

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(назва факультету)

Кафедра комп'ютерних систем та мереж

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістра

(освітній ступінь)

на тему: **Технології оптимального вибору програмних компонентів
повторного використання при проектуванні комп'ютерних систем**

Виконав: студент (ка) 6 курсу, групи СІМ-61

спеціальності 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

Керівник Дишкант І.М.
(підпис) (прізвище та ініціали)

Керівник Яцишин В.В.
(підпис) (прізвище та ініціали)

Нормоконтроль Луцик Н.С.
(підпис) (прізвище та ініціали)

Завідувач кафедри Осухівська Г.М.
(підпис) (прізвище та ініціали)

Рецензент Марценко С.В.
(підпис) (прізвище та ініціали)

Тернопіль
2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

ЗАТВЕРДЖУЮ

Завідувач кафедри Осухівська Г.М.

« _____ »

2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Дишканту Ігорю Мирославовичу
(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) Технології оптимального вибору програмних компонентів повторного використання при проектуванні комп'ютерних систем

Керівник проекту (роботи) Яцишин Василь Володимирович, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «06» грудня 2022 року №4/7-986

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Принципи застосування компонентів повторного використання, Типи програмного забезпечення в комп'ютерних системах, методи оцінювання програм

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз підходів до проектування комп'ютерних систем на основі повторного використання програмних компонентів 2. Метод оцінювання атрибутів компонентів повторного використання 3. Розробка системи автоматизації оцінювання атрибутів повторно використовуваних компонентів 4. Охорона праці та безпека в надзвичайних ситуаціях. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Актуальність і мета дослідження. 2. Задачі дослідження, об'єкт і предмет, наукова новизна і практична цінність дослідження. 3. Концепція та процес розробки ПЗ на основі ПВК 4. Модель атрибутів ЗС. 5. Вплив атрибутів компонентів на процеси повторного використання 6. Метод оцінювання атрибутів ПВК. 7. Архітектура системи оцінювання атрибутів і метрик 8. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Охорона праці та безпека в надзвичайних ситуаціях</i>	<i>Осухівська Г.М.</i>		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	<i>Прикладні програмні інтерфейси в комп'ютерних системах</i>		<i>виконано</i>
2.	<i>Імплементация методу побудови прикладних програмних інтерфейсів в комп'ютерних системах</i>		<i>виконано</i>
3.	<i>Реалізація та експериментальні дослідження API комп'ютерних систем</i>		<i>виконано</i>
4.	<i>Охорона праці та безпека в надзвичайних ситуаціях</i>		<i>виконано</i>
5.	<i>Оформлення пояснювальної записки</i>		<i>виконано</i>
6.	<i>Оформлення графічного матеріалу</i>		<i>виконано</i>
7.	<i>Попередній захист кваліфікаційної роботи магістра</i>		<i>виконано</i>
8.	<i>Захист кваліфікаційної роботи магістра</i>		

Студент

(підпис)*Душкант І.М.*_____
(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)*Яцишин В.В.*_____
(прізвище та ініціали)

АНОТАЦІЯ

“Технології оптимального вибору програмних компонентів повторного використання при проектуванні комп’ютерних систем” // Кваліфікаційна робота магістра // Дишкант Ігор Мирославович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем та програмної інженерії, група СІм-61 // Тернопіль, 2022 // с. – 84, рис. – 37 , табл. – 14 , аркушів А1 –8 , додат. – 1 , бібліогр. – 23 .

Ключові слова: технологія, оптимальність, вибір, повторне використання, програмне забезпечення, комп’ютерна система.

Метою кваліфікаційної роботи магістра є дослідження технологій оптимального вибору програмних компонентів повторного використання при проектуванні комп’ютерних систем.

Проведено аналіз особливостей процесів реалізації компонентів повторного використання у випадку не систематичного підходу до розробки програмних компонентів комп’ютерних систем.

У роботі запропоновано використання методу експертного оцінювання щодо аналізу атрибутів і вартості робіт з реалізації повторно використовуваних компонентів на основі комбінованих моделей вартості і властивостей, що дало можливість забезпечити адекватність і повноту процесу вибору потенційних кандидатів у ПВК.

Запропоновано метод оцінювання атрибутів ПВК, що дає змогу забезпечити автоматизацію процесу одержання метрик атрибутів на основі розрахунку результатів вимірювань та із застосуванням моделей ознак (атрибутів).

Створено інструмент для проведення вимірювань, пов’язаних з кількісними і статистичними критеріями програмного коду з використанням принципів теорії вимірювань, що дало змогу порівнювати оцінку експертів з автоматично визначеними значеннями метрик і мір.

ABSTRACT

"...Technologies of providing the quality of used components of computer systems software " //Master thesis // Dyshkant Ihor Myroslavovych / Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm-61 // Ternopil, 2022 // p. - 84, fig. - 37, Tab. -14, Sheets A1 – 8, Add. - 1, Ref. - 23.

Keywords: technology, optimality, choice, reuse, software, computer system.

The purpose of the master's qualification work is to research technologies for the optimal selection of reusable software components in the design of computer systems. An analysis of the peculiarities of the implementation processes of reusable components in the case of a non-systematic approach to the development of software components of computer systems was carried out.

The paper proposes the use of the expert evaluation method for the analysis of attributes and cost of works on the implementation of reusable components based on combined models of cost and properties, which made it possible to ensure the adequacy and completeness of the process of selecting potential candidates for PMC.

A method for assessing the attributes of PVC is proposed, which makes it possible to automate the process of obtaining attribute metrics based on the calculation of measurement results and using feature (attribute) models.

A tool was created for carrying out measurements related to quantitative and statistical criteria of the software code using the principles of measurement theory, which made it possible to compare the assessment of experts with automatically determined values of metrics and measures.

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ,
СИМВОЛІВ І СКОРОЧЕНЬ

БД	База Даних
ЖЦ	Життєвий Цикл
КС	Комп'ютерна Система
КПВ	Компонент Повторного Використання
ПЗ	Програмне Забезпечення
ПС	Програмна Система
ЗС	Conception Construction Context Model
ANFIS	Adaptive Network Based Fuzzy Inference System
CAESE	Computer Aided Empirical Software Engineering
CASE	Computer Aided Software Engineering
COCOMO	Constructive Cost Models
GQM	Goal Question Metrics
UML	Unified Modeling Language

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ .	6
ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ПІДХОДІВ ДО ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ ПОВТОРНОГО ВИКОРИСТАННЯ ПРОГРАМНИХ КОМПОНЕНТІВ	13
1.1. Аналіз ефективності підходів до реалізації програмних компонентів комп'ютерних систем	13
1.2. Аналіз підходу повторно використовуваних компонентів.....	18
1.3. Порівняння процесів систематичного і не систематичного створення ПВК	24
1.4. Відбір кандидатів у ПВК.....	26
1.5. Висновки до розділу	29
РОЗДІЛ 2 МЕТОД ОЦІНЮВАННЯ АТРИБУТІВ КОМПОНЕНТІВ ПОВТОРНОГО ВИКОРИСТАННЯ	30
2.1. Метод оптимального вибору компонентів повторного використання у процесі проектування комп'ютерних систем.....	30
2.2. Обґрунтування моделі представлення атрибутів компонентів повторного використання	35
2.3. Формалізація моделі атрибутів повторно використовуваних компонентів..	39
2.4. Моделі властивостей.....	45
2.5. Висновки до розділу	50
РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ АВТОМАТИЗАЦІЇ ОЦІНЮВАННЯ АТРИБУТІВ ПОВТОРНО ВИКОРИСТОВУВАНИХ КОМПОНЕНТІВ	52
3.1. Проектування архітектури системи оцінювання атрибутів ПВК	52
3.2. Опис компонентів архітектури.....	55
3.2.1. Сховища зберігання даних.....	59
3.2.2. Вимірювач атрибутів програмного коду	62
3.3. Висновки до розділу	67

РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	68
4.1. Охорона праці	68
4.2. Планування та порядок проведення евакуації населення з районів наслідків впливу НС техногенного та природного характеру	70
ВИСНОВКИ.....	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	78
ДОДАТОК А ТЕЗИ КОНФЕРЕНЦІЙ	81

ВСТУП

Актуальність теми. Сучасні технологічні можливості цифрової епохи дозволяють забезпечити ефективність проектування, модифікації та створення нових комп'ютерних систем, в тому числі з елементами штучного інтелекту та IoT пристроями. Зважаючи на це, за невеликий інтервал часу та відносно невисоку вартість можна розробити багатофункціональний апаратно-програмний комплекс, що забезпечує зручний інтерфейс користувача.

З метою пришвидшення процесу розробки розумних комп'ютерних систем без втрати якості та ефективності майбутнього функціонування доцільно застосовувати підхід, що базується повторно використовуваних компонентах. Перевагою такого підходу є те, що компоненти, які імплементуються у нову комп'ютерну систему, перевірені практикою і відповідають заявленим функціональним вимогам та критеріям якості. Підтвердженням цього факту є те, що даний підхід використовують такі IT-гіганти як IBM, Microsoft, Hewlett-Packard, що утворили цілі підрозділи для його впровадження та створили власні повторно використовувані компоненти і навіть цілі технології.

У науковій сфері також багато робіт присвячені тематиці повторного використання, створенню і розвитку компонентів, які багаторазово можна застосовувати у різномірних проектах. До когорти закордонних учених, які багато уваги приділяють дослідженню технологій повторного використання варто виділити праці V. Basili, E. Chisofsky, H. Rombach, P. Wegner, серед українських – І. Вельбіцький, М. Сидоров, В. Хоменко.

У результаті застосування технологій повторного використання одержують однойменні компоненти. При не систематичному повторному використанні, компоненти реалізуються за допомогою фрагментів існуючого ПЗ, а потреба і ефективність їх створення виражається через витрати, які не мають бути більшими за витрати передбачені на реалізацію такого функціоналу методами прямої інженерії. У процесі створення повторно використовуваних компонентів найбільш витратним є процес оцінювання властивостей потенційного кандидата у ПВК,

оскільки потребує значного обсягу часу та значної кількості фахівців з високими професійними скілами.

Окрім цього, на сьогодні відсутня процедура формалізованого вибору та оцінки властивостей КПВ і такий процес реалізується на рівні досвіду та інтуїції розробників при аналізі програмного коду ПЗ. Тому створення технології оптимального вибору програмних компонентів повторного використання при проектуванні комп'ютерних систем, є актуальною задачею у сфері ІТ.

Мета і задачі дослідження. Метою кваліфікаційної роботи магістра є дослідження технологій оптимального вибору програмних компонентів повторного використання при проектуванні комп'ютерних систем.

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

- дослідження процесів створення компонентів повторного використання при проектуванні комп'ютерних систем;
- розробка методу оцінювання атрибутів програмних компонентів повторного використання для оптимального вибору кандидатів у ПВК;
- визначення атрибутів повторно використовуваних компонентів ПЗ;
- обґрунтування моделей для опису атрибутів компонентів повторного використання;
- побудова архітектури та реалізація системи автоматизації методу оцінювання атрибутів компонентів повторного використання.

Об'єкт дослідження – процеси розробки програмних компонентів з використанням підходу повторного використання.

Предмет дослідження: моделі, методи і засоби оцінювання атрибутів компонентів повторного використання при розробці програмних компонентів комп'ютерних систем.

Методи дослідження. У процесі розв'язання поставлених задач кваліфікаційної роботи магістра використано такі методи: синтез – під час визначення метрик атрибутів ПВК; моделювання і системний аналіз – під час

створення моделей атрибутів ПВК; аналіз і проектування – під час розробки архітектури системи автоматизації оцінювання атрибутів ПВК.

Наукова новизна отриманих результатів. Наукова новизна полягає у вирішенні науково-практичної задачі оцінювання атрибутів потенційних компонентів у ПВК, при цьому одержано наступні результати:

– уперше запропоновано метод оцінювання атрибутів ПВК, що дає змогу забезпечити автоматизацію процесу одержання метрик атрибутів на основі розрахунку результатів вимірювань та із застосуванням моделей ознак (атрибутів).

– запропоновано процедуру створення моделей атрибутів, що базується на принципах підходу системного аналізу та методу «Ціль-Питання-Метрика», що дало змогу забезпечити ефективність експертного оцінювання характеристик ПВК.

Практичне значення одержаних результатів. Впровадження методу і системи автоматизації щодо оцінювання атрибутів компонентів повторного використання дають змогу підвищити ефективність процесу розробки програмних компонентів комп'ютерних систем.

Публікації. Результати кваліфікаційної роботи апробовані на X науково-технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року) як тези конференцій.

1. Яцишин В.В., Шаблій Н.Р., Дишкант І.М. Процес формування програмних компонентів повторного використання при реалізації комп'ютерних систем. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С.81.

2. Яцишин В.В., Дишкант І.М. Архітектура засобу підтримки процесу оцінювання потенційних компонентів повторного використання. Матеріали X науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С.82.

Структура роботи. Кваліфікаційна робота містить розрахунково-пояснювальну записку та графічний матеріал. До складу записки входить вступу, 4 розділи, загальні висновки, список використаних джерел і додатки. Обсяг роботи: розрахунково-пояснювальна записка – 84 арк. формату А4, графічна частина – 8 аркушів формату А1.

РОЗДІЛ 1

АНАЛІЗ ПІДХОДІВ ДО ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ ПОВТОРНОГО ВИКОРИСТАННЯ ПРОГРАМНИХ КОМПОНЕНТІВ

1.1. Аналіз ефективності підходів до реалізації програмних компонентів комп'ютерних систем

Технології розробки сучасних комп'ютерних систем та їх компонентів характеризуються застосуванням різних методів і засобів, які дають змогу підвищити ефективність їхнього проектування та реалізації. До основних таких підходів належать [1]:

- оптимізовані процеси організаційного і технічного забезпечення проектування систем;
- забезпечення якості процесів на стадіях життєвого циклу (ЖЦ);
- оптимізація та реорганізація стадій ЖЦ при реалізації комп'ютерних систем;
- зменшення затрат часу і фінансів на внесення змін до ПЗ;
- застосування підходу повторного використання компонентів.

Оптимізація, і як наслідок, вдосконалення організаційних і технічних процесів при реалізації програмних компонентів КС передбачає три види діяльності, які пов'язані з командою розробки, робочими місцями та керуванням процесу виконання проекту.

Перший вид діяльності вимагає застосування комплексу заходів щодо кількості членів команди та навиків, якими вони повинні володіти. Модель СОСОМО визначає сукупність цих заходів за рейтингом, як найвищий [2]. При формуванні команди, яка буде виконувати проект, можуть бути використанні відомі ще з 70-х років підходи бригада «chief surgeon» [3] чи «software studio», а також сучасні організаційні структури як от, наприклад, «pair programming» [4].

Наступний напрям діяльності, що дозволяє підвищити ефективність процесу імплементації ПЗ комп'ютерних систем, передбачає впровадження заходів щодо

забезпечення комфортних робочих місць розробників. Так відомі IT-гіганти, наприклад, IBM, Google, Xerox та інші, витрачають значні фінанси для їх організації, завдяки чому досягають збільшення продуктивності розробників до 12% [5].

Третій вид діяльності орієнтований на підвищення ефективності процесів управління проектом. З практики відомо, що використання поганих практик управління або його відсутність значно підвищують рівень витрат у порівнянні з іншими факторами. Для прикладу, у випадку, коли ще не визначені вимоги до компонентів ПЗ на етапі раннього планування, то в подальшому це зумовить значене їх доопрацювання, що знову ж таки негативно відобразиться на часі і фінансах його виконання [6].

Оптимізація та реорганізація стадій ЖЦ. У результаті аналізу трудомісткості виконання процесів на різних стадіях ЖЦ встановлено, що кожен з етапів характеризується різним ступенем трудозатрат. Тому, для забезпечення ефективності їх виконання необхідно впроваджувати засоби автоматизації (CASE-засоби), які дозволять підвищити продуктивність загального процесу реалізації проекту та зменшити відповідні витрати.

Імовірно, що для зниження затрат на виконання проекту, ефективним способом була б повна або часткова заміна стадії ЖЦ, наприклад, за рахунок автоматичної імплементації відповідних процесів. Проте, використання такого підходу не завжди можна забезпечити, але на практиці все ж використовується при проектуванні архітектури та генерації програмного коду. У випадку, коли предметна область є добре вивченою, то можливий автоматичний перехід від специфікації вимог до їх програмної реалізації. Проте, коли домен є недостатньо дослідженим автоматичну генерацію застосувати практично неможливо, оскільки для забезпечення якості такого процесу необхідно виділити значні фінансові ресурси. [10, 12, 14].

Заходи щодо зменшення затрат часу і фінансів при зміні вимог до ПЗ орієнтовані на розробку інструментів автоматизованого формування специфікацій

(візуальні діаграми, пошук багів, моделі компонентів і т.д.). Це пов'язано з тим, що існує проблема доопрацювання потреб на ранніх етапах проекту.

Застосування підходу повторного використання є шляхом, що сприяє підвищенню продуктивності і зниженню затрат при реалізації програмних компонентів комп'ютерних систем, що бере свій початок ще 80-х років ХХ ст. [13]. Даним підходом передбачається зменшення об'єму компонентів, які створюються, а натомість пропонується використовувати вже існуючі програмні рішення. Окрім забезпечення зростання продуктивності програмування, підхід повторного використання компонентів дозволяє підвищити надійність і якість як окремих програмних частин системи, так і ПЗ загалом. Варто відмітити, що крім зменшення витрат на написання програмного коду, знижуються затрати як часові, так і фінансові на етапах тестування, у процесі формування документації і супроводу ПЗ.

Згідно досліджень [14, 15], застосування підходу повторного використання компонентів забезпечують зростання ефективності при реалізації проектів ПЗ на рівні 10% у рік – від 6% при створенні програмних застосувань реального часу до 16% у випадку реалізації програмного забезпечення для великих корпорацій або ринку масового використання. Повторне використання при написанні програмного забезпечення застосовується такими світовими компаніями, як Ericsson, Hewlett-Packard, IBM, Motorola, а частка, що припадає на компоненти, імplementовані таким чином, згідно [16], становить: Ericsson – майже 90%, Hewlett-Packard у діапазоні від 25% до 50%; Motorola – до 85%.

Згідно джерела [16], застосування підходу повторного використання передбачає зменшення тривалості до виходу програмного продукту на ринок у від 2-ох до 5-ти раз, зменшення помилок у 4-9 разів, витрати на супровід знижуються у 5-10 разів, а загальна вартість проекту знижується від 15% до 70% [16]. Таких показників компанії досягають завдяки скороченню затрат на:

- розробку нових програмних компонентів;
- вартості на написання програмного коду.

Застосування у проектах комп'ютерних систем компонентів повторного використання забезпечує зростання надійності функціонування програмного забезпечення, оскільки використовуються уже готові та відтестовані компоненти.

Підтвердженням важливості та необхідності застосування підходу повторного використання є те, що сформовано групу моделей ЖЦ до складу якої входять три моделі, які використовують принципи цього підходу [17].

Перша модель є моделлю швидкої розробки, що описує реалізацію програмних компонентів комп'ютерної системи у вигляді зміни програмного коду існуючого ПЗ з подальшим внесення змін до інших продуктів чи сервісів на етапах ЖЦ. Модель швидкої розробки представлено на рис.1.1.

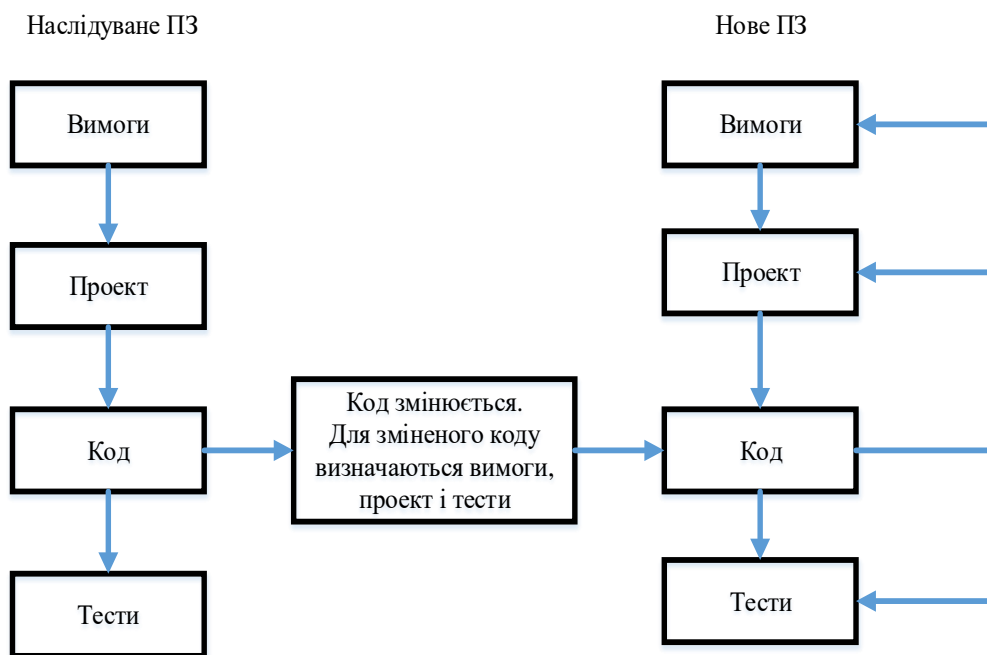


Рис 1.1. Модель розробки із заміною коду

Друга модель – ітеративна. Вона базується на підході повторного використання і передбачає аналіз ПЗ, яке буде наслідуватись, з подальшою реалізацією нового ПЗ шляхом послідовного внесення змін до існуючих частин програмного забезпечення. Ітеративну модель проілюстровано на рис.1.2.

