

# **Objektorientētas programmēšanas praktikums (studiju projekts)**

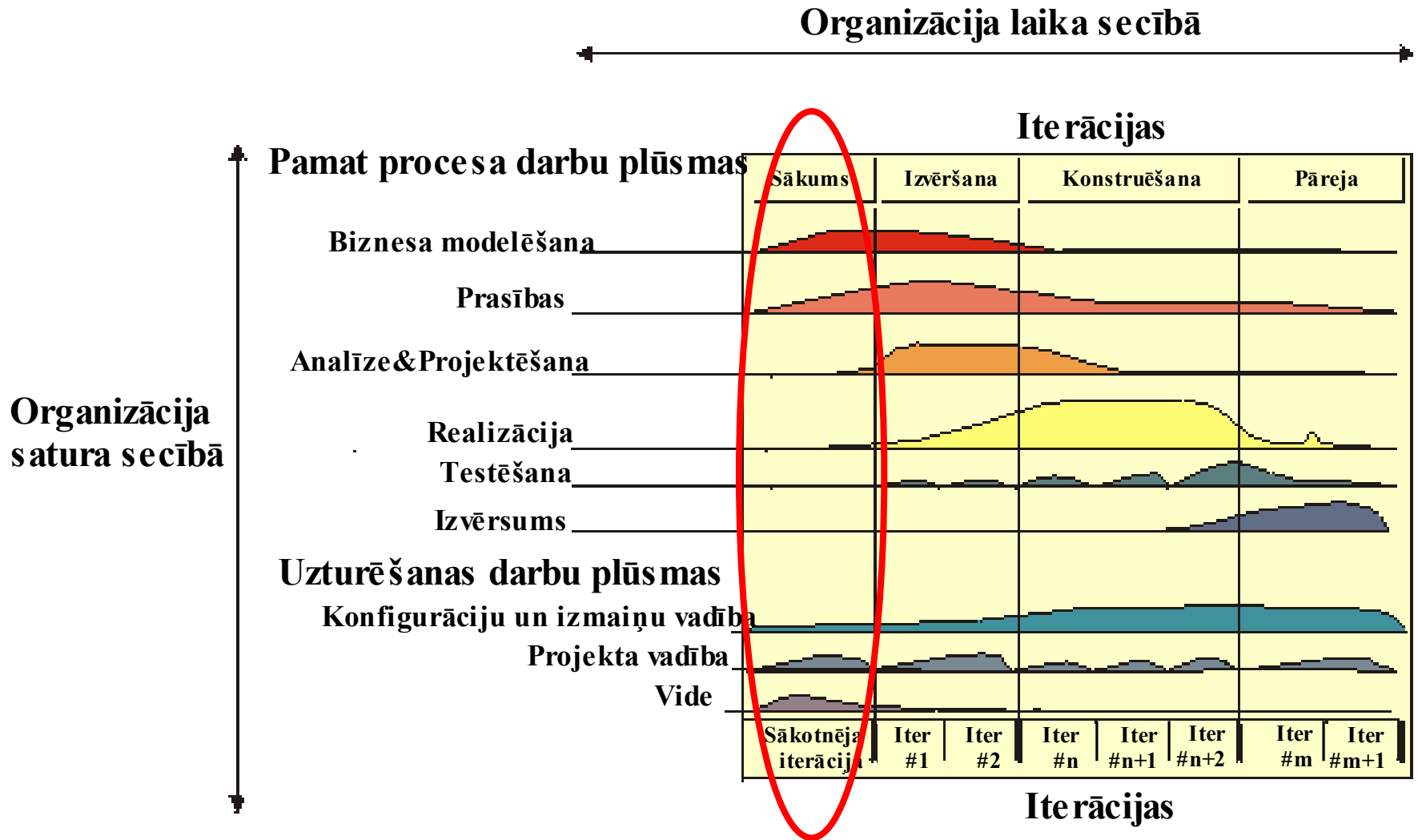
**Dr.sc.ing., prof. Oksana Nīkiforova**

**Rīgas Tehniskā universitāte  
Lietišķo datorsistēmu institūts  
Lietišķo datorzinātņu katedra**

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

**[oksana.nikiforova@rtu.lv](mailto:oksana.nikiforova@rtu.lv)**

# RUP sākuma fāze



# Sākuma fāzes mērķi

- Saprast kas ir jābūvē
- Identificēt sistēmas kritiskos lietošanas gadījumus
- Identificēt vismaz vienu iespējamo risinājumu
- Identificēt izdevumus, laika plānojumu un riskus
- Izlemt kādiem procesiem sekot un kādus rīkus izmantot

# Iterāciju skaits sākuma fāzē

- Vairums projektu ir tikai viena iterācija sākuma fāzei
- Vairākas iterācijas iespējamās gadījumos, ja:
  - Projekts ir liels
  - Sistēma ir savdabīga un/vai unikāla
  - Vairākas ieinteresētās puses, ko saista sarežģīti kontrakti
  - Tehniskie riski

# Sākuma fāzē veidoti artefakti

- Jāvienojas par projekta vīziju visaugstākajā līmenī
  - Vīzijas dokuments (Vision scope)
- Jāizstrādā “jūdzi garš, collu dziļš” sistēmas apraksts
  - “Jūdzi garš, collu dziļš” apraksts

