä, ettei heitä häiritä valmistumisen aikana tehtävissään. Ryhmänjohtajan tulisi tarjota tukea ja apua vaikeissa tilanteissa sekä varmistaa, että skeptisyys ei pilaa ryhmän henkeä. Ihmisten kunnioittaminen ja heidän työnsä tunnustaminen on yksi tapa vahvistaa joukkuetta. (Poppendieck & Poppendieck 2003, 97-121.)

2.2.6 Vastaavuus siihen, mitä on markkinoitu ja toimitettu

Asiakkaalla on oltava yleinen näkemys järjestelmästä ja siitä, miten sitä:

- markkinoidaan
- toimitetaan
- otetaan käyttöön
- käytetään

- kuinka intuitiivinen sen käyttö on
- hinnoitellaan ja kuinka hyvin se ratkaisee ongelmat.

Käsitteenä eheys tarkoittaa, että järjestelmän erilliset komponentit toimivat yhdessä kokonaisuutena ja tasapainona joustavuuden, ylläpidettävyyden, tehokkuuden sekä reagoitavuuden välillä. Tämä voidaan saavuttaa ymmärtämällä ongelma-alueet ja ratkaisemalla ne samaan aikaan peräkkäisyyden sijaan. Tarvittava tieto on vastaanotettava pieninä visuaalisina erinä. Tietokulun tulisi olla jatkuvaa asiakkaan ja kehittäjän välillä. (Poppendieck & Poppendieck 2003, 122-145.)

Yksi terveellinen tapa kohti kiinteää arkkitehtuuria on refaktorointi. Mitä enemmän ominaisuuksia lisätään alkuperäiseen koodikantaan, sitä vaikeampi on lisätä uusia parannuksia. Koodin toistot ovat merkkejä virheellisestä koodisuunnittelusta ja niitä tulisi välttää. Täydelliseen ja automatisoituun rakennusprosessiin olisi liitettävä täydellinen ja automatisoitu kehittäjä- ja asiakastestauspaketti, jolla on sama versiointi, synkronointi sekä semantiikka kuin järjestelmän nykyisessä tilassa. Loppujen lopuksi eheys tulisi tarkistaa perusteellisella testauksella varmistaen siten, että järjestelmä tekee sen, mitä asiakas odottaa. Automaattisia testejä pidetään myös osana tuotantoprosessia ja jos ne eivät tuota lisäarvoa, silloin niitä pidetään hukkana. Automatisoidun testauksen ei pitäisi olla tavoite, vaan pikemminkin keino päämäärän saavuttamiseksi, joka vähentää vikoja. (Poppendieck & Poppendieck 2003, 122-145.)

2.2.7 Optimoi kokonaisuutta

Ohjelmistojen virheillä on taipumus kertyä kehitysprosessin aikana. Hajottamalla suuret tehtävät pienemmiksi tehtäviksi ja standardisoimalla kehitysvaiheet, jolloin virheiden perimmäiset syyt tulevat helpommin löydettäväksi ja poistettaviksi. Mitä suurempi järjestelmä, sitä enemmän organisaatioita, jotka ovat mukana sen kehittämisessä. Mitä useampia osia eri ryhmät kehittävät, sitä tärkeämpää on, että eri toimittajien välillä on hyvin määritellyt suhteet. Tällöin voidaan tuottaa järjestelmä,

jossa on vuorovaikutteiset komponentit. Pidemmällä kehitysjaksolla vahvempi alihankkijaverkosto on paljon hyödyllisempi kuin lyhytaikainen voiton optimointi, mikä ei mahdollista alihankkijaverkosto höytysuhteiden kehittymistä. (Poppendieck & Poppendieck 2003, 146-168.)

3 YRITYS X:N TARJOUSPROSESSIN NYKYTILA

3.1 Tarjouspyynnön käsittely

Asiakaskäynnillä myyjä määrittelee yhdessä asiakkaan kanssa makrotasolla, mitä heidän kehitysprojektinsa sisältää. Samalla määritellään karkea aikataulu. Tämän jälkeen myyjä kysyy karkean arvion projektin kustannuksista projektiryhmän ohjelmistoarkkitehdiltä ja -suunnittelijalta. Tämän jälkeen myyjä neuvottelee projektin hinta-arviota asiakkaan kanssa. Samalla sovitaan laskutustapa riippuen projektin toteutustavasta. Tämä vaihtelee asiakkaasta sekä projektin luonteesta.