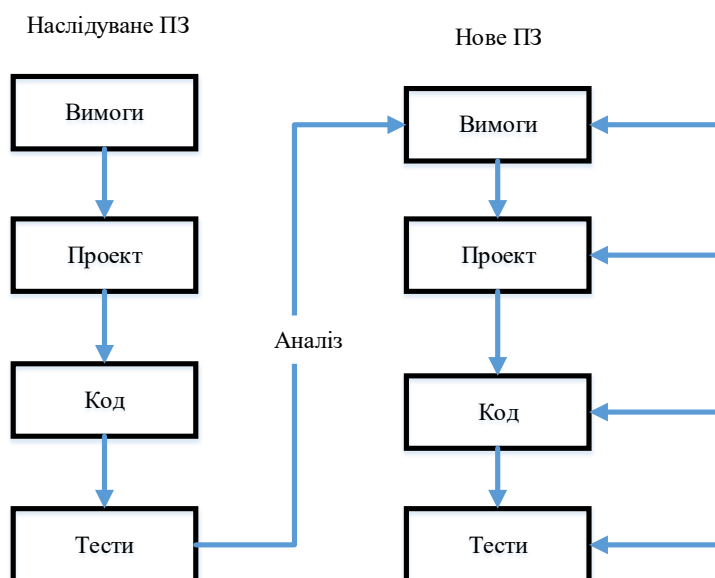


Рис. 1.2. Послідовна модель

Третій тип моделі ЖЦ, в основі якої лежить підхід повторного використання – повна модель. Дана модель використовує базу даних компонентів повторного використання, що сформована на базі робочого ПЗ, а після цього виконуються роботи над створенням нового ПЗ з використанням даних, наявних у репозиторії. Структуру повної моделі життєвого циклу при застосуванні підходу повторного використання показана на рис.1.3.

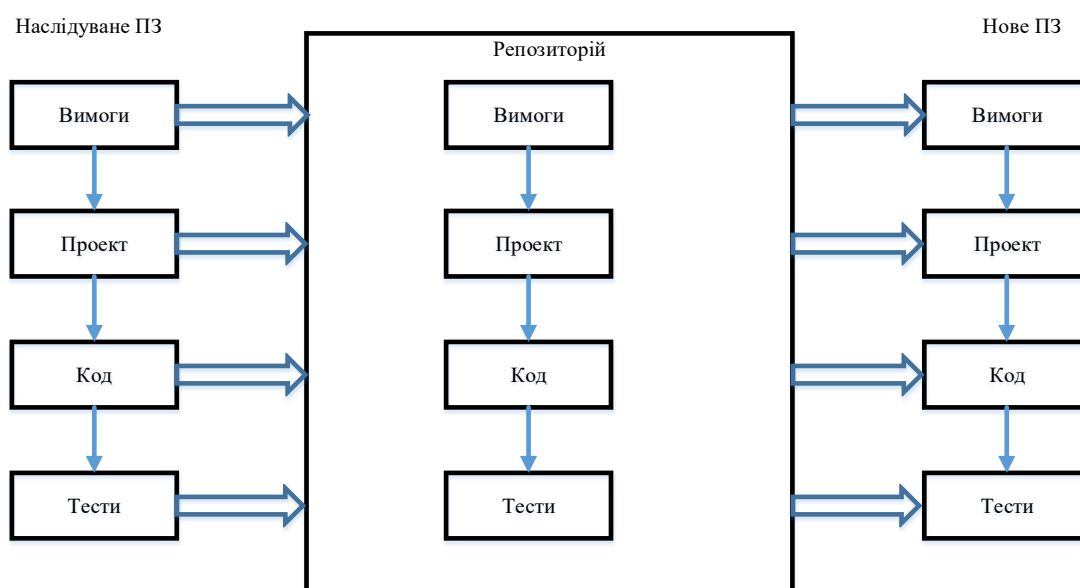


Рис. 1.3. Комплексна модель

Виходячи з проведеного аналізу публікацій присвячених підходу повторного використання, можна зробити висновок про те, що такий підхід є невід'ємним атрибутом на усіх етапах і процесах реалізації проекту програмних компонентів комп'ютерних систем.

1.2. Аналіз підходу повторно використовуваних компонентів

В основі повторного використання ПЗ лежить принцип застосування вже існуючих програмних компонентів при створенні нового програмного забезпечення [17, 19]. Суть такого підходу полягає у використанні раніше одержаних результатів у нових умовах застосування. У широкому розумінні підходу повторного використання виділяють два рівні результатів. Перший рівень представляє собою повторне використання ідей та знань, а другий – готових програмних продуктів і їх структурних елементів [18]. Виходячи з цього, специфікація вимог, аналітичні результати, результати проектування, тест-кейси і документація також можуть повторно використовуватися.

Під компонентом програмного забезпечення розуміють таку його частину, яка має ліміти застосування, володіє інтерфейсом і функціоналом використання [2]. Варто відзначити, що компонент повторного використання ПЗ є елементом, який може кілька разів бути вбудованим хоча в одну програмну систему [2].

Для імплементації нового ПЗ з використанням підходу повторного використання передбачено 2 процеси, які проілюстровано на рис.1.4.

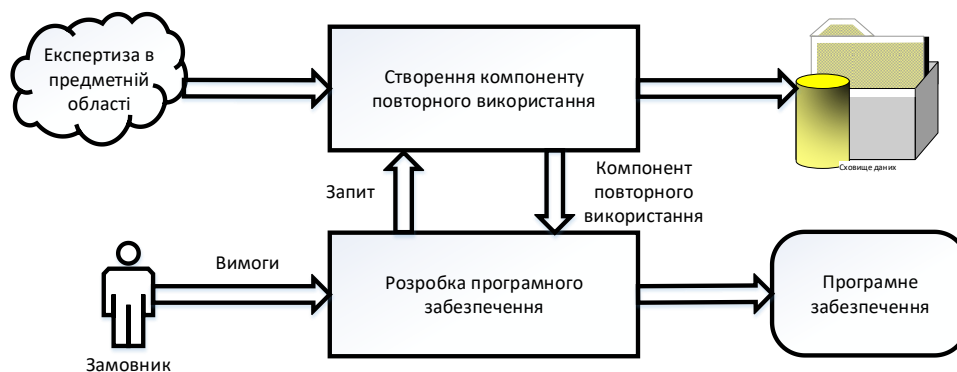


Рис. 1.4. Створення нового ПЗ на основі підходу повторного використання

Суть першого процесу полягає у створенні ПЗ, а іншого – у продукуванні повторно використовуваних компонентів (ПВК). Однак, потрібно відмітити, що перший процес передбачає використання ПВК. Розробка повторно використовуваних компонентів може виконуватися двома шляхами: синхронним і асинхронним, як показано на рис. 1.5 [4].

У відповідності із запитом розробників, спочатку виконується пошук компонентів у базі даних (репозиторії), а після цього формується відповідь, що містить результати, які найбільше відповідають вимогам, наявних у запиті.

У випадку знаходження ПВК, він, за необхідності, підлаштовується до вимог запиту, перевіряється і віддається розробнику нового ПЗ. В іншому випадку – потрібно розробляти такий компонент самостійно або використовувати більш елементарні складові, які в подальшому будуть збережені і передані для реалізації нового ПЗ [2].

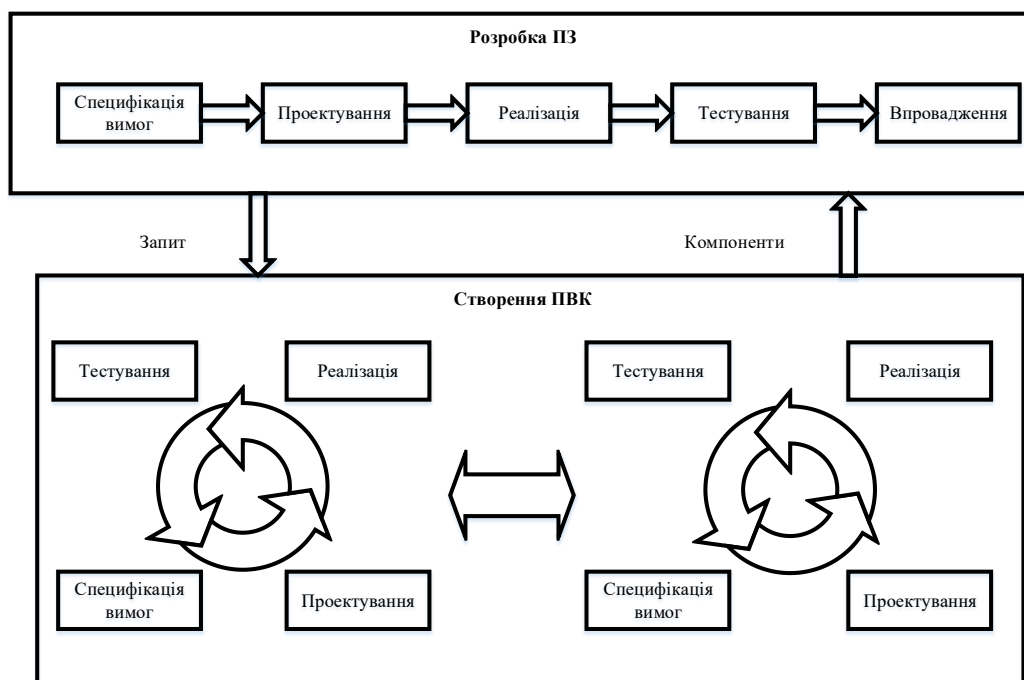


Рис. 1.5. Розробки ПЗ з використанням ПВК

База даних компонентів повторного використання повинна наповнюватися за рахунок їх створення, незалежно від безпосередньої реалізації нового ПЗ.

Асинхронне виробництво ПВК орієнтоване на наповнення бази даних якомога більшою кількістю елементів, оскільки це є критичним для успіху практичного застосування підходу повторного використання.

Окрім цього, за варіантами організації та реалізації процесів створення ПВК розрізняють систематичне і не систематичне повторне використання [2].

При систематичному повторному використанні передбачається створення компоненту як частини нового ПЗ, що забезпечує автоматизацію функцій для деякої предметної області і його подальшого використання, як ПВК. Також даний тип повторного використання передбачає вже існуючі експертні знання про домен. Необхідна умова систематичності повторного використання диктується доцільністю і необхідністю багаторазового застосування ПВК [3, 4].

По іншому, ПВК реалізовані на основі систематичного повторного використання є уніфікованими компонентами ПЗ [17], а процеси їх створення показано на рис. 1.6.

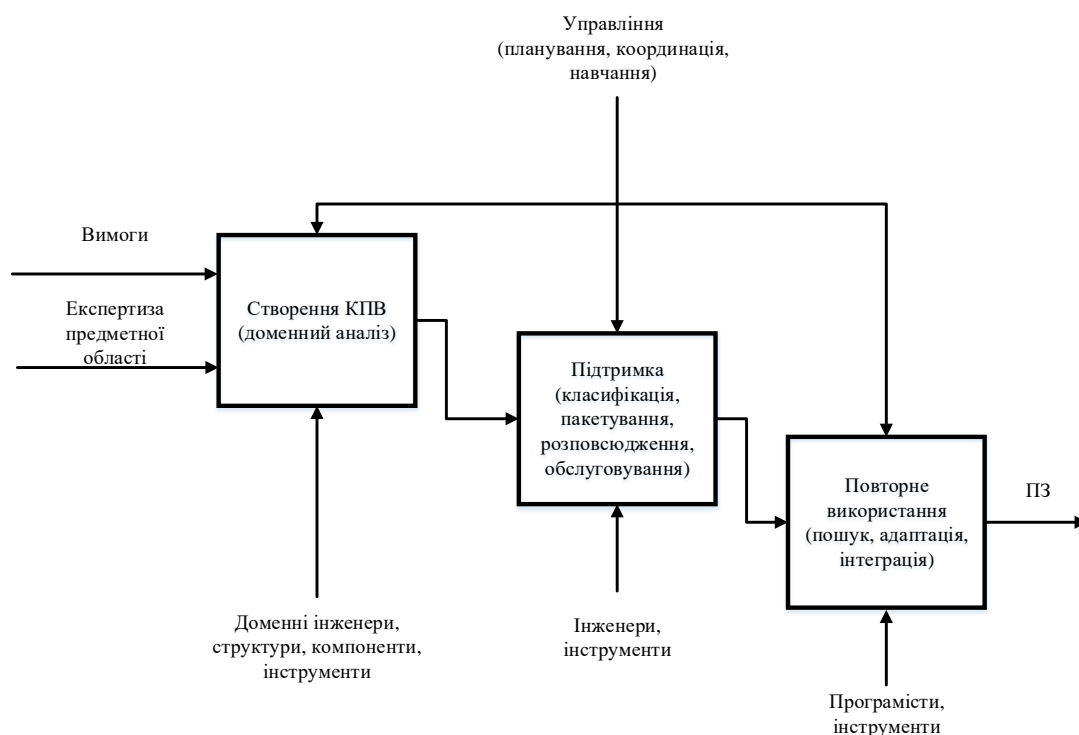


Рис. 1.6. Систематичне повторне використання

Доменний аналіз лежить в основі систематичного ПВК при якому, застосовується дедуктивний шлях реалізації компонентів [2, 5]. Створення компонентів повторного використання ведеться таким чином, щоб задовольняти усі вимоги, які висуваються у напрямку зверху до низу. Компоненти, які одержані у результаті застосування підходу, наведеного на рис. 1.6, представляють собою фрагменти програмних систем, особливості створення яких є добре вивченими і мають широке поле їх реалізацій. Прикладами компонентів, отриманих внаслідок систематичного повторного використання є складові елементи компіляторів, системи пошуку інформації, компоненти архіваторів, генератори звітів та інші.

Зберігання ПВК, які одержано таким чином, не вимагають використання значних інфраструктурних ресурсів, оскільки їх є невелика кількість. Процес розробки ПВК більш подібний до системного проектування, чим до аналізу. На рис. 1.7 показана організаційна інфраструктура, яка необхідна при застосуванні систематичного підходу [7].

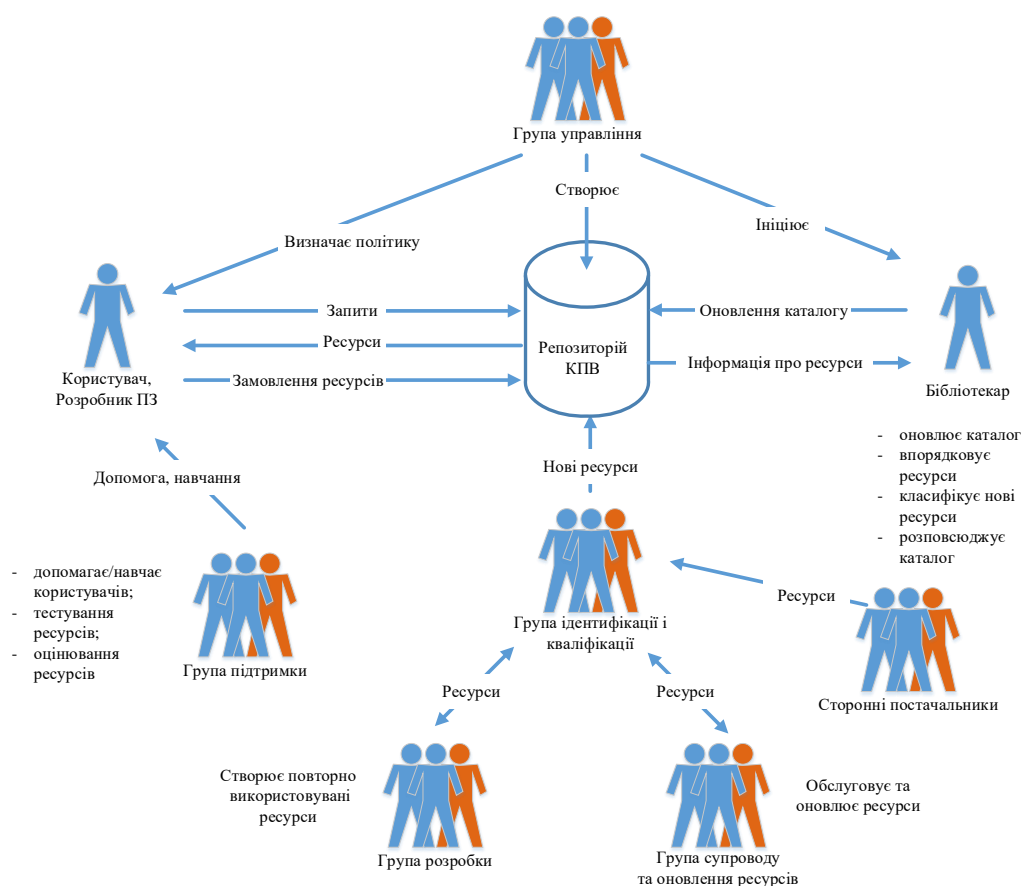


Рис. 1.7. Організаційна схема при систематичному підході

При використанні апаратного забезпечення комп'ютерних систем з часом відбувається його фізичне та моральне зношення. На відміну від технічних пристроїв, ПЗ не піддається фізичному зношенню, але відбувається його моральне старіння. Фактори, які відображають моральне зношення ПЗ, можна інтерпретувати наступним чином:

- застарілість методів, що лежать в основі роботи ПЗ;
- відсталість апаратних характеристик ПЗ, пов'язана з вдосконаленням технологій їх реалізації та супроводу;
- невідповідність вимогам нових стандартів, що використовуються при взаємодії різних програмних систем, які формують складнощі при їх інтеграції;
- зростання витрат на супровід ПЗ;

Наявність таких ознак при використанні ПЗ супроводжується необхідністю його утилізації або виводу з експлуатації. При не систематичному повторному використанні компоненти наявного програмного забезпечення інтегруються у нове. При такому підході виконується виділення і вибір компонентів з існуючого ПЗ, при потребі здійснюється їх перетворення та/або настройка з подальшою інтеграцією у нове програмне забезпечення.

У випадку не систематичного повторного використання компоненти існуючого ПЗ можуть бути використанні по аналогії до принципів тестування «чорної» або «білої скриньки» (рис. 1.8).



Рис. 1.8. Підхід не систематичного ПВК

Це означає, що при використанні компоненту як «чорної скриньки», він не зазнає змін, в іншому випадку – можливі зміни його структури та/або концепції. [19]

В залежності від компонентної структури ПЗ і потрібного рівня грануляції визначення потенційних кандидатів у ПВК може виконуватися шляхами [5], які представлені на рис. 1.9.

