МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Запорізький національний технічний університет

Кафедра програмних засобів

РЕФЕРАТ

На тему

**МЕТРИКИ СКЛАДНОСТІ ПОТОКУ**

**КЕРУВАННЯ І ДАНИХ ПРОГРАМИ**

з дисципліни

«Якість програмного забезпечення та тестування»

Виконав:

студент групи КНТ-136сп В.В. Гарковенко

2018

Зміст

[ВСТУП 3](#_Toc527384179)

[МЕТРИКИ СКЛАДНОСТІ ПОТОКУ КЕРУВАННЯ І ДАНИХ ПРОГРАМИ 4](#_Toc527384180)

[ВИСНОВОК 7](#_Toc527384181)

[СПИСОК ЛІТЕРАТУРИ 8](#_Toc527384182)

# ВСТУП

Вимоги до якості програмних засобів постійно підвищуються. Процеси розробки, впровадження складних систем, до яких відносяться зокрема програмні комплекси, повинні знаходиться під жорстким контролем. В даний час практично у всіх організаціях забезпечується контроль найважливіших характеристик, пов'язаних з виробництвом і використанням програмних продуктів, таких як час, фінансові кошти, ресурси.

В більшості випадків поза межами контролю виявляється важлива характеристика програмних продуктів – це якість продукту.

Одним з підходів для оцінки програмних засобів є оцінка атрибутів якості, які визначені у міжнародному стандарті ISO 9126-1-4[1]

Стандарти визначають загальний підхід до проблеми оцінки якості програмних засобів, що дозволяє зменшити невизначеність при спільній роботі. Застосування міжнародних стандартів зручно під час роботи з закордонними партнерами.

Відсутність можливості повного контролю викликає зростання кількості необґрунтованих рішень, збільшує фінансові витрати та ризики всередині проекту, під час розробки та впровадження продукту.

Зараз існують компанії, які мають достатньо досвіду використанні метрик для керування якістю програмного продукту. Використання протестованих підходів у керуванні якістю розробки та впровадження програмних продуктів значно підвищує передбачуваність проектів, та знижує ресурсні та фінансові витрати.

# МЕТРИКИ СКЛАДНОСТІ ПОТОКУ КЕРУВАННЯ І ДАНИХ ПРОГРАМИ

Метрики цього класу близькі як до кількісних метрик, так і до метрик складності потоку даних. Вони визначають складність програм як на основі кількісних підрахунків, так і на основі аналізу керуючих структур.

Дана категорія метрик використовується для аналізу програм, написаних з використанням високорівневих мов програмування.

Метрика Мак-Клура призначена для керування складністю структурованих програм в процесі проектування. Вона застосовується до ієрархічних схем розбивки програми на модулі, що дозволяє вибрати оптимальну схему розбиття. Метрикою виступає залежність складності програми від числа можливих шляхів виконання, числа керуючих конструкцій і числа змінних [2].

Методика розрахунку складності чітко орієнтована на структуровані програми.

Виділяють три етапи обчислення даної метрики:

1. Для кожної змінної обчислюється значення її функції складності C(i) за формулою:

,

де D(i) – величина, що стосується сфери змінної i. J(i) – міра складності взаємодії модулів через змінну i, n – кількість модулів у схемі розподілу.

1. Для модулів, що входять у сферу розбиття, визначається значення їх функцій складності M(p) за формулою:

,

де fp і gp – відповідно, кількість модулів, які передують і наступних за модулем P, X(P) – складність звернення до модулю P, Y(P) – складність керування викликом з модуля P інших модулів.

1. Загальна складність MP ієрархічної схеми розподілу програми на модулі задається формулою:

Мається на увазі що в кожному модулі одна точка входу і одна точка виходу, модуль виконує лише одну функцію, а виклик модулів здійснюється відповідно ієрархічній системі [3].

