



Enhancing the Software Development Life Cycle through Integration of Generative Artificial Intelligence

Thilakarathna WLD
MS23004740

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.

Mrs Geethanjali Wimalarathna

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is entirely my own and not of any other person, unless explicitly acknowledged (including citation of published and unpublished sources).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: *W.L.D. Tilakaratne*

Thilakarathna WLD

10/11/2025
Date:

ABSTRACT

Enhancing the Software Development Life Cycle through the Integration of Generative Artificial Intelligence

W. Lasitha Dilshan Thilakarathna

MSc. in Information Technology

Supervisor: Mrs. Geethanjali Wimalaratne

December 2025

The Software Development Life Cycle (SDLC) is a foundational framework in software engineering, yet its documentation process spanning Business Requirement Documents (BRDs), User Stories, Test Cases, and Automation Scripts remain highly manual, inconsistent, and resource intensive. This study presents a Generative Artificial Intelligence (GenAI)-driven framework designed to automate and integrate SDLC documentation from end to end, enhancing traceability, efficiency, and quality across all stages. The proposed system employs Retrieval-Augmented Generation (RAG) in combination with FAISS-based semantic **retrieval** and LangChain orchestration to extract structured requirements from BRDs, generate standardized User Stories, derive Test Cases, and produce executable Cucumber and Selenium scripts. Both qualitative and quantitative methodologies were adopted: interviews with Business Analysts (BAs) and Quality Assurance (QA) engineers identified documentation challenges, while experimental evaluation measured performance and accuracy. The results demonstrate that the framework reduces documentation time by over 90%, ensures 100% traceability between SDLC artifacts, and achieves over 95% accuracy in generated outputs. Compared to general-purpose LLMs such as ChatGPT and Gemini, the proposed approach delivers structured, consistent, and production-ready documentation with minimal human intervention. This research contributes a validated and scalable model for AI-assisted SDLC automation, offering a significant step toward intelligent, traceable, and self-sustaining software documentation pipelines. The findings have both theoretical and practical implications, supporting the broader integration of Generative AI within enterprise software engineering and quality assurance practices.

Keywords: *Generative Artificial Intelligence, Software Development Life Cycle, Business Requirement Documents, User Stories, Test Case Generation, Test Automation, Selenium, Cucumber Scripts.*

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all those who supported me throughout this thesis journey at the Sri Lanka Institute of Information Technology. First and foremost, I extend my heartfelt thanks to my advisor, Ms. Geethanjali Wimalaratne, whose mentorship has been invaluable. Her insightful advice, constructive feedback, and unwavering support significantly shaped both my ideas and this thesis. I am immensely grateful for the opportunities she provided, including access to her vast network, as well as the freedom to pursue my own research interests under her guidance. Her encouragement played a pivotal role in the successful completion of this project.

Finally, I am grateful to my family, friends, and colleagues for their constant encouragement and understanding throughout this process. Their support kept me motivated and focused, making this thesis possible.

TABLE OF CONTENTS

DECLARATION	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT	iv
TABLE OF CONTENTS.....	v
List of Figures.....	ix
List of Tables	x
Chapter 1 - Introduction.....	1
1.1 Background of Study	1
1.2 Problem Statement.....	1
1.3 Research Gap	2
1.4 Research Objectives.....	3
1.5 Research Questions.....	3
1.6 Scope of the Study	4
1.7 Significance of Study.....	4
1.8 Chapter Summery	4
Chapter 2 – Literature Review.....	5
2.1 Introduction.....	5
2.2 Traditional Challenges in SDLC Phases.....	9
2.2.1 Requirement Analysis.....	9
2.2.2 Design and Implementation	9
2.2.3 Testing.....	10
2.3 Generative AI in System Development Life Cycle.....	10
2.3.1 Automating Requirement Analysis.....	13
2.3.2 Enhancing Test Case Generation	13
2.3.3 Automating Test Script Creation	14
2.4 Economic and Strategic Impact of Generative AI	14
2.5 Future Directions and Research Gaps.....	14
Chapter 3 - Methodology.....	16

3.1 Introduction.....	16
3.2 Research Design.....	16
3.3 Research Process.....	17
3.3.1 Stage 1: Problem Identification.....	17
3.3.2 Stage 2: Literature Review.....	17
3.3.3 Stage 3: Data Collection	18
3.3.4 Stage 4: System Design and Development	18
3.3.5 Stage 5: System Evaluation.....	19
3.3.6 Stage 6: Conclusion and Recommendations	19
3.4 Data Collection	20
3.4.1 Primary Data Collection.....	20
3.4.2 Secondary Data Collection.....	21
3.5 Data Analysis	21
3.5.1 Quantitative Analysis.....	21
3.5.2 Qualitative Analysis.....	22
3.6 Proposed System Architecture	22
3.6.1 System Overview	22
3.6.2 Workflow Diagram (Conceptual Description).....	23
3.7 Tools and Technologies Used.....	23
3.8 Validation Strategy	24
3.9 Limitations and Methodology.....	25
3.10 Chapter Summary	25
Chapter 4 - Results and Discussion.....	26
4.1 Introduction.....	26
4.2 Overview of the Framework	26
4.3 Document Processing.....	27
4.3.1 Chunking Optimization.....	27
4.3.2 Outcome	28
4.4 User Story Generation.....	28

4.4.1 Given Structure	28
4.4.2 Generated Output Example	28
4.4.3 Evaluation	29
4.5 Test Case Generation	29
4.5.1 Given Structure	29
4.5.2 Example Generated Output	30
4.5.3 Evaluation	30
4.6 Automation Script Generation	31
4.6.1 Generated Example	31
4.6.2 Evaluation	32
4.7 Comparative Analysis: General LLMs vs. Proposed Framework.....	33
4.8 Quantitative Evaluation	33
4.9 Limitations and Pending Work	34
4.10 Discussion	35
4.11 Chapter Summary	35
Chapter 5 – Discussion & Conclusion	36
5.1 Introduction.....	36
5.2 Summary of the Research	36
5.2.1 Research Objectives Recap	36
5.2.2 Summary of Findings.....	37
5.3 Contributions of the Research.....	37
5.3.1 Theoretical Contributions	37
5.3.2 Practical Contributions.....	38
5.4 Implications of the Study	38
5.4.1 Academic Implications	38
5.4.2 Industrial Implications	39
5.5 Limitations of the Study.....	39
5.6 Recommendations.....	39
References.....	41

Appendix..... 43

List of Figures

Figure 1: AI Model Architecture	23
Figure 2: Logical Flow of the System.....	27
Figure 3: User Stories Generation Prompt.....	28
Figure 4: User Stories Generation.....	29
Figure 5: Test Case Generation Prompt.....	30
Figure 6: Test Case Generation.....	30
Figure 7: Script Generation Prompt.....	32
Figure 8: AI Model Accuracies.....	32

List of Tables

Table 1: Tools and Technologies Used.....	24
Table 2: Evaluation Results	32
Table 3: Developed Application Vs General-purpose LLMs (ChatGPT / Gemini)	33
Table 4: Quantitative Evaluation	34