

**პროგრამული სისტემების სასიცოცხლო ციკლის პროცესის მოდელირება
უნივერსალური და ექსტრემალური პროგრამირების პრინციპების
კომპრომისული გადაწყვეტილი**

გია სურგულაძე¹, მიხეილ გულიტაშვილი², ია კაკულია¹,
გიორგი ჩერქეზიშვილი¹, ივანე ჯავახიშვილი¹

1-საქართველოს ტექნიკური უნივერსიტეტი,
2-საქართველოს საპატრიარქოს ქართული უნივერსიტეტი

რეზიუმე

განიხილება გამოყენებითი სფეროს მართვის ავტომატიზებული სისტემების დაპროექტებისა და დეველოპინგის პროცესების დიაგნოსტიკის და IT-კონსალტინგის ამოცანები ობიექტ-ორიენტირებული მიდგომის საფუძველზე. ყურადღება გამახვილებულია პროგრამული პროექტების მენეჯმენტის საკითხებზე, უნიფიცირებული მოდელირების ენისა და ექსტრემალური პროგრამირების პრინციპების და მეთოდების კომპრომისულ გამოყენებაზე. შემოთავაზებულია პროგრამული სისტემების სასიცოცხლო ციკლის ბიზნეს-პროცესების მოდელირების აქტიურობის დიაგრამები და შესაბამისი თვალსაზრისით რეალიზებული პრაქტიკული ექსპერიმენტების შედეგები MsVisual_Studio.NET /Enterprise Architect/Ms Visio გარემოში.

საკვანძო სიტყვები: პროგრამული პროექტების მენეჯმენტი. პროგრამის სასიცოცხლო ციკლი. ბიზნეს-პროცესები. ობიექტ-ორიენტირებული მოდელირება. UML. ექსტრემალური პროგრამირება.

