

Rīgas Tehniskā Universitāte  
Datorzinātnes un informācijas tehnoloģiju fakultāte

Lietišķo datorsistēmu institūts

Objektorientētas metrikas

# **Referāts priekšmetā**

*Programmatūras metroloģijas un plānošanas modeļi*

Autors: Pāvels Kartaševs

I-DMD-1

Apliecības numurs: 061RDB231

2009. gads

## Saturs

Ievads.....	3
Objektorientētas struktūras.....	4
Metrikas priekš objektorientētam sistēmam.....	6
Objektu-orientētas specifiskas metrikas (Chidamber & Kemerer's Metics Suite ).....	8
Nosvērotas metodes pret klasi (Weighted Methods per Class (WMC)) .....	9
Klases atbilde - Response for a Class (RFC) .....	9
Sakļautības trūkums - Lack of Cohesion (LCOM) .....	10
Savienošana pāros starp objektu klasēm - Coupling Between Object Classes (CBO) .....	10
Mantošanas koka dziļums - Depth of Inheritance Tree (DIT) .....	11
Bērnu skaits - Number of Children (NOC) .....	11
Pielietojums .....	12
Secinājumi.....	17
Literatūtas saraksts.....	18

## **Ievads**

Objekt-orientēta projektēšana un izstrāde ļoti populāras koncepcijas mūsdienās programmatūras izstrādes vidē. Objektorientētā programmatūras izstrādē ir nepieciešama atšķirīga pieeja no tradicionālo funkcionālo dekompozīcijas un datu plūsmas attīstības metodes. Kaut arī funkcionālā un datu plūsmas pieejas, sākt, apsverot sistēmas uzvedības vai datu atsevišķi, objektorientētā analīze pieejas problēmu, meklējot sistēma uzņēmumiem, kas tos apvienot. Objektorientētā analīze un projektēšana vērsta uz objektiem kā uz pirmatnējiem aģentiem, kas iesaistīti šajā aprēķinā, un katrs klases datu un saistīto darbību laikā ir savāc vienotā sistēmā vienība.

Metrikas fokusējas uz iekšējo objektu struktūrām, kas attēlo katras entitijas sarežģītību – metodes, klases, un ārējo sarežģītību, kas mēra entitijas individuālas sadarbības, tādas kā savienojamību un mantojamību. Metrikas mēra skaitļošanas efektivitāti kas ietekmē algoritma efektivitāti un mašīnas resursu izmantošanu, kā arī psiholoģiskas sarežģības faktorus, kas ietekmē programmētāja īpašības rādīt, saprast, modificēt un uzturēt programmatūru.

Metrikas ir izvēlētas lai nosauktu izstrādātājiem un vadītājiem par koda un projektēšanas kvalitāti un objekt-orientētu struktūru.

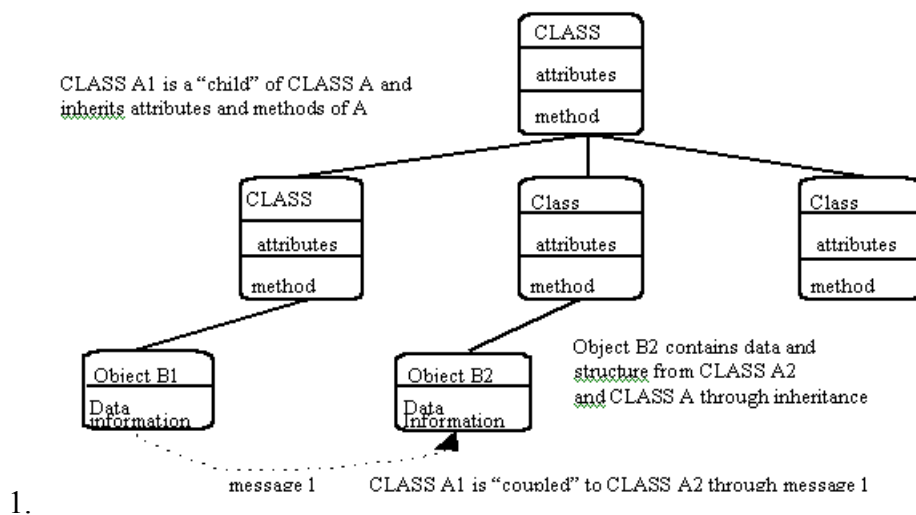
## Objektorientētas struktūras

Objektorientētu struktūru apraksts dots attēlā 1 un definīcijas tabulā 2

<b>Atribūts</b>	Definē strukturālas klašu īpašības, unikāls klasē
<b>Klase</b>	Objektu kopa, kas dala vienu struktūru un uzvedību, ko noteic metodes. Kopa servē kā objekta šablons.
<b>Sakļautība</b>	Leņķis pie kurā metodes klasē ir saistīts viens ar otru
<b>Savienošana pāros</b>	Objekts X ir savienots ar objektu Y tikai ja X nosūtīs ziņojumu objektam Y.
<b>Mantošana</b>	Attiecība starp klasēm, kad objekts klasē iegūst citas vai vairāk klases īpašības,
<b>Instantiation</b>	Process, kurā tiek radīta jauna objekta instance un saistīšana vai specifisko datu pievienošana
<b>Ziņojums</b>	Pieprasījums, kas objekts sūta citam objektam, lai izveidotu darbību
<b>Metode</b>	Operācija ar objektu, definēta, kā klases sastāvdaļa
<b>Objekts</b>	Klases iedarbināšana, kas ļauj saglabāt informāciju un piedāvā darbību skaitu, lai iedarbotos uz informāciju.
<b>Operācija</b>	Darbība, kas tiek veikta ar vai no objektam visiem klases instancēm, var arī nebūt unikāla