# Vīzijas dokuments

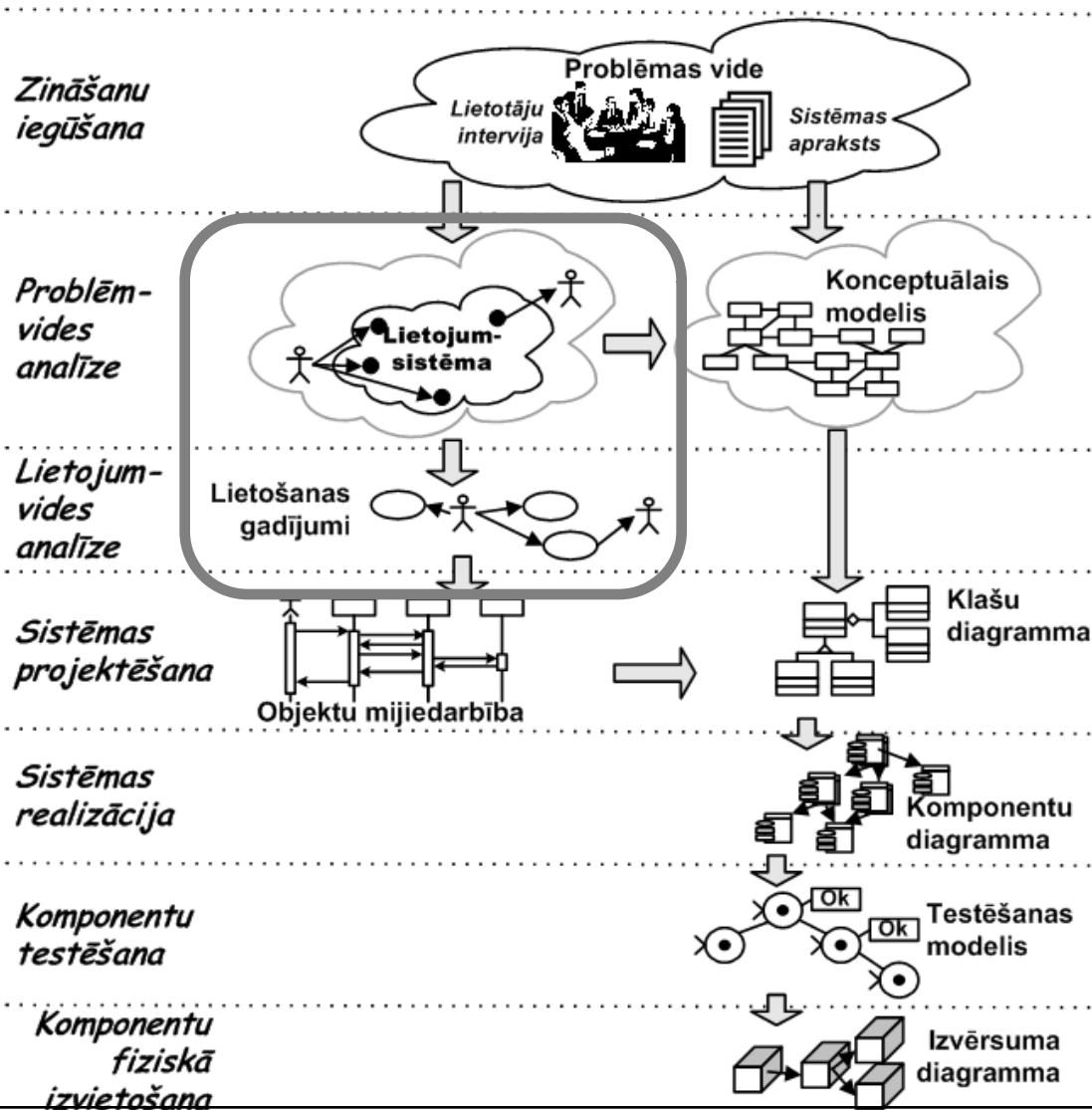
1. Ieguvumi un iespējas ko sniedz projekta izstrāde
2. Problēmas kādas programmaprodukts atrisina
3. Kurš lietos sistēmu
4. Augstā līmenī jāpasaka, ko produkts darīs. Iezīmējot galvenos lietošanas gadījumus.
5. Svarīgākās nefunkcionālās prasības, tādas kā – atbalstītās operētājsistēmas, datubāzes, sistēmas uzticamības (reliability) līmenis, licencēšana un izcenojumi, ja tie projektā ir aktuāli.

*Projektu prezentācijas 1.lekcijā – viens no vīzijas variantiem.*

# Sistēmas konteksta saprašana

- Sistēmas izstrādātāji orientējas IT jomā, taču efektīva izstrāde prasa labu orientāciju sistēmas konteksta jomā, to var panākt ar:
  - Biznes-procesu modelēšanu – aprakstā kādi procesi notiek sistēmā, kādi ir katra darbinieka funkcionālie pienākumi. Analīzes gaitā šie procesi tiks vērtēti no automatizācijas iespējas viedokļa, tie, ko ir iespējams automatizēt tiks pārveidoti par sistēmas lietošanas gadījumiem
  - Problēm-vides konceptuālo modelēšanu – apraksta kādi objekti (konceptijas) eksistē sistēmā, kā tie ir saistīti savā starpā, palīdz izveidot terminu vārdnīcu. Analīzes gaitā šīs konceptijas atlases gaitā tiks pārveidotas par klasēm.