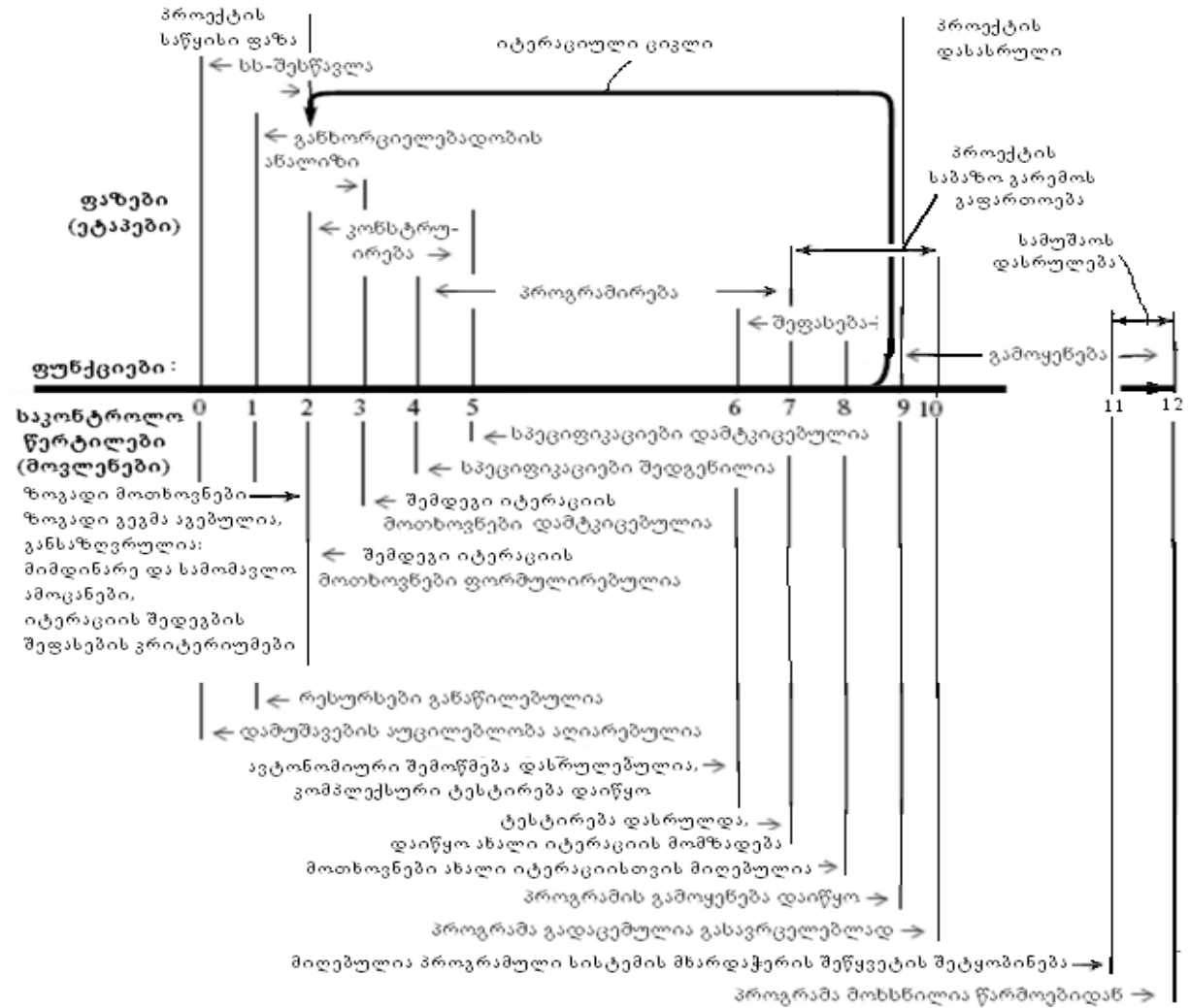
1. შესავალი

გამოყენებითი სფეროს სახით განვიხილავთ კორპორაციულ ობიექტებს, რომელთაც მიეკუთვნება სახელმწიფო ან კერძო სტრუქტურათა საწარმოო, ორგანიზაციული, საბანკო-საფინანსო და სხვა ტიპის იურიდიული სუბიექტები. მათი ფუნქციონირების კეთილდღეობა, კონკურენციის მძაფრ პირობებში, ბევრად და მოკიდებული მართვის აპარატის მოქნილობასა და საიმედოობაზე. რთული ბიზნეს-პროცესების კორპორაციული დაგეგმვისა და ოპერატიული მართვის მექანიზმების ეფექტური ორგანიზებითა და მუდმივი სრულყოფით შესაძლებელი ხდება ამ ობიექტების სასიცოცხლო ციკლის განხორციელება, რაც უდავოდ აქტუალურია. კორპორაციული ობიექტების მართვის ავტომატიზებული სისტემის (პროგრამული სისტემის) შექმნა მოიცავს IT-კონსალტინგის: დიაგნოსტიკური ანალიზის, ექსპერტული შეფასებების, ბიზნეს-პროგრამების დაგეგმვის, მათი განხორციელების ორგანიზების, ფაქტ-შედეგების აღრიცხვის, ეკონომიკური ანალიზისა და შეფასების, ობიექტზე ეფექტური ზემოქმედების მმართველი გადაწყვეტილების მიღების პროცესების ხელშეწყობი მექანიზმების შემუშავებას და მათ კომპიუტერულ რეალიზაციას [1,2].

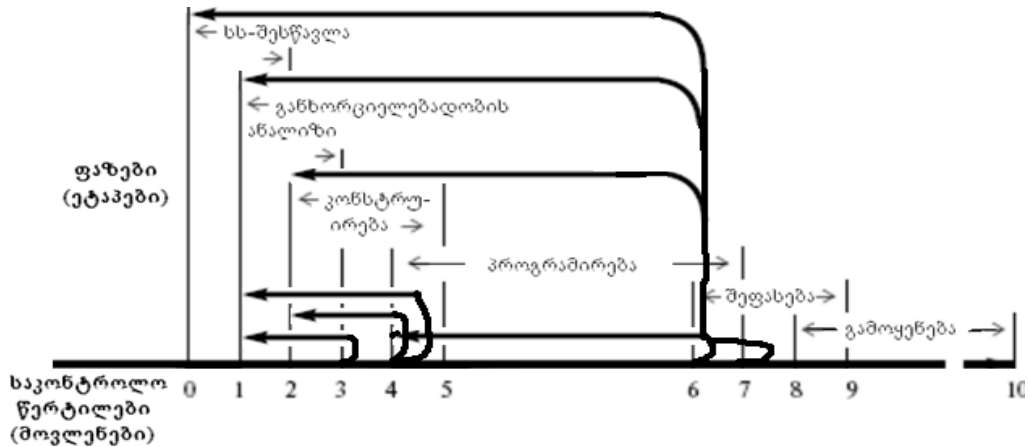
ასეთი ობიექტების მართვის მექანიზმების მოდელი საკმაოდ რთულია და მიეკუთვნება ძნელად ფორმალიზებად ღიდი სისტემების კლასს. მათი აგებისა და ეფექტური გამოყენებისათვის მიზანშეწონილია არაერთგვაროვანი, რაოდენობრივი და ხარისხობრივი მეთოდების კომპლექსური გამოყენება. კერძოდ, ერთის მხრივ, კოვნიტური მოდელის ასაგებად ექსპერტულ შეფასებათა სხვადასხვა მეთოდების ინტეგრირებული გამოყენება, და მეორეს მხრივ, ბიზნეს-პროგრამების დაგეგმვის პროცესების ოპტიმიზაცია და კვლევა პეტრის ქსელებისა და მასობრივი მომსახურების სისტემების თეორიის საფუძველზე [3,4].