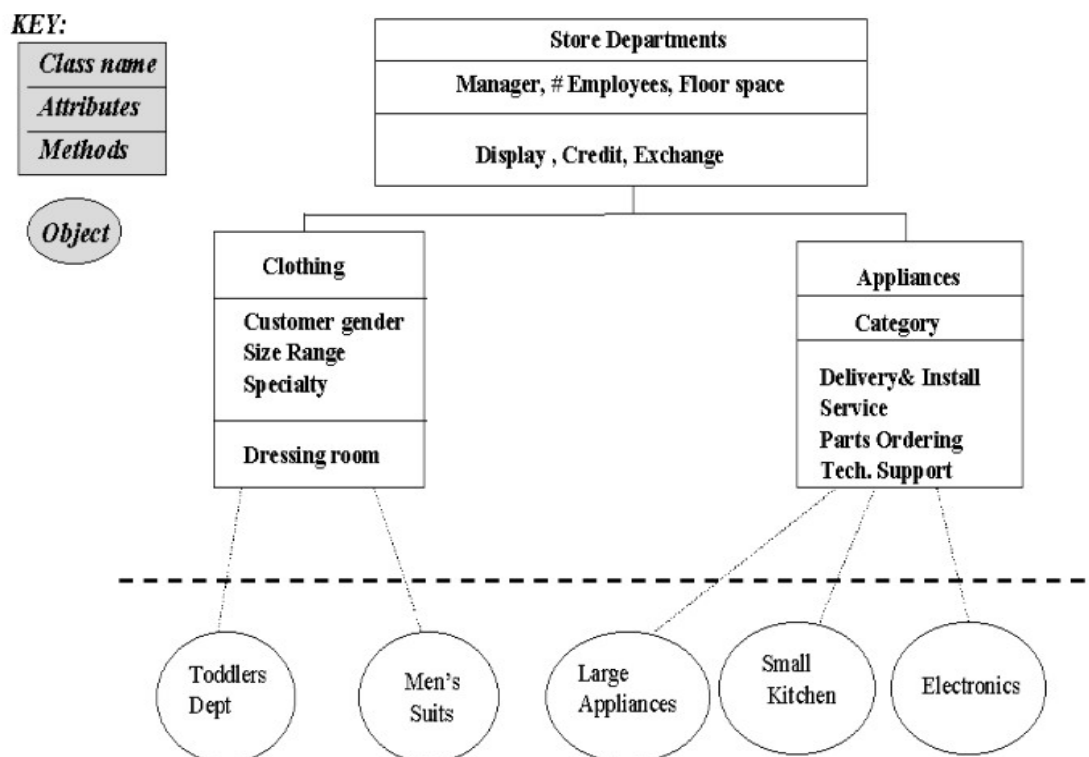
**Tabula 2: Atslēgas termini objektu-orientētam metrikām**

Jaunie objektu-orientētie izstrādes metodes ir sava terminoloģija, kas attēlo jaunus strukturālos komponentus. Attēlā 1 parādīta objektorientēta sistēmā startē parādot klasi (*Class A*) kura satur sevī atribūtus un metodes. Klases izmantotas, kā bāze objektiem. (*Object A1*) Bērnu klase manto visus atribūtus un operācijas, papildus tai ir arī savi atribūti un metodes. Bērnu klase var arī būt par vecāku klasi (*parent class*) citiem klasēm, formējot citu hierarhiska koka zaru struktūrā. Kad objekts ir izveidots, lai glabāt informāciju, tā ir klases uzsākšanas. Klases mijiedarbojās vai komunicē ar ziņojumiem. Kad ziņojums ir nosūtīts starp divām klasēm klases savienojās pāros. Termins ir definēts tabulā 2.



**Attēls 1: Izskaidrojums objektorientētiem terminiem**

Attēls 2 ir parauga lietojums ar 3 klasēm: saknes klase *Store\_dept* un divas bērnu klases *Clothing* un *Appliances*. Katrai nodaļai ir *Manager*, *# Employees* and *Floor space*; katra klase manto atribūtus *Manager*, *# Employees*, *Floor Space* no *Store\_dept*. Klasei *Clothing* ir papildus atribūti *Customer Gender*, *Size range* un *Specialty*. Klasei *Appliances* ir atribūts *Category*. Klases *Clothing* objekti ir *Toddlers Department* un *Men's Suits*. Klasei *Appliances* objekti ir *Large Appliance Department*, *Small Kitchen Appliances*, and the *Electronics Department*.



**Attēls 2: Parauga Lietojums**

# Metrikas priekš objektorientētam sistēmam

## Metodes klasē

Formula: Vidējais skaits metožu objektu klasē = Kopējais metožu skaits / Kopējais objektu-klašu skaits

- Lielāks skaits metožu uz objektu klašu skaits, sarežģītā testēšanu ar lielāku objektu izmēru un sarežģītību
- Ja metožu skaits uz objektu klasēm būs pārāk liels, tad programmu būs sarežģītāki paplašināt
- Liels skaits metožu uz objektu klasēm palielina mantošanu un palielina pirmkoda atkarotu lietošanu

## Mantošanas atkarības

Mantošanas koka dziļums = max ( mantošanas koka ceļa garums)

- Mantošanas koka dziļums ir vairāk ieteicams, pret mantošanas koku platumu. Dziļāks mantošanas koks veicina lielāku metožu sadalību, nekā plašie koki
- Dziļš mantošanas koks ir sarežģītāk testēt, nekā plašu koku
- Saprotamība var būt samazināta ar lielu mantošanas slāņu skaitu

## Savienošana pāros pakāpe starp objektiem

Vidējs atkarību numurs objektā = kopējais numurs loku / kopējais objektu skaits

loks = max(izmantotu loku skaits) -

loks - pievienots jebkuram objektam izmantošanas tīklā

- Lielāka savienošana pāros pakāpe starp objektiem sarežģītā lietojuma uzturamību, jo objektu starpmijiedarbības ir sarežģītākas
- Jo lielāka savienošana pāros objektu pakāpe, jo vairāk objektu var atkārtoti izmantot līdzīgos lietojumos
- nesavienoti objekti var ērti augt, neka objekti ar lielāku atkarību pakāpi
- Testēšana ir sarežģītāka ar vairāk savienotu objektu sistēmu
- Objektu mijiedarbības sarežģītība asociēta ar objektu savienošānu pāros var novest pie lielākā kļūdu skaita izstrādē

## **Saķedēšanas pakāpe starp objektiem - Degree of Cohesion of Objects**

Pakāpe starp objektu saķedēšanu = Kopējais loku skaits priekš visiem objektiem / Objektu skaits

## **Objektu bibliotēkas efektivitāte**

Vidējs numurs = Kopējs skaits atkārtoto objektu / Kopējais numurs bibliotēkas objektu

Ja objekti ir projektēti, lai būtu atkārtoti izmantojami lietojumā, tad efektiem vajadzētu pārdītos objektu bibliotēkas izmantošanas statistikā

## **Faktoringa efektivitāte**

Faktoringa efektivitāte = Unikālo metožu skaits/ kopējais metožu skaits

- Vairāk uzskaitīti lietojumi ir vairāk uzticami. Mazāks implementāciju izvietoējuma vidējai darbībai skaits, jo mazāk iespējamās kļūdas veicot kodēšanu
- Jo vairāk uzskaitīta mantošanas hierarhija, jo lielāka atkarotās metožu izmantojamības pakāpe
- Jo vairāk uzskaitīts lietojums, jo mazāks ir implementāciju izvietoējuma skaits vidējai metodei

