

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

Gytenis Mikulėnas

**INFORMACINĖS SISTEMOS PROGRAMINĖS
ĮRANGOS KŪRIMO DARBŲ APIMTIES
ĮVERTINIMO MODELIS**

Magistro darbas

**Vadovas
doc.dr. R. Butleris**

KAUNAS, 2006

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

**INFORMACINĖS SISTEMOS PROGRAMINĖS
ĮRANGOS KŪRIMO DARBŲ APIMTIES
ĮVERTINIMO MODELIS**

Magistro darbas

Vadovas

doc. dr. R. Butleris

2006-01-09

Recenzentas

dr. Stasys Maciulevičius

2005-12-20

Atliko

IFM 0/4 gr. stud.

G. Mikulėnas

KAUNAS, 2006

INFORMATION SYSTEMS SOFTWARE DEVELOPMENT AMOUNT ESTIMATION MODEL

SUMMARY

The main goal of this work is to analyze existing popular software size estimation and project completeness schedule methods and models, find advantages and disadvantages, try to combine them for using together, searching for common properties. The combination of these methods and models will give a new model, which will be tested with experimental research.

TURINYS

1. ĮVADAS.....	8
2. INFORMACINĖS SISTEMOS PROGRAMINĖS ĮRANGOS KŪRIMO DARBŲ APIMTIES BEI TRUKMĖS ĮVERTINIMO METODŲ ANALIZĖ.....	10
2.1. Analizės tikslas.....	10
2.2. Tyrimo sritis, objektas ir problema.....	10
2.3. Vartotojų analizė.....	11
2.3.1. Vartotojų aibė, tipai ir savybės.....	11
2.3.2. Vartotojų tikslai ir problemos.....	11
2.4. Projekto valdymo proceso analizė.....	12
2.5. Programinės įrangos kūrimo darbų apimties įvertinimo metodų analizė.....	13
2.5.1. Projekto apimties / dydžio skaičiavimo metodai.....	13
2.5.1.1. Kodo eilučių skaičiavimo metodas.....	13
2.5.1.2. Funkcinių taškų metodas.....	15
2.5.1.3. Objektinių taškų metodas.....	20
2.5.2. Projekto trukmės - laikinių įverčių skaičiavimo metodai.....	21
2.5.2.1. Analoginis vertinimo metodas.....	21
2.5.2.2. Ekspertinis vertinimo metodas.....	21
2.5.2.3. Basic COCOMO modelis.....	23
2.5.2.4. COCOMO II : Aplikacinis komponavimo modelis.....	24
2.5.2.5. COCOMO II : Ankstyvo projektavimo modelis.....	25
2.5.2.6. COCOMO II : Vėlyvo projektavimo modelis.....	27
2.5.2.7. PUTNAM modelis.....	29
2.5.2.8. Sistemine analize orientuotas metodas.....	30
2.5.3. Laikinių įverčių skaičiavimo metodų palyginimas.....	32
2.6. Analizės išvados.....	36
3. INFORMACINĖS SISTEMOS PROGRAMINĖS ĮRANGOS KŪRIMO DARBŲ APIMTIES ĮVERTINIMO MODELIS.....	38
3.1. VIVEMO-7 modelio pagrindimas ir esmės išdėstymas.....	38
3.2. VIVEMO-7 modeliui reikalingų duomenų sudėtis.....	39
3.3. Darbo su VIVEMO-7 modeliu procesas.....	40
3.4. VIVEMO-7 modelio taikymo etapai.....	41
3.4.1. Projekto reikalavimų ir analizės etapų įvertinimas.....	41
3.4.2. Projektavimo etapo įvertinimas.....	45
3.4.3. Projekto realizavimo ir vėlesnių etapų įvertinimas.....	46
3.5. Informacinės sistemos programinės įrangos darbų apimties įvertinimo modelio realizacija.....	47
3.5.1. Galimų įgyvendinimo priemonių variantų analizė.....	47
3.5.2. Prototipo duomenų koncepcinis modelis.....	47
3.5.3. Prototipo nefunkciniai reikalavimai ir apribojimai.....	48
3.5.4. Bendras prototipo elgsenos modelis.....	49
3.5.5. Prototipo komponentų modelis.....	50
3.5.6. Įdiegimo modelis.....	52
4. EKSPERIMENTINIS INFORMACINĖS SISTEMOS PROGRAMINĖS ĮRANGOS KŪRIMO DARBŲ APIMTIES ĮVERTINIMO MODELIO TYRIMAS.....	53
4.1. Projekto „X“ įverčių tyrimas.....	53
4.2. Projekto reikalavimų ir analizės etapas.....	55
4.3. Projektavimo etapas.....	66
4.4. Realizavimo ir vėlesni etapai.....	67
5. IŠVADOS.....	69
6. LITERATŪRA.....	71

Lentelių sąrašas

1 lentelė.	Eilučių atrinkimo taisyklių rinkinys.	14
2 lentelė.	Išorinio įvedimo taško sudėtingumo klasifikatorius.....	16
3 lentelė.	Išorinio išvedimo ir užklauso taško sudėtingumo klasifikatorius.....	16
4 lentelė.	Loginio vidinio ir išorinio vartotojo sąsajos taško sudėtingumo klasifikatorius.	17
5 lentelė.	Funkcinių taškų svorių klasifikatorius.	17
6 lentelė.	Sistemos charakteristikų sąrašas.	17
7 lentelė.	Sudėtingumo laipsniai ir jų reikšmės.....	18
8 lentelė.	Skaičiavimo šablonas.....	19
9 lentelė.	FP konvertavimas į SLOC.	19
10 lentelė.	Programinių modulių tipai bei sudėtingumas.....	20
11 lentelė.	Programinio modulio sudėtingumo klasifikatorius.	21
12 lentelė.	Naudojamų strategijų privalumai ir trūkumai.	22
13 lentelė.	a, b koeficientų reikšmių klasifikatorius.	23
14 lentelė.	c koeficiento reikšmių klasifikatorius.....	24
15 lentelė.	Dydis, įvertinantis programuotojo sugebėjimus.....	25
16 lentelė.	Laipsnio parametrai.	26
17 lentelė.	Laipsnio parametrų reikšmių klasifikatorius.....	26
18 lentelė.	Projekto sąnaudų bei aplinkos veiksmų parametrai.....	26
19 lentelė.	Projekto sąnaudų bei aplinkos veiksmų parametrų reikšmių klasifikatorius.	27
20 lentelė.	Projekto sąnaudų bei aplinkos veiksmų parametrai.....	28
21 lentelė.	Projekto sąnaudų bei aplinkos veiksmų parametrų reikšmių klasifikatorius.....	28
22 lentelė.	Galimų produktyvumo koeficientų klasifikatorius.....	30
23 lentelė.	Projekto dydžių koeficientų klasifikatorius.	30
24 lentelė.	Metodų palyginimo pagal skaičiavimo charakteristikas suvestinė.....	33
25 lentelė.	Metodų privalumų ir trūkumų suvestinė.....	34
26 lentelė.	Laikinių įverčių rezultatų suvestinė.....	44
27 lentelė.	Prototipo nefunkciniai reikalavimai.	48
28 lentelė.	Prototipo komponentų aprašas.	50
29 lentelė.	Projekto „X“ darbų – funkcijų sąrašas.....	53
30 lentelė.	Modulių vertinimo klasifikatorius pagal analogą.....	55
31 lentelė.	Projektui „X“ parinkti Basic COCOMO modelio parametrai.....	55
32 lentelė.	Projekto „X“ darbų – funkcijų apimties / dydžio matavimai pagal funkcinių taškų..... metodą.....	56
33 lentelė.	Projektui „X“ parinkti Putnam modelio parametrai.	59

	6
34 lentelė. Projektui „X“ parinkti COCOMO II aplikacinio komponavimo modelio parametrai. ...	60
Šiame modelyje funkcijų apimtis matuojama objektiniais taškais. Žemiau pateikiami struktūriniai.....	
skaičiavimai:	60
35 lentelė. Projekto „X“ darbų – funkcijų apimties / dydžio matavimai pagal objektinių taškų.....	
metodą.....	60
36 lentelė. Reikalavimų ir analizės etapo metodų suvestinė - projekto „X“ darbų – funkcijų.....	
skirtingi svoriai bei skirtingų metodų laikiniai įverčiai.	64
37 lentelė. Projekto „X“ rezultatų suvestinė reikalavimų bei analizės etape.....	65
38 lentelė. Projektui „X“ parinkti COCOMO II Ankstyvo projektavimo modelio parametrai.	66

Paveikslėlių sąrašas

1 pav.	Projekto vadovo bei darbuotojų funkcijų - darbų analizės veiklos procesas.....	12
2 pav.	Norden/Rayleigh kreivė.....	29
3 pav.	IS projekto gyvavimo ciklo etapai.....	39
4 pav.	VIVEMO-7 darbo procesas.	40
5 pav.	Įvykio neįvykimo tikimybės pasiskirstymo pagal z dydį grafikas.....	44
6 pav.	Prototipo duomenų koncepcinis modelis.....	48
7 pav.	Komunikavimas tarp vartotojų ir prototipo sistemos.	50
8 pav.	Prototipo programinių komponentų modelis.....	51
9 pav.	Prototipo įdiegimo modelis.....	52
10 pav.	Projekto ir projekto dalių forma.....	73
11 pav.	Projekto etapų ir funkcijų forma.....	73
12 pav.	Darbuotojų forma.....	74
13 pav.	Analoginio metodo forma.....	74
14 pav.	Basic COCOMO metodo forma.....	75
15 pav.	Basic COCOMO parametrų forma.....	75
16 pav.	Projekto ir projekto dalių programinio modulio sekų diagrama.....	76
17 pav.	Projekto etapų programinio modulio sekų diagrama.....	77
18 pav.	Ekspertinio metodo programinio modulio sekų diagrama.....	78
19 pav.	Funkcinių taškų metodo programinio modulio sekų diagrama.....	79
20 pav.	Objektinių taškų metodo programinio modulio sekų diagrama.....	80
21 pav.	Sisteminės analizės metodo programinio modulio sekų diagrama.....	81
22 pav.	Cocomo II, 1 metodo programinio modulio sekų diagrama.....	82
23 pav.	Cocomo II, 2 metodo programinio modulio sekų diagrama.....	83
24 pav.	Darbų paskyrimo programinio modulio sekų diagrama.....	84
25 pav.	Konkreto darbuotojo darbų sąrašo ir apkrovimo programinio modulio sekų diagrama.....	85
26 pav.	Įverčių skaičiavimo forma (ekspertiniu metodu).....	86
27 pav.	Įverčių skaičiavimo forma (funkciniu metodu).....	87
28 pav.	Įverčių skaičiavimo forma (COCOMO II 2 modeliu).....	88
29 pav.	Įverčių skaičiavimo forma (COCOMO II 1 modeliu).....	89
30 pav.	Įverčių skaičiavimo forma (COCOMO II 1 modeliu).....	90

1. ĮVADAS

Iki šiol Lietuvoje nepakankamai dėmesio ir pastangų skiriama kuriamos programinės įrangos laiko sąnaudoms nustatyti. Dažniausiai viską diktuoja kaina, kuri derinama pagal užsakovo finansines galimybes. Vadovaujamosi tokiais principais kaip "kainuos tiek, kiek kainuos", „užtruks tiek, kiek užtruks“. Užsakovai dar nesigilina į kainos nustatymo niuansus, apsiribodami pastangomis truputį ją sumažinti.

Šiuo metu Lietuvos IT rinka yra labai konkurencinga ir dinamiška. Firmos, norėdamos išlikti konkurencingos, siekia, kad klientams būtų pasiūlytos naujausios ir greičiausiai veikiančios technologijos arba tos technologijos, kurios geriausiai atitinka kliento veiklos specifiką. Programinės įrangos pasiūla viršija paklausą, todėl daugelis klientų organizuoja konkursus, konkrečiam projektui kurti informacinę sistemą. Kiekviename tokia konkurse yra reikalaujama, kad dalyvaujanti IT firma nurodytų terminą, iki kada būtų sukurta informacinė sistema, kiek kainuos sukūrimas, kokios technologijos bus naudojamos ir t.t. Dažnai svarbiausi atrankos kriterijai būna kaina ir sukūrimo laikas. Tačiau kiekvienas informacinės sistemos kūrimo projektas yra unikalus tiek kompiuterizuojamos veiklos, tiek technologijų panaudojimo prasme ir kuo jis unikalesnis tuo sunkiau prognozuoti būsimos sistemos apimtis bei sukūrimo laiką. Nors firmos ir nurodo konkrečias sumas bei terminus, tačiau labai dažnai vėliau atsiranda tokie dalykai kaip išsitęsę terminai, vėlavimai, viršytas projekto biudžetas ir t.t.. Praktiškai kiekviena firma nurodydama šią informaciją remiasi savo ankstesnių projektų kūrimo patirtimi. Dažnai pasitaikantis reiškinys yra, kai dalyvaujanti firma visiškai neturi patirties nei veiklos srityje, kurią nori ir pretenduoja kompiuterizuoti, nei patirties su technologijomis, kurių reikalaujama. Atsižvelgiant į rinkoje esančią konkurenciją galima bandyti netgi suprasti tokius atvejus – juk geriau laimėti konkursą, kurio pelno vos užteks padengti kūrimo kaštus, negu nelaimėti ir bankrutuoti. Tokiais atvejais pasitaiko, kad projekto pabaigoje tiek kaina, tiek sukūrimo laikas viršija iš projekto biudžetą du ir daugiau kartų. Taip nutinka tuomet, kai projekto pradžioje bei eigoje yra netinkamai įvertinama arba visai nėra įvertinama būsimos sistemos apimtis, atliekamų darbų bei viso projekto trukmės prognozavimai.

Projekto pabaigoje būna per vėlu iš esmės kažką keisti. *Brooks Law* teigia, kad projekto sudėtingumas didėja augant programuotojų skaičiui, dėl komunikavimo, versijų ir valdymo. Naujų žmoniškųjų resursų priskyrimas projekto pabaigoje tik dar daugiau užvėlina projektą. Todėl sukurta programinė įranga dažnai būna ne optimaliai sukurta, pasižymi veikimo netikslumu bei begale klaidų.

Kita vertus, įmonėms kartais nėra tikslinga rizikuoti laisvomis finansinėmis lėšomis, nuodugnai atliekant būsimos sistemos preliminarą analizę prieš paduodant paraišką konkursui. Nėra garantijos, kad įmonė, tinkamai įvertinusi būsimos sistemos apimtis, tiksliau prognozavusi kaštus, laimės konkursą. Dažnai nuodugnūs vertinimai bei skaičiavimai būna gana pesimistiniai ir veda prie

didesnės sukūrimo kainos bei ilgesnės trukmės. To klientai dažniausiai nesupranta ir įmonė tiesiog gali nelaimėti konkurso vien dėl aukštos kainos ir ilgo termino. Todėl forsavimas yra neišvengiama kiekvienos įmonės vadovybės strategijos dalis.

Nors detalus programinės įrangos apimties ir laikinių įverčių įvertinimas yra papildomos sąnaudos ir daugeliui projektų vadovų gali atrodyti kaip bereikalingas laiko gaišimas (jeigu tai tiesiogiai neatneša naudos), egzistuoja priežastys dėl, kurių verta tai daryti. Tinkami vertinimai bei skaičiavimai naudingi įvertinant faktinę projekto kainą, sekant projekto progresą, atsižvelgiant į sudarytus planus, anksčiau numatant vėlavimus ir laiku priskiriant daugiau žmogiškųjų resursų, sukaupiant informaciją apie daugkartinio panaudojimo programinius modulius ar posistemas ir panaudojant tai reinžinerijos procese. Šimtu procentu atspėti ar prognozuoti apimtį bei trukmę neįmanoma, tačiau vėlavimo paklaida daug kartų mažesnė negu visai to nedarant. Šiame darbe akcentuojama tinkamų metodų, modelių, vertinimo strategijos bei požiūrio svarba su tikslu kuo pagrįsčiau nustatyti projekto sąnaudas, laikinius įverčius bei kalendorinę trukmę tam tikra įvykio tikimybe.

Darbo tikslas – sukurti informacinės sistemos programinės įrangos darbų apimties įvertinimo modelį, kuris įgalintų visapusiškai ir pagrįstai įvertinti laiko sąnaudas. Šis modelis turi naudoti jau esamus populiarius vertinimo bei projekto laikinių įverčių skaičiavimo metodus, tačiau neprisirišti prie konkretaus vienos, naudoti metodus tinkamuose projekto etapuose, pateikti viso projekto bei atskirų darbų - funkcijų laikinius įverčius bei įvykdymo tikimybes.

Šiam tikslui pasiekti užsibrėžiami tokie uždaviniai:

1. Identifikuoti ir apibrėžti vartotojus, kuriuos liečia problema ir kuriems taikomas tyrimo objektas.
2. Pasirinkti ir išanalizuoti standartinius esamus projekto darbų – funkcijų apimties skaičiavimo metodus.
3. Pasirinkti ir išanalizuoti plačiai žinomus esamus projekto laikinių įverčių skaičiavimo metodus bei modelius.
4. Pateikti šių metodų bei modelių privalumus ir trūkumus.
5. Suformuoti ir apibrėžti informacinės sistemos programinės įrangos darbų apimties įvertinimo modelį bei pateikti jo esmę.
6. Apibrėžti modeliui reikalingus duomenis, pateikti modelio veiklos schemą bei proceso etapus.
7. Suprojektuoti prototipo statiką ir dinamiką.
8. Atlikti eksperimentinį informacinės sistemos programinės įrangos darbų apimties įvertinimo modelio tyrimą, naudojant projektą „X“.
9. Pateikti galutines išvadas.

2. INFORMACINĖS SISTEMOS PROGRAMINĖS ĮRANGOS KŪRIMO DARBŲ APIMTIES BEI TRUKMĖS ĮVERTINIMO METODŲ ANALIZĖ

2.1. Analizės tikslas

Šiuo metu pasaulyje yra daugybė metodų bei modelių skirtų projektų vertinimui, projekto apimties ir laikinių įverčių skaičiavimui. Vieni daugiau orientuoti į projekto informacinės sistemos apimties, kiti – į laikinių įverčių skaičiavimą. Šios analizės tikslas yra identifikuoti ir išanalizuoti plačiausiai žinomus bei naudojamus. Metodai bei modeliai pasirinkti pagal Pietų Kalifornijos Universiteto (*University of Southern California*) [4] bei Norvegijos *Simula* tyrimo laboratorijos (*Simula Research Laboratory*) [16] pateiktas apžvalgas bei populiarumą internete.

Analizės metu svarbu suvokti metodų tinkamumą bei pritaikomumą konkreitiems projekto etapams, projekto apimties laikinių įverčių skaičiavimo taisykles ir apribojimus, formules bei vertinimo strategijas. Trokštamas rezultatas – metodų bei modelių palyginimas tarpusavyje, privalumų ir trūkumų pateikimas.

2.2. Tyrimo sritis, objektas ir problema

Remiantis Norvegijos mokslininkų pateiktu straipsniu [16], kuriame susisteminti 10 skirtingų mokslininkų, iš skirtingų šalių, atliktų statistinių tyrimų pateikti rezultatai, apytiksliai 85% projektų buvo naudojamas ekspertinis informacinės sistemos programinės įrangos kūrimo darbų apimties bei trukmės vertinimo ir skaičiavimo metodas, kurio panaudojimas glaudžiai susijęs su analoginiu vertinimo metodu. Tai reiškia, kad arba projektų vadovas įvertindavo projekto darbų apimtį pagal analoginius praeityje buvusius projektus arba projekto komandoje esantys specialistai – ekspertai diskusijos ir konsensuso būdu atliko prognozes. Taip pat pastebėta, kad įmonės gana dažnai naudoja „*price to win*“ metodą bei principą, kurio esmė "kainuos tiek, kiek kainuos", „užtruks tiek, kiek užtruks“. Tai patys paprasčiausi ir mažai laiko sąnaudų reikalaujantys metodai, kurie taip pat dažnai yra naudojami ir Lietuvos IT įmonėse. Tačiau vien tik šie metodai nepadedą išvengti žemo laiku atliktų IT projektų procento – 16.2% [10], kad būtų atlikti laiku bei neviršytų suplanuoto biudžeto. Kituose šaltiniuose šis procentas gali skirtis, tačiau problemos tai nekeičia.

Visgi ekspertinis ir analoginis metodai nėra vieninteliai metodai. Egzistuoja ir kiti metodai, kurie ne subjektyviai, o pagrįstai atlieka informacinės sistemos programinės įrangos kūrimo darbų apimties bei trukmės vertinimą bei skaičiavimą. Kokie tie metodai? Kokie jų privalumai, kokie trūkumai, kokie panašumai? Į šiuos klausimus ir buvo bandoma atsakyti.

2.3. Vartotojų analizė

2.3.1. Vartotojų aibė, tipai ir savybės

Informacinės sistemos programinės įrangos kūrimo darbų apimtys bei trukmės įvertinimo metodai skirti informacinių technologijų projektams. Žiūrint iš žmogiškųjų resursų pusės, projektą sudaro:

- Projekto vadovas,
- Projektui priskirti darbuotojai.

Darbuotojai pagal projekte atliekamų darbų specifiką gali būti klasifikuojami pagal specializacijas:

- Analitikai,
- Duomenų bazės projektuotojai,
- Programuotojai,
- Testuotojai,
- Informacinės sistemos priežiūrą atliekantys darbuotojai,
- Kiti.

2.3.2. Vartotojų tikslai ir problemos

Projekto vadovas yra asmuo, kuris atsako už projekto darbų planavimą, jų vykdymo sklandumą bei atsiskaito prieš vadovybę. Jis vienintelis įpareigotas suvokti, identifikuoti bei nustatyti viso projekto apimtį ir suplanuoti jo trukmę. Šią užduotį jis atlieka padedamas:

- Projekto komandos narių – specialistų.
- Informacinės sistemos programinės įrangos kūrimo darbų apimtys bei trukmės vertinimo bei skaičiavimo metodų.

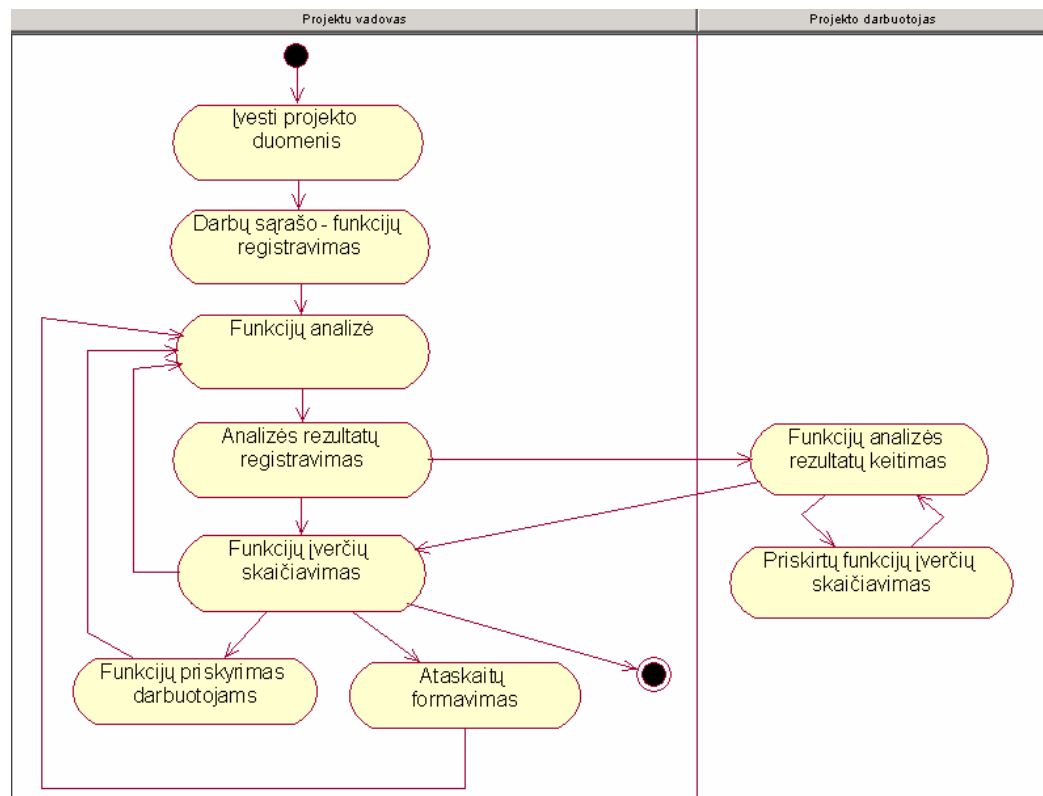
Pirmu atveju rezultatas išlieka gana subjektyvus, nes projekto specialistai – ekspertai yra žmonės ir remiasi savo ankstesne patirtimi. Nors laiko sąnaudos nedidelės, tačiau nustačius projekto darbų tiek apimtys bei trukmės rezultatus nėra, jokios garantijos, kad jie teisingi ar iš dalies teisingi. Nėra jokio ryšio su įvykio tikimybe, t.y. kiek patikimas yra rezultatas.

Antru atveju skaičiavimai yra daugiau pagrįsti, tačiau egzistuoja gana daug metodų ir neaišku, kuris metodas tinkamiausias. Viename projekte gali pasiteisinti vieno, kitame kito metodo panaudojimas. Dažnai kartą nepateisinęs lūkesčių bei sugaišto laiko metodo panaudojimas būna pasmerkiamas motyvuojant tuo, kad nėra jokios garantijos, jog jį naudojant rezultatas yra tikslesnis nei ekspertinio ar analoginio, o laiko sąnaudos yra didesnės. Todėl grįžtama prie pirmo punkto.

Nors projektų vadovas ir vykdo daugiausia vertinimų, skaičiavimų bei planavimų, tačiau tai atlieka ir patys projekto nariai, kuriems yra padalijami darbai. Kiekvienas projekto darbuotojas savarankiškai įvertina savo darbus, jų apimtį bei įsipareigoja juos atlikti laiku.

2.4. Projekto valdymo proceso analizė

Už projektą atsakingas ne tik projektų vadovas bet ir projekto nariai – specialistai. Tik vieningai dirbanti komanda gali pasiekti gerų rezultatų. Projekto pradžioje (ankstyvuose projekto etapuose) projekto vadovas įvertina visą projektą, vėliau išskaido jį funkcijomis - darbais ir preliminariai analizuoja kiekvieną iš jų. Priklausomai nuo darbo specifikos atitinkamą funkciją - darbą priskiria atitinkam projektui, kurie jas savo ruožtu analizuoja, vertina jų apimtį bei įsipareigoja atlikti laiku. Šis procesas iteracinis ir tęsiasi tol, kol visos funkcijos yra peržvelgiamos ir nustatomas pradinis darbų planas bei viso projekto trukmė. Vėlesniuose etapuose tiek darbų apimtis, tiek planai tikslinami priklausomai nuo papildomos informacijos, kuri paaiškėja eigoje. Principinė tiek projekto vadovo, tiek projekto darbuotojų darbo schema vertinant projekto darbų apimtį bei trukmes pateikta veiklos diagrama:



1 pav. Projekto vadovo bei darbuotojų funkcijų - darbų analizės veiklos procesas.

2.5. Programinės įrangos kūrimo darbų apimties įvertinimo metodų analizė

Kiekviena kuriama informacinė sistema yra unikali ir jos kūrimui yra naudojami unikalūs realizaciniai sprendimai. Todėl tikslaus modelio ar metodo, kuris leistų tiksliai atsakyti į klausimą „per kiek laiko, pagal turimus resursus, pavyks sukurti informacinę sistemą?“ ir nėra. Kiekvienas metodas turi savo privalumus ir trūkumus bei pritaikomumą. Visi vertinimo metodai gali būti skiriami į dvi vertinimo strategijas :

- Iš viršaus į apačią (*Top-down*).
- Iš apačios į viršų (*Bottom-up*).

