

საქართველოს ტექნიკური უნივერსიტეტი

ხელნაწერის უფლებით

ბექარ ურუშაძე

**პროგრამული სისტემების მენეჯმენტი UML /Agile
ბაზირებული მეთოდებით**

დოქტორის აკადემიური ხარისხის მოსაპოვებლად

წარდგენილი დისერტაციის

ავტორეფერატი

თბილისი

2014 წელი

სამუშაო შესრულებულია საქართველოს ტექნიკური უნივერსიტეტის ინფორმატიკისა და მართვის სისტემების ფაკულტეტის „მართვის ავტომატიზებული სისტემების (პროგრამული ინჟინერია)“ დეპარტამენტში

სამეცნიერო ხელმძღვანელები:

სრული პროფ. ვ. დიდმანიძე,

სრული პროფ. გ. სურგულაძე

რეცენზენტები:

სრული პროფ. თ. სუხიაშვილი

სრული პროფ. ა. ცინცაძე

დაცვა შედგება 2014 წლის ” 19 ” ივლისს, 14.00 საათზე

საქართველოს ტექნიკური უნივერსიტეტის - „ინფორმატიკისა და მართვის სისტემების“ ფაკულტეტის სადისერტაციო საბჭოს კოლეგიის სხდომაზე, კორპუსი მე-6, აუდიტორია 207-ა
მისამართი: 0175, თბილისი, კოსტავას 77.

დისერტაციის გაცნობა შეიძლება სტუ-ს ბიბლიოთეკაში,

ხოლო ავტორეფერატისა - სტუ-ს ვებგვერდზე

სადისერტაციო საბჭოს

მდივანი: სრული პროფ. თინათინ კაიშაური

ნაშრომის ზოგადი დახასიათება

თემის აქტუალურობა. თანამედროვე ორგანიზაციული მართვის ავტომატიზებული სისტემების პროგრამული უზრუნველყოფის შექმნის, დანერგვის და ექსპლუატაციის საკითხების გადაწყვეტა, ახალი ინფორმაციული ტექნოლოგიების ბაზაზე, უსაფრთხო მეთოდოლოგიებისა და საერთაშორისო სტანდარტების საფუძველზე, მეტად აქტუალურია.

გამოყენებითი საინფორმაციო სისტემების პროგრამული ინდუსტრია გვთავაზობს კორპორაციული მენეჯმენტის ბიზნეს-პროცესების მართვის არაერთ მაღალხარისხოვან და საერთაშორისო სტანდარტებით აღიარებულ პროდუქტებს, როგორებიცაა, მაგალითად, ERP, CRM, 1C:Предприятие და სხვა, რომლებიც ახორციელებს საწარმოო პროცესების სრულყოფილ ავტომატიზაციას. პროგრამული ინჟინერიის თვალსაზრისით, ასეთი პაკეტები დამუშავებულია ობიექტ-ორიენტირებული, პროცეს-ორიენტირებული და სერვის-ორიენტირებული არქიტექტურების საფუძველზე, წარმოადგენს რთულ და დიდ სისტემებს და ობიექტებზე დანერგვის დროს თხოულობს გარკვეულ საადაპტაციო სამუშაოების ჩატარებას, მოხმარებელთა სერიოზულ ტრენინგს. ამასთანავე, ამგვარი სისტემების ფასი საკმაოდ მაღალია და მათი შექმნა მხოლოდ ფინანსურად მდიდარ, სახელმწიფო ან კერძო სტრუქტურის ორგანიზაციებს შეუძლია.

ინდივიდუალური პროგრამული უზრუნველყოფის შექმნა კონკრეტული ორგანიზაციისთვის (მათ შორის მცირე და საშუალო ბიზნესისთვის) ხშირად უფრო ხელმისაწვდომია, ვინაიდან აქ მათ პრიორიტეტული ამოცანების ამორჩევის და მათ ორგანიზაციაზე თავიდანვე მორგებულად დაპროექტების საშუალება ეძლევა. თვითონ მონაწილეობენ მომავალი სისტემის მოთხოვნილებათა განსაზღვრის პროცესში, თვითონ იღებენ ექსპლუატაციაში დამუშავებულ პროგრამებს და თვითონვე განსაზღვრავენ აუცილებელი ცვლილებების ან მომავალი გაფართოების პირობებს და ამოცანებს. პროგრამული სისტემების შექმნის ახალი მეთოდოლოგიები და მხარდამჭერი CASE ტექნოლოგიები დღეს

ფართოდაა გავრცელებული აშშ-ის, ევროპის, რუსეთის და აღმოსავლეთის განვითარებულ ქვეყნებში. მათ შორის უნდა გამოვყოთ UML ენა, რომელიც დიდი პროგრამული პროექტების გადაწყვეტის მენეჯმენტის აუცილებელი მეთოდოლოგიაა, გააჩნია ინსტრუმენტული საშუალებები (Rational Rose, Visual Paradigm, Enterprise Architect, Ms Visio და სხვ.). ამ მიმართულების ფუძემდებლებია ამერიკაში მოღვაწე მეცნიერები გ.ბუჩი, ი.ჯაკობსონი, ჯ.რამბო და სხვა. ეს მეთოდოლოგია ხასიათდება პროგრამული სისტემების დაპროექტების პროცესების სტანდარტიზაციის მაღალი დონით, გრაფიკულ-ანალიზური ინსტრუმენტებით დიაგრამების ასაგებად, რა თქმა უნდა, დროის საკმაოდ ხანგრძლივობით. მეორე, საპირისპირო მიმართულებაა „მოქნილი (სწრაფი)“ მეთოდოლოგია (Agile). იგი უარს ამბობს „დიაგრამების ასაგებად“ დროის ხარჯვაზე და ძირითად აქცენტს დამკვეთზე აკეთებს. მისი პრინციპია: „მთავარია საბოლოო პროგრამული პროდუქტი და არა დიაგრამები, მათი სწრაფად შესრულება და ცვლილებების გათვალისწინება, რომლებიც საბოლოო მომხმარებელზეა გათვლილი“. Agile მეთოდოლოგია, მაგალითად, ექსტრემალური დაპროგრამება, რომლის ავტორები ასევე ამერიკელი მეცნიერები გახლავთ: კ.ბეკი, ე.გამა, რ. ჯეფრი, ჯ.სიუზერლანდი და სხვა. ასეთი სისტემებია: XP (ექსტრემალური პროგრამირება), Scrum, Rational Unified Process (RUP), Dynamic Systems Development Method (DSDM) და სხვა. აქცენტს მაღალორგანიზებულ გუნდური პროგრამირების პრინციპებზე აკეთებენ და გამორიცხავენ „ჭარბი დიაგრამების“ ხაზვის აუცილებლობას.

კომპრომისული მეთოდოლოგია, ანუ „Agile+UML“ არის ასევე აქტუალური მიმართულება პროგრამული სისტემების დეველოპინგში. მისი მიმდევრები, ამერიკელი ჯ.ფლაუერი, კანადელი ს.ამბლერი, გერმანელი ბ.რუმპე და სხვები, იმ აზრზე არიან, რომ, საჭიროა მოქნილი პროგრამირების მეთოდოლოგიას დაემატოს UML-ის ზოგიერთი, მხოლოდ აუცილებელი (და არა ყველა) დიაგრამა, რაც შესაძლებელს ხდის ე.წ. „ჰიბრიდული“ (ოპტიმალური) მეთოდოლოგიის შექმნას.

საქართველოში აღნიშნული სამეცნიერო მიმართულების დამფუძნებლები და მათი საინფორმაციო სისტემებში გამოყენების ტექნოლოგიის მიმდევრები არიან სტუ-ს პროფესორები: გ.გოგიჩაიშვილი, ე.თურქია, გ.სურგულაძე, თ.სუხიაშვილი და სხვა.

კორპორაციული ბიზნეს-პროცესების მენეჯმენტის მექანიზმების მუდმივი სრულყოფით, ITIL (Information Technology Infrastructure Library) მეთოდოლოგიით და სერვის-ორიენტირებული არქიტექტურით (SOA) შესაძლებელი ხდება მთლიანი სისტემის სასიცოცხლო ციკლის გახანგრძლივება, რაც უდავოდ აქტუალური საკითხია ინტეგრაციის და რეინჟინერინგის ამოცანების გადასაწყვეტად.

სამუშაოს მიზანი და ამოცანები. დისერტაციის მიზანია პროგრამული სისტემების მენეჯმენტის პროცესების ხელშემწყობი სისტემის აგების კონცეფციის შემუშავება, მათი სასიცოცხლო ციკლის ეტაპებისა და ფაზების სრულყოფის მიზნით, არსებული და ახალი მეთოდოლოგიების, სტანდარტების და ტექნოლოგიების ინტეგრაციის ფონზე, ინფორმაციული უსაფრთხოების და სერვის-ორიენტირებული არქიტექტურის რეალიზაციის საფუძველზე. დასმული მიზნის მისაღწევად აუცილებელია შემდეგი ძირითადი ამოცანების გადაწყვეტა:

- გამოყენებითი პროგრამული სისტემების მენეჯმენტის ბიზნეს-პროცესების დიაგნოსტიკური ანალიზი პრობლემების გამოვლენით. მათი სასიცოცხლო ციკლის პროცესების ობიექტ-ორიენტირებული მოდელირება UML ტექნოლოგიით. პრობლემების გადაწყვეტის კონცეფციის შემუშავება ახალი ინფორმაციული მეთოდოლოგიების, სტანდარტების და ტექნოლოგიების ბაზაზე;

- IT-სერვისების უწყვეტი სრულყოფის პროცესის სასიცოცხლო ციკლის ანალიზი ბიზნეს-პროცესებზე დამოკიდებულებით, ინტეგრაციის პროცესებში ITIL და Agile მეთოდოლოგიების შეთანხმებული გამოყენება სინერგეტიკის პრინციპების თვალსაზრისით. IT-სერვისების *მხარდამჭერი* და *მიწოდების* ბიზნეს-პროცესების მოდელირება UML-ის Activity-D დიაგრამებით. კერძოდ,

ინციდენტების, პრობლემების, კონფიგურაციების, ცვლილებების და რელიზების მართვის მხარდამჭერი პროცესები, აგრეთვე სერვისების დონის, სიმძლავრეების, უწყვეტობის, წვდომის, ფინანსების და უსაფრთხოების მართვის მიწოდების პროცესები;

- საპრობლემო სფეროს პროგრამული სისტემის მენეჯმენტის ბიზნეს-პროცესების გამოკვლევა პროცეს-ორიენტირებული მოდელებით (BPMN), მონაცემთა განაწილებული ბაზის კონცეპტუალური სქემების ავტომატიზებული დაპროექტება ობიექტ-როლური მოდელით (ORM), არსთა-დამოკიდებულების მოდელით (ERM) და, ბოლოს, შესაბამისი DDL-ფაილებით MsSQL Server ბაზის ავტომატიზებული რეალიზაცია. განაწილებულ სისტემაში IT-სერვისების მიწოდების და ინფორმაციის გაცვლის პროცესის იმიტაციური მოდელირება პეტრის ფერადი ქსელებით (CPN);

- წინა პუნქტში ჩამოყალიბებული საკითხების კონკრეტული გადაწყვეტის შედეგების ასახვა შემდეგი პროგრამული პროექტებისთვის: „ბანკთაშორისი საინფორმაციო სისტემის ინტეგრაციის პროცესების მენეჯმენტი სერვის ორიენტირებული არქიტექტურით“, „პროგრამული სისტემების მენეჯმენტი მულტიმედიალური აპლიკაციების დასაპროექტებლად და ასაგებად“, „შემოსავლების სამსახურში საგადასახადო დავების ბიზნეს-პროცესების მოდელირება UML-ტექნოლოგიით, შესაბამისი პროგრამული ინტერფეისების შემუშავება მასობრივი მომსახურების მეთოდების საფუძველზე;

- თეორიულად შემუშავებული მოდელების და მეთოდების პროგრამული რეალიზაცია MsVisual_Studio.NET Framework_4.0 ინტეგრირებულ გარემოში - C#, XAML, WPF, WF, WCF, MsSQL Server პაკეტების, Enterprise Architect, MsVisio და Natural ORM Architect CASE ინსტრუმენტული საშუალებების გამოყენებით.

კვლევის ობიექტი. საპრობლემო სფეროს პროგრამული სისტემის დამუშავების პროექტის ბიზნეს-პროცესები, მისი სასიცოცხლო ციკლის

ეტაპები, ფაზები და საკონტროლო წერტილები. აგრეთვე პროგრამული სისტემების მენეჯმენტის ადამიანური, ტექნიკური, ფინანსური და დროითი რესურსები.

