



Programmatūras attīstības tehnoloģijas

Dr.sc.ing., asoc. prof. Oksana Nikiforova

DITF LDI

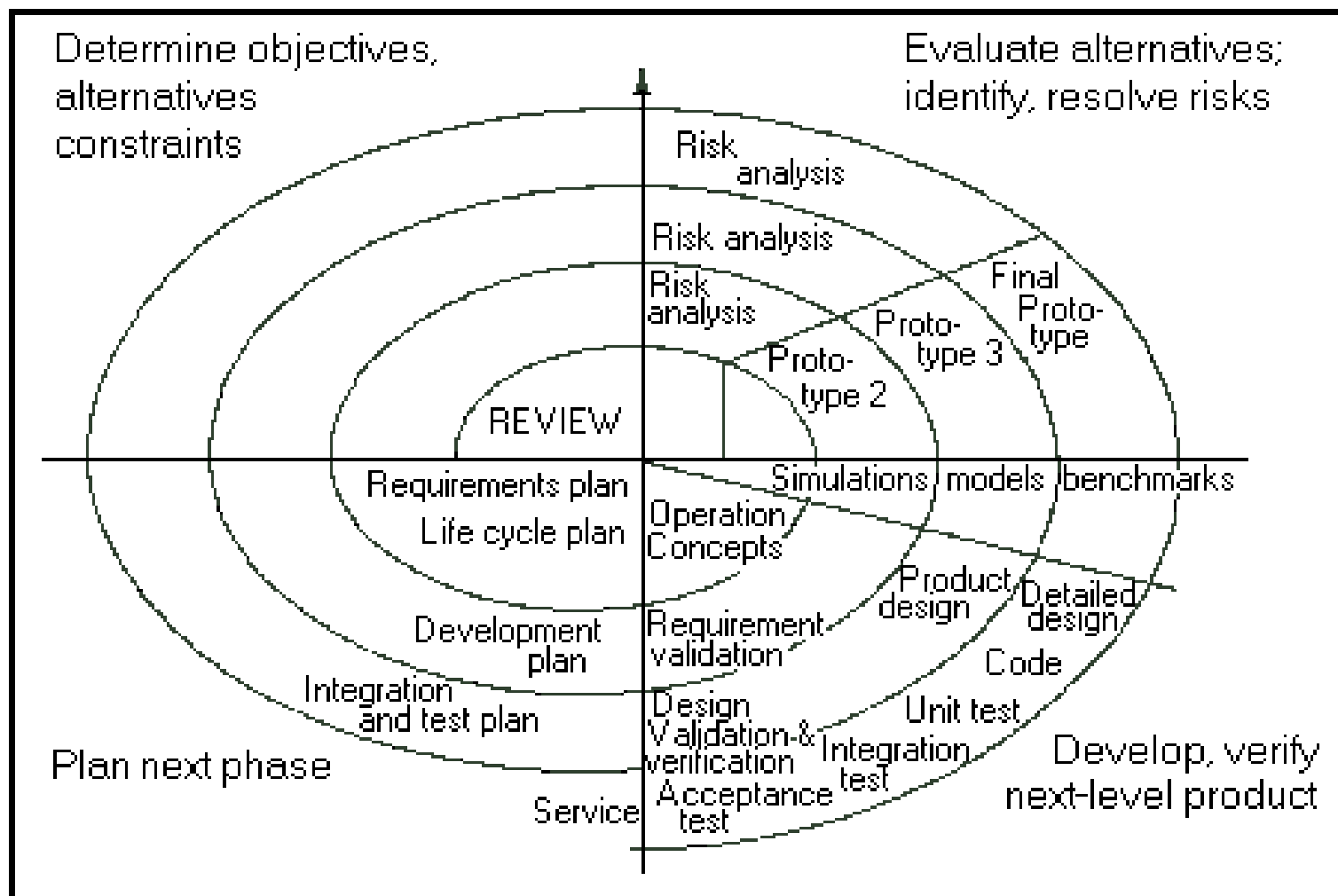
Lietišķo datorzinātņu katedra

Rīga - LV1048, Meža 1/3, 510.kab., tel.67 08 95 98

oksana.nikiforova@rtu.lv

"Spirāles" modelis

Boehm B.W. - 1988



Izstrādes stratēģija (1/2)

■ Katra no fāzēm attīstās četros posmos:

- **Plānošana**: determinēti mērķi, alternatīvas realizācijai un nosacījumi.
- **Riska analīze**: noteiktu iepriekšējā fāzē projekta attīstību alternatīvu novērtēšana attiecībā uz uzstādīto mērķi, nosacījumiem un mērķiem.