Norint atsakyti į klausimą „kokia projekto trukmė“ reikia pirma atsakyti į klausimą „kokia darbų apimtis“. Informacinės sistemos apimčiai nustatyti naudojami šie metodai:

- Kodo eilučių skaičiavimo metodas [13,17,20].
- FP - funkcinių taškų metodas [6,9,15,22].
- OP - objektinių taškų metodas [2,5,8].

Informacinės sistemos programinės įrangos kūrimo darbų trukmei nustatyti naudojami šie metodai bei modeliai:

- Analoginis vertinimo metodas [6, 11,14,19].
- Ekspertinis vertinimo metodas [6, 11,12,14,16].
- Basic COCOMO modelis [6, 11,14,18].
- COCOMO II : Aplikacinis komponavimo modelis [3, 11,14, 20].
- COCOMO II : Ankstyvo projektavimo modelis [3, 11,14, 20].
- COCOMO II : Vėlyvo projektavimo modelis [3, 11,14, 20].
- PUTNAM modelis [11,14,20].
- Sistemine analize orientuotas metodas [6,7].

2.5.1. Projekto apimties / dydžio skaičiavimo metodai

2.5.1.1. Kodo eilučių skaičiavimo metodas

Programinės įrangos apimties nustatymas yra pagrindas laikinių įverčių skaičiavimams. Dydžio skaičiavimo metodų nėra daug ir tarp jų fizinių kodo eilučių skaičiavimas vienas iš seniausiai žinomų bei naudojamų. Galimi skaičiavimo vieneto akronimai – LOC, SLOC, KLOC, KSLOC ir DSLOC, kur K žymi tūkstančius, o D – pristatytos (*Delivered*) ir realizuotos kodo eilutės. ++

Vertinimo proceso etapai:

- 1) Identifikuoti atributus objektų, kurie juos charakterizuoja ir kuriuos planuojama matuoti.
- 2) Identifikuoti šių atributų galimas reikšmes.
- 3) Susisteminti atributus, apibrėžti jų reikšmes ir sudaryti skaičiavimo taisyklių rinkinį.

4) Sudaryti duomenų užpildymo lentelę.

5) Skaičiuoti kodo eilutes.

Robert E. Park [17] atlikęs tyrimą susistemino rezultatus ir sudarė taisyklių rinkinį:

1 lentelė. Eilučių atrinkimo taisyklių rinkinys.

Matavimo vienetas:		kodo eilutės	
Nr.	Sakinio tipas	Įeina	Neįeina
1	Vykdomasis	+	
2	Nevykdomasis:		
2.1	Kintamojo deklaravimas	+	
2.2	Kompiliatoriaus kreipiniai	+	
2.3	Komentarai:		
2.3.1	Atskiroje eilutėje		+
2.3.2	Eilutėje kartu su kodu		+
2.3.3	Tušti komentarai		+
Nr.	Kaip dirbama	Įeina	Neįeina
1	Programuojama	+	
2	Generuojama	+	
3	Automatiškai konvertuojama	+	
4	Kopijuojama arba panaudojama be pakeitimų	+	
5	Keičiama	+	
6	Šalinama		+
Nr.	Šaltinis	Įeina	Neįeina
1	Pirminė inžinerija	+	
2	Reinžinerija	+	
Nr.	Panaudojimas	Įeina	Neįeina
1	Pilnas arba dalinis pagrindiniam produktui	+	
2	Išorinis arba pagrindinio produkto palaikymui		+
Nr.	Realizacija/pristatymas	Įeina	Neįeina
1	Atlikta:		
1.1	Pristatytas kodas	+	
1.2	Pristatyta sukompiliuotame arba vykdomame pavidale, bet ne pats kodas	+	

Nr.	Realizacija/pristatymas	Įeina	Neįeina
2	Nepristatyta:		
2.1	Vis dar derinama / konfigūruojama		+
2.2	Net nekonfigūruojama		+
Nr.	Funkcionalumas	Įeina	Neįeina
1	Naudojamas	+	
2	Nenaudojamas (pasenęs, nepalaikomas, neprieinamas):		
2.1	Nenaudojamas dėl specialių priežasčių	+	
2.2	Kita		+
Nr.	Kopijavimas	Įeina	Neįeina
1	Naudojamas originalo kodas	+	
2	Kopijuojami fragmentai iš originalo kodo	+	
Nr.	Programavimo kalba	Įeina	Neįeina
1	Bendri elementai:	+	
1.2	<i>Nulls, continues</i> ir <i>no-ops</i>	+	
1.3	<i>Begin...end</i> pora	+	
1.4	Loginės išraiškos ir testavimo sąlygos	+	
1.5	Paprogramių argumentai kaip išraiškos	+	
1.6	Pabaigos komandos, nutraukiančios vykdomuosius sakinius	+	
1.7	<i>Then, else</i> ir <i>otherwise</i>	+	
1.8	<i>Elseif</i> sakiniai	+	
1.9	Baziniai žodžiai tokie kaip <i>like procedure division, interface</i> ir <i>implementation</i>	+	
2	Specifiniai kalbos elementai	+	
2.1	+	
2.2	+	

Pagal šią lentelę skaičiuojamos kodo eilutės, kurios panaudojamos skaičiuojant informacinės sistemos dydį / apimtį. Plačiau apie metodą pateikta šaltiniuose [13,17,20].

2.5.1.2. Funkcinių taškų metodas

Šį metodą aprašė A. J. Albrechtas, kuris sistemos dydį / apimtį pasiūlė matuoti funkciniais taškais. Kodo eilučių skaičiavimo metode prieš skaičiuojant apimtis reikia sudaryti ir apibrėžti

taisyklių rinkinį, kuris kinta priklausomai nuo programavimo kalbos. Tuo tarpu funkcinių taškų metodas grindžiamas idėja, kad visos IS susideda iš 5 pagrindinių komponentų, kurie įvardinami kaip funkcinių taškų tipai.

Vertinimo proceso etapai:

- 1) Nagrinėti kuriamą IS identifikuojant nepritaikytus funkcinius taškus pagal 5 tipus:
 - 1) Išorinio įvedimo taškas (*External Inputs*) – tai rašymo, informacijos keitimo transakcijos. Pvz. tam tikro duomenų rinkinio registravimas. Sutrumpintai žymėsime EI.
 - 2) Išorinio išvedimo taškas (*External Outputs*) – tai informacijos išvedimo, ataskaitų ar dokumentų formavimo operacijos. Pvz. ataskaitos apie mokesčių mokėtoją formavimas ir spausdinimas. Sutrumpintai žymėsime EO.
 - 3) Išoriniai užklausų taškas (*External Inquires*) - tai vartotojo transakcijos, kurios teikia informaciją, bet nekeičia jos. Tačiau informacija neišeina už sistemos ribų. Pvz. peržiūros formos. Sutrumpintai žymėsime EQ.
 - 4) Loginiai vidiniai taškas (*Internal Logical Inputs*) – tai savotiškas esybių identifikavimas (nebūtinai duomenų bazės lentelės. Tai gali būti failų tipai, kiti operacinės sistemos objektai). Sutrumpintai žymėsime ILI.
 - 5) Išoriniai vartotojo sąsajos taškas (*External Interface Inputs*) – tai informacija arba failai, kuriais vyksta mainai su kitomis sistemomis. Pvz. duomenys apie mokesčių mokėtojo darytus pavedimus eksportuojami į Excel programą. Sutrumpintai žymėsime EII.
- 2) Įvertinamas kiekvieno taško sudėtingumas pagal bylų skaičių ir duomenų elementų skaičių:

2 lentelė. Išorinio įvedimo taško sudėtingumo klasifikatorius.

Bylų skaičius	Duomenų elementų skaičius		
	1-4	5-15	>15
0-1	Žemas	Žemas	Vidutinis
2	Žemas	Vidutinis	Aukštas
3 ir daugiau	Vidutinis	Aukštas	Aukštas

3 lentelė. Išorinio išvedimo ir užklausos taško sudėtingumo klasifikatorius.

Bylų skaičius	Duomenų elementų skaičius		
	1-5	6-19	>19
0-1	Žemas	Žemas	Vidutinis
2-3	Žemas	Vidutinis	Aukštas
> 3	Vidutinis	Aukštas	Aukštas

4 lentelė. Loginio vidinio ir išorinio vartotojo sąsajos taško sudėtingumo klasifikatorius.

Pogrūpių skaičius	Duomenų elementų pogrūpyje skaičius		
	1-19	20-50	>50
0-1	Žemas	Žemas	Vidutinis
2-5	Žemas	Vidutinis	Aukštas
> 5	Vidutinis	Aukštas	Aukštas

3) Kiekvienam taškui pagal tipą bei sudėtingumą priskiriamas svoris:

5 lentelė. Funkcinių taškų svorių klasifikatorius.

Tipas	Daugiklio koeficientas (pagal sudėtingumą)		
	Paprastas	Vidutinis	Sudėtingas
Išorinis įvedimo taškas	3	4	6
Išorinis išvedimo taškas	4	5	7
Išorinis užklausų taškas	3	4	6
Loginis vidiniai taškas	7	10	15
Išorinis vartotojo sąsajos taškas	5	7	10

4) Būsima IS nagrinėjama pagal sistemos charakteristikas bei jų sudėtingumo įtaką apimčiai:

6 lentelė. Sistemos charakteristikų sąrašas.

Nr.	Bendros sistemos charakteristikos	Komentaras
1.	Duomenų apsikeitimai	Kokie yra realizuoti patogūs ir paprasti apsikeitimo su sistema mechanizmai?
2.	Paskirstytų duomenų apdorojimas	Kaip yra paskirstomi duomenys ir apdorojamos funkcijos?
3.	Vykdymas	Ar vartotojas pateikė reikalavimus reakcijos laikui, pralaidumui arba našumui?
4.	Techninės įrangos panaudojimas	Kaip smarkiai yra naudojama dabartinė techninė įranga, kur bus patalpinta ir vykdoma sukurta sistema?
5.	Transakcijų dažnumas	Kaip dažnai yra vykdomos transakcijos? Kasdien, kas savaitę, kas mėnesį ir t.t.?
6.	Tiesioginis duomenų registravimas	Koks procentas duomenų yra tiesioginiai registruojamas (<i>On-line</i>)

Nr.	Bendros sistemos charakteristikos	Komentaras
7.	Naudingumas vartotojui	Koks sukurtos sistemos naudingumas galutiniam vartotojui?
8.	Tiesioginis atnaujinimas	Kiek vidinių loginių taškų yra atnaujinama tiesioginės transakcijos metu?
9.	Sudėtinis apdorojimas	Ar sistemoje realizuotas išplėstinis loginis ar matematinis apdorojimo mechanizmas?
10.	Reinžinerija	Ar kuriant sistemą buvo realizuoti tik vartotojo pateikti reikalavimai ar realizuota daugiau?
11.	Diegimo paprastumas	Kaip sudėtingas yra diegimo ir perėjimo procesas?
12.	Naudojimo paprastumas	Kiek patogios, paprastos ir automatizuotos naudojimui yra užkrovimo, duomenų archyvavimo ir atstatymo procedūros?
13.	Daugkartinis panaudojimas	Ar šia sistema gali naudotis skirtingos organizacijos?
14.	Pakeitimų palengvinimas	Ar kuriant sistemą buvo realizuoti kokie nors pakeitimų kūrimo mechanizmai?

5) Įvertinamas kiekvienos sistemos charakteristikos sudėtingumo laipsnis ir priskiriama reikšmė:

7 lentelė. Sudėtingumo laipsniai ir jų reikšmės.

Reikšmė, C_i	Įtakos laipsnis
0	Nėra / neturi
1	Netikėta
2	Nedidelė
3	Vidutinė
4	Reikšminė
5	Didelė

6) Skaičiuojamas pritaikymo koeficientas - VAF:

$$VAF = 0.65 + \frac{\sum_{i=1}^{14} C_i}{100}$$

Po to kai visos sistemos charakteristikos yra įvertintos tam tikromis įtakos laipsnio reikšmėmis skaičiuojamas pritaikymo koeficientas (*VAF – Value Adjustment Factor*), kuris naudojamas nepritaikytiems funkciniais taškams konvertuoti į pritaikytus funkcinis taškus.

- 7) Sudaroma skaičiavimo lentelė, kurioje registruojami visi nepritaikyti funkciniai taškai, dauginami iš jų sudėtingumo koeficientai, sumuojami ir dauginami iš pritaikymo koeficiento.

8 lentelė. Skaičiavimo šablonas.

Tipas	Sudėtingumas									VISO
	Žemas			Vidutinis			Aukštas			
	Kiek	Koef.	Viso	Kiek	Koef.	Viso	Kiek	Koef.	Viso	
EI	X	3	X*3	X	4	X*4	X	6	X*6	...
EO	X	4	X*4	X	5	X*5	X	7	X*7	...
EQ	X	3	X*3	X	4	X*4	X	6	X*6	...
ILI	X	7	X*7	X	10	X*10	X	15	X*15	...
EII	X	5	X*5	X	7	X*7	X	10	X*10	...
Nepritaikytų taškų reikšmių suma										...
Pritaikymo daugiklis										...
Pritaikytų taškų reikšmių suma										...

Galutinis pritaikytų taškų reikšmių sumos skaičius atspindi sistemos apimtį pritaikytais funkciniais taškais. Ši suma gauta naudojant pritaikymo koeficientą, kuris įvertina įtakos faktorius sistemos charakteristikoms. Tačiau tam tikri laikinių įverčių skaičiavimo metodai, skaičiuodami sistemos apimtį pagal funkcinį taškų metodą, skaičiuoja nepritaikytus funkcinis taškus nenaudojant pritaikymo koeficiento.

Galimas ryšys tarp funkcinį taškų ir fizinių kodo eilučių. Žemiau pateikta funkcinį taškų konvertavimo lentelė pagal programavimo kalbą:

9 lentelė. FP konvertavimas į SLOC.

Programavimo kalba	Kodo eilučių skaičius per funkcinį tašką
<i>Assembler</i>	320
<i>Assembler (Macro)</i>	213
<i>Algol</i>	106
<i>Cobol</i>	106
<i>Fortran</i>	106
<i>Jovial</i>	106
<i>Pascal</i>	91

Programavimo kalba	Kodo eilučių skaičius per funkcinį tašką
<i>RPG</i>	80
<i>PL/I</i>	80
<i>Ada</i>	71
<i>Lisp</i>	64
<i>Basic</i>	64
<i>4th Generation Database</i>	40
<i>APL</i>	32
<i>Smalltalk</i>	21
<i>Query Languages</i>	16
<i>Spreadsheet Languages</i>	6
<i>C</i>	150
<i>C++ default</i>	53
<i>Delphi</i>	18
<i>Html4</i>	14
<i>Java 2 default</i>	46
<i>Visual basic 6</i>	24
<i>ANSI/Quick/Turbo Basic</i>	64
<i>SQL default</i>	13
<i>Report Generator</i>	80
<i>PL/Sql</i>	45

Sistemos apimtis, išmatuota tiek fizinėmis kodo eilutėmis, tiek funkciniais taškais, naudojama laikinių įverčių ir trukmės skaičiavimo metoduose. Plačiau apie funkcinį taškų metodą pateikta šaltiniuose [6,9,15,22].

2.5.1.3. Objektinių taškų metodas

Šis metodas, vertindamas informacinę sistemą, charakterizuoja tai objekciniais taškais ir suskaido iki konkrečių programinių modulių, vertinant jų sudėtingumą:

10 lentelė. Programinių modulių tipai bei sudėtingumas.

Forma				Ataskaita			
Virtualių lentelių skaičius (View'ai)	Duomenų lentelių skaičius ir šaltinis			Virtualių lentelių skaičius	Duomenų lentelių skaičius ir šaltinis		
	<4 (<2 srvr < 3 clnt)	< 8 (2/3 srvr 3-5 clnt)	8+ (>3 srvr < 5 clnt)		<4 (<2 srvr < 3 clnt)	<8 (2/3 srvr 3-5 clnt)	8+ (>3 srvr < 5 clnt)
< 3	paprastas	paprastas	vidutinis	0 arba 1	paprastas	paprastas	vidutinis
3 - 7	paprastas	vidutinis	sudėtingas	2 arba 3	paprastas	vidutinis	sudėtingas
> 8	vidutinis	sudėtingas	sudėtingas	4 ir daugiau	vidutinis	sudėtingas	sudėtingas

Srvr – sisteminės lentelės, naudojamos ekraninėje formoje arba ataskaitoje.

Clnt – vartotojiškos arba veiklos lentelės, naudojamos ekraninėje formoje arba ataskaitoje.

Po to, kai yra identifikuojama kiekvieno programinio modulio (formos arba ataskaitos) sudėtingumas reikia priskirti svorį. Žemiau esančioje lentelėje pateiktos galimos svorių reikšmės:

11 lentelė. Programinio modulio sudėtingumo klasifikatorius.

Objekto tipas	Sudėtingumo koeficientas		
	Paprastas	Vidutinis	Sudėtingas
Forma	1	2	3
Ataskaita	2	5	8
Komponentas			10

Visi svoriniai įverčiai sudedami ir gaunamas vienas dydis - OP (*ObjectPoints*).

$$Size = \sum_i^t OP_i$$

Plačiau apie metodą pateikta šaltiniuose [2,5,8].

2.5.2. Projekto trukmės - laikinių įverčių skaičiavimo metodai

2.5.2.1. Analoginis vertinimo metodas

Nustatyti trukmę nėra itin keblu, jei įgyvendinamas projektas yra panašus į praeityje vykusį projektą. Tada galima remtis istorine patirtimi ir apytiksliai įvertinti reikalingas laiko sąnaudas. Trukmės paklaida absoliučiąja prasme nebus didelė jeigu projektas yra daugkartinio panaudojimo.

Reikalingi etapai:

- 1) Naujo projekto reikalavimų ir charakteristikų identifikavimas.
- 2) Ieškoti labiausiai „panašių“ praeityje atliktų projektų, jų charakteristikų.
- 3) Parinkti labiausiai „panašių“ projektų charakteristikų laikinius įverčius.
- 4) Susumuoti viso projekto laikinį įvertį.

Šis metodas turi daug panašumu su ekspertiniu vertinimu vien dėl to, kad vertinimai atliekami subjektyviai, t.y. priklauso nuo vertintojo. Skiriasi tuo, kad vertinimą dažniausiai atlieka tik projektų vadovas.

Plačiau apie metodą pateikta šaltiniuose [6, 11,14,19].

2.5.2.2. Ekspertinis vertinimo metodas

Vertinimas pagal intuiciją yra iš esmės pats paprasčiausias ir mažiausiai resursų reikalaujantis metodas. Geras projekto vadovas turi mokėti vertinti ir skaičiuoti laikinius įverčius ne tik naudodamas algoritminius metodus, bet ir, esant reikalui, naudodamasis savo bei ekspertų ankstesnių projektų sukauptą patirtimi bei įgūdžiais. Šis metodas glaudžiai siejasi su analoginiu metodu, kadangi ekspertas be ankstesnių projektų patirties bei įgūdžių – ne ekspertas.

Remiantis Norvegijos mokslininkų pateiktu straipsniu [16], kuriame susisteminti 10 skirtingų mokslininkų, iš skirtingų šalių, atliktų statistinių tyrimų pateikti rezultatai, apytiksliai 85% projektų buvo naudojamas ekspertinis vertinimo ir skaičiavimo metodas. Dauguma šio metodo šalininkų teigia,

jog nėra jokios garantijos, kad naudojant kitus formalius ar algoritminius skaičiavimo metodus bus gauti tikslesni laikiniai įverčiai. Taip pat teigiama, jog yra sutaupomas laikas nenaudojant kitų skaičiavimo metodų. Tačiau straipsnio autoriai pabrėžia, jog trukmės vertinimas ir skaičiavimas viso projekto metu yra sudėtingas reiškinys ir negali būti vienareikšmiškai apibrėžiamas. Taip pat reikia įvertinti žmoniškumo faktorių: projektų vadovai, kurie dalyvavo tyrime, galėjo klaidingai nurodyti tą informaciją, kuri nepriklausė nuo jų pačių, pvz. dėl kliento kaltės. O dėl firmos konfidencialumo politikos galėjo būti neatskleistos vienos ar kitos priežastys, o gal ir naudojami metodai.

Tai reiškia, kad tyrimo metu surinkti duomenys bei išvados gali būti ir klaidingi.

1) Vertinimo procesas

- a. Projekto vadovas kiekvienam ekspertui pateikia informacinės sistemos specifikaciją ir vertinimo formą.
- b. Projekto vadovas organizuoja susirinkimą, kuriame ekspertai diskutuoja apie galimas vertinimo problemas.
- c. Ekspertai įvertina projektą, jo dalis ir laikinius įverčius anonimiškai.
- d. Projekto vadovas susistemina ekspertų pateikta informaciją.
- e. Projekto vadovas organizuoja susirinkimą, kuriame akcentuoja diskutuoti dėl tų įverčių, kurie daugiausiai nesutampa ekspertų tarpe.
- f. Ekspertai vėl anonimiškai užpildo formas.
- g. Kartojama tol, kol sutariama ekspertų tarpe arba projekto vadovą tenkina gauti įverčiai.
- h. Įvertinti atskirai naudojant skirtingas „Iš viršaus į apačią“ ir „Iš apačios į viršų“ strategijas.

12 lentelė. Naudojamų strategijų privalumai ir trūkumai.

	„Iš viršaus į apačią“ (kaip visumą)	„Iš apačios į viršų“ (komponentinis)
Privalumai	Įgalina įvertinti, prognozuoti netikėtumus, sunkumus. Prognozavimas bei įvertinimas paremiamas istorine projektine patirtimi.	Šio proceso metu gaunama detalesnė informacija apie projekto sudedamąsias dalis, įgalina tiksliau atlikti projekto planavimą bei vykdymą.
Trūkumai	Dėl informacijos trūkumo sudėtinga sudaryti detalų projekto planą bei jį vykdyti. Labai priklauso kokie ankstesni projektai bus pasirinkti palyginimui iš atminties arba pagal dokumentaciją.	Gana didelė tikimybė pervertinti neprognozuojamus sunkumus. Didele dalimi priklauso nuo pasirinktų programinės įrangos kūrėjų su reikiama patirtimi.

- i. Tarpiniai įvertinimo taškai. Norint valdyti įvertinimo tikslumą būtina įvesti tarpinius laikinius taškus. Pvz. po kiekvienos projekto fazės palyginti prognozuotas laiko sąnaudas su realiomis laiko sąnaudomis.

- j. Skirtingi ekspertai. Naudojant kelis skirtingus ekspertus toms pačios užduotims įvertinti gaunami rezultatai, kuriuos apibendrinus gaunami tikslesni laikiniai įverčiai.
- 2) Išvadų analizė ir mokymasis iš klaidų. Projektas vyksta etapais ir po kiekvieno iš gali būti aptariami tarpiniai įvertinimo taškai, įvertinami faktoriai turėję įtakos trukmei. Tai įgalina mokintis iš klaidų.

Plačiau apie metodą pateikta šaltiniuose [6, 11,12,14,16].

2.5.2.3. Basic COCOMO modelis

Basic COCOMO (*CO*nstructive *CO*st *MO*del) modelis yra pagrįstas nominalių žmogaus darbo mėnesių (*Nominal person months*) skaičiavimu:

$$\text{Effort} = a \times (\text{KSLOC})^b \text{ arba}$$

Effort – projekto sąnaudos, išreikštos mėnesiais, KSLOC – skaičius parodantis darbo apimtis tūkstančiais fizinių kodo eilučių, a ir b – koeficientai, kurių reikšmės priklauso nuo pasirinkto režimo tipo:

- *Organic mode* – šis režimas skirtas mažos apimties projektams, kuriuos atlieka mažo narių skaičiaus darbinės komandos. Kuriamą sistemą yra ne sunku priderinti prie pasikeitusių reikalavimų.
- *Embedded mode* – šis režimas reiškia, kad kuriama sistema operuos su labai svarbiais duomenimis ir bet kokie pakeitimai sistemoje yra didelių finansinių išlaidų priežastis.
- *Semi – detached mode* – šis režimas tai kombinacija prieš tai paminėtų režimų.

Koeficientų reikšmės nustatomos pagal pasirinktą režimą:

13 lentelė. a, b koeficientų reikšmių klasifikatorius.

Sistemos tipas	a	b
<i>Organic</i>	2.4	1.05
<i>Semi – detached</i>	3.0	1.12
<i>Embedded</i>	3.6	1.20

Kalendorinė darbo trukmė mėnesiais apskaičiuojama pagal apskaičiuotas sąnaudas:

$$\text{Schedule} = 2.5 \times \text{Effort}^c$$

c – koeficientas, priklausantis nuo pasirinkto režimo:

14 lentelė. c koeficiento reikšmių klasifikatorius.

Sistemos tipas	c
<i>Organic</i>	0.38
<i>Semi – detached</i>	0.35
<i>Embedded</i>	0.32

Tačiau sistemos įvertinimas yra sudėtingesnis ir dažnai priklauso nuo egzistuojančių veiksmų, kurie nėra įvertinami šiame metode:

- Programinės įrangos patikimumas,
- Duomenų bazės dydis,
- Sistemos sudėtingumas,
- Programos vykdymo laiko apribojimas,
- Pagrindinės saugyklos apribojimai,
- Kompiuterinio staigaus pasikeitimo laikas,
- Analitikų sugebėjimai,
- Aplikacijų kūrimo patirtis,
- Programuotojų sugebėjimai,
- Virtualios mašinos patirtis,
- Programavimo kalbų patirtis,
- Turimos kūrimo programinės įrangos,
- Projekto plano, terminų.

Šis metodas yra paprastas naudoti, tačiau dėl to prarandamas tikslumas. Plačiau apie metodą pateikta šaltiniuose [6, 11,14,18].

2.5.2.4. COCOMO II : Aplikacinis komponavimo modelis

Keičiantis technologijoms, sistemų įvairiapusiškumui, sudėtingumui bei poreikiams, Cocomo modelis pradėjo nebetenkinti keliamų reikalavimų. Sistemos charakterizavimas vienu iš trijų (*Organic*, *Semi – detached*, *Embedded*) tipų tampa neveiksmingas ir neefektyvus.

Atsižvelgiant į šiuos trūkumus buvo atlikti patobulinimai, įtraukti nauji metodai ir sukurtos naujos modelio versijos. Kiekviena pritaikoma pagal skirtingus poreikius bei projekto etapuose.

Šis modelis skirtas vertinti sistemoms, kurioms keliami greito sukūrimo, patogios vartotojo sąsajos reikalavimai. Sistemos apimtis / dydis matuojamas objektiniais taškais, kurių aprašas pateiktas skyrelyje [2.5.1.3].

Reinžinerijos atveju būtina įvertinti ir tam tikrų komponentų panaudojimo galimybę. Tai atliekama pagrindinėje formulėje įvedant *Reuse* dydį, kuris parodo, kokią sistemos dalį (procentas) tikimasi panaudoti:

$$NOP = \frac{(ObjectPo\ int\ s) \times (100 - \% Re\ use)}{100}$$

Personalas, kuris kurs informacinę sistemą, įvertinamas įvedant PROD dydį:

15 lentelė. Dydis, įvertinantis programuotojo sugebėjimus.

Programuotojų patirtis, branda, sugebėjimai, pajėgumai	Labai žemas	Žemas	Normalus	Aukštas	Labai aukštas
PROD	4	7	10	25	50

Galutinė formulė:

$$PM = \frac{NOP}{PROD}$$

Šis modelis priskiriamas „Iš apačios į viršų“ vertinimo strategijai. Plačiau apie modelį pateikta šaltiniuose [3, 11, 14, 20].

2.5.2.5. COCOMO II : Ankstyvo projektavimo modelis

Šis modelis buvo išplėtotas iš Basic COCOMO modelio, papildant parametrais, kurių dėka vertinimo procesas tampa daug detalesnis ir tikslesnis. Vertinimo strategija - „Iš viršaus į apačią“.

