

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

Praktiskais darbs priekšmetā

„Programmatūras plānošana un metroloģija”

2. laboratorijas darbs

19. variants

Programmatūras izstrādes izmaksu novērtēšana

Izstrādāja: Sergejs Terentjevs

1. kurss, 1.grupa

apl. nr. 061rdb140

Pārbaudīja: lektore J. Bule

Rīga, 2010

Laboratorijas darba uzdevums

1. Izskaitļot programmatūras izstrādes izmaksas, lietojot trīs dažādus rīkus: *Costar*, *SEAT* un *COSMOS*. Novērtējamā programmatūra:

- Poliklīnikas vadības uzskaites sistēma (2 KLOC);
- Programmatūra, kuras koda garums sasniedz 30 KLOC;
- Programmatūra, kuras koda garums sasniedz 130 KLOC.

Projekta parametri:

- Analītiķa kvalifikācija ir zema ($ACAP = L$);
- Pieredze lietojumsfērā ir augsta ($AEXP = H$).

Salīdzināt un novērtēt iegūtos rezultātus.

2. Aprēķināt par cik procentiem palielināsies vai samazināsies darba apjoms, ja:

2.1. Pasūtītājs prasa pabeigt projektu 6 mēnešu ātrāk.

2.2. Ātrdarbības ierobežojumi ir augsti ($TIME = H$).

Saturs

1. Ievads	4
2. Rīku un modeļu apraksts	5
2.1. COCOMO modelis	5
2.2. COCOMO II modelis.....	7
2.3. Rīks <i>Costar</i>	9
2.4. Rīks <i>SEAT</i>	10
2.5. Rīks <i>COSMOS</i>	12
3. Novērtējamas programmatūras apraksts	14
3.1. Sistēmas īss apraksts	14
3.2. Sistēmas arhitektūra	15
4. Programmatūras projektu izmaksu novērtēšanas rezultāti	17
5. Laboratorijas darba uzdevuma 2. punkta izpilde	22
6. Iegūto rezultātu analīze	24
7. Secinājumi	25
Literatūra	26

1. Ievads

Kvalitatīvas programmatūras izstrādei ir jāievēro noteikti likumi. Liela uzmanība jāpievērš programmatūras izstrādes plānošanai. Viena no plānošanas procesa sastāvdaļām ir programmatūras izstrādes izmaksu novērtēšana. Pēc noteiktām formulām, zinot kāda apjoma programmatūra ir sagaidāma, var aprēķināt darba izmaksas cilvēk-mēnešos, nepieciešamo izstrādes laiku un citus lielumus. Otrajā laboratorijas darbā tiks pētītas trīs dažāda lieluma programmatūras, dažādi programmatūras izmaksu novērtēšanas modeļi un trīs dažādi rīki. Darba ietvaros tiks aprēķinātas un analizētas izmaksas, mainot noteiktus programmatūras izstrādes parametrus.

2. Rīku un modeļu apraksts

Dotajā nodaļā tiek aprakstīti COCOMO un COCOMO II modeļi, ka arī tādi izmaksu novērtēšanas rīki kā *Costar*, *COSMOS* un *SEAT*.

2.1. COCOMO modelis

COCOMO modelis ir programmatūras projektu izmaksu novērtēšanas modelis, kas tika izstrādāts balstoties uz datu sakarību analīzi par vairākiem projektiem (pamatā ir regresijas formula, kas ļauj noteikt vēlamu rezultātu attiecība pret vairākiem faktoriem) [1, 2].

COCOMO modeli ir piedāvājis zinātnieks B. Boems, kas strādājis kompānijā TRW ir izpētījis 63 projektus [1]. Modelī izstrādātais izmaksu noteikšanas mehānisms tika balstīts uz projektu koda rindiņu garumu (KLOC) novērtējumu attiecība pret kādu noteiktu programmēšanas valodu un izmantojamo programmatūras izstrādes dzīves ciklu (tanī laikā, 1981. g., galvenokārt tika lietots ūdenskrituma iteratīvais modelis) [1].

COCOMO modelis iedalās trijās detalizējamās formās [1]:

- COCOMO bāzes modelis – labi piemērots ātrai projektu izmaksu noteikšanai, kurā netiek ievēroti tādi faktori kā darbinieku pieredze, aparatūras ierobežojumi, tehnoloģiju un izstrādes tehniku pielietošana u.tml.;
- COCOMO starpmodelis – paredz izmaksu noteikšanu, uzskaitot vairākus faktorus, kurus neievēro bāzes modelis.
- COCOMO detalizēts modelis – paredz detalizēto izmaksu noteikšanu, ievērojot minēto faktoru ietekmi uz katru atsevišķo izstrādes posmu.

COCOMO bāzes modelis ievieš vairākus koeficientus, kuru vērtības ir atkarīgas no koda rindiņu garumiem (2.1 tabulu) [3].

2.1 tabula

COCOMO bāzes modeļa koeficienti

Kategorija	Garums (KLOC)	a_1	b_1	a_2	b_2
I	≤ 50	2,4	1,05	2,5	0,38
II	$> 50, \leq 300$	3,0	1,12	2,5	0,35
III	> 300	3,6	1,2	2,5	0,32

Ar doto koeficientu palīdzību ir iespējams aprēķināt tādu projektu raksturlielumus kā [3]:

- Darba apjoms cilvēk-mēnešos (skat. 2.1 formulu):

$$C = a_1 * L^{b_1}, \quad (2.1)$$

kur a_1 un b_1 – koeficienti, kuru vērtības ir atkarīgas no noteiktas kategorijas;

L – projekta apjoms (KLOC).

- Izstrādes laiks mēnešos:

$$D = a_2 * C^{b_2}, \quad (2.2)$$

kur C – aprēķinātais darba apjoms.

Modelis piedāvā arī divas papildus formulas, kuras ļauj noteikt sekojošos projektu raksturlielumus:

- Darba ražīgums LOC/mēn.:

$$DR = l / C, \quad (2.3)$$

kur l – projekta apjoms (LOC).

- Izstrādātāju vidējais skaits:

$$IS = C / D \quad (2.4)$$

COCOMO starpmodeļī darba apjoma aprēķins tiek veikts ievērojot programmas garumu un vairākus izmaksu faktorus. Dotie faktori un to novērtējuma koeficienti ir uzskaitīti 2.3 tabulā. Modeļī lietojama formula darba apjoma noteikšanai ir izskatās sekojoši [1]:

$$C = a_i * L^{b_i} * EAF, \quad (2.5)$$

kur a_i un b_i – koeficienti, kuri ir uzskaitīti 2.2 tabulā;

EAF – projektam raksturīgo izmaksas faktoru kopējais reizinājums (2.3 tabula).