3.2 Ongelmia

Nykyisessä tarjouslaskennassa jää useasti huomiotta teknologian riskit, jotka kasvattavat ajanhukkaa. Esimerkkinä tällaisesta tilanteesta on RFID-siru, jossa on teknisiä vajavuuksia, mitkä ilmenevät vasta toteutuksen testauksessa. Dokumentaatioon kuluva aika jää myös usein huomiotta ja eikä sitä huomioida suuresti nykyisessä tarjousprosessissa tai se lisätään ei laskutettaviin tunteihin. Myös muita projektin työmäärää kasvattavia tekijöitä jätetään huomioimatta tai niiden vaikutusta ei osata tiedostaa työmäärää kasvattavana tekijänä. Edellä mainitut seikat kasvattavat jo arvioitua työmäärää ja tämä aiheuttaa aikataulullisia ja työmäärällisiä ongelmia sekä psykologista stressiä.

3.3 Vaatimukset toteutukselle

Yritys X:ssä käytetään ohjelmistokehityksessä lean-ajattelua, jota toteutetaan pienessä asiantuntijatiimissä. Tämän vuoksi tulee ottaa huomioon, että ylimääräistä aikaa ja resurssia on hyvin rajoitetusti. Mahdollisesti syntyviä hukkia pyritään estämään ja vähentämään lean-ajattelulla. Samalla halutaan läpinäkyvyyttä tarjousprosessiin, jotta asiakas on tietoinen mahdollisista teknologian haasteista. Toteutuksen tulisi sisältyä normaaliin työprosessiin, jotta se olisi helposti ajan samalla helposti ajan tasalla pidettävä.

4 TUTKIMUSMENETELMÄT

4.1 Tutkimusmenetelmän rakenne

Toteutus jakaantuu kolmeen osioon. Ensimmäisessä osiossa käydään läpi projektin suunnittelupohja. Sen jälkeen tarkastellaan suunnittelupohjan soveltuvuutta COSYSMO hinta-arvioon. Lopuksi pohja päivitetään niin, että se soveltuu mallin työmäärän arvioon.

Toinen osa on COSYSMO-laskentamallin käyttö. Tässä käytetään valmista toteutusta.

Kolmannessa osassa luodaan koko prosessista työohje. Työohjeessa kerrotaan, kuinka suunnittelupohjaan täytetään oikein, jotta saadaan luvut COSYSMO-laskentaan.

4.2 Projektin suunnittelupohja

Projektin suunnittelupohjassa on seitsemän kappaletta. Ensimmäinen kappale on nimetty Yleistä. Tässä kappaleessa annetaan yleinen kuvaus projektin luonteesta sekä tavoitteesta.

Toinen kappale on nimeltään Järjestelmän vaatimukset. Tässä kappaleessa koostaan parametrit ja arvot, jotka tulee täyttää taulukkoon. Taulukon toiseen osaan kerätään standardit, joita tulee noudattaa ja täyttää. Jokaisesta standardista annetaan lyhyt selitys, mitä se pitää sisällään.

Kolmas kappale on nimeltään Suunnittelu. Tässä kappaleessa suunnitteluosat jakaantuvat eri suunnittelualan kappaleisiin, käytettäviin ohjelmistoihin, työkaluihin ja henkilöresursseihin. Tämä tarkoittaa, että pääsuunnitteluosat ovat ohjelmistosuunnittelu, elektroniikkasuunnittelu ja mekaniikkasuunnittelu. Näiden alle jakaantuvat suunniteltavat osat omiin alaotsikkoihin. Näiden otsikoiden alle jakaantuvat suunniteltavan osan ominaisuudet omiin alaotsikkoihin. Tämän jälkeen käytettävät ohjelmistot ja työkalut on eritelty taulukkoon, jossa on sarake ohjelmalle tai työkalulle ja

sen jälkeen sarake selitykselle, joka kertoo sen tehtävän. Tämän jälkeen luetellaan käytettävät resurssit, jossa on eritelty projektiryhmän henkilöt ja heidän tehtävänsä projektissa.

Neljäs kappale on nimeltään Kustannusarvio. Tämä kappale sisältää suunnittelun tunti-arvion ja materiaalikustannukset. Suunnittelun tunti-arviossa on eriteltyä osaluokkien kustannukset. Materiaalikustannuksissa on eriteltyä materiaalit ja toimitettavat osat.

Viides kappale on nimeltään Laskutus, jossa määritellään laskutusehdot ja lisäkustannukset. Kuudes kappale käsittelee projektin tekijänoikeudet ja seitsemännessä kappaleessa on tarjouksen yleiset ehdot.

4.3 COSYSMO-laskentamallin toteutus

COSYSMO-laskentamallissa kokomuuttujat kootaan omien luokkien ja vaikeustason alle. Taulukossa 6 on esitetty järjestelmän kokomuuttujat ja kuinka niiden keräys on toteutettu Excel-malliin.

Taulukko 6. Järjestelmän kokomuuttujat yksinkertaiseen COSYSMO-laskentaan.

