МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

РЕФЕРАТ

на тему:

«Набір метрик Чидамбера і Кемерера»

Виконав:

Студент групи КН-416а

Бодня Є. В.

Перевірила:

Єршова С. І.

Харків – 2020

ЗМІСТ

[Вступ…...……………………………………………………………………………3](#_Toc500926339)

[1 Класифікація метрик….………………………………………………………….](#_Toc500926341)4

[2 Метрики Чидамбера і Кемерера ……..…………………….………………..…](#_Toc500926344)..5

[Висновки](#_Toc500926353) …………………………………………………………………………...10

[Список джерел інформації](#_Toc500926354) ………………………………………………………..11

ВСТУП

У міру вдосконалення комп'ютерів, розвиваються і вдосконалюються програмні засоби для роботи з ними. На сьогоднішній день ситуація в світі інформаційних технологій складається таким чином, що більшість великих проектів розробляється на об'єктно-орієнтованих мовах програмування. При розробці таких систем необхідно заздалегідь проектувати модель програми, яка найчастіше представлена ієрархіями класів, пов'язаних між собою. Забезпечення якості проекту є головним завданням, як керівників проекту, так і самих розробників.

Дуже важливо мати можливість оцінити якість проектованої системи. Це дозволить виявити недоліки і внести корективи на ранньому етапі створення програмного продукту і, тим самим, підвищити надійність, збільшити швидкість, спростити роботу і скоротити час розробки в цілому.

Ключовий фактор у виборі моделі оцінки є точність її показників. Це може здаватися простим для вивчення атрибутів комп'ютерних програмних продуктів, процесів, людей або середовища програмування, які можуть бути виміряні.

Для ефективної оцінки програмного забезпечення використовуються метрики.

Визначення метрики в контексті інформаційних технологій, відповідно до стандарту ISO 14598 звучить як кількісний масштаб і метод, який може використовуватися для вимірювання. Інше визначення метрики – захід, що дозволяє отримати чисельне значення деякої властивості програмного забезпечення [1]. Об'єктно-орієнтовані метрики вводяться з метою: поліпшити розуміння якості продукту, оцінити ефективність процесу конструювання, поліпшити якість роботи на етапі проектування.

Для будь-якої інформаційної системи визначені набори метрик повинні орієнтуватися на особливості і унікальні характеристики. Більшість метрик, були отримані дослідним шляхом в ході роботи над проектами інформаційних систем. Важливо враховувати, не тільки особливості проекту при виборі метрик, але і його специфіку.

1. **Класифікація метрик**

З точки зору метрик виділяють п'ять характеристик об'єктно-орієнтованих систем: локалізацію, інкапсуляцію, інформаційну закритість, успадкування та способи абстрагування об'єктів. Кожна з цих характеристик безпосередньо впливає на метрики. Загальна класифікація метрик, яка найбільш часто зустрічається в науковій літературі, представлена на рисунку 1.

