





ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ,  
СИМВОЛІВ І СКОРОЧЕНЬ

БД	База Даних
ЖЦ	Життєвий Цикл
ІС	Інформаційні Системи
ПВК	Повторно використовуваний компонент
ПС	Програмні Системи
ЗС	Conception Construction Context Model
ANFIS	Adaptive Network Based Fuzzy Inference System
CAESE	Computer Aided Empirical Software Engineering
CASE	Computer Aided Software Engineering
COCOMO	COConstructive COost Models
GQM	Goal – Question – Metrics
RDD	Requirement Driven Design
RDL	Requirement Declaration Language
UML	Unified Modeling Language
XML	Extended Markup Language

## АНОТАЦІЯ

Тема дипломної роботи: “ Дослідження методів і засобів експертного оцінювання властивостей наслідуваного програмного забезпечення” // Дипломна робота // Смучок Іван Михайлович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем та програмної інженерії, група СІм-61 // Тернопіль, 2015 // с. – 123 , рис. – 35 , табл. –9 , аркушів А1 –8 , додат. –4 , бібліогр. –46 .

Ключові слова: ПОВТОРНЕ ВИКОРИСТАННЯ, НАСЛІДУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ОЦІНЮВАННЯ, ЕКСПЕРТ.

Основними задачами дипломної роботи є дослідження сучасного стану методів і засобів експертного оцінювання властивостей повторно використовуваних компонентів наслідуваного програмного забезпечення, виявлення їх недоліків, розробка методу експертного оцінювання властивостей повторно використовуваних компонентів для підвищення ефективності розробки програмного забезпечення, обґрунтування моделі оцінювання властивостей та оптимальної вартості відповідних робіт, які дали б змогу забезпечити компромісне рішення щодо доцільності застосування повторно використовуваних компонентів при несистематичному використанні.

У першому розділі дипломної роботи проведено аналіз особливостей застосування підходу повторно використовуваного програмного забезпечення при розробці нових програмних систем, наведено переваги щодо ефективності процесу створення програмного забезпечення на основі таких компонентів, проаналізовано систематичне та не систематичне повторне використання та сформульовано основні задачі та актуальність дослідження.

У другому розділі дипломної роботи розроблено метод експертного оцінювання властивостей повторно використовуваних компонентів наслідуваного програмного забезпечення, шляхом аналізу властивостей компонентів ПЗ із застосуванням ЗС - моделі компонентів ПЗ та з урахуванням

впливу властивостей на витрати для виконання процесів повторного використання, що дало змогу підвищити ефективність розробки ПЗ. Окрім цього, обґрунтовано модель оцінювання властивостей повторно використовуваних компонентів в аспекті їх застосування.

У третьому розділі наведено архітектуру та вимоги до засобів автоматизації вимірювання вихідних кодів програмного забезпечення та експертного оцінювання властивостей повторно використовуваних компонентів. Проведено експерименти на основі статистичного аналізу метрик та експертних оцінок, їх кореляції.

У четвертому розділі проведено розрахунки щодо економічної ефективності застосування розробленого у дипломній роботі магістра методу і засобів.

У п'ятому розділі дипломної роботи розкрито питання охорони праці та безпеки в надзвичайних ситуаціях.

У шостому розділі проаналізовані вимоги до приміщень для експлуатації ЕОМ та методи узагальнення екологічної інформації.

## ABSTRACT

The theme of the thesis: " Research of methods and facilities of expert evaluation of inherited software properties " //Master thesis //Smuchok Ivan Mykchaylovych/. Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm-61 // Ternopil, 2015 // p. - 123, fig. - 35, Tab. -9 -8 Sheets A1, Add. -4, Ref. -46.

**KEYWORDS: REUSE, INHERITANCE, SOFTWARE, ASSESSMENT EXPERTS.**

The main objectives of the thesis is to study the current state of methods and means of expert evaluation of properties of recycled components imitations software, identifying their deficiencies, develop a method of expert evaluation of properties of recycled components to improve the efficiency of software development, study evaluation model properties and optimal values of related works, which would provide an opportunity to compromise on the usefulness of recycled components at systematically use.

In the first chapter of the thesis analyzes the application of a re-used software in the development of new software systems, are advantages in efficiency of the process of creating software based components such analyzes systematic and not systematic reuse and formulated the basic objectives and relevance of research.

In the second chapter of the thesis the method of expert evaluation of properties of recycled components imitations software by analyzing the properties of software components using 3C - model software components and with the influence of the properties of the cost to perform the process of re-use, leading to improved efficiency of software development. In addition, reasonable evaluation model properties reusable components in terms of their application.

The third section provides the architecture and requirements for automation measurement source software and expert evaluation of properties of recycled

components. Experiments based on statistical analysis of metrics and expert estimates of correlation.

The fourth section performed calculations of economic efficiency in applying the proposed master thesis work methods and tools.

In the fifth chapter of the thesis revealed issues of safety and security in emergency situations.

In the sixth chapter analyzes the requirements for facilities and methods of operation of computers generalization environmental information.

## ЗМІСТ

ВСТУП .....	11
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ПОВТОРНО ВИКОРИСТОВУВАНОВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	14
1.1. Аналіз ефективності застосування повторно використовуваного програмного забезпечення .....	14
1.2. Аналіз принципів повторно використання.....	18
1.2.1. Синхронне створення ПВК .....	19
1.2.2. Асинхронне виробництво компонентів ПЗ .....	20
1.2.3. Систематичне повторне використання .....	21
1.2.4. Несистематичне повторне використання .....	24
1.3. Процеси систематичного і несистематичного повторного використання .....	26
1.4. Відбір компонентів-кандидатів у повторно використовувані компоненти .....	28
1.5. Висновки до розділу .....	32
РОЗДІЛ 2 МЕТОД ЕКСПЕРТНОГО ОЦІНЮВАННЯ ВЛАСТИВОСТЕЙ ПОВТОРНО ВИКОРИСТОВУВАНИХ КОМПОНЕТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	34
2.1. Розробка методу експертного оцінювання властивостей повторно використовуваних компонентів.....	34
2.2. Властивості компонентів ПЗ, важливі з точки зору повторного використання .....	40
2.2.1. Обґрунтування застосування 3С-моделі властивостей компонентів ПЗ, важливих з точки зору повторного використання.....	41
2.2.2. Уточнення 3С-моделі властивостей компонентів ПЗ в аспектах створення та використання ПВК .....	45
2.3. Моделі властивостей .....	54



2.3.1. Застосування системного підходу для побудови моделей властивостей .....	54
2.3.2. Моделі властивостей, важливих в аспекті повторного використання .....	57
2.4. Висновки до розділу .....	60
<b>РОЗДІЛ 3 РОЗРОБКА АРХІТЕКТУРИ ЗАСОБІВ ПІДТРИМКИ МЕТОДУ ЕКСПЕРТНОГО ОЦІНЮВАННЯ ВЛАСТИВОСТЕЙ ПОВТОРНО ВИКОРИСТОВУВАНИХ КОМПОНЕНТІВ ТА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ .....</b>	<b>62</b>
3.1. Задачі та архітектура засобів оцінювання властивостей повторно використовуваних компонентів програмного забезпечення .....	62
3.2. Компоненти архітектури .....	65
3.2.1. Компоненти реалізації експертного оцінювання .....	65
3.2.2. Компоненти зберігання та аналізу даних .....	69
3.2.3. Вимірювач властивостей вихідного коду .....	72
3.3. Статистичний аналіз метрик та експертних оцінок .....	78
3.4. Висновки до розділу .....	83
<b>РОЗДІЛ 4 ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ.....</b>	<b>85</b>
4.1. Розрахунок норм часу на виконання науково-дослідної роботи .....	85
4.2. Розрахунок витрат на проведення НДР .....	86
4.3. Розрахунок ціни НДР і економічна ефективність від використання методу і засобу підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення.....	93
<b>РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....</b>	<b>96</b>
5.1. Охорона праці.....	96
5.2. Уражаючі фактори вибуху та способи захисту від них .....	99
<b>РОЗДІЛ 6 ЕКОЛОГІЯ .....</b>	<b>105</b>
6.1. Вимоги до приміщень для експлуатації моніторів і ПЕОМ. Шляхи дотримання цих вимог .....	105

6.2. Методи узагальнення екологічної інформації .....	106
ВИСНОВКИ .....	110
ДОДАТОК А ТЕКСТИ НАУКОВИХ ПУБЛІКАЦІЙ ДИПЛОМНОЇ РОБОТИ МАГІСТРА .....	112
ДОДАТОК Б ШКАЛИ ЕКСПЕРТНИХ ОЦІНОК ТА ПЕРЕЛІК МЕТРИК ВИХІДНОГО КОДУ ПОВТОРНОГО ВИКОРИСТОВУВАНИХ КОМПОНЕНТІВ НАСЛІДУВАНОВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	114
ДОДАТОК В МОДЕЛІ ВЛАСТИВОСТЕЙ ПОВТОРНО ВИКОРИСТОВУВАНИХ КОМПОНЕНТІВ НАСЛІДУВАНОВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	117
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	120

## ВСТУП

**Актуальність теми.** Сучасний ринок програмних продуктів характеризується великою кількістю однотипних засобів, здатних вирішувати складні задачі у різних бізнес доменах. Рівень якості таких продуктів, залежить від репутації та технологічного забезпечення фірм-розробників. Однак для того, щоб одержати максимальну вигоду від продукту з мінімальними витратами, виробникам необхідно першими «виходити на ринок», а це в свою чергу вимагає застосування ефективних підходів до розробки програмних систем. Як показує практика, найбільш дієвим є підхід повторного використання, що дає змогу знизити витрати на створення програмного забезпечення (ПЗ) шляхом застосування накопиченого досвіду. Підхід повторного використання застосовується такими відомими розробниками ПЗ, як IBM, Hewlett-Packard, Nec, Toshiba і досліджується багатьма вченими, зокрема? Л. Бабенко, І. Вельбіцький, М. Сидоров, Г. Цейтлин, V. Basili, T. Biggerstaff, G. Caldiera, E. Chisofsky, H. Rombach, I. Sommerville, P. Wegner.

Повторне використання ПЗ вимагає створення повторно використовуваних компонентів (ПВК). Під час несистематичного повторного використання, ПВК створюються на основі наслідуваного ПЗ, а доцільність та ефективність створення ПВК визначається насамперед витратами, які не повинні перевищувати витрати на створення такого компонента „з нуля”. При цьому, найбільш витратною, що потребує великих обсягів робочого часу висококваліфікованих інженерів, є завдання щодо оцінювання властивостей компонента ПЗ – кандидата в ПВК, яке на практиці розв’язується інтуїтивно, шляхом огляду та аналізу компонентів наслідуваного ПЗ, в першу чергу – кодів програм. Тому розробка методу та засобів автоматизації процесу підтримки експертного оцінювання властивостей компонентів ПЗ є актуальною.

**Мета і задачі дослідження.** Метою дипломної роботи магістра є розробка методу і засобів експертного оцінювання властивостей повторно використовуваних компонентів наслідуваного програмного забезпечення.

Для досягнення вказаної мети, у роботі поставлено та розв'язано **наступні задачі:**

- дослідження процесів забезпечення розробки ПЗ на основі підходу повторного використання;
- розробка методу експертного оцінювання властивостей повторно використовуваних компонентів ПЗ, що передбачає прямі вимірювання вихідних текстів програм та використовує метричні моделі властивостей, суттєвих з погляду повторного використання;
- визначення множини властивостей повторно використовуваних компонентів ПЗ;
- обґрунтування моделей для опису властивостей повторно використовуваних компонентів, які забезпечують автоматизовану реалізацію запропонованого методу із застосуванням вимірювань;
- розробка архітектури засобів, які забезпечують реалізацію запропонованого методу.

**Об'єкт дослідження** – процеси створення ПЗ із застосуванням повторно використовуваних компонентів та наслідуваного ПЗ.

**Предмет дослідження:** методи та засоби оцінювання властивостей компонентів ПЗ.

**Методи дослідження** У процесі розв'язання поставлених задач дипломної роботи магістра використано такі методи: синтез – під час визначення властивостей компонентів ПЗ, суттєвих з погляду повторного використання; моделювання і системний аналіз – під час обґрунтування моделей властивостей компонентів ПЗ; первинний статистичний, кореляційний та регресійний аналізи – під час доведення адекватності обґрунтованих моделей властивостям та підтвердження встановлених в

моделях залежностей; аналіз і проектування – під час розробки архітектури засобів, що автоматизують розв’язання задач оцінювання властивостей компонентів ПЗ.

**Наукова новизна отриманих результатів.** Наукова новизна полягає у вирішенні науково-практичної задачі експертного оцінювання властивостей повторно використовуваних компонентів наслідуваного ПЗ, при цьому одержано наступні результати:

– уперше розроблено метод експертного оцінювання властивостей повторно використовуваних компонентів наслідуваного програмного забезпечення та визначено множину їх властивостей, шляхом аналізу властивостей компонентів ПЗ із застосуванням ЗС - моделі компонентів ПЗ та з урахуванням впливу властивостей на витрати для виконання процесів повторного використання, що дало змогу підвищити ефективність розробки ПЗ;

– набула подальшого розвитку модель властивостей, яка базується на прямих метриках, що дає змогу забезпечити автоматизацію процесу експертного оцінювання на основі системного підходу та методу GQM;

**Практичне значення одержаних результатів.** Впровадження методу і засобів автоматизації експертного оцінювання властивостей повторно використовуваних компонентів дають змогу підвищити ефективність розробки програмного забезпечення.

**Публікації.** Результати дослідження апробовано на VIII Всеукраїнській студентській науково - технічній конференції "Природничі та гуманітарні науки. Актуальні питання" (23-24 квітня 2015 р.) та IV Міжнародній науково-технічній конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (25-26 листопада 2015 року) Тернопільського національного технічного університету імені Івана Пулюя у вигляді тез конференцій.

## РОЗДІЛ 1

### ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ПОВТОРНО ВИКОРИСТОВУВАНОВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Аналіз ефективності застосування повторно використовуваного програмного забезпечення

Сучасні підходи до розробки ПЗ базуються на наступних шляхах підвищення ефективності процесу створення ПЗ [55]:

- вдосконалення організаційно-технічного забезпечення розробки ПЗ;
- підвищення ефективності фаз життєвого циклу (ЖЦ);
- усунення фаз ЖЦ в розробці ПЗ;
- усунення доопрацювання ПЗ;
- повторне використання ПЗ.

Удосконалення організаційно-технічного забезпечення процесу розробки ПЗ включає в себе три напрямки діяльності: персонал, робоче місце та управління.

Перший напрямок діяльності має на меті застосування заходів з комплектування персоналу, що створює ПЗ. За моделлю СОСОМО рейтинг цих заходів найвищий [ ]. Існує кілька підходів до формування груп розробників ПЗ, від широко відомих у 70-х роках бригад «головного хірурга» [7] і так званих, "програмних ательє" до «парного програмування» [1].

Другий напрямок діяльності забезпечує застосування заходів по оснащенню робочого місця розробника ПЗ. Відомо, що такі фірми як Херох, TRW, IBM йдуть на високі витрати на організацію робочого місця і завдяки цьому забезпечують зростання продуктивності праці до 11% [79].

Третій напрям діяльності забезпечує застосування заходів для ефективного управління. Досвід показує, що погане управління може знизити ефективність витрат швидше, ніж інші фактори. Наприклад, раннє планування

розробки компонентів ПЗ, до яких ще не сформовані детальні вимоги, буде призводити до їх суттєвого доопрацювання [45].

Підвищення ефективності фаз ЖЦ. Аналіз досвіду розробки і супроводу ПЗ показує, що різні фази в ЖЦ мають різну трудомісткість, тому підвищення ефективності найбільш трудомістких фаз повинно підвищити в цілому продуктивність і знизити витрати. В якості міри зниження трудомісткості фази може виступати застосування інструментів, що автоматизують її процеси [9, 15].

Усунення окремих фаз розробки ПЗ. Очевидно, що найбільш надійним способом зменшення витрат є повне усунення фази ЖЦ, наприклад, за рахунок автоматичного виконання процесів. Однак, застосування цього способу не завжди можливо. Найбільш часто він ефективно використовується в фазах проектування та кодування. Для добре вивчених доменів можна досягати автоматичного переходу від специфікації завдання до безпосередньо виконуваної програми [10, 42, 43], але для не досліджених доменів такий перехід якщо і можливий, то тільки при значних витратах.

Усунення доопрацювання ПЗ. Основні заходи цього шляху спрямовані на створення засобів, що автоматизують розробку специфікацій (візуалізація специфікацій, пошук помилок, моделювання і т.д.), так як необхідність у доопрацюванні пов'язана, в основному, із змінами, внесеними в результати виконання початкових фаз розробки ПЗ.

Повторне використання ПЗ. Цей шлях підвищення продуктивності та зменшення витрат на розробку ПЗ почав досліджуватися у 80-х роках ХХ ст. [ ]. Сутність його полягає в зменшенні обсягу новостворюваного ПЗ за рахунок застосування компонентів існуючого ПЗ. Крім підвищення продуктивності програмування, цей шлях повинен призводити до підвищення надійності та якості як окремих програм, так і ПЗ в цілому, тому, крім зниження витрат на кодування, зменшуються також витрати на тестування, документування та супроводження ПЗ. Застосування повторного використання на ранніх фазах

ЖЦ ПЗ повинно також призвести до зменшення витрат на розробку специфікацій вимог і на проектування ПЗ.

Повторне використання залишається найбільш перспективним шляхом зниження витрат і підвищення продуктивності розробки ПЗ [ ]. Наприклад, як показують дослідження [ ], в даний час, перші чотири шляхи з перерахованих дають спільний приріст ефективності створення ПЗ близько 10% на рік - від 6% в розробці додатків реального часу до 16% у створенні бізнес-додатків великих телекомунікаційних компаній. Останній шлях - повторне використання - застосовується найбільшими виробниками ПЗ - AT & T, Brooklyn Union Gas, Ericsson, GTE, Hewlett-Packard, IBM, Motorola, Nec, Toshiba, при цьому частка повторно використовуваного коду оцінюється наступними значеннями [ ]: Hewlett-Packard - 25-50%; AT & T - 40-92% для телекомунікаційних додатків; Ericsson AXE - до 90%; Motorola - 85%. Крім того, завдяки реалізації цього шляху, спостерігається скорочення часу виходу ПЗ на ринок в 2-5 разів, зниження кількості помилок в 5-10 разів, зниження вартості супроводу в 5-10 разів і зниження загальної вартості розробки від 15 до 75% [ ]. Зазначені значення досягаються за рахунок скорочення витрат на пошук рішень завдань, що виникають при розробці ПЗ, зменшенні витрат на кодування, підвищенні надійності роботи ПЗ, оскільки застосовуються готові, добре протестовані компоненти.

Важливість повторного використання для розробки ПЗ знайшла відображення також у тому, що створена група моделей ЖЦ, що складається з трьох моделей, заснованих на повторному використанні ПЗ [47]: швидка, ітеративна і повна.

Швидка модель розглядає розробку ПЗ як зміну коду наслідуваного ПЗ із подальшою заміною інших робочих продуктів фаз ЖЦ (рис.1.1).



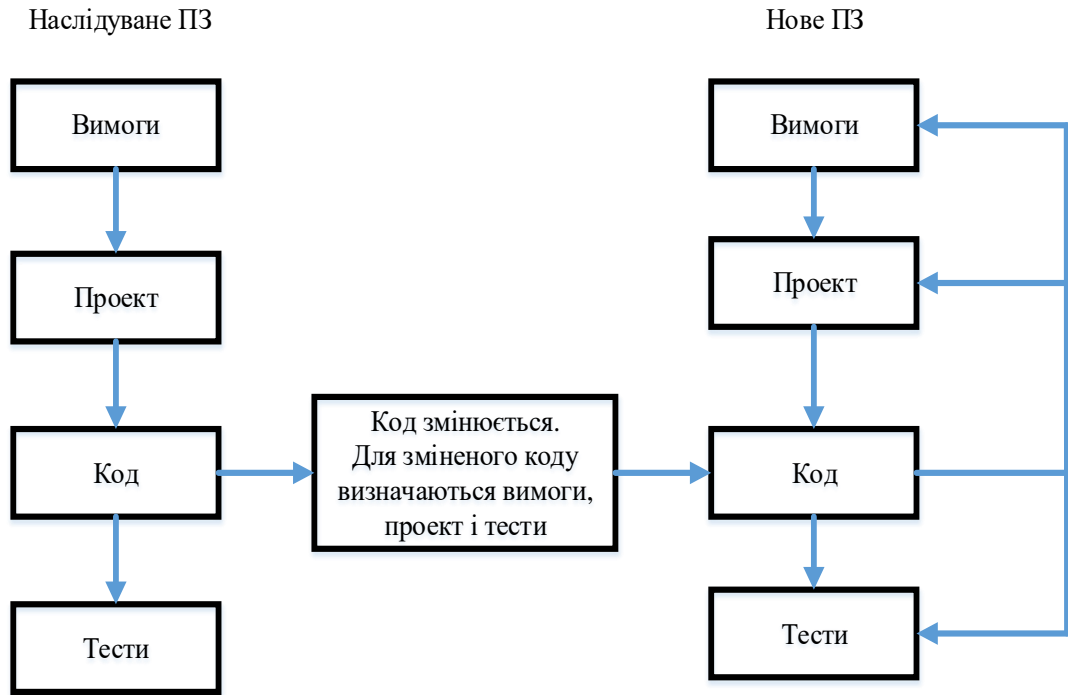


Рис 1.1. Модель швидкої розробки

Ітеративна модель спирається на аналіз наслідуваного ПЗ і побудови нового шляхом послідовних змін робочих продуктів успадкованого ПЗ (рис.1.2).

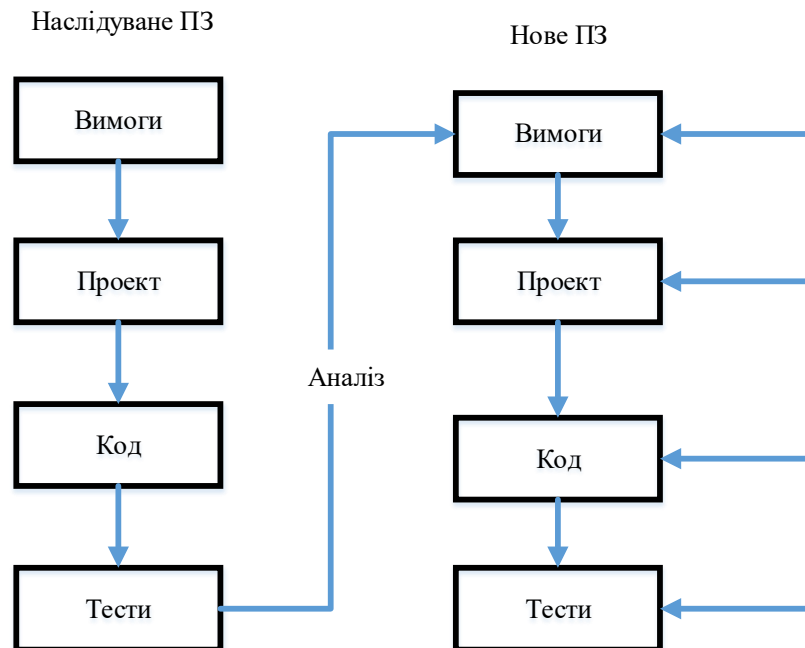


Рис. 1.2. Ітеративна модель

Повна модель ґрунтується на репозиторії повторно використуваних компонентів, що будується на основі наслідуваного ПЗ, і потім створення з допомогою репозиторію нового ПЗ (рис.1.3).

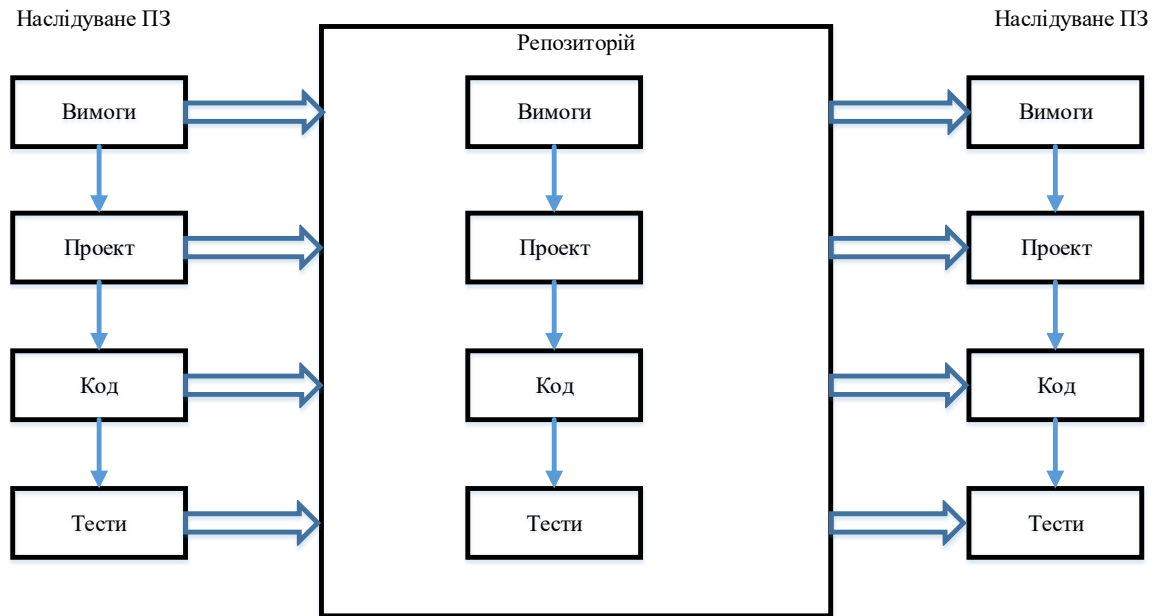


Рис. 1.3. Повна модель

Таким чином, широкомасштабне застосування повторного використання стало невід'ємною частиною всіх фаз і процесів ЖЦ.

## 1.2. Аналіз принципів повторно використання

Повторне використання ПЗ – це використання компонентів існуючого ПЗ для побудови нового [32, 51, 52]. Сутність принципу повторного використання ПЗ полягає у застосуванні в новій ситуації раніше отриманих результатів. Розглядаючи ПЗ в широкому сенсі, можна вказати на два рівні повторного використання результатів – ідей і знань; програмних продуктів та їх складових частин [ ]. Тому повторно використовуваними можуть бути не тільки компоненти коду, а й вимоги, результати аналізу, проектування, тестування і документування.

Компонент ПЗ – це складова частина (елемент) ПЗ, яка має межі, наділена інтерфейсом і методом застосування [33, 58]. При цьому, ПВК ПЗ – це елемент, який може повторно входити до складу хоча б ще однієї програмної системи [ ].

Створення нового ПЗ із застосуванням повторного використання складають два процеси (рис.1.4).

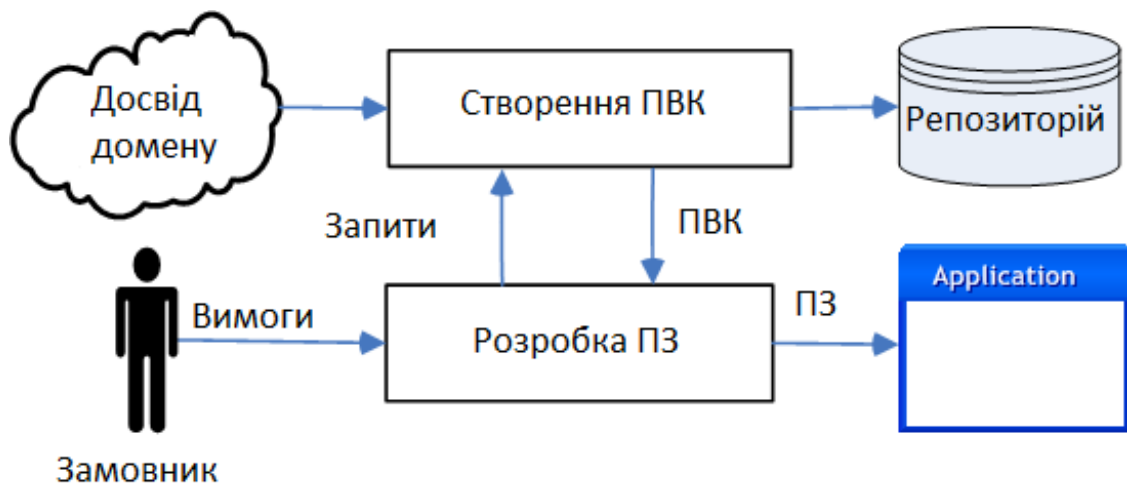


Рис. 1.4. Розробка нового ПЗ із застосуванням повторного використання

Сутність першого полягає у розробці ПЗ, а другого – у виробництві ПВК. При цьому, перший процес ґрунтується на застосуванні ПВК. По відношенню до першого процесу виробництво ПВК може здійснюватися двома способами: синхронним і асинхронним (рис. 1.5) [61].

### 1.2.1. Синхронне створення ПВК

Відповідно із запитом від розробників ПЗ, в першу чергу, реалізується пошук в репозитарії ПВК. При цьому видається результат, який найбільш точно відповідає вимогам запиту.

