

Programmatūras projekta darbietilpības prognozēšanas metodes

Programmatūras risku analīze

DITF LDK

Problēmas

◆ *Izmantojamās informācijas kvalitāte*

- ◆ Maz lietderīgas informācijas
- ◆ Daļa prasību neskaidri formulētas

◆ *Prognožu precizitāte*

◆ *Prognožu darbības sfēra*

◆ *Rezultāti nav uzskatāmi par absolūtiem skaitļiem*

Neformālās metodes

- ◆ *Nepieciešama zināšanu bāze novērtējumam*
- ◆ *Pamatojumu trūkums sarunās ar programmatūras pasūtītāju*
- ◆ *Precīzi rezultāti maziem programmatūras projektiem*
 - ◆ *< 2 KLOC*
 - ◆ *<10 darbinieki, 3-6 mēneši*

Formālās metodes

- ◆ ***Iepriekšēja pieredze apjomu vērtēšanā ir vēlama, bet nav obligāti nepieciešama***
- ◆ ***Rezultāti vidējiem un lieliem projektiem***
 - ◆ Vidējs
 - ◆ 32 KLOC
 - ◆ 20-30 darbinieki, 1-2 gadi
 - ◆ Liels
 - ◆ 128 KLOC
 - ◆ 100-300 darbinieki, 3-5 gadi
- ◆ ***Pamatojums sarunās ar programmatūras pasūtītājiem***

Formālo metožu piemēri

- ◆ *Delphi*
- ◆ *Darbu strukturēšanas metode*
- ◆ *Checkpoint*
- ◆ *SLIM (Software Lifecycle Model)*
- ◆ *COCOMO*
- ◆ *COCOMO II*

Delphi metode

- ◆ ***Ekspertu grupa meklē vienotu viedokli par projekta darbietilpību***
- ◆ ***Ekspertu grupā ietilpst***
 - ◆ Vadītājs
 - ◆ DB, lietotāja interfisa u.c. izvēlēti speciālisti
- ◆ ***Darba gaita***
 - ◆ Ekspertu grupas sākotnējā diskusija
 - ◆ Darbietilpības novērtējuma formas aizpildīšana
 - ◆ Vadītājs apkopo rezultātus
 - ◆ Ekspertu grupa apspriež atšķirīgos rezultātus
 - ◆ Process ir iteratīvs, līdz tiek sasniegts ekspertu grupas vienots viedoklis

Delphi metodes priekšrocības un trūkumi

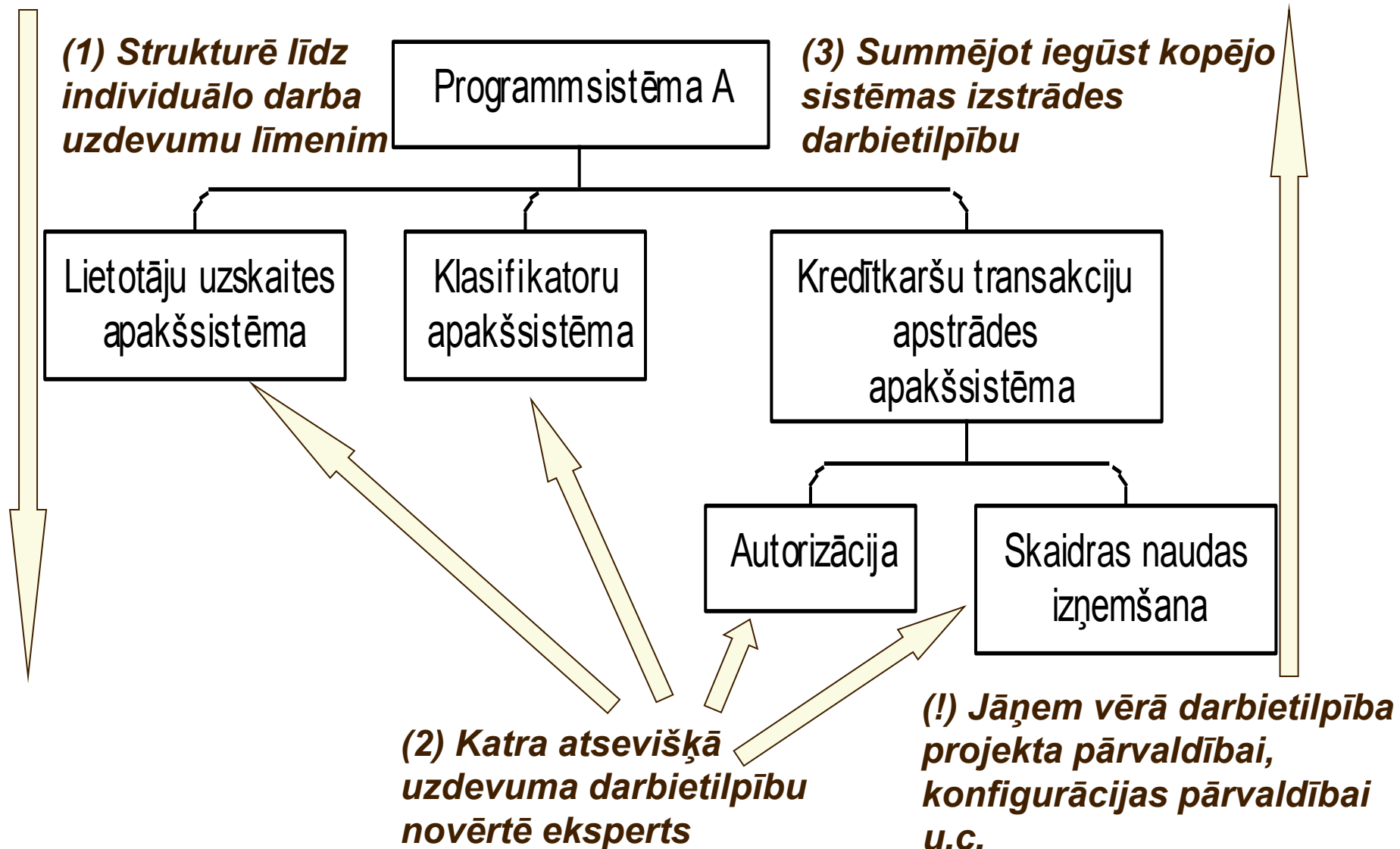
◆ ***Priekšrocības***

- ◆ Katram ekspertam ir daļējs priekšstats par kopējo darbietilpību un izstrādes laiku – metode apkopo vairākus viedokļus
- ◆ Var novērtēt ne tikai darbietilpību, bet arī riskus u.c. projekta izpildei kritiskas prasības

