

levads studiju nozarē

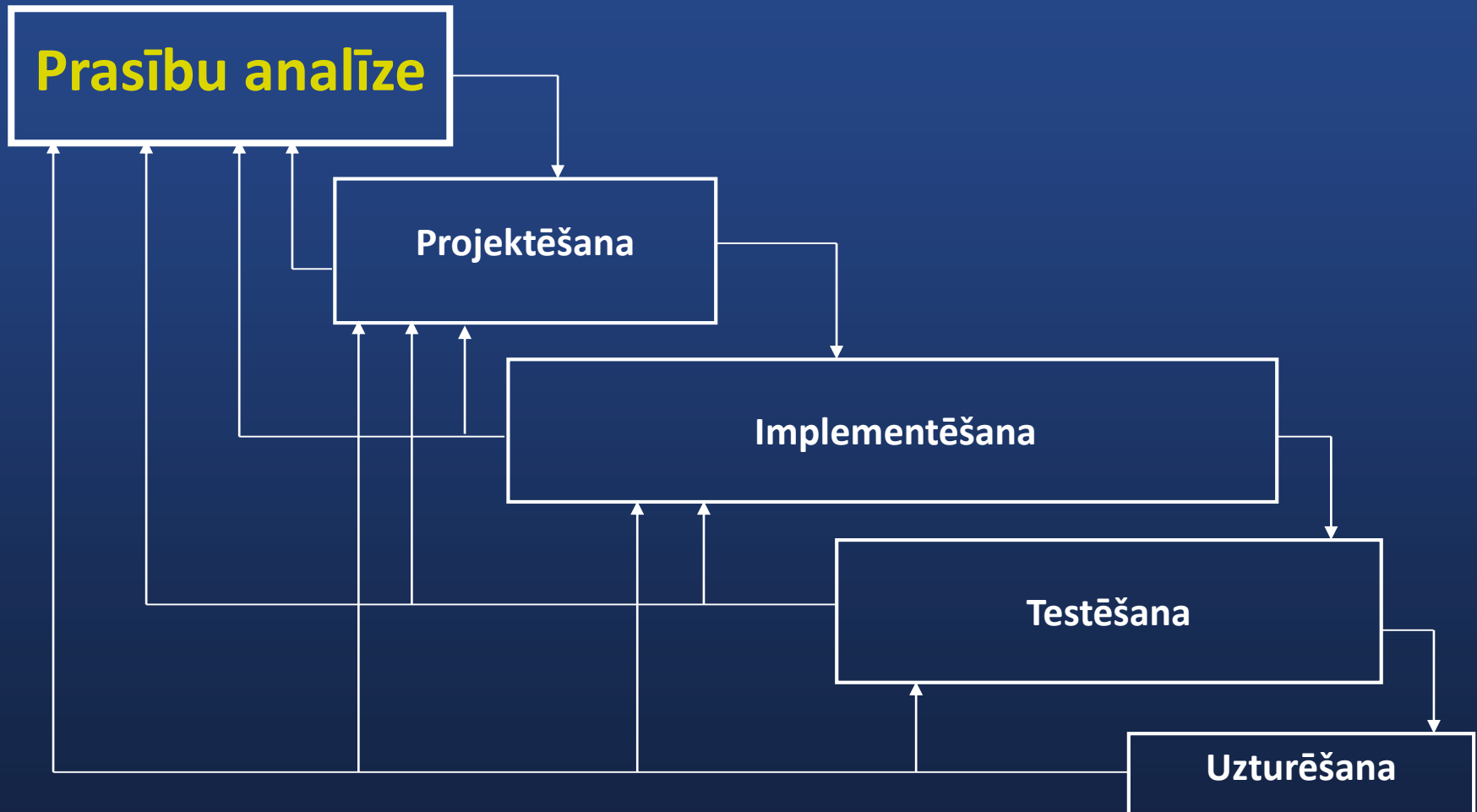
5. lekcija

Tēmu saraksts

- 1) Datorzinātnes un informācijas tehnoloģijas fakultātes misija un struktūra, pedagoģiskās un zinātniskās aktivitātes
- 2) Studiju saturs, studiju darbs un ārpus studiju aktivitātes
- 3) Studiju procesa organizācija, studentu tiesības un pienākumi
- 4) Sistēmu izstrādes projekta uzsākšana
- 5) Prasību analīzes posms sistēmu izstrādes procesā**
- 6) Projektēšanas un izstrādāšanas posmi sistēmu izstrādes procesā
- 7) Testēšanas un uzturēšanas posmi sistēmu izstrādes procesā
- 8) Sistēmu izstrādes projekta noslēgšana

PRASĪBU ANALĪZE

Ūdenskrituma modelis



Prasību analīze

Tā ir sākuma posms programmatūras izstrādes procesā, kura galvenais **mērķis** ir noskaidrot pasūtītāja prasības pret programmatūras produktu.

Prasības pret sistēmu: sistēmas funkciju, ierobežojumu un raksturojumu apraksti.

Tās nosaka, kādai ir jābūt sistēmai un kas tai ir jā dara.

Programmatūras specifikācijas līmeņi

- **Prasību definīcija** (*A requirement definition*)
Teksts dabīgā valodā apraksta funkcijas, kuras sistēma nodrošinās.
- **Prasību specifikācija** (*A requirement specification*)
Strukturēts dokuments formālajā valodā detalizēti apraksta sistēmas funkcijas.
- **Programmatūras specifikācija** (*A software specification*)
Programmatūras abstrakts apraksts sistēmas projektēšanai un implementēšanai.

Prasību analīzes soļi

1. Klientu identificēšana.

Pirms uzsākt prasību iegūšanu, ir jānosaka tie cilvēki no pasūtītāja puses, no kuriem prasības tiks iegūtas. Parasti tie ir cilvēki, kurus tieši vai netieši ietekmēs izstrādājamā programmatūras sistēma, piemēram, operatori, administratori, u.c.