სრულყოფილი და საიმედო, მოქნილი პროგრამული უზრუნველყოფის (Software Engineering) სწრაფად დაპროექტება, რეალიზაცია, დანერგვა და შემდგომი თანხლება სისტემის დაკვეთ ორგანიზაციაში მეტად მნიშვნელოვანი ამოცანაა და მისი ეფექტურად გადაწყვეტა ბევრად და მოკიდებული როგორც საპროექტო-დეველოპმენტის გუნდის შემადგენლობასა და გამოცდილებაზე, ასევე IT-ინფრასტრუქტურასა და CASE-ინსტრუმენტებზე.

ხშირად შეუძლებელია სრულყოფილი და საიმედო სისტემების აგება „სწრაფად“. ობიექტ-ორიენტირებული დაპროგრამების მეთოდი, რომელიც უნიფიცირებული მოდელირების ენის (UML) საშუალებით დამკვიდრდა, უნივერსალურია და მისი გამოყენებით პროგრამის სასიცოცხლო ციკლი მოითხოვს მისი აუცილებელი ეტაპების იტერაციულ განვითარებას [5]. 1-ა ნახაზზე ნაჩვენებია ეს ეტაპები განტერის მოდელის საფუძველზე იტერაციული ბიჯებით [6]. შედარების მიზნით, 1-ბ ნახაზზე მოცემულია იგივე ეტაპები ტრადიციული მეთოდისთვის. პროგრამული სისტემების აგების ბიზნეს-პროცესების კვლევა და მოდელირება სწორედ ამ დიაგრამების საფუძველზე ტარდება.



ნახ.1-ა. განტერის სასიცოცხლო ციკლის მოდელი იტერაციით ობიექტ-ორიენტირებული დაპროგრამების მეთოდისთვის



ნახ.1-ბ. განტერის სასიცოცხლო ციკლის მოდელი იტერაციით ტრადიციულ დაპროგრამების მეთოდისთვის

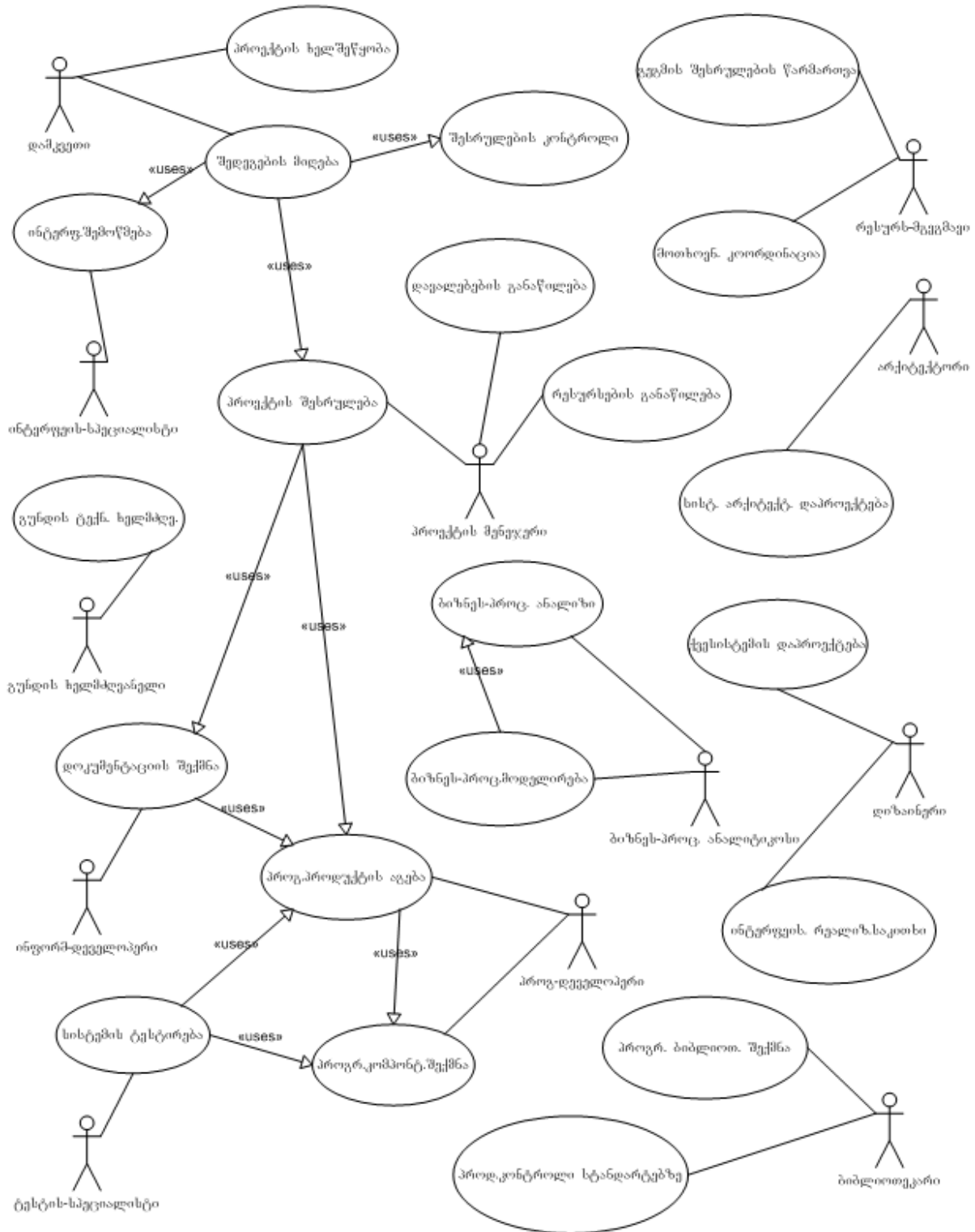
1 ა-ბ ნახაზების შედარებით ცხადი ხდება, რომ პროგრამული სისტემის საკონტროლო წერტილებში (0-12), ეტაპების მიხედვით ხორციელდება იტერაციული სამუშაოები (დაბრუნება უკანა წერტილებში განმეორებითი პროცედურების ჩასატარებლად), სისტემის ფუნქციონალობის სისრულის დაზუსტების ან გაფართოების მიზნით.