	<i>Easy</i>	<i>Nominal</i>	<i>Difficult</i>
# of System Requirements			
# of System Interfaces			
# of Algorithms			
# of Operational Scenarios			

Taulukossa 7 on COSYSMO-mallin hintamuuttujat. Valitsemalla keltainen sarake aukeaa valikko, josta valitaan hintamuuttujan taso. Taso vaikuttaa oikeanpuoleisen sarakkeen hintakertoimeen, joka kasvaa tai pienenee riippuen valitusta tasosta.

Taulukko 7. COSYSMO-taulukon hintaparametrien valinta.

SELECT COST PARAMETERS FOR SYSTEM OF INTEREST		
Requirements Understanding	N	1,00
Architecture Understanding	N	1,00
Level of Service Requirements	N	1,00
Migration Complexity	N	1,00
Technology Risk	N	1,00
Documentation	N	1,00
# and diversity of installations/platforms	N	1,00
# of recursive levels in the design	N	1,00
Stakeholder team cohesion	N	1,00
Personnel/team capability	N	1,00
Personnel experience/continuity	N	1,00
Process capability	N	1,00
Multisite coordination	N	1,00
Tool support	N	1,00
		1,00 composite effort multiplier

SYSTEMS ENGINEERING PERSON MONTHS

4.4 Työohje

Työohjeen ensimmäinen osuus käsittelee laskentaohjeita. Valittaessa kustannusparametrejä, olisi hyvä laskea osissa henkilötyökuukaudet, jotka eroavat muusta kokonaisuudesta huomattavasti. Näin saadaan tarkempi arvio, kun käytetään COSYSMO-mallia. Laskentaa varten osiot olisi hyvä numeroida otsikkoihin erikseen, jotta voidaan sisällysluetteloa lukemalla määrittää arvot COSYSMO-Exceeliin sekä helpottaa ja nopeuttaa arvion määrittelyä.

Työohjeen toisessa osuudessa käsitellään, kuinka määritellään kokomuuttujat ja miten ne sisällytetään sisällysluetteloon. Nimikettä vaikeustaso käytetään hinta-arvion laskennassa ja määritellään työmäärää. SR, SI, Alg ja OS Lyhenteillä kerrotaan, mikä määrittely on kyseessä. Kuviossa 1 on esitetty testaukseen käytetyn ohjeen sisällysluettelo.

Sisällys

1.	YLEISTÄ	3
2.	JÄRJESTELMÄN VAATIMUKSET	3
3.	SUUNNITTELU	4
3.1	Kuvaus tuotteen ohjelmisto osista	4
3.1.1	SQL tietokanta(Easy/Nominal/Difficult)(SR/SI/Alg/OS)	4
3.2	Kuvaus tuotteen elektroniikka osista	4
3.2.1	USB-C yhteys (Easy/Nominal/Difficult) (SR/SI/Alg/OS)	4
3.3	Kuvaus tuotteen mekaniikka osista	4
3.3.1	Kotelo (Easy/Nominal/Difficult) (SR/SI/Alg/OS).....	4
3.4	Käytettävät ohjelmistot ja työkalut	5
3.4.1	Ohjelmistot	5
3.4.2	Työkalut.....	5
3.5	Käytettävät resurssit	5
4.	KUSTANNUSARVIO	5
4.1.	Suunnittelu	5
4.2.	Prototyypit (Arvio)	6
5.	LASKUTUS	7
6.	TEKIJÄNOIKEIDET	7
7.	TARJOUKSEN YLEISET EHDOT.....	7

Kuvio 1. Projektisuunnitelman sisällysluettelo.

Työohjeen toinen osio toteutetaan parhaiten kuvion 2 mukaisesti. Tällä tavalla saadaan yksi dokumentti, joka sisältää asiaan kuuluvan tiedon. Samalla ohjeita on helppompaa ylläpitää. Dokumentin käyttöohjeet ovat löydettävissä helposti.

3. SUUNNITTELU

Mitä suunnitellaan?

Huom. Valittaessa kustannus parametrejä olisi hyvä laskea erikseen henkilötyökuukaudet osissa, jotka eroavat muusta kokonaisuudesta huomattavasti. Näin saadaan tarkempi arvio, kun käytetään COSYSMO-mallia. Nämä osiot olisi hyvä numeroida otsikkoihin erikseen laskentaa varten. Näin voidaan sisällysluetteloa lukemalla määrittää arvot COSYSMO-Exceeliin ja helpottaa sekä nopeuttaa arvon määrittelyä.

3.1 Kuvaus tuotteen ohjelmisto osista

Tuotteeseen tulevat ominaisuudet seuraavasti:

3.1.1 SQL tietokanta(Easy/Nominal/Difficult)(SR/SI/Alg/OS)

Otsikossa oleva vaikeustaso poistetaan asiakkaalle lähtevästä versiosta, mutta pidetään sisäisessä versiossa. Vaikeustaso nimikettä käytetään hinta-arvon laskennassa ja kun määritellään työmäärää. SR, SI, Alg- ja OS-lyhenteillä määritellään, onko kyseessä jokin seuraavista määrittelyssä:

SR = System Requirements = Käyttäjätason tehtävä

SI = System Interfaces = Vain tekniset rajapinnat. Kaksisuuntaiset rajapinnat lasketaan kahdeksi rajapinnaksi, koska ne vaativat koordinoinnin molemmissa päissä.