## **Atkārtotas lietošanas mantošanas metožu pakāpe**

Procents potenciālai metodes atkārtotas izmantojamības:PP

PP= (kopējs aktuālas metožu atkārtotas izmantošanas skaits / Kopējs skaits potenciālas metodes izmantošanai)x 100

Procents metožu potenciālai pārdefinēšanai (PM):

PM = (Kopējs pārdefinēto metožu skaits / Kopējs potenciālas metožu izmantošanas skaits x )100

- Definējot ka metodes var būt atkārtoti izmantoti, nenozīmē to aktuālo atkārtoto lietojamību

## **Lietojuma granularitāte**

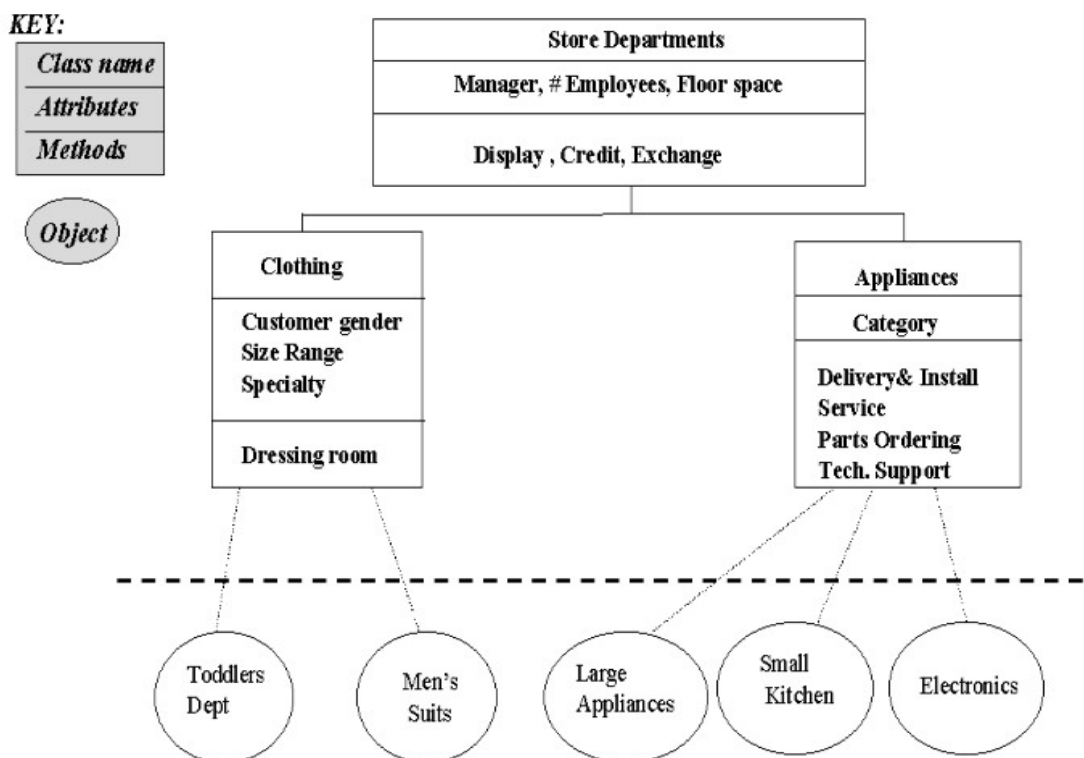
Lietojuma granularitāte = kopējs objektu skaits / kopējs funkcijpunktu skaits

- Viens no objektorientētas projektēšanās mērķiem – labāka detalizācija. Mērķis ir sasniegt lielāko abstrakcijas līmeni, salīdzinot ar procedurālo projektējumu
- Lietojums projektēts ar zemāku funkciju skaitu objektā ir vieglāk uzturams, jo objekti ir mazāki un mazāk sarežģīti
- Vairāki smalki graudaini objekti vairāk atkārtoti izmantojami. Katra objekta uzvedība vieglāk analizējas un saprotas

## **Objektu-orientētas specifiskas metrikas (Chidamber & Kemerer's Metrics Suite )**

Klase ir veidne, no kuras var tikt izveidoti jaunie objekti. Objektu kopa daļa kopējo struktūru un kopējo uzvedību, kas noteic metožu kopa. Metode definē algoritmu, kuru izpilda objekts, reaģējot uz ziņojumu. Ziņojums ir objektu sazināšanās veids. Efektīvs objektorientēts projektējums maksimizē sakļautību, jo tā veicina iekapsulēšanu. Savienošāna pāros ir novertēšana asociācijas spēka ar vienas entitijas savienojamību otrai. Klases savienotas pāros kad ziņojums ir nodots starp objektiem; kad metode deklarēta vienā klasē, izmanto citas klases metodes vai atribūtus. Mantošana ir vēsturiskas attiecības starp klasēm, kas ļauj programmētājiem atkārtoti izmantot iepriekš definētus objektus, iekļaujot operātorus un mainīgos.





3 attēls. Parauga lietojums

### ***Nosvērotas metodes pret klasi (Weighted Methods per Class (WMC))***

WMC ir metožu skaits, kas implementē klasē vai metodes sarežģītību summa. Otrais mērījums sarežģīti implementēt jo ne visas metodes ir pieejamas klases hierarhijā mantošanas dēļ. Metožu skaits un to sarežģītība ir iesaistītas kā parēģošana cik laika ir prasīts lai izveidotu un uzturētu klasi. Jo lielāks metožu skaits klasē, jo lielāka potenciālā ietekme uz bērniem; bērni manto visas metodes, kas definētas vecāku klasē. Klases ar lieliem metožu skaitiem ir vairāk lietojumu specifiskas, ierobežojot atkārtotas izmantošanas iespējas. Uz 3 attēla bāzes ir uzrakstītas metožu skaits katrā klasē.