კვლევის მეთოდები. პროგრამული სისტემების მენეჯმენტის თეორია. ობიექტ-ორიენტირებული, პროცეს-ორიენტირებული და სერვის-ორიენტირებული არქიტექტურის მეთოდები და ინსტრუმენტები. საინფორმაციო სისტემების აგების ITIL მეთოდოლოგია. მოქნილი (Agile) და უნიფიცირებული მოდელირების (UML) მეთოდოლოგიები და მათი CASE საშუალებები. საინფორმაციო სისტემების მონაცემთა განაწილებული ბაზების თეორია. ობიექტ-როლური მოდელირების მეთოდი. პეტრის ფერადი ქსელების იმიტაციური მოდელები. მასობრივი მომსახურების მოდელები და მეთოდები. ბიზნეს-პროცესების დაპროგრამების ახალი, ჰიბრიდული ტექნოლოგიები (WPF, WF, WCF).

მეცნიერული სიახლე. ნაშრომში წარმოდგენილია შემდეგი სიახლეები:

1. პირველად დისერტაციაში შემოთავაზებულ იქნა გამოყენებითი პროგრამული სისტემების მენეჯმენტის ბიზნეს-პროცესების სრულყოფა უნიფიცირებული (UML) და მოქნილი (Agile) მოდელირების მეთოდოლოგიების კომპრომისული გამოყენებით, ინფორმაციული ტექნოლოგიების ინფრასტრუქტურის ბიბლიოთეკის (ITIL) და ინფორმაციული უსაფრთხოების საერთაშორისო სტანდარტების გათვალისწინებით, სერვის-ორიენტირებული არქიტექტურით (SOA);

2. IT-სერვისების უწყვეტი სრულყოფის პროცესის სასიცოცხლო ციკლის ანალიზის საფუძველზე შემუშავდა სერვისების მიზეზ-შედეგობრივი დამოკიდებულების მოდელი ბიზნეს-პროცესებზე, განხორციელდა IT-სერვისების მხარდამჭერი და მიწოდების ბიზნეს-პროცესების მოდელირება UML-ის Activity-D დიაგრამებით;

3. პირველად ამ ნაშრომში მოხდა სხვადასხვა საპრობლემო სფეროს პროგრამული სისტემების პროექტების მენეჯმენტის ბიზნეს-პროცესების ანალიზის საფუძველზე სასიცოცხლო ციკლის გარკვეული ეტაპების

ამოცანების განზოგადოება და მათი ავტომატიზებულ პროცედურებად ჩამოყალიბება. კერძოდ, ეს ამოცანებია: ობიექტ-როლური მოდელირების და ავტომატიზებული დაპროექტების პრინციპების გამოყენებით მონაცემთა ბაზების კონცეპტუალური, ლოგიკური და ფიზიკური სტრუქტურების აგება, პეტრის ფერადი ქსელებით იმიტაციური მოდელის აგება დინამიკური პროცესების ანალიზის მიზნით, და ბოლოს, სერვის-ორიენტირებული არქიტექტურით სისტემის პროგრამული რეალიზაცია ახალი ინფორმაციული ტექნოლოგიების (WPF, WF, WCF) კომპლექსური გამოყენებით;

შედეგების გამოყენების სფერო. დისერტაციის შედეგებს აქვს პრაქტიკული ღირებულება, ვინაიდან ის შეიძლება გამოყენებულ იქნას ნებისმიერი დარგის ორგანიზაციული მენეჯმენტის ბიზნეს-პროცესების მოდელირების, დაპროექტების და პროგრამული რეალიზაციის ამოცანების გადასაწყვეტად ახალი ინფორმაციული ტექნოლოგიებით.

ნაშრომის აპრობაცია: დისერტაციის ძირითადი შინაარსი მოხსენებული იყო ინფორმატიკისა და მართვის სისტემების ფაკულტეტის „მართვის ავტომატიზებული სისტემების (პროგრამული ინჟინერია)“ კოლეგიის სამეცნიერო სემინარების სხდომებზე, ასევე VI საერთაშორისო სამეცნიერო-პრაქტიკულ კონფერენციაზე „ინტერნეტი და საზოგადოება“, INSO-2013, 6-7 ივნისი, აკ.წერეთლის სახ. უნივერსიტეტი, ქუთაისი.

პუბლიკაციები: დისერტაციის ძირითადი შედეგები გამოქვეყნებულია 8 სამეცნიერო ნაშრომში, რომელთა ჩამონათვალიც მოყვანილია დისერტაციის ბოლოს.

ნაშრომის მოცულობა და სტრუქტურა: დისერტაციის სრული მოცულობა შეადგენს 167 ნაბეჭდ გვერდს; შედგება რეზიუმეს (ორ ენაზე), სარჩევის, შესავლის, სამი თავის და დასკვნისგან. ახლავს 81 ნახაზი, 2 ცხრილი და 68 გამოყენებული ლიტერატურის სია.

დისერტაციაში წარმოდგენილია დღეისათვის არსებული მეთოდოლოგიები, საერთაშორისო სტანდარტები, მეთოდები და ინსტრუმენტული CASE-საშუალებები, რომელთა ბაზაზეც მუშავდება გამოყენებითი სფეროს პროგრამული აპლიკაციები.

განხილულია პროგრამული ტექნოლოგიების განვითარების ისტორია არსებული პარადიგმების საფუძველზე. განსაკუთრებით ექცევა ყურადღება პროგრამული სისტემების დამუშავების გუნდური მეთოდების გამოყენების ტექნოლოგიებს, პროგრამული აპლიკაციების აგებას სერვის-ორიენტირებული არქიტექტურით (SOA). გამოყენებულია პროცეს-ორიენტირებული (BPMN) და ობიექტ-ორიენტირებული (UML) მოდელირების ტექნოლოგიები, აგრეთვე მქნილი (Agile) მეთოდები და IT-ინფრასტრუქტურის ბიბლიოთეკა (ITIT).

გადასვლა საინჟინრო მეთოდოლოგიებზე და ჩანჩქერულ მოდელზე ნამდვილად იყო წინგადადგმული ნაბიჯი. მან შემოიტანა განსაზღვრული მოწესრიგება და ორგანიზებულობა დამუშავების პროცესში, ასევე აღმოფხვრა Code-and-Fix მიდგომის მრავალი პრობლემა. საჭიროა აღინიშნოს მხოლოდ პრინციპული მომენტი, რომ ყოველი პროგრამული პროდუქტი პრაქტიკულად არის უნიკალური და ახალი. მისი დამუშავება მიმდინარეობს მაღალი განუსაზღვრელობის პირობებში, და მცდელობა იმისა, რომ წინასწარ იქნას გათვალისწინებული ყველა ფაქტორი პროექტის დასაწყისში (დეტალური მოთხოვნების შედგენა, სისტემის დიზაინის სრული დამუშავება და სხვა) წინასწარაა განწირული წარუმატებლობისთვის. ჩანჩქერული მოდელის ნაკლოვანებები არა მხოლოდ ართულებს პროგრამული სისტემის დამუშავებას, არამედ იგი ავიწროვებს ადამიანურ ურთიერთობებს. მაგალითად, პროექტის დასაწყისში შედგენილი და დამტკიცებული დეტალური მოთხოვნები ხშირად ხდება დამკვეთის და მიმწოდებლის უთანხმოების მიზეზი რეალიზაციის სტადიაზე, როცა საჭიროა აუცილებელი ცვლილებების განხორციელება. რა თქმა უნდა, არსებობს პროექტებიც, რომლისთვისაც ჩანჩქერული სასიცოცხლო ციკლი მისაღებია, მაგრამ უმრავლესი პროექტისთვის ზემოაღწერილი პრობლემები აქტუალურია და მათ არგათვალისწინებას მივყავართ პროექტების ჩავარდნამდე.

შეიძლება ითქვას, რომ საინჟინრო მეთოდოლოგიებმა ვერ შეძლო პროგრამული უზრუნველყოფის კრიზისის გადაწყვეტა. პროექტების წარუმატებლობის მრავალმა შემთხვევამ, რომლებიც საინჟინრო მეთოდოლოგიების საფუძველზე ხორციელდებოდა, დღის წესრიგში დააყენა ალტერნატიული მიდგომების ძიების საკითხი, რაც შემდგომ განხორციელდა და ხორციელდება კვლავ. განსაკუთრებით სწრაფად

განვითარდა ეს პროცესი 90-ანი წლების მეორე ნახევარში, როცა დიდი პოპულარობა მოიპოვა ე.წ. მსუბუქმა (ან მსუბუქი წონის) მეთოდებმა, როგორებიცაა Scrum, DSDM, Crystal და სხვა. ეს მიდგომები მსგავსია იმით, რომ ალტერნატივის სახით მძიმე და ზედმეტად ფორმალიზებული საინჟინრო მეთოდოლოგიებისგან განსხვავებით, ისინი იძლევა ადაპტურ, იტერაციულ და ადამიანზე ორიენტირებულ მიდგომებს. ამ მიმართულებას 2001 წ. დაერქვა “Agile” (მოქნილი, სწრაფი).

მოქნილი მეთოდები - ესაა პროგრამული ინდუსტრიის თანამედროვე პასუხი თუ როგორ უნდა შესრულდეს პროექტები, რათა ისინი წარმატებით დასრულდეს მაღალი ალბათობით და მოუტანოს კმაყოფილება ყველა დაინტერესებულ მხარეს - და პირველ რიგში, დამკვეთს და პროექტის შემსრულებელ გუნდს.

მოქნილი დამუშავების მანიფესტი და პრინციპები მოიცავს მაღალი დონის კონცეფციებს იმის შესახებ თუ როგორ უნდა განხორციელდეს პროგრამული უზრუნველყოფის დამუშავების პროცესი, რათა წარმატებით დასრულდეს პროექტი, შეიქმნას მუშა გუნდები, რომლებშიც სასიამოვნო და საინტერესო იქნება მუშაობა. ეს დოკუმენტები აღწერს თუ რა უნდა გაკეთდეს ამისთვის, მაგრამ არაფერს ამბობს თუ როგორ უნდა გაკეთდეს.

კორპორაციული ბიზნეს-პროცესების მენეჯმენტის მექანიზმების მუდმივი სრულყოფით შესაძლებელი ხდება მთლიანი სისტემის სასიცოცხლო ციკლის გახანგრძლივება, რაც უდავოდ აქტუალური საკითხია ინტეგრაციის და რეინჟინერინგის ამოცანების გადასაწყვეტად. კორპორაციული ობიექტების მართვის საინფორმაციო სისტემის შექმნა მოიცავს ბიზნეს-პროცესების მოთხოვნილებათა განსაზღვრის, დიაგნოსტიკური ანალიზის, ბიზნეს-პროგრამების დაგეგმვის და პროექტირების, IT-სერვისების დადგენის, მათი განხორციელების ორგანიზების, ფაქტ-შედეგების აღრიცხვის, რისკების ანალიზის და შეფასების, ინფორმაციული უსაფრთხოების უზრუნველყოფის, ობიექტზე ეფექტური ზემოქმედების მმართველი გადაწყვეტილების მიღების პროცესების ხელშემწყობი მექანიზმების შემუშავებას და მათ კომპიუტერულ რეალიზაციას, მენეჯმენტის საინფორმაციო სისტემების აგების თანამედროვე ITIL კონცეფციის საფუძველზე.

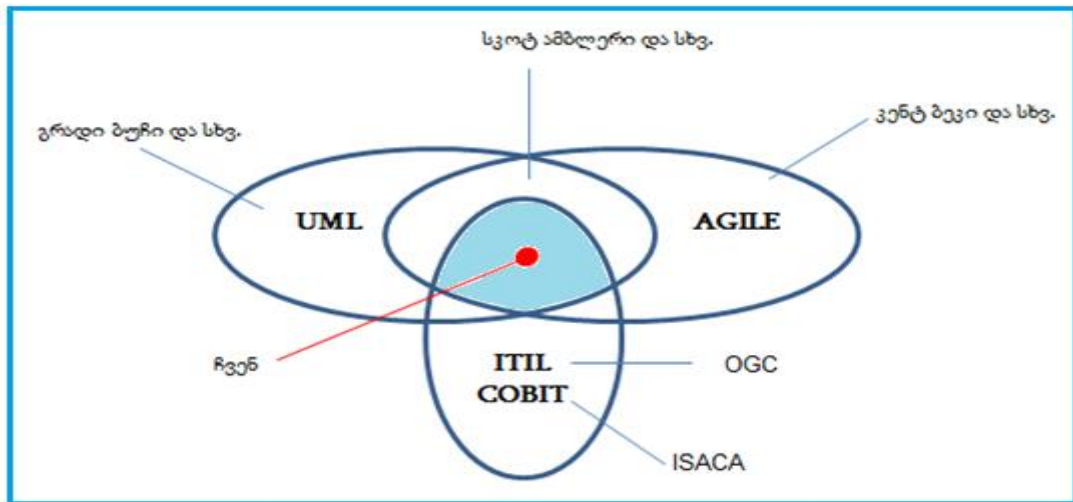
ITILv3-ის ძირითადი დანიშნულებაა IT-სერვისების ხარისხიანი წარმოდგენა და მხარდაჭერა ბიზნესის მოთხოვნების შესაბამისად. სერვისების მართვის პროცესების დაჯგუფებასთან ერთად სასიცოცხლო ციკლის ცალკეული პერიოდების მიხედვით, იგი გვთავაზობს IT-სერვისზე ვისაუბროთ ბიზნესისთვის დამატებითი ფასეულობის შეთავაზების კონტექსტში, რაც სრულიად ახალი მიდგომაა. IT-სამსახური ანალიზებს

ბიზნესის მიზნებს და ამოცანებს და, აქედან გამომდინარე, გვთავაზობს სერვისებს, რომლებიც ნამდვილად სჭირდება ბიზნესს რეალურ დროში.