◆ ***Trūkumi***

- ◆ Apgrūtināta ekspertu grupas izveide
- ◆ Tiek ignorēta iepriekšējo projektu izstrādes pieredze

Darbu strukturēšanas metode



Darbu strukturēšanas metodes priekšrocības un trūkumi

◆ Priekšrocības

- ◆ Ļoti precīza metode, ja ir laba izstrādes procesa organizācija

◆ Trūkumi

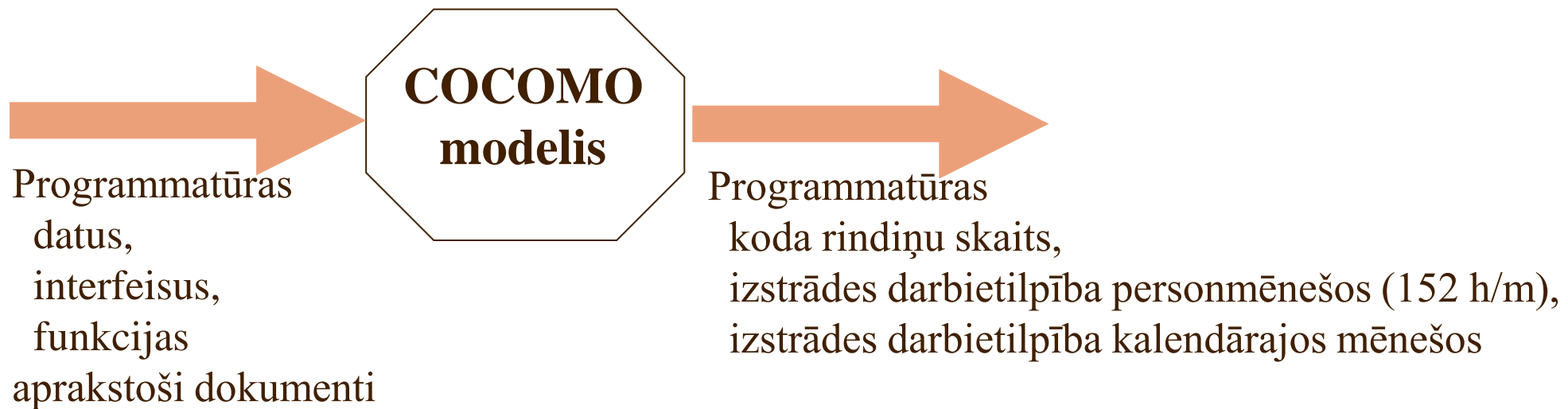
- ◆ Neder gadījumos, kad prasības nav skaidri formulētas
- ◆ Darbietilpīga, jo ietver arī plānošanu

Checkpoint metode

- ◆ *Bāze, kurā uzkrāta informācija par vairākiem tūkstošiem programmatūras izstrādes projektu (Software Productivity Research)*
 - ◆ Problēmu apgabals
 - ◆ Sistēmas izmērs
 - ◆ Izstrādes vide u.c.

COCOMO modelis

- ◆ *COCOMO - CO*nstructive *CO*st *MO*del
- ◆ Empīrisku sakarību kopums, kas iegūts apstrādājot reprezentatīvu statistisko materiālu



Darbietelpības novērtējuma sfēra



Programmatūras prasību izstrāde

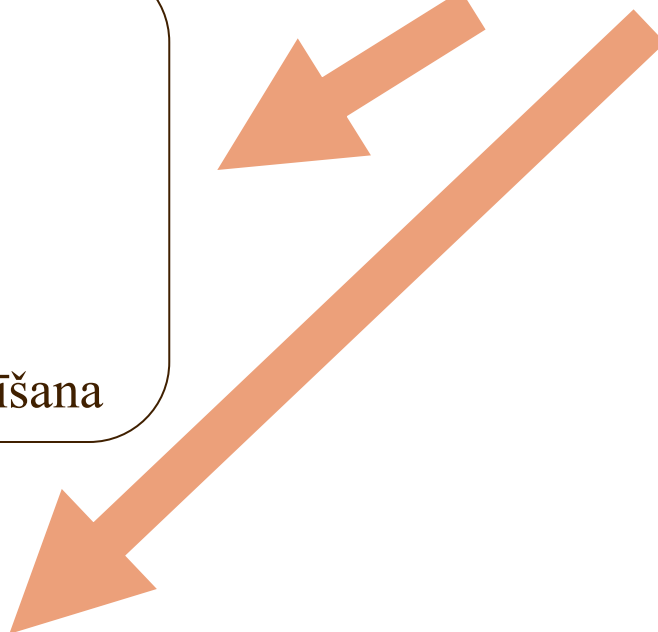
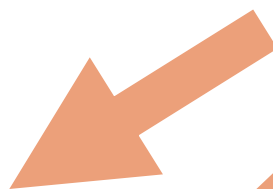
Projekta pārvaldība
Projekta dokumentēšana
Konfigurācijas pārvaldība
Programmatūras projektēšana
Programmēšana
Testēšana
Lietotāja dokumentācijas rakstīšana

Programmatūras instalācija
Apmācība

Programmatūras uzturēšana



**COCOMO vērtējuma rezultāti
attiecināmi uz**



COCOMO metodes soļi

◆ *Nepieskaņotu funkcijpunktu skaitīšana*

- ◆ Visatbildīgākais solis
- ◆ Nepieciešama kvalitatīva ieejas informācija

◆ *Programmrindiņu skaita iegūšana (LOC)*

◆ *Personmēnešu un kalendāro mēnešu skaita iegūšana*

Jēdzienu skaidrojums – funkcijs punkts (function point)