პროგრამული სისტემის რეალიზაციის სისწრაფე (დროითი ფაქტორი - ვადები) და სისრულე (იგულისხმება ფუნქციონალური, რეალიზაციის და საინტერფეისო, ხარისხობრივი ასპექტები) საყურადღებო შეზღუდვებია, რომელთა გათვალისწინება შესრულების პროცესში მეტად მნიშვნელოვანია და განაპირობებს რეალიზაციის მეთოდების და ინსტრუმენტების გამოყენების პირობებსაც. ამ კონტექსტში, იტერაციული დაპროგრამების ტრადიციული და ობიექტ-ორიენტირებული მეთოდების გვერდით საჭიროა განვიხილოთ ექსტრემალური დაპროგრამების მეთოდიც (XP, agile programming) [7,8]. მისთვის დამახასიათებელია დროითი და ფუნქციონალური სისრულის კომპრომისული გადაწყვეტა, რომლის დროსაც მოცემული იტერაციისთვის განიხილავენ იმ მინიმალური რეალიზაციის შესაძლებლობას, რაც უზრუნველყოფს სისრულეს და ფუნქციონალურ შეკრულობას. ექსტრემალური პროგრამირების მეთოდის სასიცოცხლო ციკლის მოდელში ძირითადი ყურადღება მახვილდება საპრობლემო ამოცანის სწორად ჩამოყალიბებაში დამკვეთის ბიზნეს-ანალიტიკოსთან ერთად, ნაკლებად იხარჯება დრო უნივერსალური დიაგრამების აგებასა და საანგარიშო დოკუმენტაციის გაფორმებაზე, და რა თქმა უნდა, ხდება ძირითადი ეტაპების (კონსტრუირება-დაპროგრამება) ფაზათა შერწყმა. აქედან გამომდინარე, პროგრამული სისტემის მენეჯერი, კონკრეტული პროექტის ამოცანებისა და მოთხოვნების შესაბამისად, უნდა განსაზღვრავდეს როგორც პროგრამირების მეთოდის, ეტაპთა ფაზების და იტერაციათა მოთხოვნების შერჩევა-ფორმირებას, ასევე მუშა გუნდის შემადგენლობას.