Alg = Algorithms = Ohjelmistojen kehittämisspennistelujen arvioimiseksi tulisi käyttää malleja, kuten COCOMO II. Jos Algoritmi tehdään nolasta, niin tulisi käyttää COCOMO II laskentaa tai luokitella se vähintään vaikeaksi. Hyvin tunnettu algoritmi ennustettavissa olevalla käytöllä, se katsotaan "helpoksi" algoritmiksi.

OS = Operational Scenarios = Operatiiviset skenaariot saadaan usein testitapausten tai järjestelmän käyttötapausten kautta, koska ne edustavat järjestelmän päästä päähän -toimintoa tai järjestelmän itsenäisiä ominaisuuksia. Esimerkiksi Windows XP -tietokoneen käyttötapa on toimia "vikasietotilassa".

Kuvio 2. Työohjeen integrointi suunnittelupohjaan.

5 TUTKIMUSTULOKSET

5.1 Valmistuneen projektin aikavertailu

Ensimmäisellä työohjeella saatiin 126,5 henkilötyökuukauden arvio valmistuneelle projektille. Valmistuneessa projektissa oli kehittäjiä kolmessa eri toimipisteessä. Kyseinen projekti sisältää neljä erilaista elektroniikkakorttia, joista kolmeen elektroniikkakorttiin on kehitetty ohjelmisto.

Ensimmäisellä työohjeella saatiin projektin kokomuuttujiksi määriksi seuraavat arvot. Projekti sisältää 50 helppoa, 20 normaalia ja 10 vaikeaa järjestelmävaatimusta. Järjestelmän rajapintoja projektissa on kaksi helppoa, yksi normaali ja kaksi vaikeaa. Algoritmeja on kolme normaali tasoista. Toiminnallisia tilanteita on viisi kappaletta.

Taulukko 8. Valmistuneen projektin hintamuuttujat.

SELECT COST PARAMETERS FOR SYSTEM OF INTEREST

Requirements Understanding	N	1,00
Architecture Understanding	L	1,27
Level of Service Requirements	N	1,00
Migration Complexity	N	1,00
Technology Risk	H	1,32
Documentation	L	0,91
# and diversity of installations/platforms	N	1,00
# of recursive levels in the design	VL	0,80
Stakeholder team cohesion	N	1,00
Personnel/team capability	L	1,22
Personnel experience/continuity	L	1,21
Process capability	L	1,21
Multisite coordination	N	1,00
Tool support	H	0,85
		1,85

composite effort multiplier

COSYSMO-laskennalla taulukon 8 hintamuuttujilla saatiin 126,5 työkuukautta projektin kokonaismääräksi. Projektiin kului karkeasti laskettuna noin 50-65 työkuukautta. Tämä on yhteenlaskettu määrä seitsemän eri henkilön projektiin käyttämistä työkuukausista.

Tarkentamalla ohjeistusta saatiin neljä aikaisemmin toiminnallisiin tilanteisiin kuulunutta tilannetta siirtymään neljäksi vaikeaksi järjestelmävaatimukseksi. Toiminnalliset tilanteet on ymmärretty väärin ja nämä neljä on lisätty kyseiseen kategoriaan

sen vuoksi. Tällöin alkuperäisillä hintamuuttujilla työkuukausi määrä putoaa 101,1 työkuukauteen.

Hintamuuttujissa otetaan huomioon aiemmin huomiotta jääneet palveluvaatimukset, sidosryhmien yhteenkuuluvuus ja projektikoordinaatio. Nämä on jätetty normaaliin kategoriaan, koska niihin ei ole osattu vastata. Uudelleen tarkasteltaessa saatiin palveluvaatimukset matalaksi, sidosryhmien yhteenkuuluvuus erittäin korkeaksi ja projektikoordinaatio korkeaksi. Tämä laski uudelleen määritellyn projektin työkuukausimäärän 47,1 työkuukauteen.

Kun hintamuuttujat on edellä mainitusti päivitetty vastaamaan koko projektin hintamuuttujia, saadaan alkuperäisen kokoarvion työmääräksi 58,9 työkuukautta, kun päivitetyn projektin kokoarvio oli 47,1 työkuukautta. Täten arvioiden ero on 11,8 työkuukautta. Tämä tuo vertaillessa esiin hintamuuttujien ymmärtämisen ja niiden painoarvon COSYSMO-laskennassa.