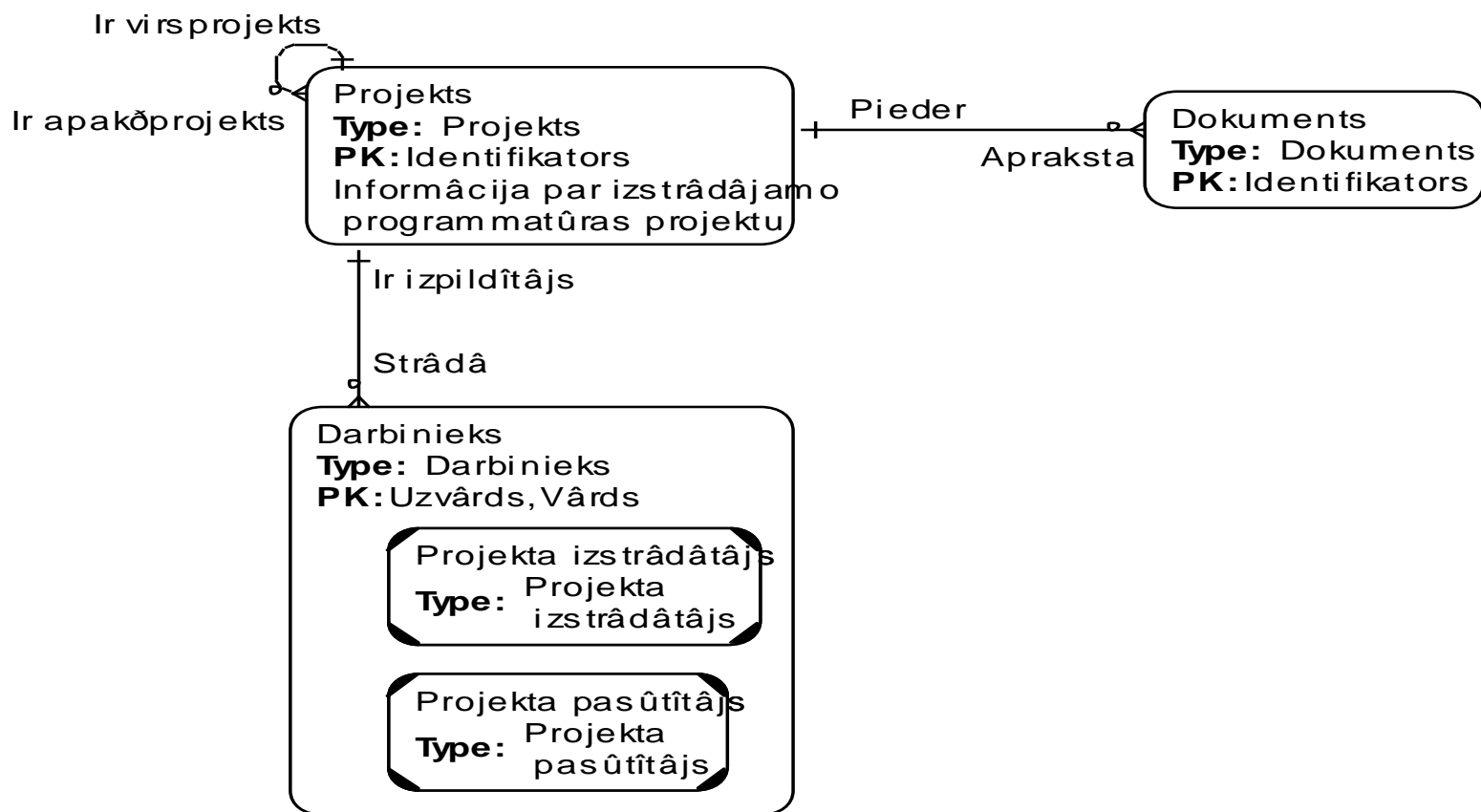
- ◆ *Sistēmas funkcionalitātes mērvienība*
- ◆ *Raksturo sistēmu no lietotāja viedokļa*
- ◆ *Funkcijs punktu skaits*
programmsistēmai nav atkarīgs no
programmatūras realizācijas vides

Nepieskaņotu funkcijpunktu skaits

	Sarežģīti	Vidēji	Vienkārši	Kopā
<i>Ievadi</i>	6 * skaits +	4 * skaits +	3 * skaits +	=
<i>Izvadi</i>	7 * skaits +	5 * skaits +	4 * skaits +	=
<i>Iekšēji datu faili</i>	15 * skaits +	10 * skaits +	7 * skaits +	=
<i>Ārēji interfeisa faili</i>	10 * skaits +	7 * skaits +	5 * skaits +	=
<i>Vaicājumi</i>	6 * skaits +	4 * skaits +	3 * skaits +	=
Nepieskaņotu funkcijpunktu skaits:				=

Jēdziena skaidrojums - iekšējs datu fails

- ◆ **Sistēmas iekšējais datu fails ir *savstarpēji loģiski saistītu datu kopa*, kuras elementus rada, labo un iznīcina sistēmas iekšienē**



lekšēju datu failu piemēri

- ◆ *Specifiskie sistēmas uzturētie dati.
Piemēram, informācija par darbinieku, ar kredītkarti veikto transakciju uzskaitījums u.c. specifiski sistēmā apstrādājami dati*
- ◆ *Sistēmas drošības, paroli, pārbaudi, HELP, kļūdu apstrādes u.c. dati un parametri, ko apstrādā sistēmas iekšienē*
- ◆ *Trasēšanas, kopēšanas u.c. vēsturiska informācija, ko sistēma īpaši apstrādā*

Jēdziena skaidrojums - ārējs interfeisa fails

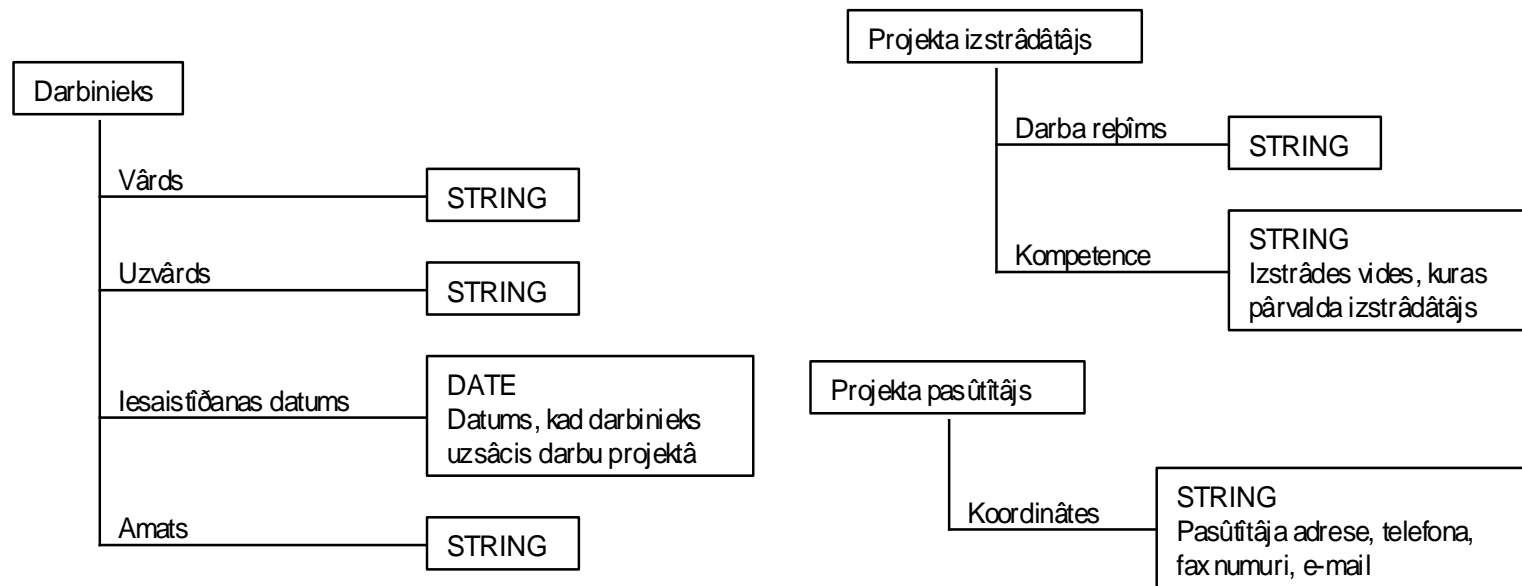
- ◆ ***Ārējā interfeisa fails ir savstarpēji loģiski saistītu datu kopa, no kuras vērtējamā sistēma nolasa datus, bet tos modificē cita sistēma***