2. Prasību iegūšana.

Šajā solī notiek cieša sadarbība ar iepriekš identificētajiem pasūtītāja pārstāvjiem. Šīs sadarbības uzdevums ir iegūt pēc iespējas pilnīgāku informāciju par sistēmu, it īpaši tās funkcijām, raksturojumiem, saskarni ar citām sistēmām, darbības ierobežojumiem un darbības vidi. Tradicionālas metodes, kas tiek izmantotas prasību iegūšanai, ir klientu intervēšanu, semināru organizēšanu, aptaujas, līdzīgu programmatūras produktu izpēti, esošās dokumentācijas analīzi, u.c.

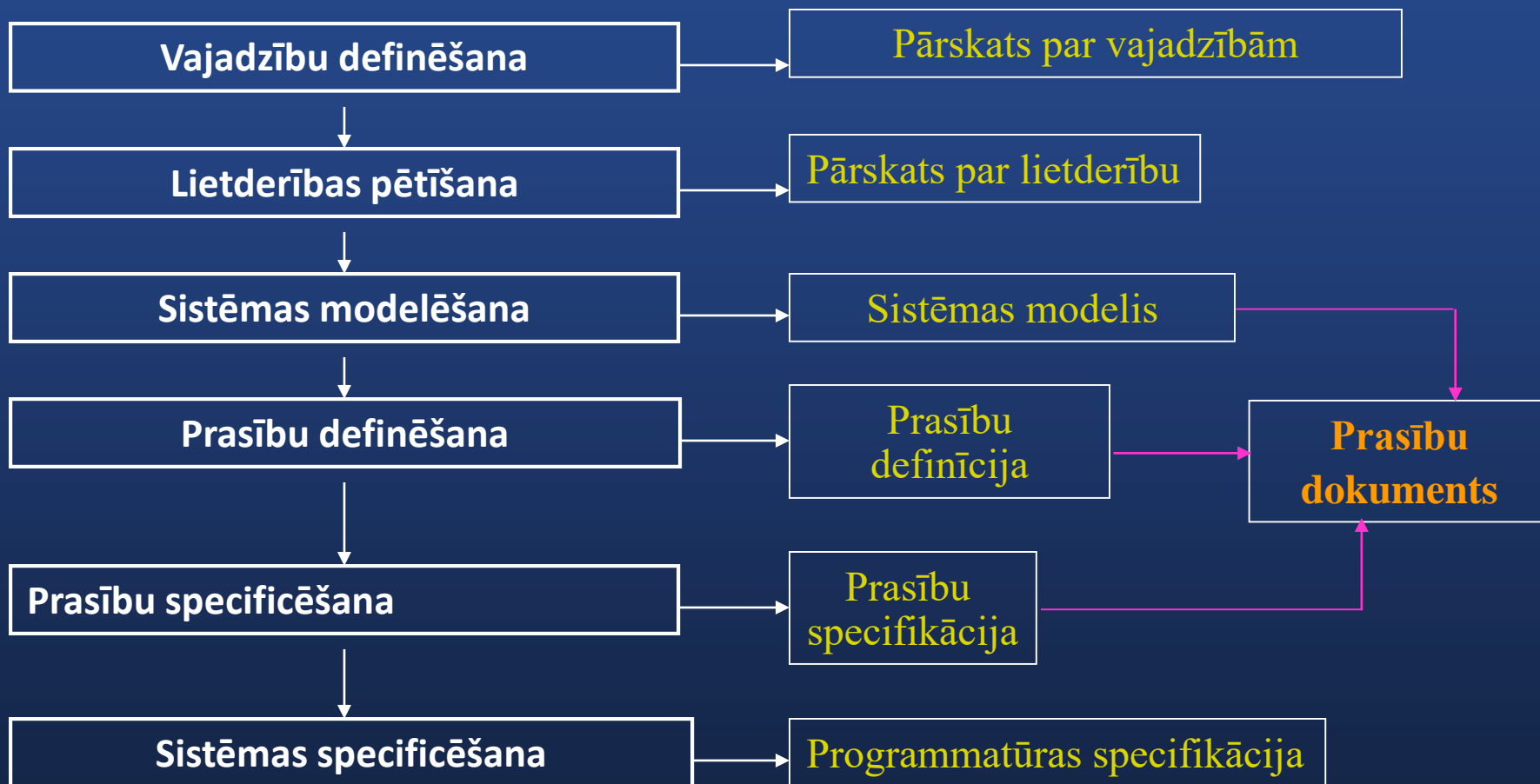
3. Prasību analīze.

Analīzes laikā prasības tiek dalītas radniecīgās grupās un tiek noteiktas to prioritātes, kā arī tiek identificētas prasības, kas konfliktē savā starpā, un paņēmieni šo konfliktu atrisināšanai.

4. Prasību dokumentēšana.

Pēc iegūto prasību analīzes prasības tiek dokumentētas, sagatavojot to formālu aprakstu- **prasību specifikāciju**.