Riskus var sadalīt piecās kategorijās:

- Riski, kas ir saistīti ar prasībām.
- Vadības riski.
- Tehniskie riski.
- Riski, kas ir saistīti ar personāla kvalifikāciju.
- Politiskie riski.

Izstrādes stratēģija

■ Katra no fāzēm attīstās četros posmos:

- **Izstrāde**: ja nolemts, ka risks ir pārāk augsts vai ir beigusies nepieciešamība pēc šī produkta, tad projekta attīstība ir pārtraukta. Ja nē, tad tiek izstrādāti šīs fāzes artefakti.
- **Novērtēšana**: projekts tiek novērtēts derīguma ziņā un notiek plānošana uz nākamo vijumu. Pasūtītājs ir iepazīstināts ar produktu un var izteikt rekomendācijas izmaiņām.

Ātrā programmatūras izstrāde (Rapid Application Development)

■ 3 pamata elementi

- Neliela izstrādātāju komanda (2-10 cilvēki)
- Īss detalizēti izstrādāts darba grafiks (2-6 mēneši)
- Izstrādātāji pakāpeniski pieprasa un realizē produktā lietotāja prasības

RAD - dzīves cikls (1/4)

■ Prasību analīze un plānošana

- ☐ Projekta realizācijas iespēju novērtēšana
- ☐ Sistēmas funkciju saraksts ar prioritātēm
- ☐ Sākotnējais sistēmas funkcionēšanas modelis
- ☐ Izstrādes grafiks un termiņi

RAD - dzīves cikls (2/4)

■ Projektēšana

- Sistēmas procesu detalizēta analīze (ekrāns, dialogs, neskaidrības un "neviennozīmības")
- Sistēmas sadalīšana apakšsistēmas (tā lai tās realizācija iekļautos 60-90 dienās)
- Definēti interfeisi starp apakšsistēmām, kas ir realizēti ar noteikta CASE rīka palīdzību

RAD – dzīves cikls (3/4)

■ Konstruēšana

- Programmas kods daļēji tiek automātiski ģenerēts balstoties uz projektēšanas informāciju
- Testēšana notiek tieši izstrādes procesā
- Lietotāji var iepazīties ar produkta tapšanu un ieviest izmaiņas
- Apakšsistēmu integrācija kopējā produktā
- Produkta dokumentācijas izstrādes pabeigšana