Savukārt, izstrādes laiks ir noteikts ar sekojošo formulu [1]:

$$D = a_i * C^{b_i} \quad (2.6)$$

2.2 tabula

COCOMO starpmodeļa koeficienti

Kategorija	Garums (KLOC)	a_i	b_i
I	≤ 50	3,2	1,05
II	$> 50, \leq 300$	3,0	1,12
III	> 300	2,8	1,2

2.2. COCOMO II modelis

COCOMO II ir uzlabots COCOMO modelis, kas ir izstrādāts jaunāko izstrādes procesu un dzīves ciklu atbalstām, piemēram, ļauj noteikt izmaksas projektam, kas tiek izstrādāts vadoties pēc IBM RUP metodoloģijas.

COCOMO II modelis ievieš vairākus projekta novērtējuma faktoros (*scale drivers*), kuri nosaka tādu projekta raksturlielumus kā izstrādei nepieciešamais laiks, arhitektūras sarežģītība, risku pārvaldība u.tml. Dotie faktori un to nozīme ir sekojoši [4]:

- Paredzamība (*precedentedness*) – nosaka izstrādes pieredzi, t.i. vai izstrādes komanda ir jau iepriekš darbojusies pie līdzīga projekta;
- Izstrādes elastība (*development flexibility*) – nosaka izvirzīto prasību noteikšanas formu (vai ir stingri noteiktas, vai arī vispārēji);
- Arhitektūras risks (*architecture / risk resolution*) – nosaka sistēmas arhitektūras noteikšanas pakāpi un kuras izmaiņas var radīt izpildes risku;
- Iesaistīto personu sadarbība (*team cohesion*) – nosaka sadarbības pakāpi starp pasūtītāju un izstrādes komandu;
- Izstrādes procesu briedums (*process maturity*) – nosaka iekšējo izstrādes procesa brieduma pakāpi, t.i. vai procesi ir haotiski, vai arī standartizēti un optimizēti.

Papildus esošiem faktoriem COCOMO II modelis, līdzīgi COCOMO starpmodelim, ievieš vairākus atsevišķo projekta procesu un komponentu izmaksu novērtēšanas faktoros (*cost drivers*), kuri ir uzskaitīti 2.3 tabulā (iekavas ir norādīti COCOMO II modeļa vērtības, kas atšķiras no COCOMO starpmodeļa vērtībām).

COCOMO II modelis paredz darba apjoma noteikšanu pēc sekojošas formulas [4]:

$$C = 2,94 * EAF * L^E, \quad (2.7)$$

kur *EAF* – izmaksas faktoru (*cost drivers*) kopējais reizinājums;

E – eksponente, kuras vērtība ir 1,0997 (tiek iegūta vadoties pēc novērtējuma faktoru vērtībām).

Savukārt, izstrādes laika aprēķināšanai tiek lietota sekojoša formula [4]:

$$D = 3,67 * C^{SE}, \quad (2.8)$$

kur *SE* – grafisko vienādojumu eksponente, kuras vērtība ir 0,3179.

COCOMO starpmodeļa un COCOMO II modeļa izmaksu koeficienti (*Cost Drivers*)

Faktori	Nosaukums	Ļoti zems	Zems	Nomināls	Augsts	Ļoti augsts	Ekstra augsts
Produkta	<u>Atribūti</u>						
RELY	Prasītā uzticamība	0.75	0.88	1.00	1.15	1.40 (1.39)	
DATA	Datu bāzes apjoms		0.94 (0.93)	1.00	1.08 (1.09)	1.16 (1.19)	
CPLX	Produkta sarežģītība	0.70 (0.75)	0.85 (0.88)	1.00	1.15	1.30	1.65 (1.66)
Aparatūras	<u>Atribūti</u>						
TIME	Ātrdarbības ierobežojumi			1.00	1.11	1.30 (1.31)	1.66 (1.67)
STOR	Atmiņas ierobežojumi			1.00	1.06	1.21	1.56 (1.57)
VIRT	Virtuālās mašīnas izmainīšanas iespēja		0.87	1.00	0.86	0.71	
TURN	Datoram griešanas cikls		0.87	1.00	1.07	1.15	
Personāla	<u>Atribūti</u>						
ACAP	Analītiķa kvalifikācija	1.46 (1.50)	1.19 (1.22)	1.00	0.86 (0.83)	0.71 (0.67)	
AEXP	Pieredze lietojumsfērā	1.29 (1.22)	1.13 (1.10)	1.00	0.91 (0.89)	0.82 (0.81)	
PCAP	Programmētāja kvalifikācija	1.42 (1.37)	1.17 (1.16)	1.00	0.86 (0.87)	0.70 (0.74)	
VEXP	Pieredze darbā ar operētājsistēmu	1.21	1.10	1.00	0.90		
LEXP (LTEX)	Pieredze darbā ar programmēšanas valodu	1.14 (1.22)	1.07 (1.10)	1.00	0.95 (0.91)	(0.84)	
Projekta	<u>Atribūti</u>						
MODP	Modernas tehnoloģijas lietošana	1.24	1.10	1.00	0.91	0.82	
TOOL	Programmatūras rīku izmantošana	1.24	1.10 (1.12)	1.00	0.91 (0.86)	0.83 (0.72)	
SCED	Izstrādes termiņa ierobežojumi	1.23 (1.29)	1.08 (1.10)	1.00	1.04 (1.00)	1.10 (1.00)	
COCOMO II	<u>papildus atribūti</u>						
RUSE	Atkārtotās lietošanas prasības		0.91	1.00	1.14	1.29	1.69
DOCU	Dokumentēšanas prasības	0.89	0.95	1.00	1.06	1.13	
PVOL	Platformas mainīgums		0.87	1.00	1.15	1.30	
PCON	Personāla maiņa	1.24	1.10	1.00	0.92	0.84	
PEXP	Platformas pieredze	1.25	1.12	1.00	0.88	0.81	
SITE	Tīmekļa programmatūra (<i>Multi-site dev.</i>)	1.25	1.10	1.00	0.92	0.84	0.78

2.3. Rīks *Costar*

Costar ir specializēts programmatūras izstrādes izmaksu novērtēšanas rīks, kas veic izmaksu novērtēšanu balstoties uz COCOMO modeli.

Costar programmas galvenais logs ir attēlots 2.1 attēlā. Tās iekļaut novērtējumu rezultātu sadaļu, kurā tiek aprēķināti sekojoši projekta atribūti:

- **Effort(PM)** – darba apjoms, cilvēk-mēnešos;
- **Duration(Mo)** – izstrādes laiks, mēnešos;
- **Cost(K\$)** – izmaksas, tūkstošos;
- **Productivity** – produktivitāte jeb darba ražīgums, LOC/mēn.

Costar - Sample (Component)

File View Reports Components Tools Preferences Help

Estimate: Sample ID: Model: COCOMO II 2000

Component: Component ID: Increment: 1

Totals for entire Project		Effort (PM)	Duration (Mo)	Cost (K\$)	Productivity	Equivalent Size
Requirements	RQ:	0.5	1.1	0.0		Total Size: 2,000
Development	PD+DD+CT+IT:	6.6	6.7	0.0	303.1	
Total	RQ+PD+DD+CT+IT:	7.1	7.8	0.0	283.3	

COCOMO II Cost Drivers for Component: Component

Personnel: ACAP... Low, APEX... High, PCAP... Nominal, PLEX... Nominal, LTEX... Nominal, PCON... Nominal

Platform: TIME... Nominal, STOR... Nominal, PVOL... Nominal

Product: RELY... Nominal, DATA... Nominal, CPLX... Nominal, RUSE... Nominal, DOCU... Nominal

Project: TOOL... Nominal, SITE... Nominal, SCED... Nominal

Size Summary: Size: 2000, Method: SLOC

User Defined: USR1... Undefined, USR2... Undefined, USR3... Undefined, USR4... Undefined

Drivers & Size / Model / REVL / Reuse / Function Points / Increments / Breakage / Costs / Rates / Maint / Filter / Descr.