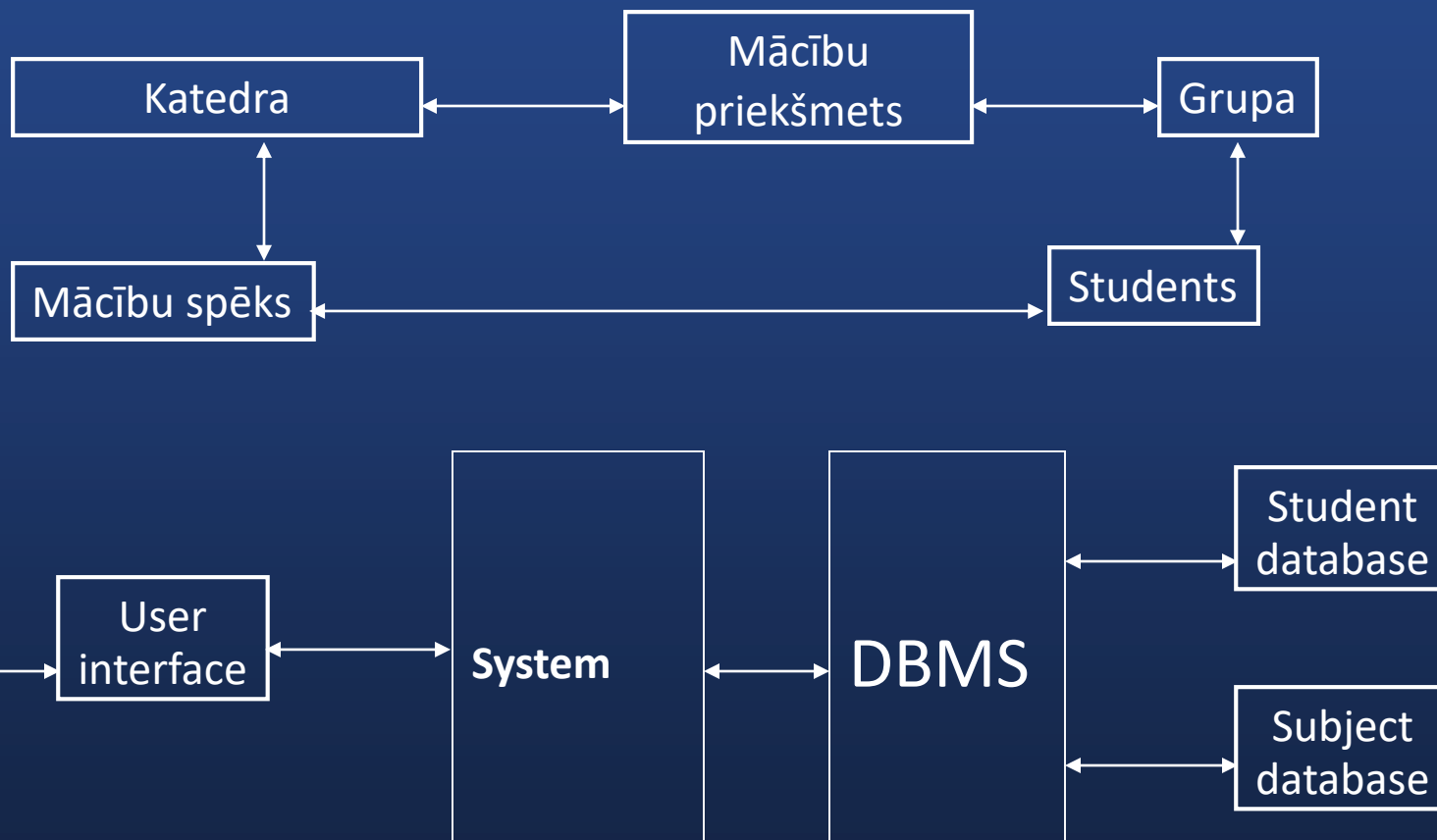
Prasību formulēšanas fāzes



Sistēmas modelēšana

- Lai sagatavotu Prasību dokumentu ir jāizstrādā **sistēmas modelis**.
- Sistēmas modelēšanas soļi:
 - Priekšmetiskās jomas informācijas modelis,
 - Sistēmas saites ar citām sistēmām un lietotājiem.