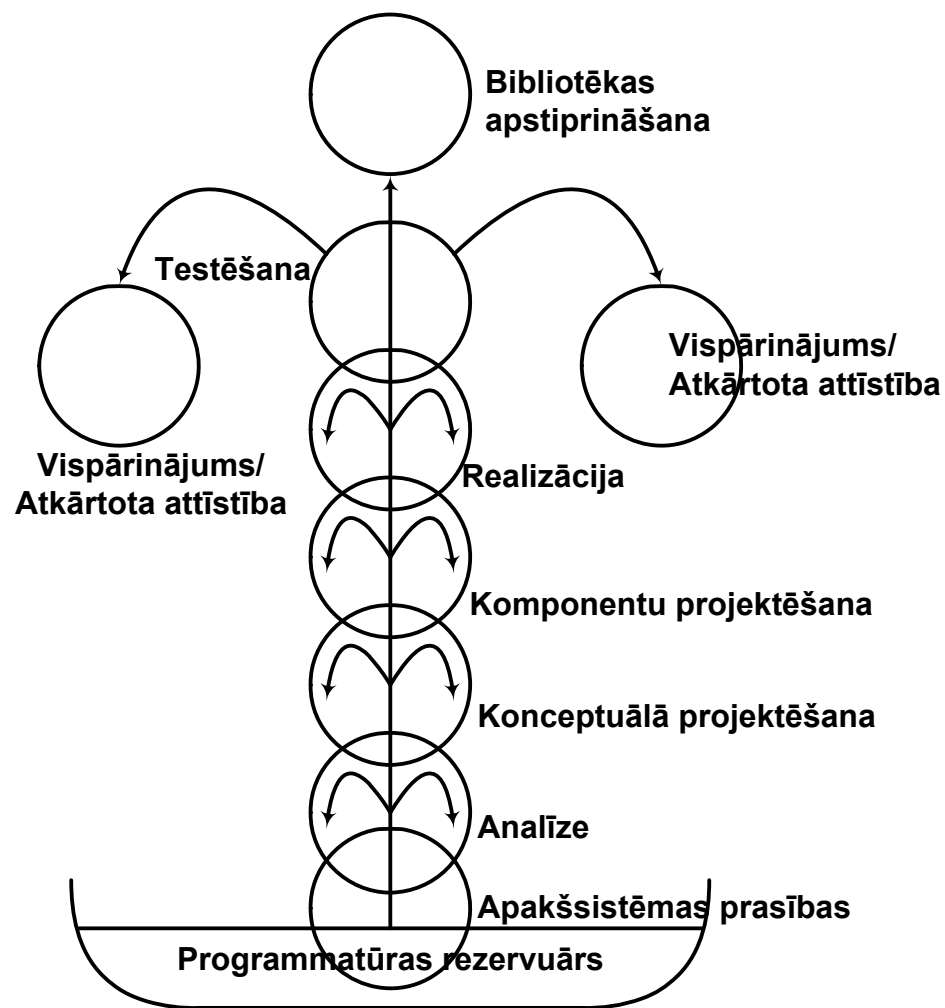
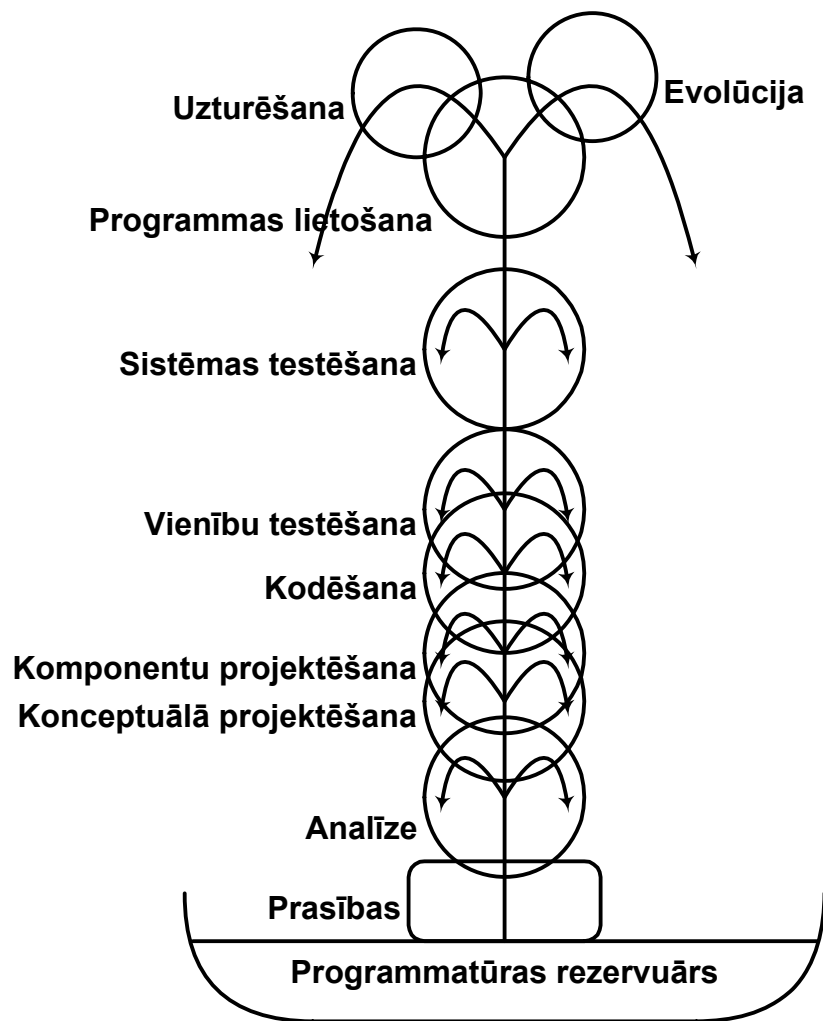
RAD - dzīves cikls (4/4)