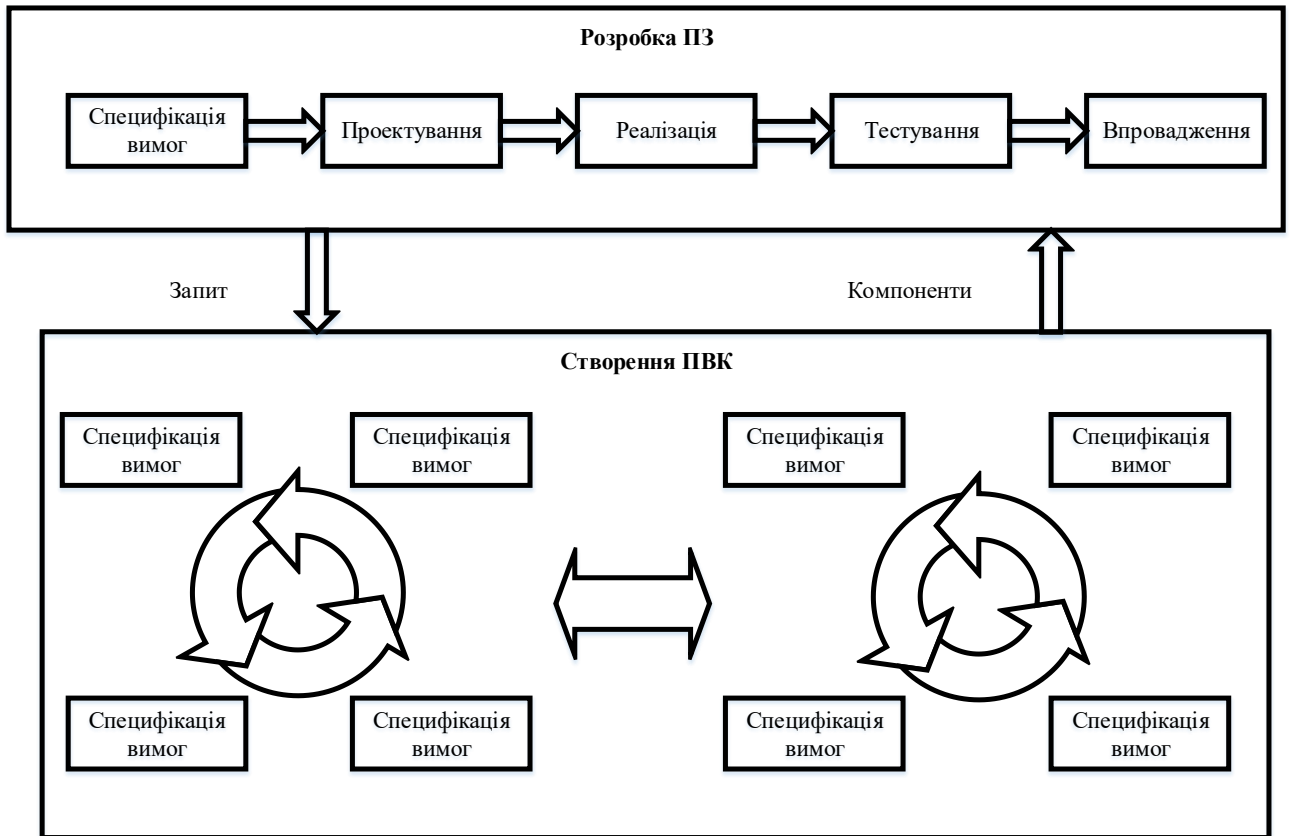


Рис. 1.5. Процеси розробки ПЗ із застосуванням повторного використання

Якщо ПВК знайдений, то він при необхідності, адаптується до вимог запиту, верифікується і надається розробникам нового ПЗ. Якщо ПВК не відповідає вимогам запиту і відсутній у репозитарії, то він розробляється «з нуля» або з більш елементарних компонентів, після цього зберігається в репозитарії ПВК і передається в розробку нового ПЗ [60].

### 1.2.2. Асинхронне виробництво компонентів ПЗ

Репозитарій ПВК планово поповнюється шляхом розробки ПВК незалежно від розробки ПЗ. При цьому реалізуються процеси планування, виробництва, пакетування і зберігання ПВК (рис.1.5).

Асинхронна діяльність спрямована, в першу чергу, на забезпечення наповненості репозиторію широким набором ПВК, оскільки ця умова є критичною для успішного застосування повторного використання.

За способом реалізації процесів виробництва ПМК повторне використання, яке використовується у розробці нового ПЗ, може бути систематичним і несистематичним [ 1 ].

### 1.2.3. Систематичне повторне використання

Суть систематичного повторного використання полягає у створенні «з нуля» компонентів, що реалізують функції для певного домену, і їх подальшого багаторазового використання при розробці ПЗ. Цей тип повторного використання припускає наявність глибоких знань про домен. Необхідною умовою доцільності систематичного повторного використання є багаторазова затребуваність створюваних ПМК [3, 4, 5, 10]. Тому компоненти, створювані шляхом систематичного повторного використання, представляють собою компоненти багаторазового використання або уніфіковані компоненти ПЗ [17]. Схема процесів систематичного повторного використання показана на рис. 1.6.

Оснoву систематичного повторного використання складає доменний аналіз.

При систематичному повторному використанні застосовується переважно дедуктивний спосіб створення ПМК ПО [52, 65]. Тому такі ПМК створюються на основі певних теорій і принципів, з використанням уявного досвіду, накопиченого окремими особистостями та колективами. Розробка ПМК ведеться з задоволенням усіх вимог, що пред'являються до них, звичайно зверху вниз. ПМК, отримані систематичним повторним використанням, - це частини програмних систем, принципи побудови яких добре вивчені, і відомо достатнє число їх реалізацій. Наприклад, частини компіляторів (сканери, синтаксичні аналізатори), інформаційно-пошукових систем (засоби відображення, сортування), проблемно-орієнтованих систем обробки даних (компоненти стиснення даних, обробки зображень, обробки запитів, фільтри, генератори звітів, репозитарії компонентів загальносистемного призначення.

Зазвичай, в домені, ПВК, отриманих таким шляхом небагато, тому для їх зберігання і застосування не потрібно потужної інфраструктури. Процес створення таких ПВК більше схожий на системне проектування, ніж на аналіз. Організаційна інфраструктура систематичного повторного використання наведена на рис.1.7 [7].

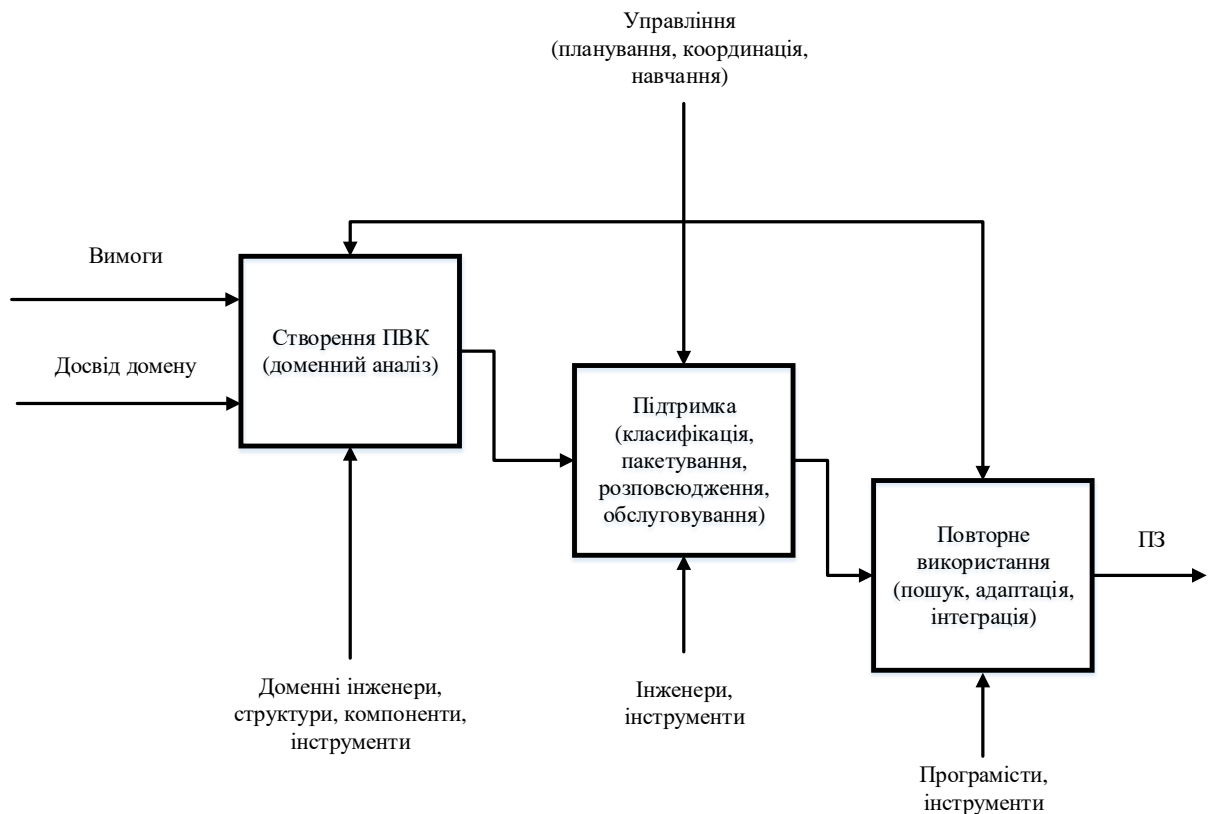
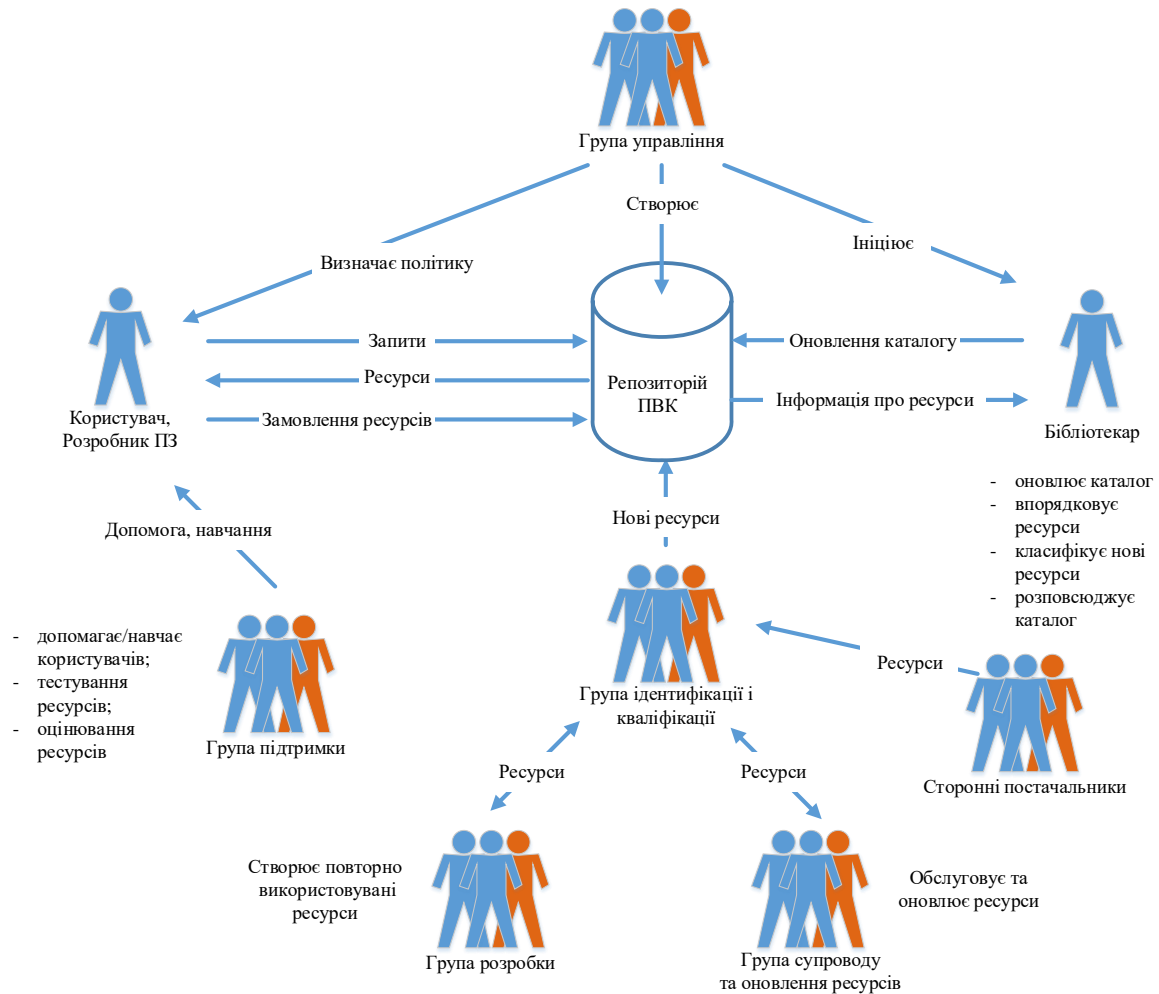


Рис. 1.6. Процеси систематичного повторного використання

Причинами виведення з експлуатації та заміни технічних засобів є фізичне і моральне зношення. ПЗ, фактично, не має фізичного зношення. До факторів морального зношення ПЗ можна віднести наступне:

- старіння методів, покладених в основу функціонування ПЗ;
- старіння технічних характеристик ПЗ у зв'язку з розвитком технологій розробки і супроводу ПЗ;

- складність інтеграції ПЗ із зовнішніми або новими компонентами у зв'язку з його невідповідністю новим стандартам взаємодії програмних компонентів;
- збільшення вартості супроводу ПЗ;
- необхідність використання нової апаратної платформи для функціонування ПЗ.



### 1.7. Організаційна структура систематичного повторного використання

Зазначені фактори призводять до необхідності виведення з експлуатації та утилізації ПЗ. Останнє може здійснюватися шляхом виконання трьох наступних процесів [30]:

- повторне використання – використання компонентів існуючого ПЗ за іншим призначенням в новому ПЗ;

- переробка – заміна неефективно працюючого або такого, що є непридатним ПЗ і побудова нового ПЗ;
- відновлення – відновлення працездатності ПЗ або його компонентів.

#### 1.2.4. Несистематичне повторне використання

Несистематичне повторне використання – це використання компонентів існуючого (наслідуваного) ПЗ в розробці нового ПЗ. Цей тип повторного використання припускає застосування елементів раніше розробленого ПЗ шляхом їх виділення з наслідуваного ПЗ, переробки та / або налаштування, та інтеграції в нове ПЗ.

При несистематичному повторному використанні компоненти наслідуваного ПЗ можуть використовуватися як «чорні скриньки» або як «білі скриньки». При повторному використанні компонента як «чорної скриньки» його не змінюють, а при повторному використанні «білого ящика» можуть змінювати як його конструкцію, так і концепцію (рис. 1.8) [59].



Рис. 1.8. Види змінюваності повторного використання



Залежно від модульної структури наслідуваного ПЗ і необхідного ступеня гранульованості програмних конструкцій виділення компонентів - кандидатів у ПВК може здійснюватися двома наступними способами [35] (рис. 1.9):

- рефакторинг ПЗ – поділ конструктивно закінчених частин ПЗ [33]. Очевидно, що таке виділення можливе тоді, коли частини інкапсульовані і капсули мають мовно розділені «стики» (інтерфейси), наприклад, коли розділяється компонент – це підпрограма, модуль або клас;
- гранулювання ПЗ – виділення окремих функцій або фрагментів з «монолітного» ПЗ і створення для них відповідних інкапсульованих конструкцій [33].

Гранулювання застосовується при відсутності в елементів, що виділяються, формальних стиків або тоді, коли обрана для виділення частина програми є не повним текстом підпрограми, модуля або класу.

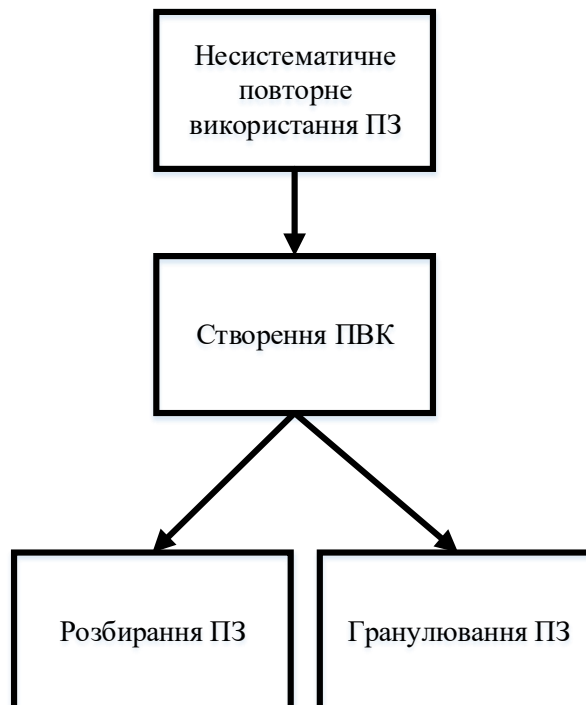


Рис. 1.9. Способи виділення компонентів для повторного використання ПЗ

### 1.3. Процеси систематичного і несистематичного повторного використання

Процеси несистематичного повторного використання утворюють дві фази (рис.1.10) [61]: створення ПВК, і використання ПВК в новому ПЗ.

Перша фаза включає наступні процеси:

- створення ПВК – виділення шляхом розбирання або гранулювання елементів наслідуваного ПЗ, які можуть бути використані повторно, і створення ПВК на їх основі (рис.1.10);
- зберігання ПВК – розміщення ПВК в репозитарії для подальшого використання.

Друга фаза включає наступні процеси:

- пошук ПВК - знаходження в репозитарії ПВК, що задовольняють вимогам запиту;
- налаштування або модифікація ПВК – доопрацювання ПВК шляхом його параметризації або модифікації перед використанням в новому ПЗ (рис.1.10);
- верифікація – перевірка правильності ПВК, у випадку, якщо виконаний попередній етап;
- вбудовування ПВК – інтеграція ПВК в нове ПЗ.

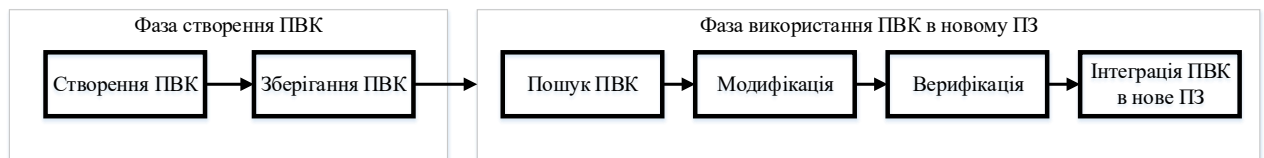


Рис. 1.10. Процеси несистематичного повторного використання, які реалізуються за допомогою ПВК

Процес створення ПВК поділяють на кілька операцій, які групують в етапи ідентифікації та кваліфікації (рис. 1.11) [61].

Етап ідентифікації включає операції виділення та відбору компонентів-кандидатів у ПВК. Виділення компонентів з наслідуваного ПЗ виконується програмістом вручну або автоматично, із застосуванням відповідних інструментів (редактори, аналізатори гранулятори). Виділені компоненти стають компонентами-кандидатами в ПВК, до яких далі застосовують операцію відбору. Відбір компонентів здійснюється шляхом їх оцінювання на придатність до створення ПВК. Відібрані компоненти переходять в етап кваліфікації (рис. 1.11).

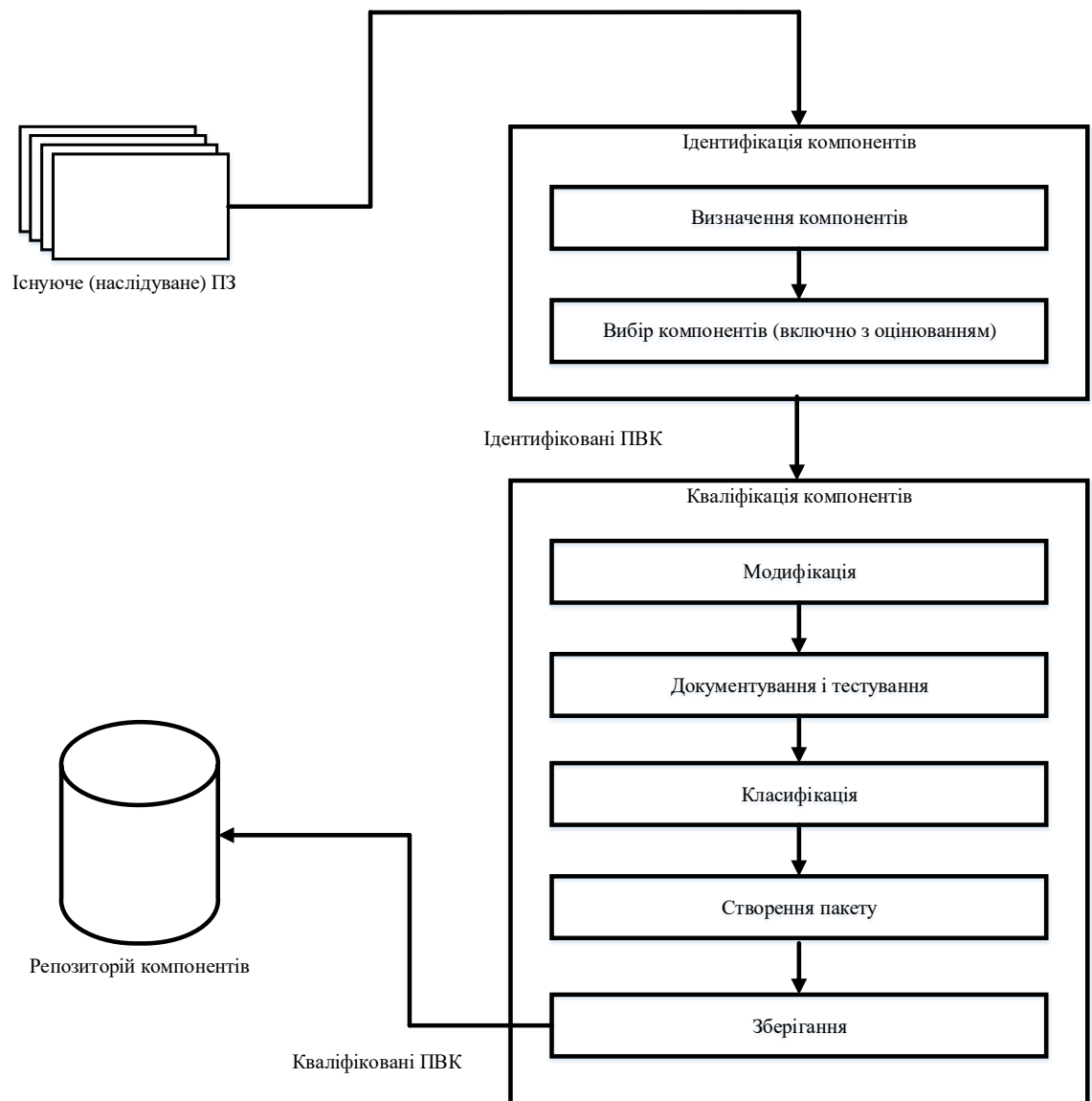


Рис. 1.11. Створення ПВК

Відбір компонентів дозволяє знижувати вартість створення ПВК шляхом виключення малопродатних для створення ПВК компонентів-кандидатів до етапу кваліфікації.

Етап кваліфікації включає операції модифікації (у випадку «білої скриньки»), документування, тестування, класифікації, пакетування та збереження компонентів. Модифікація компонента-кандидата проводиться програмістом для досягнення вимог, які застосовуються до ПВК, шляхом зміни його вихідних кодів. Документування компонента проводиться шляхом створення для нього специфікацій ПВК (коментарі в тексті програм, описи в окремих файлах), які необхідні при пошуку ПВК в репозиторії та його використанні в новому ПЗ. Тестування створеного ПВК проводиться для перевірки його правильності шляхом застосування відповідних методів і засобів. Класифікація ПВК проводиться програмістом або інженером, який обслуговує репозиторій, на основі його специфікацій і відповідно до визначеної таксономії, для забезпечення каталогізації і пошуку ПВК. Пакетування ПВК проводиться для забезпечення його цілісності при зберіганні і транспортуванні, шляхом його поміщення у файли, архівні файли або дистрибутивні. Створені пакети ПВК зберігаються в репозиторії шляхом їх приміщення у відповідні каталоги файлових систем або спеціальну БД.

#### 1.4. Відбір компонентів-кандидатів у повторно використовуваних компонентах

При несистематичному повторному використанні, на етапі ідентифікації (рис.1.11) повинен проводитися відбір компонентів ПЗ - кандидатів у ПВК.

Загальна схема відбору компонентів наведена на рис.1.12, а функції програміста, що здійснює відбір, на рис.1.13.

Відбір компонентів зазвичай будується на припущенні, що створення ПВК не повинно вимагати витрат, великих або рівних з витратами, необхідними для створення аналогічного компонента «з нуля» [52]. Витрати

можна оцінювати прямо, використовуючи відомості про вартість, необхідних для виконання процесів повторного використання, або побічно, використовуючи знання про те, наскільки властивості компонента ПЗ відповідають функціональним і нефункціональним вимогам запиту на ПВК. Тому для відбору компонентів ПЗ - кандидатів у ПВК зараз використовуються два типи моделей:

- вартості – рішення про повторне використання приймається на основі оцінювання вартості виконання процесів, пов'язаних з перетворенням компонентів ПЗ в ПВК [ ];
- властивостей – рішення про повторне використання приймається на основі оцінювання властивостей, що відображають повторну використовуваність компонентів ПЗ [55].

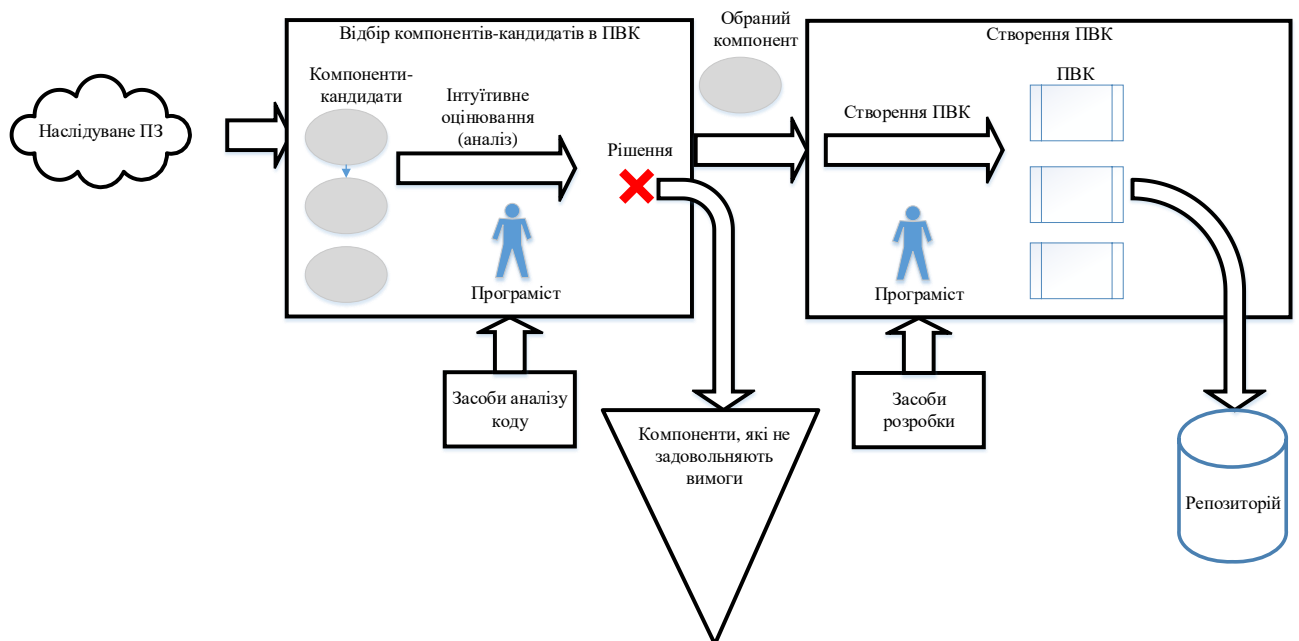


Рис 1.12. . Відбір компонентів ПЗ - кандидатів у ПВК

Модель вартості будується шляхом ідентифікації дій (операцій), що відповідають кожному типу повторного використання (розробка «з нуля», несистематичне, систематичне повторне використання) та асоційованих з

ними вартості (витрат), необхідних для виконання цих дій. Далі, розподіляючи загальну вартість розробки, заснованої на повторному використанні, на три частини (створення продукту, конструювання базового набору ПВК, підтримку інфраструктури) і будуючи для процесів повторного використання сценарії у вигляді схем переходів у двовимірній моделі «перетворення-перенесення», вартість відповідних процесів визначається як вартість кожного сценарію (сума оцінок вартостей, що входять в сценарій базових операцій).



Рис. 1.13. Функції програміста, який проводить відбір компонентів

Для несистематичного повторного використання вартість процесів має вигляд

$$C_{CA}(0, S_i) + C_{WB}(S_i, T_i), \quad (1.1)$$

де  $C_{CA}$  – вартість операцій витягання компонента з наслідуваного ПЗ;

$C_{WB}$  – вартість модифікації компонента ПЗ  $S_i$  (white-box reuse) в ПВК  $T_i$ .

Оскільки відбір компонентів ПЗ – кандидатів у ПЗ проводиться після вилучення компонента з наслідуваного ПЗ, то далі в дослідженні перша складова вартості  $C_{CA}$  не розглядається.

Моделі вартості дозволяють визначати доцільність створення ПВК на основі вартісного критерію, однак, похибка оцінки витрат може бути дуже значною, а застосування вартісної моделі для несистематичного повторного використання неефективним, беручи до уваги велику різноманітність як вихідних компонентів ПЗ ( $S_i$ ), так і операцій, виконання яких потрібно для переходу від  $S_i$  до  $T_i$ .

Модель властивостей створюється на основі PVB (property-values based) – підходу шляхом асоціювання з вимогами до компонентів так званих складних атрибутів [55] і деталізацією їх за допомогою простих атрибутів, які є прямо вимірюваними. На основі таких моделей здійснюється сертифікація компонентів, а компоненти, які відповідають вимогам поміщаються в репозиторій. Компонент вважається таким, що відповідає заданим вимогам, якщо всі його атрибути мають значення «Yes» у шкалі «Yes-No». Очевидно, що такі моделі можна застосовувати до компонентів із заздалегідь визначеними атрибутами і бінарними шкалами, тобто до компонентів систематичного повторного використання. Наприклад, в [ ] вимоги (атрибути) визначаються стандартами безпеки ISO / IEC 15408, IEC 61508.

У роботі [ ] використовується PVB-підхід для відбору компонентів ПЗ - кандидатів у ПВК при несистематичний повторному використанні. При цьому визначається модель властивостей, що базується на атрибутах компонентів, яка включає наступне:

- модель факторів повторного використання (рис.1.14), що складається з трьох факторів (корисність, якість, вартість ПВК) і субфакторів;
- чотири прямі метрики – цикломатична складність (C) [41], регулярність (R), обсяг (V), частота використання (RF) [73], які поставлені у відповідність деяким із субфакторів.