# Programmatūras izstrāde ar UML



(c) Nikiforova 2002

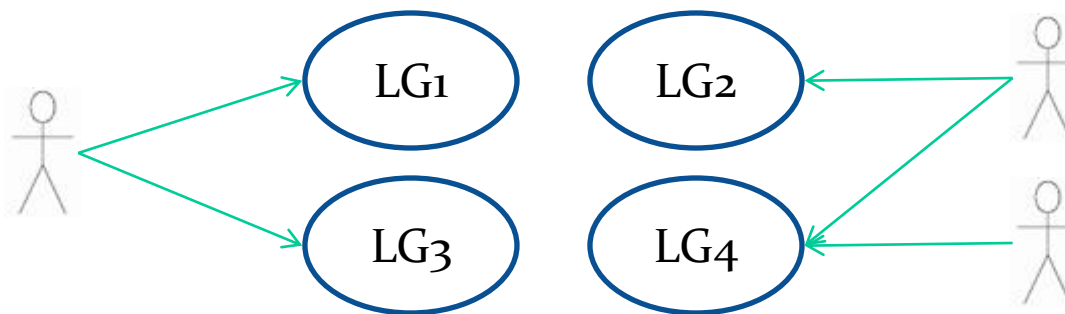


# “Jūdzi garš, collu dziļš” dokuments [1/2]

- Veicamās aktivitātes veidojot aprakstu:
  - Identificēt un vispārīgi aprakstīt aktierus
  - Identificēt un vispārīgi aprakstīt lietošanas gadījumus
- Apraksta vēlamie raksturlielumi:
  - Pāris paragrāfu par katru lietošanas gadījumu ir pietiekami
  - Izdalīt 20% kopu ar vissvarīgākajiem lietošanas gadījumiem
- Apraksta izveides tehnika
  - Maziem projektiem – Prāta vētra (Brain Storm)
  - Lielākiem projektiem – Darbnīca (Workshop)

# “Jūdzi garš, collu dziļš” dokuments [2/2]

- Dokumenta pamatu veido 4 soļos:
  1. Identificēt maksimāli daudz aktierus
  2. Asociēt aktierus ar lietošanas gadījumiem
  3. Meklēt identificētos lietošanas gadījumos trūkstošos aktierus
  4. Aprakstīt aktierus un lietošanas gadījumus

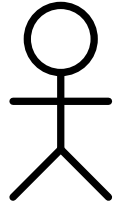


# **Lietošanas gadījumu diagramma**

- Lietošanas gadījumu diagramma (Use – Case Diagram) apraksta sistēmas ārējo uzvedību – reālus sistēmas lietotājus un funkcijas, kuras sistēma var tiem sniegt
- Lietošanas gadījumu diagrammas konstruēšanas mērķi:
  - noteikt modelējamās problēmvides kontekstu un robežas
  - formulēt sistēmas funkcionālas prasības
  - izveidot sākotnējo sistēmas modeli, kas ir paredzēts tas tālākai detalizācijai loģiskajos un fiziskajos modeļos
  - sagatavot sākuma dokumentāciju, kas ir nepieciešama sistēmas izstrādātājiem un pasūtītājiem
- Lietošanas gadījumu diagramma attēlo sistēmā paredzētas darbības (lietošanas gadījumus), to apkārtni (aktierus) un attiecības starp tiem

# Lietošanas gadījumu diagrammas elementi

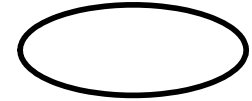
- Par aktieri tiek nosaukta jebkura būtne, kas sadarbojas ar sistēmu no ārienes
- Par aktieri var būt:
  - cilvēku lomas
  - datorsistēmas
  - elektriskās vai mehāniskās iekārtas
  - citas sistēmas
- Aktierus ir iespējams identificēt no problēmas nostādnes, uzstādot jautājumus:
  - Kas ir ieinteresēts konkrēta prasībā?
  - Kam ir labums no sistēmas lietošanas?
  - Kas piegādā sistēmu ar informāciju, lieto informāciju un dzēs informāciju?
  - Kas uztur un nodrošina sistēmu?
  - Vai sistēma izmanto ārējus resursus?
  - Vai viena persona tēlo dažās dažādās lomas?
  - Vai dažas personas tēlo vienu un to pašu lomu?



Lietotājs

# Lietošanas gadījumu diagrammas elementi

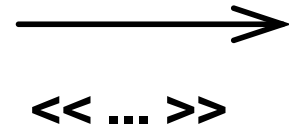
- lietošanas gadījums ir sistēmas darbību secība, kas dod ietekmīgu rezultātu noteiktam aktierim
- viens lietošanas gadījums aptver konkrētu lietotāja un sistēmas sadarbības ķēdīti, kuras rezultātā tiek realizēta konkrēta funkcija vai paveikts noteikts darbs
- Lietošanas gadījumus ir iespējams identificēt no problēmas nostādnes, uzstādot jautājumus:
  - Kas ir katra aktiera uzdevums?
  - Vai kāds aktieris radīs, mainīs, glabās, dzēs, lasīs informāciju sistēmā?
  - Kāds lietošanas gadījums radīs, mainīs, glabās, dzēsīs, lasīs šo informāciju sistēmā?
  - Vai kādu aktieri ir nepieciešams informēt par kādu notikumu sistēmā?
  - Kāds lietošanas gadījums uzturēs un nodrošinās sistēmu?
  - Vai visas funkcionālās prasības ir attēlotas ar lietošanas gadījumiem?



