

ДОДАТОК А  
Програмний код

```
object Form1: TForm1
  Left = 192
  Top = 107
  BorderStyle = bsDialog
  Caption = 'Установка параметров проекта'
  ClientHeight = 168
  ClientWidth = 301
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 8
    Top = 9
    Width = 53
    Height = 13
    Caption = 'Описание:'
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -11
    Font.Name = 'MS Sans Serif'
    Font.Style = []
    ParentFont = False
  end
  object Edit1: TEdit
    Left = 76
    Top = 7
    Width = 213
    Height = 21
    TabOrder = 0
  end
end
```

```
object RadioGroup1: TRadioGroup
  Left = 8
  Top = 40
  Width = 161
  Height = 61
  Caption = 'Задача:'
  Items.Strings = (
    'Квалиметрия'
    'Классификация')
  TabOrder = 1
end
object RadioGroup2: TRadioGroup
  Left = 8
  Top = 103
  Width = 161
  Height = 61
  Caption = 'Статус:'
  Items.Strings = (
    'Частный'
    'Общий')
  TabOrder = 2
end
object BitBtn1: TBitBtn
  Left = 204
  Top = 101
  Width = 81
  Height = 25
  TabOrder = 3
  OnClick = BitBtn1Click
  Kind = bkOK
end
object BitBtn2: TBitBtn
  Left = 204
  Top = 136
  Width = 81
  Height = 25
```

```

    TabOrder = 4
    Kind = bkCancel
end
end

```

### OpenProject

Прототип: int OpenProject (void);

Назначение: Обработчик события выбора пункта меню

Проект|Открыть... Открывает проект, инициализирует переменные CurrentProject, Records, Demands, Table.

Возвращает: 0 – в случае удачного выполнения; !0 – в противном случае.

### CreateProject

Прототип: BOOL CreateProject (void);

Назначение: Обработчик события пункта меню Проект|Создать...

Создает новый проект, в диалоговом режиме инициализирует переменные CurrentProject, Records, Demands, Table.

Возвращает: TRUE – в случае удачного выполнения; FALSE – в противном случае.

### CloseProjectChoice

Прототип: void CloseProjectChoice (void);

Назначение: Реакция на выбор пункта меню Проект|Закрыть. Закрывает открытый проект, очищает переменные CurrentProject, Records, Demands, Table, закрывает файлы.

### SaveProject

Прототип: void SaveProject (char \*);

Назначение: Сохраняет информацию о компонентах модели в соответствующих файлах.

Параметры: имя проекта.

### DeleteProject

Прототип: void DeleteProject (void);

Назначение: Удаляет файлы проекта.

### ModifyDB

Прототип: void ModifyDB (void);

Назначение: В диалоговом режиме производит модификацию перечня свойств объектов и при необходимости запускает механизм модификации баз данных

ModifyDemands

Прототип: void ModifyDemands (void);

Назначение: В диалоговом режиме производит модификацию перечня требований к объектам предметной области.

ModifyClasses

Прототип: void ModifyClasses (void);

Назначение: В диалоговом режиме производит модификацию распознаваемых классов в задаче классификации, при необходимости запускает механизм модификации баз данных.

Приведенные ниже прототипы соответствуют функциям нижних уровней.

OpenProjectFunct

Прототип: int OpenProjectFunct (char \*);

Назначение: Проверяет наличие и корректность файлов проекта и инициализирует соответствующие переменные.

Вызывающие функции: OpenProject.

Параметры: имя проекта

Возвращает: 0 – в случае успешного завершения, иначе – код ошибки.

CreateDemandTemplate

Прототип: BOOL CreateDemandTemplate (void);

Назначение: на основе диалога с пользователем инициализирует переменную Demands (создает требования).

Вызывающие функции: CreateProject.

Возвращает: TRUE – в случае удачного выполнения; FALSE – в противном случае.