Ārējo interfeisu failu piemēri

- ◆ *Dati, kas nolasīti no citas lietojumprogrammas. Piemēram, dati kurus vērtējamā sistēma nolasa no valsts nozīmes reģistriem*
- ◆ *Sistēmas drošības, paroli, pārbaužu, HELP, kļūdu apstrādes u.c. dati un parametri, ko apstrādā ārpus aplūkojamās sistēmas, bet kurus sistēma izmanto*
- ◆ *Trasēšanas, kopēšanas u.c. vēsturiska informācija, ko īpaši apstrādā cita sistēma, bet vērtējamā sistēma lasa*

lekšējo datu failu un ārējo interfeisu failu sarežģītība

◆ *Sarežģītības noteikšanai izmanto datu elementus un apakšelementus*



lekšējo datu failu un ārējo interfeisa failu sarežģītība

Apakšelementu skaits	Datu elementu skaits		
	1 - 19	20 - 50	51 +
1	Vienkārši	Vienkārši	Vidēji
2 - 5	Vienkārši	Vidēji	Sarežģīti
6 +	Vidēji	Sarežģīti	Sarežģīti

Jēdziena skaidrojums - ievads

- ◆ *levads ir sistēmas process, kas apstrādā no ārpusēs sistēmā ienākošus datus vai vadības informāciju*
- ◆ *Saņemtie dati modificē sistēmas iekšējos datu failus*

levadu piemēri

- ◆ *Ekrāna ievadformas, kas izmaina iekšējos datu failus, vai no kurām sistēmā nonāk vadības informācija*
- ◆ *Ziņojumi no citām sistēmām, kas īpaši jāapstrādā*
- ◆ *Specifiskie sistēmā apstrādājamie dati, kas sistēmā ienāk no citām sistēmām*
- ◆ *Datu failu konvertācija*

levadu sarežģītība

Datu failu skaits, kurus izmanto ievads	Datu elementu skaits		
	1 – 4	5 - 15	16 +
0 - 1	Vienkārši	Vienkārši	Vidēji
2 - 3	Vienkārši	Vidēji	Sarežģīti
3 +	Vidēji	Sarežģīti	Sarežģīti

levada piemērs

Projekts					
Identifikācija	Pamatojums	Pasūtītājs	Izpildītājs	Universāli	
Identifikators	<input type="text" value="SOFw"/>	Sākums	<input type="text" value="16.01.1997"/>	Beigas	<input type="text" value="22.04.2000"/>
Pilns nosaukums					
<input type="text" value="Funkciju bibliotēka firmai SOFTWARE"/>					
Īsais nosaukums					
<input type="text" value="SOFw"/>					
Realizācijas vide					
<input type="text" value="Visual C++"/>					
Anotācija					
<input type="text" value="Funkciju bibliotēka firmai SOFTWARE kā papildinājums jau esošajām funkcijām. Papildus jāveic esošo funkciju testēšana un problēmu apstrāde."/>					
<div>OKCancel</div>					

Jēdziena skaidrojums - izvads

- ◆ *Izvads ir sistēmas process, kas ģenerē ārpus sistēmas izejošus datus vai vadības informāciju*

Izvadū piemēri

- ◆ *Pārskati, kuru sagatavošanai nepieciešami datu apstrādes algoritmi. Piemēram, dažādi mēneša kopsavilkumi*
- ◆ *Dati, faili vai ziņojumi, kurus nosūta citai programmai. Piemēram, pārskats, kuru attēlo MS Word vai MS Excel*
- ◆ *Trasēšanas u.c. atskaites, kuras veido datu migrācijas laikā*
- ◆ *Informatīvi ziņojumi, kas nav kļūdu paziņojumi vai apstiprinājuma vaicājumi*
- ◆ *Atvasināta vai aprēķināta informācija, ko attēlo uz ekrāna*
- ◆ *Datu grafiskie attēlojumi. Piemēram, dažādas stabiņu vai riņķa diagrammas*

Izvalu sarežģītība

Datu failu skaits, uz kuriem referencējas	Datu elementu skaits		
	1 - 5	6 - 19	20 +
0 - 1	Vienkārši	Vienkārši	Vidēji
2 - 3	Vienkārši	Vidēji	Sarežģīti
4 +	Vidēji	Sarežģīti	Sarežģīti

Izvada piemērs

Microsoft Word - Document2

File Edit View Insert Format Tools Table Window Help

Nosaukums Times New Roman 12 **B** *I* U

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

OBJEKTS: PROJEKTS

Tips: Project

Objektu radīja:		Radišanas datums:	16.01.97 15:24:14
Objektu modifēja:	<u>Environment</u>	Modificēšanas datums:	23.05.00 16:47:40

Pilns nosaukums	Funkciju bibliotēka firmai SOFTWARE
Īsais nosaukums	SOFW
Anotācija	Funkciju bibliotēka firmai SOFTWARE kā papildinājums jau esošajām funkcijām. Papildus jāveic esošo funkciju testēšana un <u>problē</u>
Identifikators	SOFW
Realizācijas vide	<u>Visual C++</u>
Sākuma datums	16.01.97
Beigu datums	22.04.00
Pamatojuma dokumenta nosaukums	
Pamatojums dokumenta reģistrācijas numurs	
Pamatojuma dokumenta sastādīšanas datums	16.01.97
Pamatojuma dokumenta anotācija	