Sample: 7.1 PM, 7.8 Months Component: 7.1 PM EAF: 1.0472 Level: 1

Modeļa izmaksu faktoru līmeņi

Novērtējumu rezultāti

2.1. att. *Costar* programmas galvenais logs.

Dotie projekta atribūti tiek aprēķināti automātiski, vadoties pēc sadaļā *Drivers&Size* norādītiem izmaksu faktoru līmeņiem, ka arī pēc norādīta koda rindiņu garuma (SLOC). Saskaņā ar uzdevuma definētiem parametriem, dotajā sadaļā pretī noteiktiem izmaksu faktoriem tiek definēti atbilstoši novērtējuma līmeņi, t.i., *ACAP = Low* un *APEX = High*.

Rīka nākama sadaļā, *Model*, ir iespējams definēt programmatūras projekta izstrādes dzīves ciklu (nosaka posmus, kuriem tiks novērtētas izmaksas) un specificēt novērtējuma faktoru (*scale drivers*) vērtības. Uzdevuma izpildei ir izvēlēts COCOMO II modelis un ūdenskrituma dzīves cikls, ka arī visam novērtējuma faktoru liemeņiem ir piešķirta nomināla (*nominal*) vērtība (skat. 2.2 attēlu).

2.2. att. Sadaļa *Model*.

2.4. Rīks *SEAT*

SEAT rīks veic programmatūras izmaksu aprēķināšanu ar COCOMO starpmodeļa palīdzību. Dota mērķa sasniegšanai, rīks piedāvā divas formas.

COCOMO Effort Multipliers formā tiek specificēti izmaksu faktoru līmeņi, balstoties uz kuriem tiek aprēķināts kopējais darba apjoma reizinājums (*total effort multipliers*). Dota forma ir attēlota 2.3 attēlā.

COCOMO Estimates formā tiek aprēķināts projekta nominālais darba apjoms (*nominal PM*), izstrādes darba apjoms (*development PM*) un izstrādes laiks (*TDEV*). Doto projekta raksturlielumu vērtības tiek aprēķinātas, pamatojoties uz laukā *KSLOC* norādīto koda rindiņu garumu tūkstošos,

izvēlēto COCOMO modeļa kategoriju un aprēķināto darba apjoma reizinājumu. Dota forma ir attēlota 2.4 attēlā.

COCOMO Effort Multipliers

Product Attributes	Computer Attributes	Personnel Attributes	Project Attributes
Reliability: Nominal	Execution Time: Nominal	Analyst Capability : Low	Modern Prog Practice: Nominal
Database Size: Nominal	Storage Size: Nominal	App Experience: High	Software Tools: Nominal
Product Complexity: Nominal	Virt Mach Volatility: Nominal	Programmer Capability: Nominal	Req'd Schedule: Nominal
	Comp Turnaround Time: Nominal	Virt Mach Exp: Nominal	
		Lang Exp: Nominal	

Total Effort Multipliers (T E M) = 1.08

Buttons: Fun Count, Proc Comp, FPA Info, Effort Mult, Estimates, Cocomo Info, Project Info

2.3. att. Forma *COCOMO Effort Multipliers*.

COCOMO Estimates

Estimated Nominal Effort

EFFORT FACTOR	KSLOC	EFFORT EXPONENT	NOMINAL PM
3.20	2	exp 1.05	6.63

(Not Linked)

COCOMO Development: **Organic**

Estimated Development Effort

NOMINAL PM	TEM	DEVELOPMENTAL PM
6.63	X 1.08	7.16

KSLOC = thousands (K) of Source Line Of Code
TEM = Total Effort Multipliers
PM = Person Months
TDEV = Time to DEVELOP(Calendar Months)

Recommended Development Schedule

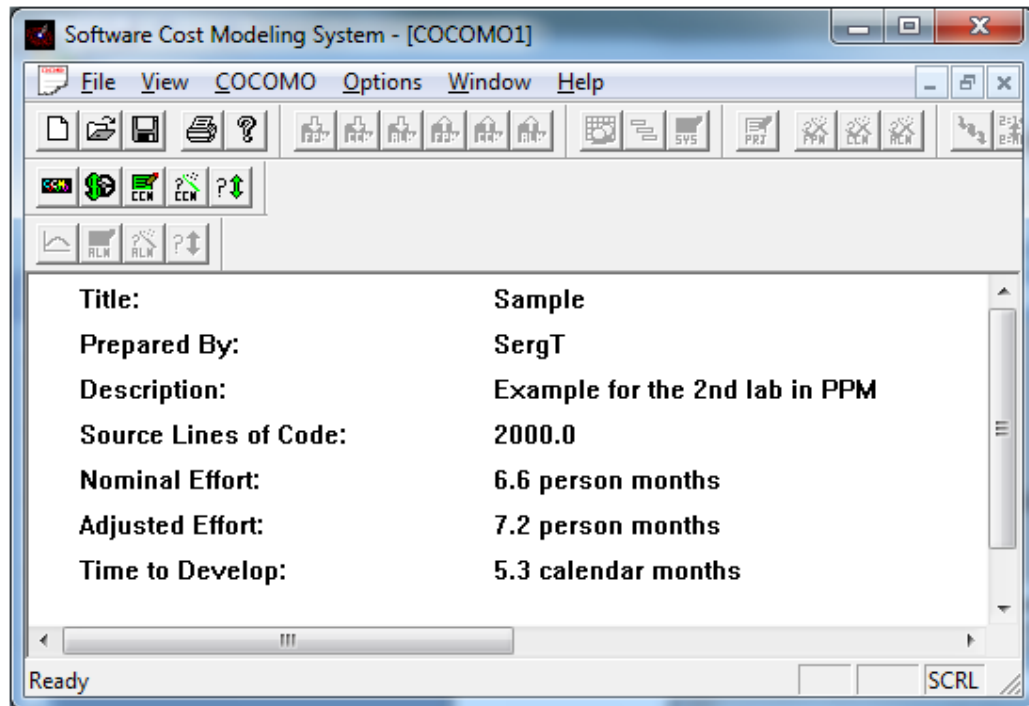
SCHEDULE FACTOR	DEVELOPMENTAL PM	SCHEDULE EXPONENT	TDEV
2.50	X 7.16	exp 0.38	5.28

Buttons: Fun Count, Proc Comp, FPA Info, Effort Mult, Estimates, Cocomo Info, Project Info


2.4. att. Forma *COCOMO Estimates*.