WMC priekš *Clothing\_dept* = 1

WMC priekš *Appliance\_dept* = 4

### ***Klases atbilde - Response for a Class (RFC)***

RFC ir visu metožu kopējais skaits kuri var būt piesauktas kā atbilde ziņojumam kas nosūtīts klases objektam ar kādu metodi klasē. Tas iekļauj visas pieejamas metodes klases hierarhijā. Šī metrika skatās uz sarežģītības kombinācijām klasē cauri metožu skaitam un komunikācijas apjomam ar citām klasēm. Jo lielāks metožu skaits, kas var izsauktas no klases cauri ziņojumiem – lielāka klases sarežģītība. Ja liels metožu skaits var tikt izsaukts kā atbilde ziņojumiem, klases testēšana un

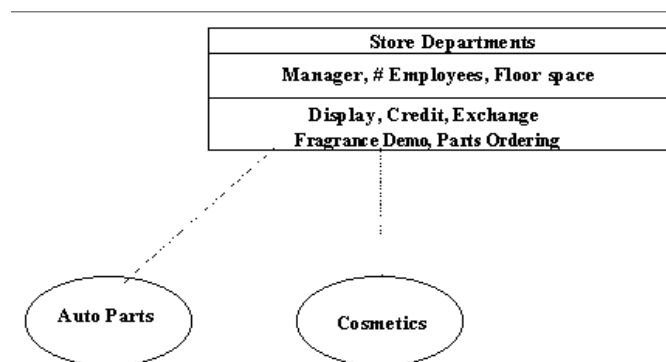
atkļūdošana kļūst sarežģīta un prasīs lielāku testētāja programmas saprotamību.

Klases atbilde priekš *Store\_dept* attēlā 3 ir metožu numurs kurai paši atsaucas uz ziņojumu (*Store\_dept*), uz *Clothing\_dept*, un *Appliance\_dept*.

Klases atbilde (RFC) priekš *Store\_dept* = 3 (self) + 1 (*Clothing\_dept*) + 4 (*Appliance\_dept*) = 8

### **Sakļautības trūkums - Lack of Cohesion (LCOM)**

Sakļautības trūkums mēra metožu nevienādību klasē instances mainīgajos vai atribūtos. Augsti sakļautīgam modulim jābūt vienam; augsta sakļautība parāda labu klašu sadalījumu. Sakļautības trūkums vai zema sakļautība paaugstina sarežģītību, paaugstina kļūdu iespēju rašanos izstrādes procesā. Klases ar zemu sakļautību varētu sadalītas divās vai vairākas apakšklasēs ar paaugstinātu sakļautību. Attēlā 4 ir parādīts alternatīvs programmas projektējums 3 attēla redzamai programmai.



Attēls 5: Alternatīvs projektējums

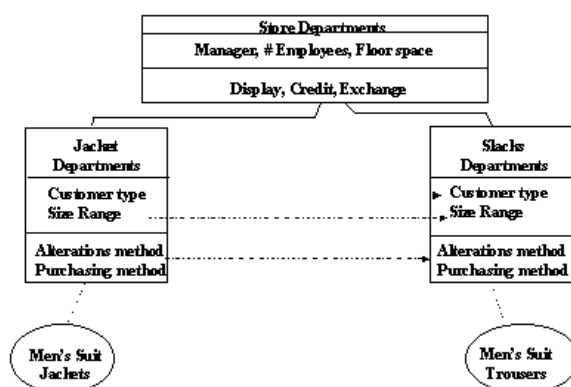
5 Attēlā divas bērnu klases *Clothing* and *Appliances* bija iznīcinātas, atribūti un metodes kombinētas klasē *Store\_dept*. Attēls 5 rada projektējumu ar lielu sakļautības trūkumu, tāpēc ka tur ir maz kopīgu atribūtu un meožu starp objektiem. *Auto\_Parts* nepieciešama metode *Parts\_ordering* bet ne *Fragrance\_Demonstrations*. *Cosmetics* nepieciešama *Fragrence\_Demonstrations* bet ne *Parts\_Ordering*. Jo objektiem maz kopējo metožu – liels sakļautības trūkums. Nepieciešamā tālākā abstrakcija – līdzīgie objekti jāgrupē kopā radot tiem bērnu klases .

### **Savienošana pāros starp objektu klasēm - Coupling Between Object Classes (CBO)**

Savienošana starp objektu klasēm ir klašu skaits kurām klase ir savienota. Tas tiek mērīts ar atšķirīgo nav mantojamo klases hierarhijas skaitīšanu, no kuras klase ir atkarīga. Pārmerīga savienošana kaitē modulārai projektēšanai un noverš atkarotu lietošanu. Jo vairāk neatkarīga klase,

jo vieglāk to atkaroti izmantot citā lietojumā. Jo lielāks pāru skaits, augstāka ir jūtība pret izmaiņām citās projektējuma daļās un uzturēšana ir sarežģītāka. Stipra savienošana sarežģī sistēmu, jo klasi ir grūtāk saprast, izmainīt vai koriģēt, ja ta ir savstarpēji saistīta ar citām klasēm. Sarežģītību var pazemināt projektējot sistēmu ar zemu iespējamu savienošānu. Tas uzlabo modularitāti un veicina iekapsulēšanu.

Attēls 5 pārspīlēts piemērs par augstu savienošānu starp objektiem. Divas nodaļas *Jackets* un *Trousers*. Tiem ir vienādi atribūti un metodes. Attēlā 6 aplūkots neefektīvs projektējums un šīs divas nodaļas vajadzētu apvienot vienā klasē.



Attēls 5: Piemērs par mērīgai savienošānai

### ***Mantošanas koka dziļums - Depth of Inheritance Tree (DIT)***

Mantošanas koka dziļums klasē mantošanas hierarhijā ir maksimāls soļu skaits no klases mezgla līdz koka saknei un mērojams priekšteču klases skaitam. Jo dziļāka klases hierarhija, jo lielākais metožu skaits var mantot cita klase darot vēl sarežģītāku to uzvedības pārēģošāna. Dziļākajam kokam ir sarežģītāks projektējums; kad vairāk metodes un klases ir iesaistītas un lielāks ir potenciāls atkārtoti izmantot. Atbalstoša metrika Mantošanas koka dziļumam ir Mantoto metožu skaits. 3 attēlā *Store\_Dept* ir sakne un koka dziļums ir 0. DIT *Clothing* ir 1.