Ottamalla toteutuneen työmäärän keskiarvon 57,5 ja laskennallisen työmäärän keskiarvon 53, saadaan 8,5 prosentin ero tuloksissa. Tämä on hyväksyttävän lähelle, koska kyseessä on arvio työmäärästä tarjousvaiheessa.

5.2 Keskeneneräisen projektin aikavertailu

Ensimmäisellä työohjeella suunnittelupohjan testiversiossa saadaan 19,3 henkilötyökuukautta keskeneneräiselle projektille. Keskeneneräisessä projektissa kehittäjät ovat samassa toimistossa. Kyseinen projekti sisältää neljä erilaista elektroniikkakorttia, joista kahteen elektroniikkakorttiin kehitetään ohjelmisto.

Ensimmäisellä COSYSMO-laskennalla saadaan projektin kokomuuttuja määräksi seuraavat arvot. Projekti sisältää kolme helppoa, viisi normaalia ja kaksi vaikeaa järjestelmävaatimusta. Järjestelmän rajapintoja projektissa on kaksi helppoa, yksi normaali ja kolme vaikeaa. Algoritmeja on viisi normaali tasoista. Toiminnallisia tilanteita on viisi kappaletta.

Taulukko 9. Keskeneräisen projektin hintamuuttujat.

SELECT COST PARAMETERS FOR SYSTEM OF INTEREST

Requirements Understanding	H	0,77
Architecture Understanding	H	0,81
Level of Service Requirements	N	1,00
Migration Complexity	N	1,00
Technology Risk	N	1,00
Documentation	L	0,91
# and diversity of installations/platforms	N	1,00
# of recursive levels in the design	L	0,89
Stakeholder team cohesion	N	1,00
Personnel/team capability	N	1,00
Personnel experience/continuity	H	0,82
Process capability	L	1,21
Multisite coordination	N	1,00
Tool support	H	0,85
		0,43

composite effort multiplier

Taulukon 9 hintamuuttujilla saadaan 19,3 työkuukautta projektin kokonaismääräksi COSYSMO-laskennalla. Projektiin on kulunut jo karkeasti laskettuna noin kuudesta kahdeksaan työkuukautta. Tämä on yhteen laskettu määrä kahden eri henkilön projektiin käyttämistä työkuukausista.

Tarkentamalla ohjeistusta saadaan neljä aikaisemmin toiminnallisiin tilanteisiin kuulunutta tilannetta siirtymään neljäksi vaikeaksi järjestelmävaatimukseksi sekä neljä algoritmia vaikeaksi järjestelmävaatimuksiksi. Toiminnalliset tilanteet on ymmärretty väärin ja nämä neljä on lisätty kyseiseen kategoriaan sen vuoksi. Projektin koko-
muuttujat ovat kolme helppoa, viisi normaalia ja kymmenen vaikeaa järjestelmävaatimusta. Järjestelmän rajapintoja projektissa on kaksi helppoa, yksi normaali ja kolme vaikeaa. Algoritmeja on yksi normaali tasoinen. Toiminnallisia tilanteita on yksi kappale. Tällöin alkuperäisillä hintamuuttujilla työkuukausi määrä putoaa 14,1 työkuukauteen.

Hintamuuttujissa otetaan huomioon aiemmin huomiotta jääneet palveluvaatimukset, sidosryhmien yhteenkuuluvuus ja projektikoordinaatio. Nämä on jätetty normaali kategoriaan, koska niihin ei ole osattu vastata. Uudelleen tarkasteltaessa saatiin palveluvaatimukset matalaksi, sidosryhmien yhteenkuuluvuus erittäin korkeaksi ja projektikoordinaatio korkeaksi. Tämä laskee uudelleen määritellyn projektin työkuukausi määrän 7,3 työkuukauteen.

Kun hintamuuttujat on edellä mainitusti päivitetty vastaamaan koko projektin hintamuuttujia, saadaan alkuperäisen kokoarvion työmääräksi yhdeksän työkuukautta.

Myös päivitetyn projektin kokoarvio pieneni 6,6 työkuukauteen. Täten arvioiden ero on 3,6 työkuukautta.

Kun huomioidaan projektin nykyinen valmiusvaihe. Voidaan todeta lasketun työmäärän olevan lähellä oikeaa.

5.3 Työohjeen parantaminen

Työohjeen vaatimusosiota parannetaan lisäämällä siihen syvempi selitys, mitä järjestelmävaatimuksella tarkoitetaan. Samalla selitys laajentuu, koska järjestelmävaatimuksia sekoitetaan operatiivisiin skenaarioihin ja algoritmeihin. Tämä sekaannus kasvattaa työmäärä arvioinnissa rajusti.

