

Автоматизація оцінювання якості програмного забезпечення

Оленева К.М. ТМ-51м

Керівник: к.е.н. Сегеда І.В.

Якість програмного забезпечення

Якість програмного забезпечення (ПЗ) – це його здатність при заданих умовах задовольняти встановленим або передбачуваним потребам.

Характеризується якість ПЗ за 8 характеристиками, згідно до моделі якості ISO 25010:2011:

- функціональність;
- ефективність;
- надійність;
- безпека;
- зручність використання;
- зручність супроводу;
- сумісність;
- портативність.

З ростом кількості сучасних технічних засобів та можливостей ПЗ зростає зона його відповідальності та рівень довіри до нього.

Уже сьогодні людина довіряє ПП вирішувати важливі завдання, часто пов'язані зі здоров'ям, безпекою та життям людей. При вирішенні завдань такого роду якість цих продуктів має перебувати на належному рівні на всіх етапах його життєвого циклу.

Якість програмного забезпечення

Для успішного контролю якості необхідні формалізовані, кількісні оцінки показників якості.

Такими оцінками виступають *метрики* ПЗ. Вони знаходяться на найнижчому – фундаментальному рівні моделі якості ПП (згідно стандартів), та є основоположними.

Саме тому починати оцінювання якості потрібно з та базуючись на метриках.



Метрика – це міра, що дозволяє отримати чисельне значення деякої властивості ПЗ або його специфікацій, для того, щоб розуміти в якій мірі система, її компоненти або процес відповідають встановленим нормам.

Існуючі програмні рішення

Для проведення оцінки якості ПЗ існують різноманітні програми.

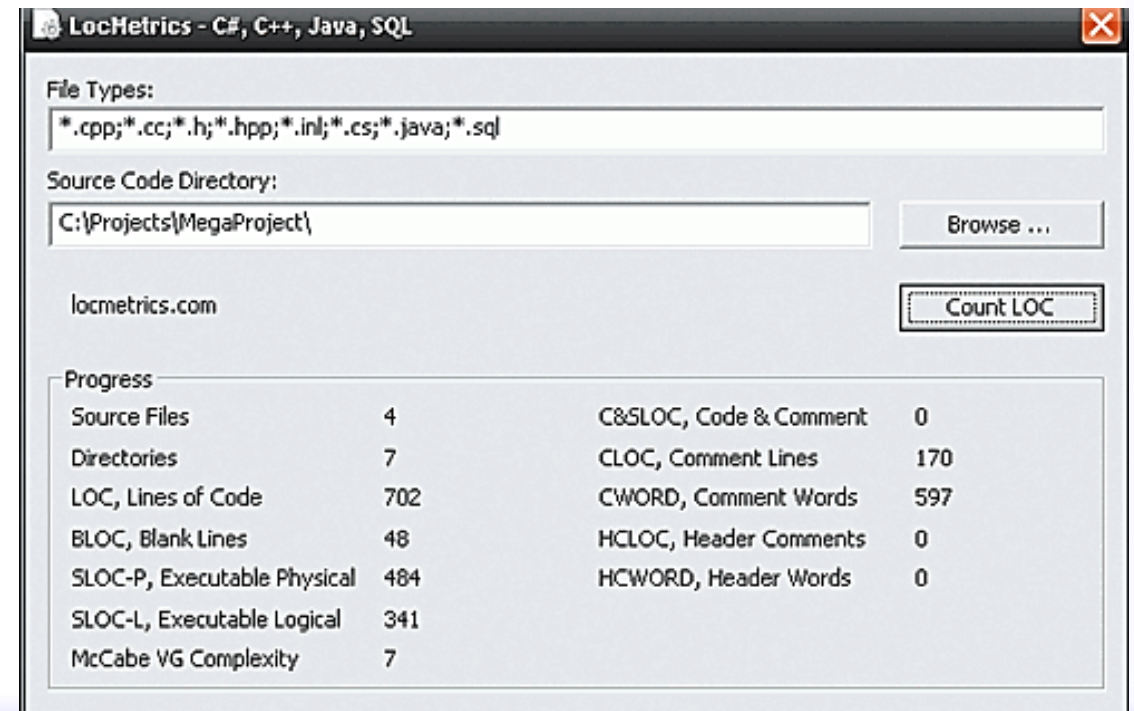
Серед них, наприклад: Locmetrics, USC Codecount, Code Counter Pro, SourceStat, Borland Together, PVS-Studio та інші.