სრულფასოვანი და საიმედო, მოქნილი გამოყენებითი პროგრამული უზრუნველყოფის (Applied Software) სრულყოფილად და სწრაფად დაპროექტება, რეალიზაცია, დანერგვა და შემდგომი თანხლება, სისტემის დამკვეთ ორგანიზაციაში, Software Engineering-ის მეტად მნიშვნელოვანი ამოცანაა და მისი ეფექტურად გადაწყვეტა ბევრადაა დამოკიდებული როგორც საპროექტო-დეველოპმენტის გუნდის შემადგენლობასა და გამოცდილებაზე, ასევე IT-ინფრასტრუქტურასა და CASE-ინსტრუმენტებზე, კერძოდ UML და Agile მეთოდოლოგიების გამოყენებაზე.

ჩვენი ნაშრომის მიზანია პროგრამული სისტემების მენეჯმენტის პროცესების ხელშეწყობი სისტემის აგების კონცეფციის შემუშავება, მათი სასიცოცხლო ციკლის ეტაპებისა და ფაზების სრულყოფის მიზნით, არსებული და ახალი მეთოდოლოგიების, სტანდარტების და ტექნოლოგიების ინტეგრაციის ფონზე, ინფორმაციული უსაფრთხოებისა და სერვისებზე ორიენტირებული არქიტექტურის რეალიზაციის საფუძველზე.

მე-2 ნახაზზე ნაჩვენებია აღნიშნული მიზნის მისაღწევად გამოყენებულ მეთოდოლოგიათა თანაკვეთა, ანუ ჩვენი მიდგომა ითვალისწინებს მათი პრინციპების კომპრომისულ გამოყენებას.



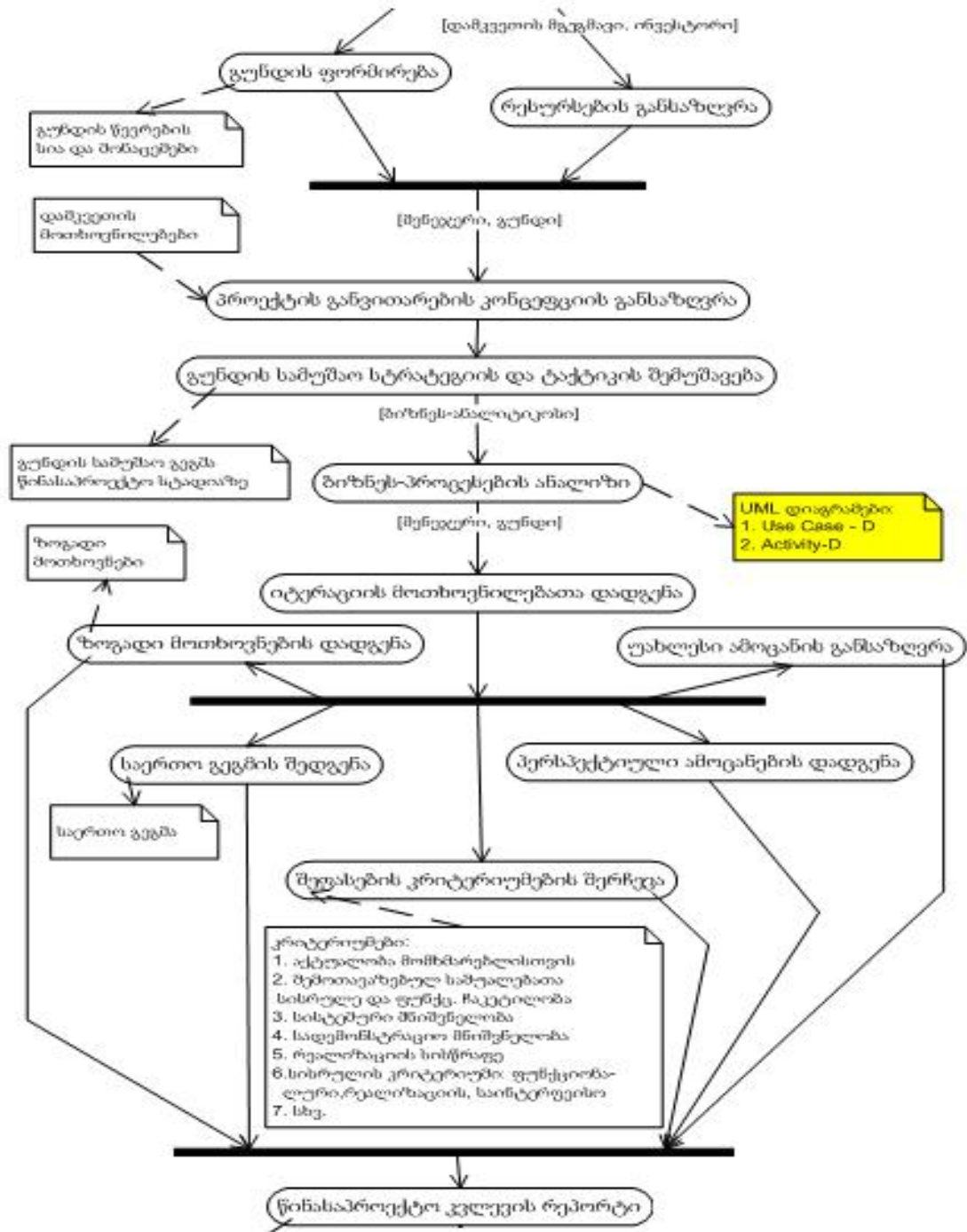
ნახ.2. პროგრამული სისტემების და IT-სერვისების მენეჯმენტის მეთოდოლოგიათა კვეთა

- OGC (Office of Government Commerce) - სახელმწიფო სავაჭრო პალატა (დიდი ბრიტანეთი),
- ISACA (Information Systems Audit and Control Association) - საინფორმაციო სისტემების აუდიტის და კონტროლის ასოციაცია (აშშ)

ამგვარად, IT-მენეჯმენტის ობიექტებია: ინფრასტრუქტურა, აპლიკაცია (დანართი), საინფორმაციო სისტემების სერვისების ორგანიზაციული სტრუქტურა და IT-პროექტები. დასმულია ამოცანა პროგრამული სისტემების ვერსიების დამუშავების მიზნით UML/Agile/ITIL

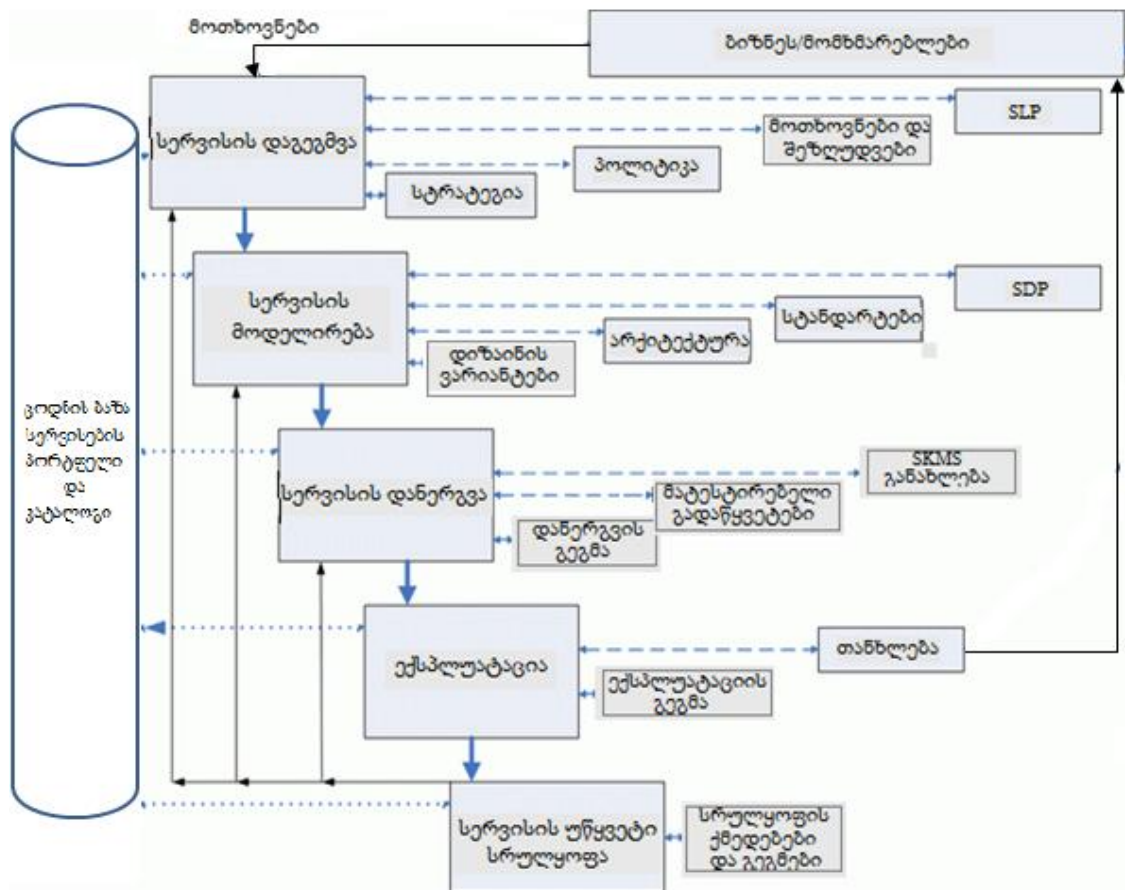
მეთოდოლოგიების კომპრომისული გამოყენების საფუძველზე, IT-სერვისებზე ორიენტირებული არქიტექტურის პროექტების ასაგებად.

მეორე თავში გადმოცემულია საინფორმაციო სისტემების პროგრამული უზრუნველყოფის ავტომატიზებული დაპროექტების და აგების საკითხები ახალი ინფორმაციული ტექნოლოგიებით. განიხილება პროგრამული სისტემების, კერძოდ IT-სერვისების სასიცოცხლო ციკლის მენეჯმენტის ბიზნეს-პროცესების მოდელირება უნიფიცირებული (UML) და მოქნილი (Agile) მეთოდოლოგიების კომპრომისულ საფუძველზე (ნახ.3).