■ Ieviešana

- ☐ Lietotāju apmācība
- ☐ Uzturēšana līdz neparādās programmas jaunā versija

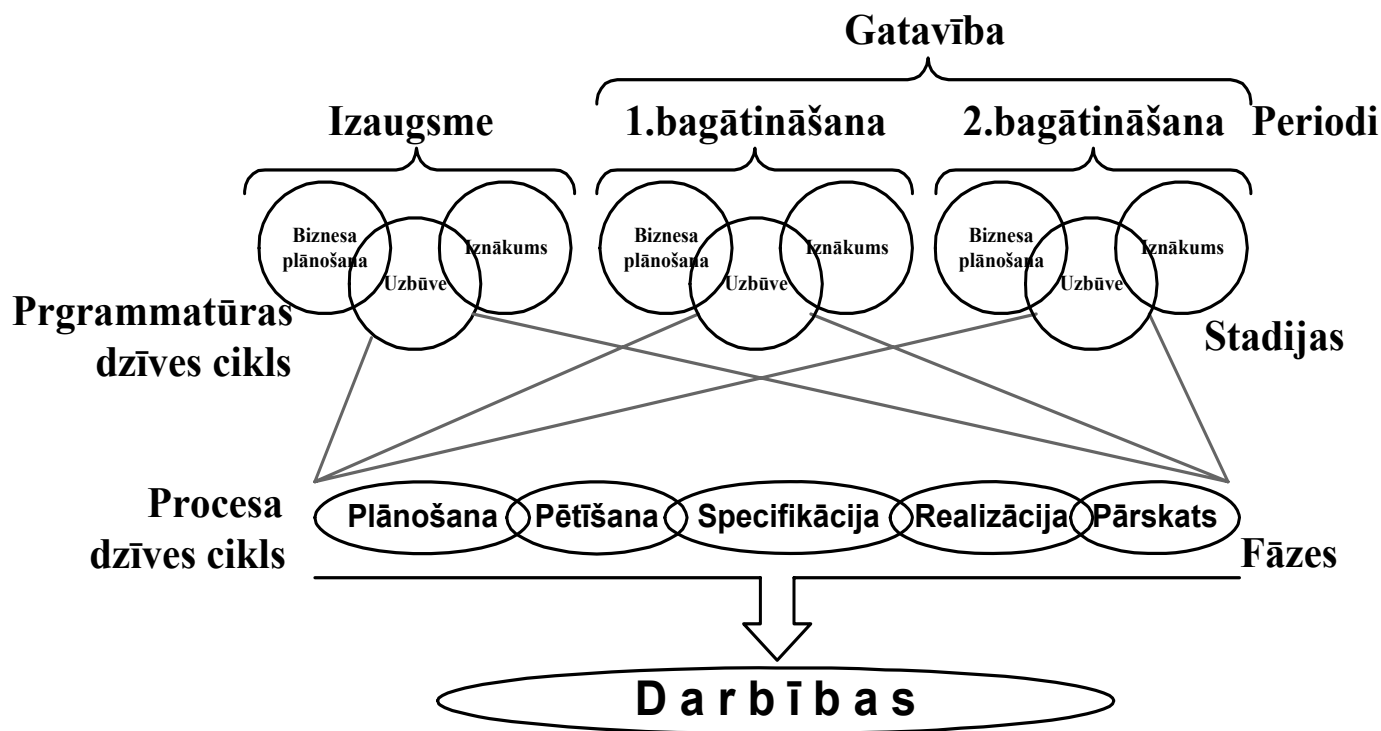
Strūklakas modelis

Henderson-Sellers B., Edwards J.M. - 1993



MOSES

Henderson-Sellers B., Edwards J.M. - 1994

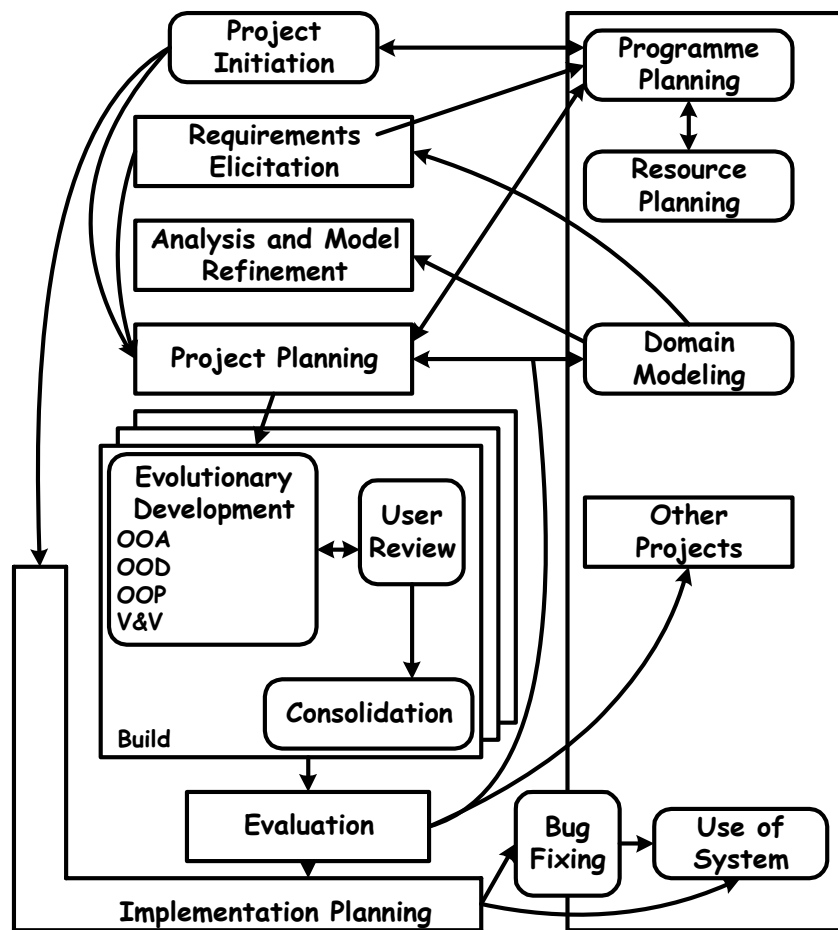


MOSES aktivitāšu piederība fāzēm

Aktivitāte	Plānošana	Pētīšana	Specifikācija	Realizācija	Pārskats
Lietotāja prasību noskaidrošana		1			
Scenārija izstrāde		2			
Objektu/klašu identifikācija			1		
Mijiedarbības specifikācija			2		
Notikumu modeļa konstruēšana			3		
Noskaņas (<i>genericity</i>) specifikācija				1	
Translācija programmēšanas valodā				2	
Pakalpojumu identificēšana			4	3	
Objektu diagrammas konstruēšana			5		
Līguma specificēšana			6		
Mantošanas identifikācija			7	4	
Klašu bibliotēku apvienošana			8		
Vispārinājums atkārtotai lietošanai			9		4
Optimizācija			10	5	
Testēšana			11	6	1
Apakšsistēmu identifikācija	1		12		
Apakšsistēmu koordinācija	2				
Iterācijas plāna izstrāde	3				
Dokumentācijas pārskats	4	3	13	7	2
Kvalitātes novērtēšana (Metrikas)	5	4	14	8	3; 5