**rezervēt istabu**

# Lietošanas gadījumu diagrammas elementi

- Asociācijas parāda attiecības starp aktieriem un lietošanas gadījumiem un lietošanas gadījumiem savā starpā
- Asociācijas starp aktieriem netiek attēlotas
- Asociācija starp aktieri un lietošanas gadījumu:
- **<<communicate>>** - attēlo komunikāciju starp aktieri un lietošanas gadījumu. Var būt vienpusēja vai abpusēja; virziens norāda, kas ir šīs komunikācijas iniciators

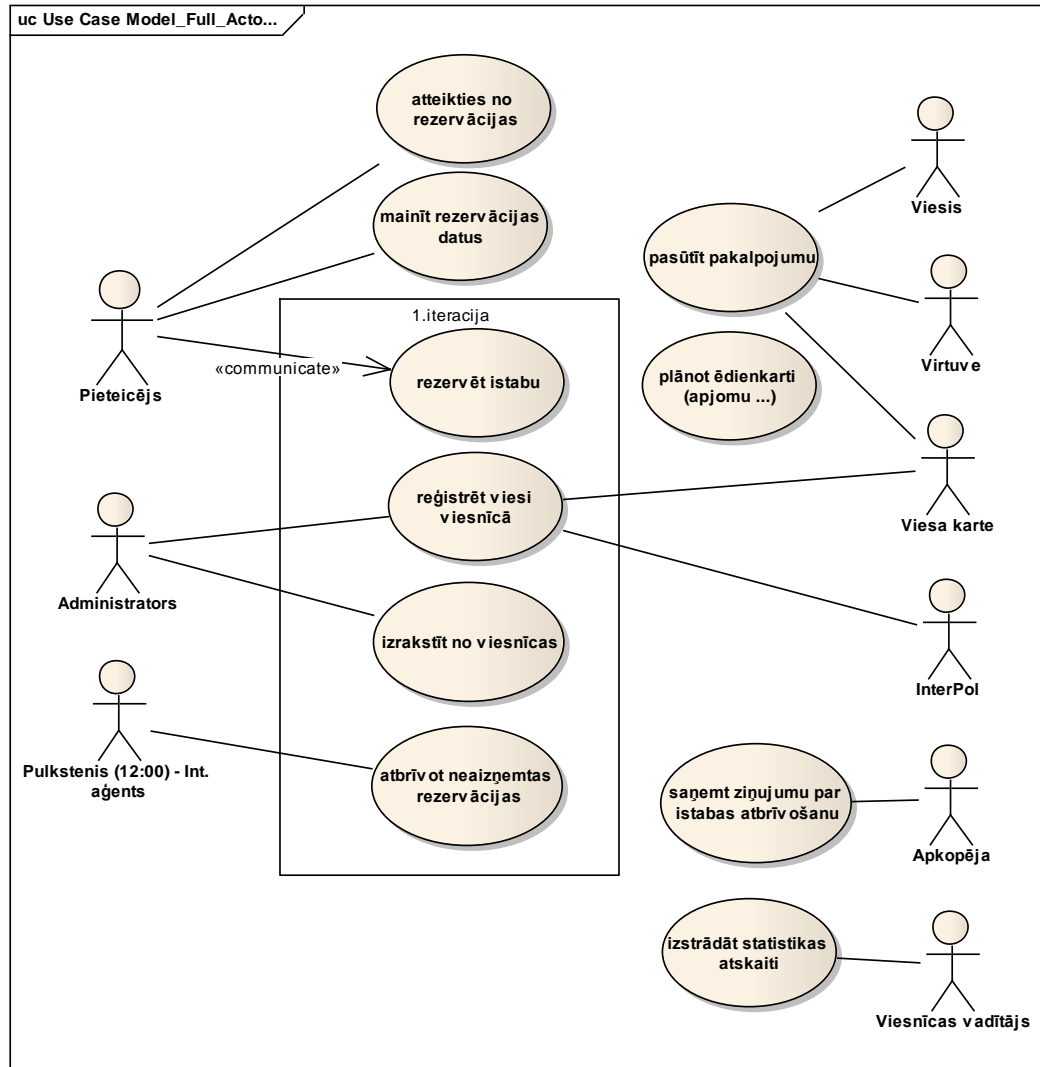


# **Problēmvides apraksts:**

## **Viesnīcas istabu rezervēšana**

- Kad klients grib rezervēt istabu uz noteiktu periodu, viņam viesnīcā (pašam vai ar administratora palīdzību) ir jāaizpilda istabas rezervēšanas anketa ar uzvārdu, vārdu, istabas tipu (single vai double) un istabas aizņemšanas periodu (sākuma un beigas datumu).
- Anketas pieprasījums tiek apstrādāts un klientam tiek paziņots vai rezervēšana ir notikusi veiksmīgi (gadījumā, ja uz šo periodu ir brīvas istabas) vai neveiksmīgi (gadījumā, ja uz šo periodu nav brīvo istabu).
- Kad klients atbrauc uz viesnīcu, administrators pārbauda rezervēšanas informāciju, ieraksta klientu attiecīgajā istabā.
- Kad klients aizbrauc no viesnīcas, administrators atbrīvo istabu turpmākai rezervēšanai.
- Katru dienu plkst. 12:00 administrators pārbauda vai katra rezervēta istaba ir aizņemta. Gadījumā, ja rezervēšanas pieprasījums nav izpildīts, attiecīga istaba ir atbrīvota priekš turpmākas rezervēšanas.