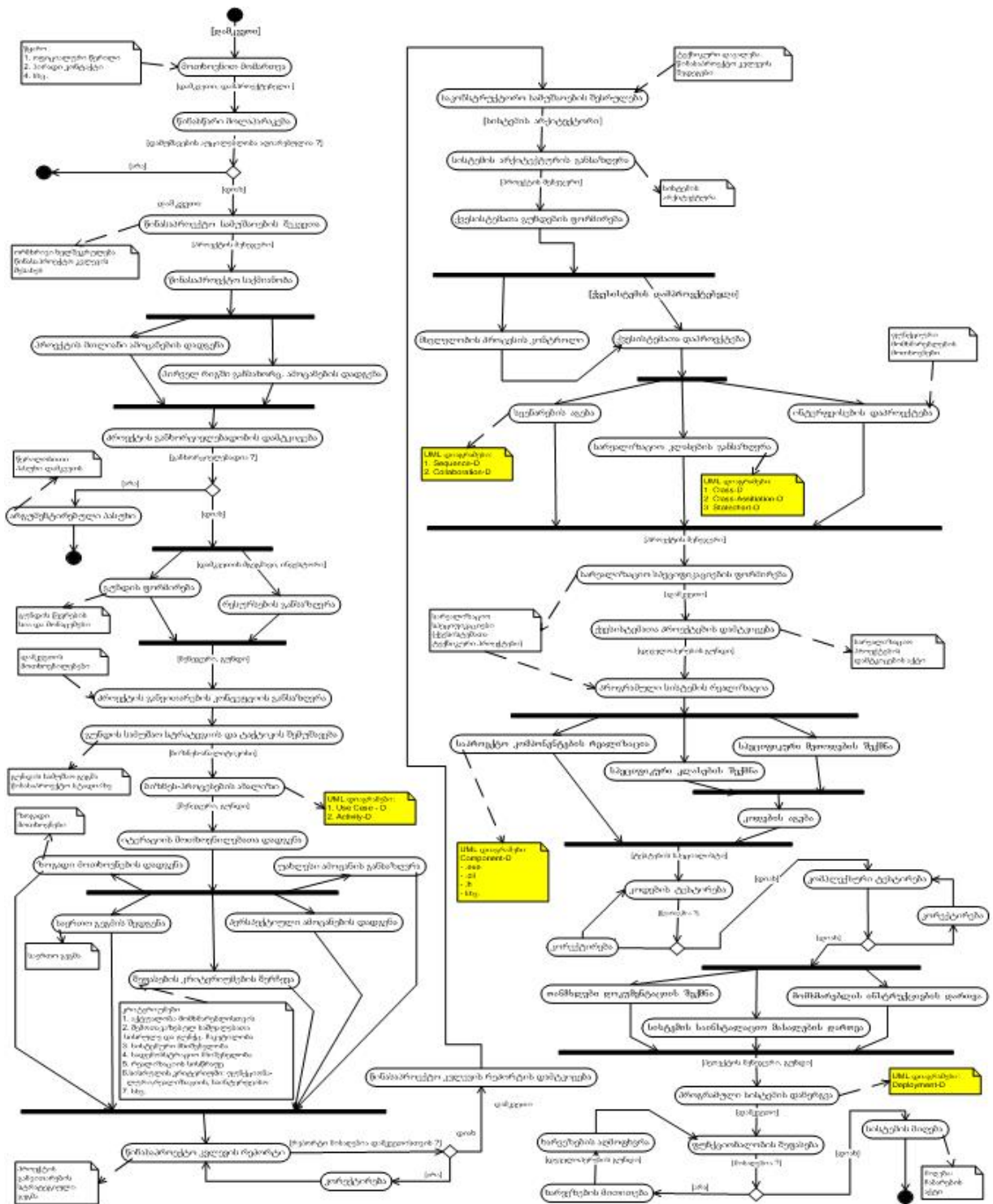
2. ძირითადი ნაწილი

პროგრამული სისტემების პროექტის მენეჯერისთვის მნიშვნელოვანია პროდუქტის სასიცოცხლო ციკლის ბიზნეს-პროცესების დეტალური ანალიზი, მის ცალკეულ ეტაპებზე ეფექტური მართვის განსახორციელებლად. 2-ა ნახაზზე მოცემულია პროგრამული პროექტების შესრულების ერთიანი პროცესის ზოგადი მოდელი, აგებული UML-ის UseCase დიაგრამის სახით. მოდელზე ასახულია ამ პროცესში მონაწილე როლები (დამკვეთი, პროექტის მენეჯერი, ბიზნეს-პროცესების სპეციალისტი, სისტემის არქიტექტორი, დეველოპერი-პროგრამისტი, ტესტირების სპეციალისტი და სხვ.) და მათი ფუნქციები. 2-ბ ნახაზზე კი ილუსტრირებულია შესაბამისი ბიზნეს-პროცესების აქტიურობათა დიაგრამა. აღწერილი ზოგადი მოდელი მენეჯერის მიერ ადაპტირდება კონკრეტული

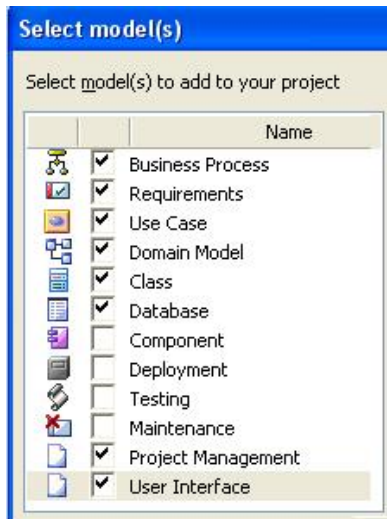
პროექტისათვის და ღებულობს კერძო სახეს. მთლიანი სისტემის ბიზნეს-პროცესების ანალიზი საფუძველია სისტემის არქიტექტურის და საერთოდ, IT-ინფრასტრუქტურის დასადგენად.