CreateProjectClasses

Прототип: BOOL CreateProjectClasses (void);

Назначение: на основе диалога с пользователем создает множество классов.

Вызывающие функции: CreateProject.

Возвращает: TRUE – в случае удачного выполнения; FALSE – в противном случае.

CreateDBStructure

Прототип: BOOL CreateDBStructure (char \*);

Назначение: на основе диалога с пользователем создает множество свойств (инициализирует переменные Records, Table).

Вызывающие функции: CreateProject.

Параметры: имя файла STC.

Возвращает: TRUE – в случае удачного выполнения; FALSE – в противном случае.

UseExistingTable

Прототип: BOOL UseExistingTable (char \*);

Назначение: Инициализирует переменные Records, Table на основе данных существующего файла STC.

Вызывающие функции: CreateProject.

Параметры: имя файла STC.

Возвращает: TRUE – в случае успешного выполнения; FALSE – в противном случае.

CreateSTCDlgProc

Прототип: BOOL FAR PASCAL \_export CreateSTCDlgProc (HWND hDlg, UINT msg, WPARAM wParam, LONG lParam);

Назначение: реализует диалог с пользователем по созданию множества свойств.

Вызывающие функции: ModifyDB, CreateDBStructure.

Параметры: hDlg – дескриптор диалогового окна; msg – код сообщения; wParam, lParam – параметры сообщения.

Возвращает: код выхода из диалога.

CreateTPTDlgProc

Прототип: BOOL FAR PASCAL \_export CreateTPTDlgProc (HWND hDlg, UINT msg, WPARAM wParam, LONG lParam);

Назначение: реализует диалог с пользователем по созданию множества требований.

Вызывающие функции: ModifyDemands, CreateDemandTemplate.

Параметры: hDlg – дескриптор диалогового окна; msg – код сообщения; wParam, lParam – параметры сообщения

Возвращает: код выхода из диалога.

ClassesDlgProc

Прототип: BOOL FAR PASCAL \_export ClassesDlgProc (HWND hDlg, UINT msg, WPARAM wParam, LONG lParam);

**Назначение:** реализует диалог с пользователем по созданию множества классов в задаче классификации.

**Вызывающие функции:** ModifyClasse.

**Параметры:** hDlg – дескриптор диалогового окна; msg – код сообщения; wParam, lParam – параметры сообщения.

## Знімки екранів електронних таблиць із основними результатами досліджень

The screenshots illustrate the iterative process of solving a linear programming problem. Each screenshot shows a spreadsheet with the following components:

- Data Table:** A table with columns: НЮ, МН, СП, ХТ, Фикст. ДО, Кол-во. The values in the 'Кол-во' column are 5, 19, 15, 19, and 23.
- Constraint Matrix Y [j\*k]:** A 5x5 matrix with values:
 

0	5	0	0	5
10	0	8	0	18
0	0	0	0	0
0	0	0	0	0
0	0	0	35	35
10	5	8	35	
- Constraint Matrix X [i\*j]:** A 5x5 matrix with values:
 

5	0	0	0	5
0	4	0	0	4
0	0	0	0	7
0	0	0	0	9
0	14	0	0	15
0	0	0	0	8
0	0	0	0	10
5	18	0	0	35
- 'Поиск решения' Dialog:** Shows the objective function (ЦФ) as 0,396552. The 'Равной:' section is set to 'максимальному значению'. The 'Имена ячеек:' section lists \$B\$3:\$B\$7, \$I\$12:\$I\$18. The 'Ограничения:' section lists constraints like \$F\$9 = \$H\$7, \$I\$12:\$I\$18 = целое, etc.

The three screenshots show the progression of the solution. The objective function value increases from 0,396552 to 0,362069, then to 0,382759, and finally to 0,388793. The constraint matrix X also changes, reflecting the updated solution.