# Sākotnējais sistēmas potenciālās funkcionēšanas uzmetums





# Lietošanas gadījumu īss apraksts

<b>Lietošanas gadījums</b>	Lietošanas gadījuma nosaukums
<b>Aktieris</b>	Aktieru saraksts, kas mijiedarbojas ar sistēmu
<b>Mērķis</b>	Kas aktierim ir jāsasniedz, izpildot lietošanas gadījumu, konkrēts labums, ko aktieris gūt no sistēmas lietošanas
<b>Īss apraksts</b>	Dažos teikumos aprakstīt lietošanas gadījuma izpildi
<b>Tips</b>	Lietošanas gadījuma tips – 1. Galvenais / otrās kārtas / papildus; 2. Ideālais / reālais
<b>Atsauces</b>	Saistīto lietošanas gadījumu nosaukumi, kuriem jāizpildās pirms vai pēc aprakstāma.
<b>Darbības iesākšana</b>	Apraksts, kā tiek palaists lietošanas gadījums, ja tas notiek specifiski
<b>Darbības beigšana</b>	Apraksts, kādā veidā lietošanas gadījums beidz savu darbību, ja tas notiek specifiski

# Piemērs – Viesnīca

Lietošanas gadījums	<b>Rezervēt istabu</b>
Aktieris	Pieteicējs
Mērķis	Rezervēt attiecīga tipa istabu uz noteiktu periodu
Īss apraksts	Pieteicējs pieteicās rezervēt attiecīga tipa istabu uz noteiktu periodu. Ja brīva istaba ir, tā ir rezervēta pieteicējam. Ja attiecīgas brīvas istabas nav, pieteicējs saņem atteikumu.
Tips	Galvenais
Atsauces	nav

Lietošanas gadījums	<b>Izrakstīt no viesnīcas</b>
Aktieris	Administrators, Viesa karte
Mērķis	Pārbaudīt vai ir veiktas visas samaksas un izrakstīt viesi no dzīvošanas.
Īss apraksts	...
Tips	Galvenais
Atsauces	nav

Oksana Nikiforova

Lietošanas gadījums	<b>Reģistrēt viesi viesnīcā</b>
Aktieris	Administrators, Viesa karte
Mērķis	Reģistrēt viesi dzīvošanai attiecīgajā istabā uz noteiktu periodu
Īss apraksts	Administrators pieprasa viesim rezervēšanas informāciju un pasi, viesis dod rezervēšanas informāciju un pases datus, administrators pierēģistrē viesi viesnīcā. Ja viesim nav iepriekšējas rezervēšanas, jāveic attiecīgas istabas meklēšana.
Tips	Galvenais
Atsauces	nav

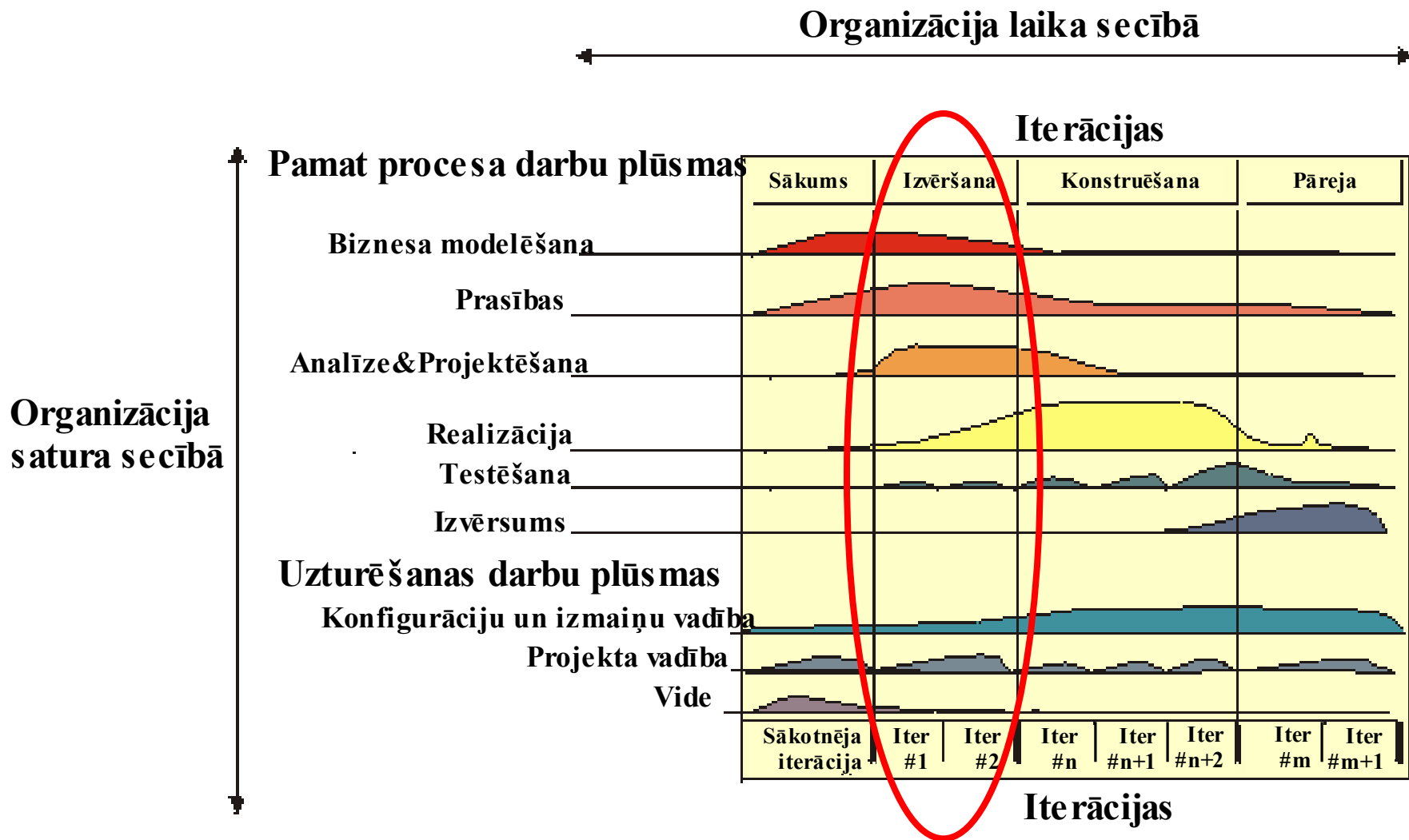
Lietošanas gadījums	<b>Atbrīvot neaizņemtās rezervācijas</b>
Aktieris	Sistēmas plkst.
Mērķis	Atbrīvot no rezervācijas tās istabas, kurās nav reģistrēts attiecīgais viesis.
Īss apraksts	...
Tips	Galvenais
Atsauces	Nav