### ***Bēru skaits - Number of Children (NOC)***

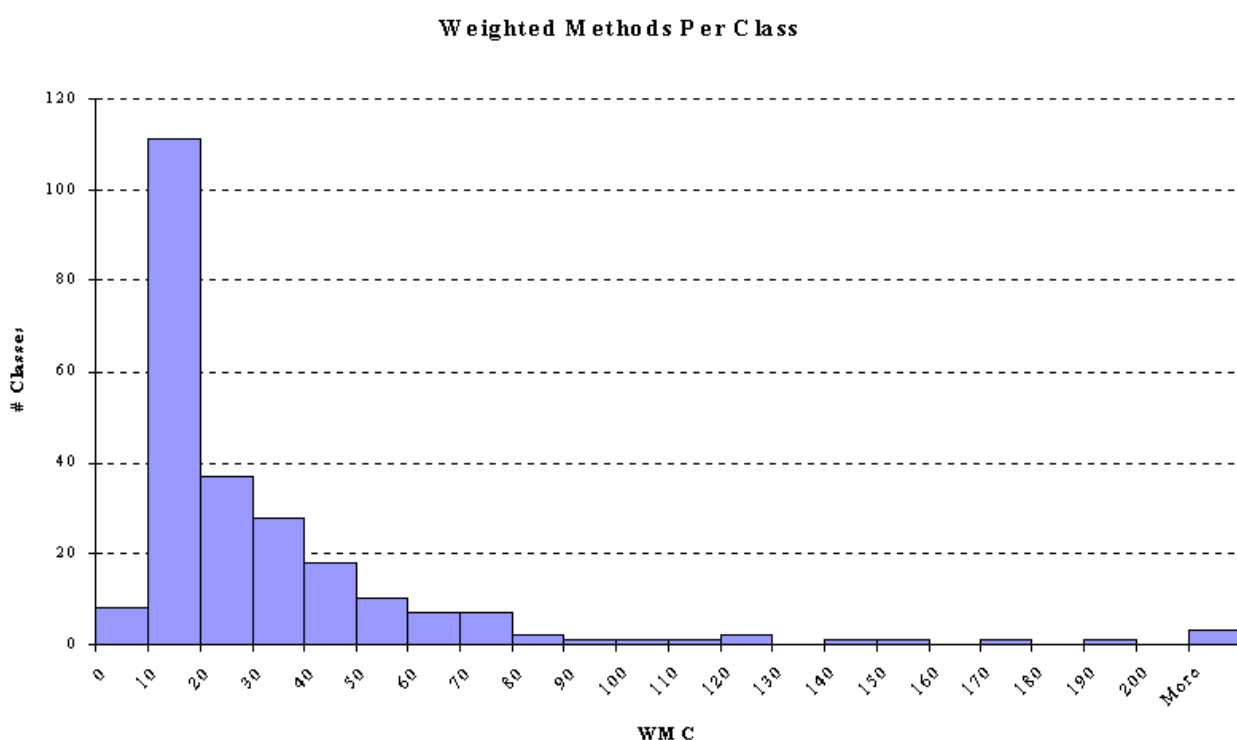
Bēru skaits ir tūlītējo apakšklašu skaits, kas pakļauti klasei hierarhijā. Tas ir indikators klases potenciālai ietekmei uz projektējumu un sistēmu. Jo lielāks bērnu skaits, jo lielāka varbūtība nepareizai vecāku abstrākcijai un var būt par iemeslu apakšklases ļaunprātīgai izmantošanai. Ja lielāks bērnu skaits, lielāka atkarota lietojamība, jo mantošana ir atkārtotas lietojamības forma. Ja klasei ir liels bērnu skaits, tas var palielināt testēšanas laiku, jo dziļas klases metožu testēšana

aizņēm vairāk laika.

Attēlā 3, *Store\_Dept* ir 2 bērnu skaits( NOC of 2.) Bērnu skaits priekš *Clothing* ir 0.

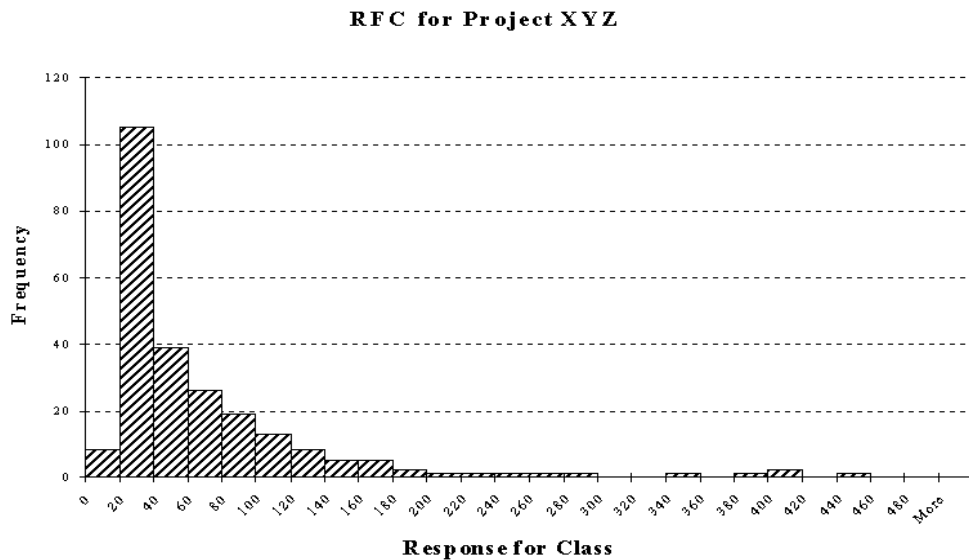
## Pielietojums

Dažām metrikām parastas histogrammas demonstrē dominējošas un ekstrēmālas vērtības. Projektam, kas attēlots 6 attēlā. Nosvērtas metodes uz klasi atklāj ka vairākos gadījumos klasēm ir WMC zemāks par 20, un dažām klasēm ir lielāks nekā 100. Dažas klases ar vislielāko WMC ir kandidāti uz inspekciju. Histogramma arī noderīga priekš sarežģītības laikā mērīšanas.