ДОДАТОК Б  
Слайди презентації

Харківський національний університет  
радіоелектроніки

Атестаційна робота магістра

Дослідження алгоритмів  
багатокритеріальної оптимізації з нечіткими  
компонентами

Виконав  
ст. гр. ІПЗмзд-18-1  
Забіяка В.С.

Науковий керівник  
проф. Шостак І.В.

1

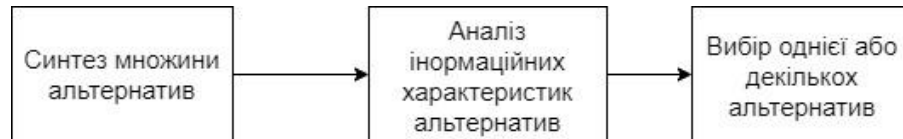
## Мета роботи

- Мета дослідження полягає в досягненні максимальної ефективності технічних систем шляхом розробки алгоритмів порівняльного аналізу на основі обробки інформації його характеристичних параметрів та при пропонувані нечітких вимогах, а також дослідженні методів оптимального вибору в системах підтримки прийняття рішень

1

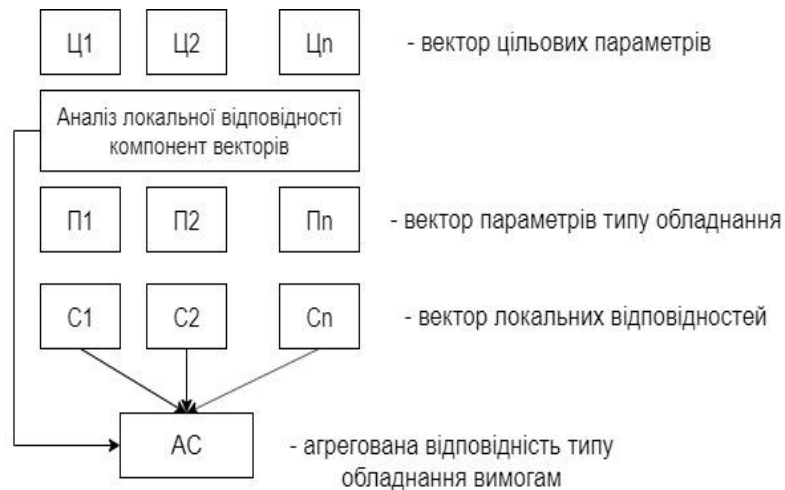
2

## Схема основних процедур прийняття рішень



3

## Схема реалізації блоку параметричного аналізу устаткування вимогам



4

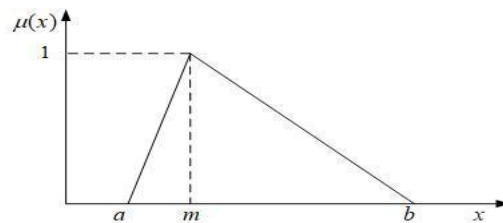
## Нечітка змінна

- характеризується трійкою  $(T, U, \mu(x))$
- проміжок  $[a; b]$  називають носієм нечіткої множини

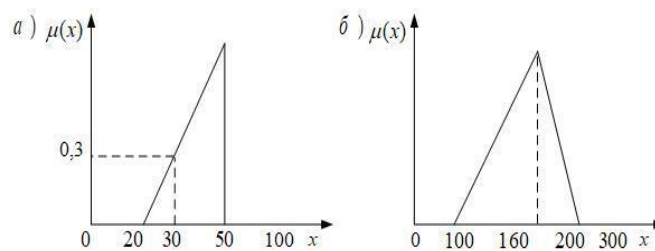
$$\mu(x) = \begin{cases} \frac{x-a}{m-a}, & \text{если } a \leq x \leq m; \\ \frac{b-x}{b-m}, & \text{если } m \leq x \leq b; \\ 0, & \text{иначе.} \end{cases}$$