Page 1 Sec 1 1/1 At 6.6cm Ln 6 Col 9 REC TRK EXT OVR WPH

Jēdziena skaidrojums - vaicājums

- ◆ ***Vaicājums ir ievada iniciēts sistēmas process, kura rezultātā atlasa datus no sistēmas iekšējiem datu failiem***
- ◆ ***Vaicājums neizmaina sistēmas iekšējos datu failus***
- ◆ ***Vaicājums ir divdaļīgs:***
 - ◆ ***Ievaddaļa*** - vaicājuma parametri
 - ◆ ***Izvaddaļa*** - vaicājuma rezultāts

Vaicājumu piemēri

- ◆ ***Sistēmas specifiskie dati, kurus pēc pieprasījuma nolasa no viena vai vairākiem iekšējiem datu vai ārējā interfeisa failiem***
- ◆ ***Lietotāja funkcijas, piemēram, View, Lookup, Display, Browse***
- ◆ ***Iebūvētie vaicājumi. Piemēram, datu atlase, lai izvēlētos, kuru ierakstu labot vai izmest***
- ◆ ***Logon informācijas ievads, ja vien ievadītā informācija neizmaina nevienu iekšēju datu failu***
- ◆ ***Katru Help līmeni ieskaita kā vienu vaicājumu. Ja ir sistēmas Help, Help par katru ievadformu un par katru ievadlauku, tad Help sistēma jāvērtē kā 3 vaicājumi***

Vaicājumu sarežģītība

Vaicājuma
ievaddaļas
sarežģītība

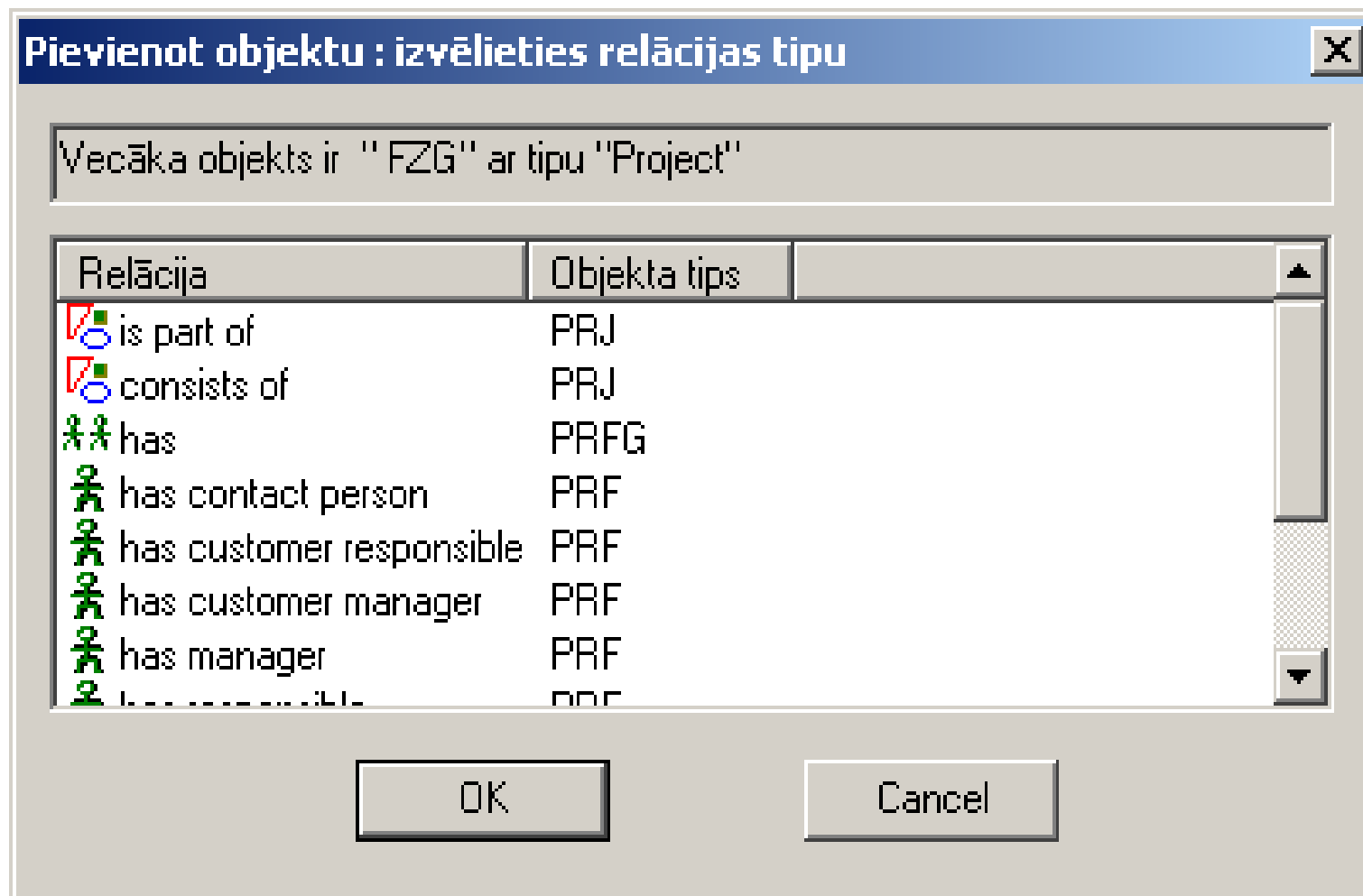
Datu tipu vai failu skaits, uz kuriem referencējas	Datu elementu skaits		
	1 - 4	5 - 15	16 +
0 - 1	Vienkārši	Vienkārši	Vidēji
2	Vienkārši	Vidēji	Sarežģīti
3 +	Vidēji	Sarežģīti	Sarežģīti

Vaicājuma
izvaddaļas
sarežģītība

Datu tipu vai failu skaits, uz kuriem referencējas	Datu elementu skaits		
	1 - 5	6 - 19	20 +
0 - 1	Vienkārši	Vienkārši	Vidēji
2 - 3	Vienkārši	Vidēji	Sarežģīti
4 +	Vidēji	Sarežģīti	Sarežģīti

Vaicājuma sarežģītība ir lielākā no vaicājuma ievaddaļas un izvaddaļas sarežģītības

Vaicājuma piemērs



Algoritmu sarežģītības novērtēšana

- ◆ *Algoritms, kurā izmantotas tikai aritmētiskas darbības, uzskatāms par vienkāršu*
- ◆ *Algoritms, kurā izmantotas sarežģītas formulas, matricu aprēķini u.c. sarežģīti matemātiski vai loģiski aprēķini, uzskatāms par sarežģītu*

Algoritmu piemēri

- ◆ *Kārtošana, ja vērtējamā sistēmā realizēti īpaši kārtošanas algoritmi*
- ◆ *Meklēšana, ja vērtējamā sistēmā realizēti īpaši meklēšanas algoritmi*
- ◆ *Nodokļu aprēķins*
- ◆ *Skaitlisko metožu izmantošana*