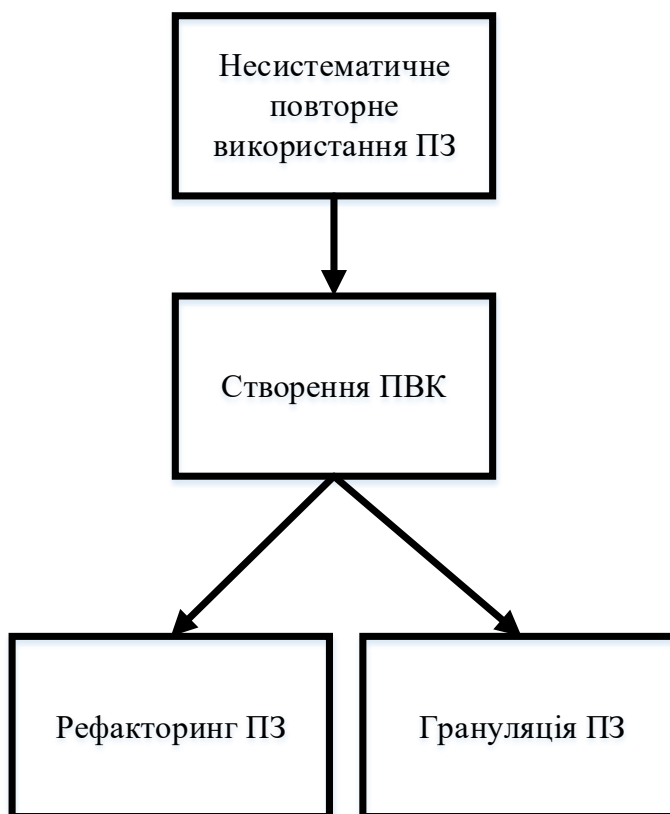


Рис. 1.9. Шляхи виявлення потенційних ПВК

На рис. 1.9 під рефакторингом ПЗ розуміють декомпозицію конструктивно завершених складових ПЗ [13]. Зрозуміло, що виділення програмних фрагментів може бути реалізоване лише тоді, коли вони є інкапсульованими, а самі капсули синтаксично розділені. Для прикладу, капсулою може служити метод, бібліотека чи клас;

Грануляція ПЗ передбачає виокремлення функцій або частин з «моноліту» і створення на основі них інкапсульованих компонентів [14].

1.3. Порівняння процесів систематичного і не систематичного створення ПВК

Не систематичне повторне використання передбачає виконання процесів, які формують два етапи: створення і застосування ПВК у новому ПЗ (рис.1.10).

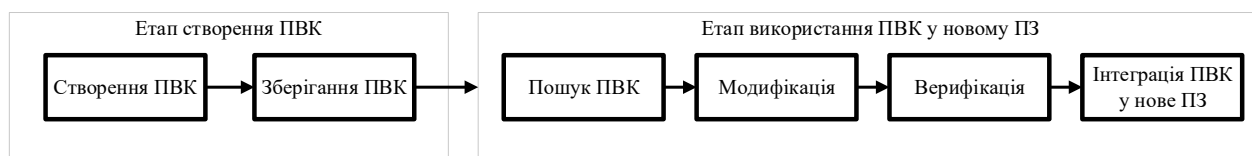


Рис. 1.10. Процеси та етапи не систематичного ПВК

Перший етап містить такі процеси:

- створення ПВК – на основі застосування рефакторингу або грануляції виконується виділення частини існуючого ПЗ, що в перспективі можна багаторазово використати і реалізувати на їхній основі ПВК;
- збереження ПВК – запис ПВК у базу даних для майбутнього використання.

Другий етап характеризується наступними процесами:

- пошук ПВК – пошук у базі даних ПВК, що відповідають умовам запиту;
- настройка чи модифікація ПВК – налаштування ПВК на основі зміни його параметрів або модифікації перед тим, як інтегрувати у нове ПЗ;
- перевірка ПВК при умові виконання попереднього етапу;
- інтеграція ПВК – впровадження ПВК.

Операції щодо створення КПВ можна згрупувати за фазами ідентифікації та кваліфікації, як показано на рис. 1.11.

Фаза ідентифікації містить операції визначення та вибору потенційних кандидатів у ПВК. Визначення компонентів з існуючого ПЗ можна виконати двома шляхами вручну розробником або автоматизованим шляхом з використанням спеціалізованих інструментів. Визначені компоненти помічаються як потенційні кандидати для ПВК.

Вибір компонентів передбачає виконання процедур оцінювання щодо придатності бути ПВК з подальшим переходом до фази кваліфікації.

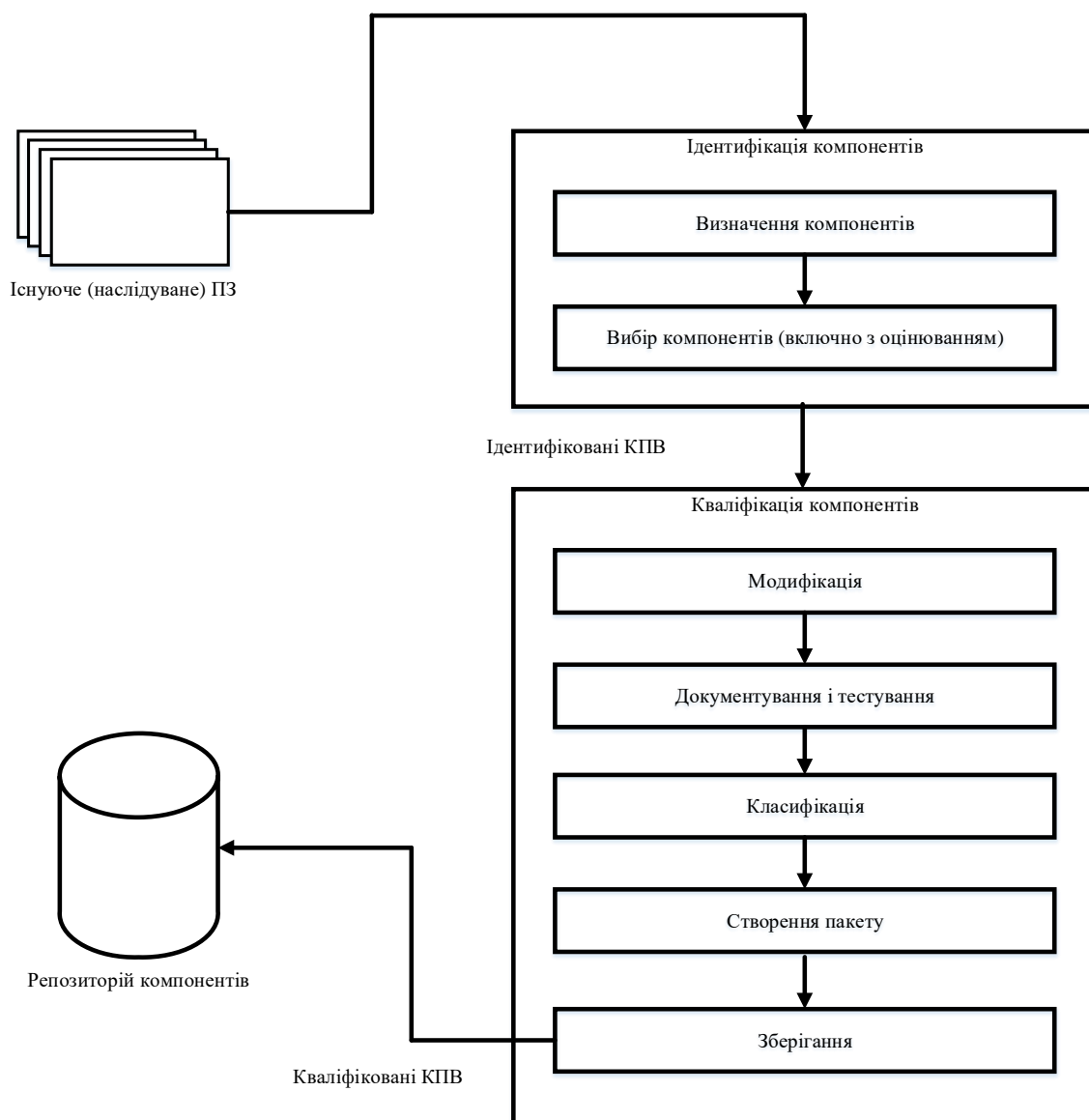


Рис. 1.11. Операції процесу створення ПВК

У результаті вибору компонентів, як потенційно повторно використовуваних, можна знизити вартість розробки внаслідок виключення компонентів, які не відповідають вимогам до формування ПВК і відповідно вони не потрапляють на стадію кваліфікації.

Стадію кваліфікації формують процеси внесення змін (модифікація) до ПВК, створення документації, тестування, групування, формування пакетів і зберігання компонентів.

Внесення змін у ПВК виконується розробником програмного забезпечення у відповідності до вимог, які висуваються до ПВК, шляхом модифікації програмного коду.

Створення документації ПВК передбачає формування його специфікації, як наприклад, коментарі стрічок коду та метаопис, які в подальшому дозволять проводити пошук компонентів у БД і застосовувати його у новому ПЗ.

Тестування реалізованого ПВК проводиться з метою його перевірки на відповідність вимог з використанням відповідних технологій. Групування чи класифікація ПВК покладається на розробників або адміністраторів бази даних, з врахуванням специфікацій компонентів, що забезпечує категоризацію і пошук ПВК. Формування пакетів або «пакетування ПВК» виконується з метою забезпечення цілісності компонентів при їх зберіганні і міграції. Розроблені пакети ПВК поміщаються у сховище шляхом їх переміщення у визначені місця файлової системи чи спеціалізовану базу даних.

1.4. Відбір кандидатів у ПВК

Не систематичне повторне використання в процесі ідентифікації передбачає виконання операції відбору потенційних компонентів, які можуть служити як ПВК.

В загальному випадку, процедура відбору компонентів представлена на рис. 1.12, а use case діаграма цієї операції наведена на рис.1.13.

При відборі компонентів припускають, що створення ПВК, по відношенню до витрат, не повинно бути більшим чи рівним у порівнянні з витратами, необхідними при реалізації такого є компоненту з нуля [12].

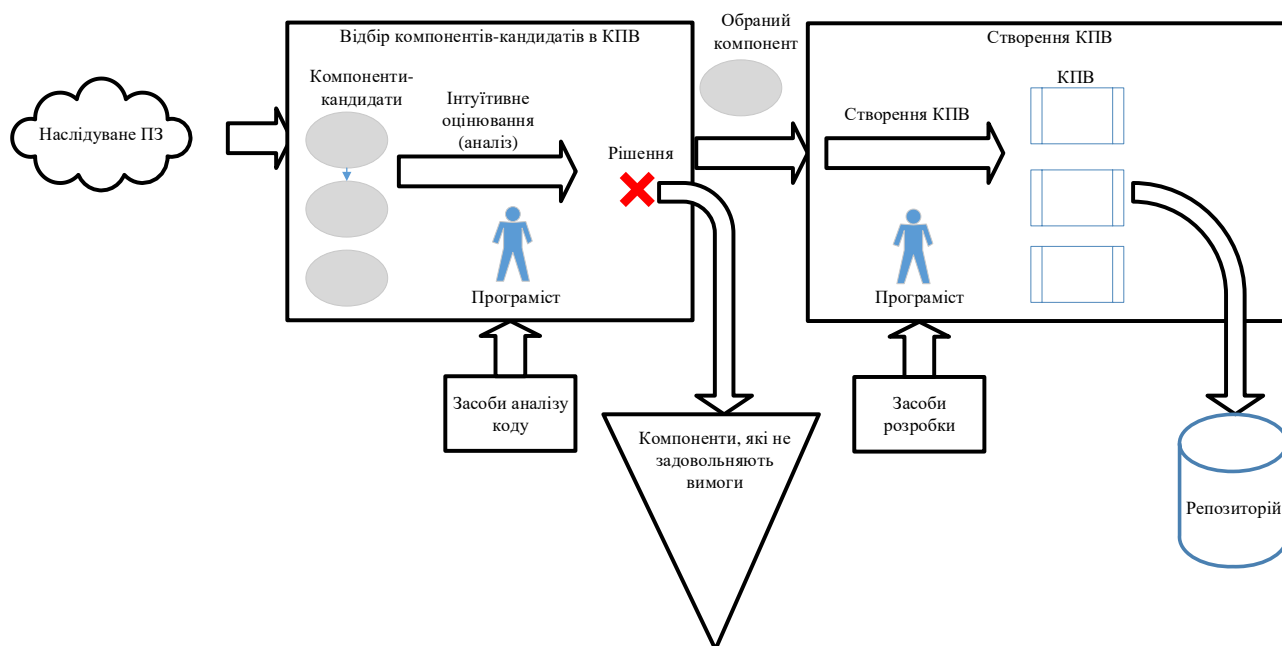


Рис 1.12. . Відбір компонентів як потенційних ПВК

Оцінювання витрат передбачає реалізацію двох можливих варіантів, які полягають у прямому їх визначенні на основі даних про затрати на виконання процесу повторного використання або непрямому – використовуються знання щодо відповідності властивостей, реалізованих у компоненті, функціональним вимогам та вимогам якості, сформованих у запиті до ПВК. У зв'язку з цим, відбір потенційних компонентів ПЗ базується на двох видах моделей:

- витрат – передбачає прийняття рішень про доцільність повторного використання компоненту з врахуванням вартості його перетворення у ПВК [7];
- атрибутів – передбачає прийняття рішень про можливість повторного використання у результаті оцінювання атрибутів, які інтерпретують здатність до подальшого багаторазового використання [9].

Вартість проведення робіт при не систематичному повторному використанні обчислюється наступним чином

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

де C_{CA} – вартість процедур при добуванні компонента з існуючого ПЗ;

C_{WB} – вартість внесення змін у компонент S_i у ПВК T_i .



Рис. 1.13. Use case діаграма при відборі потенційного ПВК

Враховуючи те, що відбір потенційних компонентів у ПВК виконується після добування компонента з існуючого ПЗ, то складова C_{CA} не розглядається.

При використанні моделей вартості визначається доцільність реалізації ПВК із застосуванням однойменного критерію, проте відхилення оцінок витрат може бути дуже великим. Окрім цього, практичне застосування таких моделей при не систематичному повторному використанні є не ефективним, оскільки існує багато S_i , та операцій, які потрібно виконати при переході від S_i до T_i .

Модель атрибутів будується за допомогою PVB («Property-values based»), що представляє собою підхід формування асоціацій вимог та компонентів із складеними атрибутами [15]. Після цього виконується деталізація складних атрибутів на основі більш простих, які можна прямо вимірювати.

У результаті проведеного аналізу можна зробити висновок про те, що моделі, які застосовуються при відборі потенційних компонентів у ПВК при несистематичному повторному використанні варто застосовувати гібридний підхід. В основі гібридного підходу лежить модель атрибутів, а при формуванні сукупності атрибутів застосовується вартісна модель.

1.5. Висновки до розділу

У даному розділі одержано наступні результати:

1. Проаналізовано сучасні технології забезпечення якості атрибутів повторно використовуваних компонентів при проектуванні розумних комп'ютерних систем, що дало змогу визначити критерії впливу на ефективність їхньої імплементації в процесі побудови та експлуатації систем.

2. Проведено аналіз особливостей процесів реалізації компонентів повторного використання у випадку не систематичного підходу до розробки програмних компонентів комп'ютерних систем, що дало змогу підтвердити необхідність впровадження процедур при виборі та ідентифікації потенційних компонентів на роль повторно використовуваних.

3. Досліджено існуючі моделі для прийняття рішення щодо доцільності використання компонентів повторного використання на основі оцінювання вартості робіт та атрибутів потенційних компонентів, що дало змогу обґрунтувати їхнє застосування при виробництві ПВК.

4. Запропоновано використання методу експертного оцінювання щодо аналізу атрибутів і вартості робіт з реалізації повторно використовуваних компонентів на основі комбінованих моделей вартості і властивостей, що дало можливість забезпечити адекватність і повноту процесу вибору потенційних кандидатів у ПВК.

РОЗДІЛ 2

МЕТОД ОЦІНЮВАННЯ АТРИБУТІВ КОМПОНЕТІВ ПОВТОРНОГО ВИКОРИСТАННЯ

2.1. Метод оптимального вибору компонентів повторного використання у процесі проектування комп'ютерних систем

Для забезпечення та оцінювання якості повторно використовуваних компонентів програмного забезпечення при проектуванні комп'ютерних систем запропоновано скористатись експертними технологіями.

Застосування експертних технологій передбачає використання оцінок, які кількісно визначають процеси або явища, які неможливо прямо виміряти [3]. Потрібно відмітити, що методи експертного оцінювання належать до групи методів прогнозування, які дозволяють створювати адекватні моделі об'єктів на основі суджень експертів [4]. В якості експертів можуть бути як окремі особи, так і колективи. В залежності від цього, розрізняють оцінки, як показано на рис. 2.1.

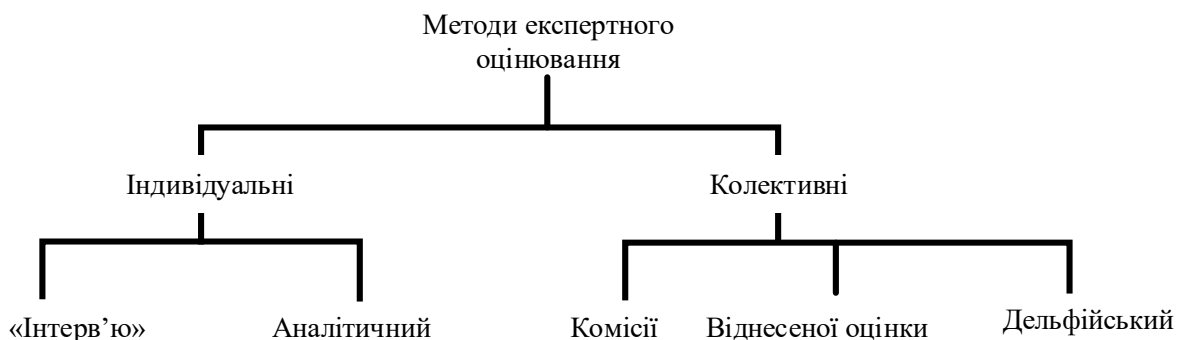


Рис. 2.1. Типи оцінок при експертному оцінюванні

Перший метод індивідуального експертного оцінювання «інтерв'ю» базується на основі результатів спілкування інженера та експерта з метою одержання відповідей на попередньо підготовлені запитання щодо прогнозу розвитку процесу чи явища.

Аналітичний метод експертного оцінювання передбачає самостійну роботу експерта щодо дослідження трендів, потенційних станів і розвитку об'єктів прогнозування.

Метод «комісії» ґрунтується на підході публічного колективного висловлення думок щодо об'єкту прогнозування і формування спільної, узгодженої позиції щодо нього. Однак при застосуванні цього методу існує імовірність впливу думок одного експерта на думки іншого, що може призвести до некоректних результатів.

Використання методу мозкового штурму передбачає розділення на дві групи експертів – перша група висловлює думки, а друга обирає найбільш оптимальне рішення.

Дельфійський метод оцінювання заснований на непрямих колективних дискусіях і реалізується за допомогою індивідуальних послідовних анкетувань, які повторюються до тих пір, поки не буде одержано узгоджений результат.

На практиці використання класу методів індивідуального експертного оцінювання характеризується високим рівнем суб'єктивного впливу на результат.

Методи колективного оцінювання дозволяють зменшити залежність результату від думок конкретного експерта, забезпечити високу точність і якість прогнозування, однак залучення великої кількості експертів передбачає значні фінансові витрати [8].

У процесі вибору потенційних компонентів у ПВК, в якості об'єкту експертного аналізу є компонент ПЗ, який описується атрибутами, що представляють цінність з точки зору повторного використання. У такому разі реалізація методу передбачає виконання процесів, які проілюстровані на рис. 2.2.:

- оцінювання атрибутів, які є важливими в контексті багаторазового повторного використання;
- оцінювання придатності потенційного компонента у ПВК і прийняття рішення щодо його реалізації чи не реалізації.



Рис. 2.2. Процеси оцінювання атрибутів потенційних ПВК

Алгоритм виконання методу експертного оцінювання, що враховує аспекти описаних вище процесів показано на рис. 2.3.