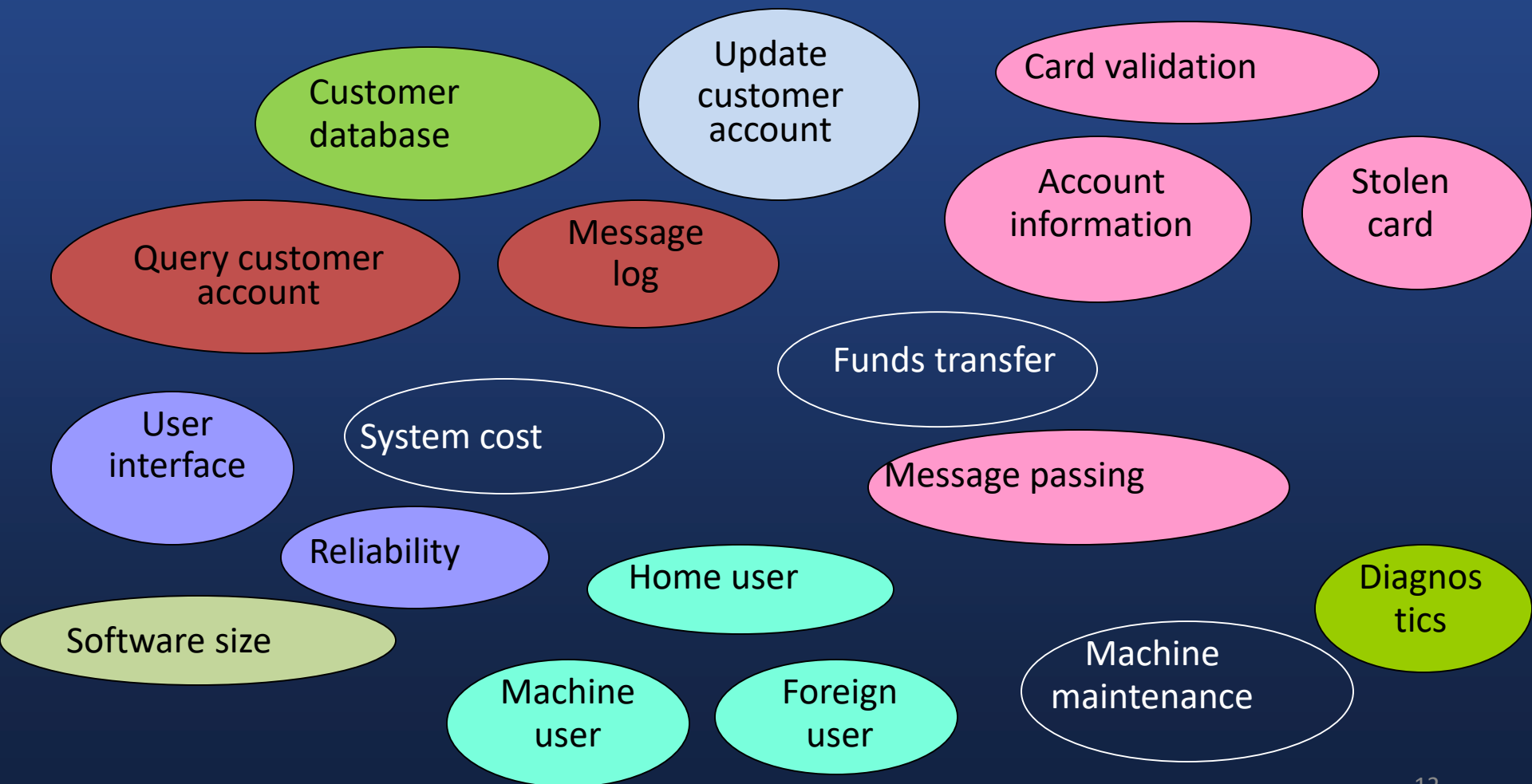
Vienkārša modeļa piemērs



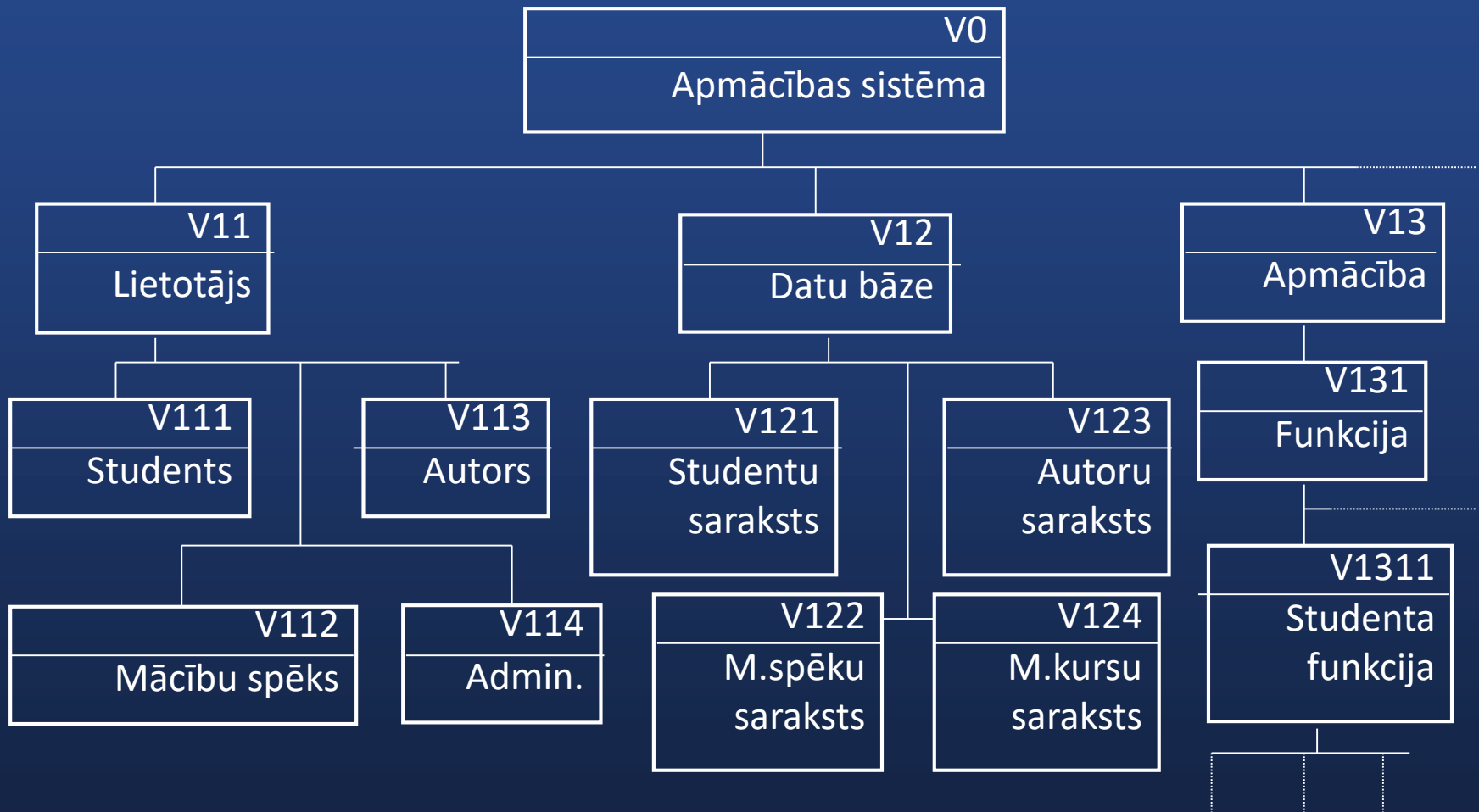
Sistēmas modeļa izstrāde

- Viedokļu savākšana un definēšana
Attēlo burbuļu diagrammas veidā
- Viedokļu analīze un klasifikācija
Datu viedokļi, nefunkcionālie viedokļi, funkcionālie viedokļi, servisa viedokļi, viedokļi par lietotāju
- Viedokļu struktūras izstrāde
Attēlo hierarhiskās diagrammas veidā

Viedokļu burbuļu diagramma



Viedokļu hierarhiskās struktūras piemērs



Tabulārās diagrammas piemērs

(Customer viewpoint)

Source	Input	Action	Output	Destination
ATM	Card input request	Input card	Card	ATM
ATM	PIN input request	Type PIN	Customer PIN	ATM
ATM	Collect cash message	Collect cash	Cash Collection confirm	Customer ATM

Datu modelēšana

Izmanto E-R (entity-relation) modeli.

Attēlo ar diagrammas palīdzību.

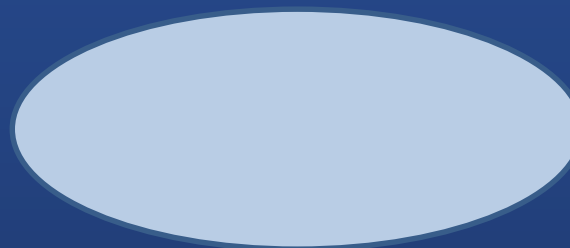
E-R modelis ietver:

- Datu entītiņu kopums,
- Relāciju kopums starp datu entītijām.