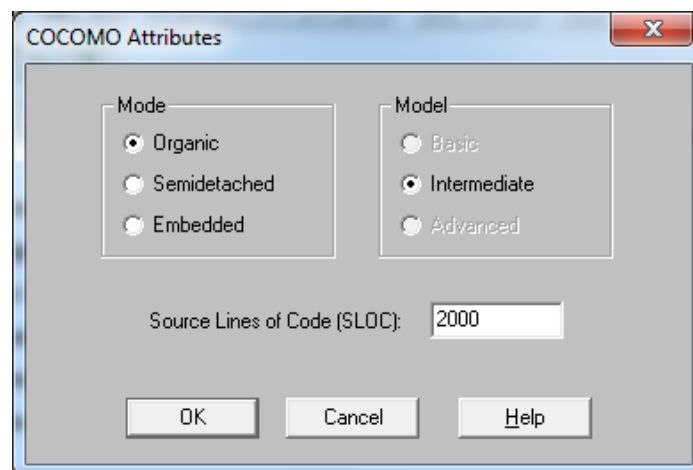
2.5. Rīks *COSMOS*

COSMOS rīks projektu izmaksu aprēķināšanai pielieto COCOMO starpmodeli. Dota rīka galvenā forma ir attēlota 2.5 attēlā.




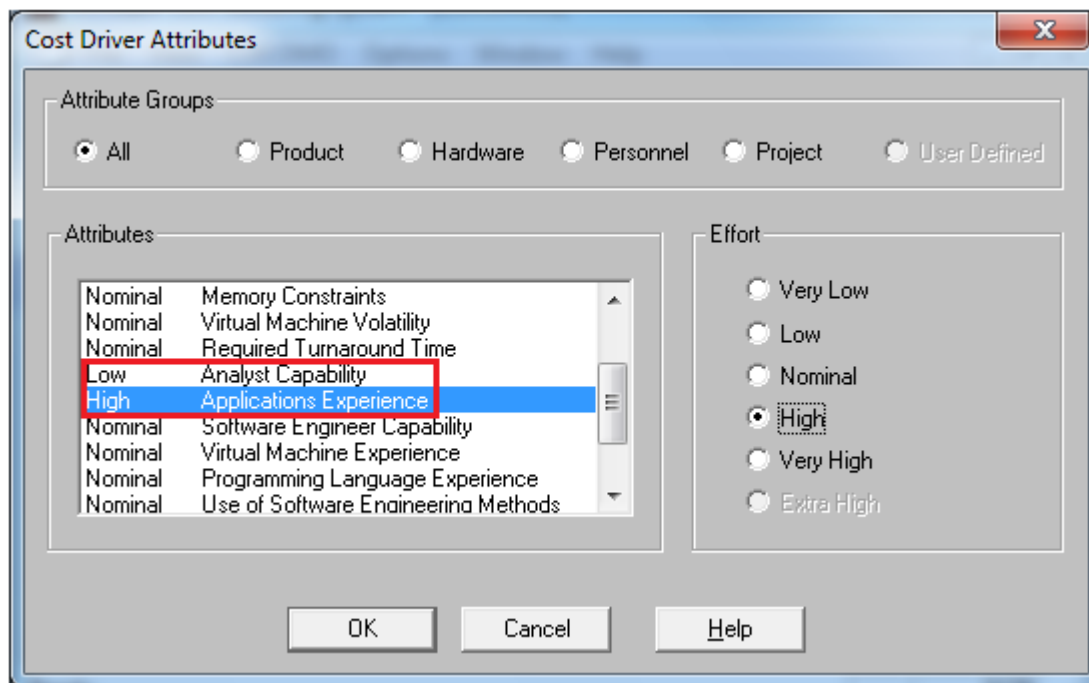
2.5. att. Rīks *COSMOS*.

Rīks piedāvā *Attributes* pogu (), ar kuras palīdzību atvērta logā *COCOMO Attributes* var iestatīt nepieciešamo koda rindiņu garumu, izvēlēties COCOMO modeļa kategoriju un formu (skat. 2.6 attēlu).



2.6. Logs *COCOMO Attributes*.

Rīks piedāvā izmaksu faktoru līmeņu definēšanas iespējas. Dotam nolūkam tiek paredzēta poga *Cost Drivers* (), kuras nospiešanas rezultātā tiek atvērta *Cost Driver Attributes* forma (skat. att. 2.7).



Cost Driver Attributes

Attribute Groups

☒ All ☐ Product ☐ Hardware ☐ Personnel ☐ Project ☐ User Defined

Attributes

Nominal	Memory Constraints
Nominal	Virtual Machine Volatility
Nominal	Required Turnaround Time
Low	Analyst Capability
High	Applications Experience
Nominal	Software Engineer Capability
Nominal	Virtual Machine Experience
Nominal	Programming Language Experience
Nominal	Use of Software Engineering Methods

Effort

☐ Very Low
☐ Low
☐ Nominal
☒ High
☐ Very High
☐ Extra High

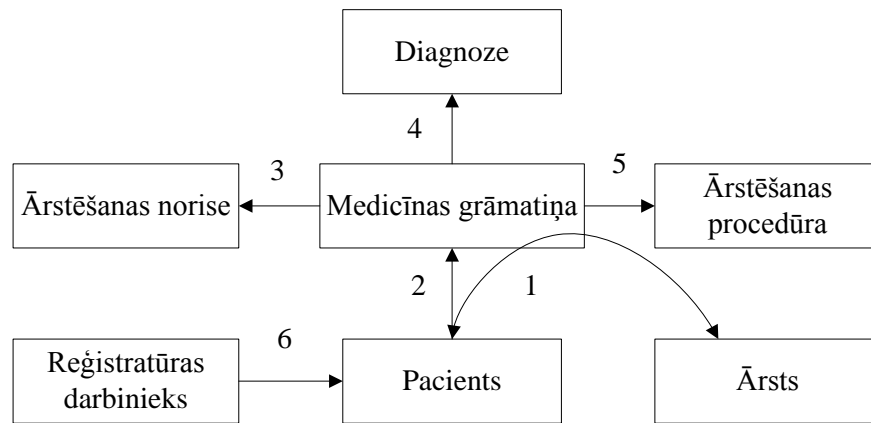
OK Cancel Help

2.7. Forma *Cost Driver Attributes*.

3. Novērtējamas programmatūras apraksts

3.1. Sistēmas īss apraksts

Praktiska darba izpildei ir izvēlēta poliklīnikas vadības uzskaites sistēma. Sistēmas mērķis ir vadīt poliklīnikas iekšējas informācijas un datu apriti, kas ir saistīta ar jauno pacientu, to slimību un ārstēšanas procedūru uzskaiti, ka arī ar personāla iekšējo datu vadību. Pašreizējās sistēmas darbības sfēras attēlojums ir parādīts 3.1. attēla.



3.1. att. Vienkāršotais sistēmas modelis.

