



Automated Analysis of Commenting Styles and Documentation Practices: A Data-Driven Approach to Software Quality and Maintainability

K.A.H.P.Sathyangani
(Reg. No.:MS24063296)

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 2025

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.

5th January 2026

Dr. Junius Anjana

Approved for MSc. Research Project:

MSc in IT 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is entirely my own and not of any other person, unless explicitly acknowledged (including citation of published and unpublished sources).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:Hansini.....

K.A.H.P.Sathyangani

Date: ...2025/12/14...

ABSTRACT

K.A.H.P.Sathyangani

MSc. in Information Technology

Supervisor: Dr. Junius Anjana

December 2025

Software maintainability is strongly influenced by the quality of code comments, which guide developers in understanding system functionality and behaviour. Poorly written, missing, or ambiguous comments reduce productivity and increase the cost of maintenance. The current study introduces an automated, data-driven approach to evaluating comment quality in Java projects. The proposed solution, implemented as a Java-based tool named Comment Quality Analyser, automatically scans source files, extracts comments, and evaluates them using four quality dimensions: grammatical correctness, readability, understandability, and meaningfulness. The tool integrates LanguageTool for grammatical analysis, the Flesch Reading Ease metric for readability, heuristic rules for understandability, and a Jaccard-similarity-based algorithm for measuring semantic alignment between comments and code identifiers. The results are presented through JSON reports and an interactive HTML dashboard that visualises the quality distribution across files. Real-world validation was conducted using the Apache Commons IO open-source repository, containing over 100 comments. Experimental results indicate that the system provides consistent scoring with an average accuracy of 86 % when compared with manual reviews. The proposed framework contributes to improving software documentation practices and offers a foundation for further research integrating Natural Language Processing (NLP) and Machine Learning (ML) to enhance software maintainability analysis.

ACKNOWLEDGEMENT

First and foremost, I express my heartfelt gratitude to my supervisor, Dr Junius Anjana, for his continuous guidance, patience, and constructive feedback throughout this research. His expertise and motivation were invaluable in shaping the direction of this study.

I also extend my appreciation to the lecturers and staff of the Sri Lanka Institute of Information Technology (SLIIT) for their academic support and resources provided during my MSc programme. Special thanks are due to my colleagues at Sysco Labs for their encouragement, technical discussions, and support during implementation and testing. Finally, I dedicate this work to my family and friends for their unwavering love, understanding, and belief in my potential throughout this journey.

TABLE OF CONTENTS

DECLARATION	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	iv
TABLE OF CONTENTS.....	v
List of Figures	x
List of Tables	xi
Chapter 1 – Introduction.....	1
1.1 Background of the Study	1
1.2 Problem Domain and Research Context	2
1.3 Research Problem Statement	3
1.4 Aim and Objectives.....	4
1.5 Scope of the Study	4
1.6 Research Significance.....	5
1.7 Methodological Overview.....	6
1.8 Thesis Structure Overview.....	7
Chapter 2 – Literature Review	8
2.1 Introduction to the Literature Review	8
2.2 Software Maintainability and Code Documentation.....	9
2.2.1 The Meaning of Maintainability.....	9
2.2.2 Documentation as a Quality Driver.....	10
2.2.3 Documentation in Modern Tool Ecosystems.....	11
2.2.4 Cognitive and Human Factors	11
2.2.5 Economic and Strategic Implications	12
2.2.6 Research Gaps and Summary	12
2.3 Code Comment Practices and Challenges	17
2.3.1 The role and evolution of comments.....	17
2.3.2 Obsolescence and drift	17
2.3.3 Redundancy, triviality and vagueness.....	18
2.3.4 Grammatical and stylistic inconsistency	18
2.3.5 Cultural and multilingual dimensions	23
2.3.6 Copy paste and structural inconsistency	24
2.3.7 Human and organizational factors.....	28
2.3.8 Industrial Automation and emerging tools.....	39
2.4 Metrics for Comment Quality	40
2.4.1 Evolution from density to semantics	40

2.4.2 Grammatical Correctness.....	40
2.4.3 Readability	41
2.4.4 Understandability.....	41
2.4.5 Meaningfulness and semantic relevance.....	42
2.4.6 Composite quality indices	42
2.4.7 Challenges in metric design	43
2.4.8 From metrics to actionable insights.....	43
Chapter 3 – Methodology	45
3.1 Introduction.....	45
3.2 Research Design and Approach	46
3.2.1 Type of Study	46
3.2.2 Conceptual Framework.....	47
3.3 System Architecture and Design.....	48
3.3.1 Overall Architecture	48
3.3.2 Input Parser.....	49
3.3.3 Preprocessing Engine	49
3.3.4 Scoring Engine	50
3.3.5 Reporting Layer	50
3.4 Data Collection and Sampling Strategy	51
3.4.1 Dataset Selection of Repository.....	51
3.4.2 Sampling and Data Volume.....	53
3.4.3 Ethical and Legal Considerations	53
3.5 Metric and Evaluation Criteria.....	53
3.5.1 Grammatical Correctness Metric.....	54
3.5.2 Readability Metric.....	54
3.5.3 Understandability Metric.....	55
3.5.4 Meaningfulness Metric	56
3.6 Experimental Procedure.....	56
3.6.1 Stage 1 – Data Acquisition	56
3.6.2 Stage 2 – Preprocessing	57
3.6.3 Stage 3 – Metric Computation	57
3.6.4 Stage 4 – Results Aggregation.....	58
3.6.5 Stage 5 – Interpretation and Reporting	59
3.7 Validation and Reliability	60
3.7.1 Internal Consistency.....	60
3.7.2 Inter-rater Reliability.....	60