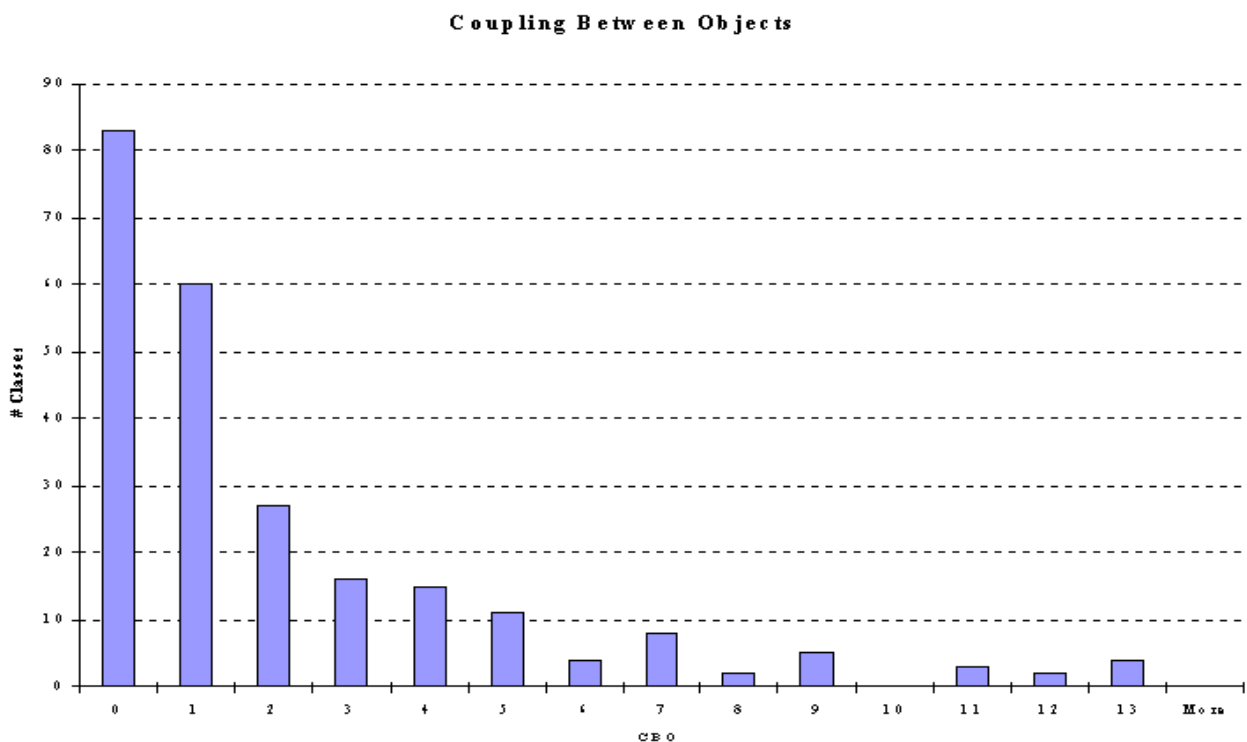
Attēls 6: Nosvērotas metodes klasei

Dažiem klasēm projektā kurš ir pārādīts 7 attēlā ir vairāk nekā 200 metodes. Klasēm ar lielo Klases atbildes skaitu ir lielāka sarežģītība un samazināta saprašana.



Attēls 7: Atbilde klasei

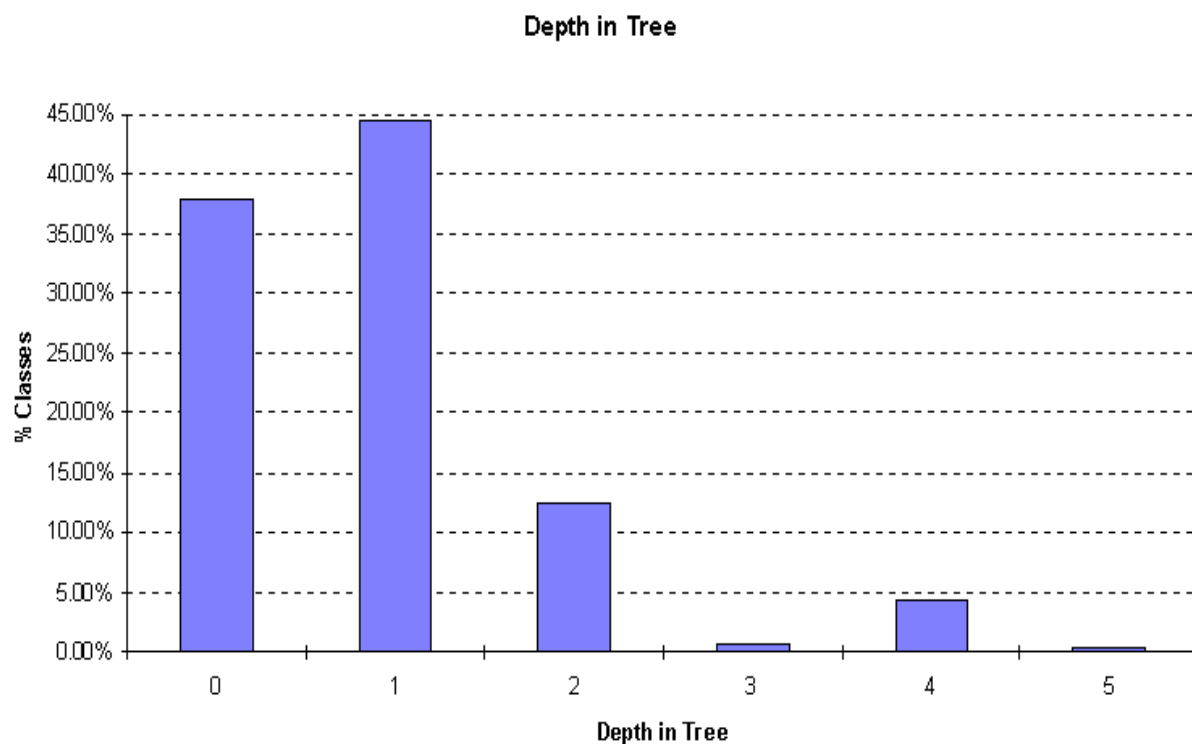
Attēls 8 attēlo Savienošānu starp objektiem CBO. Starp 240 projekta klasēm, vairāk nekā trešdaļa ir patstāvīgas. Augstāka CBO rāda grūti saprotamas klases, kuras grūti uzturēt un atkārtoti izmantot.