“Objektorientētas programmēšanas praktikums (studiju projekts)”

# Kopsavilkums

- Sākuma fāze ir pirmā RUP fāze, kurā izstrādātāji iegūst vispārēju priekšstatu par projektu, tādā apjomā lai varētu ar pietiekamu pārliecību apgalvot ka projekts izdosies:
  - Kritiskās funkcionēšanas apjoms
  - Izmantotas tehnoloģijas un rīki
  - Izmaksas
- RUP piedāvā plašu aktivitāšu klāstu. Lai neapjuku tajā, ir jākoncentrējas uz sākuma fāzes mērķiem, un jāizvēlas noteikta, katram specifiskam uzdevumam un komandai pielāgota apakškopa ar aktivitātēm šo mērķu sasniegšanai.

# RUP izvēršanas fāze

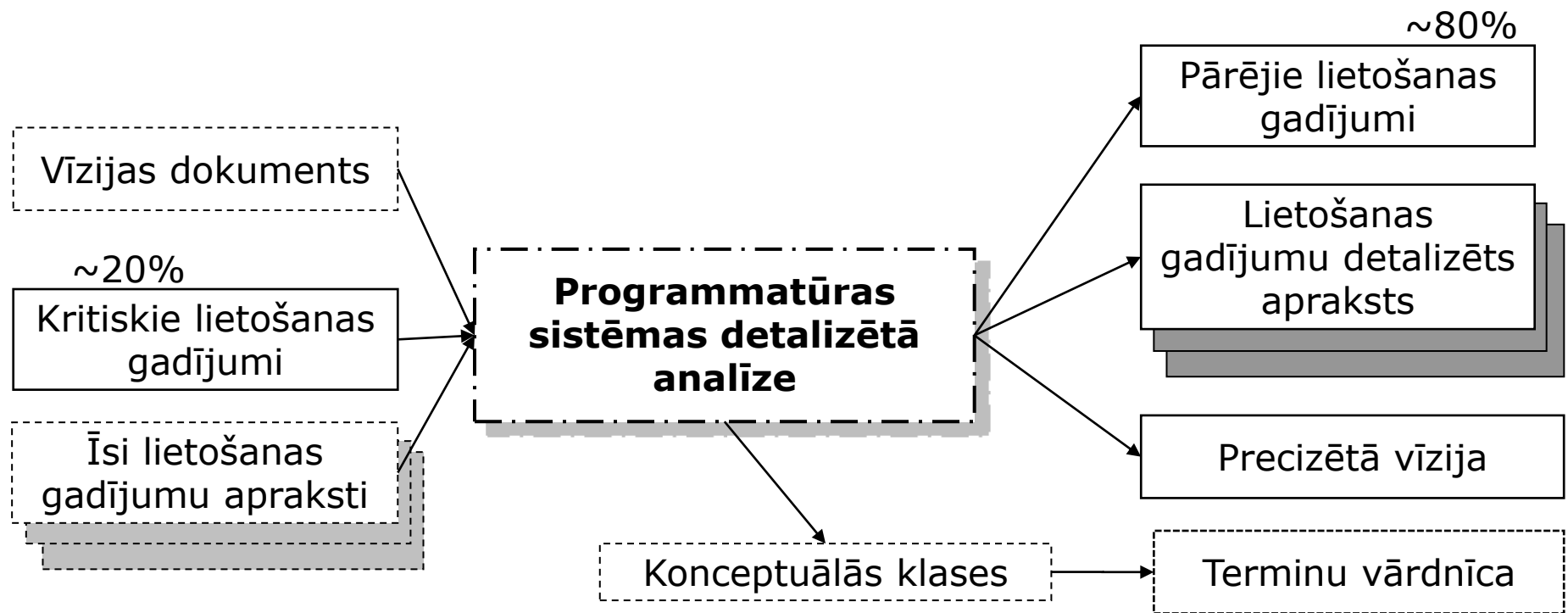


# Izvēšanas fāzes mērķi

- Vel labāk saprast prasības.
- Projektēt, realizēt un pārbaudīt arhitektūru.
- Aizsargāties no būtiskākajiem riskiem un precizēt izmaksas un termiņus.
- Uzlabot izstrādes procesu un vidi (rīkus).