Modelī definēto saišu nozīme ir sekojoša:

1. Ārsts apskata pacientu un pacients saņem no ārsta informāciju par viņa veselības stāvokli un turpmākiem ārstēšanas norīkojumiem. Tās viss tiek fiksēts medicīnas grāmatiņā.
2. Katrām pacientam ir sava medicīnas grāmatiņa.
3. Medicīnas grāmatiņā tiek aprakstīta ārstēšanas norisē.
4. Medicīnas grāmatiņā tiek norādīta nostādīta diagnoze.
5. Medicīniska grāmatiņa tiek norādīta ārstēšanas procedūra (terapijas kurss).
6. Reģistratūras darbinieks veic pacienta personīgo datu ievadi.

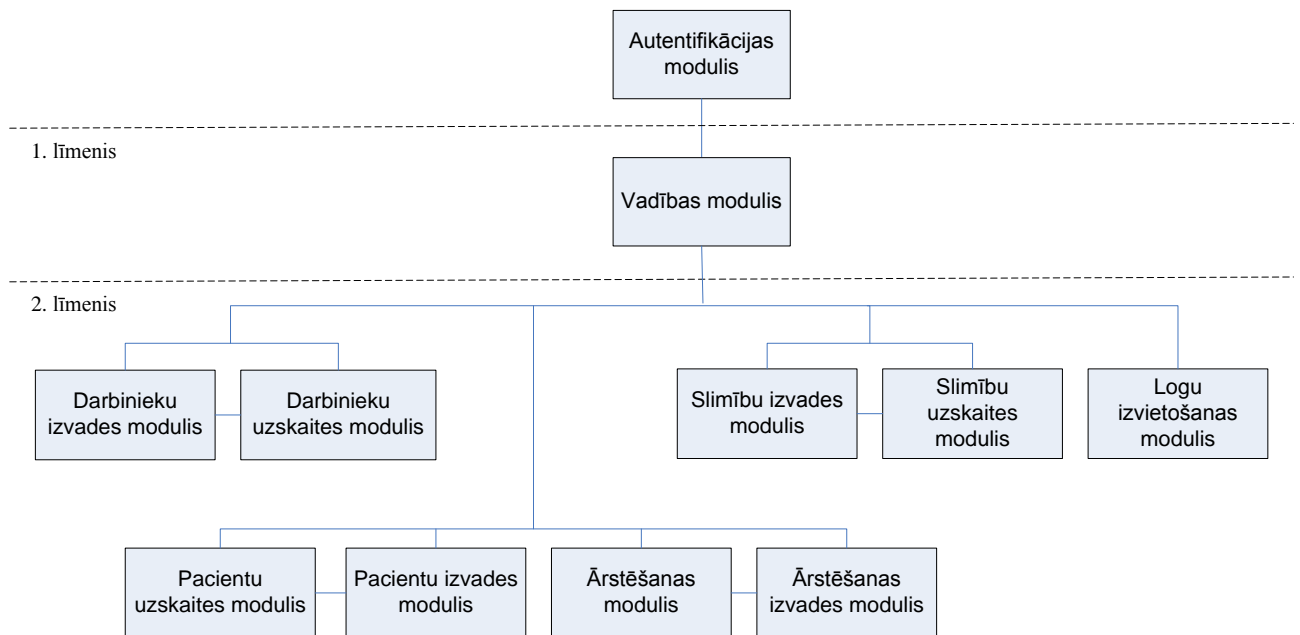
Datorizētas informācijas sistēmas uzdevums ir aizvietot medicīnas grāmatiņu. Saskaņā ar šo uzdevumu un definēto mērķi sistēmai ir jāparedz šādas funkcijas:

- Pacientu un personāla (ārsts, reģistratūras darbinieks un administrators) datu reģistrēšanu un vadību;
- Pacientu norīkošanas iespējas konkrētiem ārstiem;

- Ārstēšanas procedūru uzskaites iespējas, norādot nostādīto diagnozi un fiksējot visus nepieciešamos līdzekļus, preparātus, procedūras pacientu slimību novēršanai, ka arī ārstēšanas norises gaitu;
- Lietotāju autentifikāciju un atbilstošo pieejas tiesību nodrošināšanu.

3.2. Sistēmas arhitektūra

Balstoties uz definētam prasībām un funkcionalitāti, sistēmai tiek plānots paredzēt 3.2 attēlā parādītos moduļus.



3.2. att. Sistēmas hierarhiska diagramma.

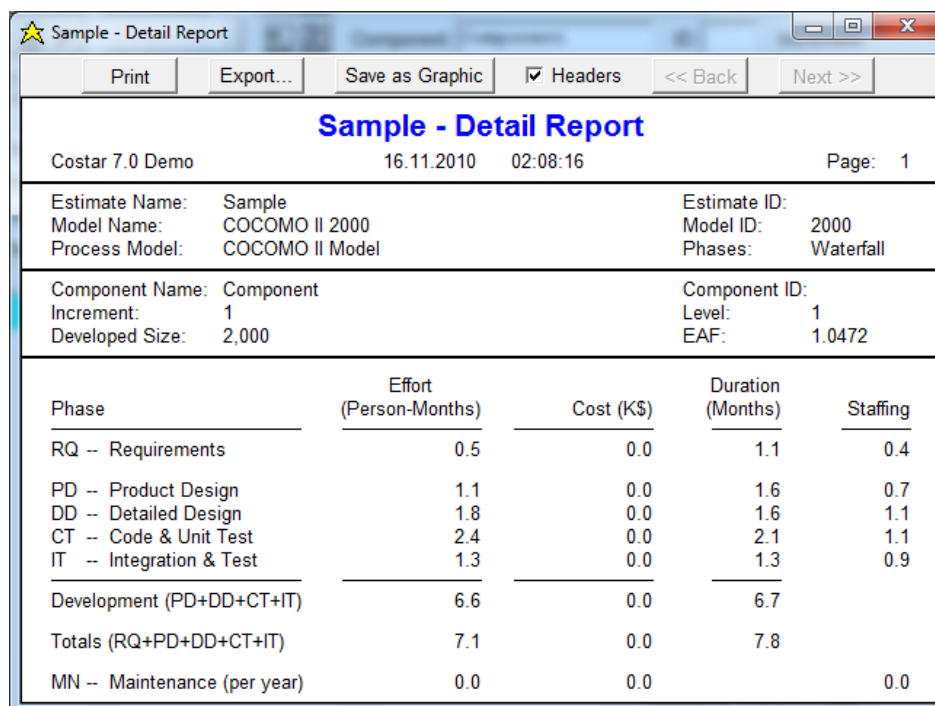
Doto moduļu nozīme ir sekojoša:

- Autentifikācijas modulis – pilda lietotāju pārbaudes funkciju, ļaujot darboties ar sistēmu tikai reģistrētiem lietotājiem.
- Vadības modulis – atkarībā no autorizēta lietotāja tipa piedāvā tikai viņam atļautas pieejamas darbības;
- Pacientu izvades modelis – pilda pacientu datu izvades funkciju, piedāvājot datu atlases un kārtošanas operācijas;
- Pacientu uzskaites modulis – piedāvā pacientu reģistrēšanas un norīkošanas iespējas;