Tarkime, projekto reikalavimai kuriamai informacinei sistemai IS yra sugrupuoti į *n* darbų-funkcijų:

$$\{f_1, f_2, f_3 \dots f_n\}$$

Darbo procesas:

- 1) Vertinama ir matuojama sistemos apimtis (*Size*): kiekviena f_i ($i \in \{1..n\}$) nagrinėjama, skaičiuojami nepritaikyti funkciniai taškai, jų reikšmės ir bendrai sumuojama.
- 2) Funkcinių taškų reikšmių suma konvertuojama į kodo eilutes.
- 3) Nustatomos 5 laipsnio parametrų reikšmės.
- 4) Nustatomos 7 projekto sąnaudų ir aplinkos veiksmų parametrų reikšmės.
- 5) Nustatoma kokia dalis programinės įrangos bus panaudota, koks panaudojimo produktyvumas.
- 6) Skaičiuojamos projekto sąnaudos nominaliais žmogaus darbinais mėnesiais (PM):

$$Effort = A \times [Size]^B \times \prod_{i=1}^7 EM_i + \frac{ASLOC(\frac{AT}{100})}{ATPROD}, \text{ kur}$$

$$B = 1.01 + 0.01 * \sum_{j=1}^5 SF_j, \text{ Size} = KSLOC + KASLOC \left(\frac{100 - AT}{100} \right)$$

- A – konstanta, lygi 2.5.
- Size – sistemos dydis, įvertintas tūkstančiais programinio kodo eilutėmis (*Source Lines Of Code*), kuris nustatoma naudojant 2.5.1.2 skyrelyje pateiktą funkcinių taškų metodo dalį. Identifikuojami nepritaikyti funkciniai taškai ir skaičiuojamos jų reikšmės pagal sudėtingumą. Skirtingai nei funkcinių taškų metode nėra skaičiuojamas VAF daugiklis ir sistemos charakteristikų įtaka.
- SF_j - 5 laipsnio parametrai (*Scale Drivers*):

16 lentelė. Laipsnio parametrai.

Laipsnio parametras	Komentaras
PREC	<i>Precedentedness</i> – Ar naujasis projektas palyginimas ir turi panašumu su ankščiau vykdytais projektais?
FLEX	<i>Development Flexibility</i> – Ar keliami reikalavimai nėra griežti, o lankstūs?
RESL	<i>Architecture/Risk Resolution</i> – Kiek išbaigta ir apibrėžta IS architektūra?
TEAM	<i>Team Cohesion</i> – Kaip apibūdintumėte santykius tarp projekto komandos narių?
PMAT	<i>Process Maturity</i> – Kokia įmonės branda projektų valdyme?

17 lentelė. Laipsnio parametrų reikšmių klasifikatorius.

Laipsnio parametras	Labai žemas	Žemas	Vidutinis	Virš vidutinio	Aukštas	Labai aukštas
PREC	4.05	3.24	2.43	1.62	0.81	0.00
FLEX	6.07	4.86	3.64	2.43	1.21	0.00
RESL	4.22	3.38	2.53	1.69	0.84	0.00
TEAM	4.94	3.95	2.97	1.98	0.99	0.00
PMAT	4.54	3.64	2.73	1.82	0.91	0.00

- EM_i – 7 projekto sąnaudų bei aplinkos veiksnių parametrai.

18 lentelė. Projekto sąnaudų bei aplinkos veiksnių parametrai.

Laipsnio parametras	Komentaras
RCPX	<i>Product Reliability and Complexity</i> - Produkto patikimumas ir sudėtingumas
RUSE	<i>Required Reusability</i> – Ar kuriamos IS komponentai bus naudojami reinžinerijoje?
PDIF	<i>Platform Difficulty</i> - Platformos sudėtingumas
PERS	<i>Personnel Capability</i> - Personalo faktoriai: analitikų sugebėjimai, programuotojų sugebėjimai, darbo vientisumas
PREX	<i>Personnel Experience</i> – Ar komandos nariai patyrę?
FCIL	<i>Facilities</i> - Produkto patogumas, naudojimo lengvumas
SCED	<i>Schedule</i> – Ar projekto kalendorinė trukmė forsuojama, t.y. mažinama?

19 lentelė. Projekto sąnaudų bei aplinkos veiksnių parametru reikšmių klasifikatorius.

Laipsnio parametras	Labai žemas	Žemas	Vidutinis	Virš vidutinio	Aukštas	Labai aukštas
RCPX	5,6	7,8	9-11	12	13-15	16-18
RUSE	0.00	0.91	1.00	1.14	1.29	1.49
PDIF	8	9	10-11	12-13	14-15	16-17
PERS	3,4	5,6	7,8	9	10,11	12-13
PREX	3,4	5,6	7,9	10,11	12,13	14,15
FCIL	2	3	4,6	7,8	9,10	11
SCED	1.29	1.10	1.00	1.00	1.00	0.00

- ASLOC – pritaikytų komponentų dydis, išreikštas
 - AT – procentas komponentų, kurie bus gauti naudojantis automatiniais kodo generatoriais.
 - ATPROD – automatinių kodo generatorių produktyvumas.
- 7) Skaičiuojama projekto kalendorinė trukmė:

$$\text{Schedule} = (3.0 * PM^{(0.33+0.2*(B-1.01))}) * \frac{SCED\%}{100}$$

Kiekvienam projektui atskirai parenkamos šių parametru reikšmės, kurių reikšmės priklausomai nuo projekto gali būti nustatomos vieną kartą arba daugiau kartų. Detalesnis aprašas yra COCOMO II modelių apraše [3].

Plačiau apie modelį pateikta šaltiniuose [3, 11,14, 20].

2.5.2.6. COCOMO II : Vėlyvo projektavimo modelis

Šis modelis skirtas vėlesniems projekto etapams, kada sukaupiama daugiau informacijos apie kuriamą IS. Vertinimo proceso etapai ir skaičiavimo formulės tokios pat kaip ir COCOMO II ankstyvo projektavimo modelyje. Skirtumai:

- Kiekvienos f_i dydis / apimtis iš karto matuojamas fizinėmis kodo eilutėmis, išskyrus dar nerealizuotus komponentus, kurie gali būti matuojami ir funkciniais taškais (vėliau konvertuojant į *SLOC*). Fizinių eilučių klasifikavimo lentelę sudarė *Software Engineering Institute*, kuri pateikta literatūros šaltinyje [17].
- Sąnaudų parametrai papildomi dar 10-čia (viso 17).

20 lentelė. Projekto sąnaudų bei aplinkos veiksnų parametrai.

Laipsnio parametras	Komentaras
RELY	<i>Required Reliability</i> – Kokie IS užstrigimo/klaidos padariniai?
DATA	<i>Database Size</i> – Kiek reikalinga duomenų, testuojant IS?
CPLX	<i>Product Complexity</i> – Kiek sudėtinga kuriama IS bus?
RUSE	<i>Required Reusability</i> – Ar kuriamos IS komponentai bus naudojami reinžinerijoje?
DOCU	<i>Documentation</i> – Kiek dokumentacijos planuojama sukurti?
TIME	<i>Execution Time</i> – Kiek CPU resursų naudos kuriama IS?
STOR	<i>Main Storame</i> – Kiek atminties resursų naudos kuriama IS?
PVOL	<i>Platform Volatility</i> – Kaip dažnai operacinė sistema / duomenų bazė bus keičiama?
ACAP	<i>Analyst Capability</i> – Kiek pajėgūs yra projekto analitikai?
PCAP	<i>Programmer Capability</i> – Kiek pajėgūs yra projekto programuotojai?
PCON	<i>Personnel Continuity</i> – Kokia metinė organizacijos apyvarta?
AEXP	<i>Applications Experience</i> – Kiek patirties su tokio pobūdžio IS turi komanda?
PEXP	<i>Platform Experience</i> – Kiek patirties turi komanda su operacine sistema / platforma, kurioje bus IS?
LTEX	<i>Language and Tool Experience</i> – Kiek patirties turi komanda su reikiama programavimo kalba ir įrankiais?
TOOL	<i>Use of Software Tools</i> – Kokio profesionalumo ir patogumo įrankiai bus naudojami?
SITE	<i>Multisite Development</i> – Ar projekto komanda fiziškai randasi vienoje vietoje? Kaip jie komunikuoja?
SCED	<i>Schedule</i> – Ar projekto kalendorinė trukmė forsuojama, t.y. mažinama?

21 lentelė. Projekto sąnaudų bei aplinkos veiksnų parametų reikšmių klasifikatorius.

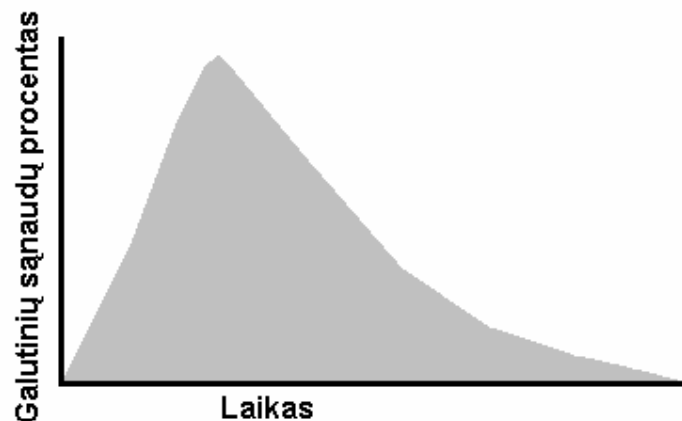
Parametras	Parametų reikšmės					
	Labai žemas	Žemas	Vidutinis	Virš vidutinio	Aukštas	Labai aukštas
RELY	0.75	0.88	1.00	1.15	1.39	0.00
DATA	0.00	0.93	1.00	1.09	1.19	0.00
CPLX	0.75	0.88	1.00	1.15	1.30	1.66
RUSE	0.00	0.91	1.00	1.14	1.29	1.49
DOCU	0.89	0.95	1.00	1.06	1.13	0.00
TIME	0.00	0.00	1.00	1.11	1.31	1.67
STOR	0.00	0.00	1.00	1.06	1.21	1.57
PVOL	0.00	0.87	1.00	1.15	1.30	0.00
ACAP	1.50	1.22	1.00	0.83	0.67	0.00

Parametras	Parametrų reikšmės					
	Labai žemas	Žemas	Vidutinis	Virš vidutinio	Aukštas	Labai aukštas
PCAP	1.37	1.16	1.00	0.87	0.74	0.00
PCON	1.24	1.10	1.00	0.92	0.84	0.00
AEXP	1.22	1.10	1.00	0.89	0.81	0.00
PEXP	1.25	1.12	1.00	0.88	0.81	0.00
LTEX	1.22	1.10	1.00	0.91	0.84	0.00
TOOL	1.24	1.12	1.00	0.86	0.72	0.00
SITE	1.25	1.10	1.00	0.92	0.84	0.78
SCED	1.29	1.10	1.00	1.00	1.00	0.00

Plačiau apie modelį pateikta šaltiniuose [3, 11,14, 20].

2.5.2.7. PUTNAM modelis

Šis modelis skirtas didelės apimties projektams, kurių apimtis viršija 70000 kodo eilučių. PUTNAM modelis pagrįstas eksperimentiniais rezultatais, kad projekto sąnaudos (*Effort*) pasiskirsto panašiai kaip *Norden/Rayleigh* kreivės integralas:



2 pav. Norden/Rayleigh kreivė.

Teigiama, jog žmonių skaičius projektui išibėgėjant auga ir krenta testavimo, eksploatacijos ir priėmimo etapais. Projekto sąnaudos apskaičiuojamos pagal formules:

$$\text{Effort} = 12^5 * B \left(\frac{SLOC}{P} \right)^3 * \frac{1}{\text{Schedule}^4}$$

arba

$$\text{Effort} = 56.4 * B \left(\frac{SLOC}{P} \right)^{\frac{9}{7}}$$

SLOC – kodo eilučių skaičius,

B – projekto dydžio koeficientas,

P – produktyvumo koeficientas,

Schedule – planuojama projekto trukmė mėnesiais.

22 lentelė. Galimų produktyvumo koeficientų klasifikatorius.

Nr	Produktyvumo parametras P	IS Tipas
1	987	Mikrokodas
2	1597	Mikroschemos (ROM)
3	1974	Realaus laiko sistemos, aviacijos elektronika
4	3194	Radarų sistemos
5	4181	Komandų valdymo pultai
6	5186	Proceso valdymas
7	8362	Telekomunikacijos
8	13530	Mokslinės sistemos
9	28657	Verslo sistemos

23 lentelė. Projekto dydžių koeficientų klasifikatorius.

Dydis / apimtis (SLOC)	B
5-15K	0.16
20K	0.18
30K	0.28
40K	0.34
50K	0.37
>70K	0.39

Projekto trukmė apskaičiuojama pagal formulę:

$$\text{Schedule} = 8.14 * \left(\frac{\text{SLOC}}{P}\right)^{\frac{3}{7}}$$

PASTABA. Ši lygybė tinka tik tada, kai projekto trukmė bent 6 mėnesiai.

Plačiau apie metodą pateikta šaltiniuose [11,14,20].

2.5.2.8. Sistemine analize orientuotas metodas

Šis metodas įvertina turimą dokumentaciją, remiasi analizės etapo diagramomis ir vertina sistemą pagal:

- Duomenų struktūras – esybes, kuriose duomenys bus saugomi, esybių atributus ir ryšius tarp skirtingų esybių. Pagrindinis šaltinis šiai analizei yra loginė duomenų struktūra (LDS).
- Procesai – duomenų srautai sistemoje, kaip ir kurie procesai tuos duomenis modifikuoja, kuriose duomenų saugyklose yra saugomi duomenys. Pagrindinis šaltinis šiai analizei yra duomenų srautų diagramos (DSD).

- Transformacijos – kaip skirtingų esybių atributai yra modifikuojami per jų gyvavimo ciklą. Todėl yra analizuojamas tam tikras kiekis sekų, kuriose atsiranda šie atributų pokyčiai bei jų pradinės sąlygos. Visa tai yra pateikiama esybių gyvavimo istorijos diagramose.

Pradžioje yra įvertinamos duomenų struktūros ir procesai, o po to transformacijos. Laikinis įvertis skaičiuojamas pagal formulę :

$$\text{Laikas} = (\text{parametro reikšmė}) * (\text{dienų skaičius, skirtas šiam parametru}) .$$

Galimi parametrai:

- A_1 – aukščiausio lygio DSD procesų skaičius,
- A_2 – loginėje duomenų struktūroje esančių esybių skaičius,
- A_3 – sistemoje egzistuojančių pagrindinių dokumentų skaičius,
- A_4 – egzistuojančių „projekto duobių“ skaičius,

Deja, ne visada yra skiriama pakankamai laiko išsamiai ir detaliam atlikti analizę, todėl žemesnio lygio, detalesnių diagramų struktūra nustatoma statistiškai:

- A_5 – būsimų interviu dėl neaiškumų („projekto duobių“) skaičius, $A_4 * 2$
- A_6 – būsimų interviu dėl sistemos palaikymo skaičius, $A_5 * 0.5$
- A_7 – antro lygio DSD procesų skaičius, $A_1 * 2.5$
- A_8 – trečio lygio DSD procesų skaičius, $A_7 * 0.75$
- d – dienų skaičius.

$$\text{Effort} = \sum_{i=1}^8 (A_i \times d)$$

Trukmė formaliai apskaičiuojama sąnaudas dalijant iš projekte dirbančių žmonių skaičiaus:

$$\text{Schedule} = \left(\frac{\text{Effort}}{PM} \right)$$

Plačiau apie metodą pateikta šaltiniuose [6,7].

2.5.3. Laikinių įverčių skaičiavimo metodų palyginimas

Aptarti projekto laikinių įverčių skaičiavimo metodai sistemos dydžiui / apimčiai vertinti naudoja skirtingus darbų – funkcijų apimties vertinimo metodus, vertinimo strategijas, formules. Realiam pasaulyje jų panaudojimas yra nuolat kombinuojamas, t.y. tam tikriems projektams ar net jo dalims gali būti naudojama viena metodika, kitiems kita.

Dauguma metodų yra ir praktiškai realizuoti bei naudojami. Šiuo metu egzistuoja daug projekto darbų apimties bei trukmės - laikinių įverčių skaičiavimo programų. Kai kurios iš jų naudoja standartinius metodus bei modelius, kai kurios remiasi tiesiog eksperimentais tyrimais. Kiekviena iš šių programų skirtingais aspektais padengia laikinių įverčių skaičiavimo poreikį. Programų, naudojamų metodų bei organizacijų trumpa apžvalga pateikta 3 priede.

Žemiau pateiktos metodų palyginimo suvestinės, kurios atskleidžia bendras metodų savybes, privalumus, trūkumus, naudojamas lygtis, dydžio metodus. Naudojamos sąvokos ir terminai:

- Nominalių žmogaus mėnesių skaičius (*Nominal person months*) – mėnesių skaičius, per kurį bus realizuotas projektas dirbant vienam žmogui.
- Dydis/apimtis – (*Size*) projekte kuriamos/modifikuojamos informacinės sistemos apimtis, kuri gali būti išreikšta:
 - SLOC – (*Source Lines Of Code*) kodo eilutėmis.
 - FP – (*Function Points*) funkciniais taškais.
 - OP – (*Object Points*) objektiniais taškais.
- Vertinimo strategija:
 - *Top-down* – „iš viršaus į apačią“ strategija vertina informacinę sistemą pagal charakteristikas.
 - *Bottom-up* – „iš apačios į viršų“ strategija vertina informacinės sistemos kiekviena modulį atskirai, po to juos apjungiant.
- Sąnaudos – (*Effort*) – dydis adekvatus nominalių žmogaus mėnesių skaičiui.
- Kalendorinė trukmė – (*Schedule*) projekto kalendorinė trukmė mėnesiais.
- KSLOC – (*Thousands Source Lines of Code*) tūkstančiai fizinių kodo eilučių.

24 lentelė. Metodų palyginimo pagal skaičiavimo charakteristikas suvestinė.

Metodas	Nominalių žmogaus mėnesių skaičius	Trukmė	Dydis/Apimtis		Vertinimo strategija
			SLOC	FP	
Analoginis vertinimo metodas	-	$Schedule = \sum ana \log iniai_ iverciai$	-	-	Top-down, bottom-up
Ekspertinis vertinimo metodas	-	$Schedule = \sum Ekspertu_ iverciai$	-	-	Top-down, bottom-up
Basic COCOMO modelis	$Effort = a \times (KSLOC)^b$	$Schedule = 2.5 \times Effort^c$	+	+	Top-down, bottom-up
COCOMO II : Aplikacinis komponavimo modelis	$Effort = \frac{\sum_{i=1}^1 OP_i \times (100 - \%Reuse)}{100 * PROD}$	$Schedule = (3.0 * PM^{(0.33+0.2*(B-1.01))}) * \frac{SCED\%}{100}$	-	-	Bottom-up
COCOMO II : Ankstyvo projektavimo modelis	$Effort = A \times [Size]^B \times \prod_{i=1}^7 EM_i + \frac{ASLOC(\frac{AT}{100})}{ATPROD}$	$Schedule = (3.0 * PM^{(0.33+0.2*(B-1.01))}) * \frac{SCED\%}{100}$	+	+	Top-down, bottom-up
COCOMO II : Vėlyvo projektavimo modelis	$Effort = A \times [Size]^B \times \prod_{i=1}^{17} EM_i + \frac{ASLOC(\frac{AT}{100})}{ATPROD}$	$Schedule = (3.0 * PM^{(0.33+0.2*(B-1.01))}) * \frac{SCED\%}{100}$	+	+	Top-down, bottom-up
PUTNAM modelis	$Effort = 12^5 * B(\frac{SLOC}{P})^3 * \frac{1}{Schedule^4}$ arba $Effort = 56.4 * B(\frac{SLOC}{P})^9$	$Schedule = 8.14 * (\frac{SLOC}{P})^{\frac{3}{7}}$	+	+	Top-down, bottom-up
Sisteminė analize orientuotas metodas	$Effort = \sum_{i=1}^8 (A_i \times d)$	$Schedule = (\frac{Effort}{PM})$	-	-	Bottom-up

25 lentelė. Metodų privalumų ir trūkumų suvestinė.

Metodas	Privalumai	Trūkumai
Analoginis vertinimo metodas	<p>1) Vertinimas yra grindžiamas realiai atlikto projekto duomenimis.</p> <p>2) Mažos įvertinimo sąnaudos, nes vertinimą realiai gali atlikti vienas žmogus – projektų vadovas, remdamasis praeities patirtimi ir žiniomis ir per daug nesileisdamas į detalius skaičiavimus.</p>	<p>1) Naudojant šį metodą reikia apsispręsti kaip tiksliausiai apibūdinti projektus. Identišku projektų nebūna, todėl netgi vienuodų funkcijų sugretinimas nereiškia, kad jos ekvivalenčios.</p> <p>2) Rizika iškreipti įvertinimą tų naujo projekto charakteristikų, kurios nėra analogų.</p> <p>3) Netgi parinkus „panašaus“ įvykdyto projekto laikinius įverčius nereiškia, kad projektas bus atliktas laiku. Tai gali sąlygoti daugelis veiksnių, tarp jų pasikeitęs darbinis personalas, reikalavimų pasikeitimas projekto eigoje ir t.t.</p>
Ekspertinis vertinimo metodas	<p>1) Ekspertai yra savo srities žinovai ir gali lyginti bei diskutuoti dėl skirtumų tarp įvykdytų projektų ir planuojamo įvykdyti projekto.</p> <p>2) Ekspertai gali diskutuoti ir prognozuoti dėl to, kaip naują projektą paveiks naujos technologijos, architektūros, taikomosios programos, programavimo kalbos, personalo pasikeitimai.</p>	<p>1) Naudojant šį metodą neįmanoma tiesiogiai ką nors išmatuoti ar apskaičiuoti skaičiais.</p> <p>2) Sunku vienareikšmiškai apibendrinti ar dokumentuoti ekspertų ar ekspertų grupės naudojamus veiksnius.</p> <p>3) Ekspertai tam tikrais klausimais gali būti per daug šališki, optimistiški, pesimistiški.</p>
Basic COCOMO modelis	<p>1) Paprastas naudoti.</p> <p>2) Gana tikslūs įvertinimai mažuose projektuose (žmonių skaičius), kurie realizuojami pažyastuose aplinkose (panašūs projektai praityje)</p>	<p>1) Ankstyvose projekto fazėse yra sunku įvertinti informacinės sistemos dydį/apimtį, todėl šis dydis gaunamas su didele paklaida.</p> <p>2) Nėra įvertinami daugelis aplinkos veiksnių tokių kaip duomenų bazės dydis, analitikų bei programuotojų pajėgumai, patirtis bei naudojami įrankiai.</p>
COCOMO II : Aplikacinis komponavimo modelis	<p>1) Ypač tinkamas ankstyvuose projekto etapuose, kada mažai turima informacijos apie architektūrinius sprendimus.</p> <p>2) Ypač tinkamas projektams, kuriuose naudojami RAD (<i>Rapid Application Development</i>) įrankiai (grafinės vartotojo sąsajos programavimo terpės).</p>	<p>1) Šis metodas naudoja „iš apačios į viršų“, vertinimo strategija, t.y. vertinimui reikalingi duomenys yra funkcijų sąrašas. Jeigu projekto pradžioje funkcijų sąrašo nėra, o yra tik bendros charakteristikos tai vertinimo procesas tampa komplikuoatas.</p>
COCOMO II : Ankstyvo projektavimo modelis	<p>1) Naudojant šį metodą ankstyvuose projekto etapuose prieš sukuriant projekto architektūrą jau yra atliekami nuodugnūs skaičiavimai, įvertinant daugelį aplinkos veiksnių.</p> <p>2) Tam tikrų parametrų reikšmės nesikeičia ir nereikia pakartotinai skaičiuoti, jeigu aplinka nuo praėito projekto nepakito / mažai pakito.</p>	<p>1) Reikalingi duomenys parametrams, kurie įmonėje gali būti nekaupiami.</p> <p>2) Skaičiavimai reikalauja daugiau sąnaudų.</p> <p>3) Nerekomenduojamas mažos apimties projektams dėl skaičiavimo sąnaudų.</p>
COCOMO II : Vėlyvo	<p>1) Skaičiavimo rezultatai gana patikimi vėlyvuose projekto etapuose po architektūros sukūrimo.</p>	<p>1) Reikalingi duomenys parametrams, kurie dažniausiai įmonėje nėra kaupiami.</p>

projektavimo modelis	<p>2) Iš visų COCOMO II modelių pats tiksliausias modelis po to, kai yra sukuriama architektūra.</p> <p>3) Tam tikrų parametrų reikšmės nesikeičia ir nereikia pakartotinai skaičiuoti, jeigu aplinka nuo praeito projekto nepakito / mažai pakito.</p>	<p>2) Skaičiavimas reikalauja daugiau sąnaudų.</p> <p>3) Dėl skaičiavimo sąnaudų nerekomenduojamas mažos apimties projektams.</p>
PUTNAM modelis	<p>1) Paprastas naudoti, nes pateikia baigtines parametrų aibes.</p> <p>2) Rezultatai yra su maža paklaida tuomet kai yra didėlės apimties projektai.</p> <p>3) Yra įvertinamas produktyvumo faktorius.</p>	<p>1) Šis modelis yra jautrus projekto trukmei : trumpinant ją žymiai padidėja nominalių žmogaus mėnesių skaičius, reikalingas sukurti informacinę sistemą.</p> <p>2) Rezultato teisingumas labai priklauso nuo tiksliai pateikiamų fizinių kodo eilučių ar loginių kodo sakinių skaičių. Kuo netikslesni įvertinami ir apskaičiuojami SLOC arba LSS, tuo netikslesnis ir rezultatas.</p> <p>3) Netinkamas mažos apimties projektams, nes pervertina laiką, t.y. per daug apskaičiuoja.</p>
Sistemine analize orientuotas metodas	<p>1) Pirminės inžinerijos metu šis metodas naudojamas vėlesniuose etapuose, t.y. po analizės etapo, kada jau pakankamai surinkta bei specifikuota reikalavimų.</p> <p>2) Šis metodas ypač tinkamas reinžinerijos metu ir kada pakeitimai yra ne esminiai. Tuomet pirminio projekto dokumentacija yra ypač naudinga bei informatyvi net jei neturima programinio kodo.</p>	<p>1) Nenaudoja standartinių darbų – funkcijų apimties skaičiavimo metodų.</p> <p>2) Labai dažnai projekto pradžioje turima per mažai dokumentacijos arba ji pasenusi arba jos visai nėra.</p> <p>3) Metodas neefektyvus pirminės inžinerijos ankstyvuose projekto etapuose, t.y. reikalavimų nustatymo ir analizės etape.</p> <p>4) Labai retai kaupiama informacija apie transformacijas, t.y. kaip skirtingų esybių atributai yra modifikuojami per jų gyvavimo ciklą.</p> <p>5) Nėra įvertinami daugelis aplinkos veiksnių tokių kaip duomenų bazės dydis, analitikų bei programuotojų pajėgumai, patirtis bei naudojami įrankiai.</p>

2.6. Analizės išvados

- 1) Egzistuoja standartiniai plačiai naudojami apimties vertinimo metodai. Atlikus literatūros šaltinių analizę pastebėta, kad plačiausiai naudojami trys informacinės sistemos programinės įrangos kūrimo darbų apimties įvertinimo metodai - funkcinių taškų, objektinių taškų ir kodo eilučių metodai.
- 2) Dauguma plačiai naudojamų laikinių įverčių skaičiavimo metodų naudoja dydžius *Effort* ir *Schedule*. Projekto sąnaudos (*Effort*) vertinamos nominalių žmogaus darbinių mėnesių dydžiu (*PM – Person Month*), kuriuo remiasi viso projekto trukmės (*Schedule*) dydžio skaičiavimas.
- 3) Dauguma plačiai naudojamų laikinių įverčių skaičiavimo metodų naudoja standartinius apimties vertinimo metodus. Išanalizavus esamus populiarius laikinių įverčių metodus bei modelius pastebėta, kad kiekvienas iš metodų pateikia laikinius įverčius pagal tam tikras skirtingas skaičiavimo formules, tačiau dauguma iš jų naudoja tuos pačius darbų - funkcijų apimties skaičiavimo metodus – funkcinių taškų, objektinių taškų ir kodo eilučių metodus.
- 4) Egzistuoja galimybė derinti tam tikrų metodų panaudojimą tarpusavyje. Viena kartą apskaičiavus projekto darbų – funkcijų apimtį arba pagal funkcinių taškų arba pagal objektinių taškų arba pagal kodo eilučių metodą, ją galima naudoti skaičiuojant laikinius įverčius tiek Basic COCOMO, tiek COCOMO II Aplikacinio komponavimo, tiek COCOMO II Ankstyvo projektavimo, tiek COCOMO II Vėlyvo projektavimo, tiek PUTNAM modeliais.
- 5) Analoginio metodo tinkamumas. Nors analoginis laikinių įverčių skaičiavimo metodas nenaudoja standartinių apimties vertinimo metodų jis išlieka kaip plačiai naudojamas metodas, kurių vienas iš privalumų yra naudojimo paprastumas. Šis metodas tinkamas naudoti projekto reikalavimų, analizės bei projektavimo etapuose.
- 6) Ekspertinio metodo tinkamumas. Šis metodas taip pat yra plačiai naudojamas bei paprastas naudoti bei nenaudoja standartinių apimties vertinimo metodų. Šis metodas tinkamas naudoti projekto reikalavimų, analizės, projektavimo, realizavimo bei vėlesniuose etapuose.
- 7) Basic COCOMO modelio tinkamumas. Šis metodas tinkamas naudoti reikalavimų, analizės bei projektavimo etapuose.
- 8) COCOMO II : Aplikacinis komponavimo modelio tinkamumas. Šis metodas tinkamas naudoti reikalavimų, analizės, projektavimo, realizavimo bei vėlesniuose etapuose.
- 9) COCOMO II : Ankstyvo projektavimo modelio tinkamumas. Šis metodas tinkamas naudoti projektavimo etape.
- 10) COCOMO II : Vėlyvo projektavimo modelio tinkamumas. Šis metodas tinkamas naudoti realizavimo bei vėlesniuose etapuose.
- 11) PUTNAM modelio tinkamumas. Šis metodas tinkamas naudoti reikalavimų, analizės, projektavimo, realizavimo bei vėlesniuose etapuose.