При побудові моделі, багато авторів роблять припущення про існування залежностей між введеними факторами і вибраними метриками, але задача формалізації та підтвердження цих залежностей не розв'язана. Для визначення

критеріїв відбору компонентів у роботі [ ], на основі експериментальних даних, отриманих за допомогою вимірювання компонентів ПЗ, визначається зв'язок між метрикою частоти повторного використання (RF) та іншими трьома метриками моделі.

Таким чином, аналіз моделей, які використовуються для відбору компонентів ПЗ – кандидатів у ПВК – показує, що для несистематичного повторного використання доцільно застосовувати комбінований підхід, основу якого складає модель властивостей, а при формуванні множини властивостей використовується модель вартості. Такий підхід використаний при розробці методу експертного оцінювання, обґрунтованого у дипломній роботі магістра.

#### 1.5. Висновки до розділу

Повторне використання є одним з найбільш ефективних шляхів зниження витрат на створення ПЗ. Суть принципу повторного використання полягає у застосуванні знань і компонентів існуючого ПЗ при створенні нового. Існують види повторного використання – систематичне і несистематичне. При несистематичному повторному використанні ПВК створюються з компонентів ПЗ, які виділяються з наслідуваного ПЗ або шляхом розбирання, або шляхом грануляції. При створенні ПВК компоненти ПЗ проходять етап ідентифікації, суть якого полягає у відборі компонентів - кандидатів у ПВК. Відбір компонентів, зазвичай, будується на припущенні, що створення ПВК не повинно вимагати витрат, більших або рівних витратам, необхідним для створення аналогічного компонента «з нуля».

У даному розділі одержано наступні результати:

1. Проведено аналітичний огляд літератури; наведено існуючі види повторного використання (систематичне, несистематичне); розглянуто та проаналізовано процеси їх реалізації та шляхи застосування при розробці ПЗ;



2. Проаналізовано процеси створення ПВК при несистематичному повторному використанні наслідуваного ПЗ; показана роль процесу відбору компонентів ПЗ як одного з ключових у забезпеченні ідентифікації компонентів ПЗ - кандидатів у ПВК;

3. Виконано аналіз існуючих моделей, які можуть бути використані для реалізації відбору компонентів ПЗ – кандидатів у ПВК; розглянуті моделі вартості (рішення про повторне використання приймається на основі оцінки вартості виконуваних процесів, пов'язаних з перетворенням компонентів ПЗ в ПВК) і моделі властивостей (рішення про повторне використання приймається на основі оцінок властивостей, що відображають повторну використовуваність компонентів);

4. На основі аналізу моделей обґрунтовано доцільність розробки та дослідження методу експертного оцінювання, заснованого на комбінації моделей властивостей і вартості.

## РОЗДІЛ 2

### МЕТОД ЕКСПЕРТНОГО ОЦІНЮВАННЯ ВЛАСТИВОСТЕЙ ПОВТОРНО ВИКОРИСТОВУВАНИХ КОМПОНЕТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 2.1. Розробка методу експертного оцінювання властивостей повторно використовуваних компонентів

Експертні оцінки – це кількісні або порядкові оцінки процесів або явищ, які не можуть бути безпосередньо вимірними [ ]. Методи експертного оцінювання – це методи науково-технічного прогнозування, які базуються на припущенні, що на основі думок експерта можна побудувати адекватну модель майбутнього розвитку об'єкта прогнозування [ ]. Залежно від типу експерта (індивідуальний чи колективний) розрізняють методи індивідуальних і колективних оцінок (рис.2.1)

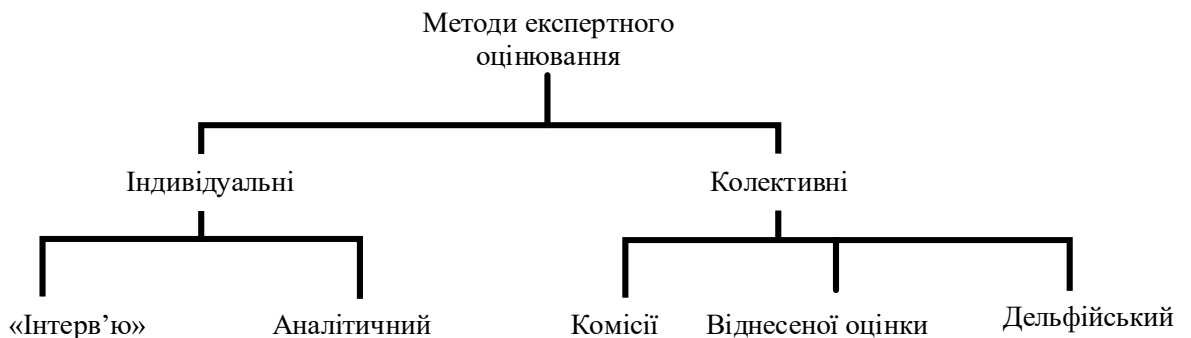


Рис. 2.2. Методи експертного оцінювання компонентів ПЗ

Метод експертного оцінювання – «інтерв'ю» ґрунтується на бесіді інженера з прогнозування та експерта в ході якої перший, відповідно до заздалегідь розробленої програми, ставить перед експертом питання щодо перспектив розвитку прогнозованого об'єкта, а експерт дає на них відповіді.

Метод експертного оцінювання – «аналітичний» ґрунтується на самостійній роботі експерта, спрямованій на аналіз тенденцій, оцінювання майбутнього стану і шляхів розвитку прогнозованого об'єкта.

Метод експертного оцінювання – «комісії» заснований на проведенні групою експертів дискусії з метою вироблення спільної позиції з питань майбутнього розвитку прогнозованих об'єктів. При використанні даного методу позначається взаємний вплив експертів, відома інерційність у відмові від одного разу висловленої публічно думки та інші фактори, що призводять до небажаних наслідків.

Метод експертного оцінювання – «віднесеної оцінки» («мозковий штурм») ґрунтується на поділі експертів на дві групи: експерти, які вільно висловлюють думки про рішення даної проблеми та експерти, які визначають оптимальне рішення.

Метод експертного оцінювання «дельфійський» – базується на відмові від прямих колективних обговорень, шляхом заміни програмою послідовних індивідуальних опитувань, які повторюються до досягнення прийнятної збіжності висловлених думок.

Застосування методів індивідуального експертного оцінювання дає високий ступінь залежності результатів від суб'єктивної думки одного експерта. Застосування методів колективного експертного оцінювання дає змогу знизити суб'єктивність, підвищити точність і ступінь конкретизації прогнозу, але, вимагає великих витрат через залучення сукупності експертів [ ].

Для вибору компонентів ПЗ - кандидатів у ПВК, об'єктом експертного оцінювання є компонент ПЗ, який представляється властивостями, які відображають його потенційну повторну використовуваність. Тому реалізація методу експертного оцінювання компонентів ПЗ вимагає виконання двох наступних процесів (рис. 2.2):

– оцінювання властивостей, важливих в аспекті повторного використання;

– оцінювання придатності компонента ПЗ кандидата в ПВК і ухвалення відповідного рішення про створення ПВК або «відкиданні» компонента.

При виконанні зазначених процесів може застосовуватися один з описаних вище методів експертного оцінювання. Загальна схема реалізації методу показана на рис.2.3.



Рис. 2.2. Процеси експертного оцінювання компонентів ПЗ

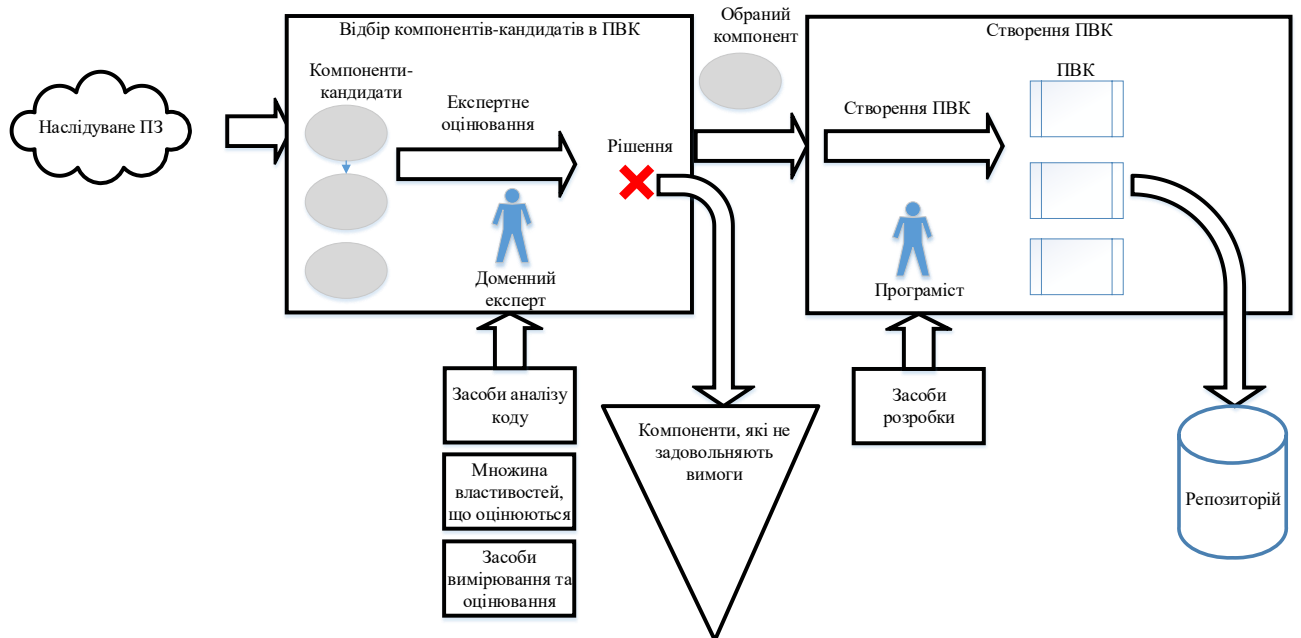


Рис. 2.3. Схема експертного оцінювання

Найбільш трудомістким при застосуванні методу експертного оцінювання є перший процес. Це пов'язано з тим, що експерту необхідно не тільки оцінювати множину властивостей, які відображають повторну використовуваність, але й, беручи до уваги несистематичне повторне використання (white-box reuse), досліджувати код компонента ПЗ. При цьому експерт повинен використовувати досить складні інструменти аналізу вихідного коду (аналізатори, діаграмери, реструктуризатори [53]), щоб одержувати обґрунтовані оцінки.

Другий процес застосування методу експертного оцінювання є менш трудомістким, однак вимагає наявності в експерта значного досвіду для уникнення значних помилок. Виконання обох процесів вимагає від експерта крім знань в області програмування ще знання домену.

Беручи до уваги трудомісткість першого процесу і можливість виконання вимірювань в ПЗ [ ], пропонується оцінки всіх або деяких властивостей отримувати автоматично і тим самим виключити частково або повністю процес оцінювання властивостей компонента ПЗ (рис.2.2). З урахуванням автоматизації першого процесу функції експерта в схемі показані на рис.2.4.

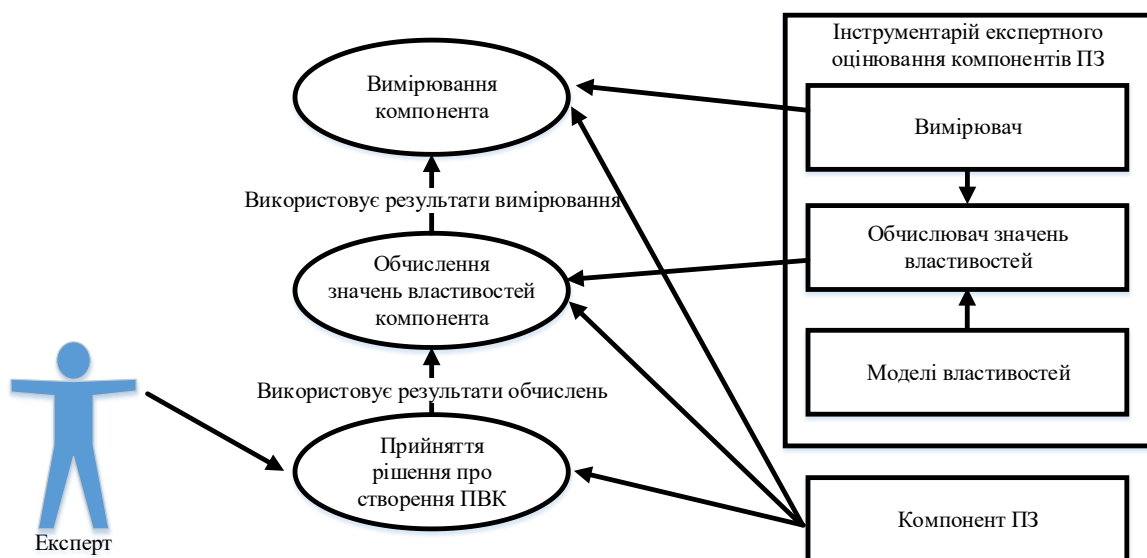


Рис. 2.4. Процеси автоматизованого експертного оцінювання

При автоматизованому оцінюванні повинно виконуватися три процеси (рис. 2.4):

- вимірювання компонента;
- обчислення значень властивостей компонента;
- прийняття рішення про створення ПВК.

Процес вимірювання компонента повинен виконуватися за допомогою засобу – вимірювача вихідних текстів програм, що надає значення прямих метрик компонента.

Процес обчислення значень властивостей компонента повинен виконуватися за допомогою засобу – обчислювача значень властивостей, що забезпечує отримання значень властивостей на основі результатів вимірювання компонента, і моделей властивостей, що відображають залежності між прямими метриками і непрямими властивостями компонентів ПЗ.

Так як, обчислення значень властивостей компонентів, і тим самим автоматизація експертного оцінювання, має проводитися на основі моделей властивостей, то необхідно забезпечити формалізацію і адекватність цих моделей.

У дипломній роботі магістра пропонується забезпечувати формалізацію і адекватність таких моделей шляхом їх уточнення за допомогою досвідчених експертів, що володіють знаннями в повторному використанні. Для цього необхідно послідовно виконувати три процесу:

- оцінку властивостей компонентів,
- вимірювання компонентів;
- уточнення моделей, за допомогою засобів, які показані на рис.2.5.

Процес оцінювання властивостей компонентів ПЗ повинен виконуватися досвідченим експертом за допомогою засобу оцінювання, що забезпечує перегляд тексту компоненту, введення і збереження результатів експертного оцінювання. Експерт повинен забезпечувати значення оцінок за результатами дослідження коду компонента одним із способів. До найбільш

часто використовуваних способів одержання значень експертної оцінки відносяться ранжування, парне порівняння і безпосередня оцінка [44].

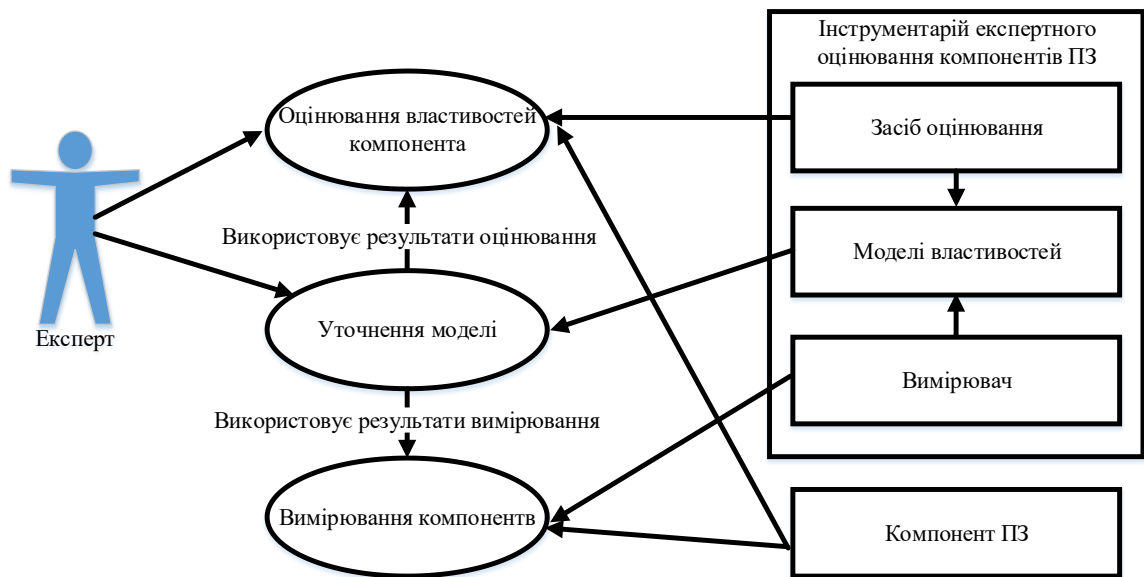


Рис. 2.5. Процеси і засоби уточнення моделей властивостей

Ранжування базується на впорядкуванні об'єктів за ступенем їх впливу на результат. На основі своїх знань і досвіду експерт повинен оцінити об'єкти в порядку переваги, керуючись одним або декількома показниками порівняння.

Парне порівняння використовує процедуру встановлення переваги об'єктів при порівнянні всіх можливих пар. На відміну від методу ранжування, відповідно до якого впорядкування здійснюється на всіх об'єктах, при парному порівнянні експерти вирішують більш просту задачу – порівнюють пари об'єктів. При цьому в парах можливі відношення або порядку, або еквівалентності.

Безпосередня оцінка базується на процедурі приписування об'єктам числових значень в шкалі інтервалів. Кожне значення відповідає ступеню впливу того чи іншого об'єкта на результат. Оцінку в шкалі інтервалів можна виконати з високим ступенем довіри тільки при хорошій інформованості експертів про властивості об'єктів і предметної області. З метою ослаблення

цих умов замість неперервної числової осі можуть використовуватися шкали з балами [73].

Оскільки способи ранжування і парного порівняння об'єктів засновані на виборі кращого об'єкта з багатьох об'єктів, то їх застосування для процесу оцінювання властивостей компонентів - кандидатів у ПВК (рис.2.5) неможливо, так як ці компоненти надходять експерту по одному. Тому в якості способу для отримання значень оцінок компонентів ПЗ обрана безпосередня оцінка їх властивостей одним або декількома експертами за допомогою числових або шкал порядку.

Процес вимірювання компонента повинен виконуватися, так само, як і при автоматизованому оцінюванні, за допомогою того ж засобу – вимірювача вихідних текстів програм, отримувати та зберігати значення прямих метрик компонента.

Процес уточнення моделей властивостей повинен виконуватися після проведення достатньої кількості оцінок і вимірювань на основі збережених значень з використанням, наприклад, статистичних методів.

Таким чином, для реалізації запропонованого методу необхідно вирішити наступні завдання:

- визначити множину властивостей компонентів ПЗ, важливих з точки зору повторного використання;
- розробити моделі для властивостей, важливих з точки зору повторного використання, що забезпечують автоматизовану реалізацію запропонованого методу із застосуванням вимірювань;
- розробити архітектуру засобів, що забезпечують реалізацію запропонованого методу.

2.2. Властивості компонентів ПЗ, важливі з точки зору повторного використання



Властивість виражає таку сторону предмета, яка обумовлює його відмінність чи спільність з іншими предметами і проявляється в його ставленні до них [40]. Задача визначення властивостей компонентів ПЗ, істотних з точки зору повторного використання, полягає у знаходженні і формулюванні таких сторін компонентів ПЗ, які обумовлюють їх істотні відмінності в процесах повторного використання.

### 2.2.1. Обґрунтування застосування 3С-моделі властивостей компонентів ПЗ, важливих з точки зору повторного використання

У дипломній роботі магістра для реалізації схеми вибору компонентів ПЗ використовується PVB - підхід. При побудові моделей властивостей, важливих з погляду повторного використання, пропонується використовувати 3С-модель компонентів ПЗ [78]. Сутність 3С-моделі компонента ПЗ полягає в тому, що онтологія властивостей компонентів ПЗ будується з урахуванням трьох аспектів: концепції (Conception), конструкції (Construction) і контексту (Context).

Концепція програмного компонента відображає його функціональність. Доменний експерт сприймає концепцію у вигляді функцій, які моделюють певні об'єкти і процеси предметної області.

З погляду повторного використання функціональність повинна бути спільною (якомога ширшого застосування) і корисною. Тому в аспекті концепції компонента істотними є властивості корисності та спільності.

Корисність відображає ступінь задоволення вимог, що висувуються користувачами до компоненту.

Спільність відображає широту можливого застосування ПВК.

Контекст відображає вимоги до середовища використання ПВК у складі ПЗ. У 3С-моделі можна виділити контекст концепції (семантичний контекст) і контекст конструкції (синтаксичний контекст). При створенні ПВК семантичний контекст формується програмістом як логіка його взаємодії з

іншими компонентами. Синтаксичний контекст формується як інтерфейс ПВК. Тому в аспекті контексту істотними є наступні властивості: закінченість інтерфейсу, синтаксична незалежність, семантична незалежність. Закінченість інтерфейсу відображає те, що інтерфейс компонента представляє всю функціональність, яка повинна бути доступна при використанні.

Семантична незалежність відображає здатність компонента мати фіксований результат при виконанні компонента, незалежно від місця, в якому він оголошується або викликається.

Синтаксична незалежність відображає здатність компонента використовуватися без зміни контексту, в який він поміщається;

Конструкція програмного компонента відображає одну з реалізацій концепції. Тому в аспекті конструкції важливими при повторному використанні є наступні властивості:

- достатність – відображає здатність конструкції так представляти концепцію, щоб цього було достатньо для повної передачі функціональності;
- закінченість конструкції – відображає те, що конструкція представляє всі можливості функціональності з концепції компонента;
- примітивність – відображає те, що в конструкцію повинні включатися тільки елементарні операції над об'єктами, передбачені в концепції;
- модульність – відображає те, що компонент має модульну внутрішню структуру, що спрощує його розуміння і модифікацію;
- структурованість – відображає використання в компоненті структурних операторів і їх складність, яка впливає на його розуміння і модифікацію;
- трасування (слідування) – відображає можливість одержувати в процесі розробки «слід» від однієї фази життєвого циклу до іншої, і використовувати ефект інклюзії;

- надійність – відображає стійку поведінку компонента, навіть якщо він неправильно використовується або використовується на межі своїх можливостей;
- ефективність – відображає однаково ефективне виконання компонента в рамках всього спектру застосувань, передбачених для нього специфікацією вимог;
- переносимість – відображає працездатність компонента в різних апаратних і програмних середовищах;
- класифікація – відображає здатність ефективного розміщення і пошуку компонента відповідно до наперед заданої класифікації;
- компактність – відображає здатність компактного зберігання компонента;
- зрозумілість – відображає суть компонента, достатню для його використання у процесі розробки нового ПЗ;
- простота модифікації – відображає простоту внесення змін у конструкцію компонента;
- коректність – відображає відсутність синтаксичних і семантичних помилок у тексті компонента;
- тестопридатність – відображає здатність ефективного тестування компонента.

Одержана у дипломній роботі ЗС-модель властивостей показана на рис.2.6.

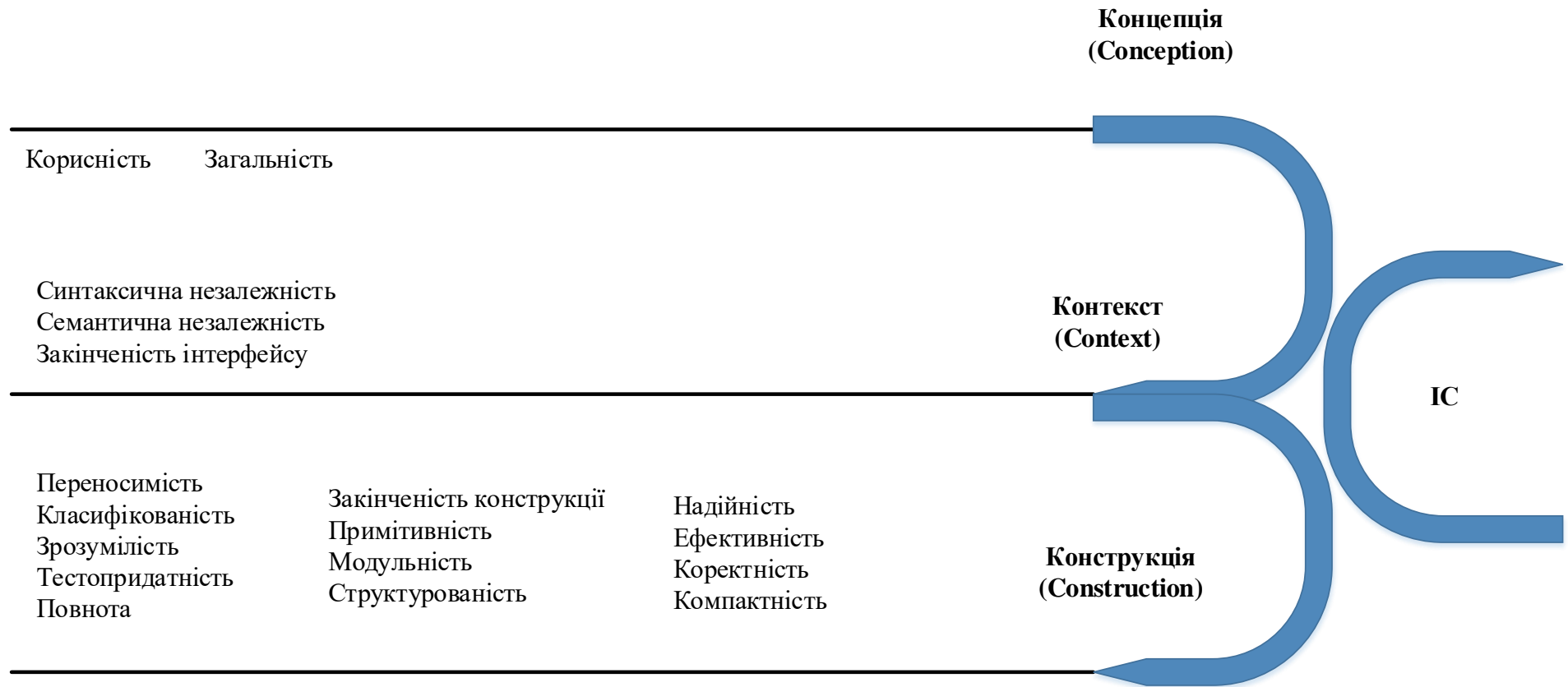


Рис. 2.6. 3С-модель властивостей, суттєвих з точки зору повторного використання

### 2.2.2. Уточнення ЗС-моделі властивостей компонентів ПЗ в аспектах створення та використання ПВК

Для визначення остаточного складу множини властивостей компонентів наслідуваного ПЗ, істотних з погляду повторного використання, використовується додатковий аналіз властивостей ЗС-моделі, з урахуванням впливу кожної властивості моделі на вартість процесів несистематического повторного використання (рис. 1.11).

Розглянемо цю задачу для процесів фаз створення і використання ПВК.

Створення ПВК проводиться відповідно до вимог, які обумовлюються реалізованим принципом повторного використання (систематичне або несистематичне повторне використання, ведеться синхронна або асинхронна розробка ПВК), а також підходами і технологіями, застосовуваними для цього в організації. Для виконання цих вимог розробник надає ПВК відповідні властивості, які роблять можливим і доцільним його повторне використання. Тому в загальному випадку в рамках моделі властивостей процес створення ПВК можна розглядати як діяльність, що складається в перетворенні множини  $P_c$  властивостей наслідуваних компонентів ПЗ до множини властивостей ПВК  $P_r$  відповідно до вимог  $R$ :

$$P_{rc} = A(P_c, R) , \quad (2.1)$$

де  $P_{rc}$  – множина значень властивостей ПВК;

$A$  – операція перетворення.

$P_c$  – множина значень властивостей компонента наслідуваного ПЗ;

$R$  – множина вимог до властивостей повторного використання ПВК.

Передбачається, що при систематичному повторному використанні ПВК створюється тільки на основі вимог, без вихідного компонента («з нуля») і поміщається в репозиторій, з якого він використовується як «чорна скринька» (black-box reuse), тому можна розглядати цей процес як звичайну розробку компонентів ПЗ ( $P_c = 0$ ). Тоді формула 2.1 має вигляд:

$$P_{rc} = A(R) , \quad (2.2)$$

де  $P_{rc}$  – множина значень необхідних властивостей ПЗ;

$A$  – операція перетворення.

$R$  – множина вимог до властивостей повторного використання ПВК;

Цей випадок у дослідженні не розглядається.

При несистематичний повторному використанні вихідним матеріалом для створення ПВК служать компоненти наслідуваного ПЗ, які, очевидно, вже мають деякі властивості, у тому числі й важливі з погляду повторного використання, тому можна розглядати процес розробки ПВК як модифікацію компонентів наслідуваного ПЗ під вимоги  $R$  (white- box reuse) без їх розміщення в репозиторії.

Очевидно, що трудомісткість перетворення наслідуваних компонентів ПЗ в ПВК залежить від «різниці» у значеннях властивостей множини  $P_c$  наслідуваного компонента і необхідного, що належить множині значень  $P_{rc}$  і задається  $R$ , ПВК і трудомісткості операцій усунення цієї «різниці». Під різницею можна розуміти різницю множин  $P_c$  і  $P_{rc}$  в теоретико-множинному сенсі (якщо розглядати наявність або відсутність властивості в множині) або різниця між значеннями відповідних властивостей однієї та іншої множини. Таким чином, витрати на створення ПВК визначаються близькістю множини властивостей  $P_c$  компонентів наслідуваного ПЗ до множини  $P_{rc}$  необхідних для ПВК і, як наслідок, витратами на операцію  $A$  щодо приведення множини значень вихідних

властивостей наслідуваного компонента  $P_c$  до множини значень властивостей ПВК  $P_{rc}$ . Критерієм вибору компонента ПЗ кандидатом в ПВК є така умова:

$$C(A(P_c, R)) < C(A(R)), \quad (2.3)$$

де  $C(A(P_c, R))$  – витрати на створення ПВК з наслідуваного ПЗ;

$C(A(R))$  – витрати на створення ПВК «з нуля».

Таким чином, при оцінюванні наслідуваних компонентів ПЗ завдання експерта полягає в тому, щоб визначити для кожного з них близькість множини властивостей  $P_c$  до множини властивостей ПВК  $P_{rc}$ , що визначаються множиною вимог  $R$ . Для визначення доцільності створення ПВК з даного компонента-кандидата експерт розробляє оцінювання трудовитрат, необхідних для зміни кожної властивості компонента-кандидата і оцінку сумарних трудовитрат:

$$C(A(P_c R)) = \sum_{i=1}^n C_i(A_i(P_{ci}, R_i)), \quad (2.4)$$

де  $C_i(A_i(P_{ci}, R_i))$  – трудовитрати на зміну  $i$ -ої властивості  $P_{ci}$  з множини  $P_c$  відповідно до вимоги  $R_i$ ;

$n$  – кількість властивостей множини  $P_c$ .