ნახ.2-ა. პროგრამული სისტემის რეალიზაციის Use Case დიაგრამა



ნახ.2-ბ. პროგრამული სისტემის რეალიზაციის Activity დიაგრამა

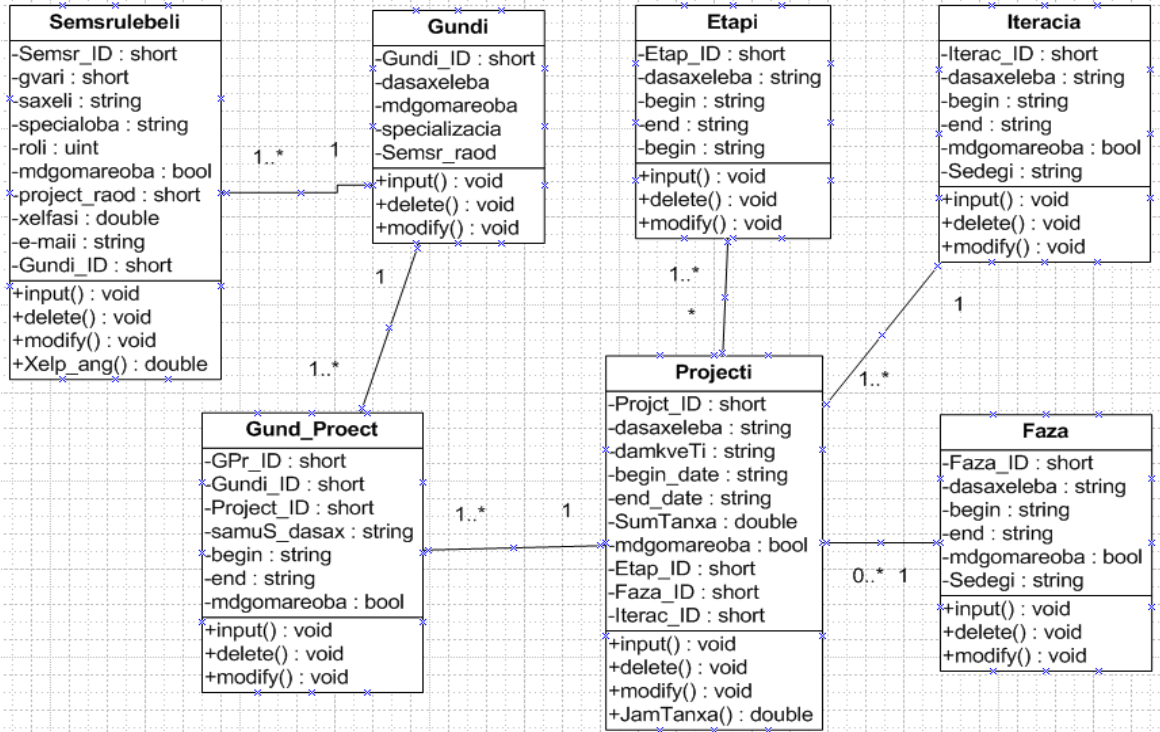


ნახ.3

პროგრამული სისტემის პროექტის მიზნების, საწყისი პირობებისა და მოთხოვნებითა გათვალისწინებით მენეჯერის მიერ შეირჩევა კომპრომისული გადაწყვეტილება და, მაგალითად, UML/2-ის ინსტრუმენტში Enterprise Architect ბიზნეს-პროცესების მოდელირების, კონსტრუირების, დეველოპმენტის და ტესტირების ეტაპების დასაგეგმად ჩაირთვება შესაბამისი “ჩეკ-ბოქსები” (ნახ.3).