- 12) Sisteminė analize orientuoto metodo netinkamumas. Šis metodas yra netinkamas dėlto, kad apimtį matuoja pagal dokumentaciją (kurios gali būti per mažai, ji pasenusi arba jos visai nėra), o ypač pagal transformacijas – kas retai yra atliekama ir dokumentuojama IT įmonėse.

3. INFORMACINĖS SISTEMOS PROGRAMINĖS ĮRANGOS KŪRIMO DARBŲ APIMTIES ĮVERTINIMO MODELIS

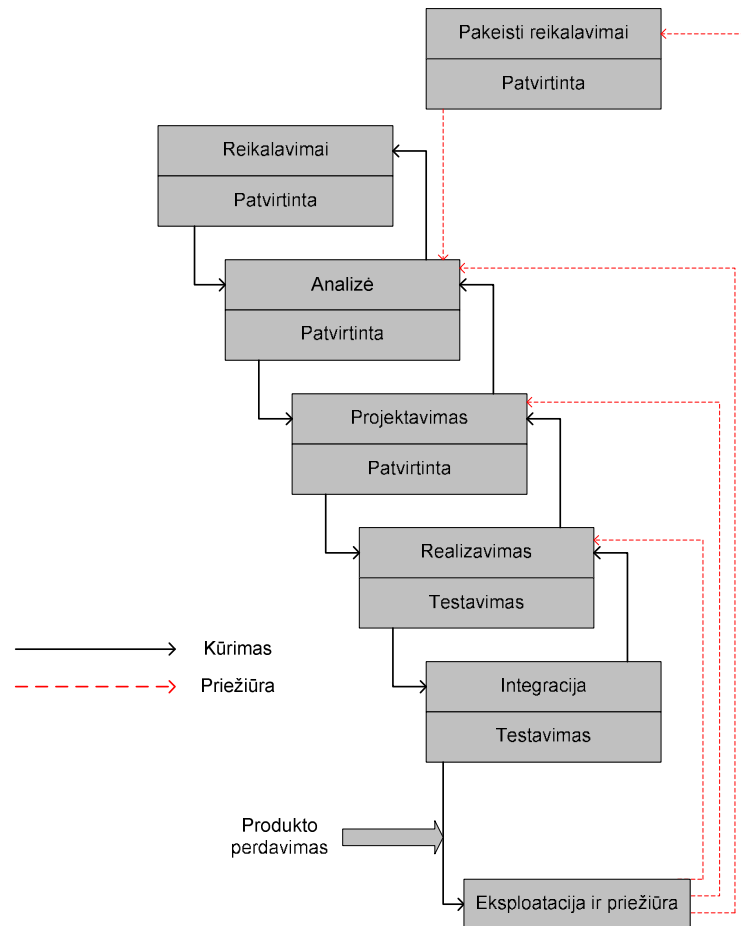
3.1. VIVEMO-7 modelio pagrindimas ir esmės išdėstymas

Analizuojant esamus plačiai naudojamus trukmės vertinimo metodus bei modelius [23] buvo pastebėta, kad kiekvienas iš metodų pateikia laikinius įverčius pagal tam tikras skaičiavimo formules, tačiau dauguma iš jų naudoja tuos pačius darbų - funkcijų apimties skaičiavimo metodus – funkcinių taškų, objektinių taškų ir kodo eilučių metodus. Logiška manyti, kad kartą apskaičiavus apimtį ją galima naudoti skirtinguose projekto laikinių įverčių skaičiavimo metoduose be didelių papildomų sąnaudų. Todėl informacinės sistemos programinės įrangos įvertinimo modelyje naudojami tie metodai, kurie naudoja standartinius apimties skaičiavimo metodus (funkcinių taškų, objektinių taškų, kodo eilučių skaičiavimo metodai) arba kurie iš karto pateikia laikinius įverčius.

Apskaičiavus projekto trukmę pagal konkretų metodą ar modelį, nėra garantijos, kad laikinis įvertis patikimas. Juolab, kad nei vienas iš projekto laikinių įverčių skaičiavimo metodų nevaldo rizikos. Šiuos trūkumus bandoma spręsti įtraukiant Bob Hughes ir Mike Cotterell [6] tikimybių grafiką, kurio dėka galima įvertinti laikinių įverčių tikimybes

Kiekvienas projektas vyksta etapais. Informacinės sistemos programinės įrangos kūrimo darbų apimties įvertinimo modelyje pritaikytas dažniausiai naudojamas klasikinis Krioklio [1] projekto modelis. Kiekviename etape panaudojami atitinkami apimties bei darbų – funkcijų bei projekto laikinių įverčių skaičiavimo metodai, sudaromi darbų – funkcijų laikinių įverčių aibės, apskaičiuojamos optimistinės, pesimistinės, labiausiai tikėtinos, tikėtinos reikšmės bei apskaičiuojamos tikimybės. Rezultate sudaroma galutinė viso projekto laikinio įverčio bei tikimybės suvestinė.

Kuriamam modeliui suteiksime trumpą pavadinimą – „VIVEMO-7“ (VISapusiškas VERTinimo MOdelis, 7 – nes naudoja 7 projekto trukmės įvertinimo metodus bei modelius).



3 pav. IS projekto gyvavimo ciklo etapai.

Projekto planavime ir darbų trukmės skaičiavime galima išskirti tokius žingsnius / etapus :

1. Projekto trukmės skaičiavimas / įvertinimas ankstyvuose projekto etapuose.
2. Projekto trukmės / įvertinimo tikslinimas projekto etapuose.
3. Faktinis įvertis po projekto pabaigos.

3.2. VIVEMO-7 modeliui reikalingų duomenų sudėtis

Tarkime, turime projektą kurti informacinę sistemą IS, sudarytą iš n funkcijų:

$$\{f_1, f_2, f_3 \dots f_n\}$$

Kiekvienai iš funkcijų f_i ($i \in \{1..n\}$) sudarome galimų laikinių įverčių aibę, panaudojant tinkamus vertinimo metodus:

$$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}, \text{ kur}$$

x_1 – Analoginiu vertinimo metodu gautas laikinis įvertis.

x_2 – Ekspertiniu vertinimo metodu gautas laikinis įvertis.

x_3 – Basic COCOMO modeliu gautas laikinis įvertis.

x_4 – PUTNAM modeliu gautas laikinis įvertis.

x_5 – COCOMO II Aplikaciniu komponavimo gautas modeliu laikinis įvertis.

x_6 – COCOMO II Ankstyvo projektavimo modeliu gautas laikinis įvertis.

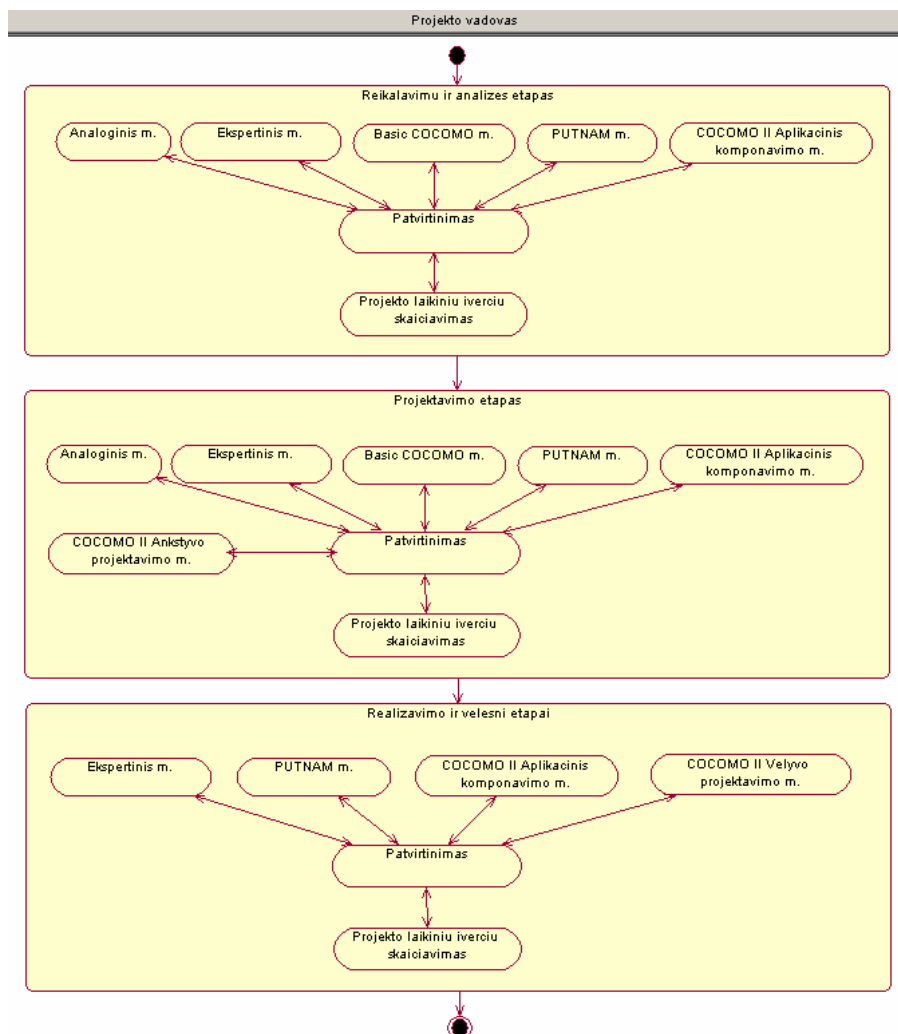
x_7 – COCOMO II Vėlyvo projektavimo modeliu gautas laikinis įvertis.

PASTABA1. Ne kiekvienai funkcijai ir ne kiekviename projekto etape galima sudaryti pilną laikinių įverčių aibę. Dėl to, ne visos f-jos turi pilną laikinių įverčių aibę.

PASTABA2. Kiekviena laikinio įverčio reikšmė pateikiama mėnesiais.

3.3. Darbo su VIVEMO-7 modeliu procesas

Žemiau pateikiama VIVEMO-7 modelio darbo schema:



4 pav. VIVEMO-7 darbo procesas.

3.4. VIVEMO-7 modelio taikymo etapai

3.4.1. Projekto reikalavimų ir analizės etapų įvertinimas

Ankstyvuose projekto etapuose naudojame:

1. Analoginį skaičiavimo metodą. Atliekame žingsnius:
 - 1.1 Pasirenkame f_i ($i \in \{1..n\}$) f-ja.
 - 1.2 Ieškome panašių praeityje buvusių projektų, panašios funkcijos faktinių įverčių.
 - 1.3 Jeigu randame, papildome f-jos f_i aibę - $\{x_1\}$, jei ne - kartojame tuos pačius žingsnius su kita f-ja.
 - 1.4 Baigiame kai visos f-jos pereitos.

Rezultatas:

$$f_1\{x_1\}, f_2\{x_1\}, f_3\{x_1\}, \dots, f_n\{x_1\}$$

2. Ekspertinį skaičiavimo metodą. Atliekame žingsnius:
 - 2.1 Organizuojame ekspertų susirinkimą.
 - 2.2 Kiekvienas ekspertas kiekvienai f_i ($i \in \{1..n\}$) įvertina ir pateikia savo laikinį įvertį.
 - 2.3 Visų ekspertų rezultatai peržiūrimi bendrai ir po diskusijos kiekvienai f_i priskiriamas suderintas laikinis įvertis.

Rezultatas:

$$f_1\{x_1, x_2\}, f_2\{x_1, x_2\}, f_3\{x_1, x_2\}, \dots, f_n\{x_1, x_2\}$$

3. Basic COCOMO modelį. Atliekame žingsnius :
 - 3.1 Kiekvienai f_i ($i \in \{1..n\}$) nustatome apimtį $KSLOC(f_i)$. Jeigu tai reinžinerijos projektas, skaičiuojame iš karto KSLOC vienetais, jeigu pirminė inžinerija – nepritaikytus FP, kuriuos verčiame į KSLOC.
 - 3.2 Pagal sistemos tipą pasirenkame vieną iš trijų režimų ir a, b reikšmes (13 lentelė.)
 - 3.3 Apskaičiuojame viso projekto sąnaudas:

$$Effort = a \times (KSLOC(f_1) + KSLOC(f_2) + \dots + KSLOC(f_n))^b$$

- 3.4 Apskaičiuojame viso projekto trukmę, parinkę c koeficiento reikšmę (14 lentelė. :

$$Schedule = 2.5 \times Effort^c$$

- 3.5 Kiekvienai f_i pagal apimtį $KSLOC(f_i)$ apskaičiuojame svorį m_i :

$$m_i = \frac{KSLOC(f_1) + KSLOC(f_2) + \dots + KSLOC(f_n)}{KSLOC(f_i)}$$

3.6 Kiekvienai f_i apskaičiuojame laikinį įvertį x_3 :

$$x_3 = Schedule * m_i$$

3.7 Baigiame kai visoms f-joms priskirti laikiniai įverčiai

Rezultatas:

$$f_1\{x_1, x_2, x_3\}, f_2\{x_1, x_2, x_3\}, f_3\{x_1, x_2, x_3\} \dots f_n\{x_1, x_2, x_3\}$$

4. PUTNAM modelį.

4.1 Kiekvienai f_i ($i \in \{1..n\}$) nustatome apimtį $SLOC(f_i)$. Jeigu tai reinžinerijos projektas, skaičiuojame iš karto SLOC vienetais, jeigu pirminė inžinerija – nepritaikytus FP, kuriuos verčiame į SLOC pagal konvertavimą (9 lentelė. .

4.2 Pagal apimties dydį parenkame projekto dydžio koeficientą (23 lentelė.).

4.3 Pagal kuriamos IS tipą parenkame produktyvumo parametą P (22 lentelė.)

4.4 Apskaičiuojame viso projekto sąnaudas:

$$Effort = 56.4 * B \left(\frac{SLOC}{P} \right)^{\frac{9}{7}}$$

4.5 Apskaičiuojame viso projekto trukmę:

$$Schedule = \frac{Effort}{k}$$

k – projekto žmonių skaičius.

4.6 Kiekvienai f_i pagal apimtį $SLOC(f_i)$ apskaičiuojame svorį m_i :

$$m_i = \frac{SLOC(f_1) + SLOC(f_2) + \dots + SLOC(f_n)}{SLOC(f_i)}$$

4.7 Kiekvienai f_i apskaičiuojame laikinį įvertį x_4 :

$$x_4 = Schedule * m_i$$

4.8 Baigiame kai visoms f-joms priskirti laikiniai įverčiai

Rezultatas:

$$f_1\{x_1, x_2, x_3, x_4\}, f_2\{x_1, x_2, x_3, x_4\}, f_3\{x_1, x_2, x_3, x_4\} \dots f_n\{x_1, x_2, x_3, x_4\}$$

5. COCOMO II aplikacinio komponavimo modelį.

5.1 Kiekvienai f_i skaičiuojame OP_i reikšmę. Bendru atveju $f_i \sim OP_i$. Kiekvienam OP_i nustatome programinio modulio tipą ir sudėtingumą (10 lentelė.) ir pagal tai parenkame OP_i reikšmę (11 lentelė.).

5.2 Apskaičiuojame projekto apimtį naujais objekciniais taškais (*New Object Points*):

$$NOP = \frac{(OP_1 + OP_2 + \dots + OP_n) \times (100 - \%Reuse)}{100}$$

5.3 Įvertiname ir parenkame PROD parametą (15 lentelė.).

5.4 Apskaičiuojame projekto sąnaudas:

$$PM = \frac{NOP}{PROD}$$

5.5 Apskaičiuojame projekto trukmę Schedule:

$$Schedule = (3.0 * PM^{(0.33+0.2*(B-1.01)}) * \frac{SCED\%}{100}$$

5.6 Kiekvienai f_i pagal OP_i apskaičiuojame svorį m_i :

$$m_i = \frac{OP_1 + OP_2 + \dots + OP_n}{OP_i}$$

5.7 Kiekvienai f_i apskaičiuojame laikinį įvertį x_5 :

$$x_5 = Schedule * m_i$$

5.8 Baigiame kai visoms f-joms priskirti laikiniai įverčiai

Rezultatas:

$$f_1\{x_1, x_2, x_3, x_4, x_5\}, f_2\{x_1, x_2, x_3, x_4, x_5\}, f_3\{x_1, x_2, x_3, x_4, x_5\} \dots f_n\{x_1, x_2, x_3, x_4, x_5\}$$

6. Kiekvienai f_i iš $\{x_1, x_2, x_3, x_4, x_5\}$ apskaičiuojame vidutinę reikšmę (vidurkį) - $laikas_{vidut}$, ilgiausią - $laikas_{pesimist}$ ir trumpiausią $laikas_{optimist}$ reikšmės. Kiekvienam darbui apskaičiuojamas tikėtinas laikas:

$$Laikas_i(f_i) = \frac{laikas_{optimist} + 4 \times laikas_{normal} + laikas_{pesimist}}{6}$$

Kiekvienai f_i nustatomas standartinis nuokrypis s_i :

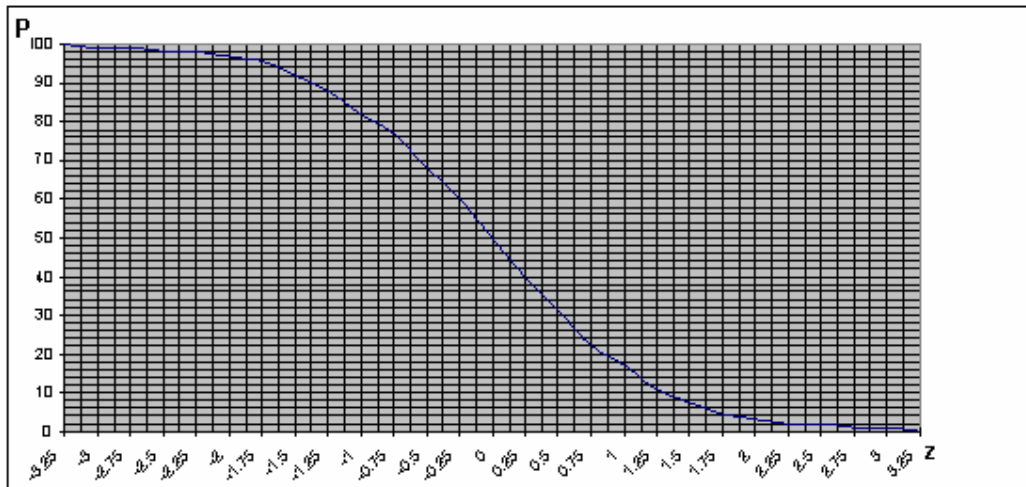
$$s_i = \frac{laikas_{optimist} - laikas_{pesimist}}{6}$$

Kiekvienai f_i skaičiuojame z_i dydį, kuris naudojamas nustatant įverčių neįvykimo tikimybę:

$$z_i = \frac{laikas_{planuoj} - laikas_{vidut}}{s_i}$$

$laikas_{planuoj}$ - planuojama darbo trukmė, kuri lygi tikėtina darbo trukmei - $Laikas_i(f_i)$.

Pagal z_i reikšmę randame darbo neįvykimo laiku tikimybę iš grafiko:



5 pav. Įvykio neįvykimo tikimybės pasiskirstymo pagal z dydį grafikas.

Sudaroma lentelė rezultatų lentelė pagal atliktus skaičiavimus.

26 lentelė. Laikinių įverčių rezultatų suvestinė.

Darbas f_i	Darbų - funkcijų trukmės mėnesiais				Standartinis nuokrypis s_i	Tikimybė, p_i
	Optimistinis $laikas_{optimist}$	Labiausiai tikėtinas $laikas_{vidut}$	Pesimistinis $laikas_{pesimist}$	Tikėtinas $Laikas_t(f_i)$		
f_1	$laikas_{optimist}(f_1)$	$laikas_{vidut}(f_1)$	$laikas_{pesimist}(f_1)$	$Laikas_t(f_1)$	$s_i(f_1)$	$p_1(f_1)$
f_2	$laikas_{optimist}(f_2)$	$laikas_{vidut}(f_2)$	$laikas_{pesimist}(f_2)$	$Laikas_t(f_2)$	$s_i(f_2)$	$p_2(f_2)$
...
f_n	$laikas_{optimist}(f_n)$	$laikas_{vidut}(f_n)$	$laikas_{pesimist}(f_n)$	$Laikas_t(f_n)$	$s_i(f_n)$	$p_n(f_n)$
Viso projekto tikimybė:						P

Viso projekto trukmė apskaičiuojama pagal:

$$Laikas_{total} = \sum_n^1 Laikas_t(f_i)$$

Projekto neįvykdymo tikimybė $P(z_{total})$ nustatoma pagal įvykio neįvykimo tikimybės pasiskirstymo pagal z dydį grafiką, o įvykdymo tikimybė P_{proj} pagal:

$$P_{proj} = 100\% - P(z_{total})$$

3.4.2. Projektavimo etapo įvertinimas

Šiame etape naudojame:

1. Analoginį vertinimo metodą. Seka 1.1. punkte.

Rezultatas:

$$f_1\{x_1\}, f_2\{x_1\}, f_3\{x_1\}, \dots, f_n\{x_1\}$$

2. Ekspertinį vertinimo metodą. Seka 1.2. punkte.

Rezultatas:

$$f_1\{x_1, x_2\}, f_2\{x_1, x_2\}, f_3\{x_1, x_2\}, \dots, f_n\{x_1, x_2\}$$

3. Basic COCOMO modelį. Seka 1.3 punkte.

Rezultatas:

$$f_1\{x_1, x_2, x_3\}, f_2\{x_1, x_2, x_3\}, f_3\{x_1, x_2, x_3\} \dots f_n\{x_1, x_2, x_3\}$$

4. PUTNAM modelį. Seka 1.4 punkte.

Rezultatas:

$$f_1\{x_1, x_2, x_3, x_4\}, f_2\{x_1, x_2, x_3, x_4\}, f_3\{x_1, x_2, x_3, x_4\} \dots f_n\{x_1, x_2, x_3, x_4\}$$

5. COCOMO II Aplikacinį komponavimo modelį. Seka 1.5 punkte.

Rezultatas:

$$f_1\{x_1, x_2, x_3, x_4, x_5\}, f_2\{x_1, x_2, x_3, x_4, x_5\}, f_3\{x_1, x_2, x_3, x_4, x_5\} \dots f_n\{x_1, x_2, x_3, x_4, x_5\}$$

6. COCOMO II Ankstyvo projektavimo modelį.

- 6.1 Kiekvienai f_i identifikuojame funkcinis taškus ir skaičiuojame apimtį nepritaikytais funkciniais taškais. Kiekvienam funkciniam taškui pagal sudėtingumą parenkame jo reikšmę FP_j ($j \in \{1..l\}$). Tarkim, kad f_i sudaryta iš vieno arba daugiau funkcinų taškų. Tada bendra f_i funkcinų taškų reikšmių suma:

$$Suma(f_i) = FP_1 + FP_2 + \dots + FP_j$$

Konvertuojame $Suma(f_i)$ į kodo eilutes SLOC(f_i) pagal programavimo kalbą [9 lentelė.]

- 6.2 Skaičiuojame viso projekto sąnaudas:

$$Effort = A \times [Size]^B \times \prod_{i=1}^7 EM_i + \frac{ASLOC\left(\frac{AT}{100}\right)}{ATPROD}$$

$$B = 1.01 + 0.01 * \sum_{j=1}^5 SF_j$$

$$Size = \sum_1^n \frac{KSLOC(f_i)}{1000} + KASLOC\left(\frac{100 - AT}{100}\right)$$

SF_j , EM_i , AT, ATPROD parametrus parenkame iš lentelių esančių skyrelyje [2.5.2.5].

6.3 Skaičiuojame projekto trukmę:

$$\text{Schedule} = (3.0 * PM^{(0.33+0.2*(B-1.01))}) * \frac{SCED\%}{100}$$

6.4 Kiekvienai f_i pagal $SLOC(f_i)$ apskaičiuojame svorį m_i :

$$m_i = \frac{KSLOC(f_1) + KSLOC(f_2) + \dots + KSLOC(f_n)}{KSLOC(f_i)}$$

6.5 Kiekvienai f_i apskaičiuojame laikinį įvertį x_6 :

$$x_6 = \text{Schedule} * m_i$$

6.6 Baigiame kai visoms f-joms priskirti laikiniai įverčiai

Rezultatas:

$$f_1\{x_1, x_2, x_3, x_4, x_5, x_6\}, f_2\{x_1, x_2, x_3, x_4, x_5, x_6\}, f_3\{x_1, x_2, x_3, x_4, x_5, x_6\}, \dots, f_n\{x_1, x_2, x_3, x_4, x_5, x_6\}$$

7. Sudarome rezultatų suvestinę pagal 6 punktą.

3.4.3. Projekto realizavimo ir vėlesnių etapų įvertinimas

Šiuose etapuose naudojame:

1 Ekspertinį vertinimo metodą, seka 1.2. punkte.

Rezultatas:

$$f_1\{x_2\}, f_2\{x_2\}, f_3\{x_2\}, \dots, f_n\{x_2\}$$

2 PUTNAM modelį. Seka 1.4 punkte.

Rezultatas:

$$f_1\{x_2, x_4\}, f_2\{x_2, x_4\}, f_3\{x_2, x_4\}, \dots, f_n\{x_2, x_4\}$$

3 COCOMO II Aplikacinį komponavimo modelį, seka 1.4

Rezultatas:

$$f_1\{x_2, x_4, x_5\}, f_2\{x_2, x_4, x_5\}, f_3\{x_2, x_4, x_5\}, \dots, f_n\{x_2, x_4, x_5\}$$

4 COCOMO II Vėlyvo projektavimo modelį. Etapai tokie patys kaip ir COCOMO II Ankstyvo projektavimo medelio tik prisideda EM_i daugiklių, kurių yra 17.