Locmetrics – продукт для обчислення LOC-метрик.

До плюсів цього продукту можна віднести те, що він безкоштовний і має простий інтерфейс.

Але мінусами цієї розробки виступають:

- відсутність документації по використанню програми;
- відсутність опису методики розрахунку метрик;
- відсутність хоча б найпростіших засобів побудови звітності або експорту даних.



Існуючі програмні рішення

SourceStat – утиліта, яка виконує розрахунок статистики (метрик) для програм, написаних на мовах C/C++.

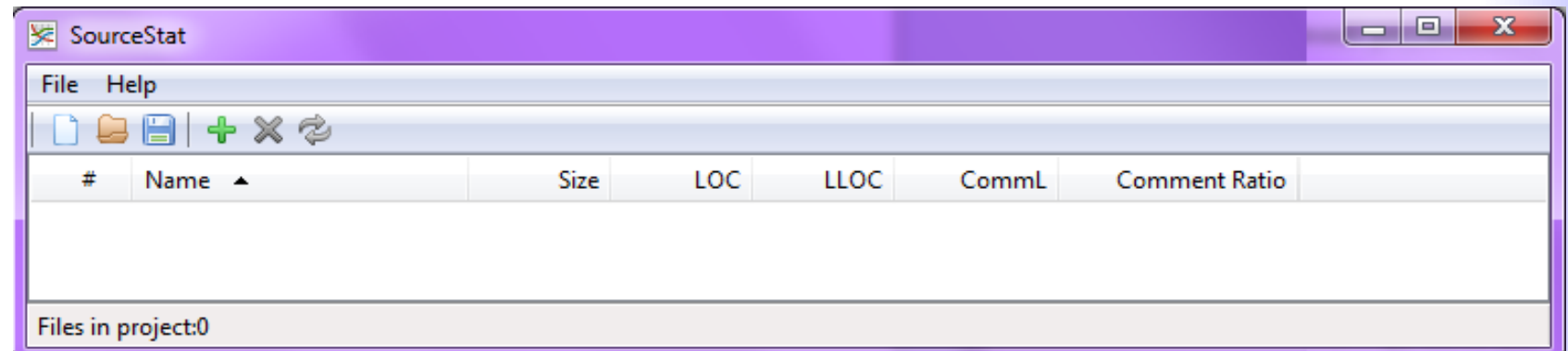
Актуальна версія SourceStat використовує наступний набір метрик: кількість рядків коду, кількість рядків коментарів, ставлення рядків коду до рядків коментарів.

До плюсів відносяться:

- простий інтерфейс, який дозволяє сортувати відображення метрик;
- збереження результатів роботи.

Серед мінусів:

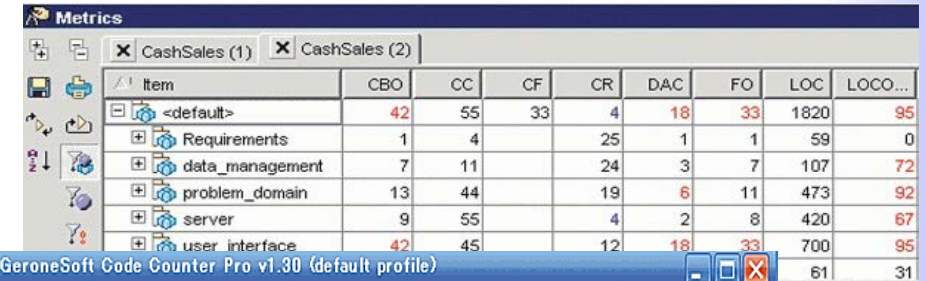
- дуже мала кількість метрик для обчислення;
- відсутність опису методики розрахунку метрик;
- програма працює лише під управлінням операційних систем Windows XP/Vista/7.