Samalla työohjeesta poistetaan COSYSMO-kokomuuttujien piilottaminen. Tällä pyritään tuomaan jo tarjousvaiheessa lean-ajattelua esiin. Saadaan prosessiin läpinäkyvä, jolla estetään turhia ominaisuuksia tai liian monimutkaisia toteutuksia. Näin tehotonta viestintää vähennetään jo prosessin alusta alkaen. Tällä pyritään helpottamaan kehityksen aikana syntyviä ongelmatilanteiden kommunikointia, joista syntyy lisätöitä, viivästyksiä, toimituksen osien toimitusestymisiä tai uudelleen suunnittelua. Kuviossa 3 on palautteen perusteella päivitetty työohje.

3. SUUNNITTELU

Mitä suunnitellaan?

Huom. Valittaessa kustannus parametrejä olisi hyvä laskea erikseen henkilötyökuukaudet osissa, jotka eroavat muusta kokonaisuudesta huomattavasti. Näin saadaan tarkempi arvio, kun käytetään COSYSMO-mallia. Nämä osiot olisi hyvä numeroida otsikkoihin erikseen laskentaa varten. Näin voidaan sisällysluetteloa lukemalla määrittää arvot COSYSMO-Exceliin ja helpottaa sekä nopeuttaa arvion määrittelyä.

3.1 Kuvaus tuotteen ohjelmisto osista

Tuotteeseen tulevat ominaisuudet seuraavasti:

3.1.1 SQL tietokanta(Easy/Nominal/Difficult)(SR/SI/Alg/OS)

Vaikeustaso nimikettä käytetään hinta-arvion laskennassa ja kun määritellään työmäärää. SR, SI, Alg- ja OS-lyhenteillä määritellään, onko kyseessä jokin seuraavista määrittelyssä:

SR = Vaatimukset voivat olla luonteeltaan toiminnallisia, suorituskykyisiä, ominaisuus- tai palvelukeskeisiä erittelyyn käytetyn menetelmän mukaan. Ne voi myös määrittää asiakas tai urakoitsija. Jokaiseen vaatimukseen voi liittyä vaivaa, kuten todentaminen ja validointi, toiminnallinen hajoaminen, toiminnallista allokointia. Järjestelmävaatimukset voidaan tyypillisesti laskemalla järjestelmässä tai markkinointieritelmässä sovellettavien vaatimusten lukumääränä.

SI = Vain tekniset rajapinnat. Kaksisuuntaiset rajapinnat lasketaan kahdeksi rajapinnaksi, koska ne vaativat koordinoinnin molemmissa päissä.

Alg = Ohjelmistojen kehittämisspennistelujen arvioimiseksi tulisi käyttää malleja, kuten COCOMO II. Jos Algoritmi tehdään nollasta, niin tulisi käyttää COCOMO II laskentaa tai luokitella se vähintään vaikeaksi. Hyvin tunnettu algoritmi ennustettavissa olevalla käytöllä, se katsotaan "helpoksi" algoritmiksi.

OS = Operatiiviset skenaariot saadaan usein testitapausten tai järjestelmän käyttötapausten kautta, koska ne edustavat järjestelmän päästä päähän -toimintoa tai järjestelmän itsenäisiä ominaisuuksia. Esimerkiksi Windows XP -tietokoneen käyttötapa on toimia "vikasietotilassa".

Kuvio 3. Päivitetty työohje.

6 JOHTOPÄÄTÖKSET JA SUOSITUKSET

Täyttämällä suunnittelupohjan ohjeiden mukaan, sisällysluetteloon syntyi lista COSYSMO-kokomuuttujista. Tästä oli helppo kerätä kokomuuttajat COSYSMO-laskenta Exceliin. Samalla saadaan esiin selkeä rakenne asiakasprojektin kokonaisuuksista ja niiden sisältämistä ominaisuuksista sekä vaatimuksista. Samalla käy ilmi myös eri kokonaisuuksien vaativuus.

COSYSMO-hintamuuttajat vaativat omat perehtymisen, mitä ei lähdetty lisäämään suunnittelupohjan ohjeistukseen. Tähän on syynä, että laskentamalli vaati ylimääräistä perehtymistä ja ymmärtämisen, jota ei pystytty lyhyeen ohjeeseen sisällyttämään. Tässä nähtiin parhaaksi, että käyttäjä perehtyy COSYSMO-verkkosivulta löytyvään ohjeeseen.

Testauksesta saatuja tuloksia tarkasteltaessa saatiin seuraavia johtopäätöksiä. Ilman syvempää perehtymistä saadaan normaalia suurempia työmääriä aikaan, kuin olisi odotettu. Kokomuuttujien suhteen on oltava tarkkana, mihin ryhmään ja vaikeusluokkaan muuttuja kuuluu. Samalla on oltava tarkkana hintamuuttujien kanssa ja luokiteltava ne projektissa arvioitavissa olevalle tasolle.