Спираючись на вищесказане, будемо застосовувати для визначення множини властивостей компонентів наслідуваного ПЗ, важливих з точки зору створення ПВК, тезу: важливість властивості вважається тим більшою, чим вищі затрати на її зміну в компоненті ПЗ у фазі створення ПВК.

Використання ПВК, якщо він розміщений в репозиторії, вимагає виконання процесів пошуку, адаптації, верифікації та вбудовування ПВК у нове ПЗ. Очевидно, що трудовитрати на здійснення цих процесів будуть тим меншими,

чим більше властивостей ПВК будуть відповідати вимогам цих процесів. Таким чином, придатність ПВК для повторного використання буде визначатися «повнотою» множини властивостей  $P_u$ , які є важливими з погляду використання ПВК та враховані при його створенні, і «різницею» значень цих і необхідних властивостей, тобто ступенем дотримання умови:

$$P'_{rc} = P_{rc}, \quad (2.5)$$

де  $P'_{rc}$  – множина значень властивостей конкретного ПВК;

$P_{rc}$  – множина значень властивостей ПВК, яке відповідає вимогам процесів його використання.

Недотримання цієї умови вимагає додаткових витрат на здійснення процесів використання ПВК в новому ПЗ, внаслідок проведення операцій, спрямованих на відповідне доопрацювання компонента (white-box reuse):

$$P'_{rc} = A(P_{rc}, R). \quad (2.6)$$

Спираючись на вищесказане, будемо застосовувати для визначення властивостей компонентів наслідуваного ПЗ, істотних з погляду повторного використання в аспекті процесів використання ПВК в новому ПЗ, тезу: важливість властивості зростає з підвищенням витрат, необхідних на доопрацювання ПВК.

Таким чином, для визначення властивостей компонентів ПЗ, важливих з точки зору повторного використання як у фазі створення, так і у фазі використання ПВК, можна застосовувати їх оцінку з точки зору впливу на вартість реалізації відповідних процесів. Спираючись на це, будемо визначати множину властивостей  $P_{rcp}$ , істотних з погляду повторного використання,



наступним чином:

- для кожної властивості з множини  $P_{cp}$  (ЗС-модель властивостей, рис.2.2) проведемо оцінювання її впливу на витрати в аспектах створення та використання ПВК;

- сформуємо множину властивостей  $P_{rc}$  шляхом вибору з множини  $P_{cp}$  тих властивостей, які важливі в аспектах створення та використання ПВК.

Ідентифікація компонентів. Для етапу ідентифікації фази створення ПВК, властивості корисності та спільності не є важливими, оскільки не впливають на витрати щодо ідентифікації та вибір компонентів. Вартість процесу ідентифікації (як у процесі виділення, так і в процесі вибору) буде знижуватися з ростом примітивності, модульності, структурованості і зрозумілості компонентів ПЗ, внаслідок спрощення виділення меж і сприйняття структури компонента. Наявність властивості закінченості контексту спрощує виділення компонента, оскільки забезпечує його чітке визначення.

Кваліфікація ПВК. Жодна з властивостей компонента-кандидата не є суттєвою для етапу кваліфікації, оскільки не впливає на вартість відповідних процесів.

Властивості впливають на вартість процесів кваліфікації наступним чином:

- достатність і закінченість – спрощують генерацію функціональних специфікацій і тестових варіантів, оскільки забезпечують повну передачу функціональності концепції компонента в його конструкції;

- примітивність – спрощує специфікацію, генерацію тестових варіантів та класифікацію за рахунок звуження функціональності компонента;

- модульність – спрощує процес упаковки і збереження ПВК, оскільки забезпечує його інкапсуляцію;

- структурованість - спрощує процес специфікації компонента, оскільки полегшує розуміння його структури;

- трасування – спрощує процеси специфікації, генерацію тестових варіантів та класифікації, оскільки забезпечує ефективне перенесення інформації між фазами життєвого циклу ПЗ;
- класифікованість – спрощує класифікацію компонента;
- компактність компонента – спрощує його упаковку і зберігання;
- зрозумілість – спрощує всі процеси, крім упаковки і збереження, оскільки зменшує трудовитрати на читання та розуміння компонента;
- коректність – спрощує генерацію тестових варіантів через зниження кількості виявлених при цьому помилок;
- тестопридатність – спрощує генерацію тестових варіантів.
- закінченість – спрощує специфікацію та генерацію тестових варіантів, оскільки забезпечує повний інтерфейс компонента.

Використання ПВК в новому ПЗ. Для фази використання ПВК в новому ПЗ важливою є його корисність, оскільки вона впливає на ефективність пошуку компонента, що задовольняє вимогам користувача.

Властивості впливають на процеси використання наступним чином:

- достатність і закінченість – спрощують модифікацію, верифікацію і вбудовування ПВК, оскільки забезпечують мінімум доопрацювань ПВК в рамках цих процесів;
- примітивність – спрощує модифікацію, верифікацію і вбудовування ПВК за рахунок зменшення його складності;
- модульність – спрощує модифікацію, верифікацію і вбудовування ПВК, оскільки забезпечує можливість ефективної роботи з частинами ПВК;
- структурованість – спрощує процеси модифікації та верифікації ПВК, оскільки забезпечує краще розуміння конструкції компонента;
- семантична і синтаксична незалежність – спрощують вбудовування ПВК внаслідок відсутності необхідності їх доопрацювання;

- трасування – спрощує модифікацію і верифікацію компонента внаслідок кращого перенесення інформації між фазами ЖЦ;
- переносимість – спрощує модифікацію і вбудовування ПВК, оскільки дозволяє налаштовувати та інтегрувати ПВК з мінімальними доопрацюваннями;
- зрозумілість – знижує вартість всіх процесів використання ПВК за рахунок менших трудозатрат на сприйняття ПВК;
- класифікованість – спрощує пошук ПВК, оскільки забезпечує критерії пошуку;
- компактність – спрощує інтеграцію ПВК за рахунок зниження вимог до ресурсів;
- простота модифікації – спрощує модифікацію ПВК;
- коректність компонента – спрощує модифікацію, верифікацію та інтеграцію ПВК, оскільки забезпечує правильне функціонування компонента;
- тестопридатність – спрощує верифікацію ПВК, забезпечуючи застосування тестів для реалізації цього процесу.
- властивість закінченості інтерфейсу – позитивно впливає на процеси верифікації та інтеграції ПВК, оскільки забезпечує повний інтерфейс для доступу до функціональності компонента.

Результати оцінювання наведені в таблиці 2.1. Як видно, на процеси створення та використання ПВК в новому ПЗ не впливають властивості спільності, надійності та ефективності компонентів ПЗ, оскільки від них не залежить вартість жодного з процесів. Ці властивості є суттєвими для процесу функціонування ПЗ, побудованого на основі ПВК. Найбільш «широким» впливом володіють властивості примітивності, модульності, структурованості, трасування, зрозумілості, коректності та закінченості. Не впливають на аналізовані процеси повторного використання властивості спільності, надійності та ефективності.

Таблиця 2.1

## Вплив властивостей компонентів ПЗ на процеси повторного використання

Властивості компонента наслідуваного ПЗ	Процеси									
	Створення ПВК						Використання ПВК в новому ПЗ			
	Ідентифікація		Кваліфікація							
	Визначення компонентів	Вибір компонентів	Специфікація	Генерація тестових варіантів	Класифікація	Упаковка та збереження	Пошук	Модифікація	Верифікація	Інтеграція
В аспектах 3С-моделі										
Концепції										
Користність							+			
Загальність										
Конструкції										
Достатність			+	+				+	+	+
Закінченість конструкції			+	+				+	+	+
Примітивність	+	+	+	+	+			+	+	+
Модульність	+	+				+		+	+	+
Структурованість	+	+	+					+	+	
Семантична незалежність										+
Синтаксична незалежність										+
Трасування			+	+	+			+	+	

Властивості компонента наслідуваного ПЗ	Процеси									
	Створення ПВК						Використання ПВК в новому ПЗ			
	Ідентифікація		Кваліфікація							
	Визначення компонентів	Вибір компонентів	Специфікація	Генерація тестових варіантів	Класифікація	Упаковка та збереження	Пошук	Модифікація	Верифікація	Інтеграція
Надійність										
Ефективність										
Переносимість								+		+
Класифікованість					+		+			
Компактність						+				+
Зрозумілість	+	+	+	+	+		+	+	+	+
Простота модифікації								+		
Коректність				+				+	+	+
Тестопридатність				+					+	
Контекст										
Семантична незалежність										+
Синтаксична незалежність										+
Закінченість інтерфейсу	+		+	+					+	+

### 2.3. Моделі властивостей

Запропонований метод експертного оцінювання властивостей компонентів наслідуваного ПЗ, важливих з погляду повторного використання, для вирішення задачі автоматичного обчислення значень оцінок властивостей вимагає визначення зв'язків між оцінками і прямими метриками, який встановлюються за допомогою моделей.

#### 2.3.1. Застосування системного підходу для побудови моделей властивостей

Застосування системного підходу до розробки системи величин. Всі властивості, важливі з точки зору повторного використання, одержані у дослідженні, не можуть бути прямо виміряні, тому відповідно до PVB - підходу потрібно деталізувати вихідні властивості компонентів і визначити величини (властивостей), які дозволяють асоціюватися з набором значень (шкалою). Для вирішення цієї задачі застосовується системний підхід, який дозволяє виявити системне ціле, його складові частини, залежності між ними [26]. При застосуванні системного підходу виходять з ієрархічної багаторівневої організації будь-якого об'єкта, що відображає його уявлення як множину його складових (деталей, ознак).

На визначення величин, придатних для вимірювання та оцінювання властивостей ПЗ впливають два основних фактори [26]:

- адекватність величини заданій властивості ПЗ;
- простота одержання значення величини.

Перший фактор обумовлюється вимогою найбільш повного і адекватного кількісного відображення відповідної властивості ПЗ за допомогою обраної величини.

Другий фактор обумовлюється необхідністю отримувати значення метрики для конкретних об'єктів, і пов'язаний з типом величини, обраної як метрики, а також способом визначення значень цієї величини. Як відомо, величини можуть бути одного з трьох основних типів – номінального, порядкового або адитивного.

Операція вимірювання може бути застосована тільки для величин адитивного типу, що накладає обмеження на вибір величин в моделях, орієнтованих на використання мір.

Системний підхід, дозволяє шляхом застосування системного аналізу виявити властивості, які відображають невимірювані властивості компонента ПЗ. Наприклад, властивість «читабельність тексту програми» можуть відображати такі властивості, як структурованість тексту, стиль, коментування. Ці властивості теж невимірювані і, в свою чергу, можуть знайти відображення на наступному рівні. Наприклад, коментування може відбиватися його властивостями: обсяг, щільність, наявність заголовків для кожної функції програми. Цей дедуктивний процес призводить до побудови ієрархічної системи пов'язаних властивостей [25], які деталізуються доти, поки на нижньому рівні ієрархії не будуть отримані вимірювані властивості [12].

Побудова ієрархічної моделі властивості шляхом системного аналізу веде до того, що на верхньому рівні ієрархії моделі знаходиться величина, що характеризує цільові властивості об'єкта, але непридатна для отримання значень прямим виміром, а на нижньому рівні виявляються величини, які можуть бути виміряні, але мають вузьку семантику у порівнянні з цільовою властивістю. Ефективність такої моделі властивостей буде залежати від того, наскільки адекватно встановлені відношення (закономірності) між її елементами, що знаходяться на сусідніх рівнях ієрархії. Для встановлення адекватності моделі властивостей необхідно перевірити зв'язки за допомогою даних, отриманих експериментально або шляхом спостереження.

Структура моделі. Результатом такого системного аналізу є модель властивостей  $P_{ij}$  та асоційованих з ними величин (метрик)  $M_{ij}$ , структура яких наведена на рис.2.7. Слід зауважити, що така модель повинна володіти певною якістю, яка характеризує її придатність до оцінювання властивостей відповідних об'єктів ПЗ.

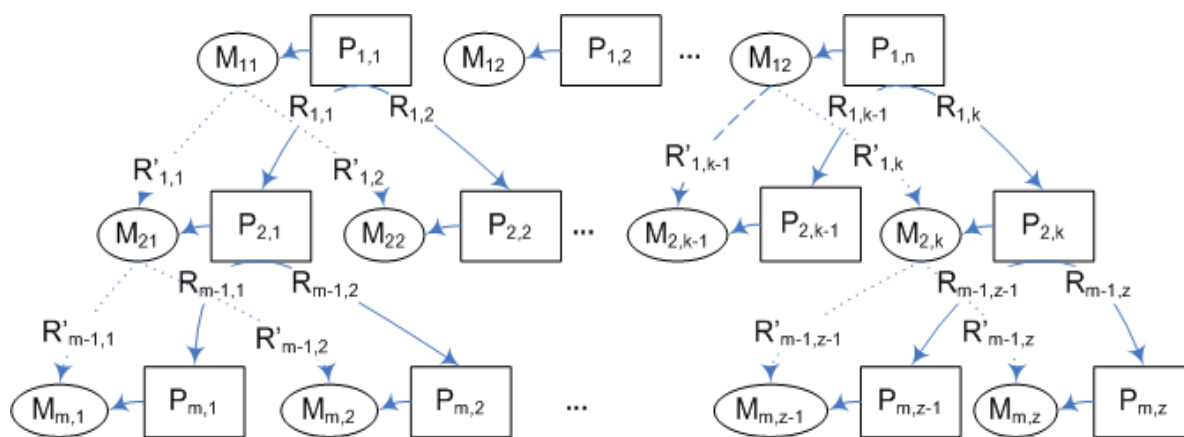


Рис. 2.7. Структура моделі властивостей

З урахуванням адекватності можна вказати наступні властивості моделі в аспекті якості:

- адекватність моделі щодо властивості;
- повнота розкриття властивості верхнього рівня властивостями нижніх рівнів;
- детермінованість і точність опису зв'язків між елементами моделі;
- придатність метрик для автоматичного отримання значень.

Для синтезу кожної величини в такій моделі необхідно визначити семантику величини, систему значень (шкалу) і спосіб порівняння значень з величиною.

Передбачається, що в моделі, величини можуть розташовуватися на наступних рівнях – цільовому, проміжному і вимірювання (рис.2.8). Проміжний рівень може вводитися для можливості додаткової структуризації системи величин.

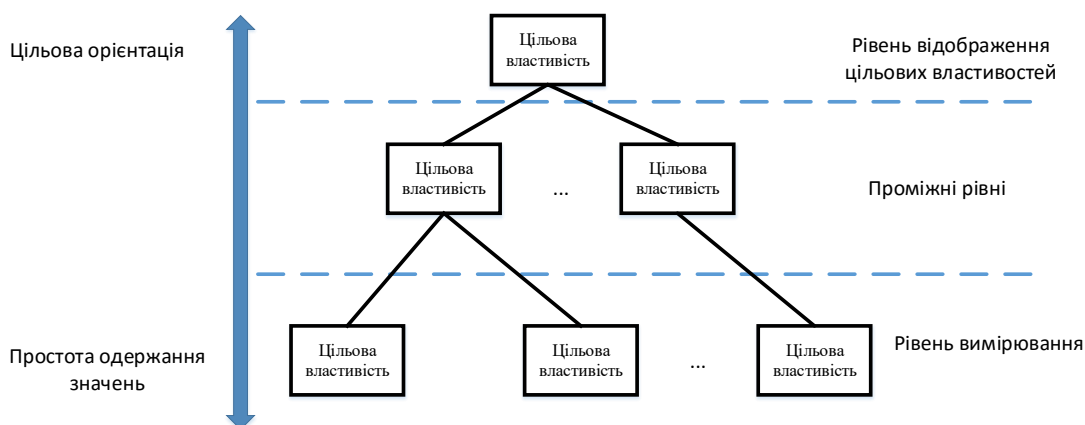


Рис. 2.8. Багаторівнева модель властивості компонентів ПЗ



### 2.3.2. Моделі властивостей, важливих в аспекті повторного використання

Таким чином, багаторівнева ієрархічна модель властивостей дозволяє будувати моделі властивостей, важливих з погляду повторного використання, для застосування в пропонованому методі. Для цього необхідно для кожної оцінюваної властивості з множини  $\{e^j_j / j=1..w\}$  сформуувати набір з  $n$  пов'язаних з ним метрик, розподілених по  $l-1$  рівнях  $\{m^k / k = 1 \dots n, k = 2 \dots l\}$ .

Для реалізації запропонованого методу будуються дворівневі моделі властивостей, важливих з погляду повторного використання. Перший рівень – рівень властивостей, що оцінюються, другий – рівень метрик, значення яких вимірюються вимірювальним пристроєм (рис.2.9).

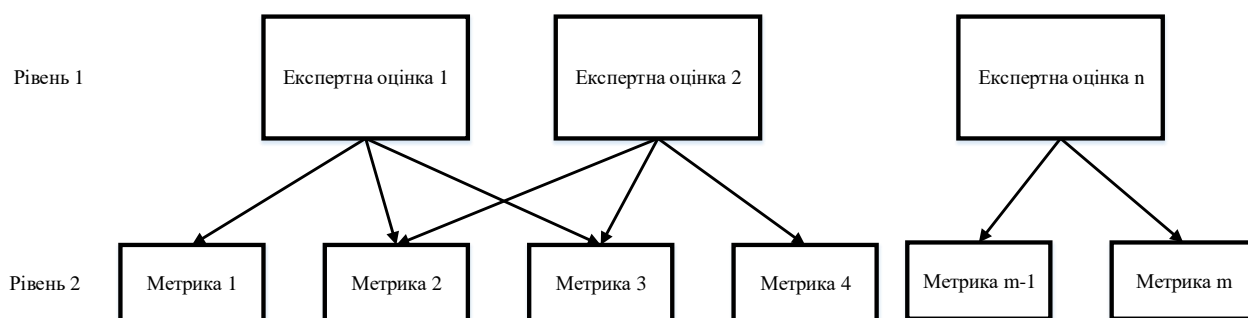


Рис. 2.9. Дворівнева модель властивості

Для реалізації системного аналізу та побудови моделей властивостей як залежностей між  $\{e_i \in E / j=1..w, m^k_i \in M / i=1..n, k=2..l\}$  у дослідженні запропоновано використовувати метод GQM («Мета-Питання-Метрика») [ ]. На першому рівні розташовуються властивості компонентів ПЗ, визначені як важливі для повторного використання (таблиця 2.1). На другому рівні розташовуються метрики, які вибираються зі списку, представленого в додатку Б. Це список сформований шляхом аналізу та відбору метрик, значення яких можуть бути отримані шляхом прямого вимірювання вихідних текстів програм (рис.2.10) [20].

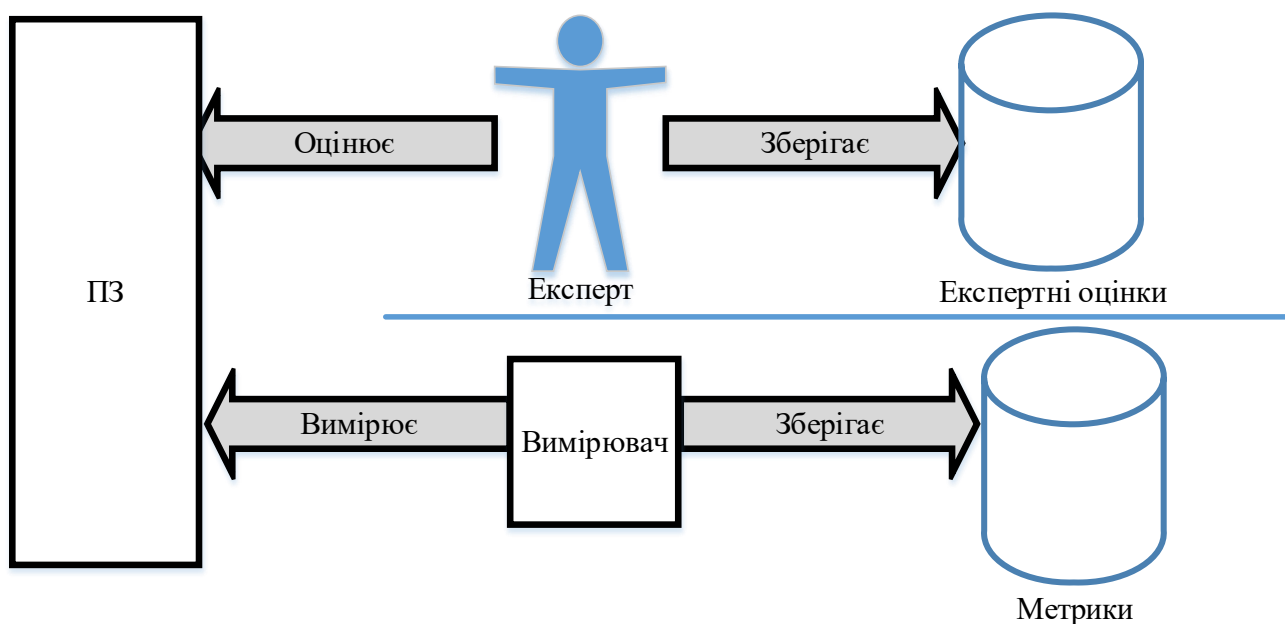


Рис. 2.10. Застосування дворівневої моделі властивостей

Відповідно до методу GQM процес аналізу формують три наступних кроки [37]:

- формулюються цілі, які повинні бути досягнуті шляхом оцінювання властивостей компонентів ПЗ;
- формулюються питання для кожної цілі, на які необхідно отримати відповідь для визначення того, чи досягається мета;
- визначаються метрики, які можуть бути використані для відповідей на питання, визначені на другому кроці.

Як приклад застосуємо методологію GQM для побудови моделі властивості «зрозумілість» компонента ПЗ. Ця властивість є одним з найважливіших і впливає, як видно з таблиці 2.1, практично на всі процеси повторного використання.

Крок 1. Мета формулюється наступним чином: «Оцінити зрозумілість і читабельність вихідного компонента наслідуваного ПЗ для розробника ПВК».

Крок 2. Питання. Для визначення того, чи досягається сформульована мета, шляхом аналізу аспектів конструкції компонентів ПЗ, що впливають на читабельність і зрозумілість тексту програм [21], формується перелік питань (стовбець 1 таблиця 2.2).

Крок 3. Метрики. З кожним питанням асоціюється, шляхом вибору з вихідної множини (додаток Б) набір метрик, які повинні забезпечувати відповіді на поставлені питання (стовбець 2 таблиці 2.2).

Таблиця 2.2

## Ціль, питання і метрики для властивості «Зрозумілість»

Ціль	Питання	Метрика		
		Семантика	Назва	№ метрики
Оцінити зрозумілість компонента-кандидата в ПВК	1.1 Наскільки добре коментований текст програми?	Відсоток модулів, що мають коментовані заголовки	Modules with Headers	45
		Відсоток модулів, які мають коментарі між заголовком и кінцем модуля	Modules with Comments	46
		Середній відсоток стрічок коментарів у всіх модулях	Ave. Percent Comments within a Module	51
	1.2.Наскільки добре структурований текст програми компонента?	Відсоток модулів, що мають горизонтальні відступи	% Mods with Horizontal Spacing	47
		Відсоток модулів, які мають вертикальні відступи	% Mods with Vertical Spacing	48
		Середній відсоток змістовних назв змінних у модулях	Ave. % Meaningful Vars within a Module	52
	1.3.Наскільки простий і добре структурований компонент?	Загальна цикломатична складність	Total V(G)	15
		Середня кількість вихідних конструкцій в модуле	Ave NCSS	14
		Загальні зусилля за Хальстедом	Total Hal's E	21
		Максимальна структурна вкладеність	Max Nesting	27

У результаті реалізації кроків для властивості «зрозумілість» одержана дворівнева модель величин, яка показана на рис.2.11.

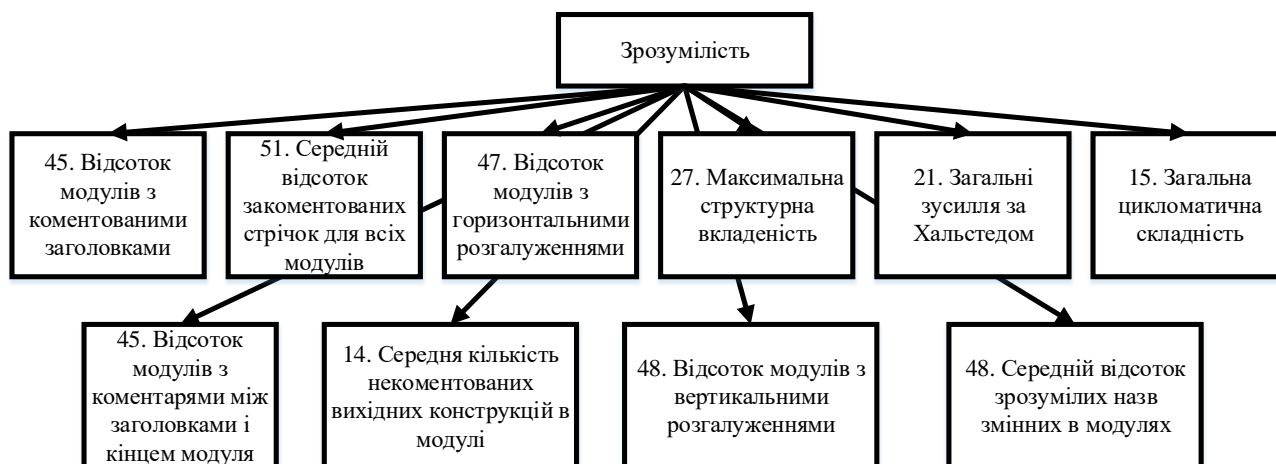


Рис. 2.11. Система метрик для властивості «Зрозумілість»

Моделі для важливих властивостей «примітивність», «модульність» і «структурованість», отримані таким же шляхом за допомогою методу GQM, наведені у додатку В.

#### 2.4. Висновки до розділу

У дипломній роботі магістра об'єктом експертного оцінювання є компонент ПЗ, який представляється властивостями, що відображають його потенційну повторно використовуваність. Реалізація експертного оцінювання компонентів ПЗ вимагає виконання двох процесів: оцінювання властивостей, важливих в аспекті повторного використання; оцінювання придатності компонента - кандидата в ПВК і ухвалення відповідного рішення про створення ПВК або «відкиданні». Найбільш трудомістким є перший процес. Беручи до уваги можливість виконання вимірювань в ПЗ, у дослідженні пропонується оцінки усіх або деяких властивостей одержувати автоматично шляхом обчислення і тим самим виключити частково або повністю процес оцінювання властивостей компонента ПЗ.

Так як, обчислення значень властивостей компонентів, і, тим самим автоматизація експертного оцінювання, повинно проводитися на основі моделей властивостей, то необхідно забезпечити формалізацію і адекватність цих моделей. У дипломній роботі магістра пропонується уточнювати їх за допомогою досвідчених експертів, що володіють знаннями про повторне використання. Для цього за допомогою відповідних засобів виконуються процеси безпосереднього експертного оцінювання та вимірювання компонентів ПЗ, і уточнення моделей з використанням статистичних методів (первинного статистичного, кореляційного і регресійного аналізу).

Множину властивостей компонентів ПЗ, важливих з погляду повторного використання, визначається в аспектах 3С-моделі компонентів ПЗ, в контексті процесів повторного використання. Будуються дворівневі моделі властивостей шляхом застосування методу GQM.

У даному розділі одержано наступні результати:

1. Запропоновано метод експертного оцінювання властивостей компонентів ПЗ - кандидатів у ПВК, що дає змогу автоматизувати процес одержання значень властивостей шляхом їх обчислення за результатами вимірювань з використанням моделей властивостей.

2. Визначено множину властивостей компонентів ПЗ, важливих з погляду повторного використання, шляхом формування вихідної множини властивостей за допомогою 3С-моделі компонентів ПЗ і вибору тих з них, які впливають на витрати, пов'язані з виконанням процесів повторного використання;

3. Визначено спосіб побудови моделей властивостей на основі системного підходу шляхом застосування системного аналізу та методу GQM; побудовано моделі властивостей «зрозумілість», «примітивність», «модульність», «структурованість».

## РОЗДІЛ 3

### РОЗРОБКА АРХІТЕКТУРИ ЗАСОБІВ ПІДТРИМКИ МЕТОДУ ЕКСПЕРТНОГО ОЦІНЮВАННЯ ВЛАСТИВОСТЕЙ ПОВТОРНО ВИКОРИСТОВУВАНИХ КОМПОНЕНТІВ ТА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

3.1. Задачі та архітектура засобів оцінювання властивостей повторно використовуваних компонентів програмного забезпечення

Для забезпечення автоматизації розробленого методу оцінювання властивостей компонентів ПЗ, важливих з погляду повторного використання, запропоновано архітектуру відповідних засобів. Ці засоби повинні забезпечувати вирішення двох основних задач:

- дослідження адекватності моделей властивостей і уточнення залежностей між властивостями компонентів-кандидатів у ПВК і прямо вимірюваними метриками;

- автоматизацію процесів експертного оцінювання компонентів ПЗ - кандидатів у ПВК.

Вирішення першої задачі відбувається у відповідності із розробленою методикою і повинно забезпечуватися такими функціями засобів:

- підтримка експертного оцінювання властивостей компонентів ПЗ і збереження оцінок в БД;

- одержання значень метрик компонентів ПЗ шляхом вимірювання вихідних текстів програм і збереження в БД;

- уточнення залежностей між оцінками і метриками на основі отриманих даних.

Вирішення другої задачі реалізується двома процесами (рис.2.2) – оцінюванням властивостей компонента і прийняттям рішення про створення ПВК або «відкиданні» компонента. Послідовність виконання цих процесів показана на рис. 3.1. Для забезпечення цих процесів засоби повинні реалізовувати такі функції:

- вимірювання компонентів;
- обчислення значень властивостей компонентів на основі результатів вимірювань.