Programmrindiņu skaits

Programmrindiņu skaitu (S) iegūst nepieskaņoto funkcijpunktu skaitu reizinot ar vidējo programmrindiņu skaitu viena funkcijpunkta realizācijai

◆ Programmrindiņu skaits viena funkcijpunkta realizācijai dažādās vidēs:

- ◆ Assembly - 320
- ◆ Visual Basic 4.0, 5.0 - 25
- ◆ C - 128
- ◆ C++ - 29
- ◆ Visual C++ - 27
- ◆ Spreadsheet - 6
- ◆ ORACLE - 25

Programmrindīņu skaita pieskaņošana

- ◆ *Cik % programmprodukta koda būs neizmantojami neprecīzas specifikācijas dēļ (BRAK)*
- ◆ *Lieliem projektiem nav ieteicams BRAK izvēlēties mazāku par 10%*
- ◆ *S - prognozējamais programmrindīņu skaits*

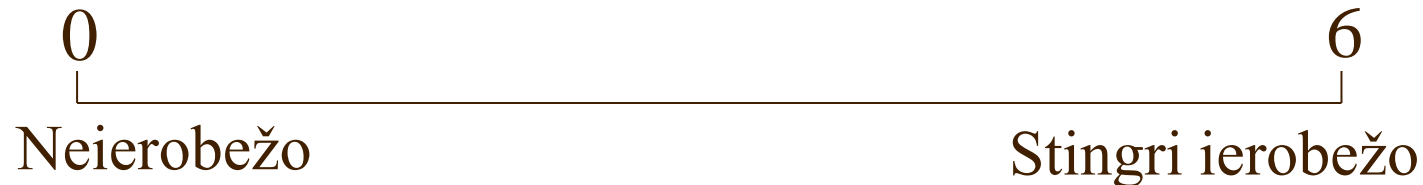
$$S = S \bullet \left(1 + \frac{\text{BRAK}}{100}\right)$$

Izstrādes risku ietekme uz darbietilpību

- ◆ *Iepriekšējā pieredze šādu uzdevumu risināšanā (SF_1)*



- ◆ *Darbu ietekmē spēcīgi aparātūras, programmatūras, laika u.c. ierobežojumi (SF_2)*

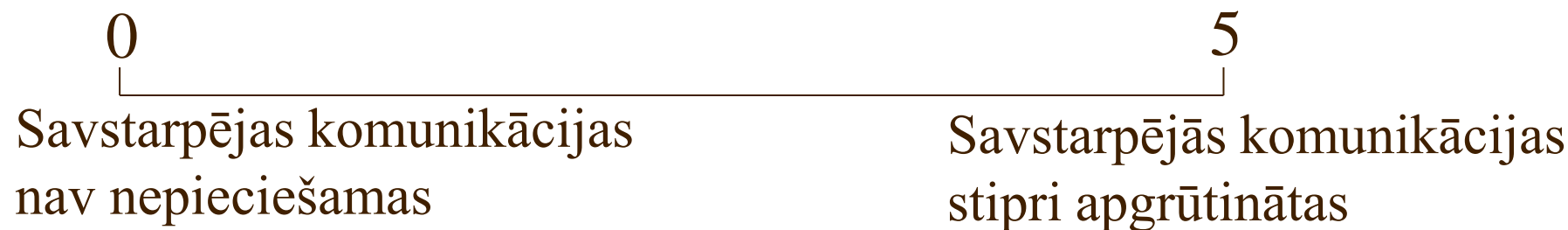


- ◆ *Visas iespējamās atkāpes no standartrisinājumiem ir definētas un iekļautas izstrādes plānos (SF_3)*

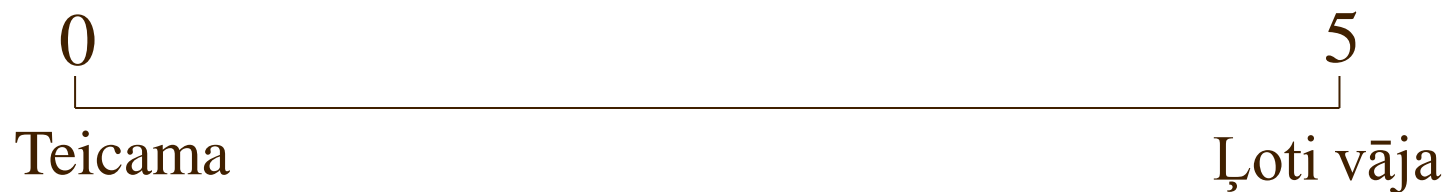


Izstrādes risku ietekme uz darbietilpību (turpinājums)

◆ *Izstrādātāju komanda (SF_4)*



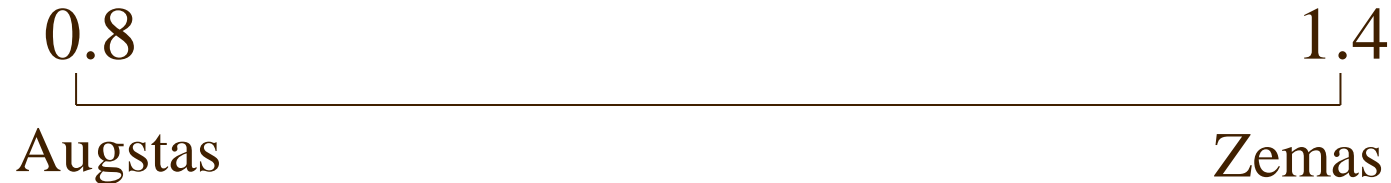
◆ *Produkta izstrādes procesa organizācija (SF_5)*



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

Izstrādes vides ietekme uz darbietilpību

◆ Izstrādātāju spējas (EM_1)



◆ Drošuma prasības (EM_2)

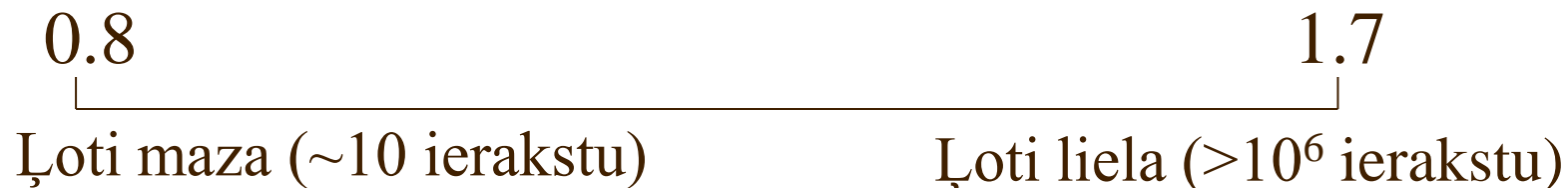
◆ Uzsvars uz drošumu un dokumentētību



◆ Izstrādājamā produkta sarežģītība



◆ Datu bāzes izmērs



Izstrādes vides ietekme uz darbietilpību (turpinājums)