Testauksessa kävi ilmi, että hintamuuttujien väärin arvioiminen vääristää hyvin paljon koko projektin arvioitua työkuukausi määrää. Tästä voidaan todeta, että subjektiivisessa arvioinnissa on erittäin tärkeää rehellinen ja todellinen tuntemus hintamuuttujien luokasta, koska muuten syntyy hyvin suuria vääristymiä arvioon.

Suosittelavaa on, että suunnittelupohja otetaan tästä eteenpäin käytettäväksi. Ohjeen mukaan täytetystä pohjasta ilmenee asiakasprojektin eri osien vaativuus ja luonne. Tämä tuo esiin jo tarjousvaiheessa mahdollisesti turhat, liian työläät tai monimutkaiset ominaisuudet, joista mahdollisesti luovutaan projektin aikana.

Ennen COSYSMO-mallin käyttöönottoa on suositeltavaa, että käyttäjä perehtyy mallin hinta- ja kokomuuttujien luokitteluun. Perehtyminen estää ja vähentää väärin tulosten syntymistä. Väärin tulkitut ja arvioidut muuttajat aiheuttavat isoja eroja tuloksissa.

7 YHTEENVETO

Työ toteutettiin alle viiden hengen insinööritoimistossa, jossa haluttiin kehittää tarjouslaadintaprosessia. Nykyisessä prosessissa tarjoukset uusista asiakasprojekteista laskee yksi asiantuntija, joka pyytää tarkentavat lisäarviot projektin eri osien tekijöiltä. Työmäärä arvioidaan kokemuksen perusteella, johon lisätään riskikertoimet. Nykyinen menetelmä ei huomioi piilotyömääriä ja tämä aiheuttaa sekä aikataulullisia että taloudellisia ongelmia insinööritoimisto X:lle. Insinööritoimisto X:n asiakasprojektit jakaantuvat tai koostuvat pääosin kahdesta osiosta, joita ovat elektroniikka ja ohjelmistokehitys.

Työn tavoitteena oli tutkia COSYSMO-mallia ja mallin soveltuvuutta insinööritoimisto X:n asiakasprojektien hinta-arvion määrittelyyn käyttäen ohjelmistokehitykseen suunnattua lean-ajattelua. Insinööritoimisto X:n pääasiallinen toimiala oli IT-konsultointi, jossa yritys oli erikoistunut sulautettujen järjestelmien kehittämiseen. Tästä syystä projektimenetelmän ja kustannusmallin oli oltava ohjelmistokehitykseen ja IT-projektiin soveltuvia. Mallia pyrittiin soveltamaan myös elektroniikka- ja mekaniikkasuunnittelun hinta-arvion määrittelyyn.

Tutkimuksen teoriaosuudessa selvitettiin ensin, mistä COSYSMO-malli rakentuu ja kuinka sen koko- ja hintamuuttujat määritellään. Tämän jälkeen perehdyttiin ohjelmistokehityksen lean-ajatteluun, jolla pyrittiin poistamaan toteutuksesta hukkatyötä. Käytännöntyön kannalta nämä oli tärkeää opiskella ennen työn aloittamista, jotta niitä pystyttiin käyttämään työtä tehdessä.

Tutkimuksen toteutus käymällä läpi nykytilanne ja nykyiset suunnittelupohjat. COSYSMO-mallista oli valmis Excel-malli. Tämän jälkeen alettiin tarkastelemaan, miten COSYSMO-mallin muuttujat ja määrittely voitaisiin toteuttaa normaalin suunnittelutyön ohessa. Otsikoihin liitettiin sulkuihin kokomuuttuja ja muuttujan vaikeustaso. Työohje nähtiin parhaaksi toteuttaa suunnittelupohjan yhteyteen, jolloin ohje on helposti löydettävissä ja ylläpidettävissä.

Työohje jakautui kahteen osaan, jossa ensimmäisessä osuudessa käsiteltiin kokomuuttujien laskentaohjeita. Valittaessa kustannusparametrejä, olisi hyvä laskea osissa henkilötöyökuukaudet, jotka eroavat huomattavasti kokonaisuudesta. Käytettäessä COSYSMO-mallia saadaan tarkempi arvio. Nämä osiot olisi hyvä numeroida otsikoihin erikseen laskentaa varten. Näin voidaan sisällysluetteloita lukemalla määrittää arvot

COSYSMO-Exceeliin ja helpottaa sekä nopeuttaa arvion määrittelyä. Työohjeen toisessa osuudessa käsiteltiin, kuinka määritellään kokomuuttajat ja kuinka ne sisällytetään sisällysluetteloon. Nimikettä vaikeustaso käytetään hinta-arvion laskennassa ja määriteltäessä työmäärää.