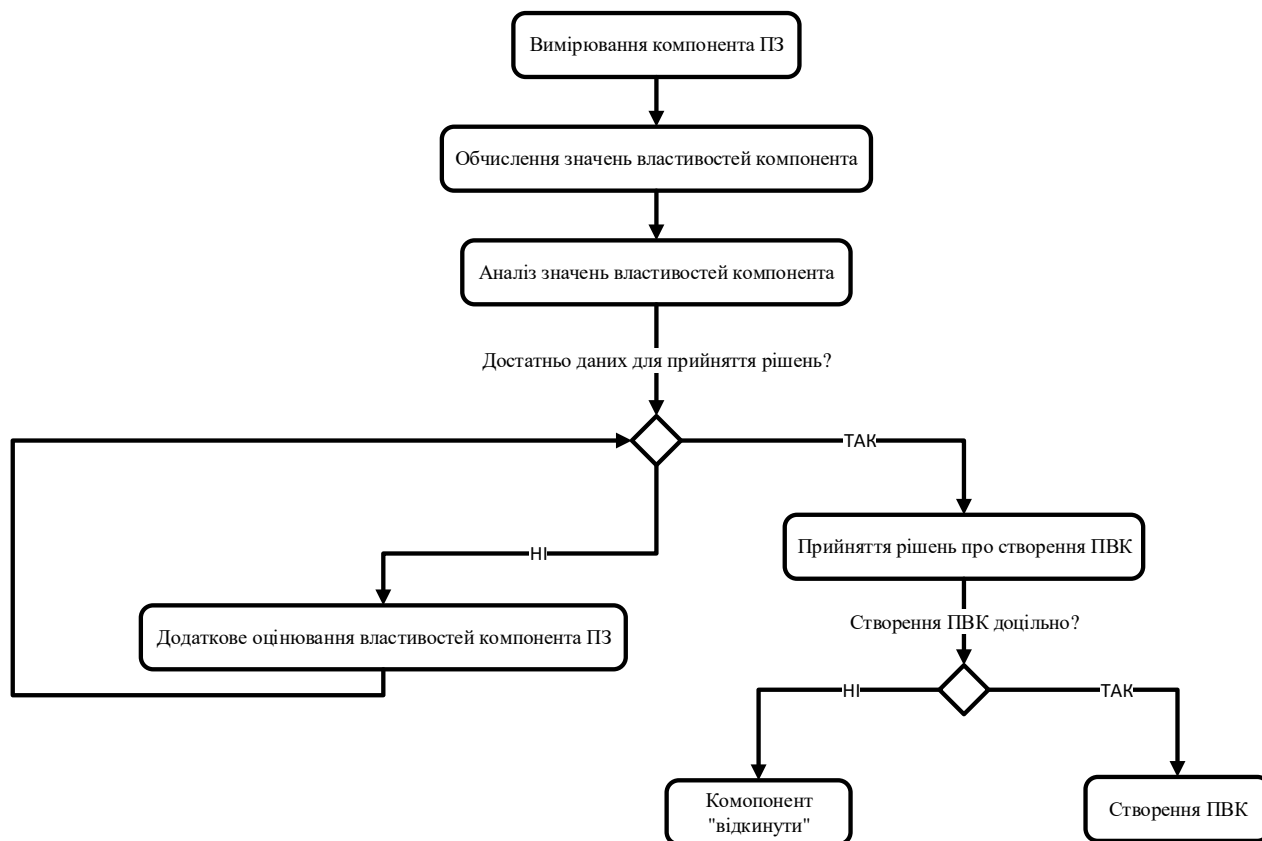


Рис. 3.1. Послідовність реалізації процесів оцінювання компонентів ПЗ

Для реалізації перерахованих вище функцій пропонується архітектура засобів, що включає наступні компоненти (рис.3.2):

- обчислювач значень властивостей – призначений для обчислення значень властивостей компонента за результатами його вимірювання;
- засоби аналізу даних – призначені для уточнення залежностей між оцінками і метриками властивостей компонентів на основі даних, одержуваних при їх оцінюванні та вимірі;
- інтерфейс експерта – призначений для забезпечення взаємодії експерта із засобами в діалоговому режимі;

- ПЗ керування – призначено для узгодження роботи всіх компонентів засобів;
- вимірювач – призначений для отримання значень метрик шляхом вимірювання вихідних кодів програм;
- база даних – призначена для зберігання наборів оцінок і метрик, результатів вимірювань, експертних оцінок компонентів ПЗ та службової інформації;
- база знань – призначена для зберігання знань про залежності між властивостями і метриками компонентів ПЗ, одержуваних у результаті аналізу даних.

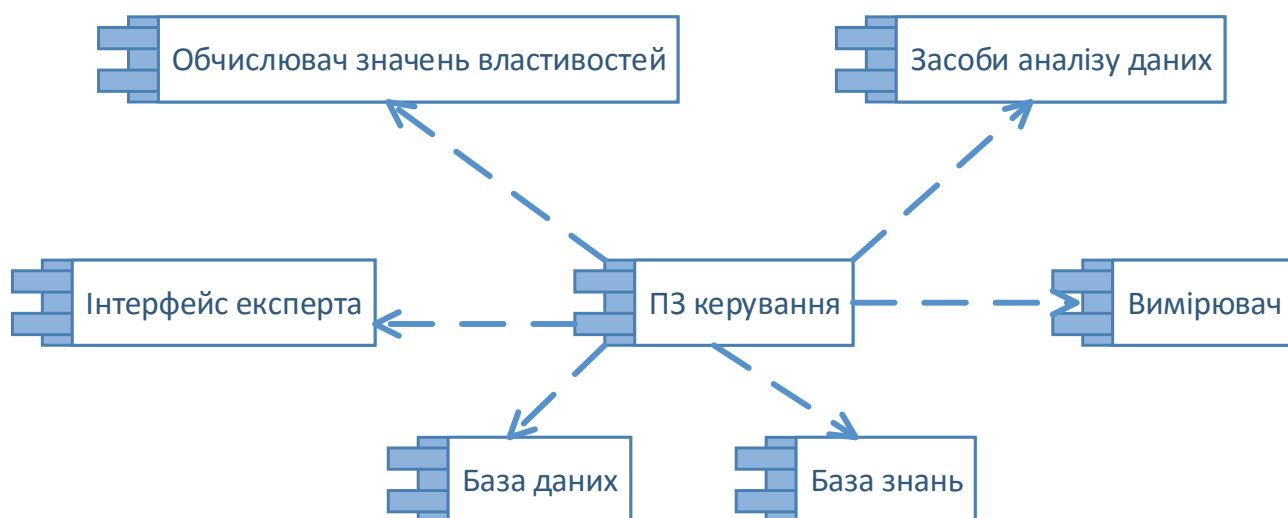


Рис. 3.2. Архітектура засобів оцінювання

Відповідно до запропонованої методики система повинна працювати в двох режимах:

- накопичення та аналізу даних – забезпечує проведення вимірювань та експертного оцінювання компонентів ПЗ, накопичення даних, їх аналіз та уточнення залежностей між властивостями і прямо вимірюваними метриками;
- оцінювання – отримання значень оцінок компонентів ПЗ на основі вимірних метрик і залежностей між властивостями і метриками.



У режимі накопичення та аналізу даних проводиться експертне оцінювання та вимірювання вихідних компонентів ПЗ, збереження значень в БД, з подальшим їх аналізом для уточнення залежностей між оцінками і метриками. Значення оцінок властивостей вводяться за допомогою інтерфейсу експерта. Цей же інтерфейс служить для управління процесом оцінювання. Значення прямо вимірюваних метрик визначаються автоматично за допомогою вимірювача вихідних текстів програм. Всі отримані значення оцінок і метрик зберігаються в БД. Уточнення залежностей проводиться за допомогою засобів аналізу даних, а результат цього уточнення зберігається в базі знань про залежності.

У режимі оцінювання компоненти ПЗ вимірюються за допомогою вимірювача вихідних текстів, а результати вимірювань подаються на обчислювач значень властивостей, який на основі залежностей між оцінками і метриками, наявних у базі знань обчислює значення властивостей компонентів. Обчислені значення оцінок відображаються інтерфейсом експерта.

## 3.2. Компоненти архітектури

Для реалізації архітектури засобів оцінювання властивостей ПВК розроблені компоненти, які наведені на рис. 3.2.

### 3.2.1. Компоненти реалізації експертного оцінювання

До компонентів, які реалізують експертне оцінювання, відносяться інтерфейс експерта, ПЗ керування і обчислювач значень властивостей.

Інтерфейс експерта забезпечує наступні основні функції:

- перегляд вихідних кодів компонентів ПЗ;
- введення і відображення значень експертних оцінок.

Крім того, інтерфейс повинен забезпечувати введення додаткових атрибутів для оцінюваних компонентів ПЗ. Функція перегляду вихідних кодів компонентів (модулів) ПЗ реалізується за допомогою текстового браузеру. Функція вводу і

відображення значень, які використовуються експертом, реалізується прикладним ПЗ, яке забезпечує:

- відображення набору оцінок і їх атрибутів;
- відображення списків можливих значень для кожної оцінки;
- вибір значення для кожної оцінки з набору і його збереження в БД;
- ввід додаткових атрибутів компонентів ПЗ і їх збереження в БД;
- відображення значень метрик для компонентів ПЗ, отриманих за допомогою вимірювача;
- відображення значень оцінок компонентів ПЗ, обчислених на основі значень метрик.

Зовнішній вигляд інтерфейсу експерта наведений у додатку В.

ПЗ керування забезпечує спільне використання всіх компонентів засобів, визначених у пропонованій архітектурі, і реалізує наступні функції:

- авторизація експерта – забезпечує ідентифікацію і авторизацію експертів перед початком роботи із засобами;
- управління роботою засобів – забезпечує задану логіку спільного використання засобів, у тому числі використання процедур авторизації, вибору компонентів ПЗ для роботи, запуску вимірювача, ввід значень оцінок, збереження атрибутів експериментів і т.п.;
- управління вимірювачем – забезпечує запуск вимірювача, передачу йому посилання на текст вихідний обраного компонента ПЗ та збереження його у БД вимірювання;
- адміністрування засобів – забезпечує ввід, редагування необхідної інформації для роботи засобів, у тому числі з метриками, оцінками і авторизованими експертами.

Логіка роботи вимірювача для режимів накопичення даних і оцінювання компонентів ПЗ показана на рис. 3.3 та 3.4.

Обчислювач значень оцінок забезпечує функцію обчислення значень оцінок на основі значень метрик. Для цього використовуються залежності, що зберігаються в базі знань.

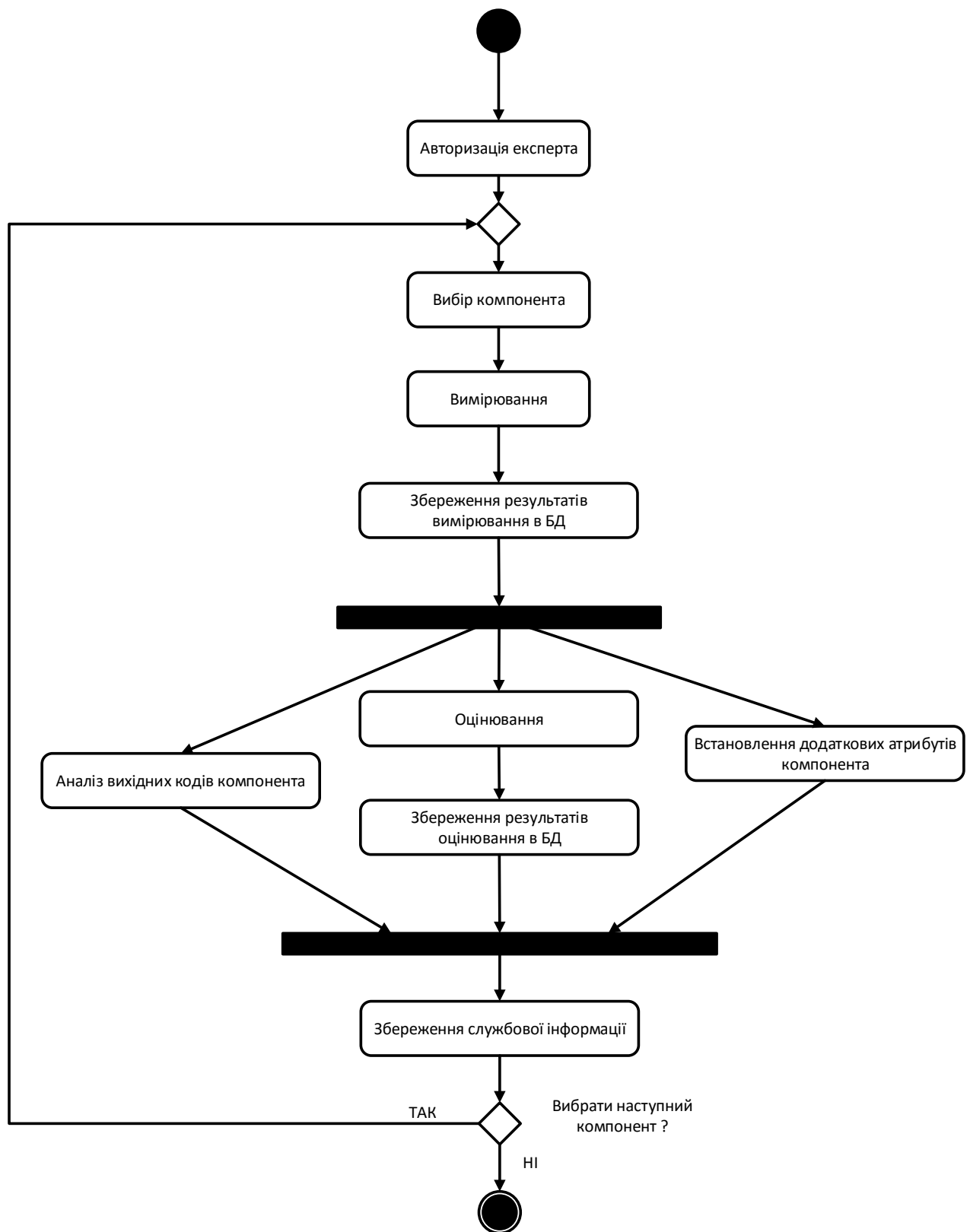


Рис. 3.3. Робота засобів в режимі накопичення даних

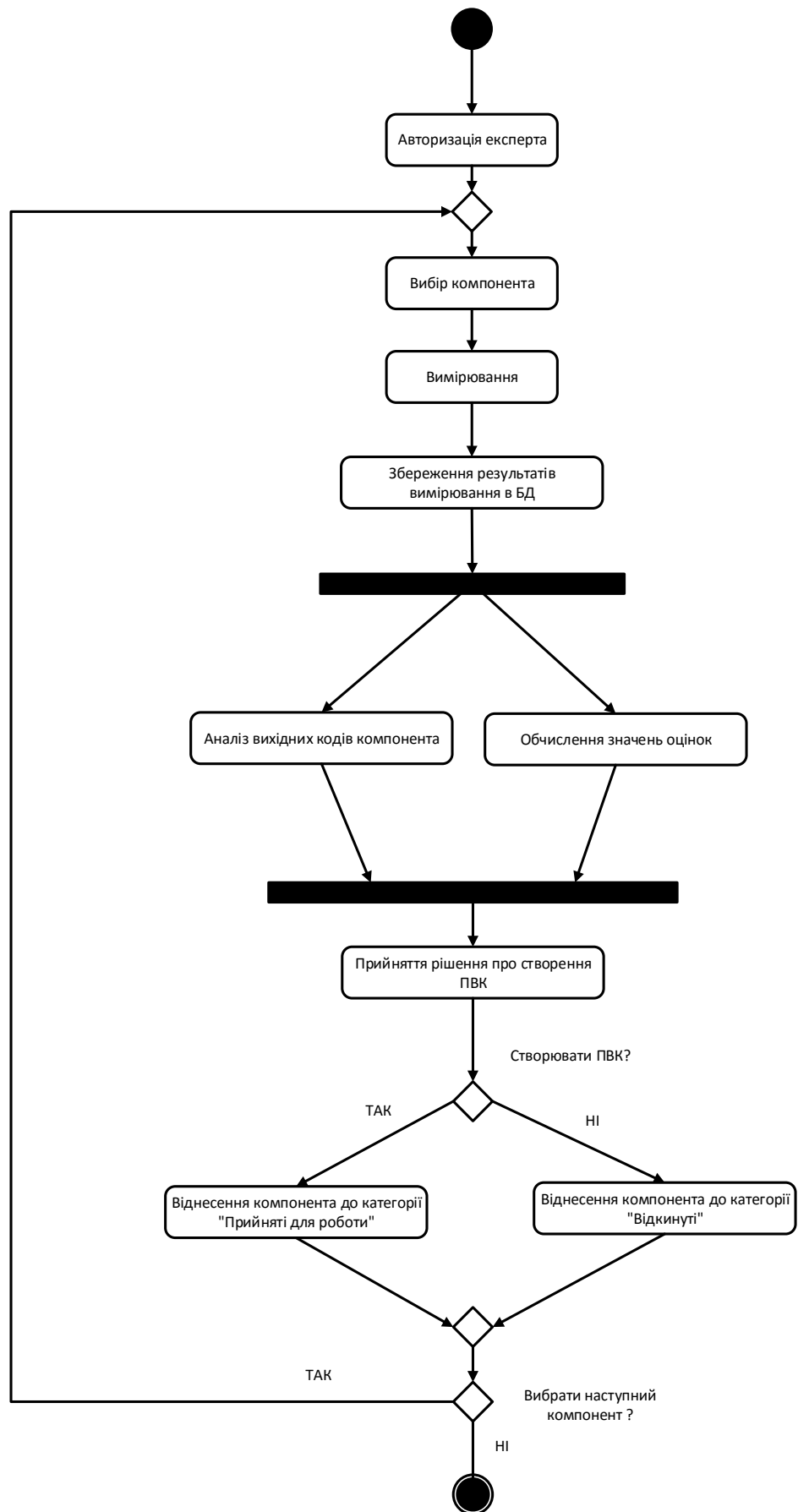


Рис. 3.4. Робота засобів в режимі оцінювання ПЗ

Обчислювач може бути побудований на різних принципах, наприклад, на обчисленні функцій багатьох змінних. У дипломній роботі пропонується використовувати для реалізації цього компонента нейронні мережі [6], які надають зручний апарат для автоматизованого виявлення та збереження залежностей між входами і виходами шляхом навчання на наборі даних. Зараз є різні види нейромереж (прямого і зворотного поширення, нейро-нечіткі і т.д.), які можуть бути використані для реалізації цього завдання [19]. Зазвичай, вид мережі підбирається, виходячи з відомих для них обмежень на умови їх застосування шляхом експериментального підбору. У дослідженні для реалізації обчислювача значень властивостей була обрана адаптивна мережа нечіткого виводу (Adaptive- Network-Based Fuzzy Inference System – ANFIS), обмеження на застосування якої можуть бути виконані при реалізації пропонованого методу.

ANFIS є варіантом гібридних нейро-нечітких мереж прямого поширення сигналу особливого типу. Архітектура нейро-нечіткої мережі ізоморфна нечіткій базі знань. У нейро-нечітких мережах використовуються реалізації диференційованих трикутних норм (множення та ймовірне АБО), а також гладкі функції приналежності, що дозволяє застосовувати для настройки нейро-нечітких мереж швидкі алгоритми навчання, засновані на методі зворотного поширення помилки. Оскільки множина значень експертних оцінок можуть приймати значення із заданих множин слів, то їх можна розглядати як лінгвістичні змінні, з відповідними множинами-термами [18].

### 3.2.2. Компоненти зберігання та аналізу даних

До компонентів зберігання та аналізу даних відносяться база даних, база знань і засоби аналізу.

База даних призначена для зберігання наборів оцінок і метрик, значень оцінок, що виставляються екпертом, і результатів вимірювань вихідних кодів програм. БД відображає наступні об'єкти (рис.3.5):

- експерти – реалізують експертне оцінювання компонентів ПЗ;

- експертні оцінки – опис експертних оцінок, які використовують експерти;
- значення експертних оцінок – наперед визначені значення, які асоціюються з експертними оцінками і вибраними екпертом при оцінюванні;
- модулі ПЗ – вихідні модулі ПЗ, підлягають оцінюванню на предмет створення ПВК;
- експерименти – відображають проведені експерименти з оцінювання компонентів ПЗ;
- метрики – опис метрик, застосовуються для виміру модулів ПЗ;
- значення метрик – значення вимірювань модулів ПЗ.



Рис. 3.5. Класи, що відображають БД

Крім того, в БД введені наступні додаткові сутності для довідкової інформації:

- кваліфікація експерта;
- клас програмного забезпечення;
- мова програмування.

Всі дані оцінювання і вимірювань асоційовані з конкретними модулями ПЗ та експертами, що проводили оцінювання. Сутності «оцінка» і «метрика» містять

атрибути, необхідні для використання метрик і оцінок в процесі оцінювання і вимірювання модулів ПЗ, а також подальшого аналізу даних. Для реалізації обрана реляційна база даних, схема якої наведена на рис.3.6 і реалізована за допомогою системи керування БД "Парадокс".

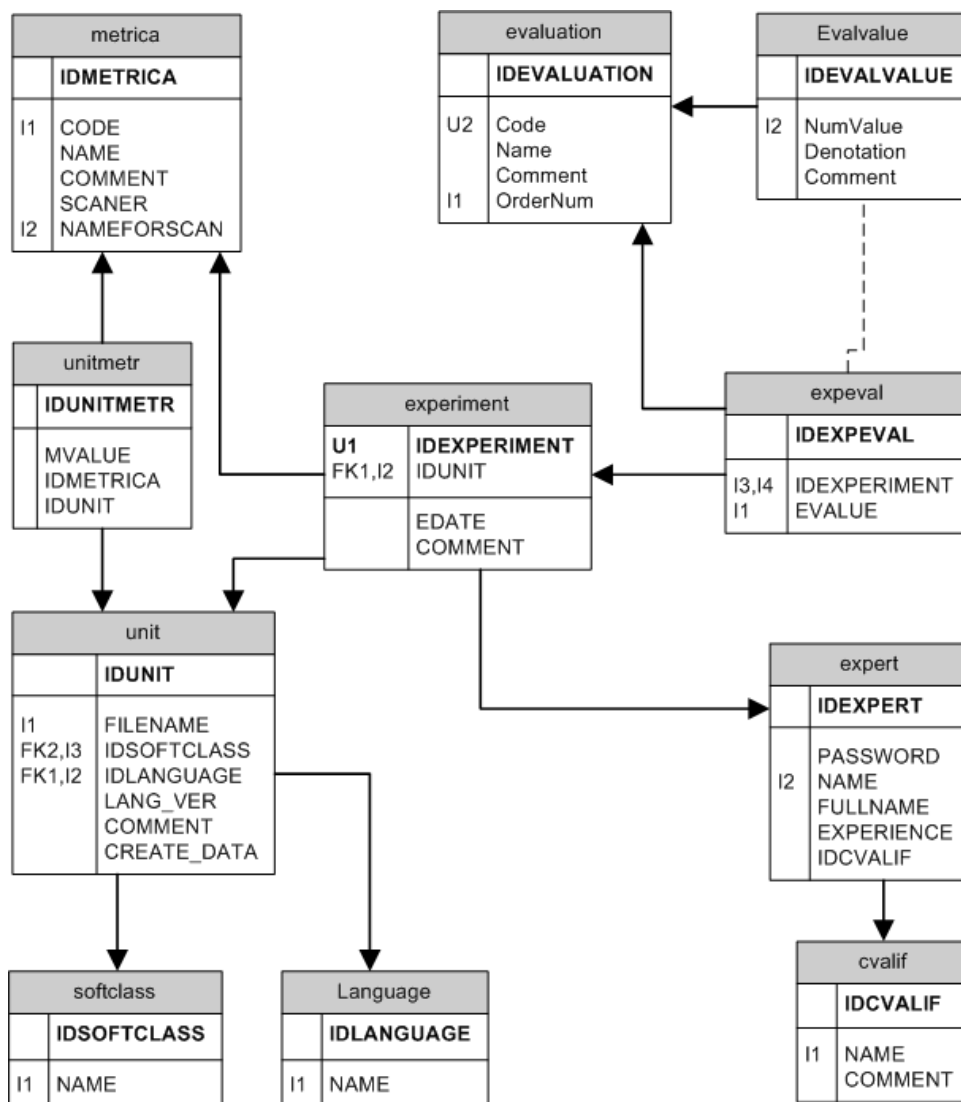


Рис. 3.6. Схема БД

Засоби аналізу даних і база знань призначені для реалізації наступних функцій:

- аналіз та уточнення залежностей між оцінками і метриками на основі експериментальних даних;

– зберігання залежностей між оцінками і метриками у формалізованому вигляді та їх редагування.

Оскільки завдання аналізу та зберігання залежностей тісно пов'язані з поданням даних, то компоненти «Засоби аналізу даних» та «База знань» розглядаються разом.

Засоби аналізу даних повинні вирішувати завдання отримання знань на основі експериментальних даних, що зберігаються в БД, тобто вирішувати завдання придбання знань, в даному випадку - про залежності між оцінками і метриками компонентів ПЗ. В експертних системах виділяють наступні стадії придбання знань [16]: ідентифікація, концептуалізація, формалізація, реалізація, тестування.

Розроблювані засоби аналізу даних повинні вирішувати завдання формалізації, реалізації та тестування знань.

База знань повинна забезпечувати зберігання знань про залежності у вигляді, визначеному на стадії розробки структури організації знань.

В принципі, запропонована загальна архітектура засобів оцінки компонентів ПЗ дозволяє використовувати різні методи аналізу даних, структуризації та уточнення залежностей, оскільки вихідні дані накопичуються в базі даних і зберігаються в реляційному представленні, яке не обмежує спосіб їх подальшої обробки. У роботі в якості структур для організації знань про залежності були обрані аналітичні функції, які визначаються та уточнюються, шляхом вирішення задачі відновлення регресії статистичними методами, а також апарат нейронних мереж і нечіткої логіки.

Для реалізації аналізу даних та уточнення залежностей була розроблена бібліотека модулів, призначених для функціонування в середовищі MathLab.

### 3.2.3. Вимірювач властивостей вихідного коду

Вимірювач вихідних текстів призначений для вимірювання вихідних кодів компонентів ПЗ. Існують вимірювачі програм, які входять до складу ряду



комерційних або вільно розповсюджуваних програмних продуктів, призначених для виробничих і дослідницьких цілей, однак, роботи, присвячені принципам побудови та функціонування вимірників ПЗ, практично відсутні. У роботі розроблені принципи роботи та структура вимірювача.

Вплив типу величини на її значення. Величина – це деяка властивість сутності, з якою можна порівняти значення. Для синтезу величини необхідно визначити властивість (семантику величини), систему значень (шкалу) і спосіб порівняння значень з величиною.

У теорії вимірювань виділяють три основні шкали [8] – номінальну, порядкову і кількісну. Номінальна (класифікаційна) шкала включає значення, які проявляють себе тільки у відношеннях еквівалентності, тобто можуть бути порівняними з властивістю сутності, однак не впорядковані одна відносно іншої. Наприклад, можна порівняти з вихідним текстом програми величину «мова програмування», значенням якої може бути назва однієї з мов (наприклад, «С», «С++», «Паскаль», «Ява» і т.д.). Такий же тип має шкала класифікації призначення модулів ПЗ (наприклад, «Бази даних», «Математичні пакети», «Операційні системи» і т.д.). До номінальних величин застосовується тільки операція перевірки на еквівалентність. Порядкова (ординальна) шкала передбачає впорядковані один відносно одного значення, до яких застосовуються операції порівняння. Порядкову шкалу можна задати для більшості експертних оцінок, наприклад, оцінки читабельності тексту програм – «незадовільно», «задовільно», «добре», «відмінно» або для оцінки рівня інкапсуляції програмних компонентів – «лексичний», «операторний», «процедурний», «класний», «модульний». Кількісна шкала включає значення, що проявляють себе у відношеннях еквівалентності, порядку та адитивності. Такі величини дозволяють застосовувати над значеннями адитивні і мультиплікативні операції (додавання, віднімання, множення, ділення). До них відносяться, наприклад, такі: число рядків коду, число рядків коментаря, оцінки трудовитрат на створення коду.

Очевидно, що найбільшу цінність з точки зору отримання та обробки значень представляють кількісні (адитивні) величини, оскільки вони не тільки

відображають властивості ПЗ, які можна використовувати для обробки та аналізу, але і володіють найбільш повним набором операцій над значеннями [11].

Отримання значень величин. Задача одержання значення величини, яка проявляє себе у відношенні еквівалентності, зводиться до задачі класифікації стану об'єкта вимірювання за заданими критеріями і порівняння формального об'єкта-позначення з цим станом. Для класифікації станів об'єкта вимірювання визначається сукупність ознак, значення яких дозволяють ідентифікувати кожний стан. Таким чином, для одержання значення номінальної величини необхідно мати засоби одержання значень відповідних ознак і пристрій класифікації, що забезпечує видачу значення величини (рис. 3.7).

Наприклад, при визначенні мови програмування, використаної при написанні програмного модуля, в якості ознаки може виступати розширення файлу модуля («С», «СРР», «ч», «па-», «Ада», «НТМ»). Датчик ознаки повинен виділяти розширення з імені файлу. Пристрій прийняття рішення повинен реалізовувати ідентифікацію значення на основі правил вигляду: «якщо Розширення = ".pas "або" .dpr ", то значення = " Паскаль "». Для отримання значення номінальної величини достатньо, щоб вибрані ознаки також володіли властивістю еквівалентності.



Рис. 3.7. Схема пристрою для одержання значень номінальних величин

Для роботи з порядковими величинами необхідна наявність хоча б однієї інтенсивної ознаки і застосування операцій порівняння у пристрої прийняття рішень. Наприклад, виберемо з набору ознак читабельності тексту множину з трьох величин {«відсоток рядків з відступами», «відсоток порожніх рядків», «середня довжина рядка»} і сформулюємо ряд правил вигляду: «" якщо відсоток рядків з відступами "> 20 і "відсоток порожніх рядків"> 10 і "середня довжина рядка" <30 символів, то значення = "задовільно" ».

Як видно з попередніх міркувань, для номінальних і порядкових величин можна говорити про «одержані значення», але не про «виміри». Власне вимір починається там, де можна визначити адитивну величину.

Оскільки, тексти програм мають дискретну природу, то для визначення значення адитивної величини необхідно мати одиничну міру, яка рівна кванту величини, і пристрій для відображення мір і підрахунку квантів. Наприклад, при вимірюванні довжин програм в якості міри може прийматися рядок, а в якості вимірювального пристрою – сканер тексту, що підраховує їх кількість.