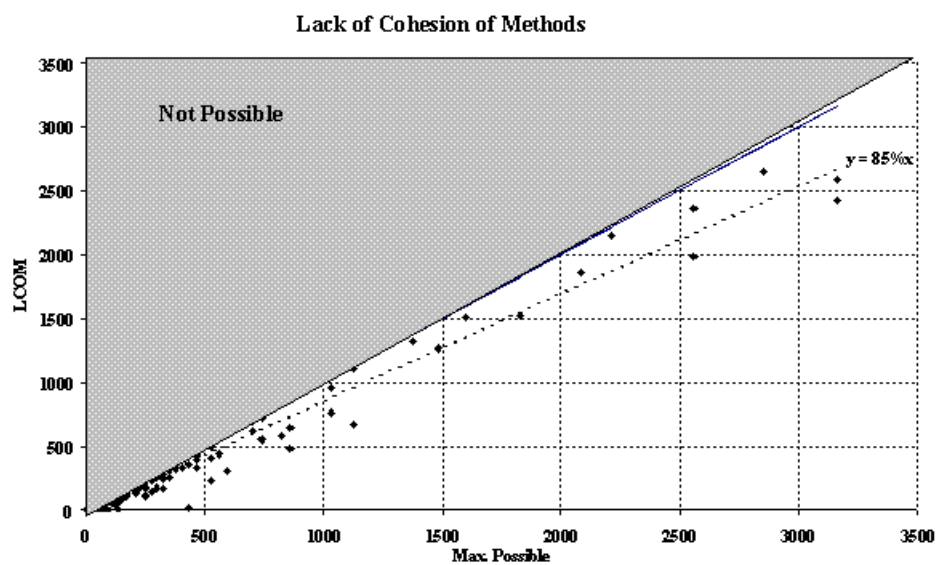
Attēls 8: Savienošāna pāros starp objektiem

Bēru skaits metrikas par hierarhisko struktūru. Bēru skaits (NOC) un Koka dziļums (DIT) var būt grafiski attēloti uz 9 attēlā. Klase ar koka dziļumu DIT = 0 ir hierarhijas sakne. Ja tas arī ir lapa,

NOC = 0, tad pirmkods ne gūst labumu no mantošanas var atkārtotas lietošanas. Gandrīz 66% no projekta klasēm ir zem citām klasēm kokā, kas veido vidējo atkārtotas lietošanas līmeni. Lielākie koka dziļuma radītāji būtu radījuši lielāko atkārtotas lietojamības pakāpi, bet arī lielāko sarežģītību. Attēlā 10 grafiks mērīto sakļautības trūkumu savieno ar iespējamo maksimumu. Mazāks sakļautības trūkuma vērtība ir labāka.

