A New Approach to Quantifying Software Code Complexity

L. H. A. N. N. Buddhadasa
(MS23002456)

A THESIS
SUBMITTED TO
SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

December 2024

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Dr. Dilshan De Silva

Approved for MSc. Research Project:

MSc. Programme Co-ordinator, SLIIT

Approved for MSc:

_____

Head of Graduate Studies, FoC, SLIIT

DECLARATION

This is to certify that the work is entirely my own and not of any other person, unless explicitly acknowledged (including citation of published and unpublished sources). The work has not previously been submitted in any form to the Sri Lanka Institute of Information Technology or to any other institution for assessment for any other purpose.

Sign: ……..…..  .....

L. H. A. N. N. Buddhadasa

Date: ……2024.11.11……..

ABSTRACT

A New Approach to Quantifying Software Code Complexity

Nirmani Buddhadasa
MSc. in Information Technology
Supervisor: Dr. Dilshan De Silva
December 2024

This research presents a novel approach to quantifying software code complexity through the development of a new metric and a web-based tool using Python. Traditional complexity metrics often fall short in addressing modern software challenges such as concurrency, dynamic memory management, and exception handling. The goal is to create a paradigm-independent metric that combines conventional factors with new dimensions for a more comprehensive assessment of code complexity. The tool enables practical application of this metric with an intuitive interface for in-depth code analysis. It includes features like a secure login system, an interactive dashboard for complexity evaluation, and review sections offering detailed feedback on Python code. Data is collected from industry evaluations and Python repositories on platforms like GitHub and Kaggle, ensuring relevance and robustness. The effectiveness of the new metric and tool is assessed by comparing them with traditional complexity metrics, supported by insights from professional testers. Results demonstrate improved accuracy, adaptability, and flexibility in capturing complex software behaviors often overlooked by traditional metrics. The tool consistently outperforms metrics like Cyclomatic Complexity and Weighted Composite Metric in evaluating Python code complexity, providing more nuanced insights into modern challenges such as concurrency and exception handling. The anticipated outcomes highlight the enhanced accuracy and practical relevance of the new metric for assessing complex software systems. The tool serves as a valuable resource for developers, offering deeper insights into Python code complexity and supporting informed decisions in software design, optimization, and maintenance.

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to my supervisor, Dr. Dilshan De Silva, at the Sri Lanka Institute of Information Technology (SLIIT) for his unwavering guidance and support throughout my research. His insightful feedback, encouragement, and valuable advice were crucial to the completion of this project. I am truly thankful for his patience and dedication in helping me navigate through the challenges of my research.

I would also like to extend my sincere thanks to SLIIT for providing the resources and a conducive environment that enabled me to carry out this work successfully.

Lastly, I am deeply grateful to my parents for their constant encouragement, love, and support throughout this journey. Without their belief in me, this accomplishment would not have been possible.

TABLE OF CONTENTS

List of Figures

List of Tables