Властивості номінальних і порядкових величин не відповідають вимогам вимірювань, тому для реалізації процесу вимірювання, в якості показників можна застосовувати тільки адитивні величини. Таким чином, процес об'єктивного визначення значення будь-якої величини повинен включати вимірювання адитивних величин.

Склад і структура вимірювача. Аналіз метрик ПЗ, які можна віднести до адитивних величин, показує, що для більшості з них значення обчислюються шляхом підрахунку об'єктів, які інтерпретуються як одиничні величини. У текстах програм об'єкти (рядок, оператор, операнд, вузол або дуга графа, програмна конструкція) виступають в якості одиниць величини і є одиничними носіями даної властивості. Наприклад, значеннями довжини програм можуть бути 100 рядків, 20 операторів або 10 підпрограм. Вищенаведені міркування, а також послідовний характер тексту програм, обумовлюють наступні операції, які повинен реалізовувати вимірювач:

- сканування тексту програми;

- виділення заданих об'єктів;
- підрахунок виділених об'єктів.

Структура вимірювача показана на рис. 3.8.

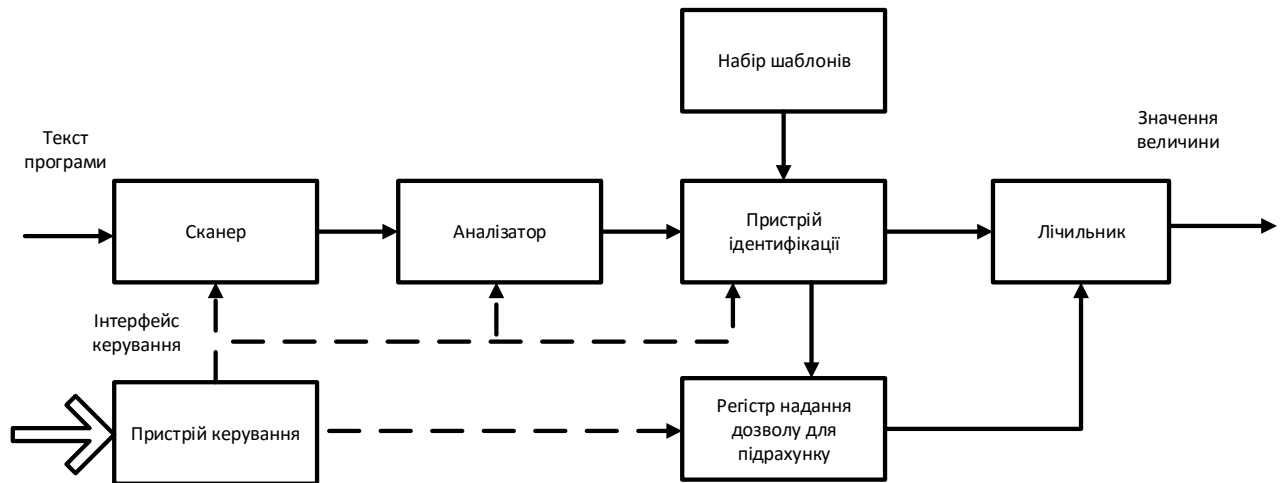


Рис. 3.8. Структура вимірювача

Сканер відіграє роль пристрою послідовного зчитування символів програми [14]. На інформаційному виході сканера послідовно з'являються коди символів тексту; керуючі лінії (показані на рис.3.8 пунктиром) забезпечують обмін старто-стопного сигналів з пристроєм управління.

Аналізатор призначений для виділення програмних конструкцій, що знаходяться на відповідному рівні розгляду програми. Для виділення об'єктів необхідно використовувати граматики як засіб відтворення одиначної величини (міри). Часто в метриці обмежується область визначення чи використання об'єктів, наприклад, «кількість локальних змінних» або «кількість змінних у структурах». Це вимагає не тільки ідентифікації самої конструкції, але і її контексту. Виходом аналізатора є потік ідентифікаторів програмних конструкцій. Управляючі лінії призначені для конфігурування аналізатора через пристрій керування.

Пристрій ідентифікації призначений для ідентифікації об'єктів підрахунку. Параметри ідентифікації задаються за допомогою шаблонів об'єктів. Набір шаблонів включає в себе опис, необхідний для ідентифікації відповідних об'єктів в тексті програми.

Лічильник призначений для підрахунку і зберігання значення кількості одиничних сигналів на виході пристрою ідентифікації. Управляючі лінії служать для скидання лічильника.

Регістр дозволу підрахунку призначений для фіксації початку і кінця частин програми, які відповідають існуванню об'єктів, що утворюють контекст для ідентифікованих атомарних об'єктів.

Пристрій управління відіграє роль конфігуратора і координуючого компонента пристрою.

Інтерфейс вимірювача. Інтерфейс вимірювача повинен забезпечувати обмін інформаційними і керуючими сигналами із зовнішнім середовищем. Інтерфейс здійснює обмін наступними категоріями даних (рис 3.9.): Текст програми, значення вимірюваних величин, керуючі дані.

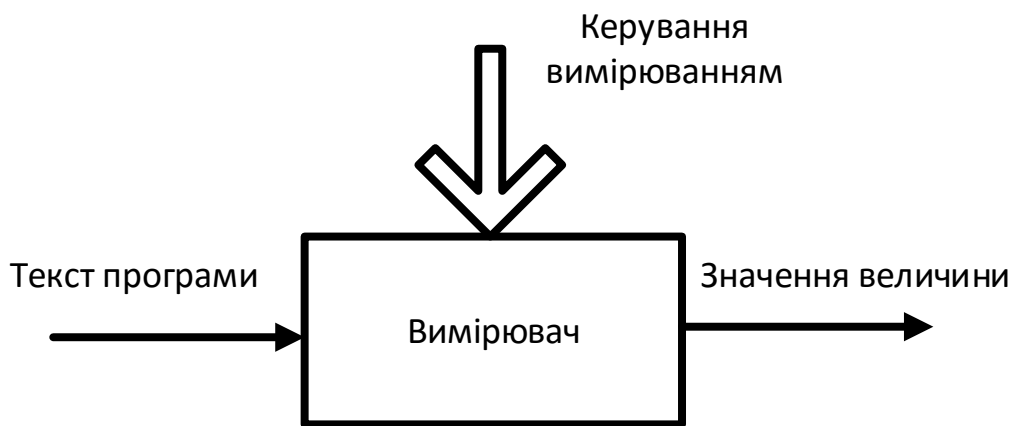


Рис. 3.9. Інформаційна взаємодія вимірювача

Для забезпечення вимірювання заданої величини вимірювач повинен отримати наступні вхідні дані:

- місце розташування, ідентифікатор і формат вимірюваного об'єкта (файлу);
- ідентифікатор класу атомарних об'єктів, що підлягають підрахунку;
- ідентифікатор класу об'єктів, що визначають область підрахунку.

Після задання всіх необхідних даних, що визначають вимір, пристрій приводиться в роботу. Після закінчення вимірювання вимірювач повинен видати сигнал завершення вимірювання та значення величини.

Перелік даних і сигналів інтерфейсу вимірювача дані в табл. 3.1.

Таблиця 3.1

### Дані та сигнали інтерфейсу вимірювача

№ з/п	Назва даних і сигналів	Напрямок передачі	Зовнішнє джерело / замовник	Інформація
1	Вхідний текст	Вхід	Текстовий файл	Текст вимірюваної програми
2	Значення величини	Вихід	Замовник вимірювання	Значення величини, що визначається
3	Назва вхідного файлу	Вхід	Замовник вимірювання	Атрибути файлу з текстом програми
4	Об'єкти підрахунку	Вхід	Замовник вимірювання	Ідентифікатор (опис) об'єктів підрахунку
5	Область підрахунку	Вхід	Замовник вимірювання	Ідентифікатор (опис) об'єктів програми, що містить об'єкти підрахунку
6	Старт	Вхід	Замовник вимірювання	Сигнал запуску вимірювання значення величини
7	Завершення вимірювання	Вихід	Замовник вимірювання	Сигнал завершення вимірювання

### 3.3. Статистичний аналіз метрик та експертних оцінок

Під час проведення первинного статистичного аналізу виконувалося паралельне дослідження значень метрик та експертних оцінок, яке включало наступні кроки:

- побудова гістограм величин;
- обчислення числових характеристик величин;

- обчислення довірчих інтервалів числових характеристик величин;
- визначення нормальності розподілу величин.

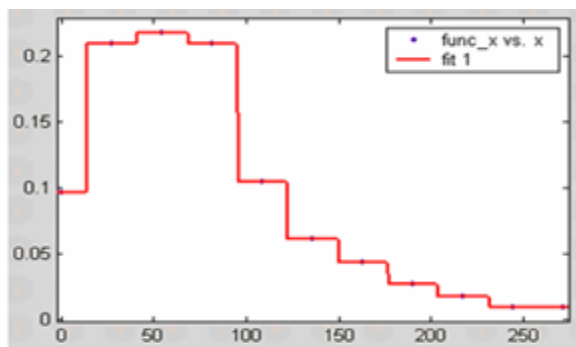
Побудова гістограм проводилась для виявлення графічного виду закону розподілу величин і виключення величин, що мають багатомодальний вид розподілу, так як статистичний аналіз залежностей побудований на дослідженні унімодальних законів розподілу.

Обчислення числових статистичних характеристик величин – проведений для уточнення законів розподілу. До таких характеристик відносяться математичне сподівання, середньоквадратичне відхилення, коефіцієнти асиметрії і ексцесу [13]. На підставі значень математичного сподівання проводилося видалення аномальних явищ (відхилень) за допомогою квантилів розподілу Стюдента шляхом визначення "грубих" значень [4]. Перевірка закону на симетричність проводилась шляхом обчислення всіх статистичних характеристик у зміщеному і незміщеному видах.

Обчислення довірчих інтервалів проводилось для визначення величин, статистичні характеристики яких не потрапляють у довірчий інтервал, і, отже, не мають нормального закону розподілу. Для всіх інших величин проводилося порівняння коефіцієнтів асиметрії і ексцесу зі значеннями, заданими теоретично для нормального розподілу.

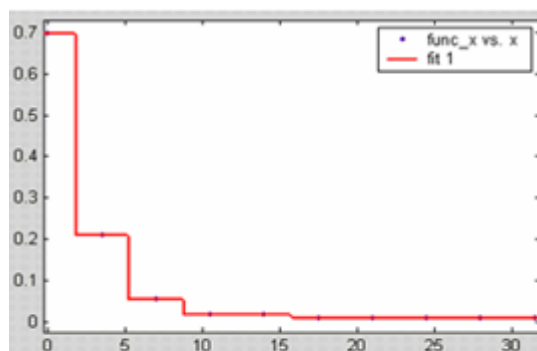
Визначення нормальності розподілу кожної величини – виконувалося на основі гістограм і числових характеристик, одержаних на попередніх кроках.

Побудова та аналіз гістограм метрик. Для кожної метрики була побудована гістограма, що показує закон розподілу її значень. На рис. 3.10 представлені чотири характерні гістограми.



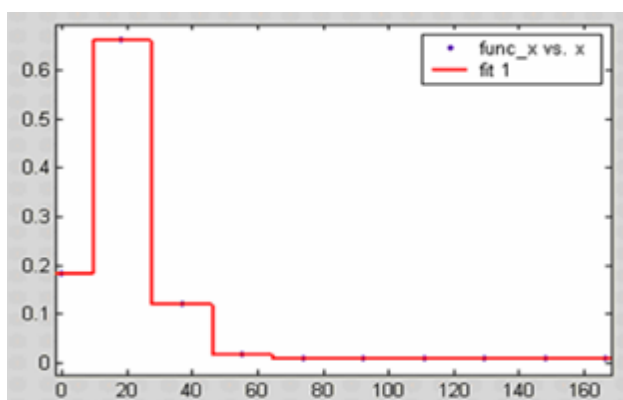
"Максимальна кількість непорожніх стрічок у модулі" (№11)

а)



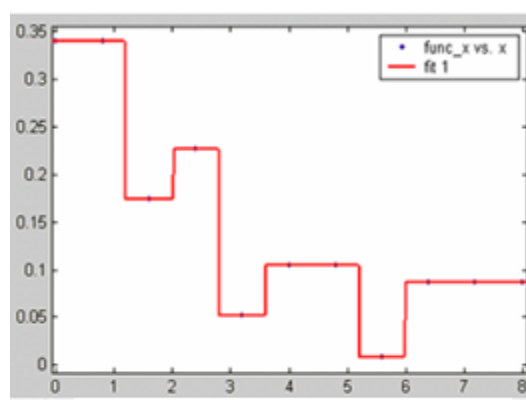
"Середня кількість викликів вводу-виводу, які використовуються у кожному модулі" (№39)

в)



"Максимальна кількість викликів інших функцій, обчислене у модулі" (№24)

б)



"Середня кількість аргументів (параметрів), використаних у кожному модулі" (№36)

г)

Рис.3.10. Характерні гістограми метрик

Аналіз гістограм, одержаних для всіх метрик, показує, що їх вигляд можна звести до наступних чотирьох типових (рис. 3.10 -г): рис 3.10а, 3.10б – гістограми з лівою асиметрією, рис 3.10в - унімодальна гістограма з сильною лівою асиметрією, рис 3.10г - багатомодальна гістограма.

Метрики з багатомодальними гістограмами і метрики, що приймають однакові значення для всієї множини вимірів, внаслідок їх непридатності для подальшого аналізу усуваються з досліджень.

Асиметрія в отриманих унімодальних гістограмах проявляється в сильній і слабкій сторонах. Отже, сильна ліва асиметрія показує, що існує значна кількість малоїмовірних подій в діапазоні для великих значень метрик. А це, в свою чергу,



говорить про те, що обраний набір програм не повною мірою відображає діапазон можливих значень. Наприклад, гістограма метрики №39 «Середня кількість викликів введення-виведення, використаних у кожному модулі» (рис.3.2в), показує, що є велика кількість досліджуваних компонентів, що містять менше 10 викликів, і мала кількість модулів з великою кількістю викликів, тобто більшість значень вибірки зосереджено в малому діапазоні метрики, що ускладнює уточнення залежностей оцінок від цієї метрики.

Для більш точного визначення закону розподілу (в першу чергу – перевірки на "нормальність") застосовувалися числові характеристики величин.

Оцінкою практичних значень виступало інтервальне оцінювання параметрів, для якого  $t_{\alpha/2, v}$  (квантиль розподілу), брався з таблиці для розподілів  $t$  - Стюдента [13]. Довірча ймовірність  $\gamma = 0,95$ , а ймовірність "промаху" у довірчому інтервалі  $\alpha = 1 - \gamma = 1 - 0,95 = 0,05$ . З таблиці  $t_{\alpha/2, v} = 1,96$ . Інтервальне оцінювання зводилося до визначення верхньої і нижньої меж, в які повинна потрапити обчислена характеристика.

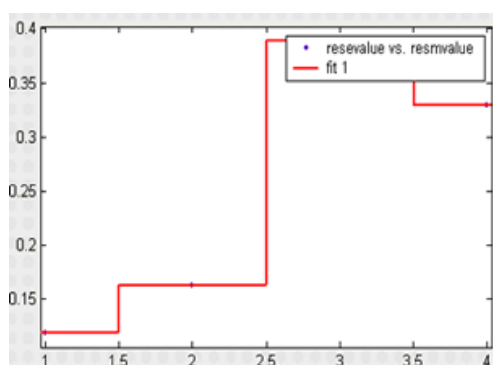
Основними критеріями, за якими визначалася приналежність даного закону розподілу до нормального, були коефіцієнти асиметрії і ексцесу, які при нормальному розподілі значень прямують до 0. Обчислені результати, як і у випадку з гістограмами, говорять про те, що нормальних розподілів немає. Деякі значення коефіцієнтів ексцесу і симетрії є близьким до нормальних, але не настільки, щоб можна було провести вирівнювання.

Можливими причинами ненормального закону розподілу досліджуваних величин могло бути наступне: недостатня кількість вимірних компонентів; підбір програм переважно з малою довжиною коду, що пояснює також велику кількість метрик з лівою асиметрією; використання набору програм, написаних тільки на одній мові (C++), внаслідок чого розподіли метрик можуть відображати особливості, пов'язані з програмуванням на цій мові.

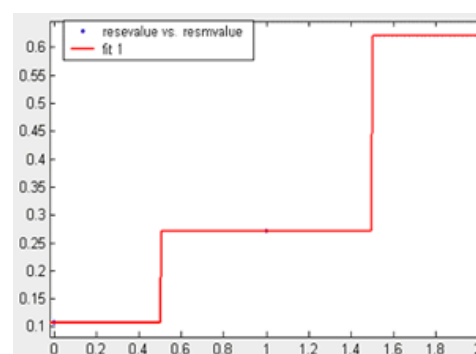
Побудова та аналіз гістограм експертних оцінок. Для кожної експертної оцінки була побудована гістограма, що показує закон розподілу її значень. На

рис.3.5. наведені гістограми для трьох експертних оцінок: "зрозумілість" (рис.3.11); "коректність" (рис.3.11б); "ефективність" (рис.3.11в).

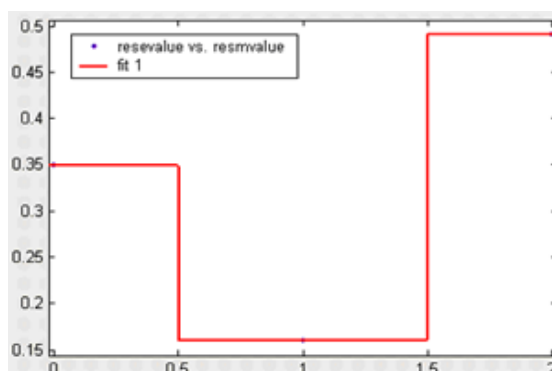
Аналіз гістограм показує, що їх вигляд можна звести до трьох типів: унімодальної гістограми з правою асиметрією (рис. 3.5а); унімодальної гістограми з сильною правою асиметрією (рис 3.11б); багатомодальної гістограми (рис.3.11в). До першого типу відноситься експертна оцінка з позначенням 'RA', до другого типу – експертні оцінки з позначеннями 'PKG', 'PRM', 'CMP', 'MOD', 'CR', 'TA', 'PF', 'CLS', 'STR', 'MOB', до третього типу - 'CF', 'HRS', 'DSN', 'UNSF', 'UR', 'RLB', 'TRK', 'SXIN', 'SMIN'.



а) «Зрозумілість»



б) «Коректність»



в) «Ефективність»

Рис.3.11. Гістограми експертних оцінок

Основним недоліком експертних оцінок з погляду застосування згладжування статистичного ряду є малий діапазон значень, який вони можуть приймати. Отримані види гістограм можуть бути обумовлені невеликою кількістю

експертів, що оцінюють програми і їх недостатньою досвідченістю. Крім того, деякі експертні оцінки можуть приймати тільки два значення. Для цих експертних оцінок подальший аналіз може стимулювати велику похибку. За результатами аналізу таких даних видно, що для них доцільно сформувати іншу шкалу оцінювання і провести оцінювання ще раз. Проте, отримані результати не слід видаляти з подальших досліджень для того, щоб надалі їх можна було порівнювати з новими оцінками.

Обчислення і перевірка числових характеристик експертних оцінок. Значення практичних початкових і центральних моментів метрик представлені в табл. В.2 додатка В, в якій наведено результати обчислень зміщених характеристик. Далі визначаються незміщені характеристики, на підставі яких проводиться порівняння та оцінювання практичних значень.

Як видно з результатів, коефіцієнт ексцесів не потрапляє в допустимі межі. Після його порівняння з нормальним розподілом можна зробити висновок, що і експертні оцінки не мають нормального розподілу, тобто числові характеристики підтверджують побудовані гістограми. Можливо, для поліпшення результатів аналізу необхідно залучити більшу кількість кваліфікованих експертів, а також більше можливих варіацій оцінок.

#### 3.4. Висновки до розділу

Для реалізації запропонованого у дипломній роботі методу розроблені засоби, які забезпечують вирішення двох основних задач: дослідження адекватності моделей властивостям і уточнення залежностей у моделях між властивостями і прямо вимірюваними метриками; автоматизація процесів експертного оцінювання компонентів ПЗ - кандидатів у ПВК.

Розроблено архітектуру, яка складається з таких компонентів: обчислювач значень оцінок; засоби аналізу даних; інтерфейс експерта, ПЗ керування; вимірювач вихідних текстів; база даних; база знань про залежності. Засоби функціонують у двох режимах – накопичення та аналіз даних, оцінювання. Одним

з ключових елементів архітектури, що забезпечує автоматизацію процесу оцінювання, є вимірювач.

У даному розділі одержано наступні результати:

1. Визначено функції та необхідні режими роботи засобів, що забезпечують застосування розробленого методу.

2. Розроблено архітектуру засобів, які забезпечують підтримку запропонованого методу.

3. Розроблено на основі загальних положень теорії вимірювань, принципи роботи та структуру вимірювача текстів програм.

4. Проведено експериментальні дослідження на прикладі програм написаних на мові C++ щодо властивостей повторно використовуваних компонентів на основі аналізу статистики метрик та експертних оцінок.

## РОЗДІЛ 4

### ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Встановлення економічної доцільності дослідження методів експертного оцінювання властивостей наслідуваного програмного забезпечення та розробка програмного засобу підтримки цього процесу, є основною метою виконання розрахунків щодо обґрунтування економічної ефективності.

#### 4.1. Розрахунок норм часу на виконання науково-дослідної роботи

Основні етапи виконання НДР можна визначити наступним чином:

- визначення актуальності теми ДР магістра;
- дослідження методів експертного оцінювання властивостей наслідуваного програмного забезпечення;
- дослідження засобів підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення;
- проектування алгоритму роботи системи підтримки прийняття рішень;
- розробка архітектури системи підтримки прийняття рішень;
- створення інструкції з реалізації програмного комплексу;
- оформлення інструкцій.

Дані витрат часу на виконання окремих стадій (етапів) можна звести у табл. 4.1.

*Таблиця 4.1*

#### Основні етапи виконання НДР

№ та назва етапу	Середній час виконання стадії інженером, год.
1 Визначення актуальності теми ДР магістра	12

Продовження табл. 4.1

№ та назва етапу	Середній час виконання стадії інженером, год.
2 Дослідження методів експертного оцінювання властивостей наслідуваного програмного забезпечення	150
3 Дослідження засобів підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення	40
4 Проектування алгоритму роботи системи підтримки прийняття рішень	30
5 Розробка архітектури системи підтримки прийняття рішень	64
6 Створення інструкції з реалізації програмного комплексу	30
7 Оформлення інструкцій	28
Разом	354

Витрати часу керівника на виконання окремих стадій (етапів) при недостатній кількості інформації доцільно приймати в межах 5% сумарних витрат часу інженерів на виконання цих стадій (етапів).

#### 4.2. Розрахунок витрат на проведення НДР

Розрахунок поточних витрат на проведення НДР проводять в розрізі таких калькуляційних статей:

- основна заробітна плата (з/п);
- додаткова (з/п);
- нарахування на (з/п);
- консультаційні витрати;
- матеріали для виконання робіт по НДР ;
- експериментально-виробничі витрати;
- загальновиробничі витрати;
- адміністративні витрати;

– позавиробничі витрати.

При системному розв'язанні питання про облік праці і заробітної плати велике значення має умовно-постійна (нормативна, довідкова та інша) інформація, яка в даному разі характеризує переважно постійних виконавців (людей і механізми) та постійні процеси (технологічні операції). Тому у першу чергу зміст по обліку праці і заробітної плати неодмінно повинна входити інформація про виконавців (облік складу працівників).

Основна з/п складається із прямої з/п і доплати, яка при укрупнених розрахунках становить 25% - 35% від прямої з/п. При розрахунку з/п кількість робочих днів в місяці слід приймати — 25,4 дні/міс., що відповідає 203,2 год./міс. Прийmemo розмір місячного окладу інженера 1536 грн..

Пряма з/п визначається:

$$ЗП = O_i * T_i / 203,2, \quad (4.1)$$

де  $O_i$  - розмір місячних окладів  $i$ -х категорій працівників;

$T_i$  -трудоємність робіт виконаних працівниками  $i$ -х категорій.

Для інженера:  $ЗП = 1536 * 354 / 203,2 = 2675,91$  грн.;

Величина доплат обраховується за формулою:

$$ЗП_1 = ЗП * K_i \quad (4.2)$$

де  $K_i$  - коефіцієнт доплат (0,25 - 0,35).

Вибираємо коефіцієнт 0,25:

Для інженера:  $ЗП_1 = 2675,91 * 0,25 = 668,98$  грн.;

Основна з/п обчислюється за формулою:

$$ЗП_0 = ЗП + ЗП_1 \quad (4.3)$$

Для інженера:

$$ЗП_О = 2675,91 + 668,98 = 3344,88 \text{ грн.}$$

Величина додаткової з/п обчислюється за формулою:

$$ЗП_Д = ЗП_О * K_Д, \quad (4.4)$$

де  $K_Д$  - коефіцієнт додаткової з/п (0,05 - 0,1).

Нехай коефіцієнт додаткової  $K_Д = 0,1$ .

Для інженера додаткова плата становить:

$$ЗП_Д = 3344,88 * 0,1 = 334,49 \text{ грн.},$$

Витрати, на проведення НДР, крім річного фонду заробітної плати, включають ще й соціальні нарахування. Всього норматив нарахувань на заробітну плату становить 37% .

Загальний норматив нарахувань на заробітну плату розраховується за формулою:

$$З_Н = (ЗП_О + ЗП_Д) * K_Н, \quad (4.5)$$

де  $ЗП_О$  - величина основної заробітної плати;

$ЗП_Д$  - величина додаткової заробітної плати;

$K_Н$  - загальний відсоток нарахувань на заробітну плату.

Для інженера загальний норматив нарахувань становить:

$$З_Н = (3344,88 + 334,49) * 0,37 = 1361,37 \text{ грн.}$$



Таблиця 4.2

## Зведена відомість витрат на заробітну плату, грн.

№ п/п	Категорія працівників	Основна заробітна плата			Додаткова заробітна плата	Нарахування на заробітну плату	Всього витрати на заробітну плату
		Пряма заробітна плата	Доплати	Всього:			
1	Інженер	2675,91	668,98	3344,88	334,49	1361,37	5040,74

Для розрахунку витрат на консультації, врахуємо, що консультації були надані в обсязі 4 год., вартість їх 360 грн.

Витрати на матеріали розраховуються на основі норм їх витрат і відповідних оптових цін:

$$M_3 = \sum_{i=1}^n H_{mi} * C_{oi} \quad (4.6)$$

де  $M_3$  – затрати на матеріали;

$H_{mi}$  – норма затрат і-их матеріалів;

$C_{oi}$  – оптова ціна за одиницю витрат і-их матеріалів;

Таблиця 4.3

## Визначення величини матеріальних витрат

Найменування матеріальних ресурсів	Одиниця виміру	Норма витрат	Ціна за одиницю, грн	Затрати матеріалів, грн	Транспортно-заготівельні витрати, грн.	Загальна сума витрат на матеріали, грн.
<b>1 Основні матеріали</b>						
Середовище MS Visio 2010	шт.	1	6000	6000	600	6600

## Продовження таблиці 4.3

Найменування матеріальних ресурсів	Одиниця виміру	Норма витрат	Ціна за одиницю, грн	Затрати матеріалів, грн	Транспортно-заготівельні витрати, грн.	Загальна сума витрат на матеріали, грн.
2 Додаткові матеріали						
Ручка	шт.	2	10	20	2	22,00
Олівець	шт.	3	2	6	0,6	6,60
Гумка	шт.	1	2	2	0,2	2,20
Компакт диски	шт.	2	4	8	0,8	8,80
Разом						6639,60

Експериментально-виробничі витрати визначаються як витрати на машинний час, який є потрібним для виконання необхідного об'єму робіт виходячи з його вартості за одиницю часу, тобто:

$$Z_{E.B.} = B_p * T, \quad (4.7)$$

де  $Z_{E.B.}$  - затрати експериментально-виробничі;

$B_p$  - витрати на користування ПК та послуги інтернет;

$T$  - час роботи ПК.

Вартість роботи на ПЕОМ і користування мережею Інтернет встановлюємо виходячи з реальних даних (5 грн./год.). Оскільки, інтернет та ПК використовувався на усіх стадіях, то експериментально-виробничі затрати становлять:

$$Z_{E.B.} = 5 * 354 = 1770,00 \text{ грн.}$$

Загальновиробничі витрати при укрупнених розрахунках приймаємо на рівні 70% - 90% від суми основної і додаткової з/п інженерів, яка була нарахована за роботу при проведенні НДР, тобто:

$$Z_{з.в.} = (ЗП_О + ЗП_Д) * K_з \quad (4.8)$$

де  $Z_{з.в.}$  - загально-виробничі затрати;

$ЗП_О$  - основна заробітна плата;

$ЗП_Д$  - додаткова заробітна плата;

$K_з$  - коефіцієнт загальновиробничих затрат.

В даному випадку прийемо коефіцієнт загально-виробничих затрат на рівні 80%, тоді сума затрат становитиме:

$$Z_{з.в.} = (3344,88 + 334,49) * 0,8 = 2943,50 \text{ грн.}$$

Аналогічно визначаються адміністративні витрати, які доцільно приймати на рівні 50% - 60% від суми основної і додаткової з/п інженерів.

$$Z_{з.а.} = (ЗП_О + ЗП_Д) * K_а, \quad (4.9)$$

де  $Z_{з.а.}$  - адміністративні витрати;

$ЗП_О$  - величина основної заробітної плати;

$ЗП_Д$  - величина додаткової заробітної плати;

$K_а$  - коефіцієнт адміністративних витрат.

Нехай коефіцієнт адміністративних витрат становить 50%, то величина адміністративних витрат буде рівна:

$$Z_{з.а.} = (3344,88 + 334,49) * 0,5 = 1839,69 \text{ грн.}$$

Позавиробничі витрати слід приймати на рівні 3% - 7% від виробничої собівартості. Виробнича собівартість включає в себе основну заробітну плату,

додаткову заробітну плату, нарахування на заробітну плату, консультації, матеріали, експериментально-виробничі та загально-виробничі витрати, тобто:

$$B_{П.В.} = (ЗП_{О} + ЗП_{Д} + З_{Н} + M_{З} + З_{Е.В.} + З_{З.В.}) * K_{П.В.} \quad (4.10)$$

де  $B_{П.В.}$  - позавиробничі витрати;

$K_{П.В.}$  - коефіцієнт позавиробничих витрат.