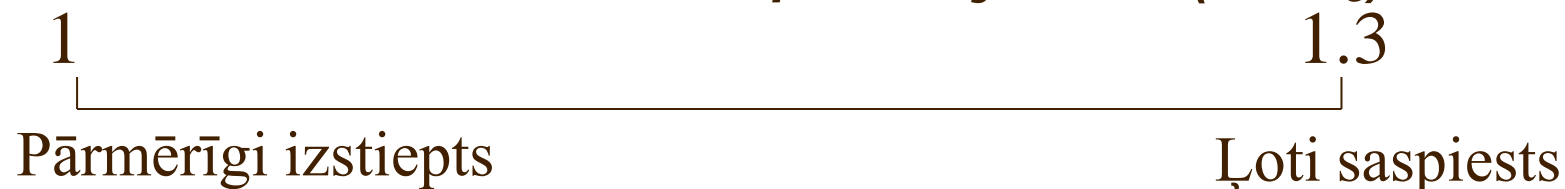
◆ *Atkārtotās pielietojamības prasības (EM_3)*



◆ *Produkta apjoma ierobežojumi un ātrdarbība (EM_4)*



◆ *Izstrādes kalendārais plānojums (EM_5)*



Izstrādes vides ietekme uz darbietilpību (turpinājums)

◆ *Programmārkāmu atbalsts (EM_6)*

◆ **Izstrādes vides stabilitāte**

0.9

1.4

Ļoti stabila

Ļoti nestabila

◆ **Integrēta vide programmaprodukta izstrādei**

0.8

1.2

Stipri integrēta

Nav

◆ **Vides atbalsts programmatūras izstrādei komandā**

0.6

1.2

Ļoti labs atbalsts

Vājš atbalsts darbam
lielā komandā

$$E = \prod_{i=1}^6 EM_i$$

Personmēnešu skaits projekta realizācijai

$$PM = 2.5 \bullet (S/1000)^B \bullet E$$

$$S = S \bullet \left(1 + \frac{BRAK}{100}\right)$$

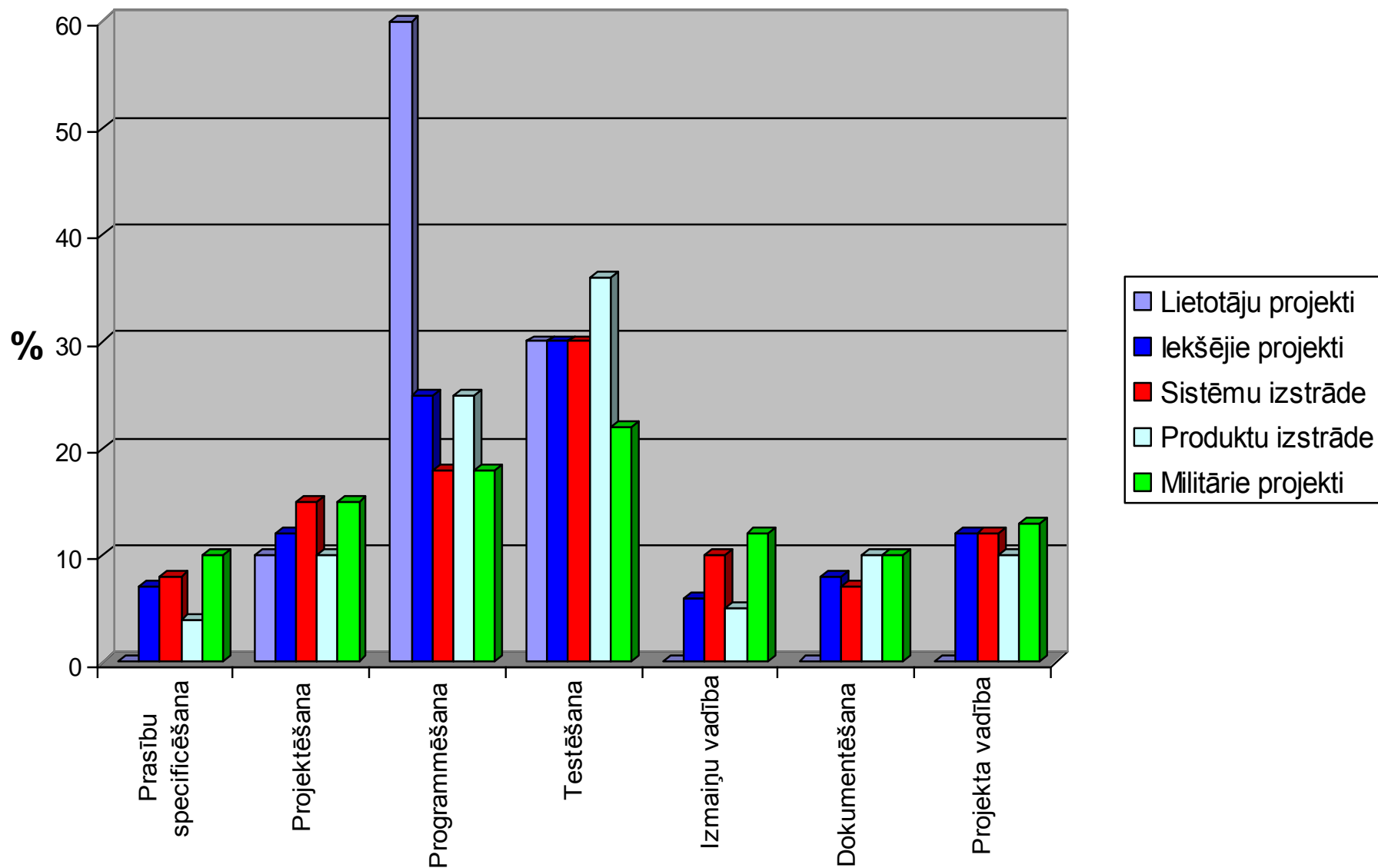
$$B = 1.01 + 0.01 \bullet \sum_{j=1}^5 SF_j$$