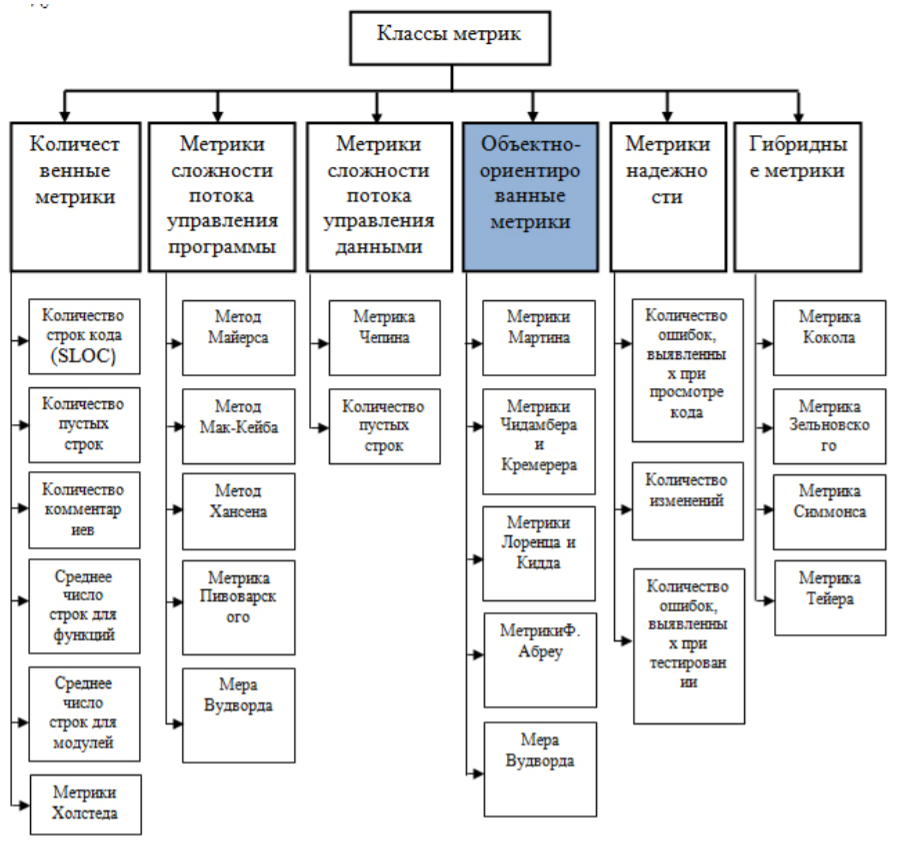


Рисунок 1 – Класифікація метрик

З рисунку видно, що класифікація метрик складається з семи основних класів. У кожному класі вказані найбільш часто використовувані види метрик. Найпростішими у використанні вважаються кількісні метрики. В даний клас входять такі метрики як: кількість порожніх рядків; кількість коментарів; відсоток коментарів (відношення числа рядків, що містять коментарі до загальної кількості рядків, виражене у відсотках); середнє число рядків для функцій (класів, файлів); середнє число рядків, що містять вихідний код для функцій (класів, файлів); середнє число рядків для модулів.

1. **Метрики Чидамбера і Кемерера**

Для аналізу і якісної оцінки проектів, створених за допомогою мов об'єктно-орієнтованого програмування, використовується клас об'єктно-орієнтованих метрик. В даному класі найбільш часто використовуваними є набір метрик Мартіна, набір метрик Чидамбера і Кемерера.

У 1994 році С. Чидамбер і К. Кемерер запропонували шість проектних метрик, орієнтованих на класи [3]. Клас – це основний, фундаментальний елемент об'єктно-орієнтованої інформаційної системи. Метрики для окремого класу, ієрархії класів і взаємодії класів є найбільш цінними для керівника проекту, який оцінює якість роботи. На рисунку 2 представлена ​​класифікація метрик С. Чидамбера і К. Кемерера.

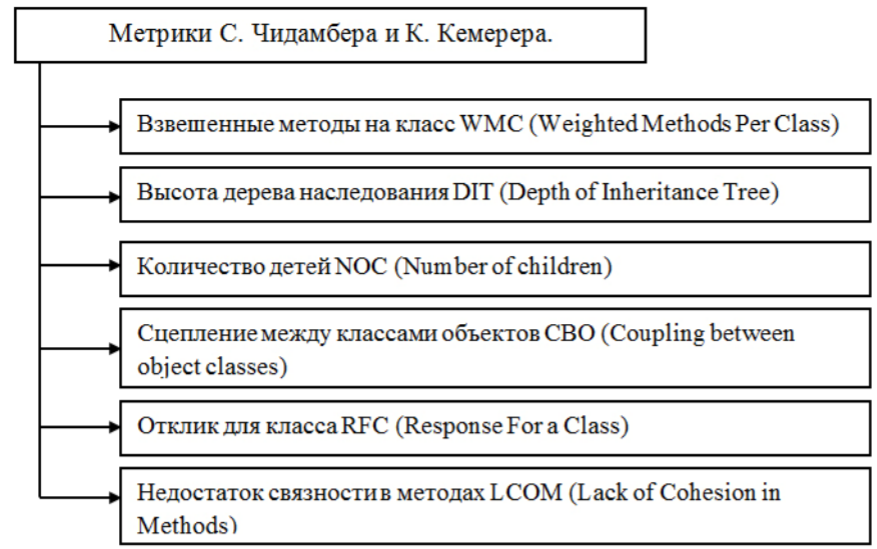


Рисунок 2 – Шість метрик С. Чидамбера і К. Кемерера

Першим видом метрик є метрика зважені заходи на клас (WMC). Використання цього заходу здійснюється за допомогою формули:

де – клас, – методи зі складністю

Складність методів може оцінюватися будь метрикою складності. Основною умовою є те, що дана метрика повинна бути нормалізована так, щоб номінальна складність для методу приймала значення 1. Кількість методів і їх складність є індикатором витрат на реалізацію і тестування класів.