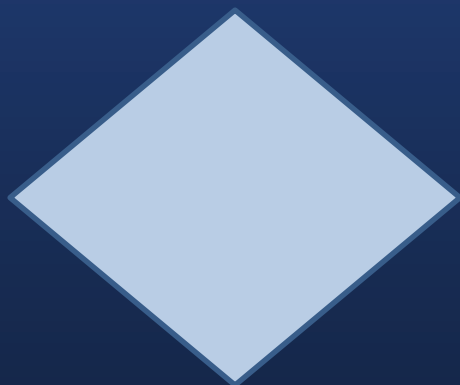
E-R modeļa notācija



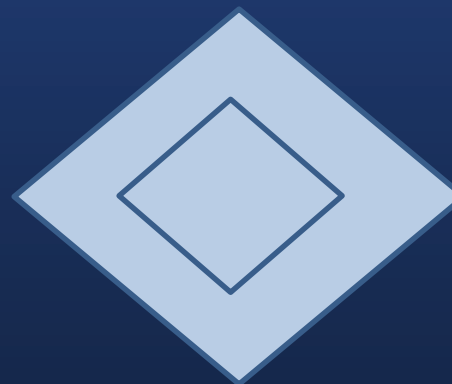
Entītija



Entītijas vai relācijas atribūts

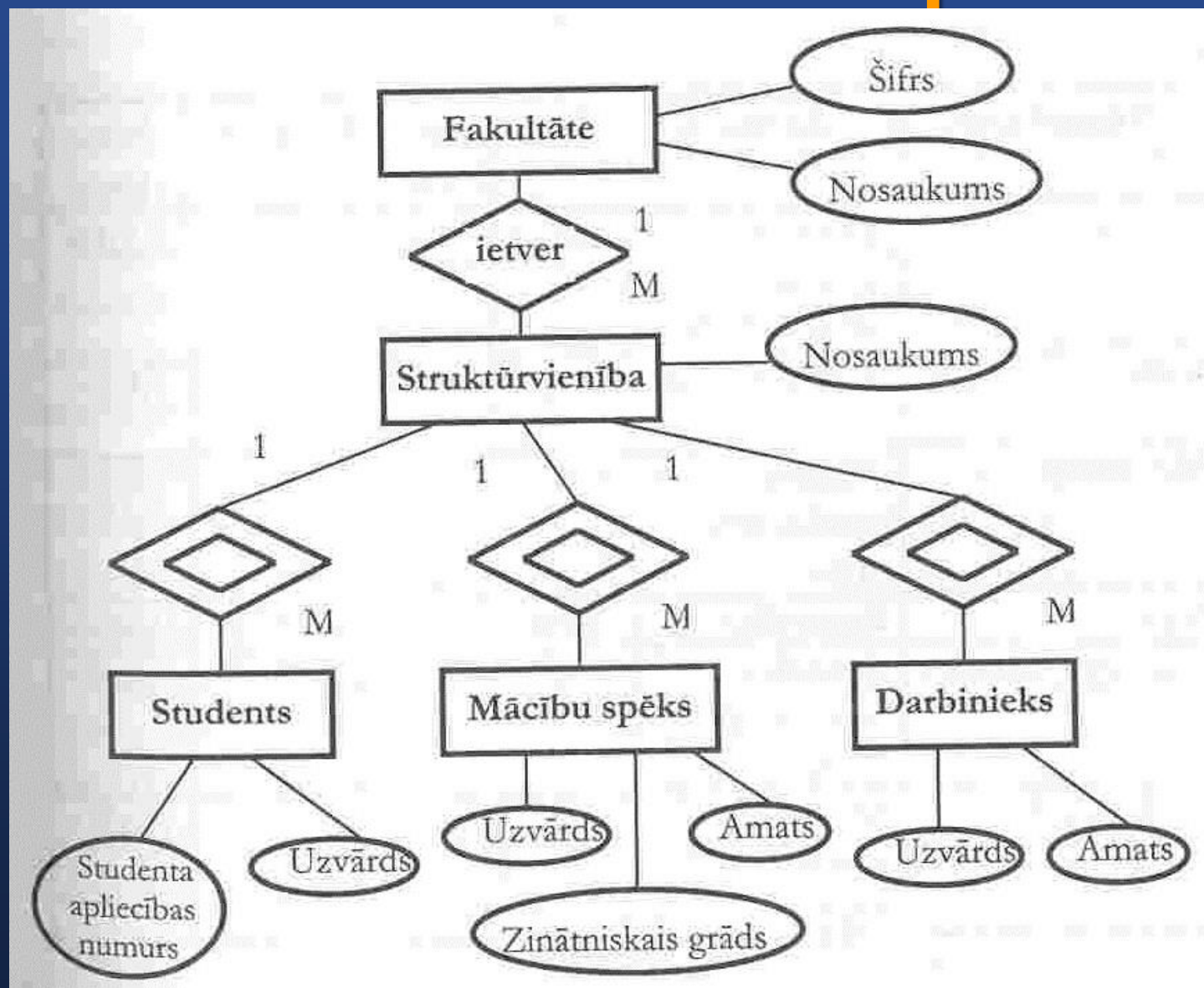


Relācija



Mantošana

Vienkāršots fakultātes piemērs



Datu vārdnīca

Datu vārdnīca ir sistēmas datu komponentu aprakstu kopums.

Var būt izveidota kā ierakstu vai datu kopums.

Parasti attēlo tabulārā veidā.

Datu vārdnīcas piemērs

Nosaukums	Apraksts	Tips	Garums	Datums	Statuss
Atzime	Studenta eksāmena atzīme	Integer	2	01.02.02	Aktīvs
Kurss	Kursa numurs	Integer	1	01.02.02	Aktīvs
MP_šifrs	Mācību priekšmeta šifrs	String	6	15.02.02	Aktīvs
Stip	Stipendija	Real	5.2	20.02.02	Aktīvs
Stud_apl	Studenta apliecības numurs	String	9	01.02.02	Aktīvs
Stud_uzvārds	Studenta uzvārds	String	30	01.02.02	Aktīvs

Datu vārdnīcas priekšrocības

- Datu tipu un garumu pārbaudes iespēja izstrādes sākumprocesā,
- Datu sarakstu vārdnīcā var izmantot jebkurš izstrādātājs,
- Unikālo datu nosaukumu nodrošināšana.

Prasību specifikācija

Programmatūras specifikācija ir sistēmas programmatūras precīzs apraksts.

Parasti šī apraksta sastādīšana notiek pēc projektējamās sistēmas aparatūras izvēles un sistēmas projektējuma specifikācijas izstrādāšanas.