ნახ.3. აქტიურობათა დიაგრამის ფრაგმენტი

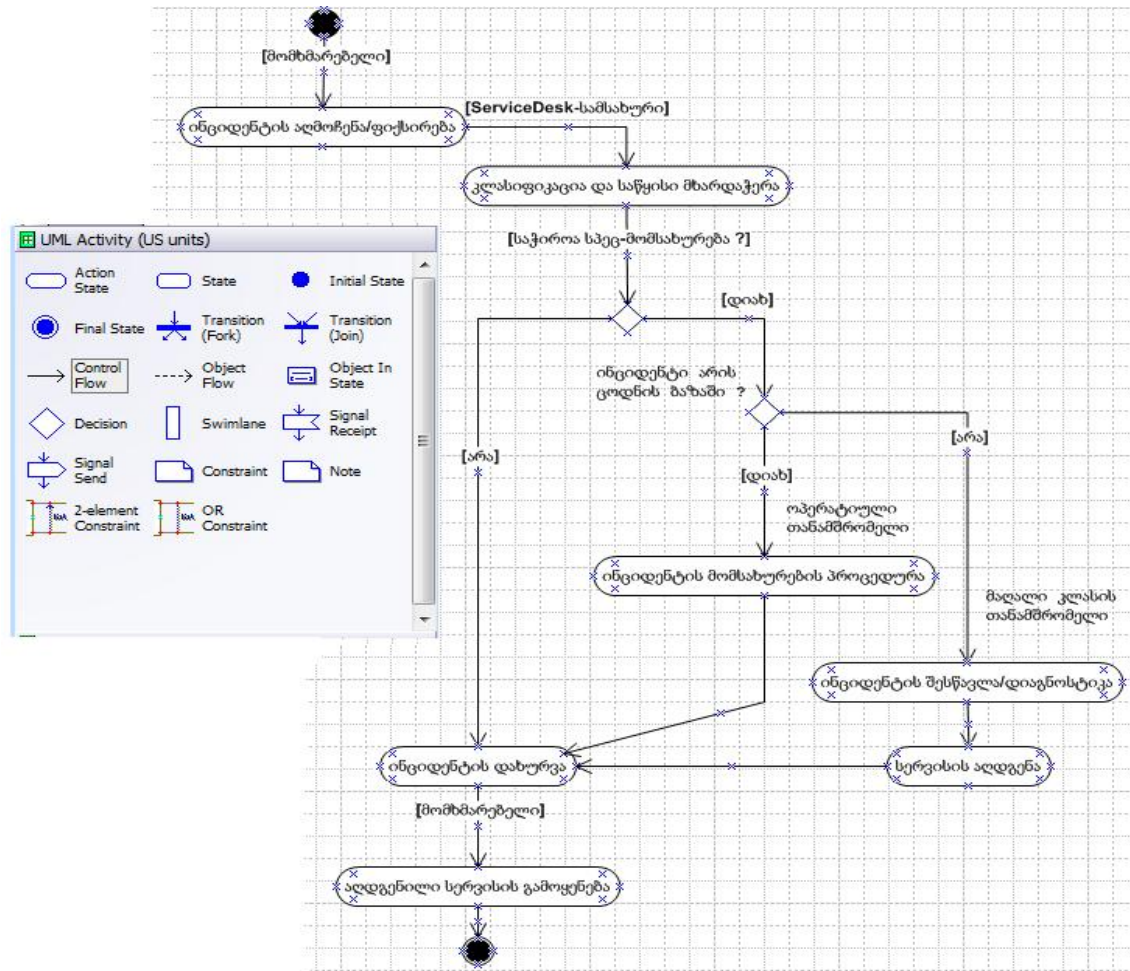
იგი მოითხოვს პროგრამული უზრუნველყოფის ისეთი გუნდის დაკომპლექტებას (მენეჯერი, სისტემური ანალიტიკოსი, დეველოპერები, ტესტირების სპეციალისტი, დამკვეთის წარმომადგენელი და სხვ.), რომლის მთავარი მიზანი IT-სერვისების (ეტაპობრივი) ვადების დაცვით შექმნა ან სრულყოფა და დამკვეთზე მიწოდებაა, ხოლო მუშაობის სტილი - მაღალპროფესიონალურ, ურთიერთშენაცვლების, დახმარების და ნდობის ფაქტორებზეა აგებული. ამასთანავე შემოთავაზებულია ITIL მეთოდოლოგიის პრინციპების გათვალისწინება, რაც მეტად მნიშვნელოვანია ინტეგრაციის პროცესების ორგანიზაციული საკითხების გადასაწყვეტად (ნახ.4). ამ პროცესში Agile მეთოდოლოგია გამოდის „კატალიზატორის“ როლში. მათი ერთობლივი გამოყენებით, სინერგეტიკის პრინციპების თვალსაზრისით, მიიღწევა უკეთესი შედეგები, ვიდრე მათი ცალ-ცალკე გამოყენების დროს.



ნახ.4. სერვისის სასიცოცხლო ციკლის ეტაპების ძირითადი კავშირები

შემდგომ, IT-სერვისების უწყვეტი სრულყოფის ციკლთან ერთად, აგებულ იქნა სერვისების *მხარდაჭერი* (ინციდენტების, პრობლემების, კონფიგურაციების, ცვლილებების და რელიზების მართვის) და *მიწოდების* (სერვისების დონის, სიმძლავრეების, უწყვეტობის, წვდომის, ფინანსების და

უსაფრთხოების მართვის) ბიზნეს-პროცესების მოდელები UML-ის Activity დიაგრამებით (ნახ.5), რომელთა საფუძველზე განისაზღვრა სისტემის ფუნქციონალური და მათი მხარდამჭერი არაფუნქციონალური (ინფრასტრუქტურული) მოთხოვნები.

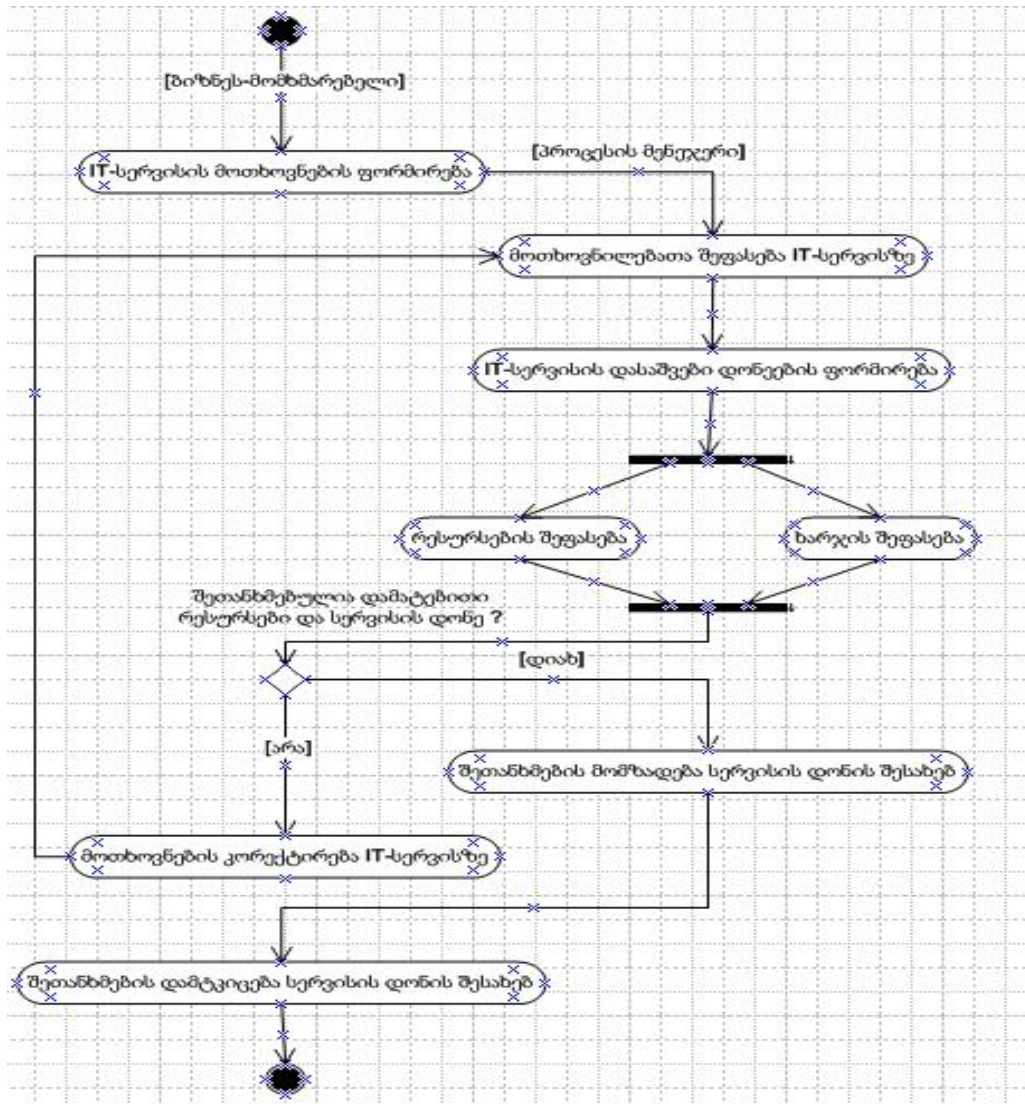


ნახ.5. ინციდენტების მართვის აქტიურობის დიაგრამა

მაგალითად, სერვისის დონის მართვის პროცესი (Service Level Management) განსაზღვრავს, ათანხმებს და აკონტროლებს IT-სერვისის პარამეტრებს, ბიზნესის თვალსაზრისით. პროცესის მენეჯერის მთავარი ფუნქციაა ბალანსის შენარჩუნება ბიზნესის მოთხოვნილებებსა და IT-ის შესაძლებლობებს შორის (ნახ.6).

სიმძლავრეების მართვის პროცესის (Capacity Management) დანიშნულებაა კი IT-ინფრასტრუქტურის რესურსების გამოყენების ოპტიმიზაცია, ბიზნესის მოთხოვნილებების შესაბამისად.

IT-სერვისების მიწოდების უწყვეტობის მართვის პროცესი (IT Service Continuity Management–ITSCM). იგი პასუხისმგებელია რისკების მართვაზე, რომლებიც გავლენას ახდენს სერვისებზე, რისკების შემცირებაზე მისაღებ დონემდე, აგრეთვე სერვისების აღდგენის დაგეგმვის შესაძლებლობაზე.



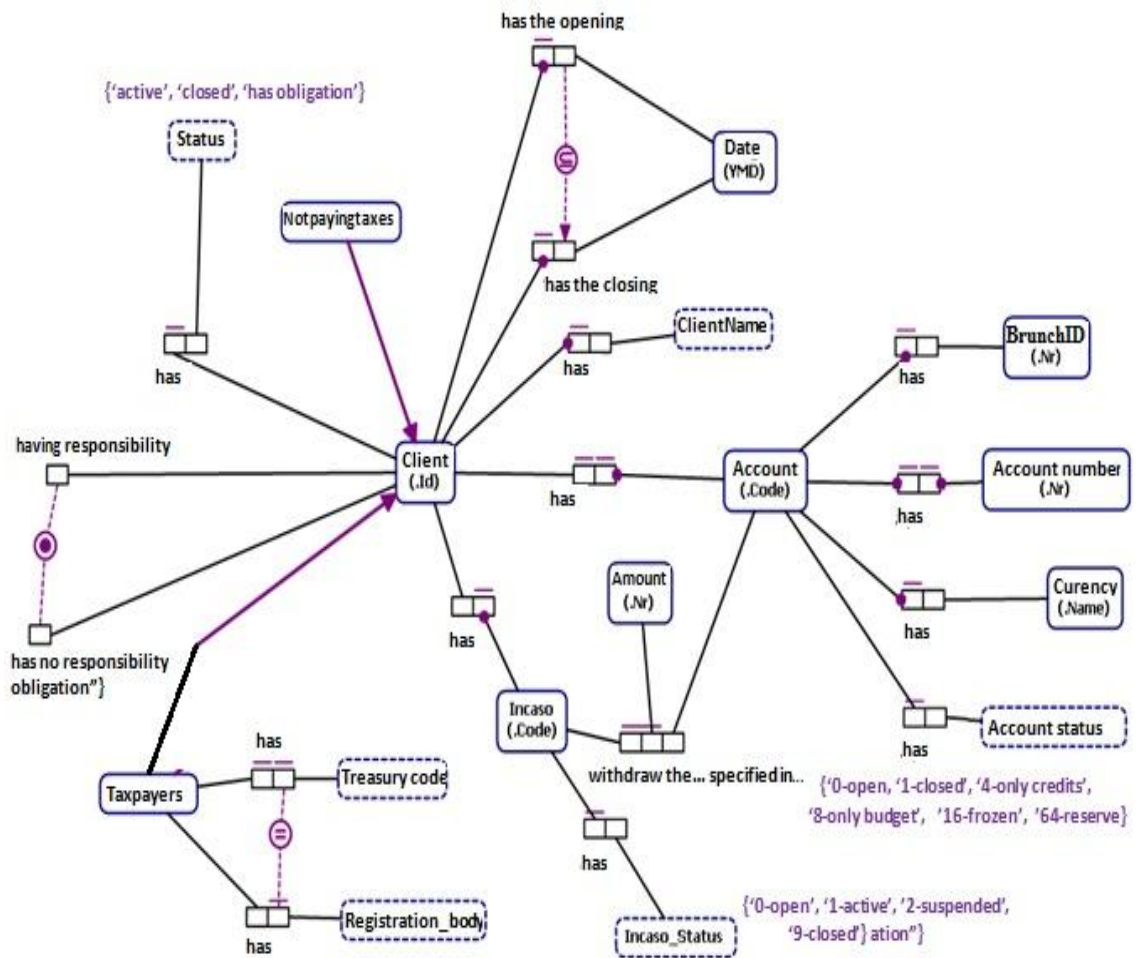
ნახ.6. სერვისის დონის მართვის აქტიურობის დიაგრამა

სერვისების უწყვეტობის მართვის ძირითადი მიზანია ბიზნესის უწყვეტობის მართვის პროცესის მხარდაჭერა. ბიზნესის უწყვეტობის მართვა (Business Continuity Management - BCM) - ესაა ბიზნეს-პროცესი, პასუხისმგებელი რისკების მართვაზე, რომელთაც შეუძლია სერიოზული გავლენა მოახდინოს ბიზნესზე.

ინფორმაციული უსაფრთხოების მართვის პროცესები (Information Security Management - ISM) არის პროცესი, რომელიც უზრუნველყოფს ორგანიზაციის აქტივების, ინფორმაციის, მონაცემების და სერვისების კონფიდენციალობას, მთლიანობას და წვდომას. მისი ძირითადი მიზანია ინფორმაციული უსაფრთხოების მართვის ეფექტური უზრუნველყოფა ყველა სერვისის და ქმედების სერვისების მართვის ფარგლებში.

დისერტაციაში კონკრეტული პროგრამული პროექტების შესრულების პროცესების ანალიზი განხორციელდა რამდენიმე, სხვადასხვა პრობლემური სფეროს მაგალითზე (სასიცოცხლო ციკლის ჩარჩოებში).