- Darbinieku izvades modulis – pilda personāla datu izvades funkciju, piedāvājot datu atlases un kārtošanas operācijas;
- Darbinieku uzskaites modulis - piedāvā ārstu, administratora un reģistratūras darbinieku reģistrēšanas iespējas, ka arī nodrošina personāla datu labošanas funkciju;
- Ārstēšanas modulis – piedāvā diagnožu ievadi un slimību apraksta iespējas, paredz nepieciešamo procedūru, medikamentu un preparātu fiksēšanu, ar kuru palīdzību tiks ārstēts pacients;
- Ārstēšanas izvades modulis – paredz pacientu slimību vēstures atspoguļošanu;
- Slimību uzskaites modulis – piedāvā slimību uzskaites iespējas, ka arī nodrošina reģistrēto datu labošanu;
- Slimību izvades modulis – pilda datu par slimībām izvadi, piedāvājot datu kārtošanas un grupēšanas operācijas;
- Logu izvietojšanas modulis – nosaka lietotāju saskarni, piedāvājot aktīvo formu (logu) izkārtošanas funkcijas.

4. Programmatūras projektu izmaksu novērtēšanas rezultāti

Costar rīka iegūtie rezultāti ir parādīti 4.1, 4.2 un 4.3 attēlos. Iegūto rezultātu izvadei tika pielietots *Detail Report* atskaišu tips.



Sample - Detail Report

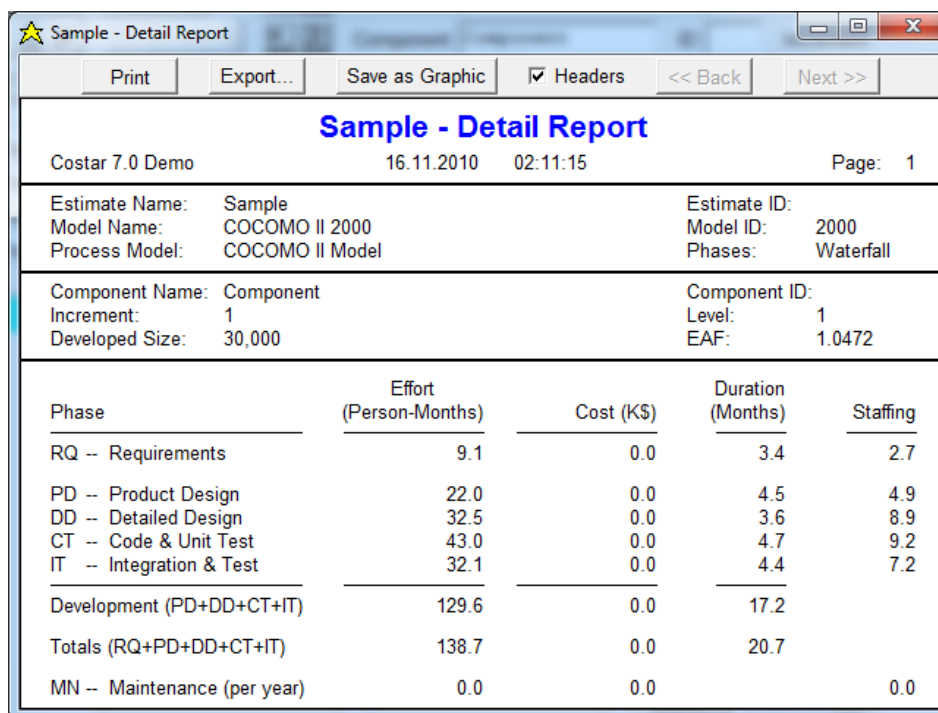
Costar 7.0 Demo 16.11.2010 02:08:16 Page: 1

Estimate Name:	Sample	Estimate ID:	
Model Name:	COCOMO II 2000	Model ID:	2000
Process Model:	COCOMO II Model	Phases:	Waterfall

Component Name:	Component	Component ID:	
Increment:	1	Level:	1
Developed Size:	2,000	EAF:	1.0472

Phase	Effort (Person-Months)	Cost (K\$)	Duration (Months)	Staffing
RQ -- Requirements	0.5	0.0	1.1	0.4
PD -- Product Design	1.1	0.0	1.6	0.7
DD -- Detailed Design	1.8	0.0	1.6	1.1
CT -- Code & Unit Test	2.4	0.0	2.1	1.1
IT -- Integration & Test	1.3	0.0	1.3	0.9
Development (PD+DD+CT+IT)	6.6	0.0	6.7	
Totals (RQ+PD+DD+CT+IT)	7.1	0.0	7.8	
MN -- Maintenance (per year)	0.0	0.0		0.0

4.1. att. Iegūti rezultāti pie 2,000 koda rindiņām.



Sample - Detail Report

Costar 7.0 Demo 16.11.2010 02:11:15 Page: 1

Estimate Name:	Sample	Estimate ID:	
Model Name:	COCOMO II 2000	Model ID:	2000
Process Model:	COCOMO II Model	Phases:	Waterfall

Component Name:	Component	Component ID:	
Increment:	1	Level:	1
Developed Size:	30,000	EAF:	1.0472

Phase	Effort (Person-Months)	Cost (K\$)	Duration (Months)	Staffing
RQ -- Requirements	9.1	0.0	3.4	2.7
PD -- Product Design	22.0	0.0	4.5	4.9
DD -- Detailed Design	32.5	0.0	3.6	8.9
CT -- Code & Unit Test	43.0	0.0	4.7	9.2
IT -- Integration & Test	32.1	0.0	4.4	7.2
Development (PD+DD+CT+IT)	129.6	0.0	17.2	
Totals (RQ+PD+DD+CT+IT)	138.7	0.0	20.7	
MN -- Maintenance (per year)	0.0	0.0		0.0

4.2. att. Iegūti rezultāti pie 30,000 koda rindiņām.

<div> Print Export... Save as Graphic <input checked="" type="checkbox"/> Headers << Back Next >> </div>				
<div> <div>Sample - Detail Report</div> <div> <div>Costar 7.0 Demo</div> <div>16.11.2010 02:22:13</div> <div>Page: 1</div> </div> </div>				
Estimate Name: Sample Model Name: COCOMO II 2000 Process Model: COCOMO II Model		Estimate ID: Model ID: 2000 Phases: Waterfall		
Component Name: Component Increment: 1 Developed Size: 130,000		Component ID: Level: 1 EAF: 1.0472		
Phase	Effort (Person-Months)	Cost (K\$)	Duration (Months)	Staffing
RQ -- Requirements	45.5	0.0	6.3	7.2
PD -- Product Design	110.5	0.0	7.8	14.2
DD -- Detailed Design	156.1	0.0	5.5	28.5
CT -- Code & Unit Test	201.6	0.0	7.2	28.0
IT -- Integration & Test	182.1	0.0	8.3	21.8
Development (PD+DD+CT+IT)	650.3	0.0	28.8	
Totals (RQ+PD+DD+CT+IT)	695.8	0.0	35.1	
MN -- Maintenance (per year)	0.0	0.0		0.0

4.3. att. Iegūti rezultāti pie 130,000 koda rindiņām.