Toteutusta testattiin kahdella eri projektilla, joista toinen oli valmis ja toinen keskeneräinen. Näin haluttiin verrata ja etsiä huomioita toteutuksen toimivuudesta. Ensimmäisellä työohjeella suunnittelupohjan testiversiossa saatiin 126,5 henkilötyökuukautta valmistuneelle projektille. Tämä oli huomattavasti enemmän, kuin oli toteutunut. Tarkastellessa huomattiin, että ohjeistusta oli tarpeen tarkentaa. Tarkentamalla ohjeistusta saatiin työkuukaudet putoamaan 101,1 työkuukauteen. Hintamuuttujissa otettiin huomioon aiemmin huomiotta jääneet palveluvaatimukset, sidosryhmien yhteenkuuluvuus ja projektikoordinaatio. Tämä laski projektin työkuukausi määrän 47,1 työkuukauteen. Vertailemalla toteutuneen työmäärän keskiarvon ja laskennallisen työmäärän keskiarvoa, tuloksissa saatiin kahdeksan ja puolen prosentin ero tuloksissa.

Ensimmäisellä työohjeella suunnittelupohjan testiversiossa saatiin 19,3 henkilötyökuukautta keskeneräiselle projektille. Tarkastellessa huomattiin, että ohjeistusta oli tarpeen tarkentaa. Tarkentamalla ohjeistusta saatiin työkuukaudet putoamaan 14,1 työkuukauteen. Hintamuuttujissa otettiin huomioon aiemmin huomiotta jääneet palveluvaatimukset, sidosryhmien yhteenkuuluvuus ja projektikoordinaatio. Tämä laski projektin työkuukausi määrän yhdeksään työkuukauteen ja päivitetyn projektiarvion 6,6 työkuukauteen. Kun huomioidaan projektin nykyinen valmiusvaihe. Voidaan todeta lasketun työmäärän olevan lähellä oikeaa.

Työohjeen vaatimusosiota parannettiin lisäämällä siihen syvempi selitys, mitä järjestelmä vaatimuksella tarkoitetaan. Tämä sekaannus kasvatti työmäärää. Samalla työohjeesta poistettiin COSYSMO-kokomuuttajien piilottaminen. Tällä pyritään tuomaan jo tarjousvaiheessa lean-ajattelua esiin. Saadaan prosessiin läpinäkyvä, jolla estetään turhia ominaisuuksia tai liian monimutkaisia toteutuksia. Näin tehotonta viestintää vähennetään jo prosessin alusta alkaen. Näin yritettiin helpottaa kehityksen aikana syntyviä ongelmatilanteiden kommunikointia, joista syntyy lisätöitä, viivästyksiä, toimituksen osien toimitusestymisiä tai uudelleen suunnittelua.

Johtopäätöksiksi saatiin tuloksista, että kokomuuttajien suhteen on oltava tarkkana mihiin ryhmään ja vaikeusluokkaan muuttuja kuuluu. Hintamuuttajien väärin arvioiminen vääristää hyvin paljon koko projektin arvioitua työmäärää. Subjektiviisessa arvioinnissa

on erittäin tärkeää rehellinen ja todellinen tuntemus hintamuuttujien luokasta, koska muuten syntyy arvioon hyvin suuria vääristymiä.

Suosittelavaa on, että suunnittelupohja otetaan tästä eteenpäin käytettäväksi. Ohjeen mukaan täytetystä pohjasta ilmenee asiakasprojektin eri osien vaativuus ja luonne. Tämä tuo esiin jo tarjousvaiheessa mahdollisesti turhat, liian työläät tai monimutkaiset ominaisuudet, joista mahdollisesti luovutaan projektin aikana. Ennen COSYSMO-mallin käyttöönottoa on suositeltavaa, että käyttäjä perehtyy mallin hinta- ja kokomuuttujien luokitteluun. Perehtyminen estää ja vähentää väärin tulosten syntymistä. Väärin tulkitut ja arvioidut muuttujat aiheuttavat isoja eroja tuloksissa.

LÄHTEET

Constructive Systems Engineering Cost Model(COSYSMO). 2020. [Verkkosivu]. Massachusetts Institute of Technology. [Viitattu 26.1.2020]. Saatavana: <http://cosysmo.mit.edu/>

Jared Fortune. 2010. COSYSMO 2.0: Constructive Systems Engineering Cost Model. ISBN-10: 3639286162

Lean. 2020. [Verkkosivu]. Turun yliopisto. [Viitattu 26.1.2020]. Saatavana: https://tech.utu.fi/embedded_kasikirja/index.html

Poppendieck, M. & Poppendieck, T. 2003. Lean Software Development: An Agile Toolkit. Boston: Addison–Wesley.

Valerdi, Ricardo. 29.1.2006. Academic COSYSMO. [Käyttöohje]. [Viitattu 26.1.2020]. Saatava: <http://cosysmo.mit.edu/wp-content/uploads/2010/02/academic-COSYSMO-User-Manual-v1.1.doc>