Tās izveidošanas gaitā galvena uzmanība tiek pievērsta tam, **kas programmatūrai jādara, nevis tam, kā šīs darbības jāveic.**

Prasību specifikācijas sastāvdaļas

Prasību specifikācija galvenokārt satur:

- **funkcionālās** prasības:

apraksta funkcijas (aprēķinus, datu apstrādi, darbības, u.c.), kas sistēmai ir jāpilda;

- **nefunkcionālās** prasības:

apraksta dažādus ierobežojumus, kas var tikt attiecināti gan uz izstrādājamo sistēmu, gan uz programmatūras izstrādes procesu.

Funkcionālo prasību piemēri

1. Sistēmai ir jāveic jauna bibliotēkas lasītāja reģistrācija, kas paredz pamatinformācijas par lasītāju ievadi un lasītāja elektroniskās kartes izveidi.
2. Sistēmai ir jāaprēķina naudas sods, ja grāmatas nodošanas laiks pārsniedz 14 dienas no grāmatas izsniegšanas datuma.
3. Sistēmai ir jānodrošina grāmatas meklēšana, pamatojoties uz tās nosaukumu un/vai autoru.

Nefunkcionālo prasību kategorijas

- Prasības pret produktu:

sistēmas raksturojumi, kā drošums, ātrdarbība, pārnesamība starp dažādām platformām, u.c. Piemēram, sistēmai ir jāizvada meklēšanas rezultāti 2 sekunžu laikā pēc meklēšanas pieprasījuma saņemšanas.

- Prasības pret izstrādes procesu:

saistītas ar izmantojamiem standartiem programmatūras izstrādei, programmēšanas vides izvēli, dokumentāciju, kas ir jānodod kopā ar produktu, u.c. Piemēram, sistēmas izstrādes dokumentācija ir jānoformē atbilstoši Latvijas valsts standartam.

- Ārējās prasības:

kas neietilpst iepriekš minētajās kategorijās un ietver sevī prasības sistēmas saskarnei ar citām sistēmām, likumdošanas prasības, prasības pret datu aizsardzību un privātumu. Piemēram, sistēma nedrīkst izvadīt nekādu citu informāciju par personu, izņemot vārdu un uzvārdu.

Nefunkcionālo prasību piemēri

1. Prasība procesam

1.3.2. Sistēmas prasību specifikācija tiks noformēta saskaņā ar Latvijas Valsts standartu LVS 68:1996

2. Prasības produktam

2.2.1. Sistēmas reakcijas laiks uz lietotāja komandu nedrīkst pārsniegt 2 sekundes

2.4.3. Lietotāju saskarnei ar sistēmu jānotiek latviešu valodā

3. Ārējā prasība

3.3.5. Sistēma tiks integrēta ar RTU eksistējošām datu bāzēm MP reģistrs un Studentu reģistrs un nekādas izmaiņas datu bāzu struktūrā netiks ienestas

Prasību mērvienības

Raksturlielums	Mērvienība
Ātrdarbība	Transakciju skaits/ sekunde Sistēmas atbildes laiks Ekrāna atsvaidzināšanas laiks
Izmērs	Kilobaits (KB)
Izmantošanas vienkāršība	Apmācības laiks Help-kadru skaits
Drošums	Vidējais laiks starp kļūmēm Neizmantojamības varbūtība Kļūmju gadījumu norma Izmantojamība
Robustums	Pēckļūmes atkopšanas laiks Notikumu procents, kas izsauc kļūmes Datu zūduma varbūtība kļūmes gadījumā
Pārnesamība	Mērķatkarīgo operatoru procents

Prasības prasību dokumentam

Dokumentam ir:

- Jāspecificē sistēmas ārejā uzvedība,
- Jāspecificē realizācijas ierobežojumi,
- Jāļauj ērti koriģēt,
- Jāietvēr norādījumi apkalpojošam personālam,
- Jāparedz ziņas par sistēmas dzīves ciklu,
- Jāparedz adekvātas reakcijas uz nevēlamiem notikumiem.

Prasību dokumenta struktūra

Latvijas valsts standarts prasību specifikācijas rakstīšanai iesaka šādu šī dokumenta struktūru:

1. **Ievads.** Šī nodaļa apraksta prasību specifikācijas nolūku un cilvēku grupu, kam tā ir domāta, identificē programmatūras produkta nosaukumu, īsi paskaidro, kas programmatūras produktam ir jādara, kā arī apraksta programmatūras produkta pielietojumu.
2. **Vispārējais apraksts.** Šeit tiek aprakstīta sistēmas sadarbība ar citām sistēmām (ja tāda ir), tiek dots kopsavilkums par sistēmas funkcijām, tiek uzskaitītas raksturiezīmes, kas raksturo programmatūras sistēmas lietotāju.
3. **Specifiskās prasības.** Šajā nodaļā apraksta sistēmas funkcionālās un nefunkcionālās prasības.
 - **Funkcionālās prasības.** Katru funkcionālo prasību dokumentē, aprakstot tās mērķi, ievaddatus un izvaddatus, kā arī visas operācijas, kuras funkcija izpilda ar ievaddatiem, lai iegūtu izvaddatus.
 - **Nefunkcionālās prasības.**

Prāsību specifikācijas speciālās valodas

- **PSL/PSA** (*Problem Statement Language / Problem Statement Analyzer*)
ietver uzdevumu specifikācijas valodu PSL un uzdevumu specifikācijas analizatoru PSA
- **SADT** (*Structured Analysis and Design Technique*)
struktūrētās analīzes un projektēšanas grafiskā sistēma prasību analīzei, programmatūras projektēšanai un sistēmas definēšanai
- **SREM** (*Software Requirements Engineering Methodology*)
sistēma ar prasību specifikācijas valodu RSL saišu noteikšanai starp objektiem

Prasību specifikācijas apraksta alternatīvas

- Lēmumu tabulas,
- Petri tīkli,
- PDL (Program description language) valoda.