В даному випадку  $K_{П.В.} = 5\%$ , тоді позавиробничі витрати становитимуть:

$$B_{П.В.} = (3344,88 + 334,49 + 1361,37 + 6639,60 + 1770,00 + 2943,50 + 360) * 0,05 = 837,69 \text{ грн.}$$

Розрахунок поточних витрат зводиться в таблицю 4.4:

Таблиця 4.4

#### Калькуляція собівартості проведення НДР

Статті витрати, грн.	Витрати, грн.	В % до загальної суми
Основна заробітна плата	3344,88	17,21%
Додаткова заробітна плата	334,49	1,72%
Нарахування на заробітну плату	1361,37	7,01%
Консультації	360	1,85%
Матеріали	6639,6	34,17%
Експериментально-виробничі витрати	1770,00	9,11%
Загальновиробничі витрати	2943,50	15,15%
Разом виробнича собівартість	16753,83	86,22%
Адміністративні витрати	1839,69	9,47%
Позавиробничі витрати	837,69	4,31%
Повна собівартість	19431,21	100%

Заключною частиною роботи на цій ділянці є показники, що необхідні для встановлення собівартості, проведення комплексного економічного аналізу затрат праці і нарахованої заробітної плати.

4.3. Розрахунок ціни НДР і економічна ефективність від використання методу і засобу підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення

Ціну НДР можна визначити [13]:

$$Ц = (C_{np} / N + C_{kop}) + П, \quad (4.11)$$

де  $C_{np}$  - собівартість НДР, грн.;

$N$  - кількість замовлень, од.;

$C_{kop}$  - собівартість копіювання (ксерокопії, дискети, компакт-диски, поштові витрати, відрядження спеціалістів для запуску та наладки програмного забезпечення тощо), грн.;

$П$  - нормативна величина прибутку (15% - 30% від собівартості  $C_{np}$ ).

Кількість замовлень при проведенні НДР:  $N = 1$ , собівартість копіювання становить:  $C_{kop} = 100$  грн. Оцінка економічної ефективності розробки НДР при створенні програмної системи рівна:

$$Ц = (19431,21 / 1 + 100) + 19431,21 * 0,2 = 23417,45 \text{ грн.}$$

Економічна ефективність від використання НДР зумовлена:

- скороченням трудовитрат при виконанні певних завдань;
- скороченням машинного часу при виконанні певних завдань.

При визначенні економічної ефективності необхідно порівняти використовуваний (базовий) програмний продукт і пропонуваній. З допомогою відповідних розрахунків (в разі значної складності використання експертних

оцінок) визначається скорочення трудовитрат і (або) машинного часу, і як наслідок – економія коштів при використанні нового програмного продукту.

Для визначення ефективності НДР розраховують чисту приведену цінність  $NPV$  і термін окупності  $T_{OK}$  [13]:

$$NPV = \sum ((D_t - B_t) / (1 + i)^t), \quad (4.12)$$

де  $D_t$  - повний дохід за рік  $t$  при використанні засобу підтримки експертного оцінювання;

$B_t$  - повні витрати за рік  $t$  при використанні засобу підтримки експертного оцінювання;

$t$  - відповідний рік проекту;

$i$  - дисконтна ставка (0,3).

Нехай повний дохід за рік при використанні методу і засобу підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення – 12000 грн., а витрати на обслуговування – 5000 грн. тоді чиста приведена цінність:

$$NPV = \sum_{t=1}^2 ((12000 - 5000) / (1 + 0,3)^t) = 9526,63 \text{ грн.}$$

Термін окупності визначається за формулою [13]:

$$T_{OK} = C / \sum (D_t / (1 + i)^t), \quad (4.13)$$

Термін окупності методу і засобу підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення становить:

$$T_{OK} = 23417,45 / (9526,63 / (1 + 0,3)^1) + 23417,45 / (9526,63 / (1 + 0,3)^2) = 1,58 \text{ року.}$$

Таблиця 4.5

## Основні показники ефективності

№ п/п	Назва показника	Один, вимір.	Величина
1	Витрати часу на розробку методу і засобу підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення	год.	354
2	Витрати на розробку методу і засобу підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення	грн.	19431,21
3	Кількість покупців системи	од.	1,00
4	Ціна розробки методу і засобу підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення	грн.	23417,45
5	Чиста приведена цінність методу і засобу підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення	грн.	9526,63
6	Термін окупності витрат по НДР	рік	1,58

Отже, собівартість методу і засобу підтримки експертного оцінювання властивостей наслідуваного програмного забезпечення складає 23417,45 грн. Термін окупності НДР становить 1,58 року, що забезпечує економічну доцільність впровадження результатів науково-дослідної роботи.

## РОЗДІЛ 5

### ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

#### 5.1. Охорона праці

Основним нормативним документом, який регламентує норми і правила експлуатації електронно-обчислювальної техніки є НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин». Правила встановлюють вимоги безпеки до обладнання робочих місць операторів ЕОМ. Вимоги НПАОП 0.00-1.28-10 є обов'язковими для роботодавців, операторів електронно-обчислювальних машин, операторів комп'ютерного набору, операторів комп'ютерної верстки та працівників інших професій, які у своїй роботі застосовують ЕОМ з ВДТ і ПП [14].

Згідно з [32, 33] площа, на якій розташовується одне робоче місце, облаштоване комп'ютером, повинна становити не менше як  $6.0 \text{ м}^2$ , об'ємом не менше як  $20 \text{ м}^3$ .

Робочі місця працівників, які використовують ПК, розташовуються на відстані від стін - на 1м; між собою на відстані 1,7 м, що відповідає вимогам до організації робочих місць, обладнаних комп'ютерами.

З метою запобігання нещасним випадкам та організації охорони праці на виробництві розробляються інструкції з охорони праці і техніки безпеки при використанні комп'ютерної техніки. Дія інструкції поширюється на всі структурні підрозділи даної установи.

До роботи на ПК допускаються особи, які пройшли спеціальне навчання, медичне обстеження, вступний інструктаж з охорони праці, інструктаж на робочому місці та інструктаж з пожежної безпеки.

Згідно із розробленою інструкцією, відповідальний працівник за охорону праці і техніку безпеки при експлуатації засобів автоматизованого експертного оцінювання властивостей повторно використовуваних компонентів, повинен дотримуватись правил внутрішнього трудового розпорядку, не допускати в робочу



зону сторонніх осіб, не виконувати вказівок, які суперечать правилам охорони праці та пам'ятати про особисту відповідальність за виконання правил охорони праці та безпеку товаришів по роботі.

Для усіх працівників поставлено вимоги щодо вміння надавати першу медичну допомогу потерпілим від нещасних випадків, користування первинними засобами пожежогасіння та дотримання правил особистої гігієни.

Основними небезпечними та шкідливими виробничими факторами, які впливають на користувачів ПК є:

- підвищений рівень шуму на робочому місці (від вентиляторів, процесорів та аудіоплат);
- можливе підвищене значення напруги в електричному колі, замикання якого може статися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітного випромінення;
- підвищена напруженість електричного поля;
- прямий та відбитий від екранів блиск; несприятливий розподіл яскравості в полі зору;
- фізичні перевантаження статичної та динамічної дії;
- нервово-психічні перевантаження (розумове перенапруження, перенапруження аналізаторів, монотонність праці, емоційні перевантаження).

При виборі кімнат для розміщення робочих місць ПК враховано ступінь відбиття світла на екранах дисплеїв, яке проходить через вікна і яке може викликати значне осліплення в тих, хто сидить перед ними, особливо влітку та в сонячні дні. Тому, ПК і оргтехніка розміщені біля стін, які не знаходяться біля вікон або навпроти них.

Оскільки, при незадовільному освітленні знижується продуктивність праці користувачів ПК, і можливі негативні впливи на здоров'я такі, як короткозорість, швидка втомленість, тому всі приміщення, які облаштовані робочими місцями з ПК, мають природне і штучне освітлення.

Розміщення робочих місць користувачів ПК повинні відповідати ГОСТ 22269-76 «Рабочее место оператора. Взаимное расположение элементов рабочего места». Не допускається розташування робочих місць з ПК в підвальних приміщеннях.

Робочі місця з ПК при виконанні творчої роботи, яка потребує значної розумової напруги чи великої концентрації уваги, ізольовано одне від одного перегородкою висотою 1,6 м.

Поверхня підлоги у приміщеннях повинна бути оздоблена керамічною плиткою і бути рівною та зручною для очищення та вологого прибирання.

Штучне освітлення у приміщеннях повинно бути виконано у вигляді комбінованої системи освітлення з використанням люмінесцентних джерел світла у світильниках загального освітлення, які розташовувати над робочими поверхнями у рівномірно-прямокутному порядку. Штучне освітлення забезпечує на робочих місцях з ПК освітленість 300 – 500 лк.

Для запобігання засвітленню екранів ПК прямими світловими потоками лінії світильників розташовані з достатнім бічним зміщенням відносно рядів робочих місць, а також паралельно до світлових отворів. При цьому кожне вікно повинно мати світлорозсіюючі штори з коефіцієнтом відбивання 0,7.

У приміщенні також необхідно забезпечити і природне освітлення, при цьому на кожному вікні закріплені жалюзі з вертикальними ламелями, що регулюються для зменшення прямого попадання сонячного світла на екран комп'ютерів.

Згідно [18] світлових відблисків з клавіатури, екрана та від інших частин ВДТ у напрямку очей оператора не повинно бути. Усі робочі місця з ПК повинні бути розташовані таким чином, щоб в поле зору користувача не потрапляли вікна або освітлювальні прилади.

З метою мінімізації відблисків від клавіатури та екрану застосовуються спеціальні захисні козирки і джерела штучного світла розташовано паралельно напрямку погляду на екран ПК з обох сторін.

Для запобігання засліпленню користувачів світильники місцевого освітлення молочного кольору. Захисний кут відбивача світильника становить  $60^{\circ}$ .

Для оздоблення приміщень використовують дифузно-відзеркалюючі матеріали з коефіцієнтами відбиття: стелі - 0,8; стін - 0,5; підлоги - 0,3, що відповідає чинним вимогам [14].

Вміст шкідливих хімічних речовин у приміщеннях з ПК не перевищує концентрацій вказаних у ГОСТ 12.1.005-88 «Общие санитарно-гигиенические требования к воздуху рабочей зоны».

На робочому місці користувачів ПК розміщується монітор, клавіатура, робочий стіл, крісло. При розташуванні елементів робочого місця враховано наступні фактори:

- робочу позу користувача;
- простір для розміщення користувача;
- можливість огляду елементів робочого місця;
- можливість огляду простору за межами робочого місця;
- можливість робити записи, розміщення документації і матеріалів, які використовує користувач.

Взаємне розташування елементів робочого місця не заважає виконанню всіх необхідних рухів та переміщень для експлуатації ПК. Отже, при дослідженні та розробці методу експертного оцінювання властивостей повторно використовуваних компонентів, враховано необхідні вимоги щодо охорони праці при використанні електронно-обчислювальної техніки і забезпечено умови для зручної та ефективної роботи працівників.

## 5.2. Уражаючі фактори вибуху та способи захисту від них

Вибух – це вивільнення значної кількості енергії в обмеженому об'ємі за короткий проміжок часу. Вибух здійснюється частіше за все за рахунок вивільнення хімічної енергії вибухових речовин [ ].

Вибухові речовини (ВР) – це порох, динаміт, тротил, нітрогліцерин та інші хімічні речовини, їх сполуки або суміші, здатні вибухнути без доступу кисню [ ].

За складом ВР поділяються на вибухові хімічні сполуки і вибухові суміші, а за призначенням – на ініціювальні (первинні) та бризантні (вторинні). Ініціювальна ВР (азид свинцю, гримуча ртуть, тетразен тощо) - високочутлива до найпростіших імпульсів маса (удар, тертя, наколювання, електрична іскра тощо), яка застосовується для збудження вибухових перетворень у зарядах вторинних ВР. Бризантні ВР мають велику швидкість детонації, тобто швидкість вибухового перетворення - до 8 км/сек; серед них можна виділити гексоген, октоген, тротил, тетрил тощо [ 1 ].

Під вибуховими пристроями (ВП) розуміють саморобні чи виготовлені промисловим способом вироби одноразового застосування, спеціально підготовлені і за певних обставин спроможні за допомогою використання хімічної, теплової, електричної енергії або фізичного впливу (вибуху, удару) створити вражаючі ефекти - спричинити смерть, тілесні ушкодження, істотну матеріальну шкоду шляхом вивільнення, розсіювання або впливу ударної хвилі, спалаху світла, теплової енергії, осколків, токсичних речовин, радіоактивних матеріалів, біологічних агентів. ВП можуть бути штатні та виготовлені з використанням побутових предметів – портфелів, іграшок, пакетів, електроприладів. Вони можуть спрацьовувати від натискування, падіння, піднімання, впливу радіохвиль певної частоти [ 1 ].

Основними елементами ВП є заряд ВР або вибухонебезпечної суміші, засоби ініціювання (підрильник, детонатор), обладнання для приведення ВП у дію та корпус ВП, хоч останній може бути й безоболонковим.

Підрибок вибухового пристрою може здійснюватись механічним, електричним або вогневим способами. Для першого використовують механічний підрильник – детонатор, ударник, пружину та запобіжну чеку. Для другого способу підрибу застосовують електродетонатори, електробатарей або інше джерело живлення: ініціювання підрибу відбувається шляхом замкнення електричної мережі і підрибу детонатора (дистанційно або за допомогою проводів). Третій спосіб підрибу передбачає для ініціювання детонатора використання вогнепровідного шнура.

Вибух – це процес надзвичайно швидкого горіння, що супроводжується швидким зростанням тиску і має велику руйнівну силу.

Якщо у повітряному середовищі виникає така концентрація пилу, парів або газів, яка досягає межі вибуху, то при наявності відкритого джерела вогню вибух станеться (табл.5.1).

Пил деяких речовин має такі значення нижньої межі вибуху (г/м<sup>3</sup>): цукор - 8,9; торф - 10,1; сіно - 20,2; тирса - 65,0.

Найбільш небезпечним є високодисперсний пил, бо він має велику сумарну поверхню, що створює підвищену хімічну активність.

Вибухи та пожежі можуть виникати за таких обставин:

- у початковий період експлуатації виробництва — період притирання елементів технологічного обладнання (недоліки допущенні у процесі проектування, неякісне виконання монтажних робіт та ін.);
- в основний період експлуатації виробництва (через несправність контрольно-вимірювальних приладів та елементів обладнання, порушення вимог безпеки, недостатній нагляд і контроль, незадовільні планово-профілактичні ремонти та ін.);
- у період так званого "старіння" елементів технологічного обладнання (через корозію матеріалів, зношеність деталей, відсутність капітальних і поточних ремонтів та ін.).

*Таблиця 5.1.*

**Концентраційні межі вибуху (займання) деяких горючих газів і парів, легкозаймистих і горючих рідин**

Речовина	Нижня межа		Верхня межа	
	% за об'ємом	г/м <sup>3</sup> при 20 <sup>0</sup> С	% за об'ємом	г/м <sup>3</sup> при 20 <sup>0</sup> С
Метилловий спирт	6,7	46,5	38,5	512,0
Етиловий спирт	3,61	50,0	19,0	363,0
Бутан	1,8	37,4	8,5	204,8
Метан	5,28	16,66	15,4	102,6
Пропан	2,31	36,6	9,5	173,8
Ацетилен	2,50	16,5	82,0	885,6

Продовження табл. 5.1

Речовина	Нижня межа		Верхня межа	
	% за об'ємом	г/м <sup>3</sup> при 20 <sup>0</sup> С	% за об'ємом	г/м <sup>3</sup> при 20 <sup>0</sup> С
Пропілен	2,30	34,8	11,1	169,0
Етилен	3,11	35,0	35,0	406,0
Бензол	1,43	42,0	9,5	308,0
Ксилол	1,0	44,0	7,6	34,0
Толуол	1,25	38,2	7,0	268,0
Ацетон	2,91	38,6	13,0	314,0
Аміак	17,0	112,0	27,0	189,0
Сірководень	4,0	61,0	44,5	628,0
Бензин паливний	2,4	137,0	4,9	281,0
Бензин розчинник	1,9	-	5,1	-
Водень	4,09	3,4	80,0	66,4

Причинами виникнення пожеж можуть бути конструктивні недоліки та порушення експлуатації пічного опалення, електрообладнання та електроустаткування, інженерних комунікацій, дефекти обладнання; організаційні порушення режимів технологічних процесів, похибки при виконанні технологічних процесів, необережні дії персоналу та низький рівень їх кваліфікації, порушення правил при поводженні з вогнем і т. ін.

Всі різноманітні причини пожеж можна об'єднати у дві великі групи, що пов'язані:

- I група - з недопустимою, з точки зору пожежної безпеки, появою горючого середовища в умовах, де є джерело вогню (розрив трубопроводів в котельнях, що працюють на рідкому паливі, підтікання паливних ліній, аварійний стан обладнання, викиди бітуму при варінні і т.д.);

- II група - з недопустимою появою джерел вогню при наявності горючої суміші і окисника.

Сюди відносяться пожежі обумовлені такими чинниками як:

- порушення режимів роботи технологічного та інженерно-технічного обладнання;

- недоліки монтажу та порушення експлуатації електрообладнання, електропроводки, електроапаратури (коротке замикання, іскріння, перевантаження проводів і ін.);
- недоліки при облаштуванні та експлуатації опалювальних систем та установок;
- недопустиме підвищенням температури при адіабатичному стисненні (робота компресорного устаткування);
- перегрів речовин, що обробляються, коли їх температура досягає температури самозаймання;
- порушення режиму зберігання та обробки самозаймистих речовин і матеріалів;
- необережне поводження з вогнем, незнання правил та норм пожежної безпеки, недбале ставлення до своїх обов'язків, навмисний підпал, накопичення електростатичних розрядів і т.д.

Безпосередньо причиною пожежі може стати у непередбачений час, у непередбаченому місці, з точки зору пожежної безпеки, поява того чи іншого компонента, який бере участь в процесі горіння, а відтак, з метою профілактики, недопустимими є такі умови, що можуть призводити до неконтрольованих процесів горіння.

Основними вражаючими факторами вибухів є:

- повітряна ударна хвиля (ПУХ), що виникає при ядерних вибухах, вибухах детонуючих та ініціюючих речовин, при вибухових перетвореннях хмар паливно-повітряних сумішей, вибухів резервуарів з перегрітої рідиною і резервуарів під тиском;
- осколкові поля, створювані уламками різного роду об'єктів.

Основними параметрами вражаючих факторів є:

- повітряна ударна хвиля – надлишковий тиск в її фронті;
- осколкове поле – кількість осколків, їх кінетична енергія і радіус розльоту.

У результаті дії вражаючих факторів вибуху відбувається руйнування або пошкодження будівель, споруд, обладнання, елементів комунікації, і загибель людей і тварин.

Вторинними наслідками вибухів є ураження об'єктів, які знаходяться всередині приміщень, уламками завалених конструкцій будівлі. У результаті вибухів можуть виникнути пожежі, витік небезпечних речовин з пошкодженого обладнання.

При пожежах і вибухах люди отримують термічні і механічні травми. Характерні опіки верхніх дихальних шляхів, тіла, черепно-мозкові травми, множинні переломи і удари, комбіновані ушкодження.

Для запобігання пожеж і вибухів необхідно виключити можливість утворення горючого і вибухонебезпечного середовища, а також запобігти появі в цих середовищах джерел загоряння.

Висновки.

Уражаюча дія вибуху, незалежно від його природи, може призводити до значних матеріальних і людських втрат. Особливо актуальним є питання збереження життя людей і забезпечення функціонування підприємств стратегічного значення в Україні в умовах війни на сході. Тому будівлі підприємств і житлового сектору повинні проектуватись і будуватись з урахуванням вимог нормативних документів (ГОСТи «Пожежна безпека» та «Вибухобезпека. Загальні вимоги»).



## РОЗДІЛ 6 ЕКОЛОГІЯ

6.1. Вимоги до приміщень для експлуатації моніторів і ПЕОМ. Шляхи дотримання цих вимог

Широке промислове та побутове використання ПК актуалізувало питання охорони праці їхніх користувачів. Найбільш повним нормативним документом щодо забезпечення охорони праці користувачів ПК є "Державні санітарні норми і правила роботи з візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин" ДСанПіН 3.3.2.007-98.

Дотримання вимог цих правил може значно знизити наслідки несприятливої дії на працівників шкідливих та небезпечних факторів, які супроводжують роботу з відеодисплейними терміналами, зокрема можливість зорових, нервово-емоційних переживань, серцево-судинних захворювань [18]. Виходячи з цього, роботодавець повинен забезпечити гігієнічні й ергономічні вимоги щодо організації робочих приміщень для експлуатації ВДТ, робочого середовища, робочих місць з ВДТ, режиму праці і відпочинку при роботі з ВДТ тощо, які викладені у Правилах.

Природне освітлення в приміщеннях з ВДТ має здійснюватися через вікна, орієнтовані переважно на північ або північний схід і забезпечувати коефіцієнт природної освітленості не нижче ніж 1,5 %. Для захисту від прямих сонячних променів, які створюють прямі та відбиті відблиски з поверхні екранів ПК і клавіатури, повинні бути передбачені сонцезахисні пристрої, при цьому вікна обладнуються жалюзіями або шторами [18].

Основні вимоги до виробничого приміщення для експлуатації ВДТ:

- заборона розташування у підвалах та цокольних поверхах;
- площа на одне робоче місце повинна становити не менше  $6,0 \text{ м}^2$ , а об'єм не менше  $20,0 \text{ м}^3$ ;
- наявність природного та штучного освітлення відповідно до діючих стандартів;

- наявність шафи для зберігання документів, магнітних дисків, полиці, стелажі, тумби тощо, з урахуванням вимог до площі приміщення;
- щоденне проведення вологого прибирання;
- поруч з приміщенням для роботи з ВДТ мають бути обладнані:
- побутова кімната для відпочинку під час роботи.

Штучне освітлення в приміщеннях з робочим місцем, обладнаним ВДТ, має здійснюватись системою загального рівномірного освітлення. Як джерело штучного освітлення мають застосовуватись люмінесцентні лампи типу ЛБ.

Вимоги до освітлення приміщень та робочих місць під час роботи з ВДТ [18]:

- освітленість на робочому місці повинна відповідати характеру зорової роботи, який визначається трьома параметрами: об'єктом розрізнення - найменшим розміром об'єкта, що розглядається на моніторі ПК; фоном, який характеризується коефіцієнтом відбиття; контрастом об'єкта і фону;
- необхідно забезпечити достатньо рівномірне розподілення яскравості на робочій поверхні монітора, а також в межах навколишнього простору;
- на робочій поверхні повинні бути відсутні різкі тіні;
- в полі зору не повинно бути відблисків (підвищеної яскравості поверхонь, які світяться та викликають осліплення);
- величина освітленості повинна бути постійною під час роботи;
- слід обирати оптимальну спрямованість світлового потоку і необхідний склад світла.

Отже, дотримання вимог стандартів щодо приміщень, де експлуатуються ПЕОМ, дає змогу забезпечити екологічну безпеку навколишнього середовища і зменшить негативний вплив на людину як біологічний об'єкт.

## 6.2. Методи узагальнення екологічної інформації

Статистичні таблиці є формою раціонального та систематизованого викладення цифрової інформації. Основною перевагою цифрової інформації,

зведеної в таблиці, є компактність, наочність, виразність. Інформація стає легкодоступною і рельєфною, компактною і раціональною.

За допомогою таблиць можна досягти систематизації цифрової інформації, полегшити і прискорити ефект її сприйняття, інтенсифікувати пізнавальний процес явища, зекономити місце при викладенні інформації.

Статистичними таблицями вважають тільки ті, що містять наслідки статистичного аналізу еколого-економічних явищ і процесів [15].

Таблиця за своїм логічним змістом розглядається як «статистичне речення», що має свій підмет і присудок [15]. Підмет таблиці характеризує об'єкт дослідження, а присудок - це система показників, що відображає підмет як об'єкт.

Вимоги до оформлення таблиць:

- кожна таблиця повинна мати назву, що відображає її зміст:
  - заголовок повинен розміщуватись над таблицею;
  - текст заголовка має бути коротким, лаконічним;
  - відривати (переносити) заголовок від таблиці не можна;
- назви рядків і граф повинні даватися повністю без скорочень;
- кожна таблиця повинна мати одиниці виміру;
- числа в таблиці не повинні бути громіздкими (не > 3-4 знаків);
- таблиці іноді потребують нумерації, яка робиться по-різному:
  - перед назвою (ставлять цифри номера);
  - над назвою в правій частині таблиці (пишуть слово “таблиця”, знак номера і сама цифра номера).

Статистична таблиця має ряд горизонтальних рядків і вертикальних граф. Перетин рядків і граф утворює клітини таблиці. Ліві бічні і верхні клітини призначені для словесних заголовків, а решта - для числових.

Кожна таблиця має три заголовки:

- загальний – відображає зміст таблиці (його місце над таблицею);
- ліві (бічні) – найменування рядків, розкривають зміст підмета;

- верхні – найменування граф, розкривають зміст присудка.

Оптимальний за розміром об'єм таблиці складає 25-30 графо-клітин, який є добутком рядків і граф.

Статистичний графік є рисунком, який описує статистичні сукупності умовною мовою геометричних знаків тієї чи іншої форми: крапок, ліній, площин, фігур та різних їх комбінацій.

«Статистичні графіки – це спосіб умовного зображення цифрової інформації у вигляді крапок, ліній, стовпчиків, кругів або фігур [15].

Графічний метод використовують для популяризації цифрової інформації, забезпечення доступності її сприйняття та узагальнення цифрових даних. У більшості випадків статистичних графіків використовують не об'ємне зображення, складне за побудовою, а площинне. Графіки за призначенням поділяють на:

- аналітичні, які дають можливість порівняти графічні образи;
- ілюстративні – допомагають при порівнянні геометричних фігур та показують зміну розмірів явищ;
- інформаційні, які містять інформацію лише про об'єкт вивчення.

Вимоги до оформлення - кожен графік повинен:

- точно відображувати початкові дані;
- найкращим чином відповідати змісту і логічній природі явищ, що зображуються;
- бути наочним, зрозумілим, легко читатися, привертати і утримувати увагу;
- бути художньо оформленим, графічні образи розмальовані різними кольорами.

Основні правила побудови графіків:

- кожен графік повинен мати назву, яка подається:
  - над графіком в демонстраційних малюнках;
  - під графіком, якщо він вміщується в тексті;
- графік повинен мати всі необхідні пояснювальні надписи:
  - наявність масштабу;

- пояснення значення забарвлень, штриховок, назви показників;
- пропорція шкал графіка має бути із співвідношенням  $y / x$ , як 1:1,5;
- на горизонтальній шкалі потрібно розміщувати незалежні змінні - роки, об'єкти, на вертикальній - залежні показники;
- масштаб потрібно вибирати таким, щоб:
  - різниця величин ясно відображалася;
  - показники помістилися на полі графіка;
  - графік легко читався, тобто був читабельним.

Графічні образи на графіках поділяють на крапкові, лінійні; площинні (квадратні, кругові), просторові (об'ємні), зображувальні (фігурні).

Застосування табличних і графічних методів узагальнення екологічної інформації дають змогу наочно представити результати екологічних досліджень, порівняти цільові показники в часі, а також задовольнити вимоги статистичного опрацювання вхідних (досліджуваних) об'єктів чи явищ.

## ВИСНОВКИ

Основні наукові та практичні результати полягають в наступному.

1. Проведено аналітичний огляд літератури; наведено існуючі види повторного використання (систематичне, несистематичне); розглянуто та проаналізовано процеси їх реалізації та шляхи застосування при розробці ПЗ;

2. Проаналізовано процеси створення ПВК при несистематичному повторному використанні наслідуваного ПЗ; показана роль процесу відбору компонентів ПЗ як одного з ключових у забезпеченні ідентифікації компонентів ПЗ - кандидатів у ПВК;

3. Виконано аналіз існуючих моделей, які можуть бути використані для реалізації відбору компонентів ПЗ – кандидатів у ПВК; розглянуті моделі вартості (рішення про повторне використання приймається на основі оцінки вартості виконуваних процесів, пов'язаних з перетворенням компонентів ПЗ в ПВК) і моделі властивостей (рішення про повторне використання приймається на основі оцінок властивостей, що відображають повторну використовуваність компонентів);

4. На основі аналізу моделей обґрунтовано доцільність розробки та дослідження методу експертного оцінювання, заснованого на комбінації моделей властивостей і вартості.

5. Розроблено метод експертного оцінювання властивостей компонентів ПЗ - кандидатів у ПВК, що дає змогу автоматизувати процес одержання значень властивостей шляхом їх обчислення за результатами вимірювань з використанням моделей властивостей.

6. Визначено множину властивостей компонентів ПЗ, важливих з погляду повторного використання, шляхом формування вихідної множини властивостей за допомогою 3С-моделі компонентів ПЗ і вибору тих з них, які впливають на витрати, пов'язані з виконанням процесів повторного використання;

7. Визначено спосіб побудови моделей властивостей на основі системного підходу шляхом застосування системного аналізу та методу GQM; побудовано

моделі властивостей «зрозумілість», «примітивність», «модульність», «структурованість».

8. Розроблено архітектуру засобів, які забезпечують підтримку запропонованого методу.

9. Проведено експериментальні дослідження на прикладі програм написаних на мові C++ щодо властивостей повторно використовуваних компонентів на основі аналізу статистики метрик та експертних оцінок.

10. Проведено обґрунтування економічної доцільності проведення НДР.

11. Проведено аналіз вимог та норм з охорони праці при використанні засобу автоматизації експертного оцінювання властивостей повторно використовуваних компонентів і досліджено уражаючі фактори вибуху та захисту від них.

12. Проведено дослідження щодо вимог до приміщень з експлуатації моніторів та ПЕОМ та методів узагальнення екологічної інформації.

## Додаток А

## Тексти наукових публікацій дипломної роботи магістра

XXI Всеукраїнська студентська науково - технічна конференція "ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ТЕМАТИКИ"

Секція: **Інформаційні технології**

УДК 004.4

Смучок І. – ст. гр. СІм-51

*Тернопільський національний технічний університет імені Івана Пулюя*

**ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ ТА РОЗВИТКУ ПІДХОДІВ  
ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ  
ПОВТОРНО ВИКОРИСТОВУВАНИХ КОМПОНЕНТІВ**

Науковий керівник: к.т.н., доцент Яцишин В.В.