# 1. mērķis – vel labāk saprast prasības

- Izvēršanas fāzē tiek analizēta lielākā lietošanas gadījumu daļa
- Līdz ar lietošanas gadījumu analīzi tiek precizēta vīzija un veidota vārdnīca



# Lietošanas gadījumu detalizēts apraksts

- Lietošanas gadījumu realizācijas scenāriji (apraksta lietošanas gadījuma realizācijas ceļu mērķa sasniegšanai):
  - Lietošanas gadījuma notikumu pamat plūsmu; tipiskā notikumu plūsma – detalizēts lietotāja dialogs ar sistēmu; atbilst sistēmas pamata darbībai, parasti tas ir “veiksmīgs” scenārijs.
  - Novirzes no pamat plūsmas – svarīgas alternatīvas un izņēmumi; atbilst retāk lietotām darbībām un izņēmumiem.
  - Kļūdu plūsmas - apraksta notikumu secību, ja kādā no soliem noticis tehnisks pārtraukums un ir nepieciešams veikt specifiskās darbības sistēmas stāvokļa atjaunošanai.

# Scenāriju noformēšana

1. Tabula tipisko notikumu plūsmas aprakstam

Aktieru darbības	Sistēmas reakcija
Darbības, kuras veic aktieris lietošanas gadījuma izpildes laikā	Sistēmas reakcija uz aktiera darbībām

2. Numurēti alternatīvie notikumi
3. Numurēti kļūdu notikumi



# Lietošanas gadījuma “Rezervēt istabu” apraksts

Lietošanas gadījuma nosaukums: **Rezervēt istabu**

Aktieri: Pieteicējs

Mērķis: Rezervēt attiecīga tipa istabu uz noteiktu periodu

Ts apraksts: Pieteicējs pieteicās rezervēt attiecīga tipa istabu uz noteiktu periodu. Ja brīva istaba ir, tā ir rezervēta pieteicējam. Ja attiecīgas brīvas istabas nav, pieteicējs saņem atteikumu.

Tips: Galvenais (jeb pamatlietoījums)

Atsauces: nav

Tipiskā notikumu secība:

	<u>Aktiera darbība:</u>		<u>Sistēmas reakcija:</u>
1.	Pieteicējs ievieš sistēmā.	2.	Piedāvā pieteicējam izvēlēties istabas tipu un uzdod periodu.
3.	Pieteicējs izvēlās istabastipu, apmēšanas sākuma datumu un izrakstīšanas datumu.	4.	Konstatē, ka ievadītie parametri ir korekti.
		5.	Sameklē brīvo istabu pēc uzdotiem parametriem
		6.	Paziņo pieteicējam, ka ir pieejama brīva istaba
		7.	Pieprasa pieteicējam ievadīt informāciju par sevi ( <i>uzvārds, vārds, e-pasts, bankas rekvizīti</i> )
8.	Pieteicējs ievada informāciju	9.	Konstatē, ka ievadīta informācija ir korekta
		10.	Pievieno pieteicēja info pie attiecīgas istabas (sameklētai brīvai istabai ir atzīmēts, ka tā ir rezervēta uz noteiktu periodu noteiktam pieteicējam)
		11.	Paziņo pieteicējam, ka istabas rezervēšana ir notikusi veiksmīgi (aizsūtīts e-pasts, piedāvāts izdrukāt rez. Informāciju, saglabāt failā, sms...)
12.	Pieteicējs veic rez. Info saglabāšanu		
13.	Pieteicējs iziet no sistēmas		

Alternatīva notikumu gaita:

3a – neizvēlās → pieteicējs iziet no sistēmas (13.solis)

4a – sistēma konstatē, ka ievadītie parametri ir nekorekti → ...

5a – brīvas istabas nav

8a – neievada → pieteicējs iziet no sistēmas

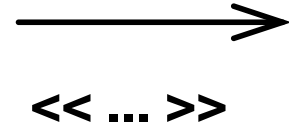
9a – ievadītie parametri ir nekorekti → ..

Kļūdu apstrāde:

5-10e – sistēmai ir jārūpējas par brīvas istabastemporālo rezervēšanu

# Lietošanas gadījumu diagrammas elementi

- Asociācijas parāda attiecības starp aktieriem un lietošanas gadījumiem un lietošanas gadījumiem savā starpā
- Asociācijas starp aktieriem netiek attēlotas
- Asociācija starp lietošanas gadījumiem:
- **<<include>>** - ja daži lietošanas gadījumi iekļauj vienādi funkcionējošus darbības fragmentus, tos iedala atsevišķā lietošanas gadījumā un izmanto iekļaušanas asociāciju; virziens ir no bāzes elementa uz iekļaujamo
- **<<extend>>** - tiek izmantotas, lai attēlotu papildus režīmus, kas tiek palaisti noteiktos apstākļos; alternatīvas plūsmas, kas tiek palaistas pēc aktiera izvēles; virziens ir no paplašinājuma uz bāzes elementu



# <<extend>> asociācijas veidošanas piemērs

Lietošanas gadījuma nosaukums: Reģistrēt viesi

Aktieri: Administrators, Viesa karte

Mērķis: reģistrēt viesi dzīvošanai attiecīgajā istabā uz noteiktu periodu

Īss apraksts: Administrators pieprasa viesim rezervēšanas informāciju un pasi, viesis dod rezervēšanas informāciju un pasas datus, administrators pierēģistrē viesi viesnīcā. Ja viesim nav iepriekšējas rezervēšanas, jāveic attiecīgas istabas meklēšana.