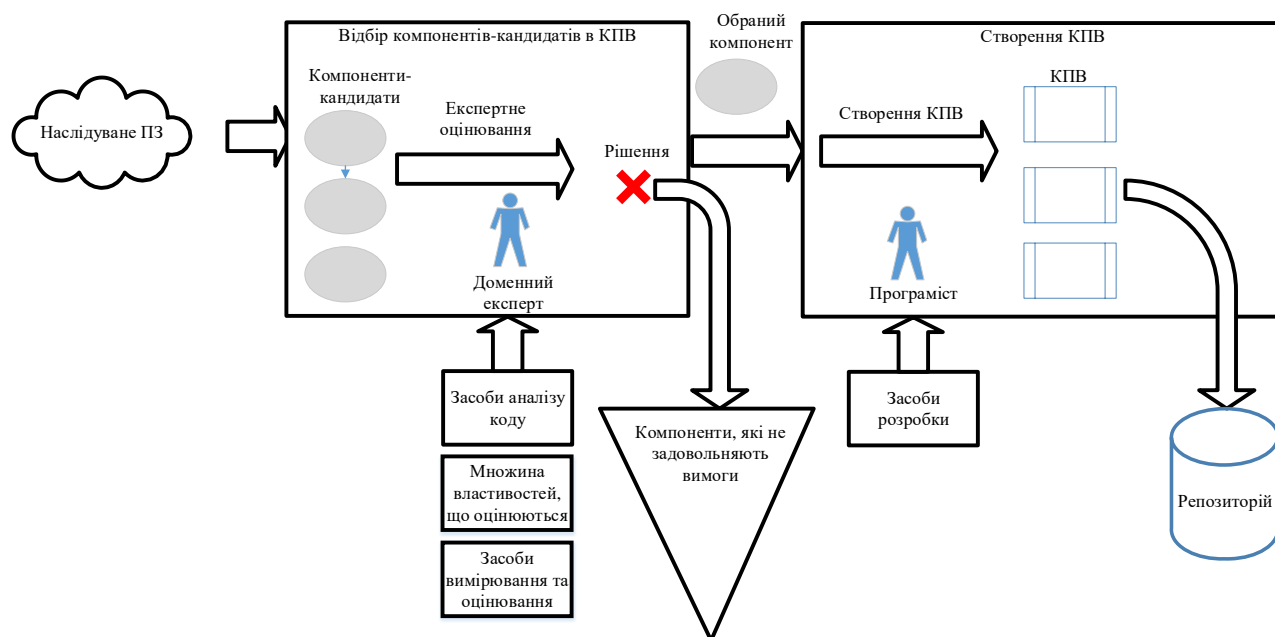


Рис. 2.3. Алгоритм експертної оцінки

Найбільш складним у даному методі є процес проведення експертного оцінювання, що зумовлено не тільки необхідністю безпосередньої оцінки атрибутів компоненту, але й врахуванням вимог до несистематичного повторного використання з необхідністю дослідження програмного коду компонента. Для одержання і забезпечення об'єктивних оцінок експерт повинен використовувати спеціалізовані, складні у використанні інструменти [6].

Процес створення ПВК не настільки трудомісткий у порівнянні з першим, проте вимагає в експерта значного досвіду і знань для зниження похибки реалізації. Особливістю процесів, наведених на рис. 2.3, є те, що від експерта вимагається володіння не тільки знання програмування, але й предметної області.

Враховуючи складність і трудомісткість процесу експертного оцінювання, пропонується автоматизувати процес аналізу програмного коду при оцінюванні атрибутів ПЗ щодо можливості повторного використання компонента і таким чином забезпечити оптимізацію цього процесу. Тоді схема експертного оцінювання матиме вигляд, як показано на рис. 2.4.

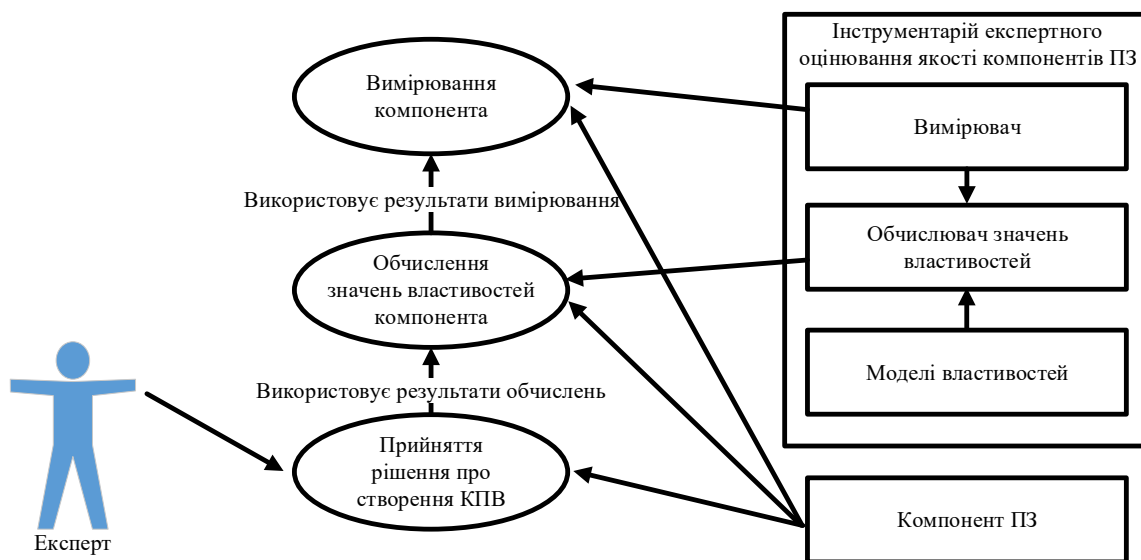


Рис. 2.4. Автоматизація процесів експертного оцінювання ПВК

Автоматизація експертного оцінювання передбачає виконання трьох основних процесів:

- вимірювання;
- обчислення метрик атрибутів;
- прийняття рішення щодо реалізації ПВК.

Вимірювання проводиться з використанням спеціального інструменту, що може аналізувати програмний код і повертати значення прямих кількісних мір компонента.

Обчислення метрик атрибутів компонента має забезпечуватись інструментом–обчислювачем, який дозволяє одержувати кількісні значення атрибутів отримані в процесі вимірювання і моделей атрибутів, які показують залежність між прямими і непрямими атрибутами.

Для того, щоб процес оцінювання атрибутів відповідав критеріям ефективності необхідно провести формалізацію моделі атрибутів (властивостей) та забезпечити її адекватність.

Формалізацію і забезпечення адекватності моделей запропоновано виконувати виходячи з досвіду експертів та шляхом уточнення самих моделей. При цьому потрібно забезпечити виконання наступних процесів при дослідженні компонентів:

- оцінювання;
- вимірювання;
- уточнення моделей.

Уточнення моделей виконується з використанням засобів, які представлено на рис.2.5.

Оцінювання атрибутів компонентів має забезпечувати досвідчений експерт з використанням інструменту, який дозволяє переглядати програмний код компоненту, вносити і зберігати результати експертизи. Для отримання експертних оцінок можна використовувати методи ранжування, попарного порівняння та прямої оцінки [4].

Враховуючи те, потенційні ПВК аналізуються експертом по одному, то застосування методів ранжування і попарного порівняння у даному випадку використовувати недоцільно, тому у роботі запропоновано виконувати пряме оцінювання атрибутів з використанням числових або порядкових шкал.

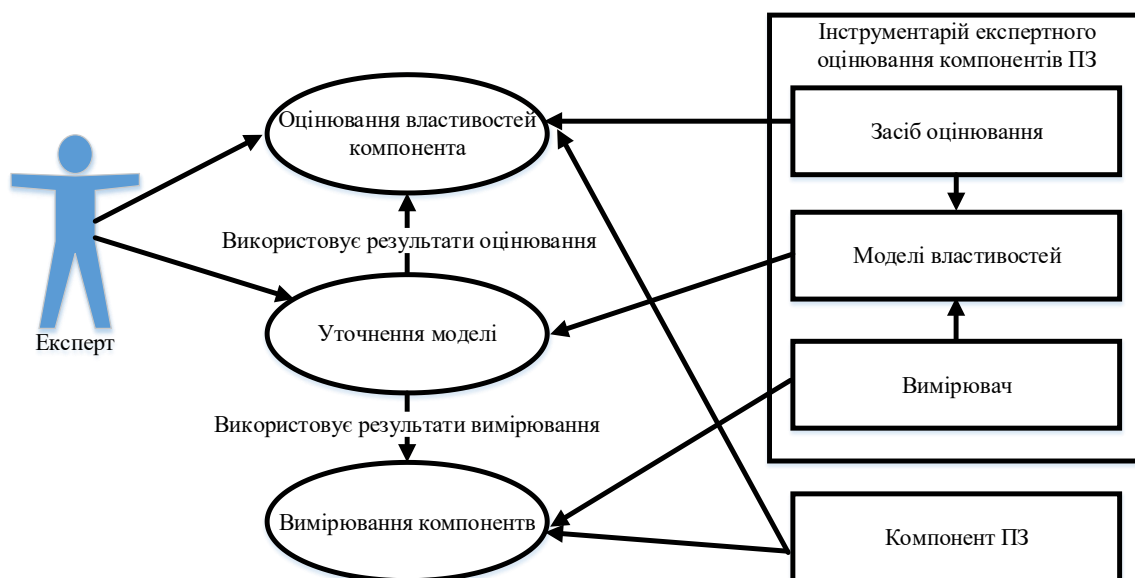


Рис. 2.5. Інструменти уточнення моделей атрибутів

Отже, при імплементації запропонованого методу потрібно забезпечити розв'язок таких задач:

- визначити сукупність атрибутів, які є важливими в контексті повторного використання;
- побудувати моделі для опису атрибутів, що дозволять забезпечити автоматизацію запропонованого методу на основі вимірювань;
- побудувати архітектуру інструментів, які забезпечать імплементацію запропонованого методу.

2.2. Обґрунтування моделі представлення атрибутів компонентів повторного використання

У кваліфікаційній роботі для вибору потенційних компонентів у ПВК пропонується використати алгоритм на основі PVB – підходу, а для побудови моделі атрибутів – 3С-модель [8]. Характерною ознакою 3С-моделі для представлення ПВК є те, що вона враховує три характеристики при формуванні онтології атрибутів – «Conception», «Construction» та «Context» (рис. 2.6.).

«Conception» дає змогу представити функціональність компонента, а експерт

у предметній області сприймає її через набір функцій, які дозволяють моделювати конкретні об'єкти чи процеси домену.

Функціональність, з точки зору повторного використання, характеризується шириною застосування і корисністю, що є найбільш важливими у контексті «Conception». Корисність інтерпретується як рівень задоволення вимог, які сформульовані користувачами до ПВК. Ширина застосування (спільність) показує масштабованість потенційного використання ПВК.

«Context» інтерпретує вимоги до середовища застосування ПВК. У запропонованій до використання моделі, виділяють семантичний і синтаксичний контексти. Семантичний контекст створюється розробником у вигляді логіки взаємодії створюваного компонента з існуючими, а синтаксичний – створюється як інтерфейс ПВК. «Context» описується такими важливими атрибутами як: завершеність інтерфейсу, синтаксична і семантична незалежність. Завершеність інтерфейсу описує наскільки інтерфейс компонента забезпечує повноту функціональності при його застосуванні. Семантична незалежність проявляється у здатності компонента генерувати сталий результат при кожному його запуску, незалежно від середовища чи місця його розташування. Синтаксична незалежність інтерпретує можливість компонента застосовуватися без зміни його контексту. «Construction» проявляється в одній з реалізацій «Conception». У випадку повторного використання для «Construction» важливими є такі атрибути:

- достатність – проявляється у здатності конструкції відобразити концепцію таким чином, щоб було забезпечити повну передачу функціональності;
- завершеність конструкції – здатність конструкції відобразити усю потужність функціональності, що наявна у концепції;
- примітивність – характеризує те, що у конструкцію можуть бути включені лише елементарні операції, наявні у концепції;
- модульність – характеризує модульну структуру компоненту для забезпечення його простоти розуміння та внесення змін;

- структурованість – описує застосування наявних у компоненті структурних операторів, їх складності, що має вплив на здатність простого внесення змін та розуміння розробником;
- трасування – проявляється у здатності отримувати мітку переходу від одного етапу ЖЦ до іншого;
- надійність – проявляється у стійкості поведінки компонента за різних умов його експлуатації;
- ефективність – характеризує постійність забезпечення ефективного виконання компонента у відповідності до визначеної сукупності вимог;
- переносимість – проявляється у коректності функціонування за умови різних середовищ виконання;
- класифікація – здатність, що проявляється в ефективному розміщенні і пошуці компонента згідно заданої рубрикації;
- компактність – проявляється у компактності при зберіганні компонента;
- зрозумілість – характеризує інтерпретацію компонента при його використанні під час створення нового ПЗ;
- простота модифікації – характеризує те, наскільки просто вносити зміни у конструктивні елементи компонента;
- коректність – проявляється у здатності не містити як синтаксичних, так і семантичних помилок у компоненті;
- тестопридатність – здатність компонента бути протестованим визначеним набором правил.

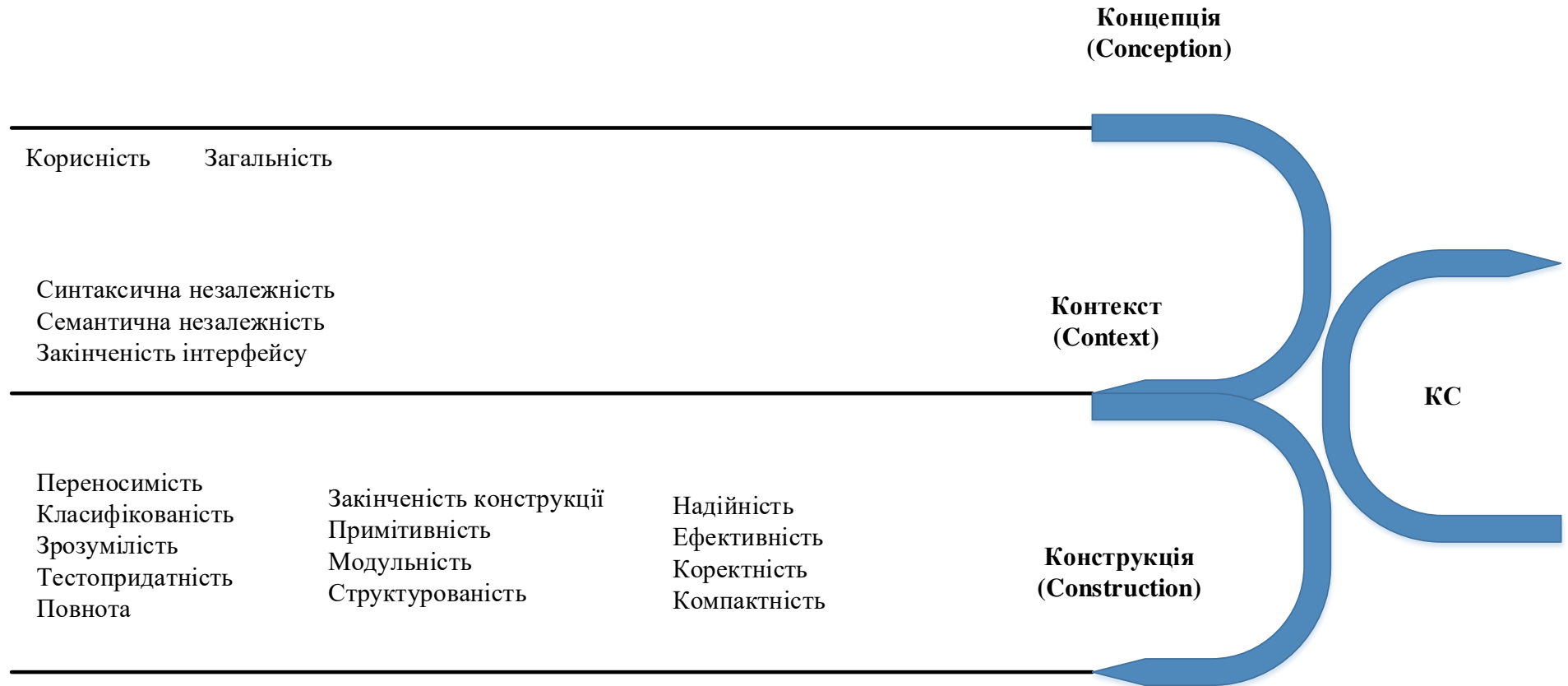


Рис. 2.6. Модель атрибутів ПКВ

2.3. Формалізація моделі атрибутів повторно використовуваних компонентів

Загалом, використовуючи модель атрибутів, процес формування компонентів повторного використання можна інтерпретувати як діяльність, щодо трансформації сукупності атрибутів P_c існуючого ПЗ у множину атрибутів ПКВ P_r у відповідності до вимог R :

$$P_r = A(P_c, R) , \quad (2.1)$$

де P_r – значення атрибутів КПВ;

A – операція трансформації;

P_c – сукупність значень атрибутів компонента існуючого ПЗ;

R – набір вимог до атрибутів КПВ.

При систематичному підході, КПВ реалізується лише з врахуванням вимог і не береться до уваги вихідний код компонента. Далі цей компонент записують у базу даних і він може використовуватися як «чорна скринька». Тому такий процес описується по аналогії до традиційної розробки ПЗ ($P_c = 0$), а вираз 2.1 набуває такого вигляду:

$$P_r = A(R) , \quad (2.2)$$

де P_r – набір потрібних атрибутів КПВ;

У випадку не систематичного повторного використання вихідними даними для реалізації компонента є частини існуючого ПЗ, які описуються певними атрибутами, тому процес створення КПВ доцільно розглядати як зміну компонентів існуючого ПЗ з адаптацією до вимог R , але без їхнього запису у базу даних.

На трудомісткість трансформації існуючих компонентів ПЗ у ПВК впливають відхилення атрибутів набору P_c та набору P_{rc} , заданих вимогами R , що відображається сукупністю робіт для забезпечення їх ідентичності. У даному випадку під відхиленням між наборами розуміють різницю атрибутів множин P_c і P_{rc} (наявність у множинах однакової кількості атрибутів) або ж різницю між значеннями атрибутів. Отже, затрати на реалізацію ПВК визначаються схожістю наборів P_c компонентів існуючого ПЗ та набору P_{rc} і як результат представляються витратами на оператор A перетворення одних атрибутів в інші. В якості критерію вибору потенційного компонента у ПВК є вираз, який задається умовою:

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

де $C(A(P_c, R))$ – затрати на формування ПВК з існуючого ПЗ;

$C(A(R))$ – затрати на реалізацію ПВК з самого початку.

Отже, у процесі оцінювання атрибутів компонентів існуючого ПЗ основна задача експерта полягає у визначенні для кожного елементу множини атрибутів P_c схожості з елементами ПВК P_{rc} у відповідності до вимог R . Доцільність формування ПВК з потенційного кандидата описується експертом через процедури оцінювання витрат, що потрібні при модифікації кожного атрибута потенційного компонента і оцінку загальних витрат:

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

де $C_i(A_i(P_{ci}, R_i))$ – витрати на модифікацію i -ого атрибута P_{ci} з набору P_c ;

n – потужність набору P_c .

Підсумовуючи наведені вище вирази, логічно зробити висновок у вигляді такого твердження: важливість атрибуту є тим вищою, чим більші витрати на його модифікацію у компоненті ПЗ на етапі формування ПВК.

Для того, щоб застосовувати компонент повторного використання у новому ПЗ, необхідно виконати його пошук у репозиторії, а після цього провести адаптацію, верифікацію та інтеграцію.

Зрозуміло, що витрати на виконання таких процесів мали б бути меншими, у випадку коли більшість атрибутів ПВК відповідатимуть вимогам процесів. Отже, для встановлення придатності ПВК при повторному використанні оцінюється «повнотою» набору атрибутів P_u , що є важливими з точки зору ПВК і були врахованими при його реалізації, а також величиною рівня відхилення щодо задоволення умови:

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

де P'_{rc} – набір атрибутів конкретного ПВК;

P_{rc} – набір атрибутів ПВК, що задовольняє вимогам процесів його застосування.

У випадку невідповідності умові, зростають додаткові витрати на забезпечення виконання процесів застосування ПВК у новому ПЗ, що пов'язано з реалізацією операцій щодо модифікації компонента:

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

Аналізуючи вищенаведені аргументи та вирази щодо витрат, необхідних при реалізації процесів представлення та визначення атрибутів компонентів існуючого ПЗ, важливих з точки зору повторного використання, справедливим

буде твердження: важливість атрибуту стає більшою із зростанням витрат, які потрібні для модифікації ПВК.

Виходячи з цього, для визначення атрибутів компонентів як на етапі їх формування, так і на стадії використання, можна проводити їх оцінювання з погляду впливу щодо вартості імплементації відповідних процесів. Враховуючи таке твердження, набір атрибутів P_{rcp} можна визначити таким чином:

- для кожного атрибуту з набору P_{cp} потрібно провести оцінювання його впливу на затрати реалізації процесів формування і застосування ПВК;
- створити набір атрибутів P_{rc} на основі вибору із сукупності P_{cp} таких атрибутів, що є важливими у контексті формування і використання ПВК.

На етапі ідентифікації процесу формування ПВК, атрибути спільності і корисності є неважливими, тому що вони не відображаються на витратах ідентифікації і вибору компонентів. Затрати на виконання процесу ідентифікації знижується у зв'язку із спрощенням виявлення лімітів і розуміння складу компонента. Існування атрибуту завершеність контексту полегшує виділення компонента шляхом забезпечення його чіткого означення.

Жоден з атрибутів потенційного компонента не є суттєвим на етапі кваліфікації, тому що не має відношення до вартості наявних процесів. Атрибути ПВК відбиваються на вартості етапу кваліфікації так, як показано у табл. 2.1.