მოცემულია მათი განზოგადოების შედეგები. კერძოდ განხილულ იქნა „ბანკთაშორის ინტეგრირებული საინფორმაციო სისტემა“, „ელექტრონული საარჩევნო მულტიმედიური სისტემა“ და „შემოსავლების სამსახურის საგადასახადო დავების მართვის სისტემა“, მათი შესაბამისი პროგრამული უზრუნველყოფების ავტომატიზებული აგების პროცესები - მენეჯმენტის თვალსაზრისით. ყველა პროექტისთვის შემუშავდა სასიცოცხლო ციკლის პროცესების ავტომატიზებული დამუშავების ერთიანი მეთოდიკა, რაც გამოიხატა, მაგალითად, საკვლევი ობიექტის ცოდნის (კანონზომიერებათა ფაქტების) ასახვით მონაცემთა რელაციურ, ობიექტ-ორიენტირებულ ბაზაში ORM/ERM/DDL ავტოგენერაციით (ნახ.7).



ნახ.7. ORM დიაგრამის ფრაგმენტი

ფაქტების სიმრავლე:

- | | |
|-----------------------------------|---------------------------------------|
| f1 - Client has Name | f6- Client has Incasso |
| f2 - Client has Account | f7-Incasso has Incasso_Status |
| f3 - Client has Type | f8- Account has Account_Status |
| f4 - Account has Account_# | f9- Account has Branch |
| f5 - Account has Currency | f10- Client has Status ... etc |

ORM-დიაგრამაზე ობიექტები გამოსახულია ელიფსებში, წყვეტილი ელიფსი აღნიშნავს მნიშვნელობას (value), პრედიკატები გამოსახულია

მართკუთხედებში. გამოყენებულია ბინარული და ტერნარული პრედიკატები. შეზღუდვები ნახაზზე გამოსახულია სპეც-აღნიშვნებით. ჩვენ მიერ პროგრამულ ინსტრუმენტში შეტანილ ფაქტებს ობიექტებისა და მათი სემანტიკური კავშირების შესახებ აქვს შემდეგი სახე:

Fact Types:

Client aqvs ClientName.

Each Client aqvs exactly one ClientName.

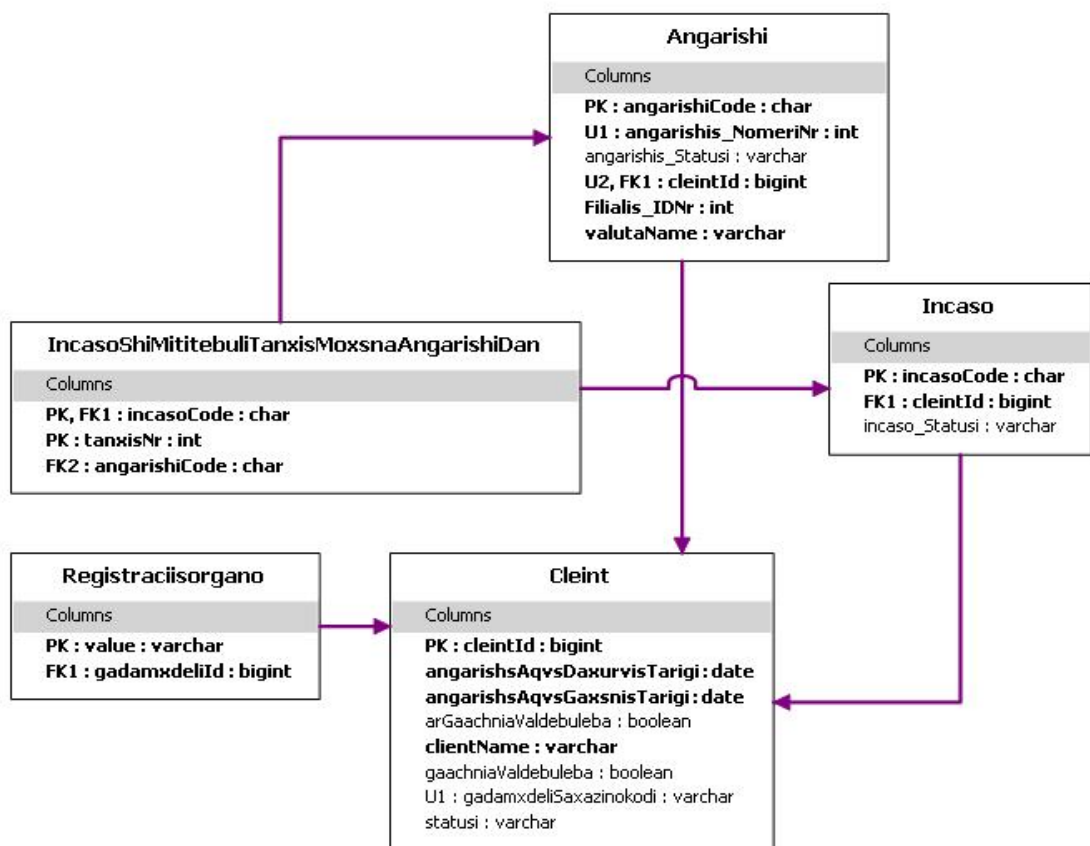
It is possible that more than one Client aqvs the same ClientName.

Gadamxdeli is an entity type.

Reference Scheme: Client has Client_Id.

Reference Mode: .Id.

... და ა.შ. ყველა ობიექტისთვის. შესაბამისად მიიღება ERM სქემა (ნახ.8).

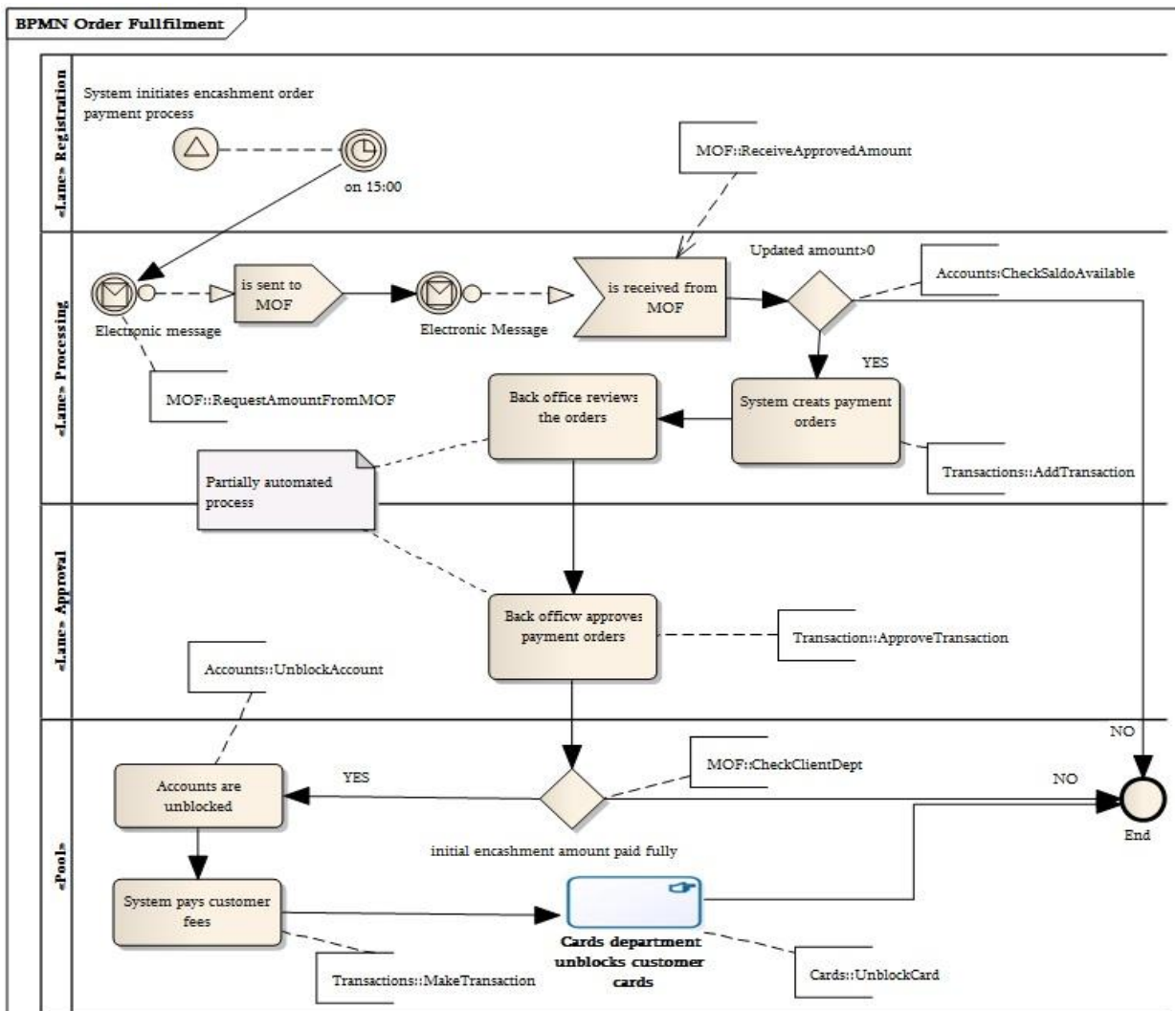


ნახ.8. ERM-დიაგრამის ფრაგმენტი

შემდეგ ბიჯზე ხორციელდება ERM-დიაგრამიდან ავტომატიზებულ რეჟიმში MsSQL Server მონაცემთა ბაზის DDL-ფაილის გენერირება, რის საფუძველზეც აიგება საბოლოო მონაცემთა ბაზა შესაბამისი ცხრილებით.

სერვისების ბიზნეს-პროცესების და ბიზნეს-წესების მოდელირება პროცეს-ორიენტირებული მიდგომით ხორციელდება BPMN ნოტაციით CASE Enterprise Architect ნსტრუმენტით. მაგალითად, მე-9 ნახაზზე ასახულია ბანკში მიღებული ინკასოების დამუშავების მოთხოვნის სერვისული ბიზნეს-პროცესის სქემა. ფუნქციონალობის აღწერა: „ბანკში

მიღებული ინკასოების დამუშავება იწყება წინასწარ განსაზღვრულ დროს. Windows სერვისის ბაზის ცხრილებში ნახულობს დასამუშავებელი ინკასოების სიას, იძახებს GetActiveOrder -ს მეთოდს, მიღებული ინკასოების ნომრები გადაეცემა RequestAmountFromMof მეთოდს, რომელიც აფორმირებს XML შეტყობინებას და იძახებს შემოსავლების სამსახურის ვებ-სერვისს. შემოსავლების სამსახური XML-ის სახის შეტყობინებას უბრუნებს ბანკის ვებ-სერვისს, რომელიც შეიცავს ინკასოების ნომრებს და შესაბამის თანხას. ეს ინფორმაცია იწერება ბაზაში. ბანკის თანამშრომელი პროგრამის საშუალებით ნახულობს ინკასოების სიას, გამოიძახება ვებ-სერვისის მეთოდი Get, ამტკიცებს მას, რის შემდეგაც გამოიძახება ბაზის პროცედურები, რომლებიც ატარებს ტრანზაქციებს, იხდიან ანგარიშებიდან თანხებს შემოსავლების სამსახურის ანგარიშზე. თუ თანხის დაფარვა მოხდება სრულად, შეიცვლება ანგარიშის სტატუსი „აქტიურზე“ და მოხდება ბარათების განზღოვვა თანამშრომლის მიერ.