Rezultatas:

$$f_1\{x_2, x_4, x_5, x_7\}, f_2\{x_2, x_4, x_5, x_7\}, f_3\{x_2, x_4, x_5, x_7\}, \dots, f_n\{x_2, x_4, x_5, x_7\}$$

5 Sudarome rezultatų suvestinę pagal 6 punktą.

3.5. Informacinės sistemos programinės įrangos darbų apimtys įvertinimo modelio realizacija

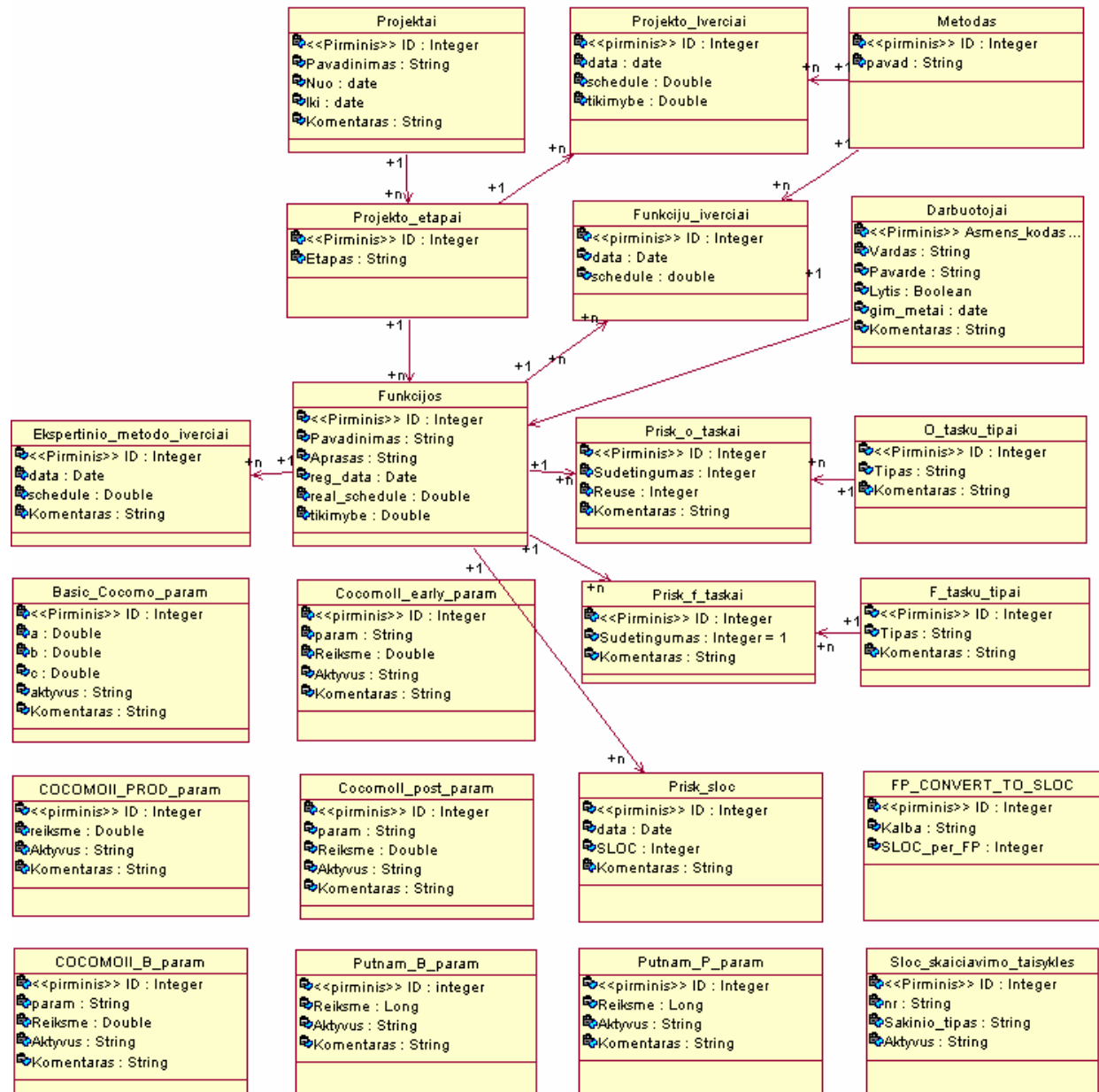
3.5.1. Galimų įgyvendinimo priemonių variantų analizė

Šio modelio realizavimas nereikalauja sudėtingų technologinių sprendimų. Prototipui realizuoti reikalinga duomenų bazė, kurioje būtų saugomi ir kaupiami su projektu susijusi informacija bei reikalinga aplikacija, kuri apdorotų duomenis. Sistemą realizuoti galima bet kokia programavimo kalba ir naudojant bet kokią duomenų bazę. Jeigu projekto apimtys nėra didelės, tai realizavimo technologijos nėra labai svarbios. Tačiau jeigu projekto apimtys didelės, funkcijų skaičius didelis, tuomet būtina pasirūpinti, kad, išaugus informacijos kiekiui apie patį projektą bei jo funkcijas, darbo su sistema ir duomenų baze vėlavimas neaugtų proporcingai padidėjusiam informacijos kiekiui.

Duomenų bazei realizuoti pasirinkta *Oracle* duomenų bazė, kuri pasižymi kaip daug galimybių turinti, efektyvi, leidžianti vienu metu su ja dirbti dideliame vartotojų skaičiui, stabili, dideliems duomenų kiekiams saugoti skirta duomenų bazė. Projekto įverčių sistemai realizuoti pasirinkta *Oracle Forms* ir *Oracle Reports* technologijos, kurios palaiko vientisumą ir betarpišką integralumą su *Oracle* duomenų baze.

3.5.2. Prototipo duomenų koncepcinis modelis

VIVEMO-7 modelis realizuotas prototipu, kurio statinė struktūra pateikta klasių diagrama:



6 pav. Prototipo duomenų koncepcinis modelis.

3.5.3. Prototipo nefunkciniai reikalavimai ir apribojimai

VIVEMO-7 modelis realizuotas prototipu, kuriam yra keliami tam tikri nefunkciniai reikalavimai:

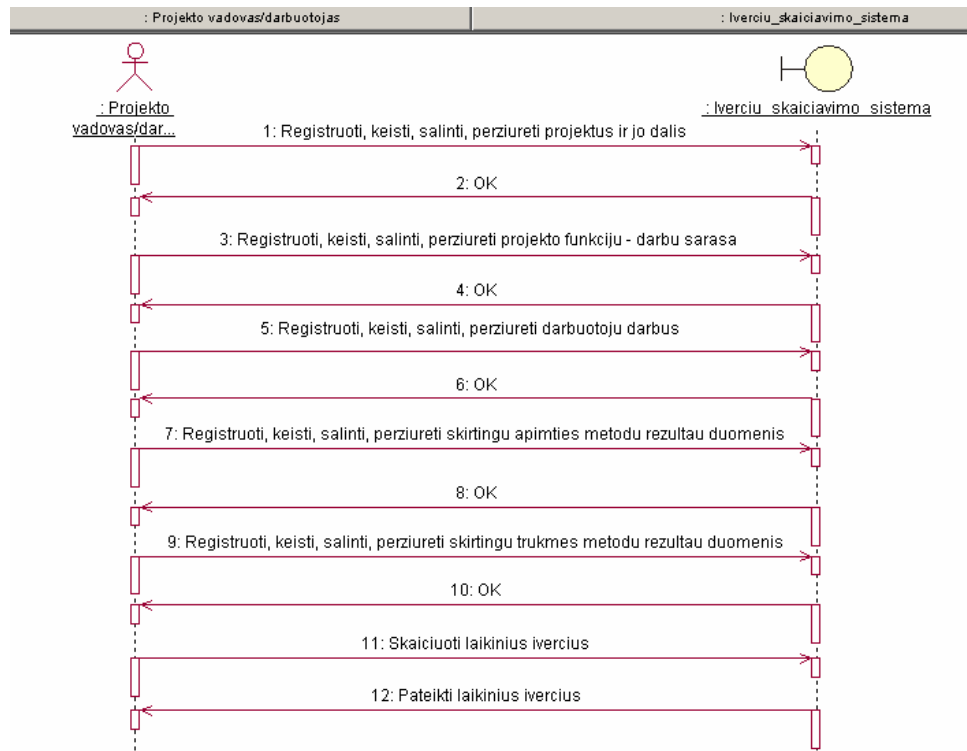
27 lentelė. Prototipo nefunkciniai reikalavimai.

Kam skirtas reikalavimas	Reikalavimo tipas	Reikalavimas	
Techninė įranga	Serveris	Operatyvi atmintis (RAM)	512 MB minimali, 1024 MB rekomenduojama
		Virtuali atmintis	2x Operatyvinės atminties

Kam skirtas reikalavimas	Reikalavimo tipas	Reikalavimas	
		Laikina vieta diske	200 MB
		Vieta diske	4 GB
		Vaizdo korta	256 spalvos
		Procesorius	200 MHz minimalus, 1 GHz rekomenduojamas
Techninė įranga	Klientas	Operatyvi atmintis (RAM)	64 MB minimali, 128 MB rekomenduojama
		Vieta diske	100 MB
		Vaizdo korta	256 spalvos
		Procesorius	200 MHz minimalus, 400 MHz rekomenduojamas
Sisteminė programinė įranga	Serveris	Operacinė sistema “Windows”, Oracle 9i duomenų bazė, Oracle aplikacijų serveris 9i.	
Sisteminė programinė įranga	Klientas	Operacinė sistema “Windows”, Interneto naršyklė, Java Runtime Environment 1.3 arba vėlesnė.	

3.5.4. Bendras prototipo elgsenos modelis

VIVEMO-7 modelis realizuotas prototipu, kurio bendras veikimo principas pateiktas seku diagrama:



7 pav. Komunikavimas tarp vartotojų ir prototipo sistemos.

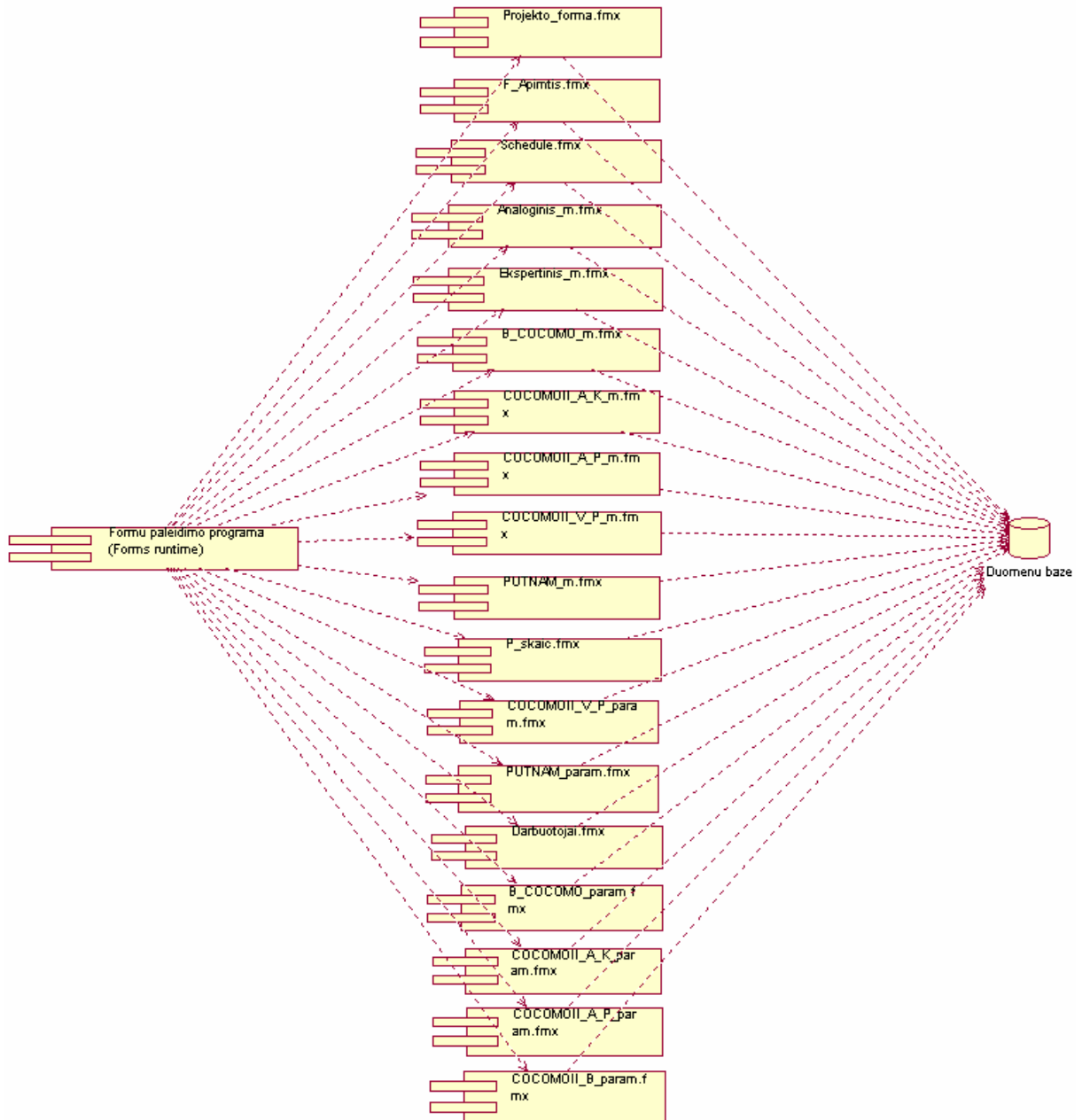
3.5.5. Prototipo komponentų modelis

Aplikacija sudaro programiniai moduliai duomenų registravimo bei apdorojimo formos, kurios yra vykdomos per formų paleidimo vykdomąsias programas. Žemiau pateiktas komponentų sąrašas bei juos vaizduojanti komponentų diagrama.

28 lentelė. Prototipo komponentų aprašas.

Nr.	Komponento pavadinimas	Komponento aprašas
1.	Projekto_forma.fmx	Informacijos apie projektą, jo etapus, funkcijas, darbuotojų paskyrimus registravimas, keitimas, peržiūra, šalinimas.
2.	F_Apimtis.fmx	Funkcijų apimčių registravimas SLOC, FP, OP metodais.
3.	Schedule.fmx	Atskirų funkcijų bei viso projekto įverčių sudarymas pagal skirtingus metodus:
1.1.	Analoginis_m.fmx	Įverčių sudarymas pagal Analoginį metodą.
1.2.	Ekspertinis_m.fmx	Įverčių sudarymas pagal Ekspertinį metodą.
1.3.	B_COCOMO_m.fmx	Įverčių sudarymas pagal Basic COCOMO modelį.
1.4.	COCOMOII_A_K_m.fmx	Įverčių sudarymas pagal COCOMO II Aplikacinio komponavimo modelį.
1.5.	COCOMOII_A_P_m.fmx	Įverčių sudarymas pagal COCOMO II Ankstyvo projektavimo modelį.
1.6.	COCOMOII_V_P_m.fmx	Įverčių sudarymas pagal COCOMO II Vėlyvo projektavimo modelį.
1.7.	PUTNAM_m.fmx	Įverčių sudarymas pagal Putnam modelį.
4.	P_skaic.fmx	Funkcijų bei viso projekto tikimybių skaičiavimas.
5.		Klasifikatoriai:
1.8.	Darbuotojai.fmx	Informacijos apie darbuotojus registravimas, keitimas, peržiūra, šalinimas.

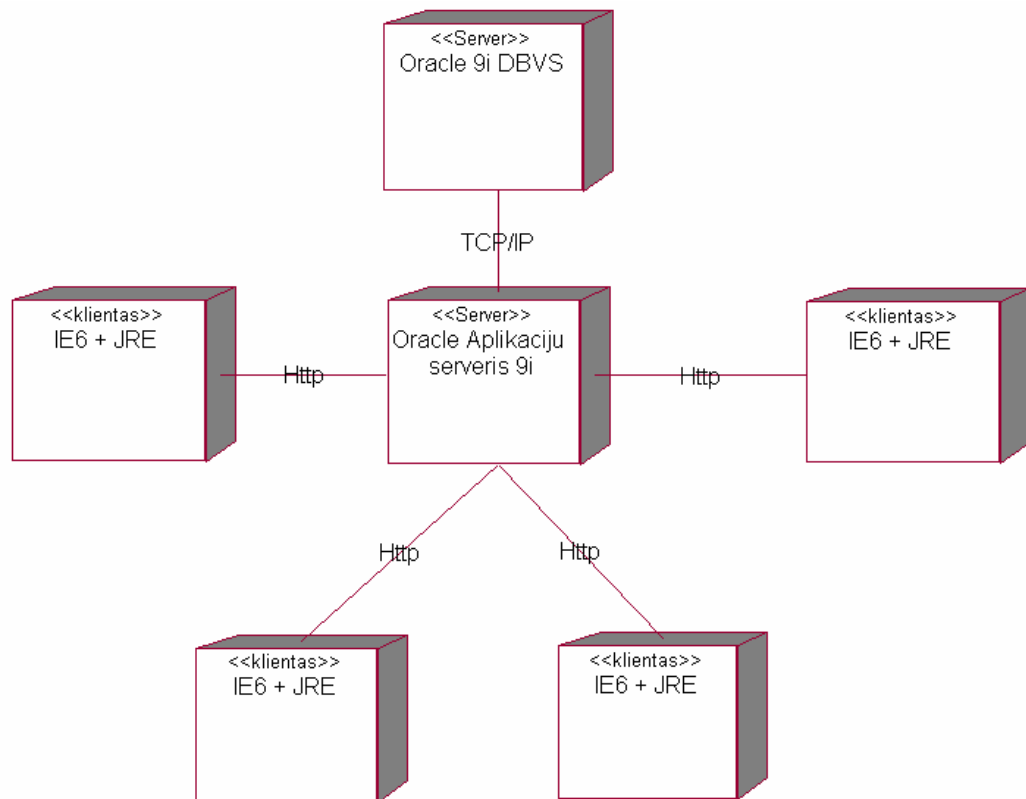
1.9.	B_COCOMO_param.fmx	Basic COCOMO modelio parametru registravimas, keitimas, peržiūra, šalinimas.
1.10	COCOMOII_A_K_param.fmx	COCOMO II Aplikacinio komponavimo modelio parametru registravimas, keitimas, peržiūra, šalinimas.
1.11	COCOMOII_B_param.fmx	COCOMO II modelio B parametru registravimas, keitimas, peržiūra, šalinimas.
1.12	COCOMOII_A_P_param.fmx	COCOMO II Ankstyvo projektavimo modelio parametru registravimas, keitimas, peržiūra, šalinimas.
1.13	COCOMOII_V_P_param.fmx	COCOMO II Vėlyvo projektavimo modelio parametru registravimas, keitimas, peržiūra, šalinimas.
1.14	PUTNAM_param.fmx	PUTNAM modelio parametru registravimas, keitimas, peržiūra, šalinimas.



8 pav. Prototipo programinių komponentų modelis.

3.5.6. Įdiegimo modelis

Šis modelis rodo fizinių techninių įrenginių išdėstymą bei vykdomųjų programų komponentų paskirstymą juose. Ši sistema skirta darbui lokaliame tinkle. Serveryje bus įdiegta *Oracle* duomenų bazė ir aplikacija, kurios vykdomuosius programinius modulius – formas bei ataskaitas per tinklą vykdys vartotojai. Vartotojų skaičius teoriškai neribojamas.



9 pav. Prototipo įdiegimo modelis.

4. EKSPERIMENTINIS INFORMACINĖS SISTEMOS PROGRAMINĖS ĮRANGOS KŪRIMO DARBŲ APIMTIES ĮVERTINIMO MODELIO TYRIMAS

4.1. Projekto „X“ įverčių tyrimas

Ekspperimentiniam tyrimui atlikti pasirenkame projektą „X“. Žemiau pateikta projekto struktūra – darbų – funkcijų sąrašas bei trumpas aprašymas:

29 lentelė. Projekto „X“ darbų – funkcijų sąrašas.

Nr.	Funkcijos pavadinimas	Funkcijos aprašas	Funkcijos santrumpa
1.	Užduoties inicijavimas / registravimas.	Pradinių duomenų registravimas / modifikavimas.	F1
2.	Užduoties dokumento spausdinimas.	Pagal patvirtintą formą.	F2
3.	Užduoties anuliavimas.	Užduoties dokumento anuliavimas, požymio registravimas.	F3
4.	Vizitavimo užduoties registravimas.	Pradinių duomenų, reikalingų vizitavimui, registravimas.	F4
5.	Vizitavimo užduoties anuliavimas.	Vizitavimo dokumento anuliavimas.	F5
6.	Veiklos kontrolės užduoties registravimas.	Pradinių duomenų, reikalingų veiklos kontrolei, registravimas.	F6
7.	Veiklos kontrolės užduoties anuliavimas.	Veiklos kontrolės dokumento anuliavimas.	F7
8.	Vizitavimo ir veiklos kontrolės dokumento formavimas ir spausdinimas.	Pagal patvirtintą formą.	F8
9.	Vizitavimo ir veiklos kontrolės rezultatų duomenų registravimas	Registruojami atlikti veiksmai, nustatyti prieštaravimai/pažeidimai,	F9

Nr.	Funkcijos pavadinimas	Funkcijos aprašas	Funkcijos santrumpa
		trūkumų pašalinimo būdai.	
10.	Pranešimo apie pažeidimus formavimas ir spausdinimas.	Pagal patvirtintą formą.	F10
11.	Pabaigos registravimas.	Pabaigai reikalingos informacijos registravimas: faktinė pabaiga, priskaičiuotos sumos, veiksmai po, išvados.	F11
12.	Rezultatų ataskaitos formavimas ir spausdinimas.	Pagal patvirtintą formą.	F12
13.	Baigtų/nebaigtų užduočių suvestinės formavimas ir spausdinimas.	Reikalingų duomenų atrinkimas ir ataskaitos formavimas.	F13
14.	Žinyno "Pažeidimai" pildymas.	Klasifikatoriaus administravimas.	F14
15.	Žinyno "„Atliekami veiksmai" " pildymas.	Klasifikatoriaus administravimas.	F15
16.	Žinyno " Veiksmai po " " pildymas.	Klasifikatoriaus administravimas.	F16
17.	Žinyno " Tikslai" " pildymas.	Klasifikatoriaus administravimas.	F17
18.	Žinyno „Nustatyti faktai“ pildymas.	Klasifikatoriaus administravimas.	F18
19.	Žinyno „Pašalinimo būdai“ pildymas.	Klasifikatoriaus administravimas.	F19

4.2. Projekto reikalavimų ir analizės etapas

Šiame etape naudojome:

- 1) Analoginį vertinimo metodą. Naudojomės praeitų panašių projektų patirtimi. Sąvoka „panašus projektas“ reiškia projektą, kuris buvo realizuotas tokia pačia technologija bei tam pačiam klientui. Įmonėje yra nusistovėjusi analoginių laikinių įverčių parinkimo reikšmių lentelė (mėnesyje yra 20 darbo dienų):

30 lentelė. Modulių vertinimo klasifikatorius pagal analogą.

	Paprastas, dienų skaičius	Paprastas, mėnesio dalis	Vidutinis, dienų skaičius	Vidutinis, mėnesio dalis	Sudėtingas, dienų skaičius	Sudėtingas, mėnesio dalis
Forma	1	0,05	2	0,1	3	0,15
Ataskaita	0,5	0,025	1	0,05	2	0,1
Komponentas	1	0,05	3	0,15	5	0,25

Funkcijų bei viso projekto laikiniai įverčiai pagal Analoginį metodą pateikti metodų suvestinėje (36 lentelė).

- 2) Ekspertinį vertinimo metodą. Naudojamės skirtingų ekspertų funkcijoms pateiktomis išvadomis ir laikiniais įverčiais. Šie rezultatai pateikti metodų suvestinėje (36 lentelė).
- 3) Basic COCOMO modelį. Pagal kuriamos IS charakteristikas parenkame reikalingus parametrus iš 2.5.2.3 skyrelio (13 lentelė. 14 lentelė.):

31 lentelė. Projektui „X“ parinkti Basic COCOMO modelio parametrai.

Parametras	Reikšmė
Režimas	<i>Organic mode</i>
a	2.4
b	1.05
c	0.38

Analizuojame ir skaičiuojame kiekvienos funkcijos apimtį / dydį pagal funkcinių taškų metodą (2.5.1.2 skyrelis):

32 lentelė. Projekto „X“ darbų – funkcijų apimties / dydžio matavimai pagal funkcinių taškų metodą.

Nr.	Funkcijos santrumpa	Funkcinių taškų tipai, jų kiekiai, sudėtingumai										
		EI	Koef.	EO	Koef.	EQ	Koef.	ILI	Koef.	EII	Koef.	Suma
1.	F1	1	6	-	-	1	6	1	10	1	10	32
Komentariai apie F1 funkcinius taškus:		Užduoties duomenų rinkinio registravimas		-		Užduočių peržiūra		Užduoties duomenų rinkinys		Ryšys su kita sistema		-
2.	F2	-	-	1	7	-	-	1	10	-	-	17
Komentariai apie F2 funkcinius taškus:		-		Užduoties duomenų rinkinio spausdinimas		-		Užduoties duomenų rinkinys		-		-
3.	F3	1	3	-	-	1	4	1	10	-	-	17
Komentariai apie F3 funkcinius taškus:		Anuliavimo požymio registravimas užduoties duomenų rinkinyje		-		Anuliuotų užduočių peržiūra.		Užduoties duomenų rinkinys		-		-
4.	F4	1	4	-	-	1	4	1	10	-	-	18
Komentariai apie F4 funkcinius taškus:		Vizitavimo užduoties duomenų rinkinio registravimas		-		Vizitavimo užduočių peržiūra		Vizitavimo duomenų rinkinys		-		-
5.	F5	1	3	-	-	1	4	1	10	-	-	17
Komentariai apie F5 funkcinius taškus:		Anuliavimo požymio registravimas vizitavimo užduoties duomenų rinkinyje		-		Anuliuotų vizitavimų peržiūra		Vizitavimo duomenų rinkinys		-		-

Nr.	Funkcijos santrumpa	Funkcinių taškų tipai, jų kiekiai, sudėtingumai										
		EI	Koef.	EO	Koef.	EQ	Koef.	ILI	Koef.	EII	Koef.	Suma
6.	F6	1	4	-	-	1	4	1	10	-	-	18
Komentariai apie F6 funkcinius taškus:		Veiklos kontrolės užduoties duomenų rinkinio registravimas		-		Veiklos kontrolės užduočių peržiūra		Veiklos kontrolės duomenų rinkinys		-		
7.	F7	1	3	-	-	1	4	1	10	-	-	17
Komentariai apie F7 funkcinius taškus:		Anuliavimo požymio registravimas veiklos kontrolės užduoties duomenų rinkinyje		-		Anuliuotų veiklos kontrolių peržiūra		Veiklos kontrolės duomenų rinkinys		-		
8.	F8	-	-	1	6	-	-	1	15	-	-	21
Komentariai apie F8 funkcinius taškus:		-		Vizitavimo ir veiklos kontrolės užduočių duomenų rinkinio spausdinimas		-		Vizitavimo ir veiklos kontrolės duomenų rinkinys		-		
9.	F9	1	6	-	-	1	6	1	15	-	-	27
Komentariai apie F9 funkcinius taškus:		Vizitavimo ir veiklos kontrolės rezultatų registravimas		-		Vizitavimo ir veiklos kontrolės rezultatų peržiūra		Vizitavimo ir veiklos kontrolės duomenų rinkinys		-		-
10.	F10	-	-	1	5	-	-	1	10	-	-	15
Komentariai apie F10		-		Pranešimo apie		-		Pranešimo apie		-		-

Nr.	Funkcijos santrumpa	Funkcinių taškų tipai, jų kiekiai, sudėtingumai										
		EI	Koef.	EO	Koef.	EQ	Koef.	ILI	Koef.	EII	Koef.	Suma
	funkcinius taškus:			pažeidimus duomenų rinkinio spausdinimas					pažeidimus duomenų rinkinys			
11.	F11	1	6	-	-	1	6	1	15	-	-	27
	Komentariai apie F11 funkcinius taškus:	Pabaigos duomenų rinkinio registravimas		-		Pabaigos duomenų peržiūra		Pabaigos duomenų rinkinys		-		
12.	F12	-	-	1	7	-	-	1	15	-	-	22
	Komentariai apie F12 funkcinius taškus:	-		Pabaigos duomenų rinkinio spausdinimas		-		Pabaigos duomenų rinkinys		-		-
13.	F13	-	-	1	5	-	-	1	10	-	-	15
	Komentariai apie F13 funkcinius taškus:	-		Baigtų / nebaigtų užduočių spausdinimas		-		Užduoties duomenų rinkinys		-		-
14.	F14	1	3	-	-	1	3	1	7	-	-	13
	Komentariai apie F14 funkcinius taškus:	Pažeidimų reikšmių registravimas		-		Pažeidimų reikšmių peržiūra		Pažeidimų duomenų rinkinys		-		-
15.	F15	1	3	-	-	1	3	1	7	-	-	13
	Komentariai apie F15 funkcinius taškus:	Atliekamų veiksmų reikšmių registravimas		-		Atliekamų veiksmų reikšmių peržiūra		Atliekamų veiksmų duomenų rinkinys		-		-
16.	F16	1	3	-	-	1	3	1	7	-	-	13
	Komentariai apie F16 funkcinius taškus:	Veiksmų po reikšmių registravimas		-		Veiksmų po reikšmių peržiūra		Veiksmų po duomenų rinkinys		-		-

Nr.	Funkcijos santrumpa	Funkcinių taškų tipai, jų kiekiai, sudėtingumai										
		EI	Koef.	EO	Koef.	EQ	Koef.	ILI	Koef.	EII	Koef.	Suma
taškus:												
17.	F17	1	3	-	-	1	3	1	7	-	-	13
Komentariai apie F17 funkcinius taškus:		Tikslų reikšmių registravimas		-		Tikslų reikšmių peržiūra		Tikslų duomenų rinkinys		-		-
18.	F18	1	3	-	-	1	3	1	7	-	-	13
Komentariai apie F18 funkcinius taškus:		Nustatytų faktų reikšmių registravimas		-		Nustatytų faktų reikšmių peržiūra		Nustatytų faktų duomenų rinkinys		-		-
19.	F19	1	3	-	-	1	3	1	7	-	-	13
Komentariai apie F19 funkcinius taškus:		Pašalinimo būdų reikšmių registravimas		-		Pašalinimo būdų reikšmių peržiūra		Pašalinimo būdų duomenų rinkinys		-		-
Viso:											341	

Konvertuojame į KSLOC dydį pagal konvertavimo klasifikatorių (9 lentelė.), kiekvienai funkcijai apskaičiuojame svorį bei laikiną įvertį. Rezultatai pagal šį metodą pateikti metodų suvestinėje (36 lentelė).