Атрибути якості компонента наслідуваного ПЗ	Процеси									
	Створення КПВ						Використання КПВ в новому ПЗ			
	Ідентифікація		Кваліфікація							
	Визначення компонентів	Вибір компонентів	Специфікація	Генерація тестових	Класифікація	Упаковка та збереження	Пошук	Модифікація	Верифікація	Інтеграція
В аспектах 3С-моделі										
Трасування			+	+	+			+	+	
Надійність										
Ефективність										
Переносимість								+		+
Класифікованість					+		+			
Компактність						+				+
Зрозумілість	+	+	+	+	+		+	+	+	+
Простота модифікації								+		
Коректність				+				+	+	+
Тестопридатність				+					+	
Контекст										
Семантична незалежність										+
Синтаксична незалежність										+
Закінченість інтерфейсу	+		+	+					+	+

2.4. Моделі властивостей

Усі атрибути, які є важливими з позиції повторного використання, не підлягають прямому вимірюванню, через що у відповідності до підходу необхідно провести деталізацію вихідних атрибутів компонентів та обчислити їх для забезпечення асоціації з множиною значень. Застосування системного підходу передбачає аналіз ієрархічного багаторівневого дерева об'єкта, що виражає його уявлення у вигляді набору елементів.

Величини, які можна використовувати у процесі вимірювання та/або оцінювання атрибутів ПЗ, залежать від двох важливих факторів [6]:

- адекватність метрики заданому атрибуту ПЗ;
- простота отримання міри метрики.

Адекватність метрики заданому атрибуту зумовлена вимогою щодо повноти та адекватності кількісного вираження атрибуту на основі обраної метрики.

Простота отримання міри обумовлена необхідністю одержувати її значення для визначених об'єктів, і пов'язана з типом метрики та способом отримання відповідних значень. Вимірювання може застосовуватися лише для метрик адитивного типу, а це в свою чергу обмежує їх вибір у моделях, що оперують мірами.

Характерною особливістю системного підходу є можливість виявлення атрибутів, які представляють не вимірювані атрибути. Для прикладу, характеристика «зручність читання програмного коду» може інтерпретувати такі атрибути, як забезпечення структурованості коду, стиль програмування, наявність та зрозумілість коментарів. Дані атрибути також не можуть бути вимірними і в подальшому відображаються на наступний рівень. Атрибут, що характеризує наявність і зрозумілість коментарів може відображатися такими атрибутами як кількість закоментованих стрічок, їх щільність, існування заголовків для методів і т.п. Застосування такого підходу при деталізації атрибутів ПВК носить дедуктивний характер і дозволяє побудувати ієрархічні структури пов'язаних

атрибутив, які деталізуються до тих пір, поки на найнижчому рівні не буде забезпечено вимірюваність атрибутів [12].

При використанні моделі у вигляді ієрархічного дерева та враховуючи особливості системного аналізу характерним є те, що у корінь моделі поміщається характеристика, яка інтерпретує цільові атрибути, однак є непридатною для прямих вимірювань, а на нижчих рівнях можуть бути наявними вимірювані величини. Проте атрибути нижчого рівня інтерпретують семантику у вузькому сенсі. Ефективність моделі атрибутів залежить від адекватності побудованих зв'язків між елементами, що розташовуються на суміжних рівнях ієрархічної структури. Перевірка на адекватність моделі атрибутів виконується шляхом перевірки відношень на основі даних, одержаних експериментальним шляхом.

Структура моделі, одержана у результаті застосування системного аналізу представляє собою модель атрибутів P_{ij} та відповідних метрик M_{ij} , яка показана на рис.2.7.

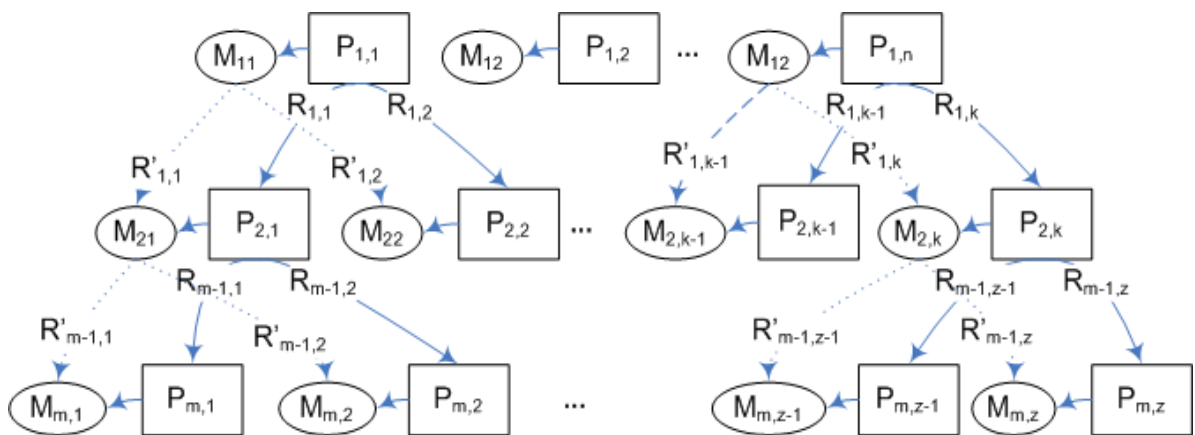


Рис. 2.7. Модель атрибутів

При формуванні елементів моделі потрібно визначити їх семантику, шкалу і процедури порівняння. На рис. 2.8 представлено модель, що передбачає наявність трьох рівнів: цільовий, проміжний і вимірювання. Проміжний рівень виконує функції забезпечення здатності допоміжного структурування системи атрибутів і метрик.

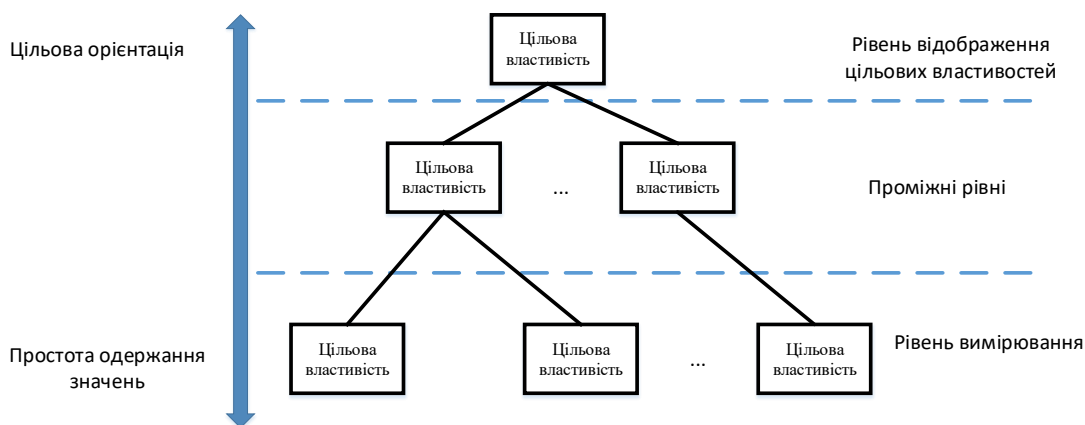


Рис. 2.8. Ієрархічна модель атрибутів

З наведених вище структур і описів можна зробити висновок про те, що ієрархічна модель атрибутів дає змогу їх представляти у вигляді важливих з точки зору повторного використання і використанням у запропонованому методі. При цьому потрібно для кожного атрибуту з набору $\{e_j^l | i = 1..w\}$ створити сукупність з n пов'язаних метрик за $l - 1$ рівнями $\{m | i = 1 \dots n, k = 2 \dots l\}$.

Реалізації методу передбачає побудову дворівневих моделей атрибутів, які є визначальними щодо повторного використання. На першому рівні розташовано атрибути, які необхідно оцінити, а на другому – метрики, кількісне значення яких вимірюються, як показано на рис.2.9.

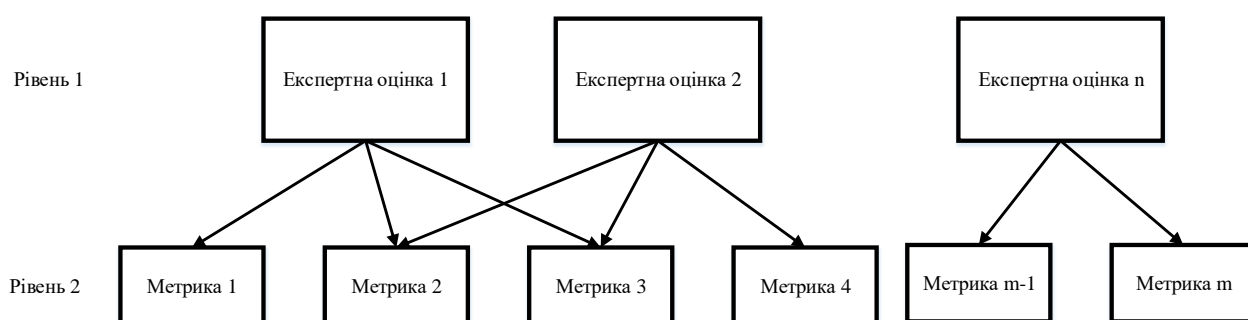


Рис. 2.9. Модель атрибута

При побудові моделей атрибутів у вигляді залежностей між атрибутами і метриками запропоновано застосувати метод «Мета-Питання-Метрика». Як було зазначено раніше, модель представляє собою дворівневу структуру з атрибутами і

метриками. Метрики можуть обиратися із довідників і які створені на основі аналізу їх придатності до того, щоб бути вимірними (рис.2.10) [20].

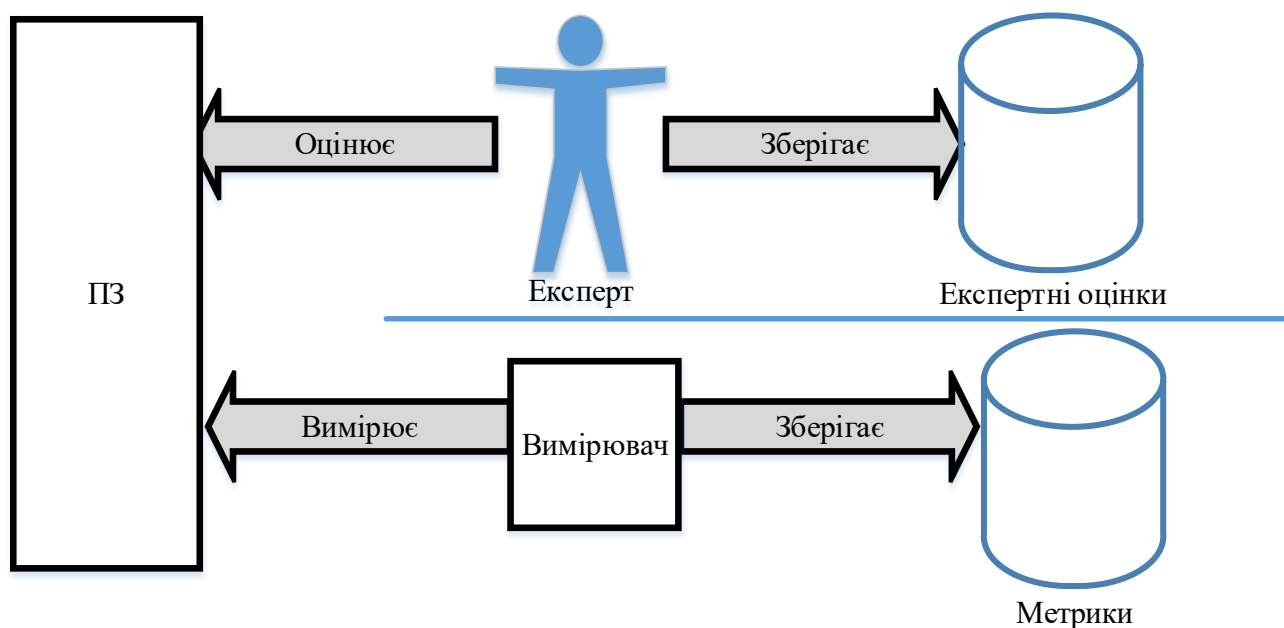


Рис. 2.10. Процеси використання дворівневої моделі на практиці

Методом «Ціль-Питання-Метрика» передбачено виконання таких кроків [17]:

- визначення цілей, які необхідно досягнути при реалізації процесу оцінювання атрибутів;
- створення питань для кожної визначеної цілі для одержання відповідей щодо досягнення мети;
- визначення метрик, які будуть використовуватися при аналізі відповідей на запитання попереднього кроку.

В якості прикладу виконаємо застосування методології «Ціль-Питання-Метрика» при побудові моделі атрибутів «зрозумілість». Цей атрибут відноситься до одних з найважливіших, оскільки впливає майже на усі процеси, характерні для повторного використання.

Крок 1. Ціль полягає у наступному: «Провести оцінювання зрозумілості та читабельності програмного коду компонента існуючого ПЗ».

Крок 2. Питання. Для встановлення факту досягнення поставленої цілі розроблено сукупність питань, які представлені у колонці 1 в табл. 2.2.

Крок 3. Метрики. Кожному запитанню поставлено у відповідність елементи з множини метрик, які відображено у колонці 2 в табл. 2.2.

Таблиця 2.2

Ціль, питання і метрики для властивості «Зрозумілість»

Ціль	Питання	Метрика		
		Семантика	Назва	№ метрики
Оцінити зрозумілість компонента-кандидата в КППВ	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 модулі	Ave NCSS	14
		Загальні зусилля за Холстедом	Total Hal's E	21
		Максимальна структурна вкладеність	Max Nesting	27

Результат застосування наведених вище кроків у графічному вигляді представлений на рис. 2.11.



Рис. 2.11. Метрики атрибуту «Зрозумілість»

По аналогії можна побудувати дворівневі моделі для не менш важливих атрибутів, зокрема, «структурованість» та «модульність», із застосуванням методу «Ціль-питання-метрика».

2.5. Висновки до розділу

У даному розділі одержано наступні результати:

1. Запропоновано метод оцінювання атрибутів ПВК, що дає змогу забезпечити автоматизацію процесу одержання метрик атрибутів на основі розрахунку результатів вимірювань та із застосуванням моделей ознак (атрибутів).
2. Визначено набір властивостей потенційних компонентів програмних складових у розумних комп'ютерних системах, що є важливими при імплементації процесів повторного використання, на основі створення вихідного набору атрибутів моделі «Concept-Construction-Context» і відбору лише тих, які безпосередньо впливають на затрати при їх повторному використанні.

3. Запропоновано процедуру створення моделей атрибутів, що базується на принципах підходу системного аналізу та методу «Ціль-Питання-Метрика», що дало змогу забезпечити ефективність експертного оцінювання характеристик КПВ.

РОЗДІЛ 3

РОЗРОБКА СИСТЕМИ АВТОМАТИЗАЦІЇ ОЦІНЮВАННЯ АТРИБУТІВ ПОВТОРНО ВИКОРИСТОВУВАНИХ КОМПОНЕНТІВ

3.1. Проектування архітектури системи оцінювання атрибутів ПКВ

Автоматизацію процесів запропонованого у роботі методу необхідно забезпечити шляхом побудови та імплементації відповідних програмних компонентів, які повинні розв'язувати дві основні задачі:

- перевірка на адекватність моделей атрибутів і формування залежності між атрибутами потенційних компонентів у ПКВ і метриками;
- автоматизація технології експертного оцінювання атрибутів ПЗ.

Для забезпечення розв'язку задачі перевірки на адекватність моделі та формування відношень між атрибутами необхідно дотримуватися правил розробленої методики, що реалізується наступним функціоналом засобів:

- підтримка процесу експертної оцінки атрибутів компонентів із подальшим їх записом у базу даних;
- отримання кількісних значень метрик на основі аналізу та вимірювання програмного коду та зберігання їх у базу даних;
- побудова залежності між одержаними експертними оцінками і метриками.

Розв'язок задачі щодо автоматизації технології експертного оцінювання реалізується двома процесами, які представлені на рис.2.2. Вони включають оцінювання атрибутів компонента, що супроводжується прийняттям рішення щодо реалізації або не реалізації КПВ. Алгоритм виконання цих двох процесів представлено на рис. 3.1. Реалізацію процесів забезпечують наступні функції:

- вимірювання;
- розрахунок значень атрибутів, що базуються на результатах вимірювань.

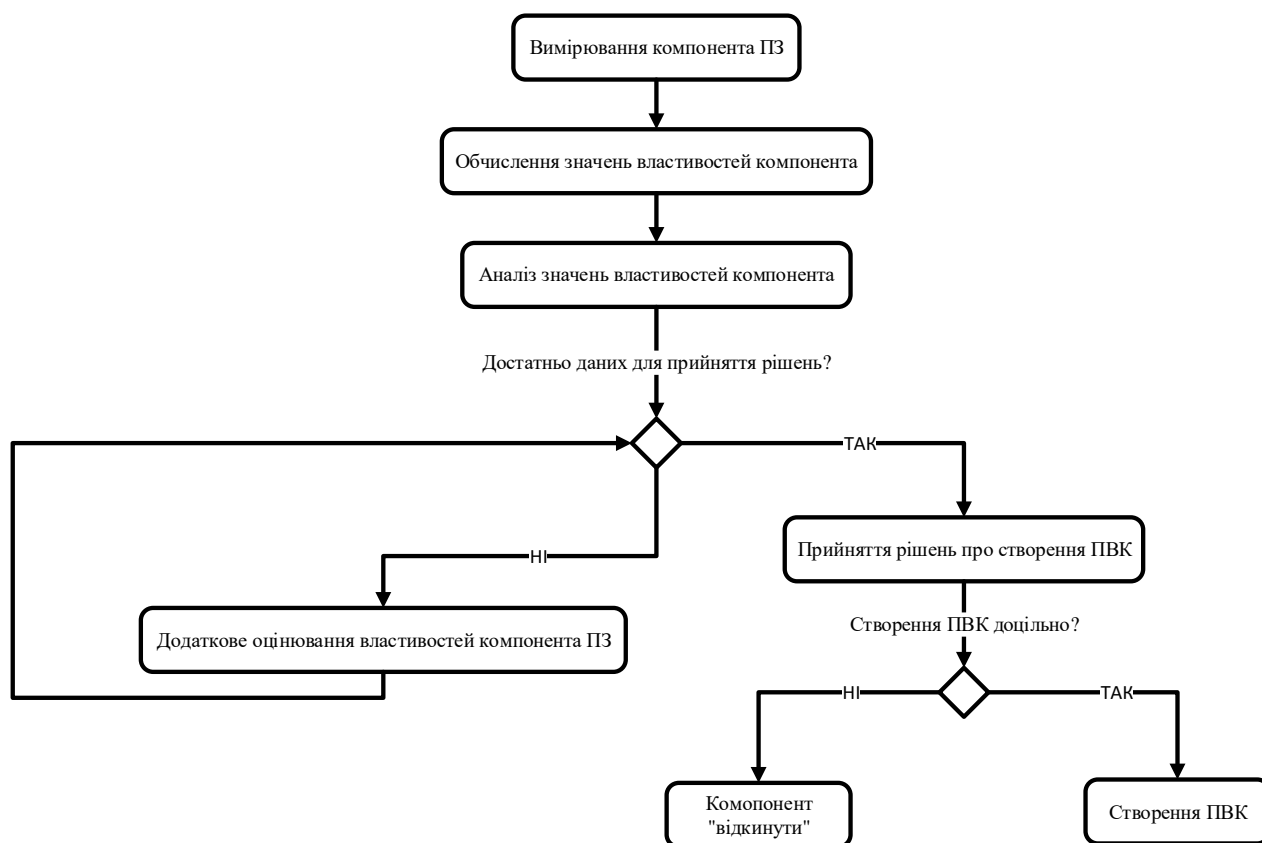


Рис. 3.1. Алгоритм оцінювання потенційних ПВК

Архітектуру системи для забезпечення реалізації перелічених вище функцій спроектовано та проілюстровано на рис. 3.2. До її складу входять:

- модуль розрахунку значень атрибутів – виконує обчислення значень атрибутів на основі результатів вимірювання;
- модулі для аналізу даних – виконують функції щодо деталізації залежності між оцінками і метриками з використанням інформації, одержаної у результаті оцінювання та вимірювання;
- користувацький інтерфейс – забезпечує діалог між системою та експертом;
- модуль управління – призначений для забезпечення узгодженості функціонування усіх програмних компонентів системи;
- вимірювач – забезпечує одержання значень метрик на основі аналізу та проведення вимірювання програмного коду програм;

- база даних – зберігає сукупність оцінок і метрик, кількісних показників проведених вимірювань і додаткових службових даних;
- база знань – зберігає знання щодо існуючих залежностей між атрибутами і метриками.