3.7.3 Cross-tool Comparison.....	60
3.7.4 Reproducibility.....	61
3.8 Tools and Technologies.....	61
3.8.1 Programming Language and Framework.....	61
3.8.2 Integrated Development Environment.....	61
3.8.3 External Libraries.....	62
3.8.4 Execution Workflow.....	62
3.8.5 Data Visualisation and Reporting.....	63
3.9 Ethical Considerations and Limitations.....	63
3.9.1 Ethical Compliance.....	63
3.9.2 Avoidance of Bias.....	63
3.9.3 Limitations.....	64
Chapter 4 – Results and Evaluation.....	65
4.1 System Overview and Architecture.....	65
4.1.1 Input and Parsing Layer.....	67
4.1.2 Analysis Engine.....	67
4.1.3 Aggregation and Reporting Module.....	68
4.1.4 Visualisation Interface.....	68
4.1.5 System Integration and Workflow.....	68
4.2 Implementation Detail.....	69
4.2.1 Development Stack and Dependencies.....	70
4.2.2 Module-Level Implementation.....	72
4.2.3 Command-Line Interface and Execution Flow.....	75
4.2.4 Performance Optimisation.....	76
4.2.5 Logging, Validation, and Error Handling.....	76
4.3 Integration Workflow.....	77
4.3.1 Stage 1 – Input Acquisition.....	77
4.3.2 Stage 2 – Parsing and Data Extraction.....	78
4.3.3 Stage 3 – Comment Scoring Engine.....	78
4.3.4 Stage 4 – Aggregation and Report Generation.....	79
4.3.5 Stage 5 – Visualization and Post-Analysis.....	80
4.3.6 Integration with Real-World Projects.....	80
4.3.7 Error Handling and Data Integrity.....	81
4.3.8 Continuous-Integration Compatibility.....	81
4.3.9 Data Flow and Pipeline Synchronization.....	82
4.4 Dataset and Experimental Setup.....	82

4.4.1 Dataset Overview	83
4.4.2 Apache Commons-IO Dataset	84
4.4.3 Curated Java Sample Dataset.....	85
4.4.4 Data Collection and Extraction Pipeline.....	85
4.4.5 Data Pre-Processing	87
4.4.6 Experimental Environment	88
4.4.7 Evaluation Scenarios	89
4.4.8 Validation Procedures.....	90
4.4.9 Ethical Considerations and Licensing.....	90
4.5 Evaluation Metrics and Methodology.....	91
4.5.1 Metric Design Principles.....	91
4.5.2 Readability Metric	92
4.5.3 Grammar Metric	93
4.5.4 Understandability Metric.....	94
4.5.5 Meaningfulness Metric	95
4.5.6 Composite Quality Index (CQI).....	96
4.5.7 Methodological Framework.....	97
4.5.8 Statistical Analysis Techniques.....	97
4.5.9 Weight Calibration and Sensitivity Testing	98
4.5.10 Reproducibility and Benchmarking	98
4.6 Results and Analysis	99
4.6.1 Overview of Experiment Outcomes	99
4.6.2 Readability Analysis.....	100
4.6.3 Grammar Accuracy.....	101
4.6.4 Understandability Evaluation.....	102
4.6.5 Meaningfulness Alignment	103
4.6.6 Composite Quality Index (CQI) Comparison	104
4.6.7 Cross-Metric Correlation Matrix.....	105
4.6.8 Error Analysis and Outlier Detection	106
4.6.9 Qualitative Observations	106
4.6.10 Statistical Significance and Validation	107
Chapter 5 – Discussion, Conclusion and Future Work.....	108
5.1 Discussion.....	108
5.2 Limitations	109
5.3 Implications for Research and Practice.....	110
5.4 Future Work.....	111

5.5 Conclusion	112
References	113

List of Figures

Figure 3.2.2 Conceptual framework diagram depicts this relationship as a chain of dependencies	47
Figure 3.3.1 System Architecture Diagram illustrates the data flow among modules.....	49
Figure 4.1 Layered architecture comprising four primary components.....	66
Figure 4.1.5 System Workflow	69
Figure 4.4.4 The data-collection pipeline	86
Figure 4.6.3 Histogram of Grammar Quality Index (GQI)	101
Figure 4.6.4 The distribution of these values	102
Figure 4.6.5 Scatter plot: Meaningfulness vs Understandability.....	103
Figure 4.6.6 Boxplot comparison of Composite Quality Index (CQI).....	104

List of Tables

Table 3.8.3 Key libraries supported the CQA tool	62
Table 4.2.1 Libraries and Plugins Used in the Prototype Implementation	71
Table 4.4.1 Overview of Datasets Used for System Evaluation.....	83
Table 4.4.6 Experimental Environment Configuration.....	88
Table 4.5.2 Readability Score Interpretation Scale	92
Table 4.5.6 Composite Quality Index (CQI) Interpretation Scale	96
Table 4.6.1 Average Metric Scores and Composite Quality Index Across Datasets.....	99
Table 4.6.6 Statistical Summary of Composite Quality Index (CQI) Scores	104
Table 4.6.7 Cross-Metric Correlation Matrix	105