5

## Функція приналежності



### Функції приналежності нечітких локальних вимог



6

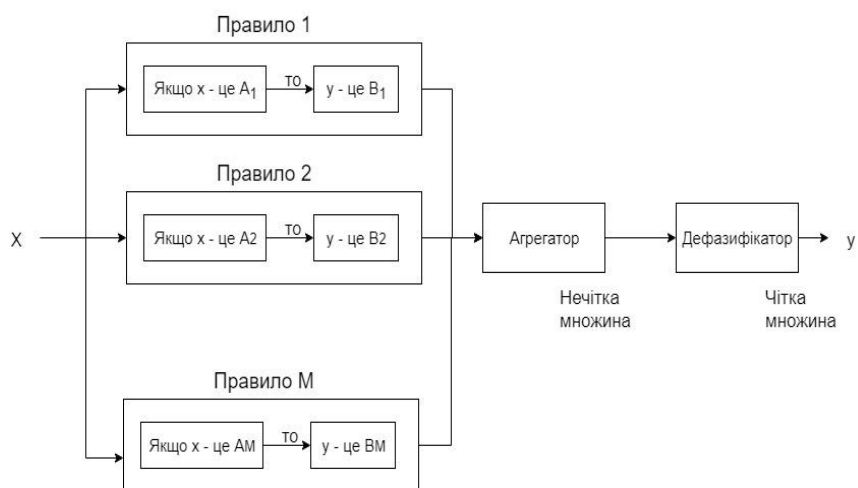
## Нечітке продукційне правило,

- зазвичай на підставі експертних висновків:

«ЯКЩО  $A$ , ТО  $B$ » = «ЯКЩО  $R_{\text{пос}} \in T_{\text{пос}}$ , ТО  $R_{\text{зам}} \in T_{\text{зам}}$ »

7

## Система нечіткого логічного виводу



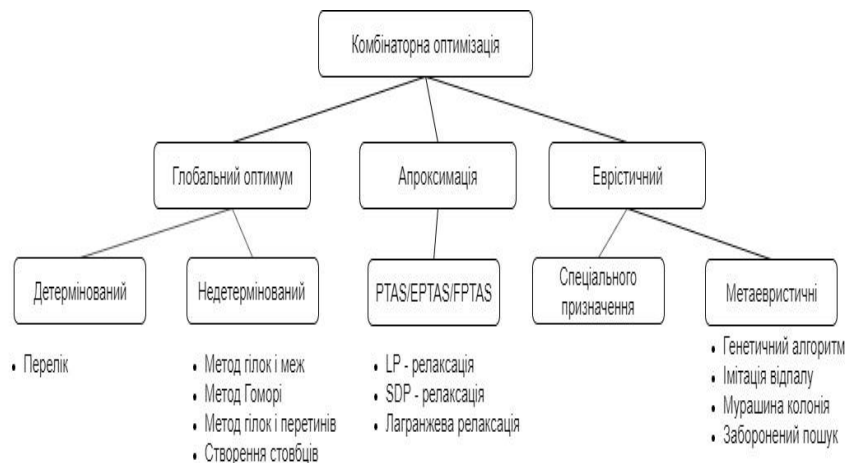
8

## Постановка задач дослідження

- Проведений аналіз особливостей представлення інформації про вибір складних об'єктів і вимог до їхнього вибору, а також умов вибору й релевантних математичних методів прийняття оптимальних рішень й показав, що для досягнення поставленої мети необхідно розв'язати наступні завдання:
- аналіз методик моделювання нечітких вимог до вибору об'єктів й оцінки сукупної по параметрах відповідності типів альтернативного набору об'єктів пропонованим вимогам на основі експертизи характеристичних параметрів в термінах нечітких виразів;
- розробка алгоритму обчислення агрегованої відповідності об'єктів по сукупності локальних відповідностей характеристичних і цільових параметрів з використанням багатомірних функцій приналежності;
- чисельна апробація запропонованих методів і алгоритмів на прикладах вибору устаткування для виробничих процесів.