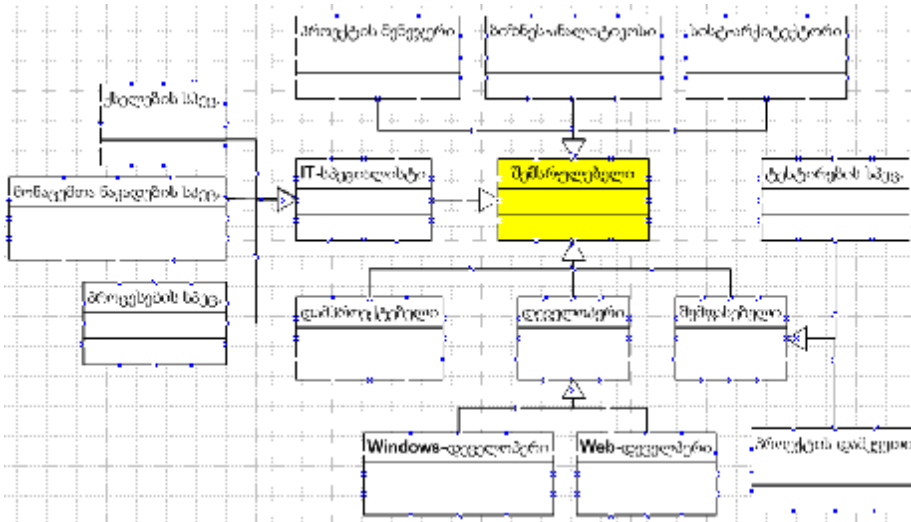
დიდი პროექტებისათვის, რომელშიც რესურსები და დროითი ფაქტორები, შედარებით კრიტიკული არაა, ხდება ობიექტ-ორიენტირებული მიდგომის ყველა ეტაპის და ფაზის გამოყენება შესაბამისი საკონტროლო წერტილების აუცილებელი მონიტორინგით და რეპორტებით (ნახ.1-ა). ექსტრემალური მიდგომის მეთოდის გამოყენებისას, მცირე ან საშუალო პროექტებში, სადაც რეალიზაციის დრო კრიტიკულია, ძირითადი ყურადღება ამოცანის სწორად ჩამოყალიბების, დეველოპინგის და ტესტირების ეტაპებზე გადადის.

მე-4 ნახაზზე მოცემულია პროგრამული პროექტების მენეჯმენტის სისტემის კლასთა-ასოციაციის დიაგრამის ფრაგმენტი, რომელშიც ასახულია ინფორმაცია პროექტების, გუნდების, შემსრულებლების, იტერაციების, ეტაპებისა და ფაზების შესახებ, აგრეთვე მათი ურთიერთკავშირები და თითოეულის მდგომარეობა. ასეთი ინფორმაცია სასარგებლოა პროექტების მენეჯმენტის განსახორციელებლად, კერძოდ საპროექტო სამუშაოების გასანაწილებლად შემსრულებლებს შორის მათი დატვირთვისა და კალენდარული ვადების გათვალისწინებით, რომელიც წყდება, მაგალითად, MsProject პაკეტით.



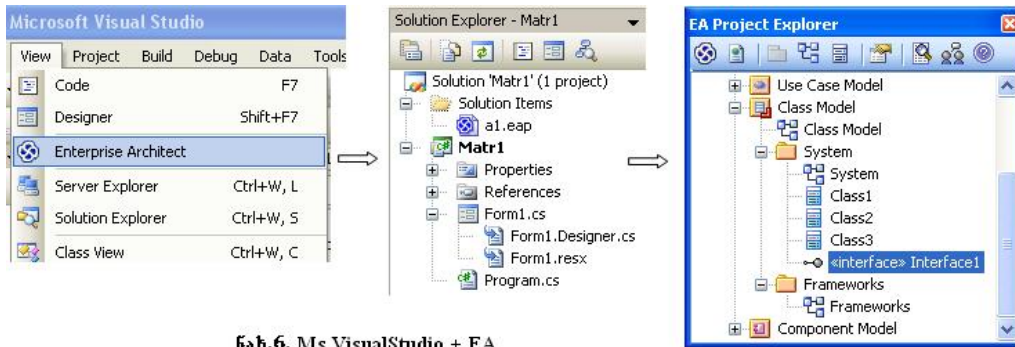
ნახ.4. კლასთა-ასოციაციის დიაგრამის ფრაგმენტი

მე-5 ნახაზზე ილუსტრირებულია პროექტის შემსრულებელთა გუნდის წევრების იერარქიული სქემა მემკვიდრეობითობის (Generalization) კავშირების საფუძველზე. პროექტის მენეჯერი თვითონაც გუნდის წევრია, რომელსაც ეხება გუნდის დაკომპლექტება პროექტის მიზნებიდან, სირთულიდან და განხორციელების ვადებიდან გამომდინარე. იგი გაითვალისწინებს აგრეთვე, საჭიროების შემთხვევაში, ექსტრემალური პროგრამირების პრინციპებსაც [8].