- 4) PUTNAM modelį. Pagal kuriamos IS charakteristikas parenkame reikalingus parametrus iš 2.5.2.7 skyrelio (22 lentelė, 23 lentelė.):

33 lentelė. Projektui „X“ parinkti Putnam modelio parametrai.

Parametras	Reikšmė
P	28657
B	0.16

Šiam metodui reikalingus funkcijų dydžius (SLOC) bei svorius parenkame iš prieš tai buvusio skyrelio. Funkcijų bei viso projekto laikiniai įverčiai pagal šį metodą pateikti metodų suvestinėje (36 lentelė).

- 5) COCOMO II Aplikacinį komponavimo modelį. Pagal kuriamos IS charakteristikas parenkame reikalingus parametrus iš 2.5.2.4 skyrelio (15 lentelė.):

34 lentelė. Projektui „X“ parinkti COCOMO II aplikacinio komponavimo modelio parametrai.

Parametras	Reikšmė
PROD	25
REUSE	0
SCED	1.00
PREC	0.81
FLEX	3.64
RESL	0.84
TEAM	0.99
PMAT	2.73
B	9.01

Šiame modelyje funkcijų apimtis matuojama objekciniais taškais. Žemiau pateikiami struktūriniai skaičiavimai:

35 lentelė. Projekto „X“ darbų – funkcijų apimtys / dydžio matavimai pagal objektinių taškų metodą.

Nr.	Funkcijos santrumpa	Objektinių taškų tipai, jų kiekiai, sudėtingumai							
		Forma	Koef.	Ataskaita	Koef.	Komp.	Koef.	Suma	
1.	F1	1	2	-	-	1	10	12	
Komentaras apie F1 objektinius taškus:		Vidutinė forma, nes duomenų lent. ska 3, view'ų 3			-		Sudėtingas komponentas, nes ryšys su kita sistema		-
2.	F2	-	-	1	5	-	-	5	
Komentaras apie F2 objektinius taškus:		-			Vidutinė ataskaita, nes duomenų lent. ska 2, view'ų 4		-		-
3.	F3	1	1	-	-	-	-	1	
Komentaras apie F3 objektinius taškus:		Paprasta forma, nes duomenų lent. ska			-		-		-

Nr.	Funkcijos santrumpa	Objektinių taškų tipai, jų kiekiai, sudėtingumai						
		Forma	Koef.	Ataskaita	Koef.	Komp.	Koef.	Suma
	taškus:	1, view'ų 1						
4.	F4	1	1	-	-	-	-	1
	Komentariai apie F4 objektinius taškus:	Paprasta forma, nes duomenų lent. ska 2, view'ų 3		-		-		-
5.	F5	1	1	-	-	-	-	1
	Komentariai apie F5 objektinius taškus:	Paprasta forma, nes duomenų lent. ska 1, view'ų 2		-		-		-
6.	F6	1	1	-	-	-	-	1
	Komentariai apie F6 objektinius taškus:	Paprasta forma, nes duomenų lent. ska 2, view'ų 3		-		-		-
7.	F7	1	1	-	-	-	-	1
	Komentariai apie F7 objektinius taškus:	Paprasta forma, nes duomenų lent. ska 1, view'ų 2		-		-		-
8.	F8	-	-	1	5	-	-	5
	Komentariai apie F8 objektinius taškus:	-		Vidutinė ataskaita, nes duomenų lent. ska 3, view'ų 5		-		-
9.	F9	1	3	-	-	1	10	13
	Komentariai apie F9 objektinius taškus:	Sudėtinga forma, nes duomenų lent. ska 5, view'ų 7		-		Sudėtingas rezultatų tikrinimo komponentas		-
10.	F10	-	-	1	2	-	-	2
	Komentariai apie F10 objektinius taškus:	-		Paprasta ataskaita, nes duomenų lent. ska 1, view'ų 3		-		-
11.	F11	1	1	-	-	-	-	1
	Komentariai apie F11 objektinius	Paprasta forma, nes duomenų lent. ska		-		-		-

Nr.	Funkcijos santrumpa	Objektinių taškų tipai, jų kiekiai, sudėtingumai						
		Forma	Koef.	Ataskaita	Koef.	Komp.	Koef.	Suma
	taškus:	3, view'ų 2						
12.	F12	-	-	1	8	-	-	8
	Komentariai apie F12 objektinius taškus:	-		Sudėtinga ataskaita, nes duomenų lent. ska 4, view'ų 16		-		-
13.	F13	-	-	1	2	-	-	2
	Komentariai apie F13 objektinius taškus:	-		Paprasta ataskaita, nes duomenų lent. ska 0, view'ų 2		-		-
14.	F14	1	1	-	-	-	-	1
	Komentariai apie F14 objektinius taškus:	Paprasta forma, nes duomenų lent. ska 1, view'ų 0		-		-		-
15	F15	1	1	-	-	-	-	1
	Komentariai apie F15 objektinius taškus:	Paprasta forma, nes duomenų lent. ska 1, view'ų 0		-		-		-
16.	F16	1	1	-	-	-	-	1
	Komentariai apie F16 objektinius taškus:	Paprasta forma, nes duomenų lent. ska 1, view'ų 0		-		-		-
17.	F17	1	1	-	-	-	-	1
	Komentariai apie F17 objektinius taškus:	Paprasta forma, nes duomenų lent. ska 1, view'ų 0		-		-		-
18.	F18	1	1	-	-	-	-	1
	Komentariai apie F18 objektinius taškus:	Paprasta forma, nes duomenų lent. ska 1, view'ų 0		-		-		-
19.	F19	1	1	-	-	-	-	1
	Komentariai apie F19 objektinius	Paprasta forma, nes duomenų lent. ska		-		-		-

Nr.	Funkcijos santrumpa	Objektinių taškų tipai, jų kiekiai, sudėtingumai						
		Forma	Koef.	Ataskaita	Koef.	Komp.	Koef.	Suma
	taškus:	1, view'ų 0						
							Viso:	59

Funkcijų bei viso projekto laikiniai įverčiai pagal šį metodą pateikti metodų suvestinėje (36 lentelė).

- 6) Rezultatų suvestines. Panaudojus aukščiau aprašytus metodus ir modelius buvo gauti skirtingi projekto darbų trukmės laikiniai įverčiai bei funkcijų svoriai. Šie rezultatai pateikti projekto reikalavimų ir analizės etapo metodų suvestinėje (36 lentelė.).

36 lentelė. Reikalavimų ir analizės etapo metodų suvestinė - projekto „X“ darbų – funkcijų skirtingi svoriai bei skirtingų metodų laikiniai įverčiai.

Nr.	Funkcijos santrumpa	m _i , pagal FP metodą	m _i , pagal OP metodą	m _i , pagal SLOC metodą	Laikinis įvertis pagal Ekspertinį m., mėnesiais	Laikinis įvertis pagal Analoginį m., mėnesiais	Laikinis įvertis pagal Basic COCOMO m., mėnesiais	Laikinis įvertis pagal Putnam m., mėnesiais	Laikinis įvertis pagal COCOMO II app. m., mėnesiais
1.	F1	0.09	0.203	0.10	0.5	0.812	0.45	0.531	0.459
2.	F2	0.04	0.085	0.05	0.1	0.34	0.2	0.236	0.204
3.	F3	0.04	0.017	0.05	0.1	0.068	0.2	0.236	0.204
4.	F4	0.05	0.017	0.06	0.1	0.068	0.25	0.295	0.255
5.	F5	0.04	0.017	0.05	0.2	0.068	0.2	0.236	0.204
6.	F6	0.05	0.017	0.06	0.15	0.068	0.25	0.295	0.255
7.	F7	0.04	0.017	0.05	0.1	0.068	0.2	0.236	0.204
8.	F8	0.06	0.085	0.07	0.4	0.34	0.3	0.354	0.306
9.	F9	0.08	0.220	0.09	0.5	0.88	0.4	0.472	0.408
10.	F10	0.04	0.034	0.05	0.1	0.136	0.2	0.236	0.204
11.	F11	0.08	0.017	0.09	0.5	0.068	0.4	0.472	0.408
12.	F12	0.07	0.136	0.08	0.3	0.544	0.35	0.413	0.357
13.	F13	0.04	0.034	0.05	0.1	0.136	0.2	0.236	0.204
14.	F14	0.03	0.017	0.04	0.2	0.068	0.15	0.177	0.153
15.	F15	0.03	0.017	0.04	0.1	0.068	0.15	0.177	0.153
16.	F16	0.03	0.017	0.04	0.15	0.068	0.15	0.177	0.153
17.	F17	0.03	0.017	0.04	0.1	0.068	0.15	0.177	0.153
18.	F18	0.03	0.017	0.04	0.15	0.068	0.15	0.177	0.153
19.	F19	0.03	0.017	0.04	0.1	0.068	0.15	0.177	0.153
Projekto trukmė pagal Ekspertinį metodą:					3.9				
Projekto trukmė pagal Analoginį metodą:						4.1			
Projekto trukmė pagal Basic COCOMO modelį:							4.5		
Projekto trukmė pagal PUTNAM modelį								5.9	
Projekto trukmė pagal COCOMO II Aplikacinį komponavimo modelį:									5.1

Išanalizavus šiuos duomenis ir kiekvienam darbui - funkcijai apskaičiavus optimistinį, pesimistinį, labiausiai tikėtiną, tikėtiną laikinius įverčius, standartinį nuokrypį ir tikimybę sudaryta projekto reikalavimų ir analizės etapo rezultatų suvestinė (37 lentelė.):

37 lentelė. Projekto „X“ rezultatų suvestinė reikalavimų bei analizės etape.

Darbas f_i	Optimistinis $laikas_{optimist}$	Labiausiai tikėtinas $laikas_{vidut}$	Pesimistinis $laikas_{pesimist}$	Tikėtinas $Laikas_i(f_i)$	Standartinis nuokrypis s_i	Dydis z_i	Tikimybė p_i
F1	0.450	0.550	0.812	0.577	-0.030	0.900	80
F2	0.100	0.216	0.340	0.217	-0.001	1.300	90
F3	0.068	0.162	0.236	0.158	-0.005	-0.600	25
F4	0.068	0.194	0.295	0.189	-0.040	-0.100	45
F5	0.068	0.183	0.236	0.172	0.008	1.300	90
F6	0.068	0.204	0.295	0.196	0.006	1.150	85
F7	0.068	0.163	0.236	0.158	0.003	0.900	80
F8	0.300	0.340	0.400	0.343	-0.006	0.550	70
F9	0.400	0.532	0.880	0.568	-0.144	0.250	60
F10	0.100	0.175	0.236	0.173	-0.020	-0.100	45
F11	0.068	0.370	0.500	0.341	-0.047	-0.600	25
F12	0.300	0.393	0.544	0.403	0.025	-0.400	35
F13	0.100	0.175	0.236	0.173	0.002	1.300	90
F14	0.068	0.150	0.200	0.144	0.006	0.900	80
F15	0.068	0.130	0.177	0.127	0.008	0.250	60
F16	0.068	0.140	0.177	0.134	0.004	1.150	85
F17	0.068	0.130	0.177	0.127	0.004	0.550	70
F18	0.068	0.140	0.177	0.134	0.009	0.550	70
F19	0.068	0.130	0.177	0.127	0.004	0.550	70
Projekto tikėtina trukmė: 4.461							
Projekto įvykdymo tikimybė:							53%

Viso projekto gauti laikinių įverčių rezultatai:

- optimistinė trukmė – 2.566 mėnesių,

- pesimistinė trukmė – 6.331 mėnesių,
- vidutinė trukmė – 4.477 mėnesių,
- planuojama (tikėtina) trukmė – 4.461 mėnesių,
- standartinis nuokrypis – -0.6275,
- z reikšmė – -0.0255,
- projekto neįvykdymo tikimybė – 47%,
- projekto įvykdymo tikimybė – 53%.
- Faktinė projekto trukmė - 5.4 mėnesių

4.3. Projektavimo etapas

Šiame etape naudojome:

- 2) Analoginį vertinimo metodą. Seka tokia pati kaip ir 4.2 skyrelyje, 1 punkte.
- 3) Ekspertinį vertinimo metodus. Seka tokia pati kaip ir 4.2 skyrelyje, 2 punkte.
- 4) Basic COCOMO modelį. Seka tokia pati kaip ir 4.2 skyrelyje, 3 punkte.
- 5) PUTNAM modelį. Seka tokia pati kaip ir 4.2 skyrelyje, 4 punkte.
- 6) COCOMO II Aplikacinį komponavimo modelį. Seka tokia pati kaip ir 4.2 skyrelyje, 5 punkte.
- 7) COCOMO II Ankstyvo projektavimo modelį. Parenkame reikalingus parametrus:

38 lentelė. Projektui „X“ parinkti COCOMO II Ankstyvo projektavimo modelio parametrai.

Parametras	Reikšmė
PREC	0.81
FLEX	3.64
RESL	0.84
TEAM	0.99
PMAT	2.73
B	9.01
RCPX	10
RUSE	1.14
PDIF	9
PERS	8
PREX	12
FCIL	7

Parametras	Reikšmė
SCED	1.00
ASLOC	0
AT	40
ATPROD	2000

8) Rezultatų suvestinė. Išanalizavus šiuos duomenis ir kiekvienam darbui - funkcijai apskaičiavus optimistinį, pesimistinį, labiausiai tikėtiną, tikėtiną laikinius įverčius, standartinį nuokrypį ir tikimybę sudaryta analogiška suvestinė.

Viso projekto gauti laikinių įverčių rezultatai:

- optimistinė trukmė – 3.147 mėnesių,
- pesimistinė trukmė – 6.547 mėnesių,
- vidutinė trukmė – 4.91 mėnesių,
- planuojama (tikėtina) trukmė – 4.889 mėnesių,
- standartinis nuokrypis – 0.056,
- z reikšmė – 0.50,
- projekto neįvykdymo tikimybė – 31%,
- projekto įvykdymo tikimybė – 69%.
- Faktinė projekto trukmė - 5.4 mėnesių

4.4. Realizavimo ir vėlesni etapai

Šiame etape naudojome:

- 1) Ekspertinį vertinimo metodus. Seka tokia pati kaip ir 4.2 skyrelyje, 2 punkte.
- 2) PUTNAM modelį. Seka tokia pati kaip ir 4.2 skyrelyje, 4 punkte.
- 3) COCOMO II Aplikacinį komponavimo modelį. Seka tokia pati kaip ir 4.2 skyrelyje, 5 punkte.
- 4) COCOMO II Vėlyvo projektavimo modelį.
- 9) Rezultatų suvestines. Išanalizavus šiuos duomenis ir kiekvienam darbui - funkcijai apskaičiavus optimistinį, pesimistinį, labiausiai tikėtiną, tikėtiną laikinius įverčius, standartinį nuokrypį ir tikimybę sudaryta analogiška.

Viso projekto gauti laikinių įverčių rezultatai:

- optimistinė trukmė – 4.912 mėnesių,
- pesimistinė trukmė – 5.647 mėnesių,
- vidutinė trukmė – 5.271 mėnesių,
- planuojama (tikėtina) trukmė – 5.1 mėnesių,
- standartinis nuokrypis – 0.147,
- z reikšmė – 1.25,
- projekto neįvykdymo tikimybė – 11%,
- projekto įvykdymo tikimybė – 89%.
- Faktinė projekto trukmė - 5.4 mėnesių

Projekto vėlesniuose etapuose jau pakankamai turima informacijos apie konkrečius programinius modulius ar komponentus. Visi aplinkos veiksmi bei integralumo faktoriai įvertinami per metodus bei modelius ankstesniuose etapuose. Todėl jeigu neprisideda naujų funkcijų laikiniai įverčiai, o taip pat viso projekto trukmė nesikeičia.

5. IŠVADOS

- 1) Atlikta projekto darbų apimties vertinimo metodų analizė. Atrinkti trys plačiausiai naudojami standartiniai darbų apimties vertinimo metodai.
- 2) Atlikta projekto darbų trukmės vertinimo metodų analizė ir nustatyta, kad plačiausiai naudojami trukmės vertinimo metodai naudoja standartinius darbų apimties metodus.
- 3) Atlikta projektų darbų trukmės vertinimo metodų tinkamumo projekto etapams analizė ir nustatyti šių metodų panaudojimo deriniai.
- 4) Sukurtas informacinės sistemos programinės įrangos darbų apimties bei trukmės įvertinimo modelis – VIVEMO-7. Modelyje integruoti trys standartiniai projekto darbų apimties bei septyni projekto trukmės vertinimo metodai.
- 5) Atskirai lyginant VIVEMO-7 modelį su jį sudarančiais metodais nustatyti privalumai:
 - a. Informacinės sistemos programinės įrangos kūrimo projekto darbų – funkcijų apimtis nustatoma skirtingais pjūviais. Apimtis nustatoma tiek subjektyviai pagal panašius projektus praeityje, tiek ekspertų dėka, tiek funkcinių taškų, objektinių taškų bei kodo eilučių skaičiavimo metodais.
 - b. Apsauga nuo neteisingo metodo panaudojimo. Jeigu tam tikras metodo panaudojimas duoda klaidingus arba netikslius laikinius įverčius galutinis laikinis įvertis yra išlyginamas kitų metodų bei modelių dėka.
 - c. Įdiegtas rizikos valdymas. Egzistuoja tikimybė, kad prielaida, jog apskaičiuoti bei įvertinti informacinės sistemos programinės įrangos kūrimo darbų apimtis bei trukmės laikiniai įverčiai yra klaidingi arba netikslūs. Šis netikslumas yra įvertinamas naudojant tiek atskirų darbų – funkcijų, tiek viso projekto įvykdymo tikimybę.
- 6) VIVEMO-7 modelio trūkumas lyginant su jį sudarančiais metodais. VIVEMO-7 modelį sudaro panašūs metodai bei modeliai, kurių dėka optimizuotos šio modelio panaudojimo laiko sąnaudos. Tačiau šios sąnaudos, visgi yra didesnės nei atskirai naudojant kuri nors vieną iš septynių metodų bei modelių.
- 7) Atliktas VIVEMO-7 modelio eksperimentinis tyrimas naudojant projektą „X“. Išanalizavus tyrimo rezultatus paaiškėja, kad reikalavimų ir analizės etape, buvo netiksliai išmatuota darbų – funkcijų apimtis. Dėl to viso projekto planuojamas laikinis įvertis buvo trumpesnis negu faktinis. Vėlesniuose etapuose sukaupus daugiau informacijos apie konkrečias funkcijas buvo gauti tikslesni skaičiavimai. Kiekviename etape viso projekto trukmė vis ilgėjo, tačiau įvykdymo tikimybė didėjo. Dėl to daroma išvada, kad projekto laikinių įverčių tikslumas didėja su vėlesniu projekto etapu.
- 8) Didžiausia rizika yra blogai apskaičiuoti darbų – funkcijų apimtį. Informacinės sistemos programinės įrangos kūrimo darbų apimčiai nustatyti naudojami keli apimties nustatymo

metodai bei septyni skirtingi trukmės laikinių įverčių nustatymo metodai bei modeliai.

Tačiau neteisingai ar klaidingai nustačius bei apskaičiavus visais trimis apimties metodais darbų – funkcijų apimtį bus gauti neteisingi bei klaidingi trukmės laikiniai įverčiai. Tokiu atveju tiek projekto darbų – funkcijų, tiek viso projekto įvykdymo tikimybės bus klaidingos. Apimties nustatymas yra vienas iš svarbiausių uždavinių.

- 9) Magistro darbo tematika perskaitytas pranešimas informacinių technologijų magistrantų ir doktorantų konferencijoje ir paskelbtas straipsnis [23].

6. LITERATŪRA

1. Als A. and Greenidge C. The waterfall model. Prieiga per internetą : http://scitec.uwichill.edu.bb/cmp/online/cs221/waterfall_model.htm [žiūrėta 2005.05.08]
2. Arlene Minkiewicz F. Measuring Object Oriented Software with Predictive Object Points. Prieiga per internetą : <http://www.pricystems.com/downloads/pdf/pops.pdf>. [žiūrėta 2004.11.02]
3. Boehm B., Abts C. and Clark B. Cocomo II Model Definition. Prieiga per internetą : <http://www.classes.ccc.wustl.edu/~cse528/modelman.pdf>. [žiūrėta 2005.04.10]
4. Boehm B. Abts C. Software Development Cost Estimation Approaches. Prieiga per internetą : <http://sunset.usc.edu/publications/TECHRPTS/2000/usccse2000-505/usccse2000-505.pdf>. [žiūrėta 2004.10.10]
5. Champeaux D., Lea D. and Faure P. Object-Oriented System Development. Prieiga per internetą : <http://gee.cs.oswego.edu/dl/oosdw3/> [žiūrėta 2004.10.17]
6. Cotterel M. and Hughes B. Software Project Management. Oxford and Northampton, Great Britain, Alden Press Limited, 1995.
7. Famelong J. Software Cost Estimation. Prieiga per internetą : [http://olaf.crema.unimi.it/cost1.ppt#256,1,Software Cost Estimation](http://olaf.crema.unimi.it/cost1.ppt#256,1,Software%20Cost%20Estimation) [žiūrėta 2005.03.22]
8. Fetcke T., Abran A. and Tho-Hau Nguyen. Mapping the OO-Jacobson Approach into Function Point Analysis. Prieiga per internetą : <http://www.cs.unibo.it/people/faculty/cianca/wwwpages/ids/lecture/>. [žiūrėta 2004.12.22]
9. Gramantieri F., Lamma E., Mello P. and Riguzzi F. A System for Measuring Function Points from Specifications. Prieiga per internetą : <http://dcm.cl.uh.edu/capf01gp4/delivers/gramantieri97system.pdf>. [žiūrėta 2004.12.03]
10. IT Cortex. Statistics over IT projects failure rate. Prieiga per internetą : [http://www.it-cortex.com/Stat_Failure_Rate.htm#The%20Conference%20Board%20Survey \(2001\)](http://www.it-cortex.com/Stat_Failure_Rate.htm#The%20Conference%20Board%20Survey%20(2001)). [žiūrėta 2005.05.24]
11. Johnson K. Software Cost Estimation: Metrics and Models. Prieiga per internetą : <http://sern.ucalgary.ca/courses/seng/621/W98/johnsonk/cost.htm> [žiūrėta 2005.04.29]
12. Jorgensen M. A Review of Studies on Expert Estimation of Software Development Effort.. Prieiga per internetą : http://www.simula.no/photo/experts/submitnovember2002_copy.pdf [žiūrėta 2005.02.10]
13. Kalb George E. Counting Lines Of Code, Confusions, Conclusions, And Recommendations. Prieiga per internetą : http://sunset.usc.edu/research/CODECOUNT/documents/3rd_REVIC.pdf [žiūrėta 2005.09.25]

14. Liming W. The Comparison of the Software Cost Estimating Methods. Prieiga per internetą :
<http://www.compapp.dcu.ie/~renaat/ca421/LWu1.html> [žiūrėta 2005.04.12]
15. Longstreet D. Fundamentals of Function Points Analysis. Prieiga per internetą:
<http://www.ifpug.com/files/Fundamentals%20of%20Function%20Point%20Analysis.pdf> [žiūrėta 2004.12.15]
16. Mookken K. and Jorgensen M. A Review of Surveys on Software Effort Estimation. Prieiga per internetą :
http://www.ifi.uio.no/~isu/INCO/Papers/Review_final8.pdf [žiūrėta 2005.02.17]
17. Park Robert E. Software Size Measurement: A Framework for Counting Source Statements, 1996. Prieiga per internetą :
<http://www.sei.cmu.edu/publications/documents/92.reports/92.tr.020.html> [žiūrėta 2005.09.18]
18. Pressman R. S. Basic Cocomo Model Prieiga per internetą:
<http://www.engin.umd.umich.edu/CIS/course.des/cis525/js/f00/gamel/help.html> [žiūrėta 2004.11.29]
19. Shepperd M. and Schofield C. Effort Estimation by Analogy: A Case Study. Prieiga per internetą:
<http://dec.bournemouth.ac.uk/ESERG/ANGEL/ESCOM96.html> [žiūrėta 2005.02.25]
20. Sencer S. and Karakab U. Cost Estimation Models. Prieiga per internetą :
<http://yunus.hun.edu.tr/~sencer/cocomo.html> [žiūrėta 2005.09.25]
21. University of Southern California. COCOMO Model Definition Manual. Prieiga per internetą :
<http://sunset.usc.edu/research/COCOMOII/Docs/modelman.pdf>. [žiūrėta 2004.11.25]
22. Vienneau R. Function Points. Prieiga per internetą :
<http://www.softwaretechnews.com/fall95/function.html> [žiūrėta 2004.11.10]
23. Mikulėnas G. Informacinės sistemos programinės realizacijos apimties įvertinimo modelis// Informacinės Technologijos. 2005, p. 205-208.

1 priedas. Prototipo ekraniniai vaizdai.