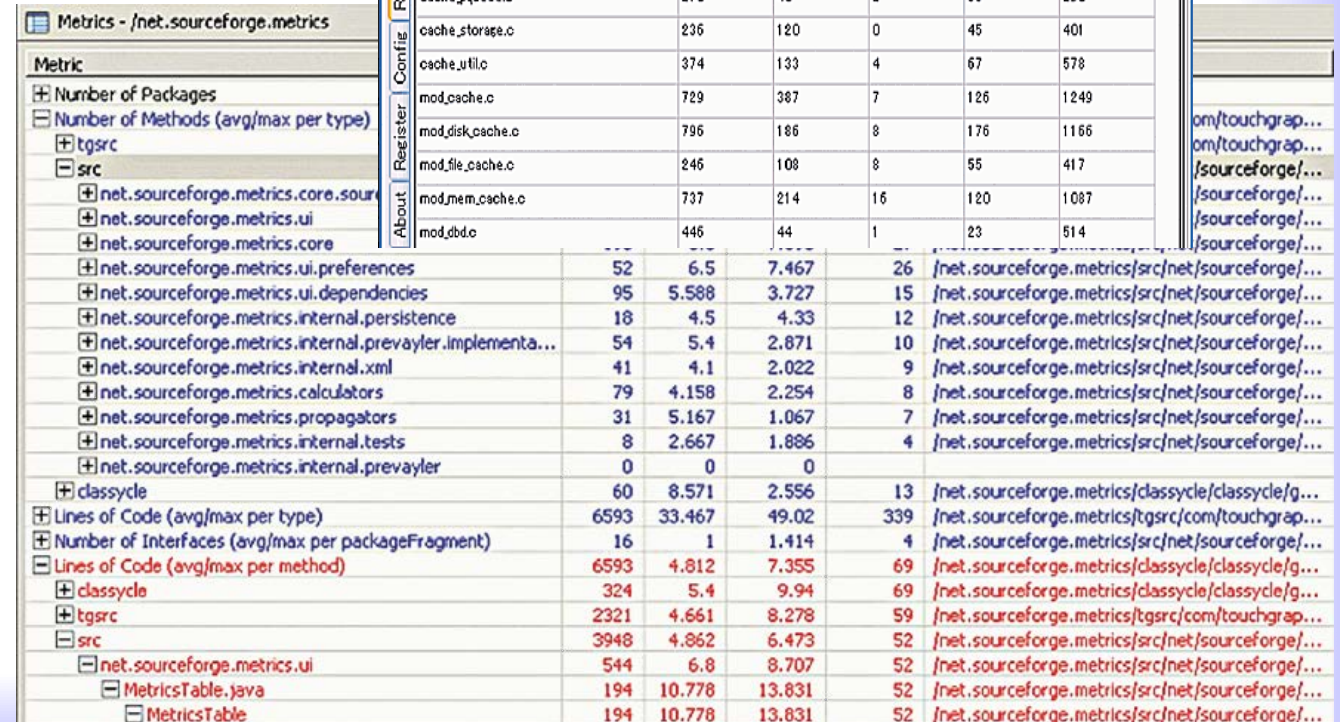
Але більшість продуктів, які призначені для оцінки ПЗ не володіють презентабельним інтерфейсом, а супроводжує їх досить скупа або зовсім відсутня документація.

Одні з них абсолютно безкоштовні, але при цьому розраховують невелику кількість метрик, а переважна більшість – є комерційними рішеннями та бувають неймовірно дорогими, які до того ж не можна попередньо випробувати.

Також, серед мінусів усіх програм для оцінювання якості ПЗ є те, що майже всі існуючі рішення застарілі та вже не підтримуються розробниками, а також не призначені для роботи на сучасних операційних системах.



Item	CBO	CC	CF	CR	DAC	FO	LOC	LOCO...
<default>	42	55	33	4	18	33	1820	95
Requirements	1	4		25	1	1	59	0
data_management	7	11		24	3	7	107	72
problem_domain	13	44		19	6	11	473	92
server	9	55		4	2	8	420	67
user_interface	42	45		12	18	33	700	95
							61	31