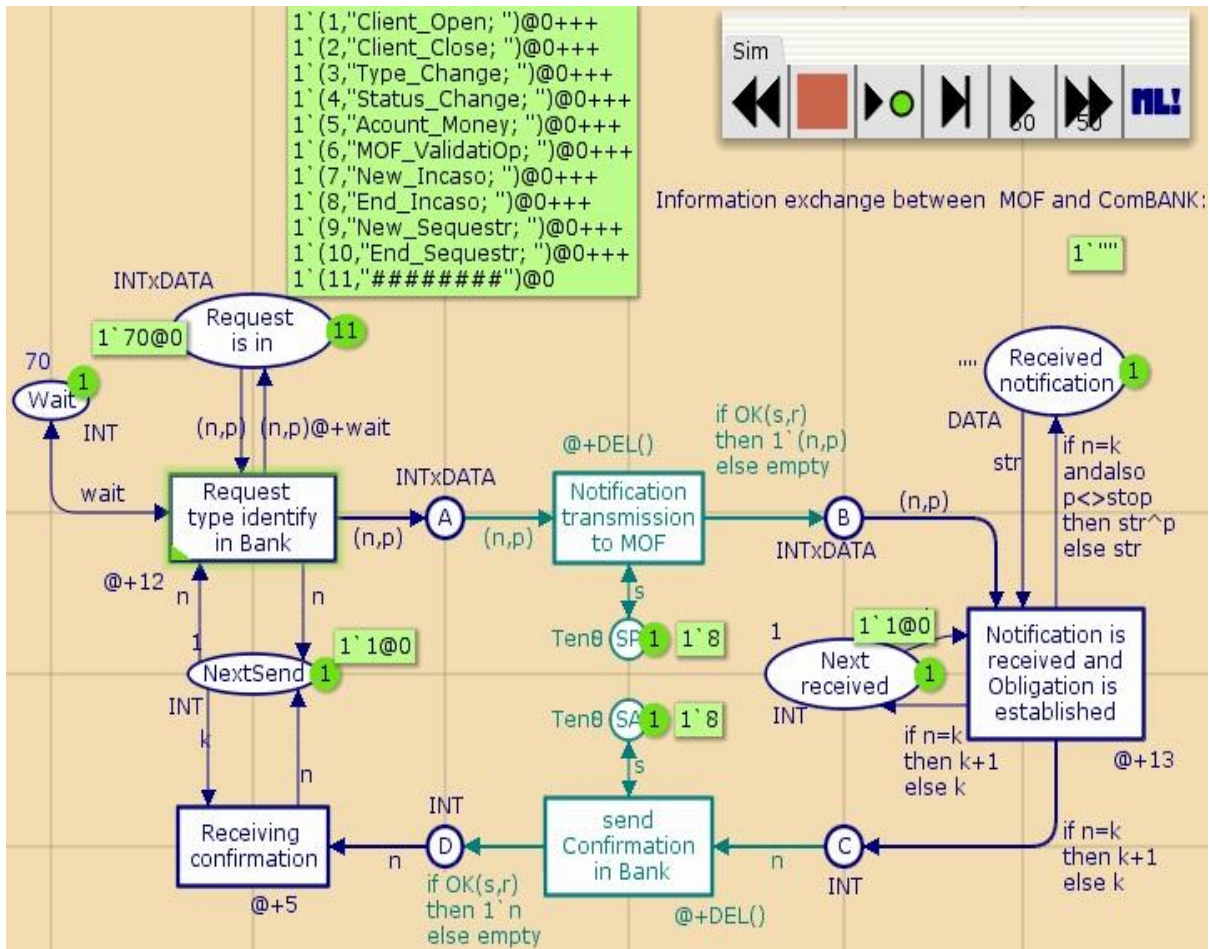
ნახ.9. Incoming Order Fulfilment (): სერვისებით

სერვის-ორიენტირებული არქიტექტურა აფართოებს ობიექტ-ორიენტირებული ტექნოლოგიის გამოყენების საზღვრებს და ინტეგრაციის შესაძლებლობას ქმნის. მთავარი განსხვავება ამ ორ ტექნოლოგიას შორის მეთოდების განზოგადებისა და წვდომის ასპექტში იკვეთება. პრინციპული განსხვავება ობიექტ-ორიენტირებულ და სერვის-ორიენტირებულ მიდგომებს შორის არის ორიენტირება ბიზნეს-პროცესზე და არა ობიექტზე. ამ პრინციპით, ვებ-სერვისში ინკაფსულირებულია ბიზნეს-ფუნქცია აბსტრაქციის მაღალი დონით, რაც უნიფიცირებული პროცესის შაბლონად შექმნის შესაძლებლობას ქმნის. სერვის-ორიენტირებული არქიტექტურის ერთ-ერთი პრინციპია ტიპიზირებული ვებ-სერვისების ბიბლიოთეკის არსებობა, მასში ამ შაბლონების შენახვა ვებ-სერვისის სახით და მისი მრავალჯერადად გამოყენება.

IT-სერვისების, შეტყობინებების და სხვა ინფორმაციის გადაცემის (გაცვლის) დინამიკური პროცესების კვლევის მიზნით იმიტაციური მოდელების აგება პეტრის ფერადი ქსელების (CPN) საფუძველზე განხორციელდა ყველა პროექტისთვის (ბანკები, ელ-არჩევნები, საგადასახადო დავები). მაგალითად, საფინანსო ბანკებსა და შემოსავლების სამსახურს შორის ინფორმაციის გაცვლა სერვისული ბიზნეს-პროცესების საფუძველზე ხორციელდება შეტყობინებათა და მონაცემთა პაკეტების გაცვლის სერვისების ხშირი გამოყენებით (ყოველდღიურად ბანკმა შეიძლება მიიღოს (ან გადასცეს) 1000-ზე მეტი მოთხოვნა კლიენტების შესახებ). ასეთი ინფორმაციის მენეჯმენტი მოითხოვს საიმედო აღრიცხვისა და რისკების გამორიცხვის პროცედურების გათვალისწინებას. შეტყობინებათა ერთობლიობა, რომელიც მუდმივად გადაიცემა ქსელის საშუალებით, არ უნდა დაიკარგოს და ყოველი მათგანი უნდა ექვემდებარებოდეს მკაცრ კონტროლს, უნდა შეიძლებოდეს აღდგენა, ანუ განმეორებითი პროცედურის შესრულება. ასეთი სერვისული მოთხოვნების დამუშავების მართვის პროცესის მოდელირება ჩვენ განვახორციელეთ CPN ინსტრუმენტით (ნახ.10).

პროცესების შესრულების დრო (დაყოვნება) აისახება გადასავლელთან სიმბოლოს და დროის ერთეულის (მაგალითად, @+7, @+wait) მითითებით, სადაც wait წინასწარ განსაზღვრული კონსტანტაა.

ამავე ნახაზზე ასახულია არადეტერმინირებული ლოგიკური გამოსახულება (პირობის ბლოკი) ფერადი პეტრის ქსელის რკალებზე, რომლებიც გადასასვლელთა გაშვების სხვადასხვა პირობებს და შედეგებს ასახავს. ლოგიკური პირობის ჭეშმარიტებისას გადასასვლელს განსხვავებული მნიშვნელობა მიეწოდება (ან გადასასვლელიდან განსხვავებული მნიშვნელობა გამოვა), მცდარობისას – განსხვავებული.



ნახ.10. MOF-BANK კორპორაციული კავშირების პროცესების იმიტაციური მოდელი CPN-ის გარემოში: „მოთხოვნების დამუშავება“

მაგალითად, გადასასვლელს „ნოტიფიკაციის გადაგზავნა MOF-ში“ გამოსასვლელ რკალზე აქვს ლოგიკური პირობა - თუ გამოგზავნილი ნოტიფიკაციის ნომერი (n) ემთხვევა კლიენტის ანგარიშის ნომერს (k), მაშინ გვაქვს "true", წინააღმდეგ შემთხვევაში „false“, რაც იმას ნიშნავს, რომ საჭირო ნოტიფიკაცია არაა მისული MOF-ში. თუ ყველაფერი წესრიგშია, მაშინ MOF (მიმღები) უგზავნის ბანკს შეტყობინებას გადასასვლელით „დასტურის_გამოგზავნა“. ნოტიფიკაციის და შეტყობინების გადაცემათა ქსელში შემთხვევითი პროცესის არსებობა განპირობებულია დაყოვნების ცვლადი დროის გამო, რაც აისახება colset NetDelay=int with 25..75, fun DEL() =NetDelay.ran() random-ფუნქციით. ლოგიკური პირობის მნიშვნელობა სხვადასხვა შემთხვევებში სხვადასხვანაირად განისაზღვრება. ინტერაქტიულ სიმულატორებში ჭეშმარიტება-მცდარობას თავად მომხმარებელი განსაზღვრავს, ავტომატური სიმულაციისას – შემთხვევით რიცხვთა გენერატორი.

დროითი CPN -მოდელით შეიძლება გამოვიკვლიოთ კორპორაციათა შორის (მაგალითად, ბანკსა და შემოსავლების სამსახურს შორის)

შეტყობინებათა გაცვლის პროცესის მახასიათებლები. შეტყობინებათა განმეორებითი გადაცემის დაყოვნების დროის (wait) სხვადასხვა მნიშვნელობისათვის. ხანმოკლე დაყოვნება ზრდის შანსს განმეორებითი გადაგზავნების თავიდან ასაცილებლად. იგი ასევე ზრდის შანსს, რომ ოპერაცია Dasturis_migeba გადაიდოს, რადგან პროცესი Motxovnis tipis dadgena bankshi დაკავებულია განმეორებითი გადაგზავნით. გრძელი დაყოვნება ნიშნავს, რომ საჭირო იქნება დიდხანს ცდა, სანამ ბანკი დარწმუნდება, რომ შეტყობინება ან დასტური იქნა დაკარგული. სიმულაციის პროცესში, სხვადასხვა wait-მნიშვნელობით შეიძლება დადგინდეს ოპტიმალური მნიშვნელობა განმეორებითი გადაცემის დაყოვნებისათვის.

Notifikaciis_gadagzavna გადასასვლელიდან B-პოზიციაში შემავალი რკალი აკონტროლებს ლოგიკურ პირობას, ანუ დასაშვებია ორი შემთხვევა:

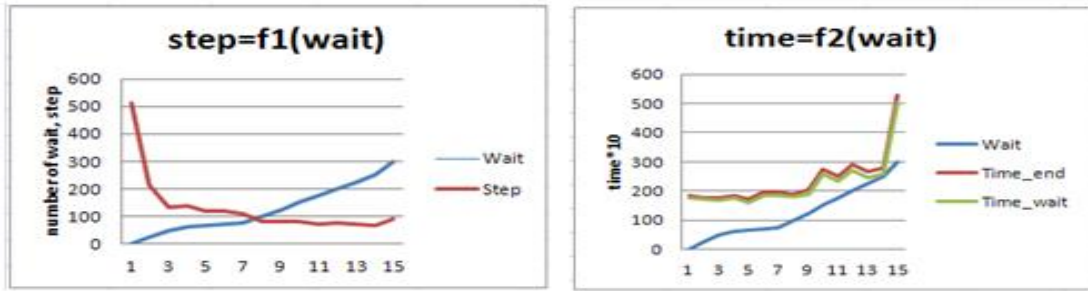
TP + = (Notifikaciis_gadagzavna, <n=1,p="Client_Open", success=true>),
TP - = (Notifikaciis_gadagzavna, <n=1,p="Client_Open", success=false>).

ეს ორი დამაკავშირებელი ელემენტი TP+ და TP- იმყოფება კონფლიქტში ერთმანეთთან, ანუ ერთის შესრულება მეორეს გამორიცხავს. პირველით მოდელირდება ქსელში ნოტიფიკაციის წარმატებით გადაცემა, ხოლო მეორეთი კი – ამ შეტყობინების დაკარგვა.

პეტრის ქსელის იმიტაციური მოდელის საფუძველზე პროცესების კვლევის შედეგად, როდესაც ვცვლით მოთხოვნების ანალიზის და შესრულების დასაწყისის დაყოვნების დროს (wait - პოზიცია) ინტერვალში [0-:300], მივიღეთ 1-ელი ცხრილი და მე-11 ნახაზზე ნაჩვენები დიაგრამები.

ცხრ.1

N	Wait	Step	Time_end	Time_wait
1	0	512	1856	1773
2	25	213	1759	1734
3	50	134	1771	1690
4	60	139	1833	1773
5	65	118	1714	1588
6	70	120	1977	1864
7	75	111	1956	1839
8	100	83	1887	1786
9	125	79	2036	1907
10	150	80	2729	2579
11	175	73	2514	2339
12	200	77	2893	2693
13	225	71	2686	2461
14	250	67	2794	2594
15	300	89	5289	4989



ნახ.11. იმიტაციის შედეგები: wait დროის ცვლილებით.
1: 15 ექსპერიმენტის ნომერი; wait-დაყოვნების დრო;
step-ბიჯების რაოდენობა; Time- სერვისების დამუშავების
საბოლოო დრო

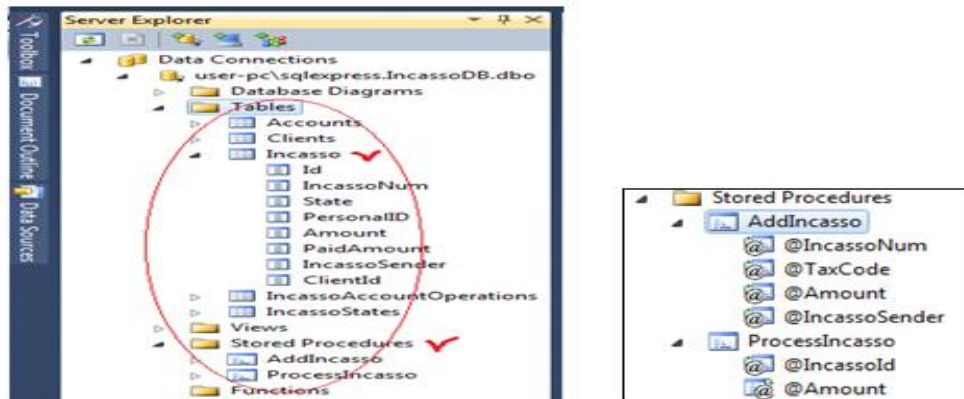
ჩვენ მიერ დამუშავებული სხვა სფეროებისთვის, კერძოდ, ელ-საარჩევნო მულტიმედიაური საინფორმაციო სისტემის და საგადასახადო დავების მართვის სისტემის პროექტებისთვის მიღებულ მსგავს შედეგებს აქ აღარ განვიხილავთ, ისინი აღწერილია როგორც სადისერტაციო ნაშრომში, ასევე ავტორეფერატის ბოლოში, გამოქვეყნებულ ნაშრომები.