SEAT rīka iegūtie rezultāti ir attēloti 4.4, 4.5 un 4.6 attēlos.

COCOMO Estimates

Estimated Nominal Effort

EFFORT FACTOR

3.20

X

KSLOC

2,000

EFFORT EXPONENT

exp

1.05

=

NOMINAL PM

6.63

(Not Linked)

COCOMO Development

Organic

Estimated Development Effort

NOMINAL PM

6.63

X

T E M

1.08

=

DEVELOPMENTAL PM

7.16

KSLOC = thousands (K) of Source Line Of Code

T E M = Total Effort Multipliers

PM = Person Months

TDEV = Time to DEvelop(Calendar Months)

Recommended Development Schedule

SCHEDULE FACTOR

2.50

X

DEVELOPMENTAL PM

7.16

exp

SCHEDULE EXPONENT

0.38

=

TDEV

5.28

Fun Count

Proc Comp

FPA Info

Effort Mult

Estimates

Cocomo Info

Project Info

4.4. att. Iegūti rezultāti pie 2,000 koda rindiņām.

COCOMO Estimates

Estimated Nominal Effort

EFFORT FACTOR		KSLOC		EFFORT EXPONENT		NOMINAL PM
3.20	X	30.00	exp	1.05	=	113.80
(Not Linked)						

COCOMO Development
Organic

Estimated Development Effort

NOMINAL PM		T E M		DEVELOPMENTAL PM
113.80	X	1.08	=	122.90

KSLOC = thousands (K) of Source Line Of Code
T E M = Total Effort Multipliers
PM = Person Months
TDEV = Time to DEvelop(Calendar Months)

Recommended Development Schedule

SCHEDULE FACTOR		DEVELOPMENTAL PM		SCHEDULE EXPONENT		TDEV
2.50	X	122.90	exp	0.38	=	15.56

Fun Count Proc Comp FPA Info **Effort Mult** Estimates Cocomo Info Project Info

4.5. att. Iegūti rezultāti pie 30,000 koda rindiņām.

COCOMO Estimates

Estimated Nominal Effort

EFFORT FACTOR		KSLOC		EFFORT EXPONENT		NOMINAL PM
3.20	X	130	exp	1.05	=	530.63
(Not Linked)						

COCOMO Development
Organic

Estimated Development Effort

NOMINAL PM		T E M		DEVELOPMENTAL PM
530.63	X	1.08	=	573.08

KSLOC = thousands (K) of Source Line Of Code
T E M = Total Effort Multipliers
PM = Person Months
TDEV = Time to DEvelop(Calendar Months)

Recommended Development Schedule

SCHEDULE FACTOR		DEVELOPMENTAL PM		SCHEDULE EXPONENT		TDEV
2.50	X	573.08	exp	0.38	=	27.93

Fun Count Proc Comp FPA Info **Effort Mult** Estimates Cocomo Info Project Info

4.6. att. Iegūti rezultāti pie 130,000 koda rindiņām.

COSMOS rīka iegūtie rezultāti ir attēloti 4.7, 4.8 un 4.9 attēlos.

The screenshot shows a window titled 'COCOMO1' with a list of project metrics. The metrics are as follows:

Title:	Sample
Prepared By:	
Description:	
Source Lines of Code:	2000.0
Nominal Effort:	6.6 person months
Adjusted Effort:	7.2 person months
Time to Develop:	5.3 calendar months

4.7. att. Iegūti rezultāti pie 2,000 koda rindiņām.

The screenshot shows a window titled 'COCOMO1' with a list of project metrics. The metrics are as follows:

Title:	Sample
Prepared By:	
Description:	
Source Lines of Code:	30000.0
Nominal Effort:	113.8 person months
Adjusted Effort:	123.2 person months
Time to Develop:	15.6 calendar months

4.8. att. Iegūti rezultāti pie 30,000 koda rindiņām.

The screenshot shows a window titled 'COCOMO1' with a list of project metrics. The metrics are as follows:

Title:	Sample
Prepared By:	
Description:	
Source Lines of Code:	130000.0
Nominal Effort:	530.6 person months
Adjusted Effort:	574.6 person months
Time to Develop:	28.0 calendar months

4.9. att. Iegūti rezultāti pie 130,000 koda rindiņām.

Iegūtie rezultāti ir apkopoti 4.1 tabulā.

4.1 tabula

Programmatūras izmaksas, lietojot trīs dažādus rīkus

Programmatūra	Garums, KLOC	Costar		SEAT		COSMOS	
		Darba apjoms	Izstrādes laiks	Darba apjoms	Izstrādes laiks	Darba apjoms	Izstrādes laiks
1. projekts	2	7,1	7,8	7,16	5,28	7,2	5,3
2. projekts	30	138,7	20,7	122,9	15,56	123,2	15,6
3. projekts	130	695,8	35,1	573,08	27,93	574,6	28

5. Laboratorijas darba uzdevuma 2. punkta izpilde

1. Pasūtītājs prasa pabeigt projektu 6 mēnešu ātrāk:

Pirmā punkta izpildei ir izvēlēts *Costar* rīka iegūtais izstrādes laiks, jo doto aprēķinu pamatā ir pielietots COCOMO II modelis. Dotais modelis apskata vairākus papildus faktorus (*scale drivers*) un veic izmaksu novērtēšanu katram dzīves cikla modeļa posmam, ka arī ir labāk piemērots mūsdienu izstrādes procesu novērtēšanai.

Pirmā projekta darba apjoma aprēķins:

- $k = (7.8 - 6) / 7.8 = 1.8 / 7.8 = 0.23$;
- Par cik pieļaujama vērtība ir robežās no 0,75 līdz 1,6, tad projektu ātrāk par 6 mēnešiem nav iespējams pabeigt.

Otrā projekta darba apjoma aprēķins:

- $k = (20.7 - 6) / 20.7 = 0.71$;
- Projektu nav iespējams pabeigt ātrāk par 6 mēnešiem.

Trešā projekta darba apjoma aprēķins:

- $k = (35.1 - 6) / 35.1 = 0.83$;
- SCED $k_{low} = 1.08$;
- Darba apjoms palielināsies par 8%.

2. Ātrdarbības ierobežojumi ir augsti ($TIME = H$):

Uzdevuma izpildei tika pielietots *Costar* rīks, kurā atbilstoši uzdevuma nosacījumiem tika definēti 5.1 attēlā definēti izmaksu faktoru līmeņi.

The screenshot shows the 'Personnel' and 'Platform' sections of the Costar tool. The 'Personnel' section lists six factors: ACAP... (Low), APEX... (High), PCAP... (Nominal), PLEX... (Nominal), LTEX... (Nominal), and PCON... (Nominal). The 'Platform' section lists three factors: TIME... (High), STOR... (Nominal), and PVOL... (Nominal). Each factor has a corresponding dropdown menu with up and down arrows.