Filename:	Source:	Comment:	Both:	Blank:	Total:
cache_cache.c	124	17	0	30	171
cache_hash.c	168	97	1	24	290
cache_queue.c	210	40	2	38	290
cache_storage.c	236	120	0	45	401
cache_util.c	374	133	4	67	578
mod_cache.c	729	387	7	126	1249
mod_disk_cache.c	796	186	8	176	1166
mod_file_cache.c	246	108	8	55	417
mod_mem_cache.c	737	214	16	120	1087
mod_obj.c	446	44	1	23	514
net.sourceforge.metrics.core.sources	52	6.5	7.467	26	/net.sourceforge.metrics/src/net/sourceforge/...
net.sourceforge.metrics.ui.preferences	95	5.588	3.727	15	/net.sourceforge.metrics/src/net/sourceforge/...
net.sourceforge.metrics.internal.persistence	18	4.5	4.33	12	/net.sourceforge.metrics/src/net/sourceforge/...
net.sourceforge.metrics.internal.prevaler.implementa...	54	5.4	2.871	10	/net.sourceforge.metrics/src/net/sourceforge/...
net.sourceforge.metrics.internal.xml	41	4.1	2.022	9	/net.sourceforge.metrics/src/net/sourceforge/...
net.sourceforge.metrics.calculators	79	4.158	2.254	8	/net.sourceforge.metrics/src/net/sourceforge/...
net.sourceforge.metrics.propagators	31	5.167	1.067	7	/net.sourceforge.metrics/src/net/sourceforge/...
net.sourceforge.metrics.internal.tests	8	2.667	1.886	4	/net.sourceforge.metrics/src/net/sourceforge/...
net.sourceforge.metrics.internal.prevaler	0	0	0		
classycle	60	8.571	2.556	13	/net.sourceforge.metrics/classycle/classycle/g...
Lines of Code (avg/max per type)	6593	33.467	49.02	339	/net.sourceforge.metrics/tgsrc/com/touchgrap...
Number of Interfaces (avg/max per packageFragment)	16	1	1.414	4	/net.sourceforge.metrics/src/net/sourceforge/...
Lines of Code (avg/max per method)	6593	4.812	7.355	69	/net.sourceforge.metrics/classycle/classycle/g...
classycle	324	5.4	9.94	69	/net.sourceforge.metrics/classycle/classycle/g...
tgsrc	2321	4.661	8.278	59	/net.sourceforge.metrics/tgsrc/com/touchgrap...
src	3948	4.862	6.473	52	/net.sourceforge.metrics/src/net/sourceforge/...
net.sourceforge.metrics.ui	544	6.8	8.707	52	/net.sourceforge.metrics/src/net/sourceforge/...
MetricsTable.java	194	10.778	13.831	52	/net.sourceforge.metrics/src/net/sourceforge/...
MetricsTable	194	10.778	13.831	52	/net.sourceforge.metrics/src/net/sourceforge/...

Вирішення проблеми

Після проведення аналізу існуючих методів проведення оцінки якості ПЗ та вивчення існуючих програмних рішень, можна робити висновки про недосконалість існуючих рішень.

Що призводить до виникнення необхідності розробки нових або вдосконалення існуючих методів та розробки сучасних програмних рішень для оцінки якості ПЗ.

Тому було поставлене завдання створення автоматизованої системи, яка зробить проведення оцінювання якості доступним, а також, дозволить удосконалити спосіб визначення якості.