$$E = \prod_{i=1}^6 EM_i$$

Kalendāro mēnešu skaits projekta realizācijai

$$CM = 3 \bullet PM^{(0.33 + 0.2 \bullet (B - 1.01))} \bullet \frac{EM_6}{100}$$

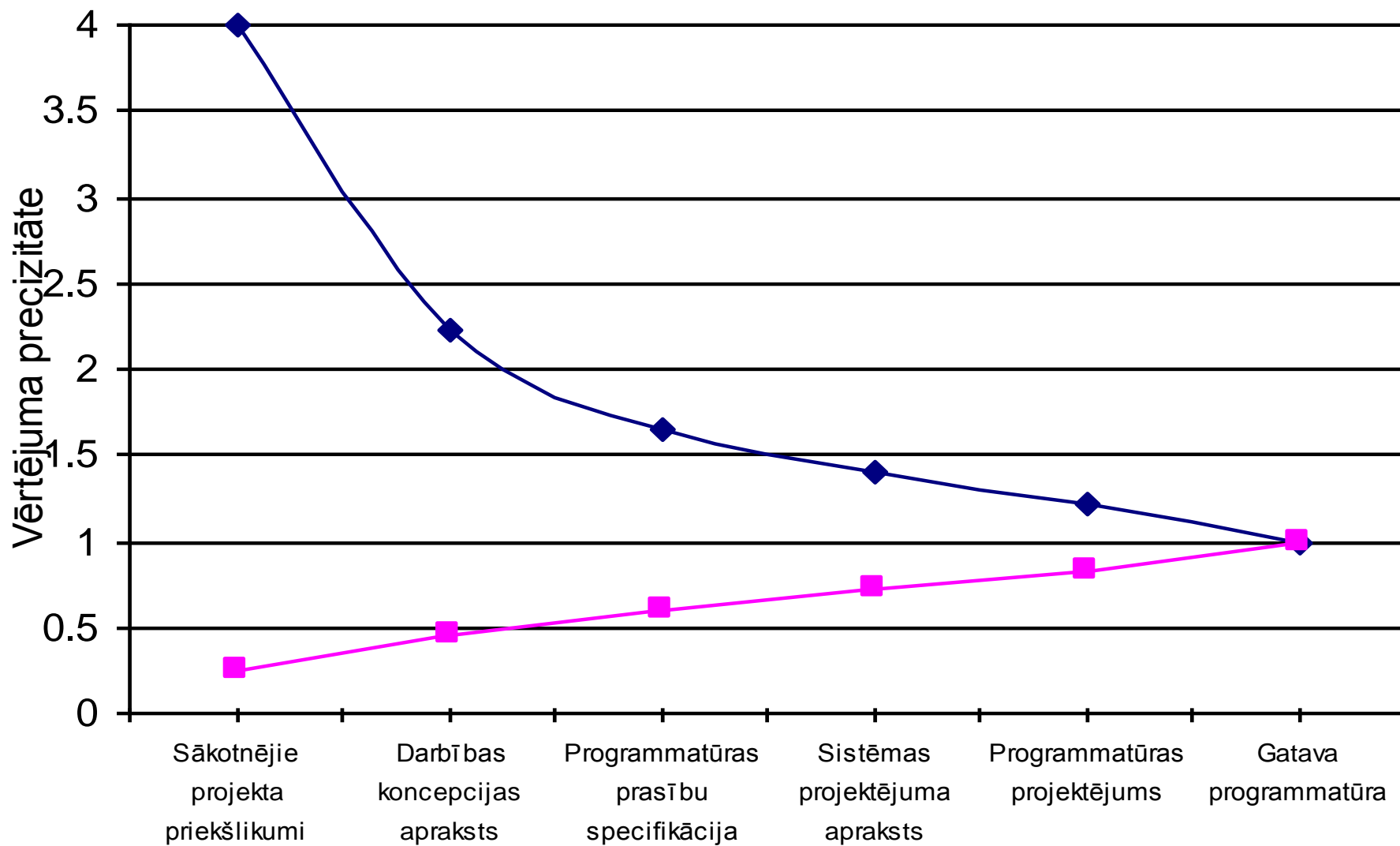
Darbietilpības sadalījums pa aktivitātēm



Darbietelpības sadalījums pa aktivitātēm

Activities Performed	Web	MIS	Outsource	Commercial	System	Military
01 Requirements	5.00%	7.50%	9.00%	4.00%	4.00%	7.00%
02 Prototyping	10.00%	2.00%	2.50%	1.00%	2.00%	2.00%
03 Architecture		0.50%	1.00%	2.00%	1.50%	1.00%
04 Project plans		1.00%	1.50%	1.00%	2.00%	1.00%
05 Initial design		8.00%	7.00%	6.00%	7.00%	6.00%
06 Detail design		7.00%	8.00%	5.00%	6.00%	7.00%
07 Design reviews			0.50%	1.50%	2.50%	1.00%
08 Coding	30.00%	20.00%	16.00%	23.00%	20.00%	16.00%
09 Reuse acquisition	5.00%		2.00%	2.00%	2.00%	2.00%
10 Package purchase		1.00%	1.00%		1.00%	1.00%
11 Code inspections				1.50%	1.50%	1.00%
12 Independent verification and validation						1.00%
13 Configuration management		3.00%	3.00%	1.00%	1.00%	1.50%
14 Formal integration		2.00%	2.00%	1.50%	2.00%	1.50%
15 User documentation	10.00%	7.00%	9.00%	12.00%	10.00%	10.00%
16 Unit testing	30.00%	4.00%	3.50%	2.50%	5.00%	3.00%
17 Function testing		6.00%	5.00%	6.00%	5.00%	5.00%
18 Integration testing		5.00%	5.00%	4.00%	5.00%	5.00%
19 System testing		7.00%	5.00%	7.00%	5.00%	6.00%
20 Field testing				6.00%	1.50%	3.00%
21 Acceptance testing		5.00%	3.00%		1.00%	3.00%
22 Independent testing						1.00%
23 Quality assurance			1.00%	2.00%	2.00%	1.00%
24 Installation/training		2.00%	3.00%		1.00%	1.00%
25 Project management	10.00%	12.00%	12.00%	11.00%	12.00%	13.00%
Total	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Activities	7	18	21	20	23	25

Vērtējumu precizitāte



Literatūras avoti

- ◆ http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html
- ◆ **COCOMO II Model Definition Manual, University of Southern California, 86 p.**
- ◆ **T. Capers Jones. Estimating Software Costs, McGraw-Hill, USA, 1998**
- ◆ **A.J.Albrecht. Measuring Application Development Productivity._Proc. IBM Applic. Dev. Symposium, Monterey, California, 1979**
- ◆ **R.Pressman. Software Engineering: A Practitioner's Approach._McGraw-Hill, 1992, pp 41-91**