IT-ინფრასტრუქტურის ეფექტური მართვის მიზნით შემოთავაზებულია System Center 2012 Service Manager ინსტრუმენტის გამოყენება, რაც შესაძლებელია პრაქტიკულად ნებისმიერი მასშტაბის ორგანიზაციისათვის. სისტემა აგებულია Microsoft-ის WPF-კლასების ბიბლიოთეკის გამოყენებით, რაც მისი ახალი კლასებითა და ფორმებით დინამიკურად გაფართოების შესაძლებლობას იძლევა.

მესამე თავში გადმოცემულია პროგრამული სისტემების მენეჯმენტის ბიზნეს-პროცესების პროგრამული რეალიზაცია .NET პლატფორმაზე, Windows Presentation Foundation (WPF), Workflow Foundation (WF) და Windows Communication Foundation (WCF) ტექნოლოგიებით, ობიექტ-ორიენტირებული დაპროგრამების C#.NET და აპლიკაციების დიზაინის აგების XAML ენებით. მონაცემთა განაწილებული ბაზის სახით გამოყენებულია MsSQL Server პაკეტი და ADO.NET დრაივერი.

ნაჩვენებია ზემოაღნიშნული პროექტების პროგრამული რეალიზაციის ფრაგმენტები, ძირითადად კლიენტ-სერვერული და სერვის-ორიენტირებული არქიტექტურების ბაზაზე. მოცემულია ასევე მომხმარებელთა ინტერფეისების საილუსტრაციო მაგალითები შესაბამისი პროგრამული ლისტინგები. მაგალითად, მე-12 ნახაზზე ნაჩვენებია Incasso.DB მონაცემთა ბაზის სტრუქტურა Accounts, Clients, Incasso, IncassoAccountOperatio და IncassoStates.

მონაცემთა ბაზის შენახვად პროცედურებში (Stored Procedures: AddIncasso, ProcessIncasso) რეალიზებულია სერვის-ფუნქციები. ქვემოთ, 1-ელ ლისტინგში ნაჩვენებია ამ პროცედურის ტექსტი:



ნახ.12. IncassoDB ბაზა და შენახვადი პროცედურები

```
-- ===== AddIncasso =====ლისტი-1=====
-- Description: ახალი ინკასოს შემოსვლა
-- =====
ALTER PROCEDURE [dbo].[AddIncasso]
    @IncassoNum varchar(15),
    @TaxCode varchar(15),
    @Amount money,
    @IncassoSender tinyint
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
    SET NOCOUNT ON;
    DECLARE @State tinyint = 0 -- გადაუხდელი
    DECLARE @ClientId int
    DECLARE @PaidAmount money
    Declare @incasoId int
    DECLARE @PersonalID varchar(12)
    --კლიენტის მოძებნა პირადი ნომერის მიხედვით
    SELECT @ClientId = Id, @PersonalID = PersonalID FROM dbo.Clients where
    TaxCode = @TaxCode
    IF ISNULL(@ClientId,0)=0
    SET @State = 1--არ არის ბანკის კლიენტი/არ არის გადასახდელი
    -- Insert statements for procedure here
    INSERT INTO [dbo].[Incasso]
        ([IncassoNum]
        ,[State]
        ,[PersonalID]
        ,[Amount]
        ,[IncassoSender]
        ,[ClientId])
    VALUES
        (@IncassoNum
        ,@State
        ,@PersonalID
        ,@Amount
        ,@IncassoSender
        ,@ClientId)
    if @@ROWCOUNT = 1 RETURN SCOPE_IDENTITY()
    RETURN -1
END
```


მე-12 ნახაზზე ილუსტრირებულია აპლიკაციის ამუშავების შემდეგ მიღებული შედეგები ბანკში კლიენტების მიხედვით ინკასოს მდგომარეობა.

კლიენტი	ინკასო	ანგარიში/თანა
კლიენტი გამრეჯლა/10589789	#89-66 / GEL 2500.00	ანგ# 108897305 / GEL 500.00
ნიონი ჭელიძე/58109634	#89-66 / GEL 250.00	
ტარიელ ტალახვერდოვი/879450100	#89-66 / GEL 300.00	
ელენე გობეჩია/10989533		

კლიენტი	ინკასო	ანგარიში/თანა
გიორგი გამრეჯლა/10589789	#89-66 / GEL 2500.00	ანგ# 108897301 / ანგ 250.00
ნიონი ჭელიძე/58109634	#89-66 / GEL 250.00	
ტარიელ ტალახვერდოვი/879450100	#89-66 / GEL 300.00	
ელენე გობეჩია/10989533		

ნახ.12. ინტერფეისის მაგალითი ერთი კლიენტის 2 ინკასოს შემთხვევით

ბიზნეს-პროცესების შესრულებისას ერთ-ერთი მნიშვნელოვანი საკითხია კომუნიკაცია აპლიკაციებს შორის, კლიენტებსა და სერვერებს შორის, აგრეთვე ბიზნეს-პროცესებსა და ჰოსტ-დანართებს შორის.

მთავარი ქმედებები, რომლებიც კომუნიკაციისთვის გამოიყენება არის Send და Receive ქმედებები (და მათი ვარიაციები: SendReply და ReceiveReply). ეს ქმედებები გამოიყენებს Windows Communication Foundation (WCF) ტექნოლოგიას შეტყობინებათა გადასაცემად და მონიტორინგისთვის. ბიზნეს-პროცესები თავსდება ვებ-სერვისში, რომელიც უზრუნველყოფს იდეალურ საშუალებას მუშა პროცესის გადაწყვეტილების მისაწოდებლად არა-მუშა პროცესის კლიენტებისთვის, როგორცაა ვებ-აპლიკაციები. ვებ-სერვისი იღებს მოთხოვნას, ასრულებს მის სათანადო გადამამუშავებას და აბრუნებს პასუხს. ეს, ბუნებრივია, სრულდება Receive და Send ქმედებებით. ვინაიდან ეს აქტიურობები ინტეგრირებულია Windows Communication Foundation (WCF) -თან, ჩვენ შეგვიძლია ადვილად შევქმნათ WCF სერვისები. მე-13 ნახაზზე მოცემულია ბანკთაშორისი საინფორმაციო სისტემის მომხმარებლის ინტერფეისის ერთი მაგალითი.

Reservations

საფინანსო ბანკი

მოთხოვნათა სია

ფიზიკური პირი:

პირადი N:

ანგარიშის N:

Event Log

ნახ.13. მომხმარებლის ინტერფეისი

ფორმის ზედა ნაწილში „მოთხოვნების სია“ ასახავს შემოსულ მოთხოვნებს, რომლებიც მოქმედებაშია. მოთხოვნის გასაგზავნად სხვა ფილაში გამოიყენება ველები ფორმის შუაში. აქ მიეთითება „ფიზიკური_პირი“, „პირადი_N“ და „ანგარიშის_N“, შემდეგ გაგზავნის ღილაკი „პასუხი“. Event Log ქვედა მარცხენა კუთხეში ასახავს ბიზნეს-პროცესის შეტყობინებას. ფორმის დიზაინი აგებულია BankRequest.xaml ფაილით, რომლის ფრაგმენტიც მე-2 ლისტინგშია:

```
<!-- ლისტინგი_2 ---- -->
<Window x:Class=" BankRequest.MainWindow"
...
Loaded="Window_Loaded" Unloaded="Window_Unloaded">
<Grid>
<Label Height="40" HorizontalAlignment="Left" Margin="12,0,0,0"
Name="lblBranch" FontSize="24" VerticalAlignment="Top" Width="276"
FontStretch="Expanded">საფინანსო ბანკი</Label>
<ListView x:Name="მოთხოვნების სია" Margin="12,42,12,5" Height="150"
VerticalAlignment="Top" ItemsSource="{Binding}">
<ListView.View>
<GridView>
<GridViewColumn Header="მოთხოვნების სია" Width="610">
<GridViewColumn.CellTemplate>
<DataTemplate>
<StackPanel Orientation="Horizontal">
<TextBlock Text="{Binding Requester.BranchName}" Width="100"/>
<TextBlock Text="{Binding ფიზიკური_პირი}" Width="95"/>
<TextBlock Text="{Binding პირადი_N}" Width="180"/>
<TextBlock Text="{Binding ანგარიშის_N}" Width="90"/>
<Button Content="Reserve" Tag="{Binding InstanceID}"
Click="Reserve" Width="65"/>
<Button Content="Cancel" Tag="{Binding InstanceID}"
Click="Cancel" Width="60"/>
</StackPanel>
</DataTemplate>
</GridViewColumn.CellTemplate>
</GridViewColumn>
</GridView>
</ListView.View>
</ListView>
...
HorizontalContentAlignment="Right">ფიზიკური_პირიAuthor:</Label>
HorizontalContentAlignment="Right">პირადი_N:</Label>
HorizontalContentAlignment="Right">ანგარიშის_N:</Label>
...
Click="btnRequest_Click">პასუხი</Button>
</Grid>
</Window>
```

სერვისის პროგრამული რეალიზაცია (ClientService.cs) ნაჩვენებია მე-3 ლისტინგში C# კოდით.

```
// ===== ლისტინგი 3 =====
```

```

using System;
using System.ServiceModel;
namespace LibraryReservation
{ public class ClientService : ILibraryReservation
  {
    public void RequestBook(ReservationRequest request)
    {
      ApplicationInterface.RequestBook(request);
    }
    public void RespondToRequest(ReservationResponse response)
    {
      ApplicationInterface.RespondToRequest(response);
    }
  }
}

```

აპლიკაციისთვის აუცილებელია ServiceHost-ის რეალიზაცია შემავალი შეტყობინებების მისაღებად. BankRequest.xaml.cs ფაილის ფრაგმენტი ნაჩვენებია მე-3 ლისტინგში.

```

// ===== ლისტინგი 3 =====
public partial class MainWindow : Window
{
  private ServiceHost _sh;  //!
  public MainWindow()
  {
    InitializeComponent();
    ApplicationInterface._app = this;
  }
}

```

იწყება ServiceHost მაშინ, როცა ფანჯარა ჩატვირთულია და იხურება, როცა ფანჯარა ამოტვირთულია. მეთოდების დამატება ნაჩვენებია მე-4 ლისტინგში MainWindow კლასისთვის ჩატვირთვის და ამოტვირთვის მოვლენათა დამმუშავებლების სარეალიზაციოდ.

```

// ===== ლისტინგი 4 =====
private void Window_Loaded(object sender, RoutedEventArgs e)
{
  // გაიხსნას config ფაილი, მიეცეს ბანკის სახელი და მისი ქსელური მისამართი
  Configuration config =
  ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);
  AppSettingsSection app =
  AppSettingsSection)config.GetSection("appSettings");
  string adr = app.Settings["Address"].Value;
  // ფილიალის სახელის გამოტანა ფორმაზე
  lblBranch.Content = app.Settings["საფინანსო ბანკი"].Value;
  // ServiceHost-ის შექმნა
  _sh = new ServiceHost(typeof(ClientService));
  // დასასრულის წერტილის (Endpoint) დამატება
  string szAddress = "http://localhost:" + adr + "/ClientService";
  System.ServiceModel.Channels.Binding bBinding = new BasicHttpBinding();
  _sh.AddServiceEndpoint(typeof(ILibraryReservation),bBinding, szAddress);
}
private void Window_Unloaded(object sender, RoutedEventArgs e)
{
  // service host-ის დატოვება
  _sh.Close();
}

```

მოვლენის დამმუშავებელი Loaded ხსნის კონფიგურაციის ფაილს და ათავსებს ფილიალის სახელს lblBranch -მართვის ელემენტში, ამიტომაც ფორმა ასახავს ლოკალური ფილიალის სახელს. შემდეგ იქმნება ServiceHost თანამგზავრი (passing) ClientService კლასის. იგი აკონფიგურირებს დასასრულის წერტილს ServiceHost-თვის, იყენებს რა ცნობილი მისამართის, მიმმის და კონტრაქტის სამეულს. Unloaded მოვლენის დამმუშავებელი უბრალოდ ხურავს ServiceHost-ს, ასე რომ მეტი აღარ მოხდება შეტყობინებების მიღება.