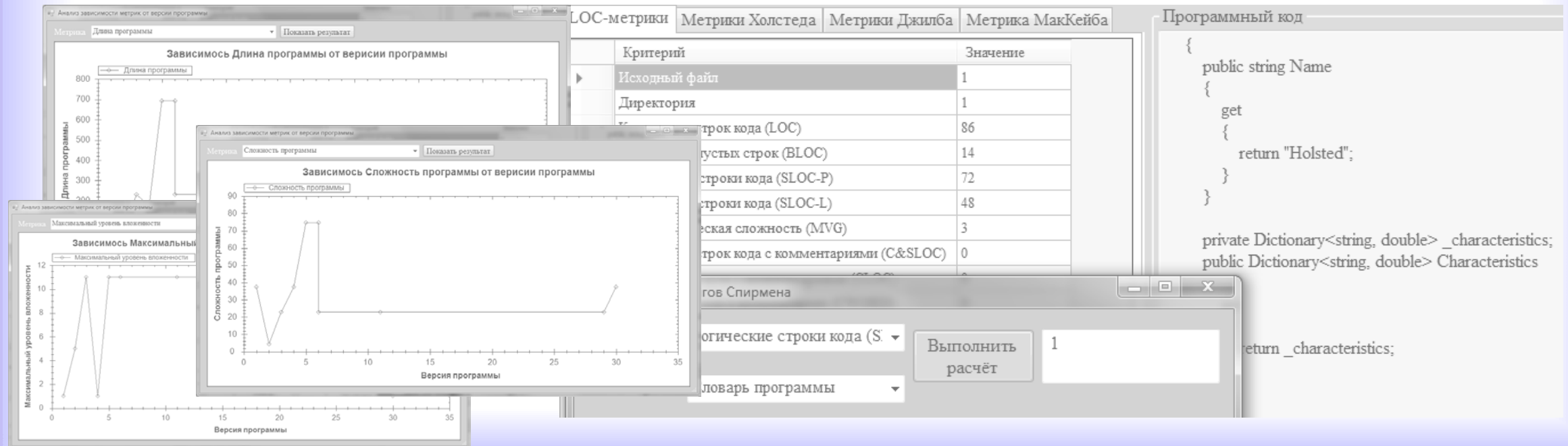


automated system of quality assessment software

Запропоноване рішення проблеми

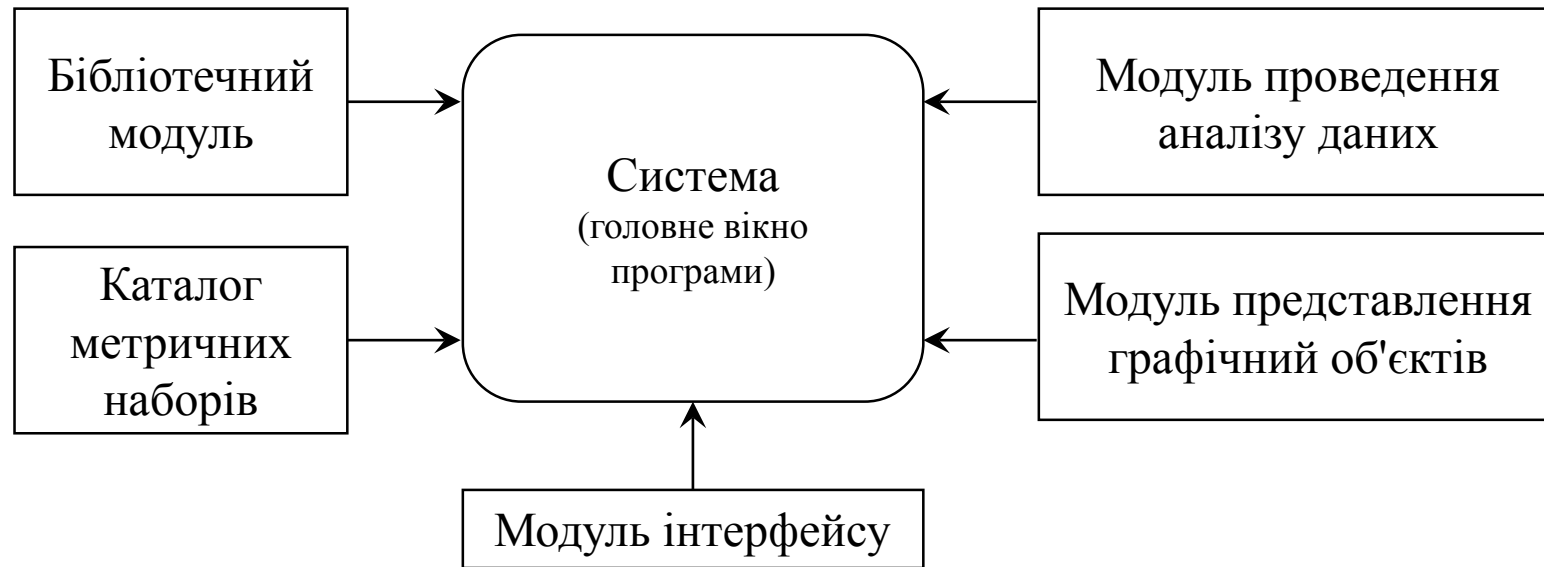
Робота з системою проходить у декілька простих етапів:

- користувач повинен відкрити програмний код та дати старт розрахунку метрик;
- надалі проводиться аналіз отриманих результатів, використовуючи коефіцієнти кореляції рангів Спірмена;
- також, проводиться графічний аналіз для відстеження залежності певної оцінки від змін у програмному коді;
- усі результати можна зберігати у вигляді звітів.