З ростом кількості методів в класі, його застосування стає все більш специфічним, тим самим обмежується можливість багаторазового використання. За цих причин метрика WMC повинна мати розумно низьке значення.

Також застосовують спрощену версію метрики. При цьому вважають , після чого WMC прирівнюється до кількості методів в класі.

Для підрахунку кількості методів в класі використовують два протилежні варіанти обліку:

1. Підраховуються тільки методи поточного класу. Успадковані методи ігноруються. Обґрунтування – успадковані методи вже підраховані в тих класах, де вони визначалися.

2. Підраховуються методи, визначені в поточному класі, і всі успадковані методи.

На практиці застосовуються обидва варіанти. Основною умовою застосування є постійне використання одного варіанту обліку методів. Тільки в цьому випадку забезпечується коректний збір метричних даних [3].

Метрика WMC дає відносну міру складності класу. Можна припустити, що клас який має максимальну кількість методів серед класів одного з ним рівня, є найбільш складним.

Другий вид метрик – висота дерева спадкування (DIT). Вона визначається як максимальна довжина шляху від листа до кореня дерева успадкування класів. Приклад дерева успадкування класів схематично представлений на рисунку 3.

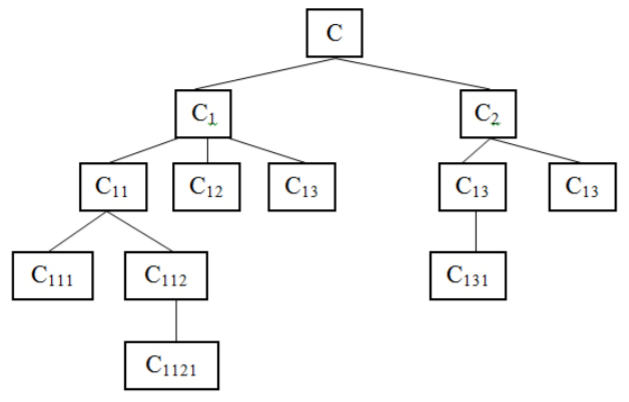


Рисунок 3 – Дерево успадкування класів

Для прикладу дерева класів, представленого на рисунку 3, метрика DIT = 4. Для окремого класу DIT, це довжина максимального шляху від даного класу до кореневого класу в ієрархії класів G.

По мірі зростання DIT, ймовірно, що класи нижнього рівня будуть наслідувати багато методів. Це призводить до труднощів в прогнозі поведінки класу. Висока ієрархія класів (велике значення DIT) призводить до більшої складності проекту, так як означає залучення більшої кількості методів і класів.

Разом з тим, велике значення DIT має на увазі, що багато методів можуть використовуватися багаторазово.

Третій вид метрик – кількість дітей NOC. Підкласи, які безпосередньо підпорядковані класу, називаються його дітьми.

Значення NOC дорівнює кількості дітей, тобто кількості безпосередніх спадкоємців класу в ієрархії класів. На рисунку 3 клас має двох дітей – підкласи і . У свою чергу підклас має одного спадкоємця – . Кількість дітей характеризує потенційний вплив класу на проект.

По мірі зростання NOC, зростає кількість тестів, необхідних для перевірки кожної дитини.

Четвертий вид метрик – зчеплення між класами об'єктів (СВО). Дана міра визначає кількість класів, з якими взаємодіє розглянутий клас. Дана метрика характеризує статичну складову зовнішніх зв'язків класів. Зі збільшенням значення СВО багаторазовість використання класу, зменшується. Чим більша незалежність класу, тим легше його повторно використовувати в інших програмах. Високе значення СВО ускладнює модифікацію і тестування, яке слід за виконанням модифікації.

П'ятий вид метрик – відгук для класу (RFC). RFC – це кількість методів класу плюс кількість методів інших класів, що викликаються з даного класу.