Action Edit Query Block Record Field Help Window ORACLE

Projektai ir projekto dalys

Projektai

Id	Pavadinimas	Nuo	Iki
1	Projektas "Y"	2005.11.01	2005.12.31

Komentaras: Šis projektas yra jau pabaigtas ir visa informacija saugoma archyvuose. Projektas yra mažos apimties ir i dalis neskaidomas

Projekto dalys

Id	Pavadinimas	Nuo	Iki
1	Projektas "Y"	2005.11.01	2005.12.31

Komentaras: Šis projektas

Record: 1/1

10 pav. Projekto ir projekto dalių forma.

Action Edit Query Block Record Field Help Window ORACLE

Etapai ir funkcijos

Projektai

Id: 1 Pavadinimas: Projektas "Y"

Projekto dalys

Id	Pavadinimas
1	Projektas "Y"

Etapai

Id	Pavadinimas
1	Reikalavimų etapas
2	Analizės etapas
3	Projektavimo etapas
4	Programavimo etapas

Funkcijos

Id	Reg Data	Pavadinimas	Faktinis vertis	Vardas	Pavarde	Aprasas
1	2005.12.10	Užduoties inicijavimas / re		Gytenis	Mikuler	Aprasas
2	2005.12.11	Užduoties dokumento sp		Gytenis	Mikuler	Aprasas
3	2005.12.12	Vizitavimo užduoties regis		Gytenis	Mikuler	Aprasas

Record: 1/1

11 pav. Projekto etapų ir funkcijų forma.

Action Edit Query Block Record Field Help Window ORACLE

Basic COCOMO metodai

Projektai

Id	Pavadinimas
1	Projektas "Y"

Projekto sudedamosios dalys

Id	Pavadinimas
1	Projektas "Y"

Projekto etapai

Id	Pavadinimas

Funkcijos

Id	Pavadinimas	Faktinis Ivertis	Reg Data	Aprasas

Record: 1/1

14 pav. Basic COCOMO metodo forma.

Action Edit Query Block Record Field Help Window ORACLE

Basic COCOMO metodo parametru klasifikatorius

Parametrai ir ju reiksmes

ID	Sistema	a	b	c	Aktyvus
1	Organic	2.4	1.05	.38	<input type="checkbox"/>
2	Semi - detached	3	1.12	.35	<input checked="" type="checkbox"/>
3	Embedded	3.6	1.2	.32	<input type="checkbox"/>

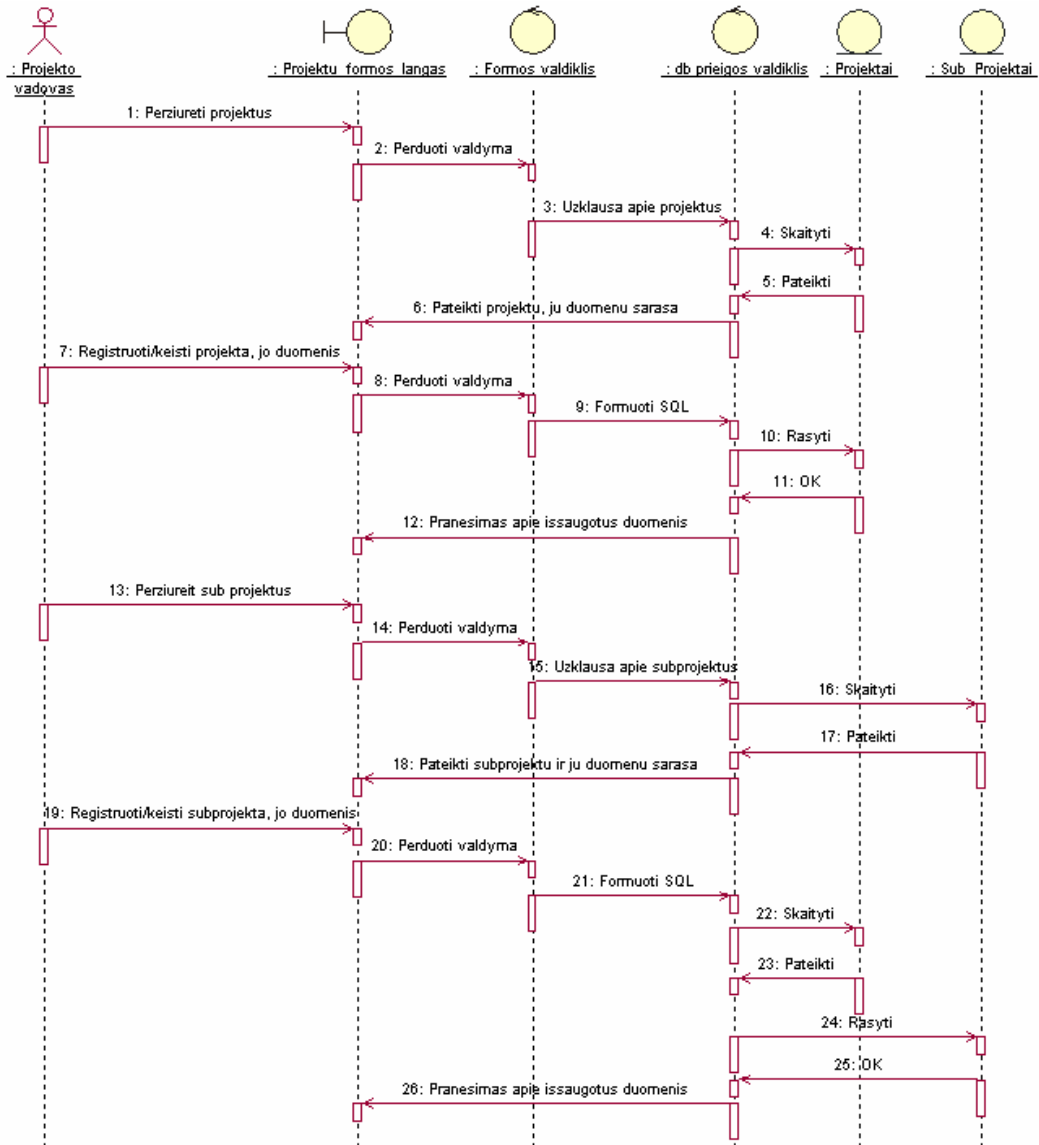
Record: 1/3

15 pav. Basic COCOMO parametru forma.

2 priedas. Dinaminis prototipo modelis

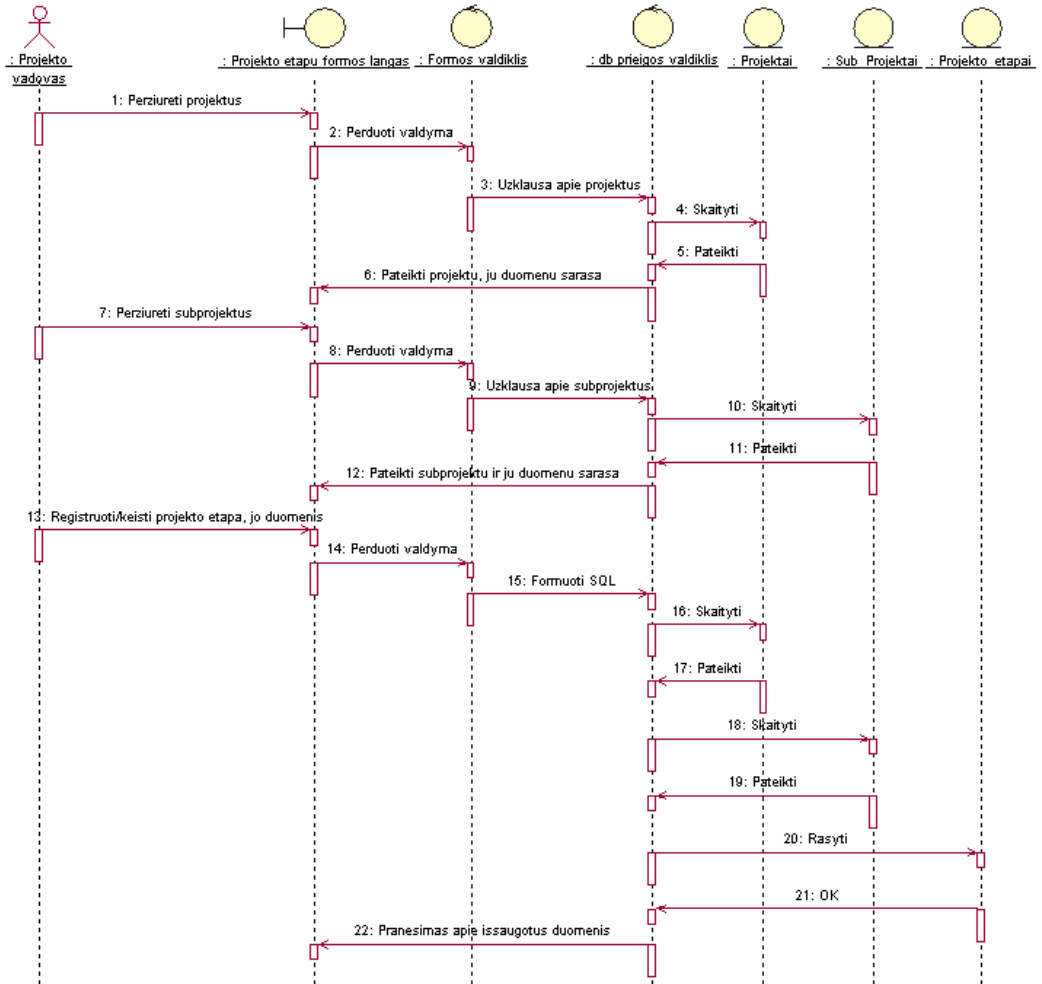
Žemiau pateiktos prototipo dinaminis modelis, naudojant sekų diagramas. Kiekviena iš jų vaizduoja programinių modulių funkcionalumą bei elgseną.

1. Projekto ir projekto dalių programinio modulio sekų diagrama.



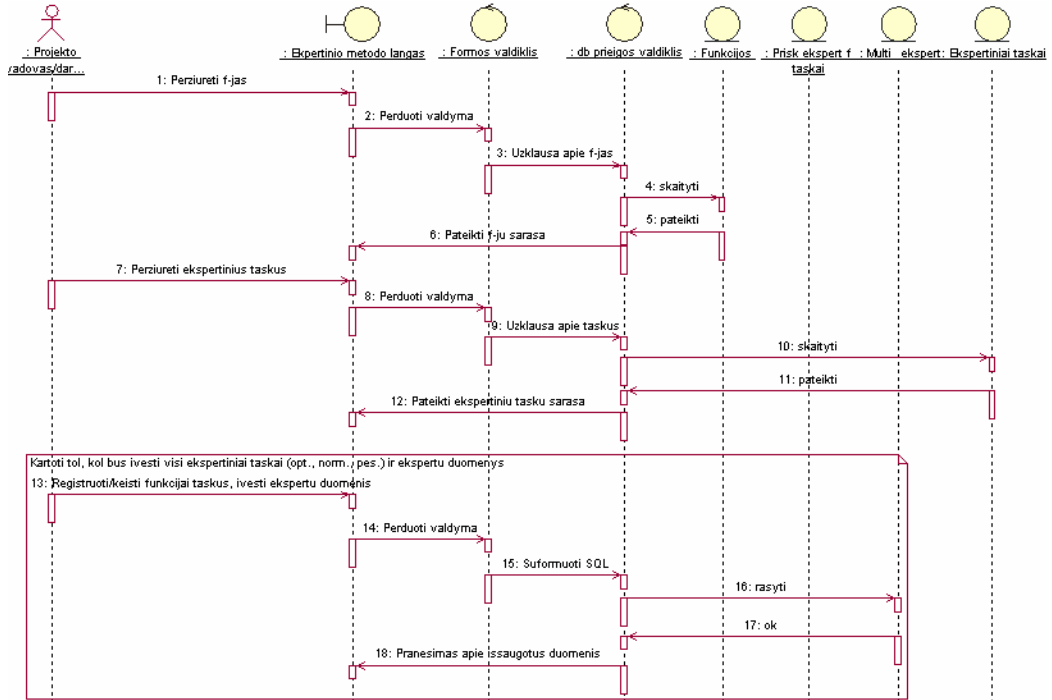
16 pav. Projekto ir projekto dalių programinio modulio sekų diagrama.

2. Projekto etapų programinio modulio sekų diagrama.



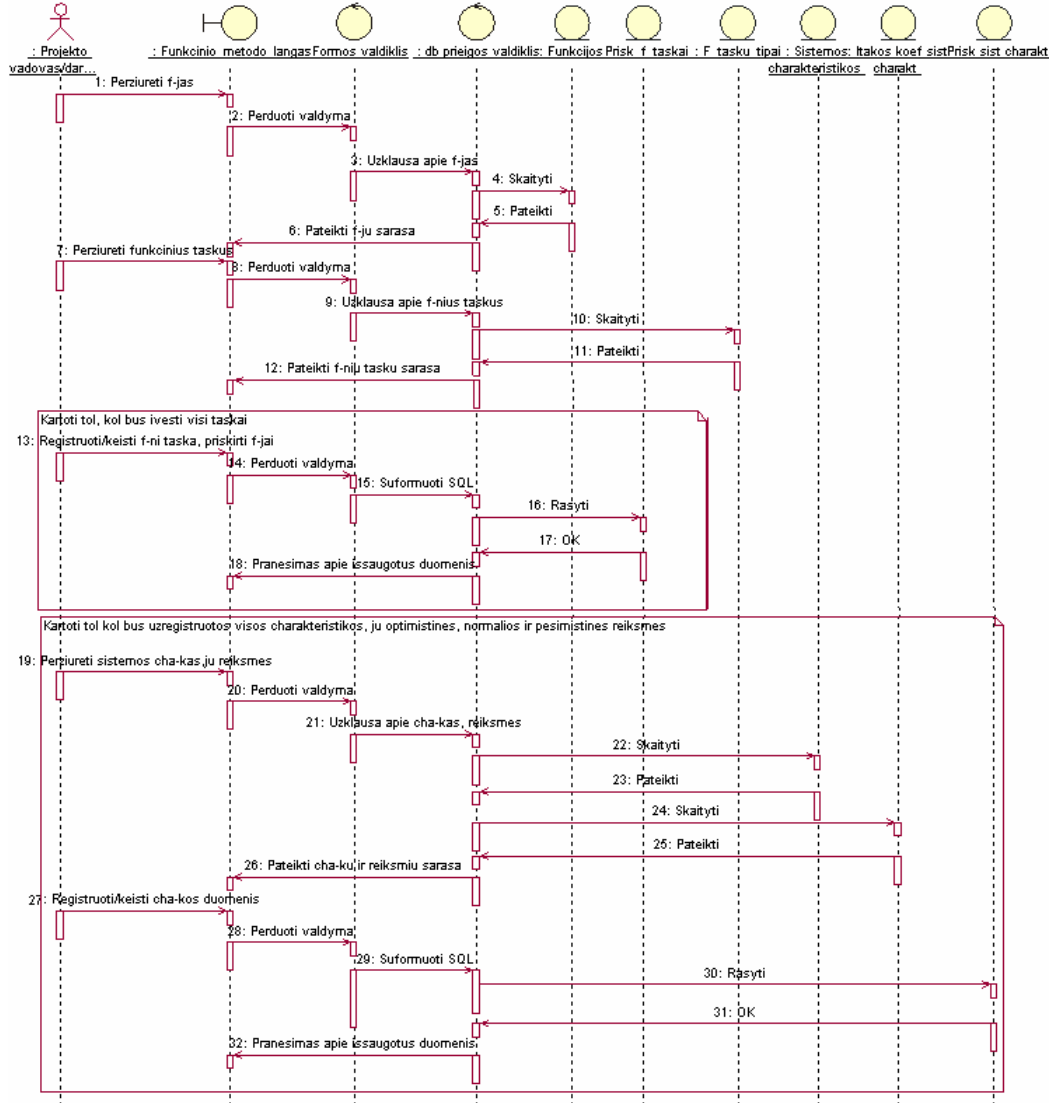
17 pav. Projekto etapų programinio modulio sekų diagrama.

3. Ekspertinio metodo programinio modulio sekų diagrama.



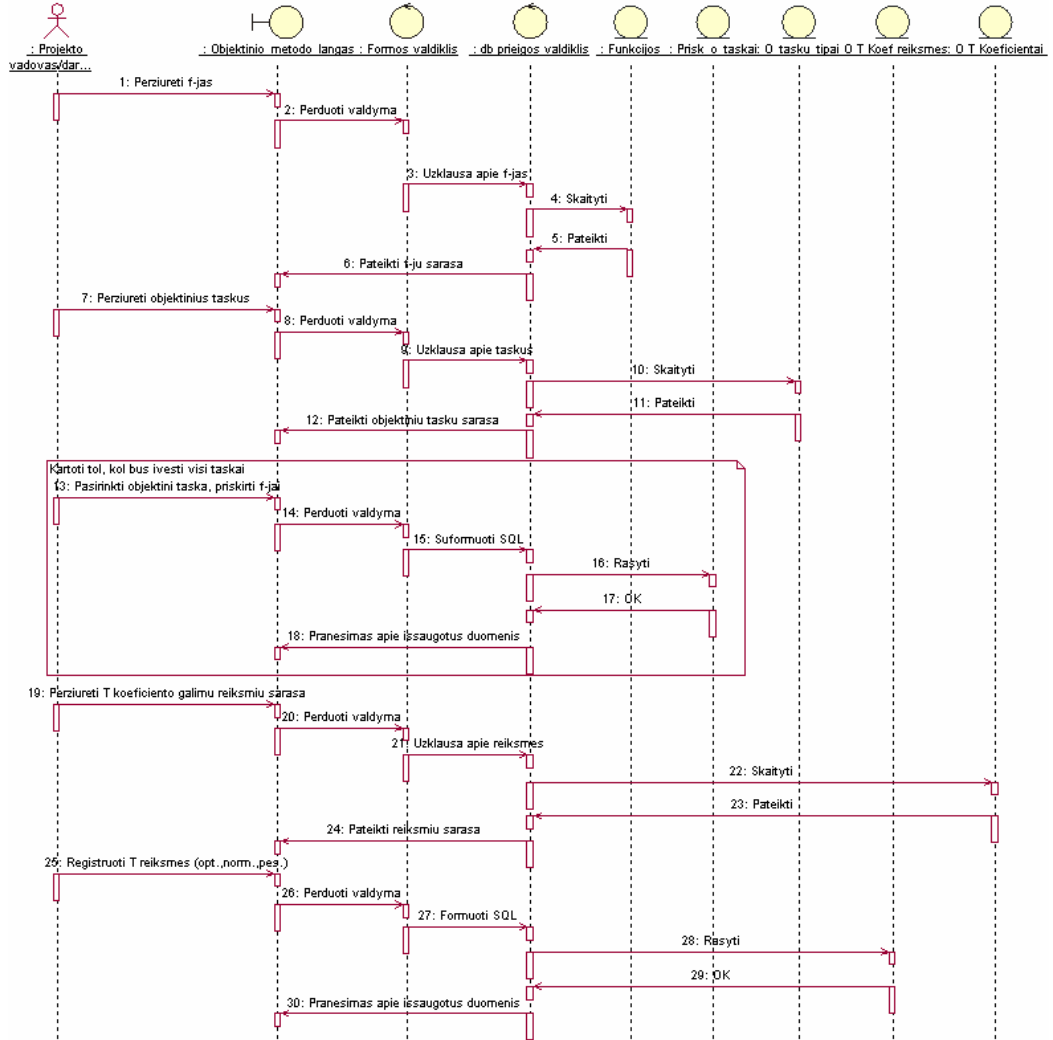
18 pav. Ekspertinio metodo programinio modulio sekų diagrama.

4. Funkcinių taškų metodo programinio modulio sekų diagrama.



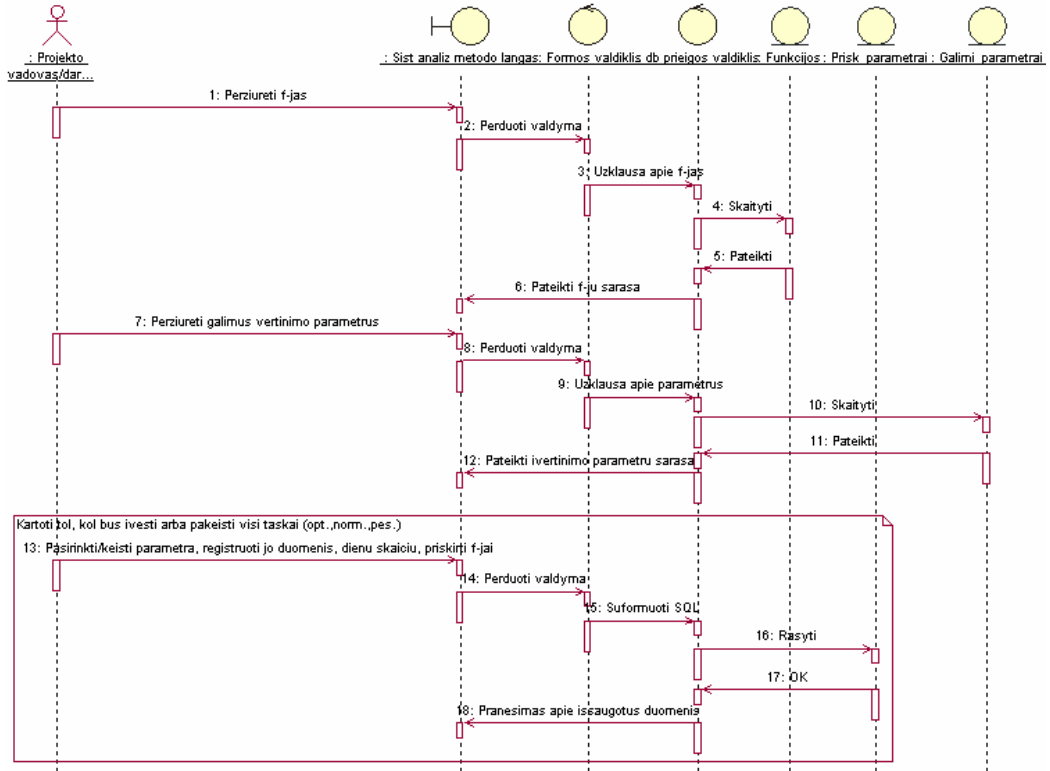
19 pav. Funkcinių taškų metodo programinio modulio sekų diagrama.

5. Objektinių taškų metodo programinio modulio sekų diagrama.



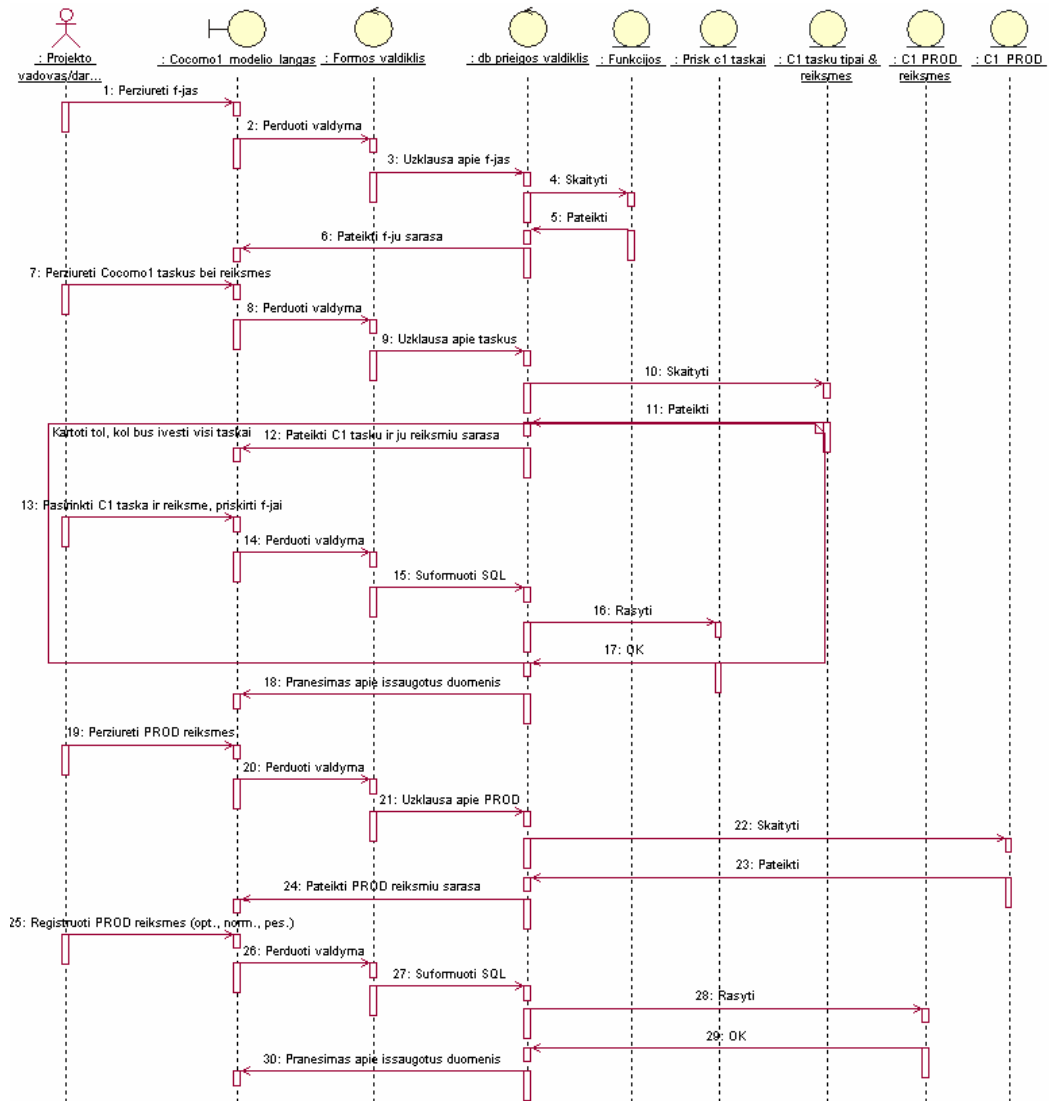
20 pav. Objektinių taškų metodo programinio modulio sekų diagrama.

6. Sisteminės analizės metodo programinio modulio sekų diagrama.



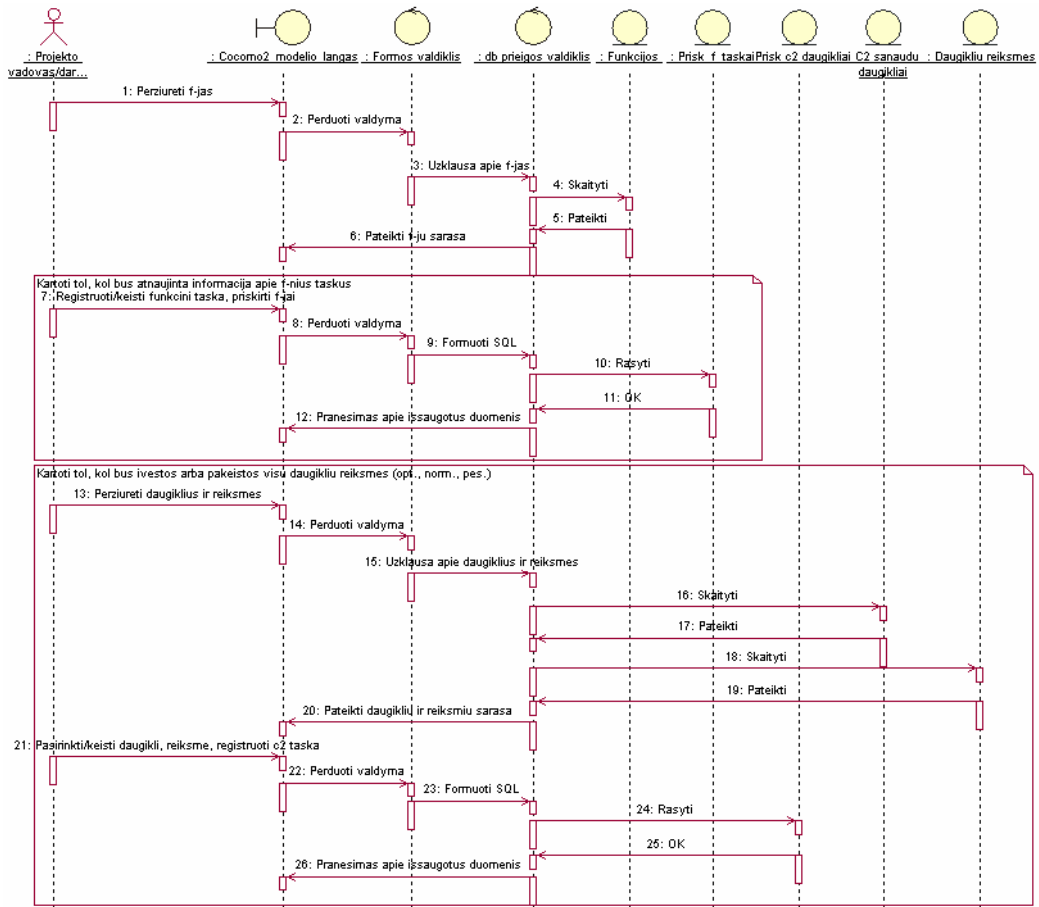
21 pav. Sisteminės analizės metodo programinio modulio sekų diagrama.