Object-Oriented Process, Environment, Notation (OPEN)

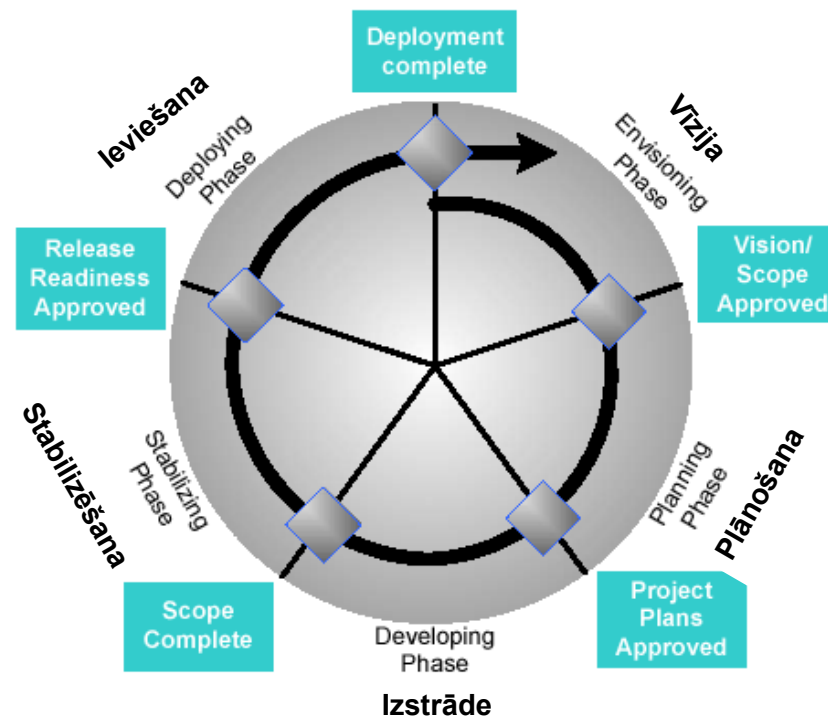
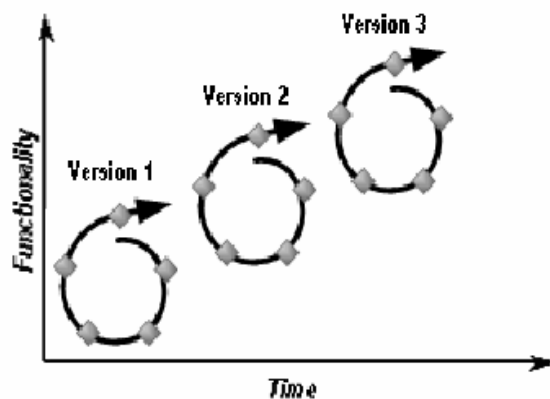
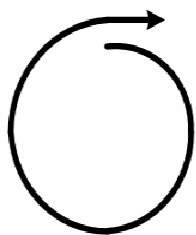
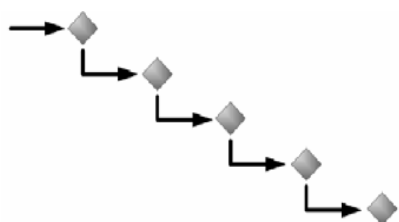
Henderson-Sellers B., Graham I., Nerson J.M., 1996



- OPEN process OPF (OPEN Process Framework) ir definēts karkass, no kura var tikt ģenerēti specifiskie procesi (jeb procesa eksemplāri).
- Process ir nodrošināts ar noteikto aktivitāšu skaitu, kas ir objekti ar pirms- un pēc- nosacījumiem.
- Katrai aktivitātei ir definēti uzdevumi izpildīšanai

Microsoft risinājumu karkass

Microsoft Solutions Framework – MSF (1995)



IBM/Rational vienots process

IBM/Rational Unified Process – RUP (1998)

Organizācija laika secībā