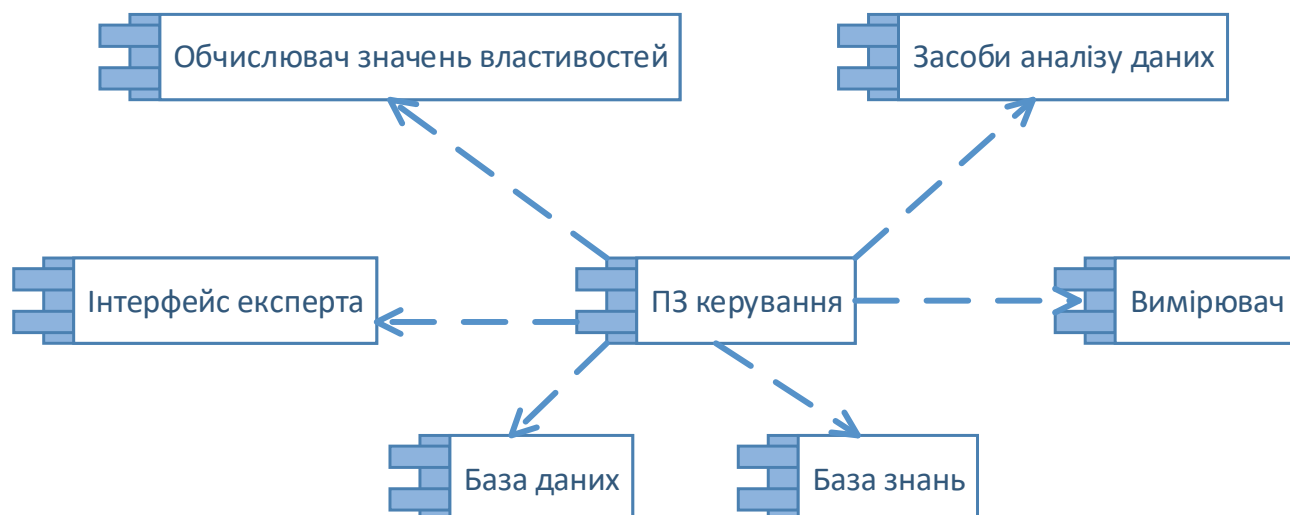


Рис. 3.2. Архітектура системи оцінювання якості повторно використовуваних компонентів

При використанні системи із запропонованою архітектурою передбачається два режими її функціонування:

- акумуляція та аналіз даних – дозволяє проводити вимірювання та експертне оцінювання, акумулювати інформацію, забезпечувати аналіз даних щодо виявлення залежності між атрибутами і метриками;
- оцінювання – одержання кількісних оцінок компонентів з використанням метрик, отриманих у результаті вимірювання, і залежностей між атрибутами та метриками.

Режим акумуляції та аналізу даних передбачає проведення експертного оцінювання та виміру існуючих компонентів ПЗ, зберігання даних у базу даних, що в подальшому будуть проаналізовані з метою побудови залежності між кількісними значеннями оцінок і метриками. Для введення даних оцінок експертів використовується користувацький інтерфейс експерта, який також забезпечує

керування самим процесом оцінювання. Застосування вимірювача метрик програмного коду забезпечує одержання кількісних мір в автоматичному режимі. У базу даних записують усі одержані оцінки та метрики, а залежності, виявлені в процесі аналізу з відповідними результатами поміщаються у базу знань.

У випадку функціонування системи у режимі оцінювання, вимірювачем виконуються аналіз програмного коду, а одержані результати передаються у модуль проведення розрахунків значень атрибутів. Цей модуль за допомогою виявленої залежності між оцінками і метриками виконує обчислення кількісної міри атрибутів, що відображається в інтерфейсі експерта.

3.2. Опис компонентів архітектури

Як було зазначено раніше, реалізацію архітектури системи при оцінюванні ПВК забезпечують модулі, які показані на рис. 3.2.

До модулів, які дають можливість реалізувати процес експертного оцінювання, належать: користувацький інтерфейс, модулі управління та обчислювач міри атрибутів.

Функціональність користувацького інтерфейсу експерта реалізує такі функції:

- попередній перегляд програмного коду компонентів існуючого ПЗ;
- ввід та представлення оцінок експерта.

Окрім цього, інтерфейс експерта дозволяє ввід інших властивостей при оцінюванні компонентів ПЗ. Текстовий браузер забезпечує попередній перегляд програмного коду модулів ПЗ. Для введення та представлення значень, які використовує експерт, реалізовано за допомогою прикладного ПЗ, що дозволяє:

- відображати набір оцінок за відповідними атрибутами;
- відображати перелік доступних значень конкретної оцінки;
- вибирати значення оцінки з переліку з подальшим їх зберіганням у БД;
- введення додаткових властивостей компонентів з можливістю їхнього зберігання у БД;

- візуальне представлення значень метрик, одержаних вимірювачем;
- візуальне представлення значень оцінок, розрахованих за допомогою метрик.

Зовнішній вигляд інтерфейсу експерта наведений на рис.3.3.

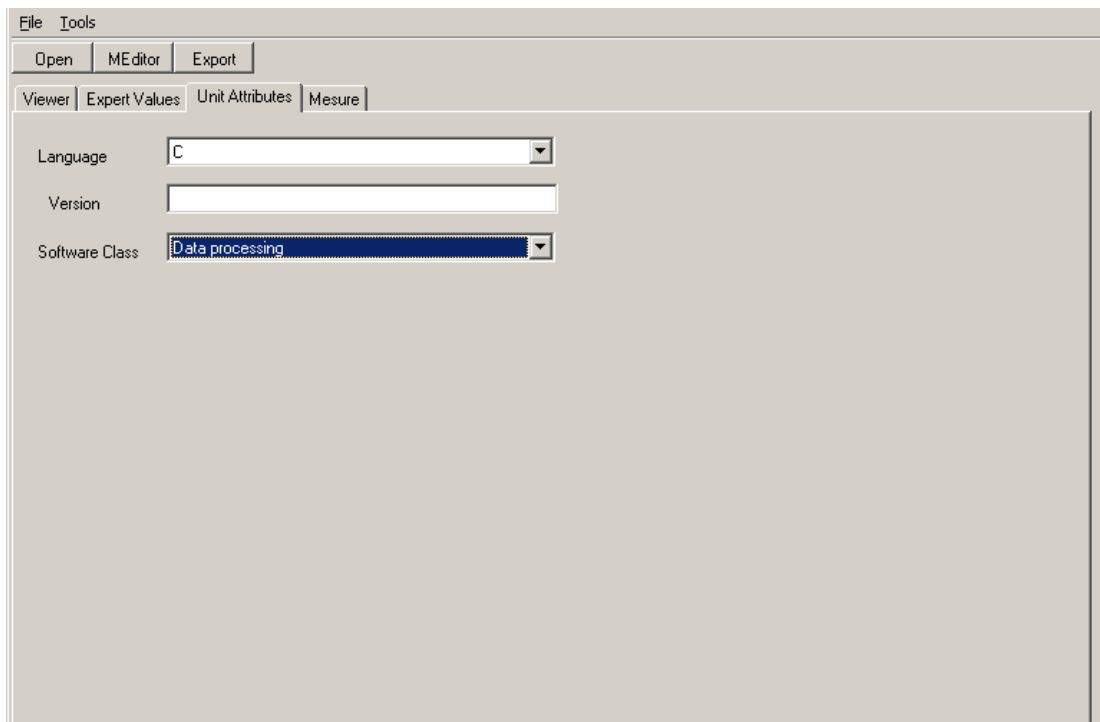


Рис. 3.3. Інтерфейс експерта

Модулі управління роботою системи оцінювання атрибутів забезпечують коректність функціонування усіх елементів системи, які визначені архітектурою та і реалізують такі функції:

- авторизація – дозволяє проводити перевірку ідентифікаційних та авторизаційних даних експертів на початку роботи з системою;
- керування іншими модулями – забезпечує виконання заданої логіки функціонування модулів з врахуванням особливостей процедури авторизації, обрання компонентів, старту вимірювача і т.п.;
- керування вимірювачем – виконує старт вимірювача, передаючи посилання на програмний код відповідного компонента та зберігає значення до бази даних;

– підтримка модулів – дозволяє вводити і редагувати потрібну для коректного функціонування інформацію.

Алгоритм виконання функцій вимірювачем у режимі акумуляції даних проілюстровано на рис. 3.4., а в режимі оцінювання – показано на рис. 3.5.

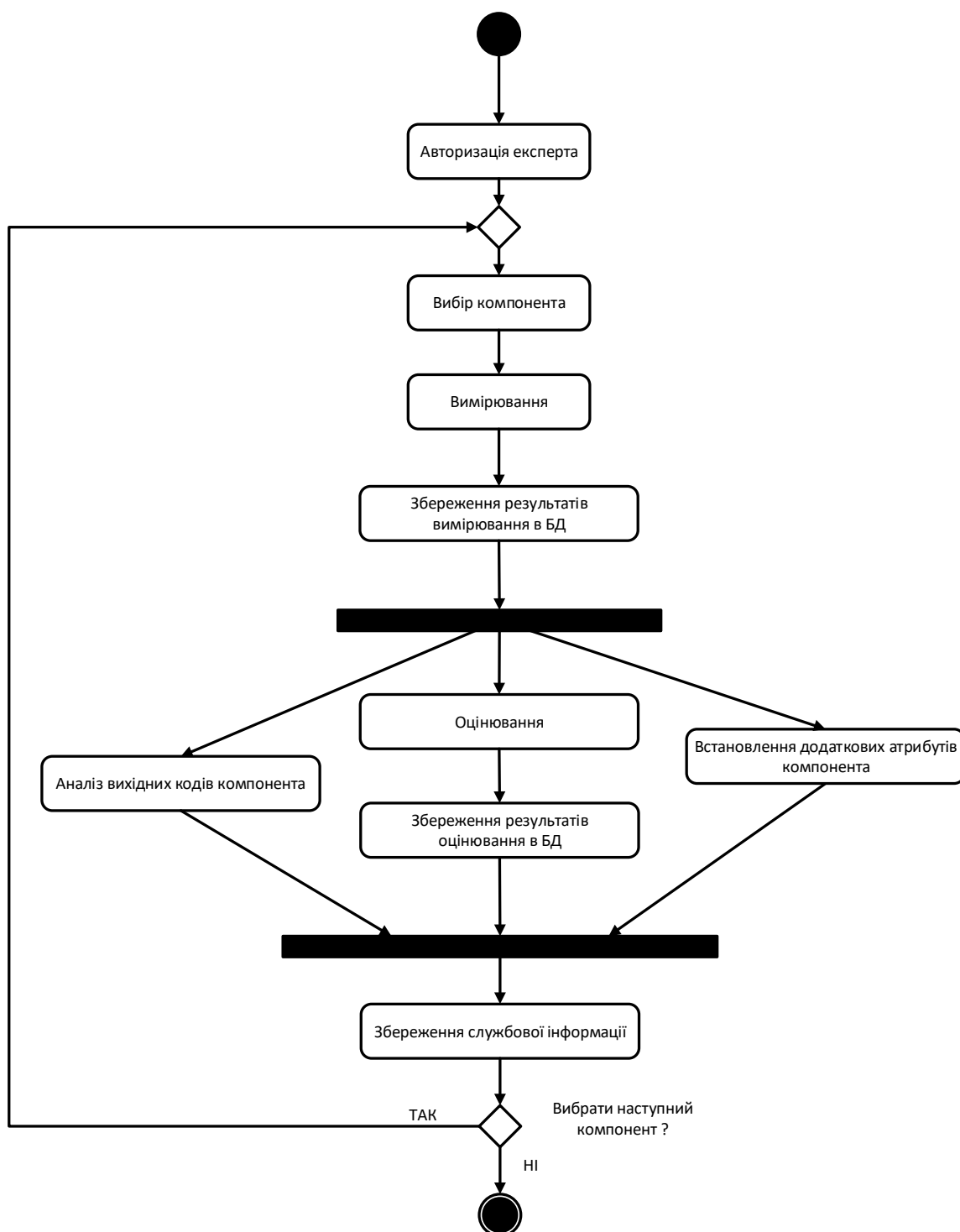


Рис. 3.4. Алгоритм функціонування системи в режимі акумуляції даних

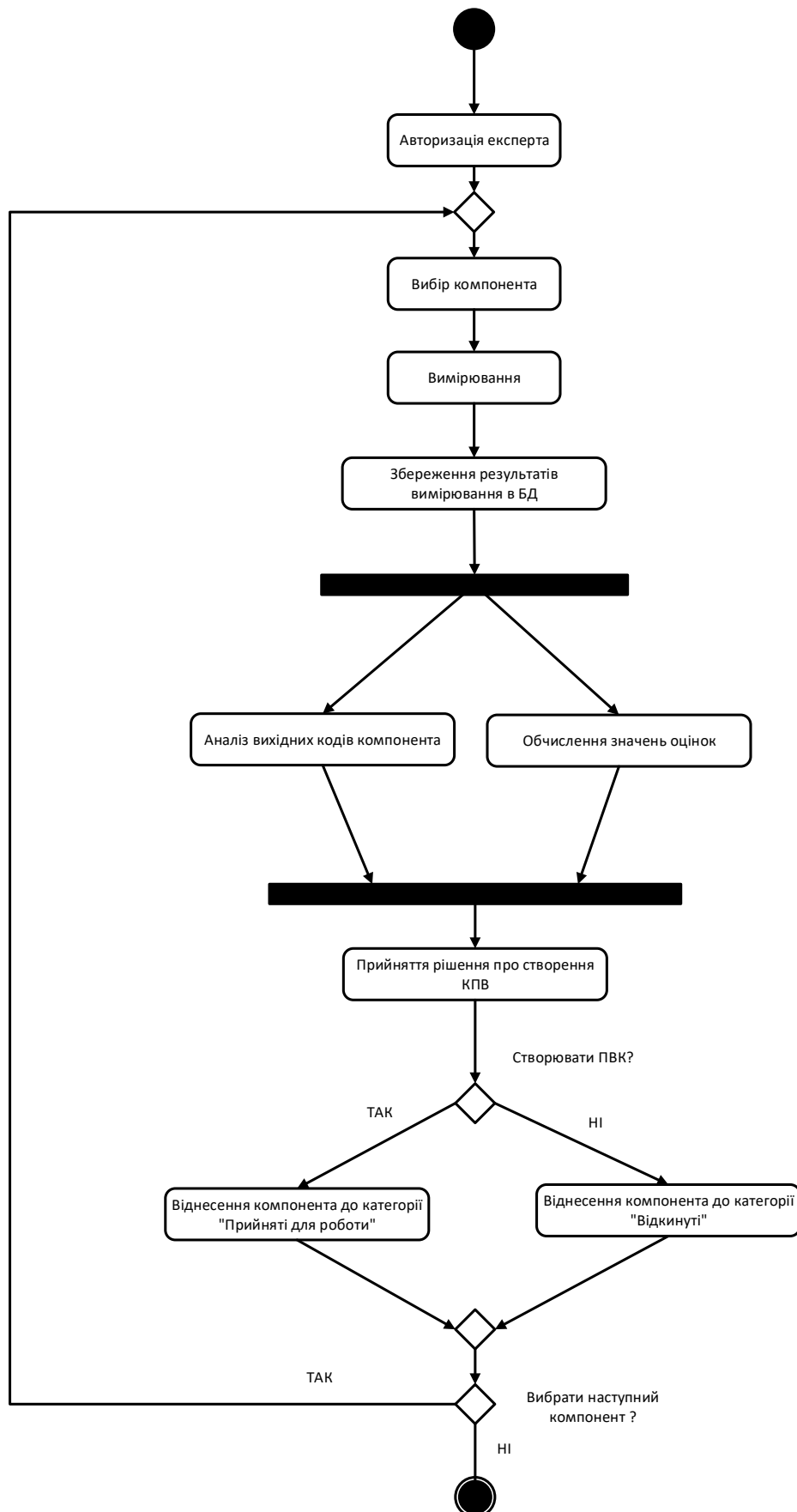


Рис. 3.5. Алгоритм функціонування системи у режимі оцінювання

Модуль обчислення може бути спроектований за допомогою різних підходів, однак у кваліфікаційній роботі запропоновано використання нейронних мереж [6], що забезпечують зручність автоматизації виявлення і зберігання залежності між вхідними і вихідними шарами при навчанні на дата сетах. Сьогодні існує багато видів нейромереж з різною архітектурою та принципом навчання, що дають змогу виконати поставлене завдання для встановлення залежності між метриками та оцінками. Вибір тієї чи іншої нейронної мережі зумовлений фактором експериментального і практичного застосування. У даному випадку при реалізації модуля проведення розрахунків атрибутів обрано адаптивну нейромережу з нечітким виводом.

Така адаптивна мережа є способом організації гібридних нечітких нейромереж прямого поширення. Структура нейромережі ізоморфна до структури нечіткої бази знань. В основі таких нейромереж лежать диференційовані трикутні норми та функції приналежності, які дають змогу забезпечити гнучкість налаштування параметрів для швидких алгоритмів навчання, які використовують метод зворотного поширення помилки. Враховуючи те, що набір значень при експертному оцінюванні може формувати значення на основі заданої множини слів, то вони інтерпретуються як лінгвістичні змінні [18].

3.2.1. Сховища зберігання даних

Основними компонентами, що забезпечують збереження та можливість проведення аналізу даних є БД, БЗ та інструменти аналізу.

У кваліфікаційній роботі база даних представляється об'єктами, які продемонстровано на рис.3.6:

- експерт – забезпечує оцінювання атрибутів компонента;
- оцінки експерта – нотації оцінок, які можуть застосовувати експерти при оцінюванні компонентів;
- значення оцінок – попередньо встановлені значення, які мають відношення до результатів експертизи проведеної під час оцінювання;

- компоненти ПЗ – представляють собою модулі, які необхідно оцінити щодо можливості формування ПВК;
- експеримент – відображає виконані процедури щодо оцінки компонентів;
- метрика – опис процедури і шляхів вимірювання показників програмних модулів;
- значення метрик – кількісна міра.



Рис. 3.6. Представлення класів бази даних

Окрім сутностей, наведених на рис. 3.6, у БД наявні допоміжні сутності, які носять інформативний характер:

- досвід експерта;
- тип ПЗ;
- мова програмування.

Усі дані, одержані у результаті проведеного оцінювання та вимірювання, мають відношення до конкретного модуля ПЗ та експерта, який їх проводив.

Реалізацію бази даних виконано на основі реляційної бази даних з використанням СКБД MySQL. ER-діаграму БД представлено на рис. 3.7.