7. Cocomo II, 1 metodo programinio modulio sekų diagrama.



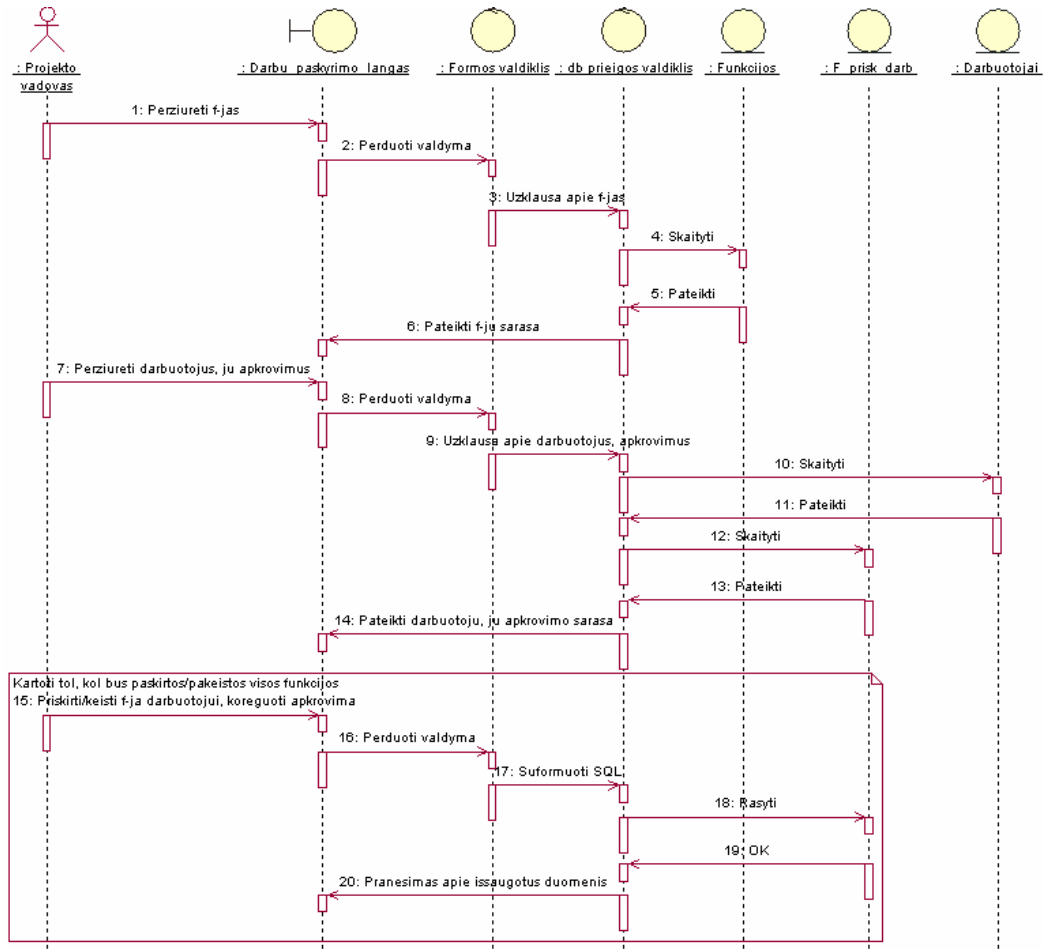
22 pav. Cocomo II, 1 metodo programinio modulio sekų diagrama.

8. Cocomo II, 2 metodo programinio modulio sekų diagrama.



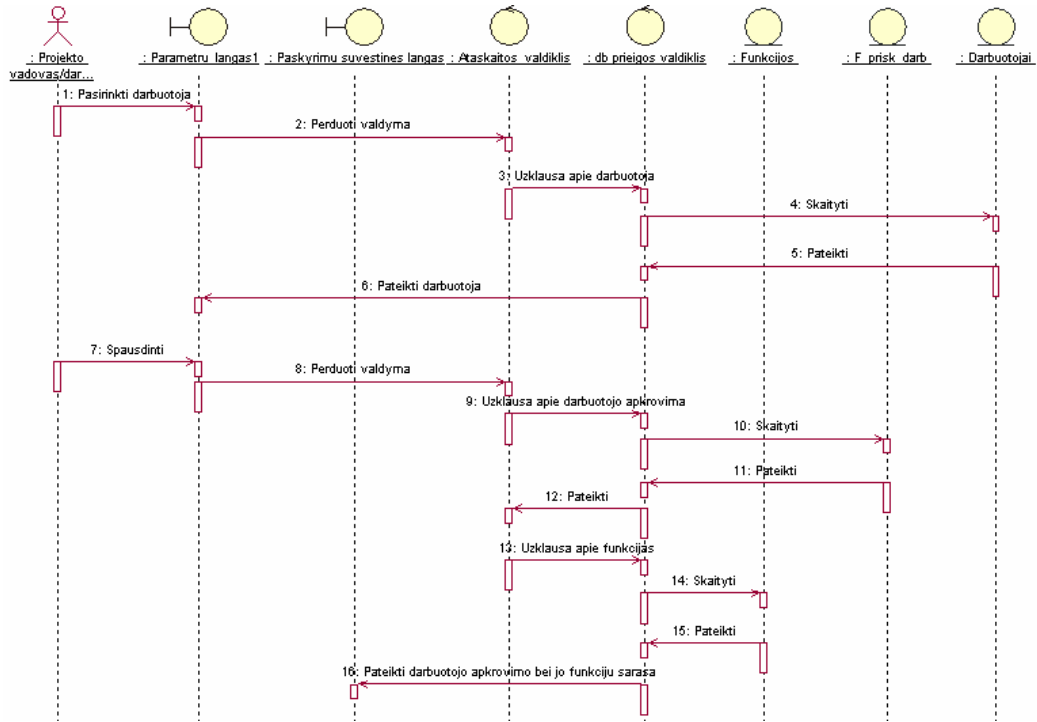
23 pav. Cocomo II, 2 metodo programinio modulio sekų diagrama.

9. Darbų paskyrimo programinio modulio sekų diagrama.



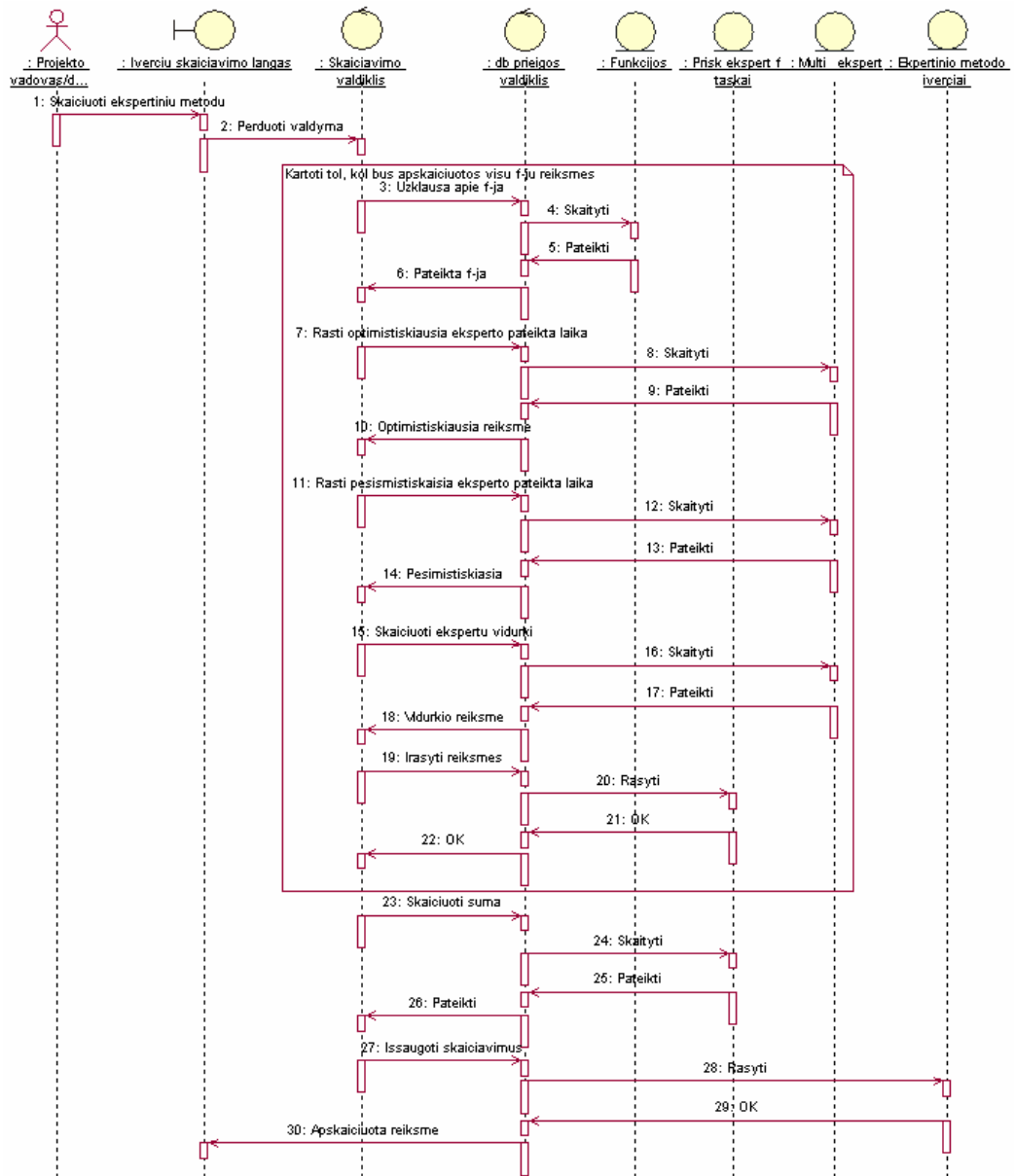
24 pav. Darbų paskyrimo programinio modulio sekų diagrama.

10. Konkretaus darbuotojo darbų sąrašo ir apkrovimo programinio modulio sekų diagrama.

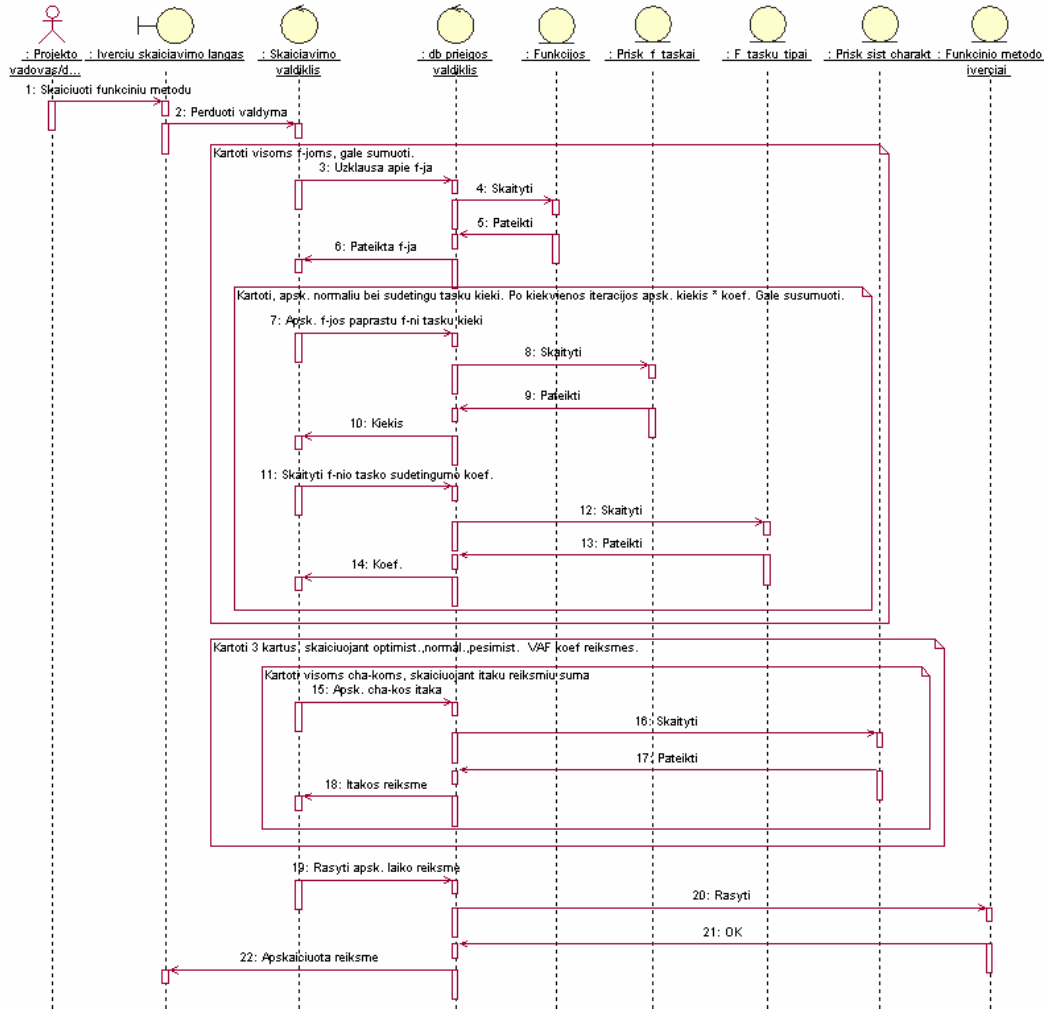


25 pav. Konkretaus darbuotojo darbų sąrašo ir apkrovimo programinio modulio sekų diagrama.

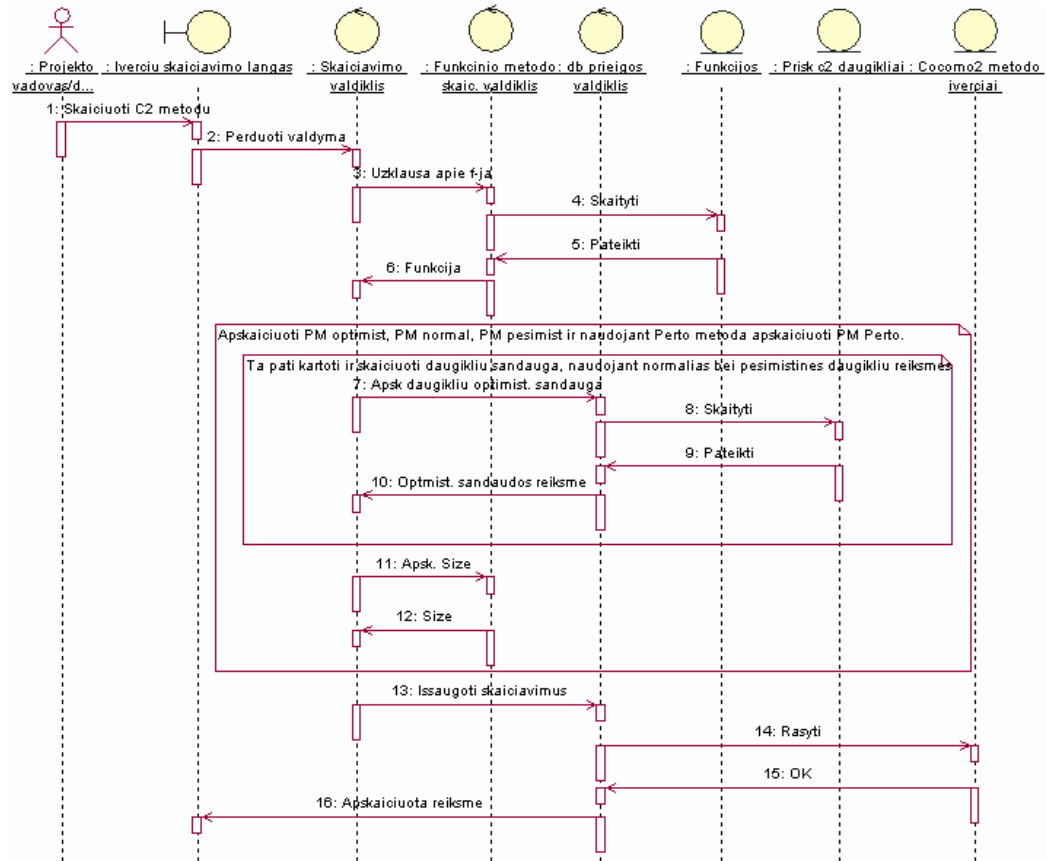
11. Įverčių skaičiavimo forma yra vienas iš sudėtingiausių sistemos komponentų, kurios veikimo mechanizmui pavaizduoti sukurtos 6 sekų diagramos:



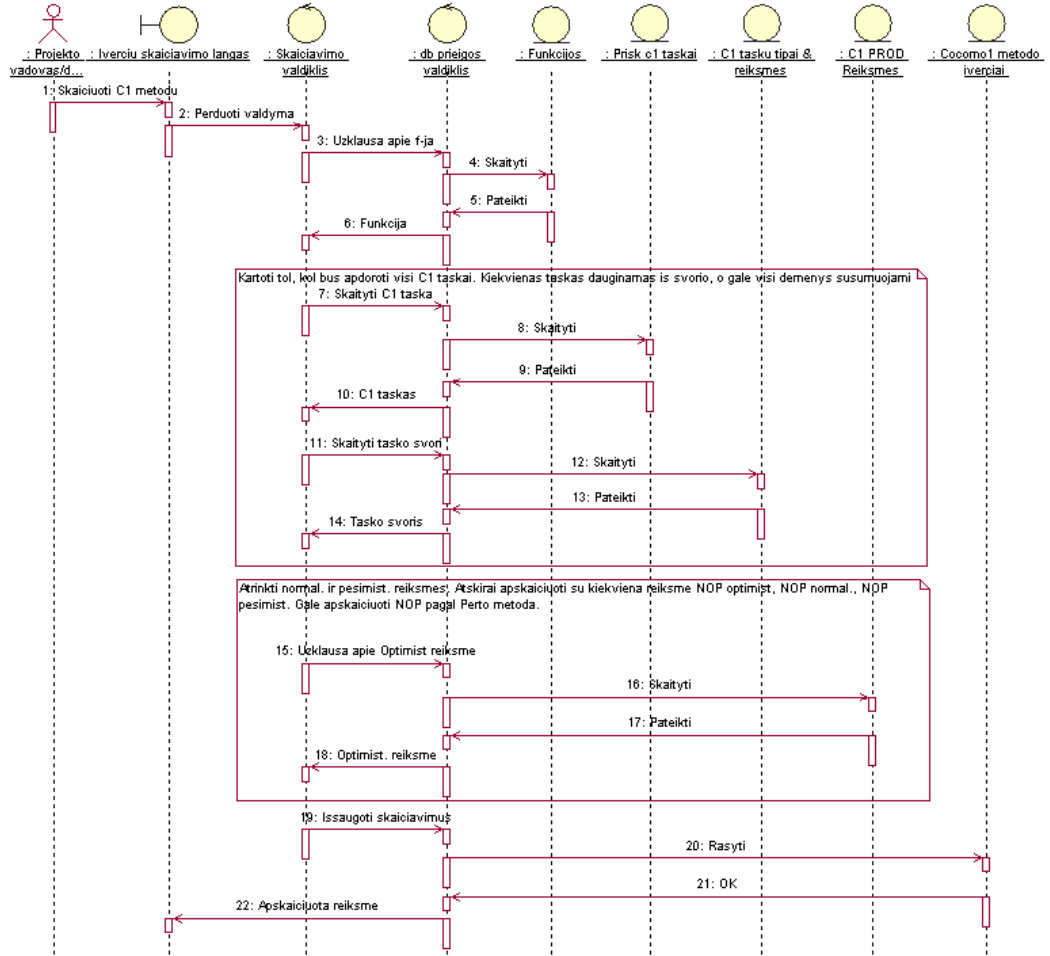
26 pav. Įverčių skaičiavimo forma (ekspertiniu metodu)



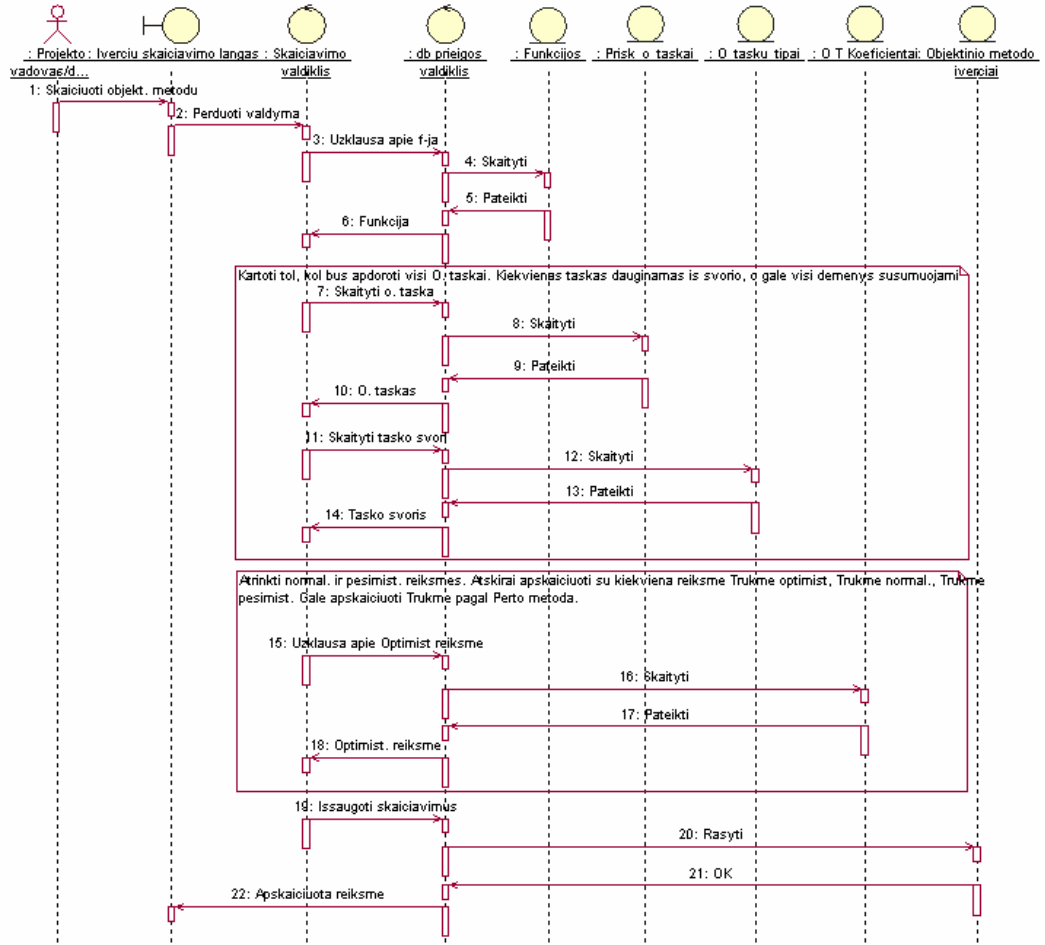
27 pav. Įverčių skaičiavimo forma (funkciniu metodu)



28 pav. Įverčių skaičiavimo forma (COCOMO II 2 modeliu)



29 pav. Ivercių skaičiavimo forma (COCOMO II I modeliui)



30 pav. Įverčių skaičiavimo forma (COCOMO II I modeliu)

3 priedas. Informacinės sistemos programinės įrangos kūrimo darbų apimtys bei trukmės įvertinimo įrankiai.

	Įrankio pavadinimas	Platforma	Įrankio kūrėjo interneto svetainė	Komentaras
1.	EZEstimate	Windows	Strive Logic http://www.openrage.com/main/ezestimate_full.htm	Use case based effort estimation tool
2.	20s Reference Estimator	Excel	20smackers http://www.20smackers.com	Creates estimates by referring to historical projects
3.	20s Estimation Calculator	Excel	20smackers http://www.20smackers.com	Excel-based for making early effort and cost estimates
4.	Estimate Easy Use Case	Windows	Duversa Software http://www.duversa.com/	Effort estimation tool based on use cases
5.	Crystal Ball Predictor	Excel	Decisioneering www.decisioneering.com	Excel based tool for forecasting
6.	ESTIMACS		Computer Associates International Inc. http://www.ca.com/products/estimacs.htm	
7.	CA-FPXpert		Computer Associates International Inc. http://www.ca.com/products/estimacs.htm	FPA
8.	SLIM-Estimate		Quantitative Software Management Inc. http://www.qsm.com	PUTNAM
9.	Costar		Softstar Systems Inc. http://www.softstarsystems.com	COCOMO
10.	Checkpoint		Software Productivity Research Inc. http://www.spr.com	Guides the user through the development of a software project estimate and plan using a ge base of more than 6,700 projects
11.	Function Point Workbench	Windows	Software Productivity Research Inc. http://www.spr.com	FPA
12.	KnowledgePlan	Windows	Software Productivity Research Inc. http://www.spr.com	ad hoc (“sizing by analogy”), support also schedule planning
13.	ESTIMATE Professional	Windows	Software Productivity Center http://www.spc.com	Putnam, COCOMO II
14.	SEER-SEM	Windows	Galorath http://www.galorath.com	Predicts, measures and analyzes resources,

	Įrankio pavadinimas	Platforma	Įrankio kūrėjo interneto svetainė	Komentaras
				staffing, schedules, risk and cost for software projects
15.	ePM.Ensemble	Windows	InventX http://www.inventx.com	Support effort estimation, among other things.
16.	PRICE S		Price Systems http://www.pricesystems.com	
17.	CostXpert		Marotz, Inc. http://www.costxpert.com	COCOMO
18.	USC COCOMO II		USC Center for Software Engineering http://sunset.usc.edu/research/COCOMOII/index.html	COCOMO
19.	LEAP	Windows, Unix, MacOS	University of Hawaii http://csdl.ics.hawaii.edu/Tools/LEAP	freeware
20.	Construx Estimate	Windows	Construx Software http://www.construx.com	freeware
21.	CoCoPro	Mac	Iconix Software Engineering, Inc. http://www.iconixsw.com	COCOMO
22.	SoftEST	Windows	SPAWAR Systems Center http://sepo.spawar.navy.mil/sepo/index2.html loacte "Process" and under that click on "Software Estimation"	Ad hoc (free)
23.	REVIC Cost Model	Windows	SPAWAR Systems Center http://sepo.spawar.navy.mil/sepo/index2.html loacte "Process" and under that click on "Software Estimation"	COCOMO (free)
24.	ACEIT (Automated Cost Estimation Tool)	Windows	http://www.aceit.com	Ad hoc
25.	SEAT (Software Estimation and Analysis Tool)	Windows	http://www.cs.umd.edu/RTSL/dsstud/dsstudio.htm	Ad hoc (free)