Funkcijas specifikācijas piemērs strukturētā valodā

Funkcija	Kredītkartes_pārbaude
<u>Apraksts</u>	Funkcijas mērķis ir pārbaudīt kredītkartes korektumu
<u>Ieeja</u>	Bankas identifikators, konta numurs, pēdējās operācijas datums
<u>Avots</u>	Datus nolasa no kredītkartes magnētiskās joslas
<u>Izeja</u>	Kartes_statuss = (OK, Nepareizi)
<u>Norādījums</u>	Rezultāts tiks pārvietots uz citām programmas daļām
<u>Prasības</u>	Konta formāts, datums
<u>Pirms stāvoklis</u>	Kartes ievades laikā dati no joslas tika nolasīti
<u>Pēc stāvoklis</u>	Bankas identifikators ir sarakstā <u>and</u> Konta numurs atbilst formātam <u>and</u> Pēdējās operācijas datums < = Šodienas datums Kartes_statuss = OK <u>or</u> (ja kaut kas neatbilst) Kartes_statuss = Nepareizi__

Funkcijas specifikācijas piemērs

PDL valodā

```
PROCEDURE Kartes_parbaude (  
    Bankalidentifikators : BANKA;  
    Banku_Saraksts      : BANKU_KOPUMS;  
    KontaNumurs         : KONTA_NUMURS;  
    PedOperacDatums     : DATUMS;  
    KartesStatuss       : IN_OUT_STATUSS;  
/* Funkcijas mērķis ir pārbaudīt kredītkartes pareizību */  
BEGIN if PedOperacDatums<=ŠodienasDatums and  
    KorektsFormats(KontaNumurs) and  
    IrSaraksta (Banku_Saraksts, Bankalidentifikators)  
then    KartesStatuss = OK;  
else    KartesStatuss = Nepareizi;  
END Kartes_parbaude;
```


Sistēmanalītīķis

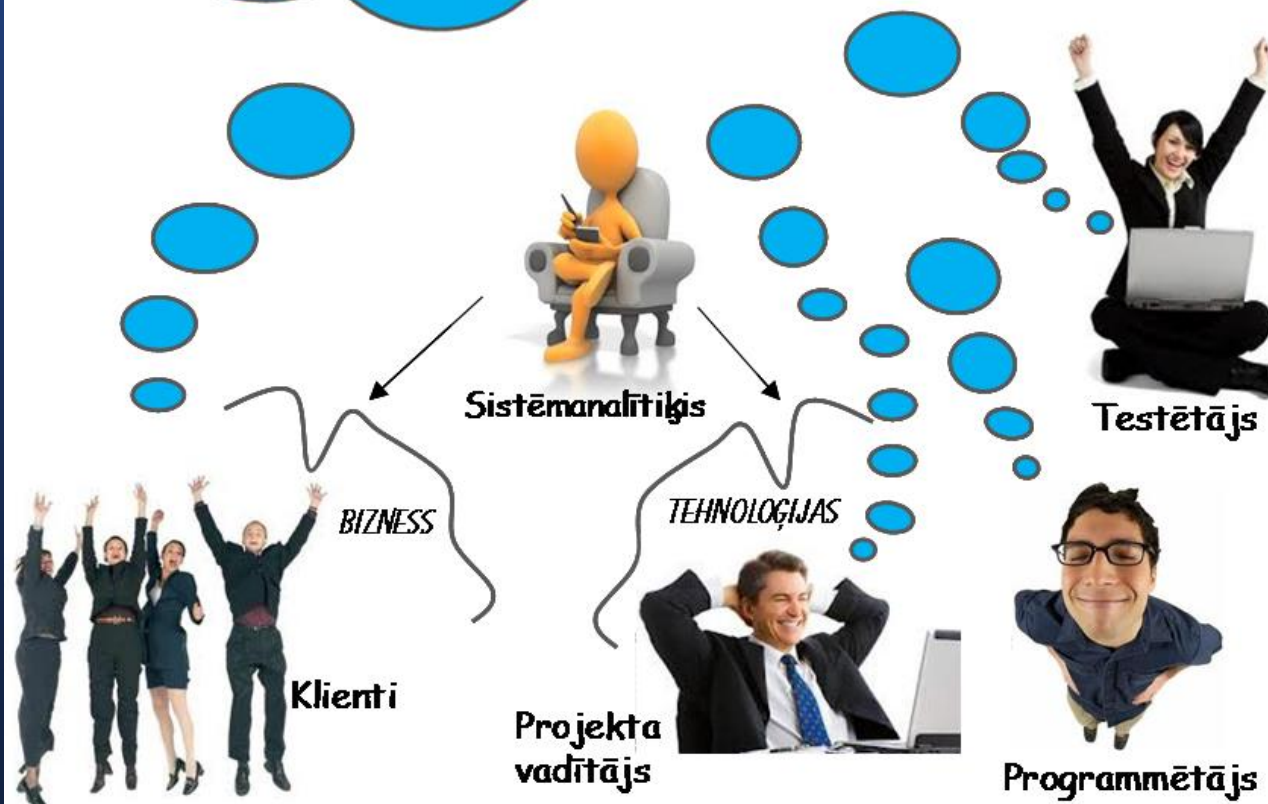
DZĪVE BEZ SISTĒMANALĪTIKA

- Nav vienotas izpratnes,
t.i., zināšanas nav vienādotas
- Visi runā dažādās valodās
- Pastāv risks, ka pasūtītājs nebūs
apmierināts ar rezultātu
- Pastāv konfliktējošas prasības



DZĪVE AR SISTĒMANALĪTIKI

Gan Klients, gan Izstrādātāji
domā par vienu, nevis
katrs par savu



Sistēmanalītiķis un citi darbinieki



Karjeras iespējas

Tabula 1.

Profesijas novērtējums darba tirgū	
Labākais darbs tehnoloģiju jomā	#2
100 labākie darbi	#2

Tabula 2.

Cik var nopelnīt sistēmanalītiķis?	2013	2012	Pieaugums
Vidējā gada alga sistēmanalītiķim ASV	\$ 78197	\$ 77139	1.4 %

Kas ir sistēmanalītiķis?

Sistēmanalītiķis – speciālists, kas analizē un izvērtē klientu prasības attiecībā uz informācijas tehnoloģijām, procedūrām vai problēmām, veic zinātniskās pētniecības darbus, kā arī attīsta un īsteno priekšlikumus, ieteikumus un izstrādā plānus, lai uzlabotu pašreizējo vai jaunās informācijas sistēmas.