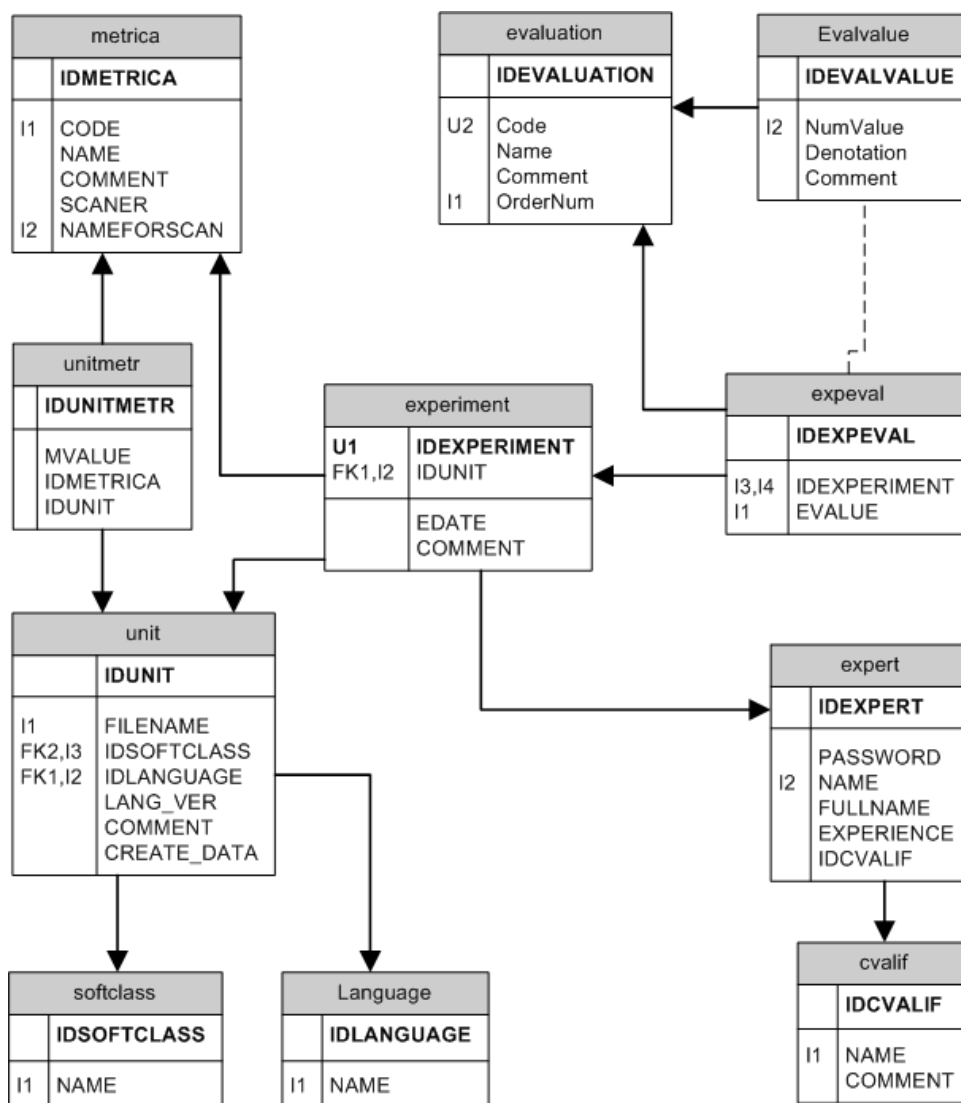


Рис. 3.7. ER-діаграма бази даних

Функції, які покладені на базу знань і відповідний аналізатор передбачають виконання:

- аналізу та формування залежності між оцінками і метриками з врахуванням результатів експериментів;
- можливість збереження формальної залежності між оцінками і метриками і забезпечення здатності до їх модифікації.

Інструменти для проведення аналізу даних мають розв'язувати задачі добування знань на основі існуючих результатів експерименту, які наявні у базі даних, тобто забезпечувати набуття нових знань щодо залежностей між оцінками і метриками.

Основне завдання бази знань полягає у забезпеченні здатності збереження знань щодо існуючих залежностей у такому вигляді, який заданий на етапі проектування структури знань. У даному випадку, структура знань щодо залежностей представляється у вигляді аналітичних функцій, визначених та уточнених із застосуванням регресійних і статистичних методів, а також методів побудови нейромереж і fuzzy-логіки. Такі інструменти реалізовані у вигляді бібліотеки, яка імплементована у середовище MathLab.

3.2.2. Вимірювач атрибутів програмного коду

Міра – це певна властивість об'єкту, що може бути порівняна з деяким значенням. Для створення міри потрібно визначити атрибут (семантичне навантаження міри), шкалу і систему порівняння.

Для того, щоб одержати номінальне значення міри, потрібно володіти інструментами отримання значень атрибутів і засобом класифікації, що дає змогу видавати значення міри, як показано на рис. 3.8.

Для прикладу, у випадку визначення мови програмування, якою реалізований потенційний компонент повторного використання, ознакою міри може бути розширення файлу.

Сенсор ознаки призначений для виявлення та виділення розширення з назви файлу. Інструмент прийняття рішення має забезпечувати виконання ідентифікації значення за допомогою асоціативних правил типу: «if Extension = ".c" or ".cpp ", then value = " C/C++ "». Для того, щоб одержати номінальне значення міри достатньо, щоб обрані ознаки відповідали властивості еквівалентності.



Рис. 3.8. Структура засобу одержання номінальних значень мір

У випадку, коли використовуються порядкові міри необхідно, щоб була наявною хоча б одна інтенсивна ознака і використовувалась операція порівняння при прийнятті рішення. Як приклад, виберемо із сукупності ознак, що визначають атрибут читабельність програмного коду множину з трьох величин {«відсоток стрічок, які містять відступи», «відсоток пустих стрічок», «середнє значення довжини стрічки»} і створимо правила типу: «" if percent of strings with indentation"> 20 and "percent of empty strings"> 10 and "average length of string " <30, then value = " satisfactory" ».

Виходячи з результатів попереднього аналізу, можна зробити висновок що номінальні і порядкові міри інтерпретують «отримані значення», але не «виміряні значення», тобто вимірювання застосовується у випадку визначення адитивної міри.

Враховуючи те, що програмний код володіє дискретною природою, то для знаходження адитивної міри потрібно, щоб існувала одинична міра. Одинична міра дорівнює кванту міри, тобто необхідно мати інструмент для представлення мір і обчислення квантів. Як приклад, вимірювання довжини програмного коду може використовувати міру у вигляді стрічки коду, а інструментом вимірювання може виступати спеціальний сканер, який обчислює їхню кількість.

Отже, для забезпечення об'єктивності визначення будь-якої міри, необхідно забезпечити адитивність величин.

При аналізі метрик, які володіють адитивними мірами, виявлено, що для великої їх частини значення розраховуються як сума об'єктів, які представляються у вигляді одиничних мір. У програмному коді такі об'єкти, як рядок, програмна конструкція, оператор та інші, можуть використовуватися як одиничні міри, що є елементарними носіями деякого атрибуту. Для прикладу, значення довжини програмного коду можуть визначатись у вигляді 500 стрічок, 100 операндів або 50 функцій. Враховуючи вище викладені аспекти та впорядкованість при виконанні програмного коду, вимірювач повинен відповідати структурі та забезпечувати виконання операцій, наведених на рис. 3.9.

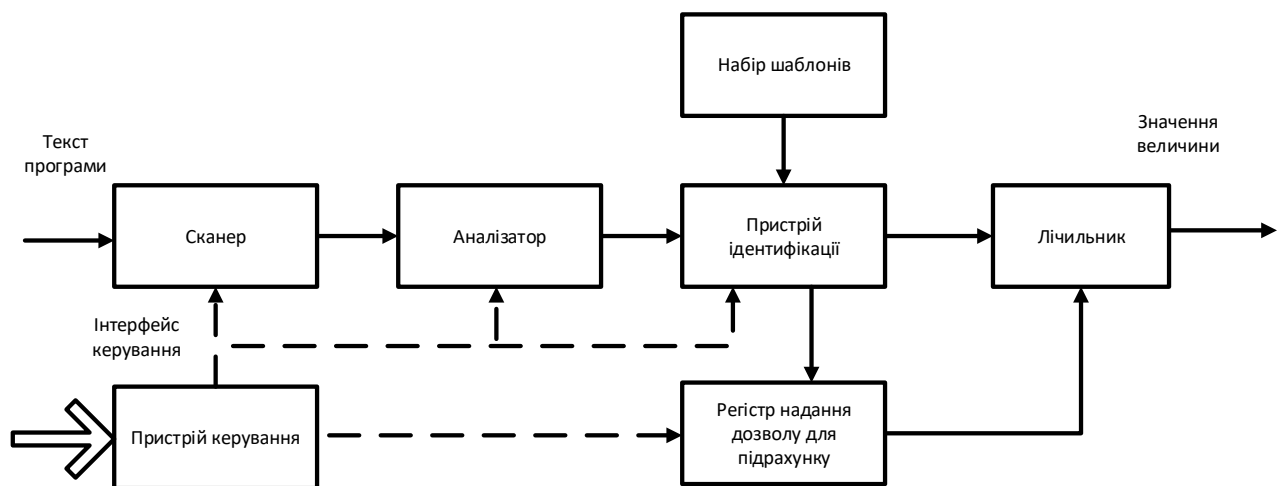


Рис. 3.9. Архітектура вимірювача

Згідно архітектури вимірювача, сканер забезпечує послідовне зчитування програмних знаків [14], тобто результатом роботи сканера на виході отримує коди символів та лінії управління, призначені для обміну сигналами початку і завершення з пристроєм керування.

Аналізатор застосовується в якості інструменту виділення конструкцій, які розташовані на деякому етапі аналізу програми. З метою забезпечення процесу виділення використовують граматики в якості інструменту відображення міри. Доволі частою практикою є обмеження області визначення об'єктів, наприклад,

«count of local variables» чи «count of variables in structures», що потребує не лише визначення власне конструкції коду, але і його контексту. На виході аналізатора одержують сукупність ідентифікаторів, які описують відповідні програмні конструкції. Лінії керування виконують функції конфігурації аналізатора за допомогою пристрою управління.

Інструмент ідентифікації забезпечує визначення об'єктів при їх підрахунку, а параметри при цьому встановлюються на основі патернів об'єктів. До складу патерну входить опис, що використовується при визначенні приналежності об'єкту до деякої групи.

Лічильник забезпечує підрахунок і запис значень кількості разових сигналів, лінії керування використовуються для його обнулення.

Компонент реєстр дозволяє фіксувати початок і кінець фрагментів програмного модуля, що відображає існування об'єктів.

Пристрій керування виступає у ролі конфігуратора та координатора при функціонуванні системи.

Функції обміну інформаційними сигналами, а також сигналами керування по відношенню до зовнішнього середовища покладено на інтерфейс вимірювача. Він забезпечує обмін такими даними, які показано на рис. 3.10, тобто програмний код і значення метрик і дані управління.

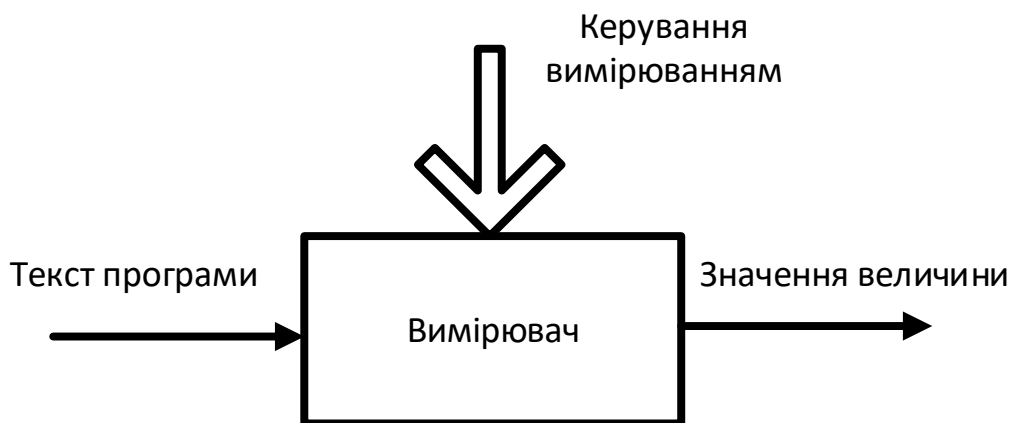


Рис. 3.9. Інтерфейс вимірювача

Для того, щоб забезпечити вимірювання встановленої метрики або атрибуту вимірювач має одержувати на вхід такі дані:

- шлях до розташування файлу, ідентифікатор і тип об'єкту для вимірювання;
- ідентифікатор приналежності до класу елементарних об'єктів, підрахунок яких потрібно виконати;
- ідентифікатор для визначення області розрахунків.

Після того, як встановлено усі необхідні вхідні дані, можна запускати вимірювач на виконання. Після завершення процесу вимірювання видається відповідний сигнал, що інтерпретується системою як вимірювання завершено. Сукупність інформації щодо опису інтерфейсу представлено у вигляді 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. Висновки до розділу

У даному розділі одержано наступні результати:

1. Створено функції та визначено режими функціонування інструментів, які дають змогу забезпечити ефективність прикладного застосування запропонованого методу оцінювання атрибутів ПВК.

2. Спроектовано архітектуру системи автоматизації процесу ідентифікації та кваліфікації програмного коду потенційних компонентів повторного використання при реалізації розумних комп'ютерних систем, що дало змогу скоротити час експертного оцінювання.

3. Створено інструмент для проведення вимірювань, пов'язаних з кількісними і статистичними критеріями програмного коду з використанням принципів теорії вимірювань, що дало змогу порівнювати оцінку експертів з автоматично визначеними значеннями метрик і мір.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Охорона праці

Під час проведення досліджень технологій оптимального вибору програмних компонентів повторного використання при проектуванні комп'ютерних систем усі процеси повинні виконуватись із врахуванням вимог техніки безпеки на робочому місці, пожежної безпеки, відповідно з діючими нормативно-правовими актами та встановленими нормами.

Перед початком усіх робіт проведено інструктаж з техніки безпеки на робочому місці відповідно до вимог Типового положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України ДБН В.1.1-7-2016 „Пожежна безпека об'єктів будівництва та НПАОП 0.00-4.36-05.

Розробка програмного забезпечення та проведення досліджень відбувались відповідно до Закону України "Про охорону праці" та "Правил охорони праці при експлуатації електронно-обчислювальних машин" та Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, а також НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [24].

Робоче місце відповідає ергономічним, організаційним вимогам, вказаних у вищеперерахованих документах.

Електробезпека на робочому місці, повинна відповідати вимогам Правил безпечної експлуатації електроустановок споживачів, затверджених наказом Держнаглядохоронпраці від 09.01.98 N 4 , зареєстрованих у Міністерстві юстиції України 10.02.98 за N 93/2533 (НПАОП 40.1-1.21-98) [20].

Для забезпечення електробезпеки при роботі з ЕОМ під час проведення досліджень електромережа відповідала правилам:

- лінія електромережі живлення виконана як окрема групова трипровідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників;

- нульовий захисний провідник використовується для заземлення(занулення) електроприймачів;

- електричне устаткування, електропроводи та кабелі мають апаратуру захисту від струму короткого замикання та інших аварійних режимів;

- під час монтажу та експлуатації електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежено застосування проводів з легкозаймистою ізоляцією та застосовано негорючу ізоляцію;

- електричне устаткування підключено до мережі лише за допомогою справних штепсельних з'єднань і розеток заводського виготовлення;

- у штепсельних з'єднаннях та електророзетках є спеціальні контакти для підключення нульового захисного провідника [20].

Електробезпека під час розробки прототипу системи оцінювання атрибутів компонентів повторного використання при проектуванні комп'ютерних систем забезпечувалась завдяки використанню безпровідних технологій та напруги живлення в діапазоні 3.3 – 9 В, що зменшує шанс ураження струмом та його вплив при виникненні контакту з мережею чи аварійних ситуацій.

Пожежна безпека будівель та приміщень, де розміщені робочі місця, обладнані ЕОМ з ВДТ і ПП відповідає вимогам, встановленим:

- державними будівельними нормами ДСТУ Б.В.1.1-36:2016 „Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою”;

- ДБН В.1.1-7-2016 „Пожежна безпека об'єктів будівництва”;

Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення відповідно до ДБН В.2.5-28-2006. Природне освітлення забезпечується через світлові прорізи, орієнтовані переважно на північ чи північний схід і

коефіцієнт природною освітленості (КПО) не нижче ніж 1,5%. Розраховується КПО за методикою, викладеною в ДБН В.2.5-28-2006.

Штучне освітлення в приміщенні з робочими місцями, обладнаними ВДТ ЕОМ та ПЕОМ, забезпечувалось системою загального рівномірного освітлення. В якості джерела світла для штучного освітлення використані люмінесцентні лампи типу ЛБ.

Також під час дослідження технологій оптимального вибору програмних компонентів повторного використання не було порушено наступних вимог:

- самостійного ремонту ЕОМ, ВДТ чи ПП;
- внесення змін до конструкції ЕОМ;
- не торкатись задньої панелі системного блоку при включеному живленні;
- вживання напоїв та їжі на робочому місці;
- нагромадження біля ЕОМ, ВДТ, ПП предмети, які не використовуються для поточної роботи.

Дослідження технологій оптимального вибору програмних компонентів повторного використання при проектуванні комп'ютерних систем проведено з дотриманням вимог техніки безпеки та охорони праці, електробезпеки, пожежної безпеки та санітарних норм при роботі з ЕОМ та ВДТ.

4.2. Планування та порядок проведення евакуації населення з районів наслідків впливу НС техногенного та природного характеру

В умовах неповного забезпечення захисними спорудами в містах та інших населених пунктах, що мають об'єкти підвищеної небезпеки, основним засобом захисту населення є евакуація і розміщення його у зонах, які є безпечними для проживання людей.

Евакуації підлягає населення, яке проживає в населених пунктах, що знаходяться у зонах можливого катастрофічного затоплення, можливого небезпечного радіоактивного забруднення, хімічного ураження, в районах

виникнення стихійного лиха, аварій і катастроф (якщо виникає безпосередня загроза життю та здоров'ю людей). Залежно від обставин, які склалися на час надзвичайної ситуації, може бути проведено загальну або часткову евакуацію населення тимчасового або безповоротного характеру. Загальна евакуація проводиться за рішенням Кабінету Міністрів України для всіх категорій населення і планується на випадок:

- можливого небезпечного радіоактивного забруднення територій навколо атомних електростанцій (якщо виникає безпосередня загроза життю та здоров'ю людей, які проживають в зоні ураження);
- виникнення загрози катастрофічного затоплення місцевості з чотиригодинним добіганням проривної хвилі.

Часткова евакуація проводиться за рішенням Кабінету Міністрів України у разі загрози або виникнення надзвичайної ситуації техногенного та природного характеру.

Під час проведення часткової евакуації завчасно вивозиться не зайняте у сферах виробництва та обслуговування населення: діти, учні навчальних закладів, вихованці дитячих будинків, разом з викладачами та вихователями, студенти, пенсіонери та інваліди, які утримуються у будинках для осіб похилого віку, разом з обслуговуючим персоналом і членами їх сімей.

У сфері захисту населення і територій від надзвичайних ситуацій техногенного та природного характеру евакуація населення планується на випадок:

- аварії на атомній електростанції з можливим забрудненням території; усіх видів аварій з викидом сильнодіючих отруйних речовин; загрози катастрофічного забруднення місцевості :
- лісових і торф'яних пожеж, землетрусів, зсувів, інших геофізичних і гідрометеорологічних явищ з тяжкими наслідкам, що загрожують населеним пунктам.

Загальна евакуація проводиться шляхом вивезення основної частини населення з міст і небезпечних районів усіма видами наявних транспортних

засобів на відповідній адміністративній території та виведення найбільш витривалої його частини пішки. Часткова евакуація проводиться з використанням транспортних засобів, що експлуатуються за діючим графіком. На органи виконавчої влади, органи місцевого самоврядування та керівників об'єктів, які проводять евакуацію населення, покладається:

- планування і проведення евакуації працівників та членів їх сімей;
- подання до відповідних транспортних органів розрахунків потреби у транспортних засобах для вивезення працівників і членів їх сімей до безпечних районів;
- контроль за плануванням, підготовкою і проведенням евакуаційних заходів підвідомчими об'єктами;
- визначення та підготовка безпечного району для розміщення евакуйованих працівників і членів їх сімей.

Інші заходи та порядок проведення евакуації викладено у постанові Кабінету Міністрів від 26 жовтня 2001р. № 1432 про затвердження Положення про порядок проведення евакуації населення у разі загрози або виникнення надзвичайних ситуацій техногенного та природного характеру.

У плані евакуації, складовою частиною якого є карта (схема), зазначаються:

- висновки з оцінки обстановки у разі виникнення надзвичайної ситуації;
- порядок оповіщення населення про початок евакуації;
- кількість населення, яке підлягає евакуації, за віковими категоріями;
- терміни проведення евакуації;
- склад евакуаційних органів і терміни приведення їх у готовність;
- кількість населення, яке вивозиться різними видами транспортних засобів окремо і виводиться пішки;
- розподілення об'єктів за збірними евакуаційними пунктами, пунктами посадки, районами (пунктами) розміщення та евакуаційними напрямками; маршрути евакуації;

- райони (пункти) розміщення евакуйованого населення; пункти посадки на транспортні засоби, пункти висадки у безпечному районі, порядок доставки населення з пунктів висадки до районів (пунктів) розміщення;

- заходи щодо організації приймання, розміщення, захисту та життєзабезпечення евакуйованого населення у безпечному районі;

- порядок організації управління і зв'язку.

Розділ плану, в якому визначаються види забезпечення евакуації, розробляється відповідними службами. До цього розділу включаються:

- основні завдання служби;

- перелік сил і засобів, які залучаються для виконання евакуаційних заходів;

- терміни виконання завдань.

Евакуаційна комісія відповідного органу виконавчої влади, на території якої планується розміщення евакуйованого населення, розробляє план його приймання і розміщення у безпечному районі з картою (схемою).