Category	Factor	Level
Personnel	ACAP...	Low
	APEX...	High
	PCAP...	Nominal
	PLEX...	Nominal
	LTEX...	Nominal
	PCON...	Nominal
Platform	TIME...	High
	STOR...	Nominal
	PVOL...	Nominal

5.1. att. Izmaksu faktoru iestatījumi.

Iegūtie rezultāti, definējot augstu ātrdarbības ierobežojuma līmeni, ir attēloti 5.2, 5.3 un 5.4 attēlos.

Totals for entire Project	Effort (PM)	Duration (Mo)	Cost (K\$)	Productivity	Equivalent Size
Requirements RQ:	0.5	1.1	0.0		Total Size: 2,000
Development PD+DD+CT+IT:	7.3	6.9	0.0	273.1	
Total RQ+PD+DD+CT+IT:	7.8	8.0	0.0	255.2	

5.2. att. Iegūti rezultāti pie 2,000 koda rindiņām.

Totals for entire Project	Effort (PM)	Duration (Mo)	Cost (K\$)	Productivity	Equivalent Size
Requirements RQ:	10.1	3.5	0.0		Total Size: 30,000
Development PD+DD+CT+IT:	143.9	17.8	0.0	208.5	
Total RQ+PD+DD+CT+IT:	154.0	21.3	0.0	194.8	

5.3. att. Iegūti rezultāti pie 30,000 koda rindiņām.

Totals for entire Project	Effort (PM)	Duration (Mo)	Cost (K\$)	Productivity	Equivalent Size
Requirements RQ:	50.5	6.5	0.0		Total Size: 130,000
Development PD+DD+CT+IT:	721.8	29.7	0.0	180.1	
Total RQ+PD+DD+CT+IT:	772.3	36.3	0.0	168.3	

5.4. att. Iegūti rezultāti pie 130,000 koda rindiņām.

Iepriekš iegūto un jauniegūto rezultātu apkopojums ir sastādīts 5.1 tabulā, kur:

- Cj – jaunais darba apjoms;
- Cv – iepriekšējais darba apjoms;
- Dj – jaunais izstrādes laiks;
- Dv – iepriekšējais izstrādes laiks.

5.1 tabula

Iepriekš iegūto un jauniegūto rezultātu salīdzinājums

Programma	Garums, KLOC	Costar					
		Vecais darba apjoms	Jaunais darba apjoms	Attiecība $k = C_j/C_v$	Vecais izstrādes laiks	Jaunais izstrādes laiks	Attiecība D_j/D_v
1. projekts	2	7.1	7.8	1.1	7.8	8.0	1.03
2. projekts	30	138.7	154.0	1.1	20.7	21.3	1.03
3. projekts	130	695.8	772.3	1.1	35.1	36.3	1.03

6. Iegūto rezultātu analīze

Praktiska darba izpildes procesā ir iegūti divi atšķirīgi rezultāti, kas tiek pamatots ar divu atšķirīgu COCOMO modeļu lietošanu izstrādes izmaksu novērtēšanai.

Rīku *SEAT* un *COSMOS* aprēķini ir veikti balstoties uz COCOMO starpmodeļi, kas darba apjoma noteikšanai lieto 2.5 formulu un izstrādes laika noteikšanai – 2.6 formulu. Dotas formulas atšķirībā no bāzes COCOMO modeļa formulām, ievēro arī vairākus izmaksu faktorus, uz kuru pamatā tiek konstruēta *EAF* mainīga vērtība. Dotais mainīgais ietekme uz kopējo darba apjoma un izstrādes laika noteikšanu, šādi piedāvājot detalizētāko pieeju izmaksu novērtēšanai. Taču šāda pieeja ir labi piemērota iepriekš lietotiem paņēmieniem programmatūras izstrādē (ūdens krituma modelim u.tml.), kuri pārsvara vairs netiek pielietoti dotiem nolūkiem. Rīku *SEAT* un *COSMOS* iegūtie rezultāti ir analogiski, būtiskās atšķirības ir novērojamas noapaļošanas ziņā.

Rīka *Costar* iegūti rezultāti ir balstīti uz COCOMO II modeli, kas papildus izmaksu faktoriem (*cost drivers*) ievēro arī novērtējuma faktorus (*scale drivers*), kas sniedz vēl detalizētāku pieeju izmaksu novērtēšanā (skat. 2.7 un 2.8 formulas). Atšķirīgs rezultāts ir iegūts arī dēļ izmaksu faktoru koeficientu atšķirībām, ka arī modelis veic katra atsevišķa izstrādes posma detalizēto novērtēšanu. COCOMO II modelis ir uzlabots COCOMO modelis, kas tika izstrādāts mūsdienu programmatūras dzīves ciklu procesu atbalstām.

No visiem rezultātiem, uzskatu, ka precīzākus rezultātus ir sniedzis *Costar* rīks. Jo tās ar COCOMO II modeļa palīdzību ir novērtējis arī izstrādes elastību, arhitektūras sarežģītību, sadarbības pakāpi starp izstrādātāju un pasūtītāju u.c. svarīgos faktorus, kurus *SEAT* un *COSMOS* rīki nav ņēmuši vērā.

SEAT un *COSMOS* rīku rezultātus es sniegtu klientam gadījumos, kad nebūtu lielas nepieciešamības pēc augstas detalizācijas pakāpes, t.i., rezultāti būtu domāti ātrai novērtēšanai paša projekta izstrādes sākumā. *Costar* rīka rezultāti tiktu sniegti klientam izmaksu detalizētai analīzei.

7. Secinājumi

Praktiska darba izpildes gaitā ir iegūts praktisks priekšstats programmatūras izmaksu novērtēšanā. Darba izpildei tika pielietoti *COSMOS*, *SEAT* un *Costar* rīki. No visiem rīkiem, viss intuitīvāki šķita *COSMOS* un *SEAT* rīki, jo piedāvā salīdzinoši maz iespēju izmaksu novērtējuma iestatīšanā salīdzinājumā ar *Costar* rīku. Ka arī sniedz mazāk rezultātu, t.i., neparedz darba ražīguma jeb produktivitātes un izmaksu kā tādu aprēķināšanu.

Atšķirība no pārējiem rīkiem, *Costar* rīks piedāvā izmaksu norādīšanu katram komandas dalībniekam, atskaišu izvades dažādību, atbalsta vairākus programmatūras izstrādes dzīves ciklus u.tml., kas raksturo to kā profesionālo un specializēto rīku programmatūras izmaksu novērtēšanas ziņā.

Darbs īpašās grūtības nesagādāja, tās tika izpildīts pusotras dienās laikā.

Literatūra

1. Wikipedia: COCOMO / Internets – <http://en.wikipedia.org/wiki/COCOMO>.
2. Wikipedia: Regression analysis / Internets – http://en.wikipedia.org/wiki/Regression_analysis.
3. Larisa Zaiceva, lekciju konspekts „Programmatūras metroloģijas un plānošanas modeļi” priekšmetā, 110 lpp, RTU 2010.
4. Overview of COCOMO / Internets - <http://www.softstarsystems.com/overview.htm>.