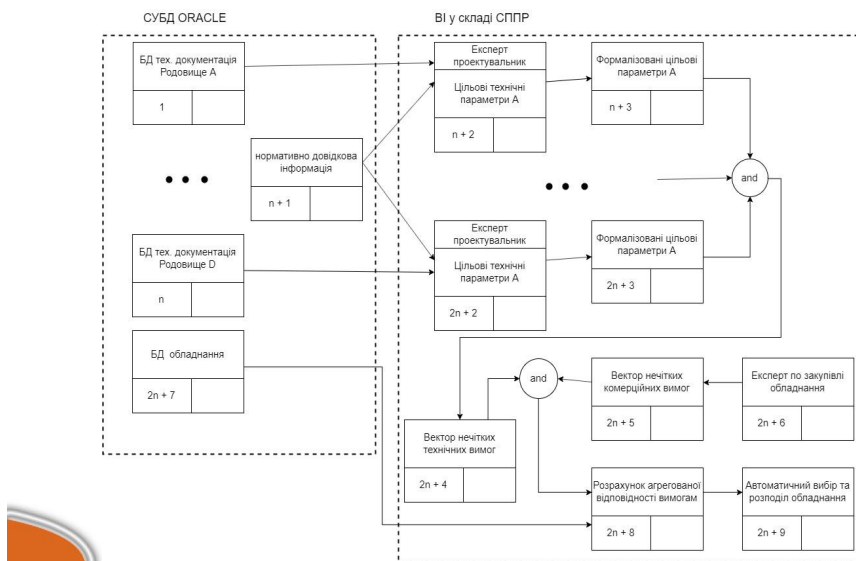
9

## Класифікація методів комбінаторної оптимізації (одержання цілочисельних рішень )



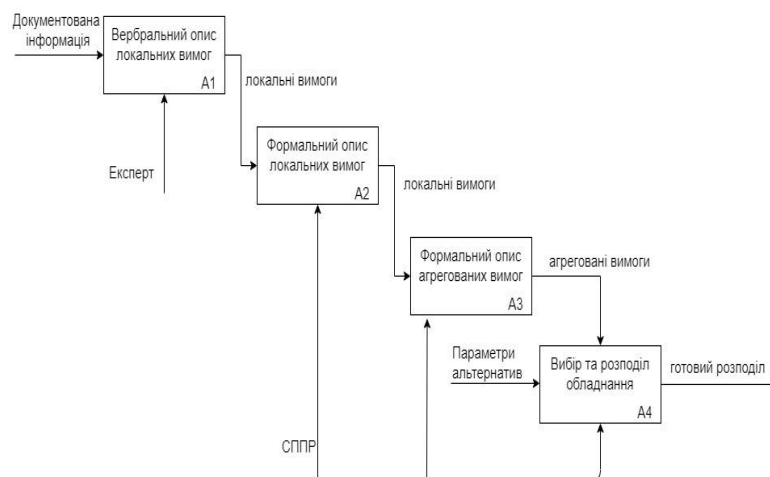
10

## Структурна модель інформаційних процесів у складі системи SAP ERP



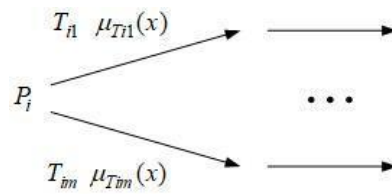
11

## Структурна модель обробки інформації при формуванні та аналізі вимог до вибору об'єктів



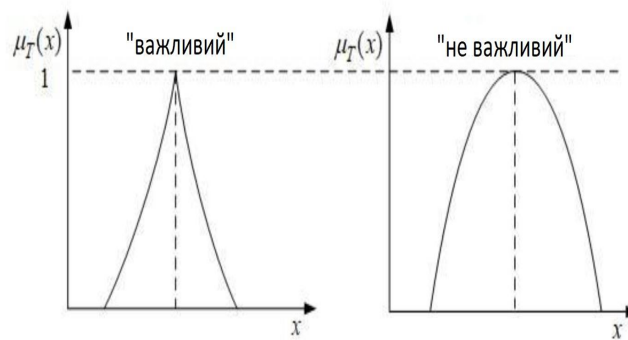
12

## Схема завдання нечітких вимог для $i$ -го цільового параметра



13

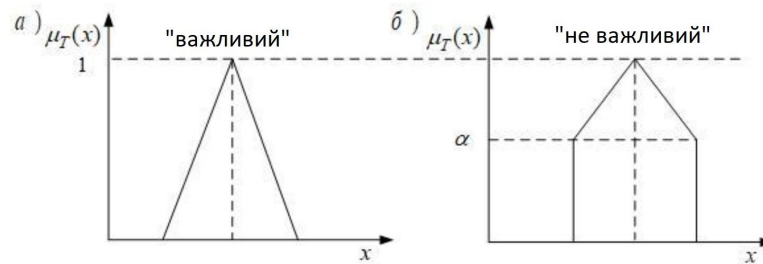
## Функції приналежності адекватні поняттям «важливий» параметр і «не важливий» параметр