У плані зазначаються:

- кількість евакуйованого населення за віковими категоріями, яке прибуває у район, місто, район у місті, селище, село;

- кількість об'єктів і їх розподіл за районами у місті, сільськими і селищними радами, населеними пунктами;

- чисельність населення, яке проживає на відповідній території;

- будівлі і споруди для розміщення об'єктів господарювання;

- пункти висадки евакуйованого населення;

- порядок і терміни доставки евакуйованого населення з приймальних евакуаційних пунктів до районів (пунктів) розміщення;

- порядок розміщення евакуйованого населення;

- порядок забезпечення евакуйованого населення продуктами харчування, водою, предметами першої необхідності, медичним та іншими видами обслуговування;

– порядок оповіщення посадових осіб, які відповідають за приймання евакуйованого населення, про початок евакуації і терміни прибуття населення.

План приймання і розміщення евакуйованого населення включає також розділ з транспортного забезпечення евакуації, в якому зазначається:

- кількість транспортних засобів кожного виду і термін їх подачі до пунктів посадки;
- кількість населення, яке підлягає евакуації;
- терміни відправлення евакуйованого населення у безпечні райони;
- терміни прибуття евакуйованого населення до пунктів посадки;
- маршрути руху транспортних засобів;
- кількість рейсів.

На всіх громадян, які підлягають евакуації, завчасно складаються списки за об'єктами і житлово-експлуатаційними організаціями у трьох примірниках, один з яких залишається на об'єкті або в житлово-експлуатаційній організації, другий (у разі одержання рішення про проведення евакуації) після уточнення списків надсилається на збірний евакуаційний пункт, третій - до евакуаційної комісії району (пункту) розміщення.

З отриманням рішення (сигналу) про проведення евакуації евакуаційні комісії уточнюють завдання керівникам об'єктів щодо проведення евакуаційних заходів, контролюють стан оповіщення населення, його збору, формування колон (через начальників маршрутів), забезпечують переміщення їх до пунктів евакуації, а також разом з транспортними службами - готовність транспортних засобів до перевезень, уточнюють порядок їх використання, підтримують постійний зв'язок з начальниками маршрутів та з органами виконавчої влади безпечних районів, інформують їх про хід евакуації.

У райони розміщення евакуаційних органів та населення, яке підлягає евакуації, направляються представники евакуаційних комісій для вирішення питань приймання, розміщення і життєзабезпечення евакуйованого населення.

Керівник органу виконавчої влади і евакуаційна комісія безпечного району, організовують підготовку пунктів висадки, розгортають приймальний

евакуаційний пункт, уточнюють кількість прибулих і порядок подачі транспортних засобів для їх вивезення з пунктів висадки, а також з проміжних пунктів евакуації до пунктів розміщення, контролюють роботу керівників об'єктів безпечних районів з прийому і розміщення евакуйованого населення.

У разі оголошення евакуації громадяни самостійно на міських транспортних засобах, які у цей період працюють цілодобово, прибувають на збірні евакуаційні пункти. Працівники цих пунктів розподіляють громадян, які підлягають евакуації, за транспортними засобами, інструктують їх і забезпечують посадку на транспортні засоби.

Евакуйовані громадяни повинні мати при собі паспорт, військовий квиток, документ про освіту, трудову книжку або пенсійне посвідчення, свідоцтво про народження, гроші і цінності, продукти харчування і воду на 3 доби, постільну білизну, необхідний одяг і взуття загальною вагою не більш як 50 кілограмів на кожного члена сім'ї. Дітям дошкільного віку вкладається у кишеню або пришивається до одягу записка, де зазначається прізвище, ім'я та по батькові, домашня адреса, а також ім'я та по батькові матері і батька.

ВИСНОВКИ

Основні наукові та практичні результати полягають в наступному.

1. Проаналізовано сучасні технології забезпечення якості атрибутів повторно використовуваних компонентів при проектуванні розумних комп'ютерних систем, що дало змогу визначити критерії впливу на ефективність їхньої імплементації в процесі побудови та експлуатації систем;

2. Проведено аналіз особливостей процесів реалізації компонентів повторного використання у випадку не систематичного підходу до розробки програмних компонентів комп'ютерних систем, що дало змогу підтвердити необхідність впровадження процедур при виборі та ідентифікації потенційних компонентів на роль повторно використовуваних.

3. Досліджено існуючі моделі для прийняття рішення щодо доцільності використання компонентів повторного використання на основі оцінювання вартості робіт та атрибутів потенційних компонентів, що дало змогу обґрунтувати їхнє застосування при виробництві ПВК.

4. Запропоновано використання методу експертного оцінювання щодо аналізу атрибутів і вартості робіт з реалізації повторно використовуваних компонентів на основі комбінованих моделей вартості і властивостей, що дало можливість забезпечити адекватність і повноту процесу вибору потенційних кандидатів у ПВК.

5. Запропоновано метод оцінювання атрибутів ПВК, що дає змогу забезпечити автоматизацію процесу одержання метрик атрибутів на основі розрахунку результатів вимірювань та із застосуванням моделей ознак (атрибутів).

6. Визначено набір властивостей потенційних компонентів програмних складових у розумних комп'ютерних системах, що є важливими при імплементації процесів повторного використання, на основі створення вихідного набору атрибутів моделі «Concept-Construction-Context» і відбору лише тих, які безпосередньо впливають на затрати при їх повторному використанні.

7. Запропоновано процедуру створення моделей атрибутів, що базується на принципах підходу системного аналізу та методу «Ціль-Питання-Метрика», що дало змогу забезпечити ефективність експертного оцінювання характеристик КПВ.

8. Створено функції та визначено режими функціонування інструментів, які дають змогу забезпечити ефективність прикладного застосування запропонованого методу оцінювання атрибутів ПКВ.

9. Спроектовано архітектуру системи автоматизації процесу ідентифікації та кваліфікації програмного коду потенційних компонентів повторного використання при реалізації розумних комп'ютерних систем, що дало змогу скоротити час експертного оцінювання.

10. Створено інструмент для проведення вимірювань кількісних і статистичних критеріїв програмного коду з використанням принципів теорії вимірювань, що дало змогу порівнювати оцінку експертів з автоматично визначеними значеннями метрик і мір.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бабенко Л.П. Информационная поддержка повторного использования в программной инженерии на базе UML. Кибернетика и системный анализ. 2003. №1. с.74-80.
2. Бабенко Л.П. Адаптивные компоненты многократного использования в системах генерации программ. Кибернетика и системный анализ. 2012. №5. с.145-149.
3. Бабенко Л.П. Онтологічні моделі опису готових ресурсів у розробці програм. Проб. прогр., №2-3, 2014. с.173-179.
4. Вельбицкий И.В. Технология программирования. К.: Техніка, 1984. 279 с.
5. Rod D. Kuhns. Strategies for Designing and Building Reusable GIS Application Components. URL: <https://proceedings.esri.com/library/userconf/proc98/PROCEED/TO600/PAP557/P557.HTM> (дата звернення 11.11.2022 р.)
6. Грищенко В.Н. Компонентно-ориентированное программирование. Состояние, направление и перспективы развития. Проб. прогр. 2002.№1-2. с.80-90.
7. Whitfield T. What is a reusable software component? URL: <https://www.lynx.com/embedded-systems-learning-center/what-is-reusable-software-component-rsc> (дата звернення 16.11.2022 р.)
8. Jalender B., Govardhan A., Premchand P. Designing code level reusable software components. International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.1, January 2012. URL: <https://airccse.org/journal/ijsea/papers/3112ijsea16.pdf> (дата звернення 25.11.2022 р.)
9. Крамар Ю. М. Автоматизация процесса контроля применения стиля языка программирования. Проблемы программирования. Материалы четвертой международной научно-практической конференции по программированию «УкрПРОГ2004». Спец. вып. 2014. № 2-3. с. 208-214.
10. Бабак В.П. Основи теорії ймовірностей та математичної статистики: Навчальний посібник. К.: КВІЦ, 2013. 432 с.

11. Сидоров Н.А. Инженерия утилизации программного обеспечения. Методы проектирования интеллектуальных прикладных программных систем. Киев. 2012, с.15-22.
12. Сидоров Н.А. Применение принципов программной инженерии в преподавании основ программирования. УСиМ, март-апрель 2009. с.78-86.
13. Цейтлин Г.Е. Алгебра логики та конструирования программ. К. Наук. Думка. 1994. 90с.
14. Лаврищева К.М. Програмна інженерія. К. 2008. 312 с.
15. Лаврищева Е.М. Методы программирования. Теория, инженерия, практика.К.: Наук. Думка. 2006. 451 с.
16. Яцишин В.В., Шаблій Н.Р., Дишкант І.М. Процес формування програмних компонентів повторного використання при реалізації комп'ютерних систем. Матеріали Х науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С.81.
17. Яцишин В.В., Дишкант І.М. Архітектура засобу підтримки процесу оцінювання потенційних компонентів повторного використання. Матеріали Х науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (8-9 грудня 2022 року). Тернопіль: ТНТУ. 2022. С.82.
18. Вершина А. Модель процесса разработки программного обеспечения. Пробл. програмув. № 2-3 [спец. вип.]. 2006. С. 269-274.
19. Хоменко В.А. Метод та засоби експертного оцінювання властивостей повторного використовуваних компонентів програмного забезпечення: автореф. дис... канд. техн. наук.: 01.05.03. Національний авіаційний ун-т. К., 2006. 16 с.
20. AL-Badareen A., Selamat M., Jabar A., Din J. Reusable Software Components Framework. URL: https://www.researchgate.net/publication/252066722_Reusable_Software_Components_Framework (дата звернення 20.11.2022 р.).

21. Sametinger J. Software Engineering with Reusable Components. URL: https://se.jku.at/wp-content/uploads/1997/01/1997.book_.pdf (дата звернення 25.11.2022 р.)

22. НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації ЕОМ». Наказ Держгірпромнагляду від 26.03.2010 № 6

23. Атаманчук П.С. Безпека життєдіяльності та охорона праці (Практичний курс): Навчальний посібник. Кам'янець-Подільський: "Думка". 2010. 152 с.

Додаток А
Тези конференцій

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

Х НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



7–8 грудня 2022 року

**ТЕРНОПІЛЬ
2022**

О. Гуменюк АНАЛІЗ РОБОТИ ІНСТРУМЕНТУ ДЛЯ УПРАВЛІННЯ ТА АНАЛІЗУ ЖУРНАЛІВ GRAYLOG	
О. Humeniuk PERFORMANCE ANALYSIS OF THE GRAYLOG LOG MANAGEMENT AND ANALYSIS TOOL	76
О. Гуменюк АНАЛІЗ РОБОТИ СТАНДАРТУ ЖУРНАЛЮВАННЯ SYSLOG	
О. Humeniuk ANALYSIS OF THE OPERATION OF THE SYSLOG JOURNALING STANDARD	77
А. Лупенко, С. Куліков, Д. Денисов КЛАСИФІКАЦІЯ ТА ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ПРИКЛАДНИХ ПРОГРАМНИХ ІНТЕРФЕЙСІВ ПРИ РЕАЛІЗАЦІЇ КОМП'ЮТЕРНИХ СИСТЕМ	
A. Lupenko, S. Kulikov, D. Denysov CLASSIFICATION AND FEATURES OF THE APPLICATION PROGRAMMING INTERFACES IN COMPUTER SYSTEMS IMPLEMENTATION	79
В. Яцишин, Н. Шаблій, Д. Денисов ПРИЗНАЧЕННЯ І ДОЦІЛЬНІСТЬ ВИКОРИСТАННЯ API GATEWAY У КОМП'ЮТЕРНИХ СИСТЕМАХ	
V. Yatsyshyn, N. Shabliy, D. Denysov PURPOSE AND FEASIBILITY OF USING API GATEWAY IN COMPUTER SYSTEMS	80
В. Яцишин, Н. Шаблій, І. Дишкант ПРОЦЕС ФОРМУВАННЯ ПРОГРАМНИХ КОМПОНЕНТІВ ПОВТОРНОГО ВИКОРИСТАННЯ ПРИ РЕАЛІЗАЦІЇ КОМП'ЮТЕРНИХ СИСТЕМ	
V. Yatsyshyn, N. Shabliy, I. Dyshkant THE PROCESS OF FORMING REUSABLE SOFTWARE COMPONENTS IN THE IMPLEMENTATION OF COMPUTER SYSTEMS	81
В. Яцишин, І. Дишкант АРХІТЕКТУРА ЗАСОБУ ПІДТРИМКИ ПРОЦЕСУ ОЦІНЮВАННЯ ПОТЕНЦІЙНИХ КОМПОНЕНТІВ ПОВТОРНОГО ВИКОРИСТАННЯ	
V. Yatsyshyn, PhD; Assoc. Prof., I. Dyshkant ARCHITECTURE OF THE SUPPORT TOOL FOR THE EVALUATION OF POTENTIAL REUSE COMPONENTS	82
А. Паламар, В. Дьомін, В. Волоський ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ДЛЯ МОНІТОРИНГУ СТАНУ ПРИСТРОЇВ БЕЗПЕРЕБІЙНОГО ЖИВЛЕННЯ	
A. Palamar, V. Domin, V. Voloskyi COMPUTER SYSTEM SOFTWARE FOR CONDITION MONITORING OF UNINTERRUPTIBLE POWER SUPPLY DEVICES	83
А. Паламар, В. Дьомін СТРУКТУРА МОДУЛЯ ДЛЯ МОНІТОРИНГУ СТАНУ ПРИСТРОЮ БЕЗПЕРЕБІЙНОГО ЖИВЛЕННЯ	
A. Palamar, V. Domin MODULE STRUCTURE FOR CONDITION MONITORING OF UNINTERRUPTIBLE POWER SUPPLY DEVICE	84
А. Паламар, І. Курпатий СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я ПАЦІЄНТІВ НА ОСНОВІ ІНТЕРНЕТУ МЕДИЧНИХ РЕЧЕЙ	
A Palamar, I. Kurpatyi PATIENT HEALTH REMOTE MONITORING SYSTEM BASED ON INTERNET OF MEDICAL THINGS	85

УДК 004.4

В. Яцишин, Н. Шаблій, І. Дишкант

Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ПРОЦЕС ФОРМУВАННЯ ПРОГРАМНИХ КОМПОНЕНТІВ ПОВТОРНОГО ВИКОРИСТАННЯ ПРИ РЕАЛІЗАЦІЇ КОМП'ЮТЕРНИХ СИСТЕМ

UDC 004.4

V. Yatsyshyn, N. Shabliiy, I. Dyshkant

THE PROCESS OF FORMING REUSABLE SOFTWARE COMPONENTS IN THE IMPLEMENTATION OF COMPUTER SYSTEMS

Процес створення компонентів повторного використання при реалізації комп'ютерних систем поділяють на операції, які формують два етапи: визначення і затвердження (рис. 1).

До стадії визначення або, по-іншому ідентифікації, входять операції з виявлення та вибору кандидатів у компоненти повторного використання. Процес виконання операції визначення може виконуватись в ручному режимі або за допомогою засобів автоматизації фахівцем з програмування. Визначені програмістом компоненти можна помітити маркером потенційного компонента до повторного використання і до них застосовується наступна операція – вибору. Оскільки, при аналізі можливе виявлення кількох потенційних компонентів, то для вибору кращих необхідно провести їх оцінювання щодо придатності використання як КПВ. Обрані програмні компоненти переміщують на стадію затвердження (рис. 1).

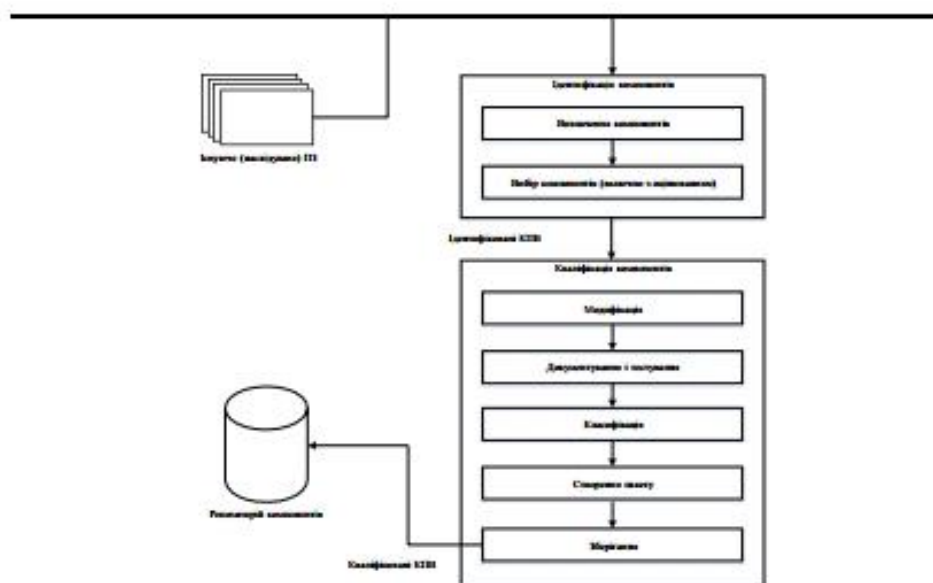


Рисунок 1. Процедура формування компонентів повторного використання

Вибір оптимального КПВ дає змогу знизити вартість реалізації КПВ, оскільки забезпечує виключення «гірших» альтернатив серед потенційних компонентів ще до стадії їх затвердження.

УДК 004.4

В. Яцишин, І. Дишкант

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

**АРХІТЕКТУРА ЗАСОБУ ПІДТРИМКИ ПРОЦЕСУ ОЦІНЮВАННЯ
ПОТЕНЦІЙНИХ КОМПОНЕНТІВ ПОВТОРНОГО ВИКОРИСТАННЯ**

UDC 004.4

V. Yatsyshyn, I. Dyshkant**ARCHITECTURE OF THE SUPPORT TOOL FOR THE EVALUATION OF
POTENTIAL REUSE COMPONENTS**

З метою визначення та затвердження компонентів, які в подальшому можуть бути використані в якості повторного використовуваних при проектуванні комп'ютерних систем, розроблено засіб автоматизації, архітектуру якого представлено на рис. 1.

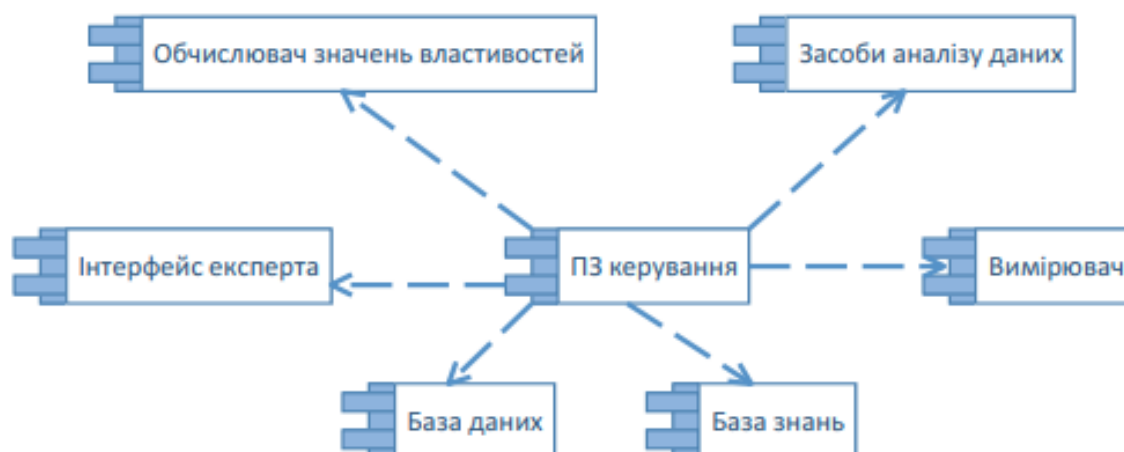


Рисунок 1. Архітектура засобу підтримки процесу оцінювання потенційних компонентів повторного використання

Як видно з рис. 1, основними компонентами засобу автоматизації є:

- вимірювач кількісних значень атрибутів компонента – призначений і використовується при обчисленні кількісних значень атрибутів компонента і формує результат вимірювання;
- інструменти аналізу даних – дають змогу забезпечити та уточнити існуючі залежності між оцінками і метриками атрибутів компонентів;
- користувацький інтерфейс експерта – інтерфейс, що забезпечує діалог експерта з системою оцінювання;
- ПЗ управління – відповідає за узгодженість роботи компонентів системи;
- вимірювач значень атрибутів – забезпечує одержання та інтерпретацію значень метрик на основі аналізу програмного коду компонента;
- база даних – відповідає за збереження експертних оцінок і метрик, результатів вимірювань та іншої додаткової і службової інформації;
- база знань – використовується для зберігання та опрацювання знань про залежності між властивостями і метриками компонентів повторного використання.