Smuchok I.

*Ternopil Ivan Pul'uj National Technical University*

**PROSPECTS OF DEVELOPMENT AND APPROACHES TO  
SOFTWARE DEVELOPMENT BASED ON REUSABLE COMPONENTS**

Supervisor: PhD, Ass. Prof. Yatsyshyn V.V.

Ключові слова: програмне забезпечення, повторно використовувані компоненти  
Keywords: software, reusable components

Створення сучасних програмних систем, які були б конкурентоздатними на ринку і задовольняли потреби замовників є доволі складною задачею і вимагає від розробників застосування знань з багатьох розділів науки і практики, зокрема, технологій програмування, експертних технологій, математичної статистики та ряду інших. Для того, щоб програмний продукт був конкурентоздатним необхідно, щоб у ньому були реалізовані, принаймні, усі переваги прототипів, які існують на ринку, і затрати фінансів та часу на його розробку були мінімальними. Одним із шляхів зменшення витрат на розробку програмних продуктів є використання підходу наслідуваного програмного забезпечення. Даний підхід передбачає використання компонентів програмного забезпечення, які були створені раніше і показали свою працездатність протягом довготривалого періоду часу. Однак, при цьому розробники стикаються із проблемами експертного оцінювання властивостей компонентів, які плануються повторно використати для нових проєктів.

У процесі застосування повторного використання ПЗ необхідно створювати повторно використовувані компоненти. Під час несистематичного повторного використання повторно використовувані компоненти створюються на основі наслідуваного ПЗ, а доцільність та ефективність їх створення визначається насамперед витратами, які не повинні перевищувати витрати на створення такого компонента „з нуля”. При цьому, найбільш витратною, що потребує великих обсягів робочого часу висококваліфікованих інженерів, є завдання оцінювання властивостей компонента ПЗ – кандидата в повторно використовувані компоненти, яке сьогодні розв’язується інтуїтивно. Для оцінювання властивостей повторно використовуваних компонентів запропоновано скористатись рекомендаціями міжнародних стандартів серії ISO 25000. На модульному рівні, для визначення властивостей компонентів дієвим є використання стандарту ISO 25010 (ISO 9126). Експертне оцінювання властивостей компонентів запропоновано проводити на основі методу SQFD.



УДК 004.4

В.В. Яцишин канд. техн. наук, доц., І.М.Смучок

Тернопільський національний технічний університет імені Івана Пулюя, Україна

#### ПІДХОДИ ЩОДО ЗАБЕЗПЕЧЕННЯ ЕФЕКТИВНОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

V.V. Yatsyshyn PhD, Assoc. Prof., I.M.Smuchok

#### APPROACHES FOR PROVIDING EFFECTIVENESS OF SOFTWARE DEVELOPMENT

Для підвищення ефективності розробки програмного забезпечення існують різні підходи:

- удосконалення організаційно-технічного забезпечення розробки ПЗ;
- підвищення ефективності фаз життєвого циклу;
- усунення стадій життєвого циклу в розробці ПЗ;
- мінімізація повторного виконання процесів ПЗ;
- повторне використання ПЗ.

Удосконалення організаційно-технічного забезпечення розробки ПЗ передбачає підбір фахівців на виконання визначених в проекті задач, організацію оптимальних для розробників робочих місць та інтеграція процесу управління в процес розробки програмного забезпечення.

Підвищення ефективності фаз життєвого циклу може реалізовуватись шляхом застосування засобів автоматизації і CASE-засобів, які здатні скоротити часові затрати на виконання процесів на найбільш трудомістких етапах.

Усунення фаз життєвого циклу передбачає злиття окремих процесів на різних етапах життєвого циклу або в ідеалі злиття цілих фаз. Однак на практиці реалізувати це практично неможливо чи надто затратно. Можливе злиття фаз проектування та кодування шляхом застосування засобів автоматизації – генерація коду модулів з визначених архітектурних шаблонів.

Мінімізація повторного виконання процесів ПЗ націлена на створення засобів, які дають можливість автоматизувати розробку специфікацій на різних стадіях життєвого циклу. Ці засоби носять інформаційний та моніторинговий характер, оскільки повторне виконання процесів зумовлюється змінами анесеними у результати виконання окремих процесів.

Суть повторного використання ПЗ полягає у зменшенні об'єму новостворюваного ПЗ за рахунок використання компонентів існуючого ПЗ. Крім підвищення продуктивності розробки ПЗ, цей шлях, як показує практика веде до підвищення надійності та якості як окремих програм, так і ПЗ в цілому. Застосування такого підходу дає змогу знизити витрати на кодування, тестування, документування та супровід ПЗ. Застосування повторного використання на ранніх фазах ЖЦ ПЗ призводить до зменшення витрат на розробку специфікацій вимог і на проектування ПЗ.

Тому оптимальним варіантом, щодо підвищення ефективності розробки ПЗ є застосування композиції наведених вищє підходів, однак найбільш раціональним є застосування підходу повторно використовуваних компонентів.

## Додаток Б

Шкали експертних оцінок та перелік метрик вихідного коду повторного використання компонентів наслідуваного програмного забезпечення

Таблиця Б.1

## Експертні оцінки

№ з/п	Код	Назва	Шкала
1.	RA	Зрозумілість	незадовільна, задовільна, хороша, відмінна
2.	CF	Загальність	для проекту, для класу проектів, для класу ПЗ, загального призначення
3.	PKG	Компактність	незадовільна, задовільна, хороша
4.	PRM	Примітивність	непримітивний, примітивний
5.	CMP	Завершеність	незавершений, завершений
6.	MOD	Модульність	незадовільна, задовільна, хороша
7.	CR	Коректність	незадовільна, задовільна, хороша
8.	TA	Тестопридатність	незадовільна, задовільна, хороша
9.	PF	Ефективність	низька, середня, висока
10.	CLS	Здатність до класифікації	ускладнена, хороша
11.	STR	Структурованість	незадовільна, задовільна, хороша
12.	MOB	Переносимість	незадовільна, задовільна, хороша
13.	UNSF	Корисність (частота повторного використання в новому ПЗ)	рідко, інколи, часто
14.	UR	Достатність	недостатня, достатня, надлишкова
15.	RLB	Надійність	ненадійний, надійний
16.	TRK	Трасування	низьке, середнє, високе
17.	SXIN	Синтаксична незалежність	низька, середня, висока
18.	SMIN	Семантична незалежність	низька, середня, висока

## Перелік метрик

№ з/п	Категорія	Назва	№ метри-ки	Код	Опис
1.	Метрики управління потоком (Control Flow Metric)	Percent Modules with Exceptions	5	PME	Відсоток модулів з виключеннями
2.		Average Exceptions per Modules	6	AVGEM	Середній відсоток виключень у модулі
3.		Total Modules	8	TM	Загальна кількість функцій (модулів) компонента
4.		Total LOC	9	TLOC	Загальна кількість непорожніх стрічок, включаючи стрічки коментарів та заголовки.
5.		Max LOC	10	MLOC	Максимальна кількість непорожніх стрічок в будь-якому модулі
6.		Ave LOC	11	AVLOC	Середня кількість непорожніх стрічок в модулі
7.		Total NCSS	12	TNCSS	Загальна кількість вихідних конструкцій. Не включає порожні стрічки, стрічки з коментарями або заголовки у всіх модулях
8.		Max NCSS	13	MNCSS	Максимальна кількість вихідних конструкцій у будь-якому модулі
9.		Ave NCSS	14	AVNCSS	Середня кількість вихідних конструкцій в модулі
10.		Total V(G)	15	TVG	Загальна цикломатична складність у всіх модулях V(G). обчислюється шляхом підрахунку кількості ключових слів розгалуження у кожному модулі
11.		Max V(G)	16	MVG	Максимальна цикломатична складність у модулі
12.		Ave V(G)	17	AVVG	Середня цикломатична складність модулів у компоненті
13.		Total Hal's V	18	THV	Загальний об'єм за Хальстедом, обчислений для модулів Volume = (загальна кількість операторів + загальна кількість операндів) / log <sub>2</sub> (кількість унікальних операторів + кількість унікальних операндів)
14.		Max Hal's V	19	MHV	Максимальний об'єм V для модуля за Хальстедом
15.		Ave Hal's V	20	AVHV	Середній об'єм V модулів в компоненті системи за Хальстедом
16.		Total Hal's E	21	THE	Загальні зусилля за Хальстедом для компонента.
17.		Max Hal's E	22	MHE	Максимальне значення зусиль за Хальстедом, обчислене для модуля
18.		Ave Hal's E	23	AVHE	Середнє значення зусиль за Хальстедом для модулів у компоненті
19.		Max Fan-out	24	MFO	Максимальна кількість викликів інших функцій, обчислене у модулі
20.		Max Fan-in	25	MFI	Максимальна кількість викликів даного модуля, обчислене для компонента
21.		Ave Fan-in	26	AVFI	Загальна кількість викликів інших функцій, поділена на кількість модулів у компоненті
22.		Max Nesting	27	MN	Найвищий рівень структурної вкладеності, обчислений у компоненті через підрахунок (``{``)

## Продовження табл. Б.2

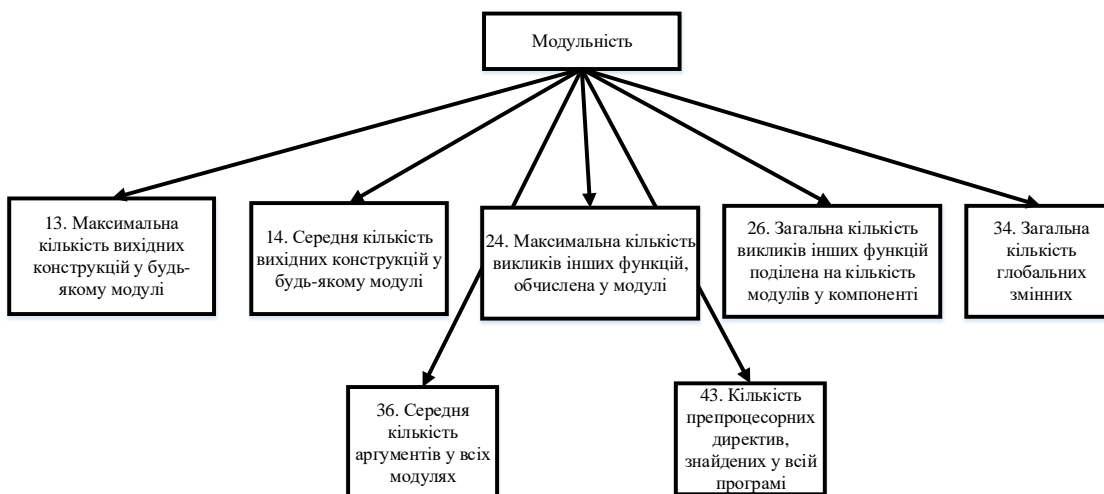
№ з/п	Категорія	Назва	№ метри-ки	Код	Опис
23.	Метрики управління потоком (Control Flow Metric)	Ave Nesting	28	AVN	Середній максимум структурної вкладеності модулів у компоненті
24.		Max Recursive Calls	29	MRC	Максимальна кількість рекурсивних викликів у модулі
25.		Number of Recursive Modules	30	NRM	Загальна кількість модулів компонента, що містять рекурсивні виклики
26.	Метрики потоку даних (Data Flow Metrics)	Total Vars	32	TV	Загальна кількість змінних у всіх модулях
27.		Total Local	33	TL	Загальна кількість локальних змінних у всіх модулях
28.		Total Global	34	TG	Загальна кількість глобальних змінних. Не включає препроцесорні директиви
29.		Total Unused	35	TU	Загальна кількість змінних, оголошених, але не використовуваних у всіх модулях
30.		Ave. Number of Arguments	36	AVNA	Середня кількість аргументів (або параметрів), що використовуються у всіх модулях
31.		Tot Init Vars	37	TIV	Загальний відсоток всіх змінних, ініціалізованих перед використанням
32.		Total Number of External Files	38	TNEF	Загальна кількість файлових змінних у всіх модулях
33.		Ave. Number of I/O Calls per Module	39	AVNIOC	Середня кількість викликів вводу-виводу, використаних у кожному модулі
34.		Ave. Variable Span per Module	40	AVVSM	Середня кількість стрічок між першим та останнім використанням кожної змінної у модулі
35.		Max. Variable Span per Module	41	MVSM	Максимальна кількість стрічок між першим та останнім використанням кожної змінної у модулі
36.		Total Percent Well-Declared Variables	42	TPWDV	Відсоток усіх змінних у програмі, оголошених безпосередньо перед використанням.
37.	Number of Preprocessor Directives	43	NPD	Кількість препроцесорних директив у компоненті	
38.	Типографічні метрики (Typographic Metrics)	Modules with Headers	45	MH	Відсоток модулів, що мають коментовані заголовки (коментарі на початку модуля)
39.		Modules with Comments	46	MC	Відсоток модулів, що мають коментарі між заголовком і кінцем
40.		Percent Modules with Horizontal Spacing	47	PMHS	Відсоток модулів, які мають вбудовані горизонтальні відступи (два і більше пробілів або табуляції)
41.		Percent Modules with Vertical Spacing	48	PMVS	Відсоток модулів, що мають вертикальні відступи (порожні стрічки)
42.		Percent Modules with Indentation	49	PMI	Відсоток модулів із структурованим розміщенням тексту
43.		Ave. Percent Uncrowded Statements within a Module	50	AVPUS	Сума неущільнених конструкцій (не більше однієї конструкції на стрічку) для кожного модуля, поділена на кількість модулів
44.		Ave. Percent Comments within a Module	51	APCM	Середній відсоток коментованих стрічок для всіх модулів
45.		Ave. Percent Meaningful Vars within a Module	52	APMV	Середній відсоток змінних, які мають змістовну назву (назва довжиною більшою, ніж 4 символів)

## Додаток В

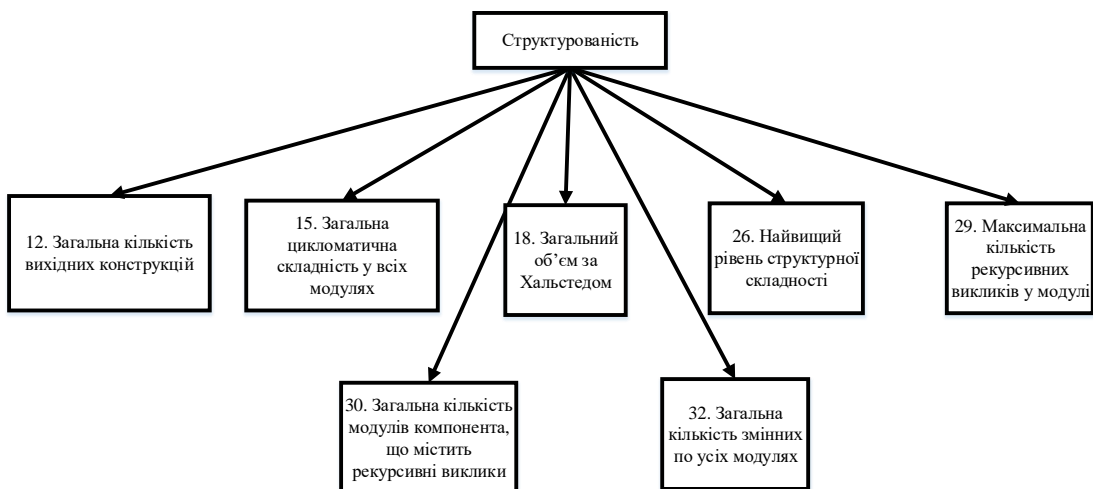
Моделі властивостей повторно використовуваних компонентів наслідуваного програмного забезпечення



Модель властивості «Примітивність»



Модель властивості «Модульність»



Модель властивості «Структурованість»

## Додаток Д

## Інтерфейс програмного засобу для проведення експертного оцінювання властивостей повторно використовуваних компонентів

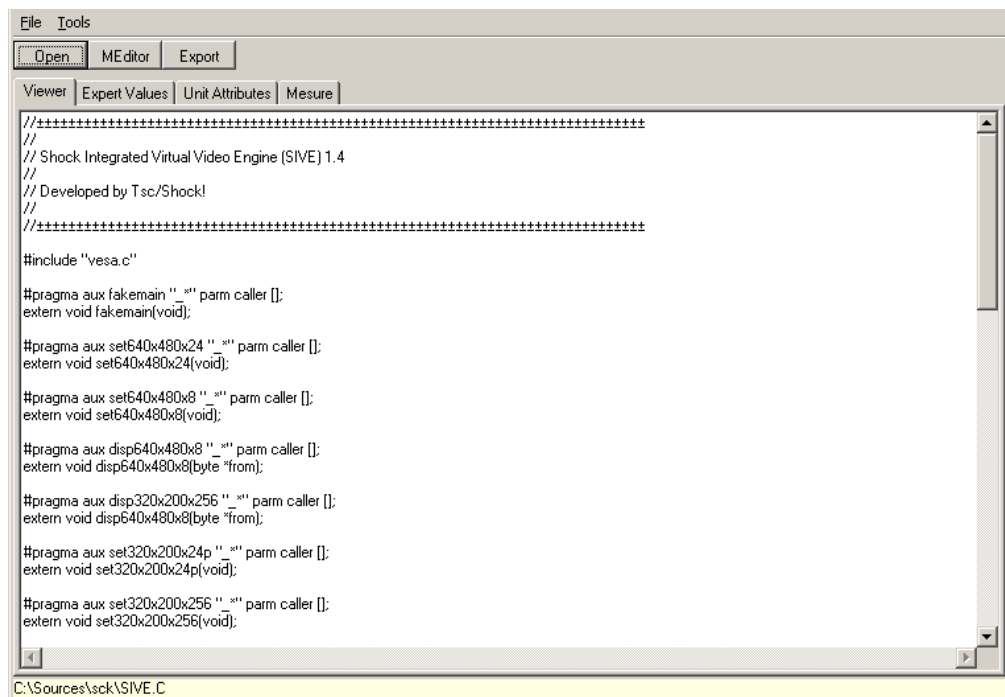


Рис. Д.1. «Браузер» для перегляду текстів програмних компонентів

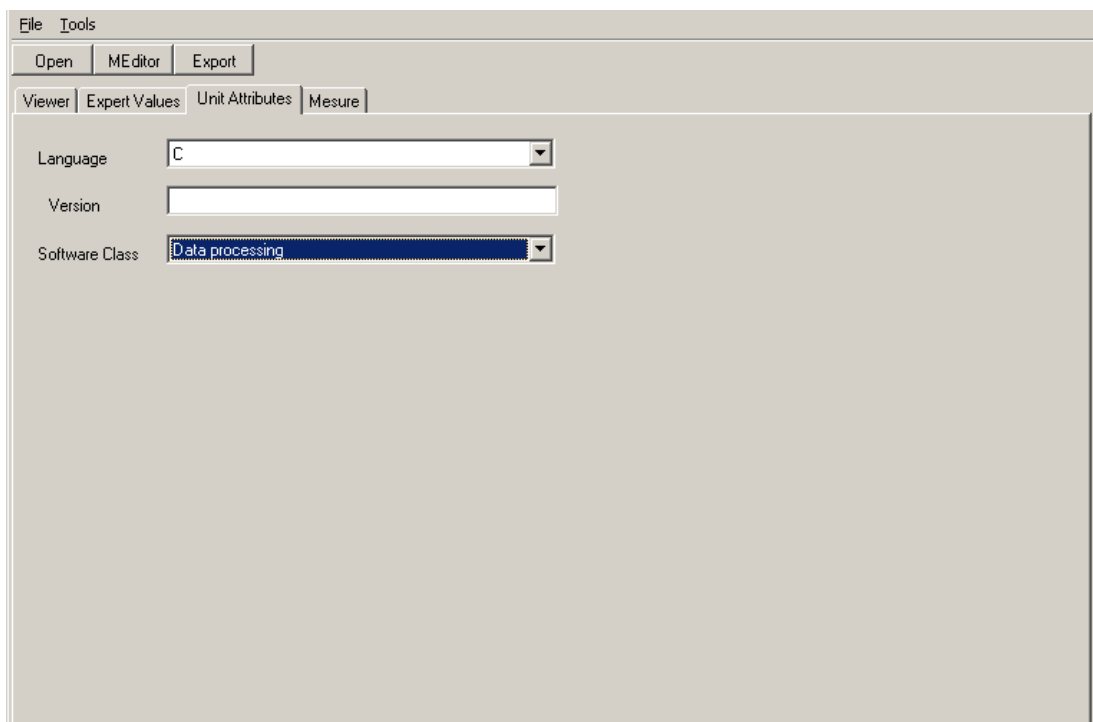


Рис. Д.2. Вкладка для вводу значень додаткових атрибутів компонентів

Code	Name	NAMEFORSCAN	Value
AVRPF	AVERAGE PERCENT FIT	AVERAGE PERCENT FIT	39.28
PME	Percent Modules with Exceptions	Percent Modules with Exceptions	0.00
AVGEM	Average Exceptions per Module	Ave. Exceptions per Module	0
CFF	CONTROL FLOW FIT	CONTROL FLOW FIT	91.92
TM	Total Modules	Total Modules	1
TLOC	Total LOC	Total LOC	59
MLOC	Max LOC	Max LOC	6
AVGLOC	Average LOC	Ave. LOC	59
TNCSS	Total NCSS	Total NCSS	59
MNCSS	Max NCSS	Max NCSS	6
AVGNCS	Ave. NCSS	Ave. NCSS	59
TVG	Total V(g)	Total V(g)	1
MVG	Max V(g)	Max V(g)	1
AVGVG	Ave. V(g)	Ave. V(g)	1
THV	Total Hal's V	Total Hal's V	26
MHV	Max Hal's V	Max Hal's V	26
AHV	Ave Hal's V	Ave Hal's V	26
THE	Total Hal's E	Total Hal's E	52
MHE	Max Hal's E	Max Hal's E	52
AVGHE	Ave Hal's E	Ave Hal's E	52
MFO	Max Fan-out	Max Fan-out	3

Рис. Д. 3. Вкладка для перегляду значень метрик,  
одержаних автоматичним вимірювачем

Code	Name	NAMEFORSCAN
AVRPF	AVERAGE PERCENT FIT	AVERAGE PERCENT FIT
PME	Percent Modules with Exceptions	Percent Modules with Exceptions
AVGEM	Average Exceptions per Module	Ave. Exceptions per Module
CFF	CONTROL FLOW FIT	CONTROL FLOW FIT
TM	Total Modules	Total Modules
TLOC	Total LOC	Total LOC
MLOC	Max LOC	Max LOC
AVGLOC	Average LOC	Ave. LOC
TNCSS	Total NCSS	Total NCSS
MNCSS	Max NCSS	Max NCSS
AVGNCS	Ave. NCSS	Ave. NCSS
TVG	Total V(g)	Total V(g)
MVG	Max V(g)	Max V(g)

Control panel: Navigation buttons (back, forward, search, etc.) and an 'Add Scanner' button.

Рис. Д. 4. Інтерфейс редактора метрик

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Астелс Д. Практическое руководство по экстремальному программированию. Пер. с англ./ Д. Астелс, Г. Миллер, М. Новак// – М.: «Вильямс», 2002. - 320с.
2. Бабак В.П. Статистична обробка даних / В.П. Бабак, А.Я. Білецький / – К.: „МІВВЦ”, 2001 - 388с.
3. Бабенко Л.П. Информационная поддержка повторного использования в программной инженерии на базе UML // Кибернетика и системный анализ. – 2003. – №1. – с.74-80.
4. Бабенко Л.П. Адаптивные компоненты многократного использования в системах генерации программ/ Л.П. Бабенко, Е.М. Кохова, В.В. Синяговская // Кибернетика и системный анализ. - 1991. - №5. – с.145-149.
5. Бабенко Л.П. Онтологічні моделі опису готових ресурсів у розробці програм / Л.П. Бабенко, С.Л. Поляничко // Проб. прогр., №2-3, 2004. - с.173-179.
6. Барский А.Б. Нейронные сети: Распознавание, управление, принятие решений./ А.Б. Барский // - М.: Финансы и статистика. – 2002. – 176с.
7. Брукс Ф. Как проектируются и создаются программные комплексы: мифический человек-месяц/ Ф. Брукс// - М.:Наука, 1979. - 321 с.
8. Брянский Л.Н. Шкалы измерений. Прошлое и настоящее/ Л.Н. Брянский // - Законодательная и прикладная метрология. - 2000, №5. – с.51-55.
9. Вельбицкий И.В. Технология программирования / И.В. Вельбицкий // – К.: Техніка, 1984. – 279с.
10. Вельбицкий И.В. Генератор ассемблеров. Принципы построения быстродействующих синтаксически управляемых ассемблерных систем/ И.В. Вельбицкий, Л.И. Шолмов // УСиМ. – 1974. - №1. – с.88-98.
11. Вентцель Е.С. Теория вероятностей: Учеб. для вузов. – 7-е изд. стер./ Е.С. Вентцель// – М.: Высш. шк., 2001. – 575 с.



12. Гаврилова Т.А. Извлечение и структурирование знаний для экспертных систем / Т.А. Гаврилова, К.Р. Червинская// - М.: Радио и связь – 1992– 200с.
13. Гадышев И. Анализ и обработка данных: специальный справочник/ И. Гадышев// – СПб: Питер, 2001. – 752с.
14. Грис Д. Конструирование компиляторов для цифровых вычислительных машин/ Д. Грис// – М.: Мир – 1975. – 544 с.
15. Грищенко В.Н. Компонентно-ориентированное программирование. Состояние, направление и перспективы развития/ В.Н. Грищенко, Е.М. Лаврищева // Проб. прогр. - 2002.- №1-2 – с.80-90.
16. Джексон П. Введение в экспертные системы.: Пер. с англ. : Уч. пос. – М. : Издательский дом «Вильямс», 2001. – 624 с.
17. Жоголев Е.А. Принципы построения многоязыковой системы модульного программирования // Кибернетика. - 1974.- №4.- с.78-83.
18. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. - М.: Мир, 1976. - 165 с.
19. Заенцев И.В. Нейронные сети: основные модели. - Воронеж: Воронежский Гос. университет. - 1999. – 76с.
20. Изосимов А.В. Метрическая оценка качества программ – М.: МАИ, 1989.- 96с.
21. Крамар Ю. М. Автоматизация процесса контроля применения стиля языка программирования // Проблемы программирования. Материалы четвертой международной научно-практической конференции по программированию «УкрПРОГ2004». Спец. вып. – 2004. - № 2-3. - с. 208-214.
22. Лазарев Ю. Моделирование процессов и систем в MATLAB. Учебный курс. - ВНУ – К., 2004. - 512 с.
23. Математика и кибернетика в экономике. Словарь-справочник. – М.: «Экономика», 1975 – 699с.
24. Математический энциклопедический словарь. Гл.ред. Прохорова Ю.В. - М.: Сов. энциклопедия, 1988. – 847с.

25. Месарович М. Теория иерархических многоуровневых систем/ М. Месарович, Д. Мако, И. Такахара // – М.: «Мир», 1973. – 344с.
26. Николаев В.И., Брук В.М. Системотехника: методы и приложения, Машиностроение/ Николаев В.И., Брук В.М. // - Л., 1985. - 198 с.
27. Бабак В.П. Основи теорії ймовірностей та математичної статистики: Навчальний посібник/ В.П. Бабак, А.Я. Білецький, О.П. Приставка// – К.: КВІЦ, 2003. – 432 с.
28. Поспелов Г.С. Искусственный интеллект – основа новой информационной технологии – М.; Наука, 1988.- 280 с.
29. Потемкин В. Система инженерных и научных расчетов MatLab 5.x (в 2-х томах). Диалог-МИФИ, 1999. – 531с.
30. Сидоров Н.А. Инженерия утилизации программного обеспечения. // Методы проектирования интеллектуальных прикладных программных систем, Киев, 1992, - с.15-22.
31. Сидоров Н.А. Повторное использование - метод утилизации модульного программного обеспечения // Синтез модульных систем обработки данных. - Кишинев, 1988.- 35с.
32. Сидоров Н.А. Повторное использование программного обеспечения // Кибернетика. – 1989. - №3 – с.46-51.
33. Сидоров Н.А. Применение принципов программной инженерии в преподавании основ программирования // УСиМ, март-апрель 1999. – с.78-86.
34. Сидоров Н.А. Система повторного использования программного обеспечения. // Прикладная информатика систем проектирования. - Калининград, 1987.- с. 67.
35. Сидоров Н.А. Средства для утилизации программного обеспечения // УСиМ, 5, 1990.- с. 50-57.
36. Системы управления базами данных и знаний: Справ. Изд. / А.Н. Наумов, А.М.Вендров, В.К.Иванов и др.; Под ред. А.Н.Наумова// - М.: Финансы и статистика, 1991. – 352с.

37. Советов Б.Я., Яковлев С.А. Моделирование систем: Учеб. Для вузов/ Б.Я. Советов, С.А. Яковлев – 3-е изд. – М.: Высш. Шк., 2001. – 343с.
38. Советский энциклопедический словарь / Гл.ред. А.М.Прохоров.- 3-е изд. –М.: Сов. Энциклопедия, 1984. – 1600с.
39. Соммервилл И. Инженерия программного обеспечения/ И. Соммервилл // – М.: ”Вильямс”, 2007 - 624с.
40. Уемов А.И. Вещи, свойства, отношения, Изд-во АН СССР, 1963. – 350с.
41. Холстед М.Х. Начала науки о программах / Пер, с англ..- М.: Финансы и статистика, 1981.- 128 с.
42. Цейтлин Г.Е. Алгебра логики та конструирования программ. – К. Наук. думка – 1994. – 90с.
43. Цейтлин Г.Е., Ющенко Е.Л. Формализованные спецификации и трансформационный синтез программ // Кибернетика и системный анализ. – 1993. - №1. – с.127-153.
44. Черчмен У и др. Введение в исследование операций. М., «Наука», 1968. – 384с.
45. Шафер Д.Ф, Фатрелл Р.Т. Управление программными проектами: достижение качества при минимуме затрат/ Д.Ф. Шафер, Р.Т. Фатрелл// – М.: «Вильямс», 2003. – 1185с.
46. Энциклопедия кибернетики. – К.: Главная редакция украинской советской энциклопедии.