14

## Апроксимовані функції

- приналежності адекватні поняттям «важливий» параметр і «не важливий» параметр



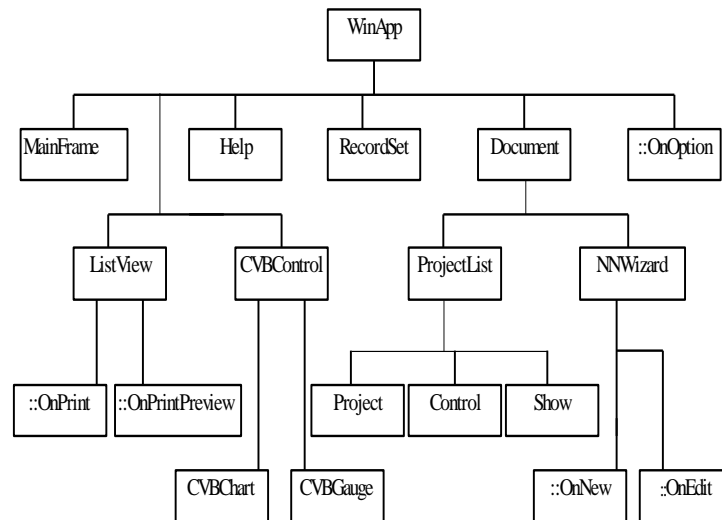
15

## Схема алгоритму обчислення локальних і агрегованих відповідностей альтернативних типів пропонуваним вимогам



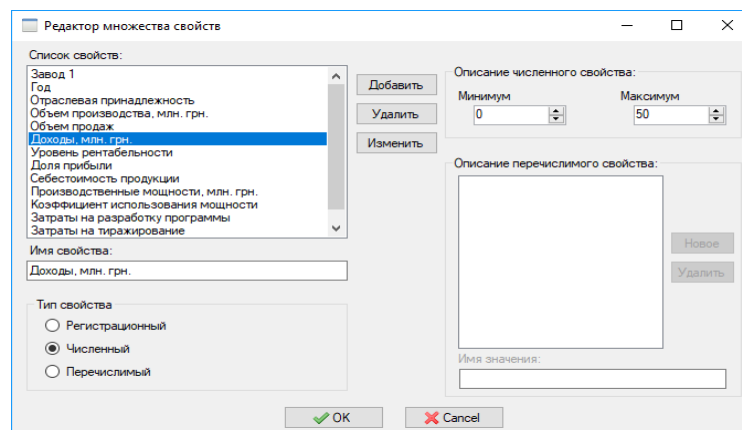
16

## Структурна модель програми



17

## Визначення множини властивостей об'єктів



18

## Вікно редактора вимог

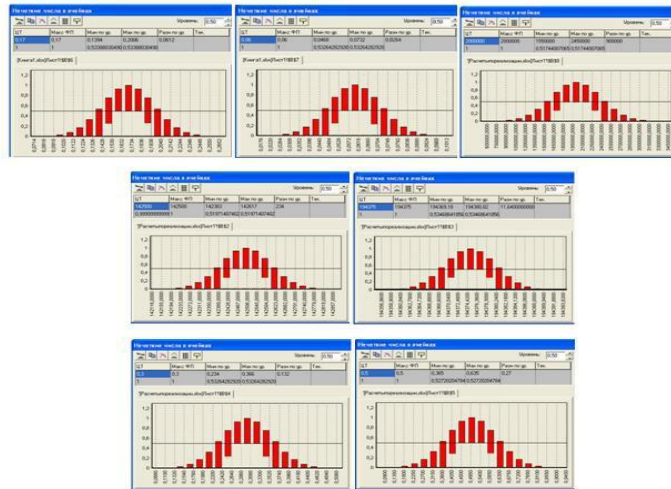
19

## Чисельне подання вхідних параметрів

1	1 альтернатива	2 альтернатива	3 альтернатива	
2	Начальные инвестиции	142500	142500	184030
3	Затраты на внедрение	194375	272500	232500
4	Время на внедрение	0,3	0,5	0,6
5	Планируемый срок окупаемости	0,5	0,7	0,9
6	Ставка дисконтирования	0,17	0,17	0,17
7	Ставка инфляции	0,06	0,06	0,06
8	Объем чистой прибыли	2000000	2000000	2000000

20

## Подання вхідних параметрів, що завдане у вигляді функцій приналежності НМ



21

## Нечіткі значення показників ефективності

1.	Індекс рентабельності	5,33		0,2
2.	Строк окупності	0,23		0,3
3.	Повернення інвестицій	10,11		0,1

22



## Висновки

Обґрунтована можливість використання нечітких логічних виразів і операцій нечіткої логіки для формалізованого опису експертних вимог, що забезпечило одержання чисельних оцінок локальних відповідностей параметрів альтернативних типів заданим вимогам.