Тестуючою мірою М називається міра складності, що задовольняє наступним умовам:

1. Міра складності простого оператора дорівнює 1.
2. ))

Міра зростає з глибиною вкладеності і враховує протяжність програми. Найближчже до тестуючої міри прилягає міра на основі регулярних виразів. Її ідея в підрахунку кількості операторів, операндів з мінімальною кількістю дужок. Тестуюча М-міра та міра на основі регулярних виразів чутливі до вкладеності конструкцій та до протяжності програми.

Також мірою якості програмного забезпечення є зв’язність модулів програми. Якщо модулі сильно зв’язані, то програма стає важкомодифікованою та важкою в розумінні. Дана міра не виражається чисельно.

Види пов’язаності модулів:

Зв'язаність за даними - якщо модулі взаємодіють через передачу параметрів і при цьому кожен параметр є інформаційним об'єктом. Це найкращий тип пов'язаності.

Зв'язаність по структурі даних - якщо один модуль посилає іншому інформаційний об'єкт для обміну даними.

Зв'язаність з керуванням - якщо один модуль посилає іншому прапор, призначений для керування його внутрішньою логікою.

Модулі пов'язані з загальною областю в тому випадку, якщо вони посилаються на одну область глобальних даних. Зв'язаність по загальній області є небажаним, так як, по-перше, помилка в модулі, що використовує глобальну область, може проявитися в іншому модулі. По-друге, такі програми важкі для розуміння, так як програмісту важко визначити які саме дані використовуються конкретним модулем [4].

Зв'язаність за вмістом - якщо один з модулів посилається всередину іншого модуля. Це неприпустимий тип зчеплення, так як повністю суперечить принципу модульності.

Зовнішня зв'язаність - два модуля використовують зовнішні дані, наприклад комунікаційний протокол.

Зв'язаність за допомогою повідомлень - найбільш вільний вид пов'язаності, модулі безпосередньо не пов'язані один з одним, та спілкуються через повідомлення, які не мають параметрів.

Відсутність пов'язаності - модулі не взаємодіють між собою.

Підкласова зв'язаність - відношення між класом-батьком і класом-нащадком, причому нащадок пов'язаний з батьком, а батько з нащадком – не пов’язаний.

Зв'язаність за часом - дві дії згруповані в одному модулі лише тому, що з огляду на обставини вони відбуваються в один час [5].

Також існує метрика, заснована на інформаційній концепції – Міра Берлінгера. Міра складності розраховується:

де fi – частота появи символу, pi – ймовірність появи символу.

Недоліком є те, що програма може містити багато унікальних символів, але в малій кількості, буде мати однакову складність з програмою що містить невелику кількість унікальних символів, але у великій кількості [6].

# ВИСНОВОК

Підводячи підсумок, можна відзначити, що жодної універсальної метрики не існує.

Бідь-які контрольовані метричні характеристики програми повинні контролюватися залежно один від одного, або в залежності від завдання. Крім того існують комбіновані метрики, але вони так само залежать від простіших метрик і також не можуть бути універсальними.

Будь-яка метрика – це лише показник, який сильно залежить від мови програмування, стилю програмування, тому жодну міру неможна приймати як абсолютну та приймати рішення, ґрунтуючись тільки на ній

# СПИСОК ЛІТЕРАТУРИ

1. ISO 9126-1-4 [Електрон. ресурс] – Режим доступу: https://studbooks.net/2410710/informatika/standart\_9126
2. Метрика Мак-Клура [Електрон. ресурс] – Режим доступу https://studopedia.ru/12\_94181\_metrika-makkeyba.html
3. Інженерія якості програмного забезпечення: навчальний посібник / Г.В Табунщик, Р.К. Кудерметов, Т.І. Брагіна. – Запоріжжя: ЗНТУ, 2013. – 180 с.
4. Тестуюча М-міра [Електрон. ресурс] – Режим доступу https://habr.com/company/intel/blog/106082/
5. Види пов’язаності модулів [Електрон. ресурс] – Режим доступу https://www.viva64.com/ru/a/0045/#ID0EKZAE
6. Міра Берлінгера [Електрон. ресурс] – Режим доступу http://www.ispras.ru/preprints/docs/prep\_25\_2013.pdf