Tips: Galvenais (jeb pamatlietoījums)

Atsauces: lietošanas gadījums „reģistrēt viesi, kam nav iepriekšēja rezervācija” paplašina šī lietošanas gadījuma darbību.

Tipiskā notikumu secība:

	Aktiera darbība:		Sistēmas reakcija:
1.	Administrators ieiet viesnīcas datorsistēmā, lai reģistrētu viesi dzīvošanai	2.	Sistēma prasa ievadīt rezervācijas numuru
3.	Administrators ievada rezervācijas numuru	4.	....

Alternatīva notikumu gaita:

3a – viesim nav iepriekš rezervētas istabasneizvēlās → izpildīt lietošanas gadījumu „reģistrēt viesi, kam nav iepriekšēja rezervācija”

Kļūdu apstrāde:



<<extend>> asociāciju veido, ja tiek konstatēts, ka kāda soļa izpildes alternatīva pati pa sevi ir sarežģītā un tas aprakstā būs iespējamās savas novirzes.

# <<include>> asociācijas veidošanas piemērs

Lietošanas gadījuma nosaukums: Rezervēt istabu

	Aktiera darbība:		Sistēmas reakcija:
1.	Pieteicējs ieiet viesnīcas sistēmā.	2.	Piedāvā pieteicējam izvēlēties istabas tipu un uzdod periodu.
3.	Pieteicējs izvēlas istabas tipu, apmēšanas sākuma datumu un izrakstīšanas datumu.	4.	Konstatē, ka ievadītie parametri ir korekti.
		5.	Sameklē brīvu istabu pēc uzdotiem parametriem
		6.	Paziņo pieteicējam, ka ir pieejama brīva istaba
		7.	Pieprasa pieteicējam ievadīt informāciju par sevi (uzvards, vārds, e-pasts, bankas rekvizīti)
8.	Pieteicējs ievada informāciju	9.	Konstatē, ka ievadīta informācija ir korekta
		10.	Pievieno pieteicēja info pie attiecīgas istabas (sameklētai brīvai istabai ir atzīmēts, ka tā ir rezervēta uz noteiktu periodu noteiktam pieteicējam)

<<include>> asociāciju veido, ja tiek konstatēts, ka kāda soļu secība atkārtojas divos vai vairāk lietošanas gadījumu aprakstos.

Lietošanas gadījuma nosaukums: Reģistrēt viesi, kam nav iepriekšējas rezervācijas

Aktieri: Administrators

Mērķis:

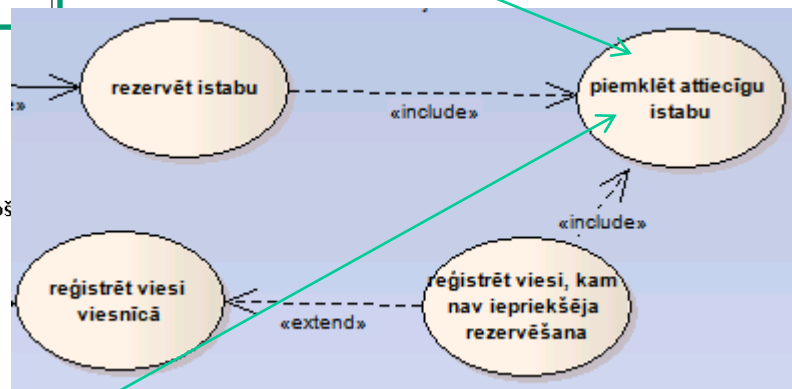
Īss apraksts: Administrators piemeklē attiecīga tipa istabu uz noteiktu periodu. Ja brīva istaba ir, tā ir reģistrēta dzīvošā attiecīgas brīvas istabas nav, viesis saņem atteikumu.

Tips: Galvenais (jeb pamatlietoījums)

Atsauces: paplašinājuma lietošanas gadījums lietošanas gadījumam „reģistrēt viesi”

Tipiskā notikumu secība:

	Aktiera darbība:		Sistēmas reakcija:
1.	Administrators atrodas viesnīcas sistēmā, turpinot lietošanas gadījuma „reģistrēt viesi” realizāciju.	2.	Piedāvā izvēlēties istabas tipu un uzdod periodu.
3.	Administrators izvēlas istabas tipu, apmēšanas sākuma datumu un izrakstīšanas datumu.	4.	Konstatē, ka ievadītie parametri ir korekti.
		5.	Sameklē brīvu istabu pēc uzdotiem parametriem
		6.	Paziņo administratoram, ka ir pieejama brīva istaba
		7.	Pieprasa administratoram vai reģistrēt viesi dzīvošanai viesnīcā.
8.	Administrators paziņo, ka vēlas reģistrēt dzīvošanai	9.	Turpina realizēt lietošanas gadījumu reģistrēt viesi.



# Realizācijai atlasīto lietošanas gadījumu un to aktieru diagramma