- Запропонований алгоритм розрахунку агрегованої по параметрах відповідності типів альтернативного устаткування, заснований на обчисленні значень модифікованих трикутних функцій приналежності, що запобігає невиправдане заниження оцінок відповідності устаткування вимогам.
- Проведена чисельна апробація розроблених моделей і алгоритмів, яка показала їхню працездатність і можливість значного підвищення ефективності функціонування виробничих технічних систем. Застосування методів і алгоритмів вибору устаткування технічних систем з використанням нечітких технічних і комерційних цільових характеристик в інформаційно-керуючій системі.

ДОДАТОК В  
Апробація результатів роботи

# Подано тези на Молодіжний форум «Радіоелектроніка і молодь в XXI сторіччі»

## ДОСЛІДЖЕННЯ АЛГОРИТМІВ БАГАТОКРИТЕРІАЛЬНОЇ ОПТИМІЗАЦІЇ З НЕЧІТКИМИ КОМПОНЕНТАМИ

Завіска В.С.  
Науковий керівник - проф. Шостак І.В.  
Харківський національний університет радіоелектроніки  
(61100, Харків, пр. Науки, 14, каф. Програмної інженерії,  
т. (037) 702-14-64)  
E-mail: i.v.shostak@gmail.com

In the work the method of estimating the maturity (perfection) of the projects performed in the testing of software systems is developed. The method is based on TQM's five-level maturity model. Also, the use of mathematical model, the basic process of testing software data processing systems under the conditions of limited resources, as well as several methods aimed at improving the quality of software products, have been developed and substantiated.