Метрика RFC є мірою потенційної взаємодії даного класу з іншими класами, дозволяє судити про динаміку поведінки відповідного об'єкта в системі. Дана метрика характеризує динамічну складову зовнішніх зв'язків класів. Якщо в , зростанням RFC збільшується складність класу. Найгірша величина відгуку може використовуватися при визначенні часу тестування [3].

Шостий вид метрик – недолік зв'язності в методах (LСOM). Кожен метод всередині класу звертається до однієї або декількох властивостей. Метрика LCOM показує, наскільки методи не пов'язані один з одним через властивості. Якщо всі методи звертаються до однакових властивостей, то LCOM = 0.

Для наочного уявлення використання метрик введемо позначення:

кількість пар методів без загальних екземплярних змінних;

кількість пар методів із загальними екземплярними змінними;

набір екземплярних змінних, які використовуються методом .

Тоді формула для обчислення нестачі зв'язності в методах набуде вигляду:

Зв’язність методів всередині класу повинна бути високою. Якщо LCOM має високе значення, то методи слабо пов'язані один з одним через властивості. Це збільшує складність, що збільшує ймовірність помилок при проектуванні [3].

Високі значення LCOM означають, що клас, ймовірно, треба спроектувати краще (розбиттям на два або більше окремих класів). Будь-яке обчислення LCOM допомагає визначити недоліки в проектуванні класів, так як ця метрика характеризує якість упаковки даних і методів в оболонку класу. Зв’язність в класі бажано зберігати високу, тобто слід домагатися найнижчого значення LCOM.

Набір метрик Чидамбера-Кемерера є основою оцінки проекту реалізованого на мові об'єктно-орієнтованого програмування. На даний момент часу існує велика кількість модифікацій, які пропонують оптимальні рішення оцінки проекту. Зокрема доповненням до набору метрик Чидамбера-Кемерера вважається метрика – поведінкова закритість інформації BIH (Behaviourial Information Hiding).

Формула застосування метрики:

де зважені зовнішні операції на клас (фактично це WMC), зважені внутрішні і зовнішні операції на клас.

Якщо , клас показує іншим класам всі свої можливості. Якщо , поведінка класу ховається від інших класів. може розглядатися і як міра складності. Складні класи, мають малі значення . Прості класи мають значення, які близькі до 1.

ВИСНОВОК

Використання наборів метрик Чидамбера-Кемерера дозволяє не тільки визначати складність проекту, але модифікувавши певні характеристики, врахувати особливості розроблюваного проекту, провести ефективний аналіз інформаційної системи, виявити вразливі місця. В даний час при оцінці проектів, реалізованих на мовах об'єктно-орієнтованого програмування мало використання одного набору метрик Чидамбера-Кемерера.

Поряд з розглянутими метриками використовуються метрики Лоренца і Кідда (розмір класу CS (Class Size), кількість операцій, доданих підкласом NOA (Number of Operations Added by a Subclass), індекс спеціалізації SI (Specialization Index), середній розмір операції OSAVG (Average Operation Size ), складність операції ОС (Operation Complexity), середня кількість параметрів на операцію NPAVG (Average Number of Parameters per operation), кількість описів сценаріїв NSS (Number of Scenario Scripts), кількість підсистем NSUB (Number of SUB system)) [4].

Набір метрик MOOD (Metrics for Object Oriented Design), запропонований Ф. Абреу (фактор закритості методу (МНF), фактор закритості властивості (AHF), фактор успадкування методу (MIF), фактор спадкування властивості (AIF), фактор поліморфізму (POF), фактор зчеплення (СОF)) і багато інших. Велика кількість метрик, розроблених для оцінки інформаційних систем на мовах об'єктно-орієнтованого програмування, дозволяє сформувати необхідний набір заходів для оцінки проекту.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Міжнародний стандарт ISO/IEC 25040:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Evaluation process. [Електронний ресурс] – Режим доступу: URL – http://webstore.iec.ch/preview/info\_isoiec25040%7Bed1.0%7Den.pdf
2. Орлов С. А. Технологии разработки программного обеспечения. Разработка сложных программных систем: учебник. – СПб.: Питер, 2002. – 464 с.
3. Черноножкин С. К. Методы и инструменты метрической поддержки разработки качественных программ: автореферат, – Новосибирск, 1998.
4. Богданов Д.В. Стандартизация жизненного цикла программных средств. – СПб., 2000. – 210 с.