Функціональна схема

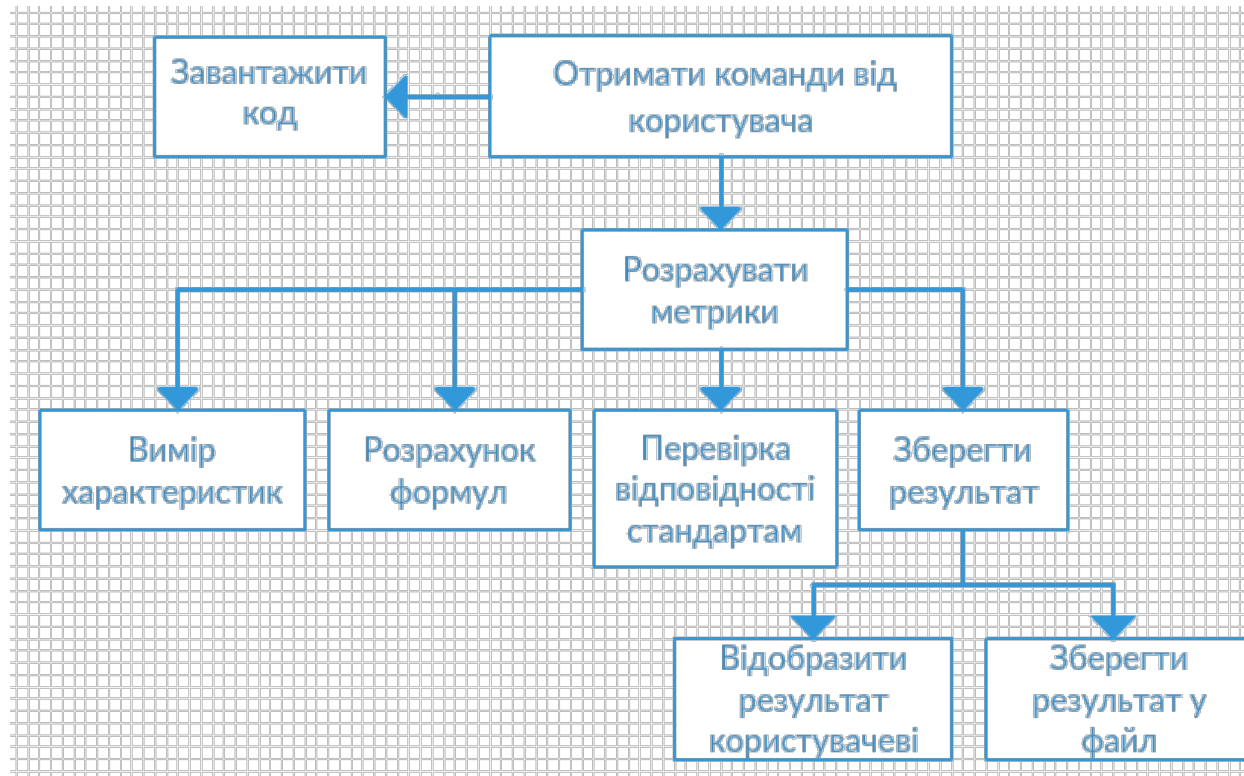
Для досягнення та реалізації сформульованих задач створену таку функціональну схему, для зображення зв'язку між модулями програмного рішення.



Функціональна декомпозиція системи

Архітектура створеної системи досить проста. Основними складовими можна виділити власне саму систему та її контент, а управління створеним програмним рішенням виконує безпосередньо користувач.

Склад автоматизованої системи представлений, використовуючи функціональну декомпозицію системи, що дозволяє побачити всі основні компоненти програми та їх взаємодію.



Метрики: LOC, Холстеда, Джилба

LOC-метрики

Ґрунтуються LOC-метрики на розрахунку прямих оцінок кількості критеріїв коду, при цьому розглядається не розмір, а функціональність або корисність продукту.

До цього набору метрик відносять:

- кількість рядків вихідного коду;
- кількість «фізичних» рядків коду;
- кількість «логічних» рядків коду;
- число порожніх рядків і число рядків, що містять коментарі;
- відсоток коментарів;
- середнє число рядків для функцій;
- середнє число рядків, що містять вихідний код для функцій;
- середнє число рядків для модулів.

Метрики Холстеда

Метрики Холстеда відносяться до метрик, які обчислюються на підставі аналізу числа рядків і синтаксичних елементів вихідного коду програми.

Оснoву метрики Холстеда становлять вимірювані характеристики програми, такі як: число унікальних / загальне число операторів програми, теоретичне число операторів, число унікальних / загальне число операндів програми, теоретичне число операндів.

З огляду на введені позначення, можна визначити:

- словник (n) і довжину програми (N);
- теоретичні словник (n`) і довжина програми (N`);
- обсяг (V) та теоретичний об'єм програми (V`);
- рівень якості програмування (L);
- рівень якості програмування, заснований лише на параметрах реальної програми без урахування теоретичних параметрів (L`);
- складність розуміння програми (EC);
- трудомісткість кодування програми (D);
- рівень мови вираження (y`).