დისერტაციის ბოლოს მოცემულია **დასკვნა** და გამოყენებული ლიტერატურის სია. სადისერტაციო თემაზე ავტორის მიერ გამოქვეყნებულია რვა სამეცნიერო ნაშრომი

დასკვნა

სადისერტაციო თემის ფარგლებში ჩატარებული საპროექტო-კვლევითი სამუშაოების შედეგების საფუძველზე შესაძლებელია შემდეგი დასკვნების გაკეთება:

1. საპრობლემო სფეროს მართვის საინფორმაციო სისტემის პროგრამული უზრუნველყოფის შექმნის ეფექტური მენეჯმენტის განსახორციელებლად საჭიროა პროგრამული აპლიკაციის სასიცოცხლო ციკლის ცალკეული ეტაპების და ფაზების სრულყოფა და საკონტროლო წერტილებში მონიტორინგის რეალიზაცია იტერაციათა შედეგების შესაფასებლად. უპირველეს ყოვლისა საჭიროა ამ სფეროს ბიზნეს-პროცესების ანალიზის ჩატარება, ფუნქციონალურ და არაფუნქციონალურ მოთხოვნილებათა განსაზღვრის მიზნით, პროცეს-ორიენტირებულ (BPMN), ობიექტ-ორიენტირებულ (UML) და მოქნილი პროგრამირების (Agile) მიდგომათა ტექნოლოგიებით;

2. UML და Agile მეთოდოლოგიები პროგრამული სისტემების მენეჯმენტის „ორი სხვადასხვა პოლუსის“ კონცეფციებია, რომელთაც თავ-თავისი დადებითი და ნაკლოვანი მხარეები გააჩნია. პირველი მათგანი აუცილებელი და ეფექტურია დიდი პროგრამული პროექტების სასიცოცხლო ციკლის სამართავად, მეორე კი - მცირე და საშუალო პროექტებისთვისაა ხელსაყრელი (პროექტების შესრულების შეზღუდული დროის გამო). ერთ-ერთი ახალი, ჰიბრიდული მიმართულება პროგრამული სისტემების მენეჯმენტისთვის, ითვალისწინებს UML-ის და ექსტრემალური დაპროგრამების (Agile) მეთოდოლოგიათა კომპრომისულ გამოყენებას. იგი ეფუძნება პროგრამული უზრუნველყოფის დეველოპერების მოქნილი გუნდის ჩამოყალიბებას, რომლის მთავარი მიზანი IT-სერვისების

(ეტაპობრივი) ვადების დაცვით შექმნა და დამკვეთზე მიწოდებაა, ხოლო მუშაობის სტილი - მაღალპროფესიონალურ, ურთიერთშენაცვლების, დახმარების და ნდობის ფაქტორებზეა აგებული;

3. ITIL მეთოდოლოგია ინტეგრაციის პროცესების ორგანიზაციულ საკითხებს ემსახურება, ამ პროცესში Agile მეთოდოლოგია გამოდის „კატალიზატორის“ როლში. მათი ერთობლივი გამოყენებით, სინერგეტიკის პრინციპების თვალსაზრისით, მიიღწევა უკეთესი შედეგები, ვიდრე მათი ცალ-ცალკე გამოყენების დროს;

4. მნიშვნელოვნად უნდა ჩაითვალოს IT-სერვისების (პროგრამების) რელიზების მართვის პროცესის ეფექტურად ორგანიზაცია, IT-ინფრასტრუქტურაში განხორციელებული ცვლილებების შეთანხმების უზრუნველყოფის მიზნით. რელიზი არის ახალი ან შეცვლილი კომპონენტების ერთობლიობა, რომლებიც გადის დამუშავების, ტესტირების და დანერგვის ეტაპებს (მაგალითად, აპლიკაციათა ვერსიები იტერაციების მიხედვით);

5. საკვლევი სფეროებისთვის, როგორცაა ბანკთაშორისი ინტეგრირებული საინფორმაციო სისტემა, მულტიმედიური ელექტრონული საარჩევნო სისტემა და საგადასახადო დავების წარმოების სისტემა, განხორციელდა მონაცემთა განაწილებული ბაზის კონცეპტუალური, ლოგიკური და ფიზიკური სტრუქტურების მოდელირება და დაპროექტება CASE ინსტრუმენტული საშუალებით (Natural ORM Architect ბაზაზე). კერძოდ, კონკრეტული საპრობლემო სფეროსთვის განსაზღვრულ იქნა „ფაქტების სიმრავლე“ (ცოდნა მართვის ობიექტის შესახებ), რომლის საფუძველზეც ავტომატიზებულ რეჟიმში დაპროექტდა ობიექტ როლური მოდელი (ORM დიაგრამა - კონცეპტუალური მოდელის 1-ელი დონე). შემდეგ მის საფუძველზე აგებულ იქნა არსთა დამოკიდებულების მოდელი (ERM დიაგრამა - კონცეპტუალური მოდელის მე-2 დონე) და ბოლოს, ავტომატიზებულად გენერირდა მონაცემთა აღწერის ენის DDL-ფაილი და Ms SQL Server მონაცემთა ბაზის შექმნა;

6. განაწილებული გამოყენებითი საინფორმაციო სისტემებისთვის, ბანკთაშორისი-, ელ-საარჩევნო და საგადასახადო დავების წარმოების ბიზნეს-პროცესების მაგალითებზე, დამუშავდა პეტრის ფერადი ქსელების იმიტაციური მოდელები. მათი დახმარებით შესაძლებელი გახდა შესაბამისი IT სერვისების, შეტყობინებების და სხვა ინფორმაციული ერთეულების კლიენტ-სერვერ კვანძებს შორის (დამკვეთი-მიმწოდებელი) გადაცემა/გაცვლის პროცესების გამოკვლევა იმიტაციურ რეჟიმში. წარმოდგენილია ექსპერიმენტების ჩასატარებელი შესაბამისი პეტრის

ფერადი ქსელების გრაფული სქემები (CPN) და კვლევის შედეგად მიღებული შეფასების ცხრილები და დროითი დიაგრამები.

7. თეორიულად შემუშავებული საკითხები, მოდელების და მეთოდების სახით, რეალიზებულია პროგრამულად Ms Visual Studio.NET Framework 4.0 ინტეგრირებულ გარემოში - C#.NET,HTML, XAML ენების, WPF, WF,WCF ტექნოლოგიების, ADO.NET დრაივერის, MsSQL Server მონაცემთა ბაზის მართვის სისტემის, Ms Visio და Natural ORM Architect CASE ინსტრუმენტული საშუალებების გამოყენებით.

8. IT-ინფრასტრუქტურის ეფექტური მართვის მიზნით System Center 2012 Service Manager წარმოადგენს მოხერხებულ ინსტრუმენტს პრაქტიკულად ნებისმიერი მასშტაბის ორგანიზაციისათვის. სისტემა აგებულია Microsoft-ის WPF-კლასების ბიბლიოთეკის გამოყენებით, რაც მისი ახალი კლასებითა და ფორმებით დინამიკურად გაფართოების შესაძლებლობას იძლევა.

გამოქვეყნებული ლიტერატურა:

1. Gogichaishvili G., Surguladze G., Topuria N., **Urushadze B.** Construction of Management Information Systems of Distributed Business Processes Based on Petri networks and Object-Role Modeling. Bulletin "Moambe", vol. 8, N. 1. Georg. National Academy of Sciences. Tbilisi, 2014. pp.58-64

2. სურგულაძე გ., ბულია ი., **ურუშაძე ბ.** საინფორმაციო სისტემების ინტეგრაციის პროცესების მენეჯმენტი სერვის-ორიენტირებული არქიტექტურით და UML/AGILE ბაზირებული მეთოდებით. სტუ შრ.კრ.: „მას“-N2(13). თბილისი, 2012, გვ. 7-22.

3. **ურუშაძე ბ.** პროგრამულ აპლიკაციათა ინტეგრაცია AGILE და ITIL მეთოდოლოგიების საფუძველზე. სტუ შრ.კრ. “მართვის ავტომატიზებული სისტემები”, N3(16), თბ., 2013. გვ. 101-106

4. სურგულაძე გ., გულუა დ., **ურუშაძე ბ.**, კაშიბაძე მ. ორგანიზაციის საინფორმაციო ინფრასტრუქტურის ავტომატიზების თანამედროვე მეთოდები. სტუ-ს შრ.კრ. „მას“-N 1(14). 2013. გვ.109-114

5. ბიტარაშვილი მ., რამიშვილი ა., **ურუშაძე ბ.**, მიქიაშვილი ნ. საგადასახადო დავების ბიზნეს-პროცესების მოდელირება რიგების თეორიის საფუძველზე და პროგრამული რეალიზაცია. სტუ-ს შრ.კრ. „მას“-N 2(15). 2013. გვ.174-178

6. სურგულაძე გ., თოფურია ნ., ბასილაძე გ., **ურუშაძე ბ.**, ლომიძე მ., გაბინაშვილი ლ. პროგრამული სისტემების მენეჯმენტი მულტიმედიალური აპლიკაციების დასაპროექტებლად და ასაგებად. VI საერთ. სამეცნ.პრაქტ. კონფერენცია. „ინტერნეტი და საზოგადოება“. აკ.წერეთლის სახ.უნივ. ქუთაისი, სექტ., 2013. გვ. 66-70

7. სურგულაძე გ., ურუშაძე ზ. საინფორმაციო სისტემების მენეჯმენტის საერთაშორისო გამოცდილება (BSI, ITIL, COBIT). სტუ, თბ., 2014. 320 გვ. http://gtu.ge/book/gia_sueguladze/1giasurg.pdf

8. Surguladze G., Topuria N., Chikovanu d. Towards an Integration of Process-Modeling: from Business Method: from Business-Content to the Software Implementation. Journal of Business, IBSU, vol.2. Black Sea Univ. Tbilisi, 2013. pp. 15-18

ABSTRACT

In the dissertation “Program System Management with UML/Agile base Methods” the tasks for the management of software provision for the information systems of the corporate institutions are investigated and solved with process-oriented projection (BPMN) and service-oriented architecture (SOA) realization approaches, using unified modelling and agile modelling methodologies in a compromising manner, based on the information technology infrastructure library and international standards of information safety.

The work also characterizes the tasks of vital cycle stages of the applied program system management, their modelling and analysis, main concepts of both methodologies (unified modelling and extreme programming) and the concept elaborated on the basis of their compromised interaction, on the base of new Microsoft informational technologies for designing a support system of organized management.

Organized management systems design has been classified into big, medium and small projects and proper solutions have been formed for their program realization in order to work out effective methodological and technological concepts. Unified modelling language (UML)–as an obligatory methodology for fulfilling big and medium projects, with appropriate graphical-analytical CASE instruments and flexible (fast) modelling methods (Agile)–as a technology for small and medium projects, with formation of highly qualified, friendly project-development team and with the principle of programming of gradual, iterative works.

The work suggests the use of ITIL (information technology infrastructure library) for designing applied information systems, the organizational issues for the IT service procession, continuous upgrading and for the fulfilment of business procedures, establishing the effective analysis and management principles of lifetime cycle of IT services and providing the business with iterative and incremental realization.

The unified use of ITIL methodologies and Agile methods significantly improves the implementation quality of program projects. ITIL methodology serves the organizational issues of the integration processes, in this process Agile method has a role of a “catalyst”. Better results shall be achieved by their correlation, from the synergy point-of-view, than when used separately.

The dissertation gives the details of business-processes supporting IT services and the models of business processes for “provision”, which have been designed

with the help of unified modelling language (UML), namely in the form of its activity diagrams. These are the processes supporting incidents, problems, configurations, changes and realization management, as well as management and provision processes of service levels, capacities, continuity, access, finances and safety management.

Improvement of the management processes of the applied programs system projects according to their lifetime stages and from the point-of-view of realizing architectural programs oriented to services, has been practically carried out within the certain scientific projects in the department of automated systems management at the Technical University. The following has been suggested: “Management of the integration processes in the between bank information system with service oriented architecture”, “Management of program systems for the projection and design of multi-media application”, “Modelling of business processes of tax disputes at the revenue service with UML technology and on the basis of mass service methods and their program realization”. Relevant results are also presented in the work.

The work also provides contemporary methods of automated designing of IT infrastructure and their concrete instrumental means. The issues of using Microsoft production - System Centre 2012 Service Manager are also discussed, together with its structure and functions.

For the spheres of investigation, such as between bank integrated informational system, multi-media electronic election system and the system of tax dispute processes, conceptual, logical and physical structure modelling and design of distributed database was carried out by means of CASE instruments (for example: Natural ORM Architect). Namely, for the concrete problematic sphere “the multitude of facts” (knowledge about the management object) has been determined, based on which an object role model (ORM– a diagram 1 level of conceptual model) has been projected in the automated regime. Then on its basis essence relations model (ERM diagram 2 level of conceptual model) using NORMA package was designed. Finally, we carried out the automated generation of DDL file from the ERM diagram and inserted it into the Ms SQL Server database system.

For the distributed applied information systems, based on the examples of between banks, electronic election and tax dispute business processes, the Petri colour network imitational models have been elaborated. It allowed us to investigate the processes of handing over/exchanging IT services, messages and other informational units between client-server (client-provider) blocks in the imitation regime. The graphical schemes of the Petri colour networks (CPN) and the assessment tables and temporary diagrams as a result of investigations are presented.

The issues processed theoretically in the forms of models and methods are realized in the program Ms Visual Studio.NET Framework 4.0, in the integrated environment - C#, HTML, XAML, MsSQL Server packages, using Ms Visio and Natural ORM Architect CASE instruments.