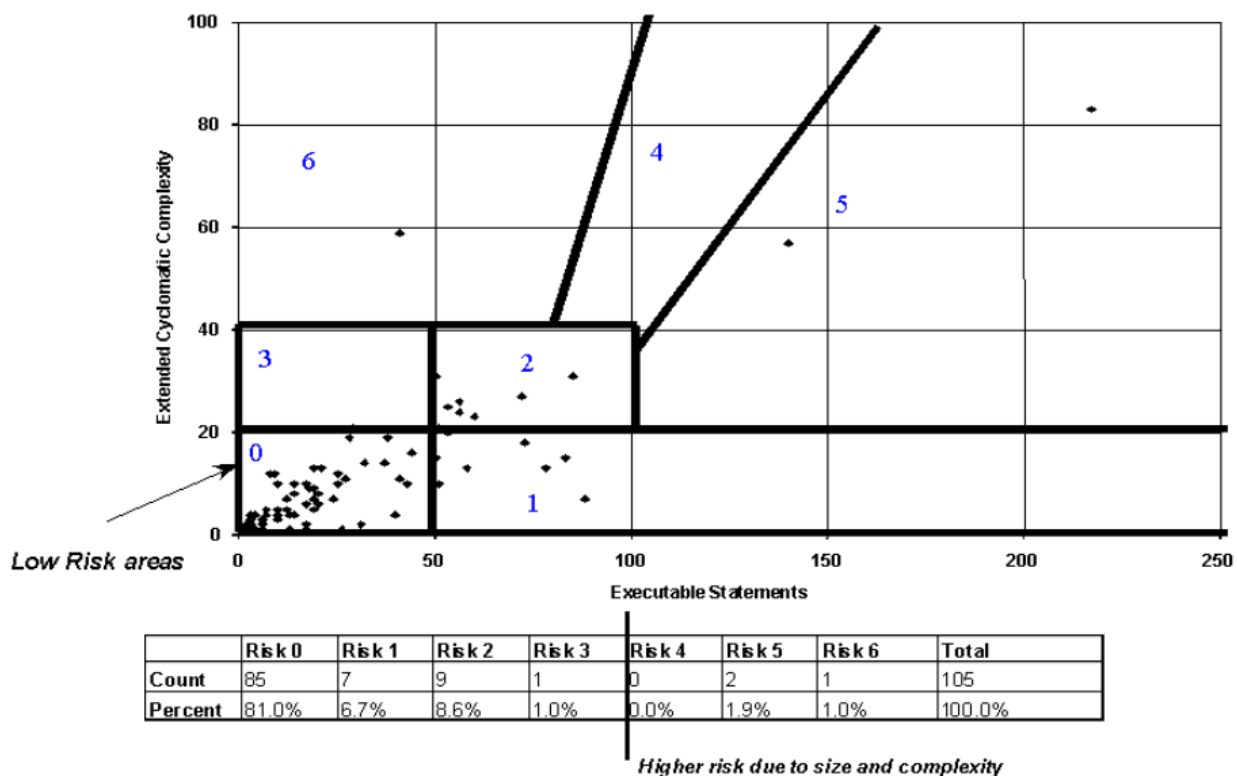


Attēls 9 : Koka dziļums



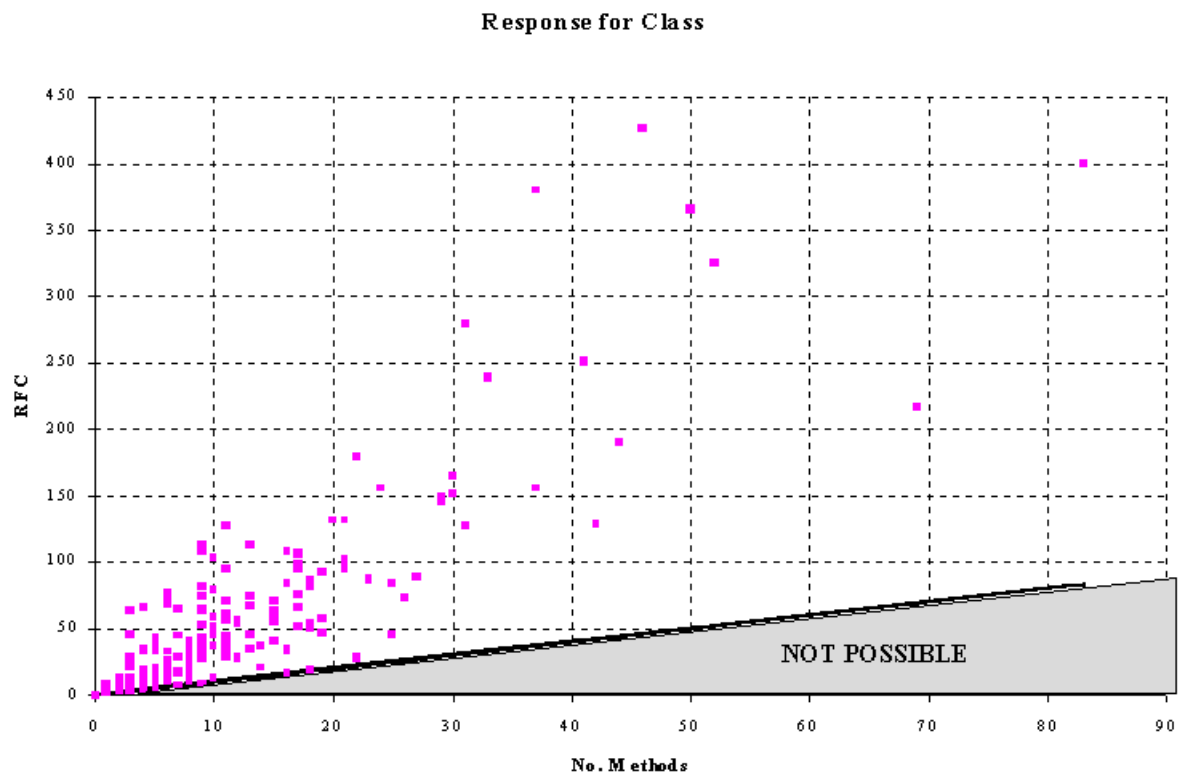
Attēls 10: Metožu sakādes trūkums

Daudziem metrikām ir labāk analizēt moduļus izmantojot 2 metrikas. 11 Attēlā ir parādītas metodes bāzētas uz izmēra un sarežģītības. Riska reģioni norāda, kur metodēm ir potenciāla zema kvalitāte un tas var ietekmēt uzturamību, atkārtoto lietošanu un lasāmību.



Attēls 11: Izmēra un sarežģītības salīdzināšana

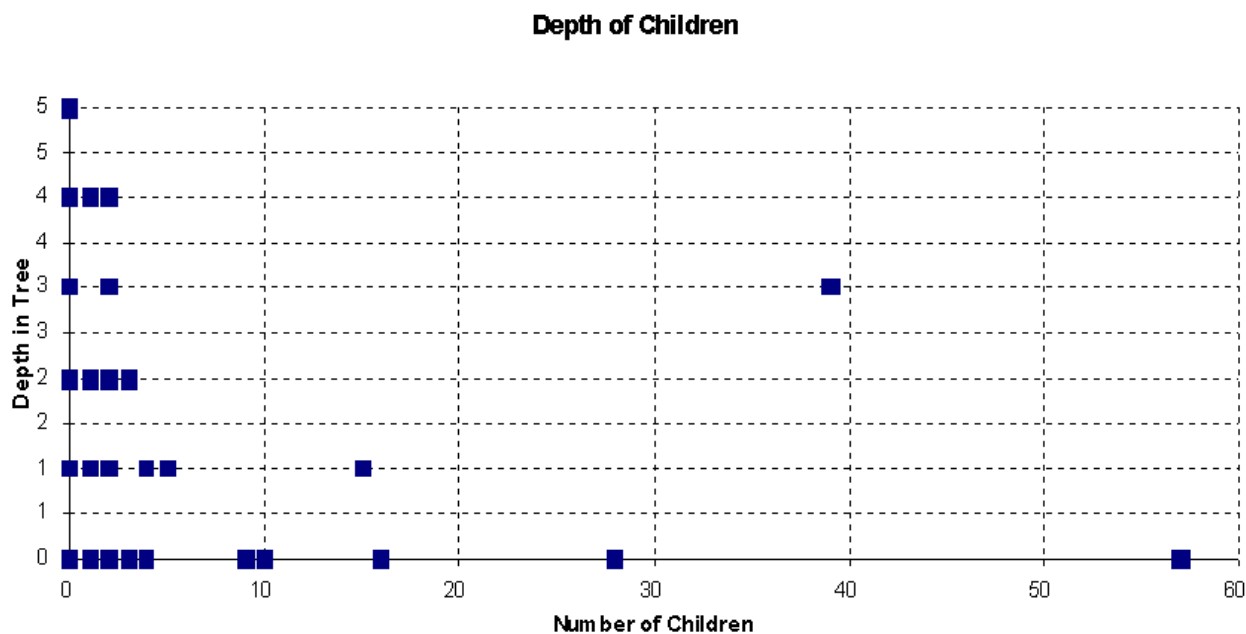
Attēlā 12 Klases ir izteiktas grafikā pret metožu skaitu. Punkti līdzās tuvai “iespējamai” līnijai attēlo klases kuras nav iekļauj ārpusejas metodes. Tas norāda izstrādātājiem, kā dažas klases ar vairāk nekā 40 metodēm ietekmē objektus citās klasēs. Tie arī ir kandidāti uz testēšanu un inspekciju.



Attēls 12: Klases atbildes metožu skaits

Ir kompromiss, kad nosaka bērnu skaitu un koka dziļumu. Lielākais koka dziļuma skaits nosaka kompromisu starp lielāko sarežģītību un atkaroto lietojamību. Lielāks bērnu skaits norāda atkārtoto lietojamību, bet pieļauj vairāk testēšanas. Attēlā 13 demonstrē ka var tikt identificēta klase, tālākai pētīšanai ar 40 bērniem.





Attēls 13: Hierihiskā novērtēšana

## Secinājumi

Objektorientētas metrikas var novērtēt kādas izstrādes un testēšanas pūles vajadzīgas, saprotamību, uzturamību un atkārtotu lietošanu. Objektorientētas metrikas eksistē, lai piedāvātu vērtīgu informāciju objektu-orientētiem izstrādātājiem un projektu vadītājiem. Objektorientētas metrikas piedāvā daudzus paņēmienus, priekš vērtīgas izstrādes informācijas iegūšanu, kas atvieglo izstrādātājiem darbu.

## **Literatūtas saraksts**

1. Software Measurement Page, Object Oriented Metrics -  
<http://yunus.hacettepe.edu.tr/~sencer/oom.html>
2. Applying and Interpreting Object-oriented Metrics -  
[http://satc.gsfc.nasa.gov/support/STC\\_APR98/apply\\_oo/apply\\_oo.html](http://satc.gsfc.nasa.gov/support/STC_APR98/apply_oo/apply_oo.html)