Метрики Джилба

Дають можливість оцінити логічну складність програми, тобто дають поняття про те, наскільки складний код, але в той же час, як добре він зрозумілий.

Показують складність ПЗ на основі насиченості програми умовними операторами або операторами циклу

До цього набору входять:

- абсолютна складність програми;
- відносна складність ПП;
- максимальний рівень вкладеності.

Метрики: цикломатична складність МакКейба

Найпоширеніший і точний показник складності ПЗ, який вказує необхідну кількість прогонів програми, що потрібні для вичерпного тестування.

Визначається за формулою:

$$C = e - n + 2$$

де e – число ребер, а n – число вузлів на графі керуючої логіки.

У процесі автоматизованого обчислення показника цикломатичної складності, зазвичай, застосовується спрощений підхід, відповідно до якого побудова графа не здійснюється, а обчислення показника проводиться на підставі підрахунку числа операторів керуючої логіки (if, switch і т.д.) і можливої кількості шляхів виконання програми.

Відповідно до моделі якості, обрані метричні оцінки визначають такі критерії якості:

	LOC-метрики	Метрики Холстеда	Метрики Джилба	Метрика МакКейба
Функціональна придатність	+	+	+	+
Сумісність			+	
Надійність		+	+	+
Ефективність	+	+		+
Безпека			+	+
Портативність			+	
Зручність супроводу	+		+	
Зручність і простота використання	+	+	+	+

Коефіцієнт кореляції рангів Спірмена

Коефіцієнт рангової кореляції Спірмена – це кількісна оцінка статистичного вивчення зв'язку між явищами.

Розрахунок коефіцієнта складається з наступних етапів:

- ранжування ознак по зростанню (присвоєння рангів – порядкових номерів – відповідним значенням);
- визначення різниці рангів кожної пари;
- піднесення до квадрату різниці та знаходження загальної суми, $\sum d^2$;
- обчислення коефіцієнта кореляції рангів за формулою:

$$p = 1 - 6 \frac{\sum d^2}{n^3 - n}$$

де d_2 – квадратів різниць між рангами; N – кількість ознак під час ранжування.

Результати роботи

На етапі роботи з системою

Файл | D:\ДРИНА КРП\Диплом (магістратура)\ASQAS\Примеры 4 Form1.cs | Графический анализ | Справка

Версия программы: 32

Выполнить расчёты

Выполнить корреляционный анализ рангов Спирмена

Показать код

LOC-метрики | Метрики Холстеда | Метрики Джилба | Метрика МакКейба

Критерий	Значение
Исходный файл	1
Директория	1
Количество строк кода (LOC)	189
Количество пустых строк (BLOC)	20
Физические строки кода (SLOC-P)	161
Логические строки кода (SLOC-L)	94
Цикломатическая сложность (MVG)	13
Количество строк кода с комментариями (C&SLOC)	4
Количество строк с комментариями (CLOC)	8
Количество слов в комментариях (CWORD)	37

Программный код

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
using System.Runtime.Serialization.Formatters;
Server

partial class Form1 : Form
{
    protected Game game = null;
    protected Player player = null;
    bool admin = false;
    bool gamer = false;