Sistēmanalītiķu veidi

- **Informācijas sistēmanalītiķis**

programmēšanas speciālists, kurš var strādāt dažādu zinātņu un prakses priekšmetu nozarēs uz informācijas sistēmu un tehnoloģiju vispusīgas zināšanas pamata.

- **Sistēmanalītiķis-programmētājs**

projektē un rada praksei nepieciešamos konkurentspējīgos programlīdzekļus un datoru informācijas sistēmas, tās nodrošinot ar nepieciešamu dokumentāciju un ekspluatācijas pavadīšanu.

- **Biznesa sistēmu analītiķis**

pēta biznesa procesus un informācijas plūsmas, izmantojot biznesa modelēšanu (biznesa procesu modelēšanu) un datu modelēšanu.

- **Datu bāzes analītiķis**

izstrādā un uztur datu bāzes, garantē informācijas pieeju lietotājiem, izstrādā datu bāžu informācijas sistēmas. Veic datu bāžu analīzi dažādās nozarēs: ražošana, vadība, informācija.

Sistēmanalītiķa pienākumi

- analizēt uzņēmuma darbību un tajā jau izmantojamos IT risinājumus, kuram tiek izstrādāts programmatūras produkts;
- plānot, sagatavot un vadīt intervijas ar pasūtītājiem, kā arī apkopot interviju rezultātus;
- iegūt un aprakstīt prasības pret programmatūras produktu, izmantojot šim nolūkam speciālas metodikas un rīkus;
- izstrādāt programmatūras sistēmas prasību specifikāciju;
- organizēt un vadīt pakļauto cilvēku grupas;
- sniegt konsultācijas darba kolēģiem: projektētājiem, programmētājiem un testētājiem;
- sekot tam, kā attīstās informācijas tehnoloģijas nozare.

Darbības posms

Darbojas prasību analīzes posmā:

- noskaidro pasūtītāja prasības,
- dokumentē pasūtītāja prasības,
- seko līdzi sistēmas attīstībai cauri visiem programmatūras izstrādes posmiem, lai kontrolētu un nodrošinātu programmatūras sistēmas izstrādi atbilstoši pasūtītāju prasībām.

Sistēmanalītiķim nepieciešamās zināšanas

Ir jābūt:

- teorētiskās un praktiskās zināšanas dažādās IT nozares virzienos,
- prasmei strādāt ar IT nozares standartiem, speciālo tehnisko literatūru un pieņemto terminoloģiju,
- izpratnei par to sfēru, kurai tiek izstrādāts konkrēts programmatūras produkts,
- lieliskai komunikācijas prasmei.

Sistēmanalītiķim nepieciešamās prasmes

Ir jāprot:

- pielietot intervēšanas metodika un prasību iegūšanas metodes,
- domāt abstrakti un izmantot radošu pieeju, lai rastu risinājumu tam, kādam ir jābūt programmatūras produktam,
- izmantot dažādas diagrammas un shēmas, lai attēlotu programmatūras sistēmas struktūru un funkcijas.
- efektīvi un patstāvīgi strādāt komandā un sadarboties ar citiem kolēģiem un pasūtītāju.

Dokumentētājs

Tehniskais dokumentētājs

Tehniskais dokumentētājs ir tehniskās dokumentācijas izstrādes speciālists.

Tehniskie dokumentētāji:

- projektē,
- izstrādā,
- uztur un atjauno programmatūras produkta dažāda veida tehnisko dokumentāciju.

Dokumentētāju var darboties visos programmatūras izstrādes procesa posmos.

Tehniskā dokumentācija

Programmatūras sistēmas tehniskā dokumentācija ir programmatūras sistēmas izstrādes un lietošanas instrukcija.

Programmatūras sistēmas tehniskā dokumentācija ietilpst:

- programmatūras sistēmas prasību specifikāciju,
- Projektējumu,
- lietotāja ceļvežu,
- produkta aprakstus.

Dokumentētāja pienākumi

- apkopot informāciju par programmatūras sistēmu un veikt tās analīzi;
- izstrādāt tehniskajā dokumentācijā ietilpstošo dokumentu struktūru un formātu;
- izstrādāt tehniskajā dokumentācijā ietilpstošos dokumentus un veikt to modifikāciju nepieciešamības gadījumā.

Nepieciešamās zināšanas

Ir jābūt:

- augsta līmeņa rakstīšanas prasmei,
- dziļām zināšanām IT joma, ir īpaši lietotajā terminoloģijā,
- prasmei vizualizēt informāciju, izmantojot dažādas metodes, piemēram, attēlus, grafikus, u.c.
- labām svešvalodu (vismaz angļu valodas) zināšanām – sarunvalodas līmenī (pārrunām ar pasūtītāju), dokumentācijas rakstīšanas un tehniskās literatūras lasīšanas līmenī.

Nepieciešamas prasmes

Ir jāspēj:

- domāt analītiski un izklāstīt informāciju loģiskā un strukturētā veidā,
- efektīvi un patstāvīgi strādāt komandā un sadarboties ar citiem kolēģiem un pasūtītāju.,
- Lieliski komunicēties.

Standarti un vadlīnijas

- SWEEBOK,
- IEEE Standards (piemēram IEEE Recommended Practice for Software Requirements Specifications),
- ANSI / IEEE Std 1063-1987
- Programmatūras izstrādes LV standarti.

Dokumentēšanas rīki

- MS Visio,
- CASEWISE – corporate modeler tool,
- ROBOHelp – tool for developing help systems,
- Adobe Dreamweaver,
- Rīki ar BPMN.

Studiju priekšmeti

- Sistēmu analīze un zināšanu iegūšana (intervēšanas māksla);
- Diskrētas struktūras datorzinātnēs;
- Datu bāzes tehnoloģijas;
- Programmēšanas valodas;
- Objektorientēta sistēmanalīze.

Kam tas patiktu

- Cilvēkiem, kam patīk komunicēt un mācīties jauno;
- Cilvēkiem, kam patīk/kas mīl tehnoloģijas;
- Cilvēkiem, kas neuzskata, ka rakstīt ir garlaicīgi;
- Cilvēkiem, kas prot apkopot, vispārināt un detalizēt.
- ...

PALDIES PAR UZMANĪBU!