До головних напрямків програмної інженерії відноситься завдання автоматизація процесів контролю якості ПС, зокрема процесу тестування. Неухильне підвищення якості програмних продуктів - основна мета програмної інженерії та предмет турботи розробників ПС.

Відповідь на питання, як підвищити конкурентоспроможність українських програмних продуктів, як знизити ризик провальної якості, як досягти балансу серед продуктивності, швидкості та вартості - цілі проекту, протягом останніх років намагається знайти відповідь не лише керівникам організацій-розробників ПС і менеджери проєктів, але і вчені, і спеціалісти, що використовують програмні продукти низької якості. Наприклад поставивши програмний продукт низької якості - це завжди збитки користувачів не лише матеріальні і фінансові витрати, але і психічна травма. Ці проблеми в свою чергу, викликають відображення на конкурентоспроможності організацій-розробників програмних продуктів. Для забезпечення необхідного рівня якості ПС в міжнародній практиці знаходять застосування два підходи: продукто-орієнтований і процесно-орієнтований. В першому випадку робиться на оцінку якості шляхом тестування готового програмного продукту. Цей підхід базується на припущенні, що чим більше знань і усвідомлено дефіцитів в ПС при тестуванні, тим вище його якість.

Тестування - важливий етап розроблення ПС, особливо, з одного боку, вимагає значних витрат на проведення, а з іншого - робить великий внесок у його якість. Чим теоретично доведено неможливість вичерпного тестування та велику повільність вартість витрат часу в ПС, потрібен чітко визначений та ефективний процес менеджера, базований на умовленні об'єктивних рішень щодо тривалості та вартості тестування для досягнення необхідного рівня довіри до якості ПС.

Методи визначення кількісних критеріїв завершення тестування та керування процесом тестування в використання кількісних вимірів не мало використовуються в проєктах створення ПС, що призводить до того, що якість та надійність залишаються не передбачуваними. Лише за наявності достовірної та своєчасної інформації щодо стану ПС, після ризиків і можливих витрат часу вилучити можна бути забезпечене ефективне виконання процесу тестування.

Метою роботи є проведення комплексного дослідження з інженерії тестування ПС, оброблення даних, формування ефективної стратегії тестування програмних систем, спрямованої на зникнення ризику вилучити час виконувати. Для цього в роботі розв'язуються наступні задачі:

дослідження сучасних підходів до процесу тестування ПС;  
аналіз існуючих моделей надійності ПС, розроблення алгоритмів та програм їх реалізації;  
побудова моделі визначення оптимального часу тестування модулів ПС з урахуванням ризику вилучити.

Визначення структури базового процесу, що регламентує всі дії з підготовки, проведення та оцінювання результатів тестування, та розроблення методик виконання процесу тестування ПС оброблення даних.

проєктування та реалізації програмного комплексу підтримки інженерії тестування.

аналіз сучасних підходів до визначення рівня цілісності процесу тестування ПС та розроблення методик оцінювання процесу тестування, впровадження запропонованих підходів, моделей та методик інженерії тестування в проєкти з розроблення ПС оброблення даних та опис результатів впровадження запропонованих моделей оцінювання оптимального часу тестування та методу оцінювання ризику вилучити програмних модулів.

З метою визначення ефективності впровадження базового процесу тестування був розроблений метод оцінювання цілісності (досконалості) виконуваних у проєктах дій з тестування ПС. Цей метод ґрунтується на використанні моделі цілісності TQM.

Впровадження моделі аналізу виконуваних в конкретній інформаційно-аналітичній системі підтримки управління управлінських рішень, яка має складатися з програмних комплексів, об'єднаних діючим процесом оброблення даних. В ході практичної реалізації аналіз впровадження системи показав, що в більшій кількості інформації найбільший внесок у ризик її вилучити робить ПК контролю виведення даних до БД. ОБЛАСНІ дані використовуються та однократно функціонують на 10 робочих місцях. Для 9-ти модулів цього ПК були виконані оцінки ризику вилучити та часу тестування.