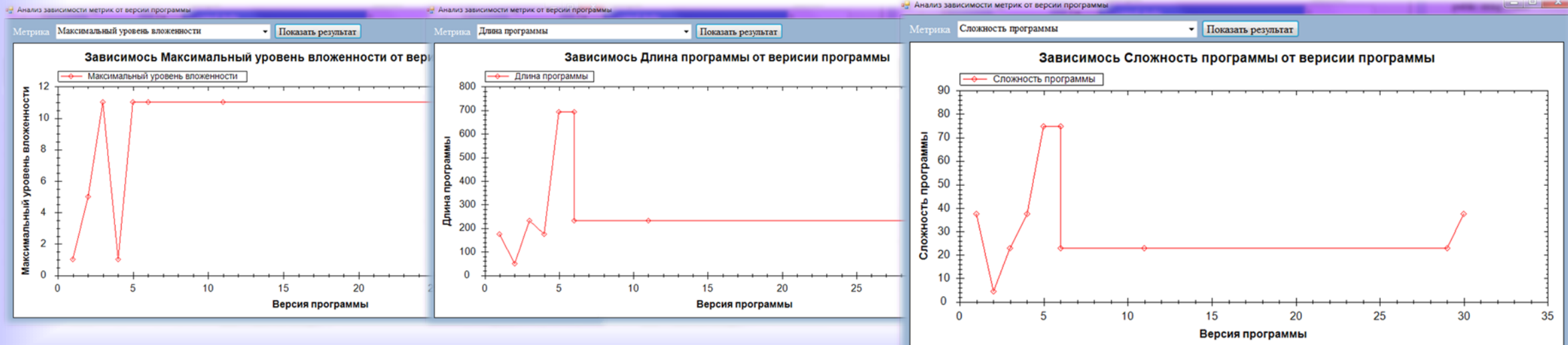
    Form1()
    {
        InitializeComponent();
    }
}
```

Корреляция рангов Спирмена

Метрика 1: Логические строки кода (S) | Выполнить расчёт | 0,0743148833397499

Метрика 2: Цикломатическая сложность (MVG)

Коэффициент = 0, связь отсутствует
Коэффициент (0;1] - связь прямая
Коэффициент [-1;0) - связь отрицательная



Результати роботи

Корреляция рангов Спирмена

Метрика 1: Количество строк кода (LC) 0,0743148833397499

Метрика 2: Цикломатическая сложнос

Корреляция рангов Спирмена

Метрика 1: Количество строк кода с к 0,778936180334248

Метрика 2: Объем программы

Корреляция рангов Спирмена

Метрика 1: Длина программы 0,146067415730337

Метрика 2: Абсолютная сложность пр

Коэффициент = 0, связь отсутствует
Коэффициент (0;1] - связь прямая
Коэффициент [-1;0) - связь отрицательная

Метрика: LOC

Исходный файл	1
Директория	1
Количество строк кода (LOC)	61
Количество пустых строк (BLOC)	6
Физические строки кода (SLOC-P)	55
Логические строки кода (SLOC-L)	29
Цикломатическая сложность (MVG)	10

Метрика: Holsted

Словарь программы	20
Длина программы	233
Объем программы	1007,009246
Сложность программы	22,75

Метрика: Jilb

Абсолютная сложность программы	10
Относительная сложность программы	0
Максимальный уровень вложенности	0,064935

Метрика: McCabe

Цикломатическая сложность программы	11
-------------------------------------	----

Збереження результатів

Висновки

Із результатів аналізу сучасного стану галузі оцінювання якості ПЗ впливають потреба й актуальність наукових досліджень у галузі оцінювання та прогнозування якості ПЗ на усіх етапах його життєвого циклу.

А також є необхідність покращення вже існуючих рішень оцінки якості.

Моделі оцінювання якості ПП включають метрики для визначення рівня кожної характеристики якості. Якщо характеристики якості обрані правильно, такі вимірювання можуть підтримати якість (рівень якості) багатьма способами. Метрики можуть допомогти в управлінні процесом прийняття рішень, сприяти пошуку проблемних аспектів і вузьких місць в процесах.

За час виконання магістерської дисертації була розроблена автоматизована інформаційна система, яка дозволяє забезпечити оцінку якості ПЗ за різними показниками якості та на усіх стадіях життєвого циклу ПП.

А також, покращено спосіб розрахунку трьох розмірно— та однієї об'єктно—орієнтованої метрик програмного коду.

Хоча, як кількісні оцінки (в даному випадку мова йде про результати оцінок, а не про процес вимірювань) характеристик якості можуть бути корисні самі по собі (наприклад, число незадоволених вимог і пропорція таких вимог), так можуть ефективно застосовуватися математичні та графічні техніки, які полегшують інтерпретацію значень метрик.

Тому у висновку хотілося б відзначити, що наявність необхідних метрик і графіків, які відображають зміни стану проекту в плинні часу, дозволить поліпшити не тільки процес тестування, а й розробки в цілому, а також полегшить процедуру проведення аналізу виконаного проекту, що дозволить в подальшому не допускати минулих помилок.

Гідність створеної реалізації, що лежить в основі розробки даної системи полягає в тому, що проведення порівняння двох об'єктів за одним критерієм не викликає ускладнень, а в результаті надає простий, зрозумілий та швидкий спосіб визначення якості ПЗ, через оцінку його коду.

Також подальшою перспективою досліджень є забезпечення повноти та цілісності показників і характеристик якості ПЗ. Для вирішення подібних питань необхідно визначити нові можливості покращення існуючих метрик, а також розглянути варіанти створення абсолютно нових методів оцінки якості ПП.



Дякую за увагу!