ნახ.5. პროექტის შემსრულებელი: როლების მემკვიდრეობითი კავშირები

UML-დიაგრამების (მაგ., კლასების, აქტიურობათა, მიმდევრობითობის) დაპროექტების შემდეგ საჭიროა კოდის გენერაციის პროცედურების შესრულება. ჩვენს შემთხვევაში შესაძლებელია MDG (Model Driven Generation) გამოყენება. MDG Link for Visual Studio .NET საშუალებას იძლევა Enterprise Architect ინსტრუმენტით ვიმუშაოთ .NET გარემოში და განვახორციელოთ პირდაპირი და რევერსული დაპროექტება (ნახ.6).



ნახ.6. Ms VisualStudio + EA

3. დასკვნა

პროგრამული სისტემების შექმნის თანამედროვე ინტეგრირებული ტექნოლოგიები საშუალებას იძლევა არა მხოლოდ დავაჩქაროთ ხარისხიანი პროგრამული პროდუქტების წარმოება, არამედ განვახორციელოთ მათი სასიცოცხლო ციკლის მონიტორინგი და ვმართოთ მისი პროცესები, რომლებიც დაკავშირებულია კომპიუტერული და ადამიანური რესურსების ეფექტიან ორგანიზებასა და თვით პროდუქტის გამოყენების პერიოდის გახანგრძლივებასთან. შემოთავაზებულია ტრადიციული, CASE და XP მეთოდების პრინციპების ანალიზი და მათი ბიზნეს-პროცესების მოდელური, მენეჯერისა და გუნდის წევრების ფუნქციები და პროექტის შესრულების სტრატეგიული გეგმების ფორმირება კომპრომისული გადაწყვეტილებების საფუძველზე.

ლიტერატურა:

1. ფრანგიშვილი ა., სურგულაძე გ., ვაჭარაძე ი. ბიზნეს-პროგრამების ექსპერტულ შეფასებებში გადაწყვეტილებათა მიღების მხარდამჭერი მეთოდები და მოდელები. სტუ. მონოგრ., თბილისი, 2009
2. სურგულაძე გ., კაშიბაძე მ. ორგანიზაციულ სისტემებში ინფორმაციული რესურსების მართვა. სტუ, თბილისი, 2009
3. სურგულაძე გ., ოხანაშვილი მ., სურგულაძე გრ. მარკეტინგის ბიზნეს-პროცესების უნიფიცირებული და იმიტაციური მოდელირება. სტუ, თბ., სტუ, 2009
4. სურგულაძე გ., გულუა დ. განაწილებული სისტემების ობიექტ-ორიენტირებული მოდელირება უნიფიცირებული პეტრის ქსელებით. სტუ. თბ., 2005
5. Booch G., Jacobson I., Rumbaugh J. Unified Modeling Language for Object-Oriented Development. Rational Software Corporation, Santa Clara, 2006
6. Скопин И.Н. Основы менеджмента программных проектов. www.intuit.ru/department/se/msd. 2004
7. Бек К. Шаблоны реализации корпоративных приложений. Экстремальное программирование: Пер. с англ. М.: Вильямс, 2008.
8. Martin C R, Martin M. Agile Principles, Patterns and Practices in C#. Prentice Hall. 2006. □

MODELING PROCESSES OF LIFE CYCLE OF COMPUTER SYSTEMS WITH THE COMPROMISE DECISION OF PRINCIPLES OF UNIVERSAL AND EXTREME PROGRAMMING

Surguladze Gia, GulitaSvil Mixeil, kakulia Ia,
Cherqezishvili Giorgi, Djavakhishvili Ivane
Georgian Technical University

Summary

Problems of diagnostics and IT-consulting of processes of designing and development of the Management Information System applications on the basis of the object-oriented approach are considered. The main attention is focused on issues of management of program systems, compromise application of principles of universal (UML) and extreme programming (XP). Results of the analysis and modeling of business processes of life cycle of program systems and corresponding practical experiments on platform MsVisual_Studio.NET/Enterprise Architect/Ms Visio are offered.

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ЖИЗНЕННОГО ЦИКЛА КОМПЬЮТЕРНЫХ СИСТЕМ С КОМПРОМИССНЫМ РЕШЕНИЕМ ПРИНЦИПОВ УНИВЕРСАЛЬНОГО И ЭКСТРЕМАЛЬНОГО ПРОГРАММИРОВАНИЯ

Сургуладзе Г., Гулиташвили М., Какулия И.,
Черкезишвили Г., Джавахишвили И.
Грузинский Технический Университет

Резюме

Рассматриваются задачи диагностики и ИТ-консалтинга процессов проектирования и разработки АСУ прикладной сферы на основе объектно-ориентированного подхода. Основное внимание уделяется вопросам менеджмента программных систем, компромиссного использования принципов универсального (UML) и экстремального программирования (XP). Предложены результаты анализа моделирования бизнес-процессов жизненного цикла программных систем и соответствующих практических экспериментов на платформе MsVisual_Studio.NET/Enterprise Architect/Ms Visio.