Independent University **Bangladesh (IUB)**

IUB Academic Repository

Graduate Thesis 2026

2025-12-14

A lightweight framework for implementing ISO/IEC 26550 in small and medium-sized software enterprises in Bangladesh

Hafiz, Zinia

Independent University, Bangladesh (IUB)

https://ar.iub.edu.bd/handle/11348/1568

Downloaded from IUB Academic Repository



A Lightweight Framework for Implementing ISO/IEC 26550 in Small and Medium-sized Software Enterprises in Bangladesh

By

ZINIA HAFIZ

1822332

Spring 2025

Supervisor:

Dr. Saadia Binte Alam

Associate Professor & Head

Department of Computer Science & Engineering

Independent University, Bangladesh

14 December 2025

Dissertation submitted in partial fulfillment for the degree of Master of Science in Software Engineering

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

I affirm that the work presented in this research paper is the result of my independent research and analysis. The insights, approaches, and outcomes discussed are based on my independent thinking, supported by the literature review and analysis of the data collected during the study. I have dedicated all of my efforts to maintaining the originality and integrity of this research. All external sources, including scholarly articles, relevant materials, books, and reports, have been properly cited and referenced in accordance with academic standards and guidelines. The primary data collection, obtained through surveys, interviews, or observations, was conducted by me and has been accurately documented and analyzed. The conclusions drawn reflect my meticulous interpretation of these findings. Throughout the research process, I have adhered to ethical principles, ensuring confidentiality and anonymity where necessary and obtaining the required permissions and approvals for data collection. The chosen methodology for the research was carefully designed and executed to ensure both rigor and validity.

.....
ZINIA HAFIZ

.....
Date

Acknowledgement

First and foremost, I would like to express my deepest gratitude to Almighty Allah for granting me the ability, patience, and perseverance needed to accomplish this research. His blessings have been my greatest source of encouragement throughout this academic path.

I sincerely thank the faculty of Independent University, Bangladesh (IUB), whose expertise and dedication greatly contributed to my academic advancement. I am especially grateful to my supervisor, **Dr. Saadia Binte Alam**, for his invaluable guidance, insightful feedback, and consistent encouragement throughout the research process. Furthermore, my heartfelt appreciation goes to **Ms. Farzana Sadia** for her kind support and assistance during various stages of my research.

I am also thankful for the support and collaboration of my fellow researchers, co-authors, and friends of IUB. Their teamwork and encouragement created a motivating and intellectually enriching environment that inspired me to achieve the goal.

Finally, I am profoundly grateful to my family and friends for their unconditional support, patience, and continuous encouragement. Their belief in me has been a great source of strength and motivation throughout this research journey.

Zinia Hafiz
December, 2025

Letter of Transmittal

To
Dr. Saadia Binte Alam
Associate Professor
Department of Computer Science & Engineering
Independent University, Bangladesh

Subject: Letter of Transmittal for Research Paper.

Dear Sir,

I am pleased to submit my thesis paper titled **"A Lightweight Framework for Implementing ISO/IEC 26550 in Small and Medium-sized Software Enterprises in Bangladesh"** for your review. This thesis has been completed as part of the requirements for the MSc in Computer Science (Software Engineering Thesis) at Independent University, Bangladesh.Bangladesh.

The study investigates the main challenges Bangladeshi software SMEs face in adopting the ISO/IEC 26550 standard and proposes a practical, lightweight framework that suits their limited resources. The findings are based on a detailed literature review, along with surveys and interviews conducted with industry stakeholders. Additional secondary sources, such as official publications and research articles, were used to support the analysis.

This work reflects my sincere effort to contribute to improving software process practices in developing economies. I would also like to express my heartfelt gratitude to my supervisor, Dr. Mahady Hasan, for his continuous guidance and thoughtful feedback throughout this research journey. Thank you very much for your time and consideration. I look forward to your valuable feedback.

Sincerely,
Zinia Hafiz
Email address: 1822332@iub.edu.bd

Evaluation Committee

Supervision Panel

.....	.....
Supervisor	Co-supervisor

External Examiners

.....	.....
External Examiner 1	External Examiner 2

Office Use

.....	.....
Program Coordinator	Head of the Department

.....

Director Graduate Studies, Research and Industry Relations

.....

Library

Abstract

This study explores the challenges faced by Bangladeshi software Small and Medium-sized Enterprises (SMEs) in adopting ISO/IEC 26550 for Software Process Improvement (SPI). Data collected from 12 SMEs revealed uneven implementation across key process areas, with high familiarity in domain requirements (up to 95%) but limited engagement in testing and validation practices. To address these gaps, the study proposes a lightweight PDCA (Plan–Do–Check–Act) framework tailored to the unique resource constraints of SMEs. Unlike traditional models such as CMMI or ISO/IEC 12207, the proposed approach emphasizes simplicity, cost-effectiveness, and role-based adaptability, making it more practical for SMEs in emerging economies. The framework supports scalable and gradual process improvement, enhances quality assurance, and facilitates stronger alignment with ISO/IEC 26550 in resource-limited environments.

Keywords— Small and Medium-sized Enterprises (SMEs), Capability Maturity Model Integration (CMMI), Plan–Do–Check–Act (PDCA).

Contents

Attestation i

Acknowledgement ii

Letter of Transmittal iii

Evaluation Committee iv

Abstract v

1 Introduction 1

1.1 Overview . . . 1
1.2 Contribution of the thesis . . . 2
1.2.1 My Contribution in the thesis . . . 2
1.3 Organization of the thesis . . . 3

2 Literature Review 5

2.1 Literature Selection Procedure . . . 5
2.2 Keywords Searched . . . 5
2.3 Databases for Search . . . 5
2.4 Inclusion and Exclusion Criteria . . . 6
2.5 Filtering Results . . . 6
2.6 Impact of the Project . . . 6
2.7 Literature Reviews of Paper . . . 7

3 Project management 14

3.1 Project Management overview . . . 14
3.2 Activity List . . . 14
3.3 WBS . . . 15
3.4 Gannt chart . . . 15
3.5 Network Diagram . . . 16

4 Published Paper 18

4.1 Problem Statement . . . 18
4.2 Research methodology . . . 21

4.2.1 Purpose of the Study 21
4.2.2 Data Collection 22
4.2.3 Data Analysis 23
4.2.4 Research Method 24
4.3 Proposed Solution 25
4.4 Result Analysis 28
4.4.1 Demographic Result 29
4.4.2 Exploratory Analysis 29

5 Sustainability **34**
5.1 Sustainability Overview 34
5.2 Economical Sustainability 34
5.3 Social and Environmental Sustainability 35
5.4 Ethics 35

6 Conclusion **36**
6.1 Project Summary 36
6.2 Future Work 36

Bibliography **36**

List of Figures

3.1 WBS 15
3.2 Gannt chart 16
3.3 Network Diagram 17

4.1 Participant Demographics by Role 22
4.2 Regression line Showing practice Item/Area Implemented by Organization 23
4.3 Diagram of Proposed Work 26
4.4 Regression Line Showing Implementations in Field of Practice By Organizations 31

List of Tables

3.1 Practices aim to solve the problems 14

4.1 Participant Demographic 24
4.2 Practice Items and their adaptation rate 30
4.3 Adaptation in field of practice 31
4.4 Cronbach's alpha 32
4.5 Practice, sub-areas, six sigma phases and action phase 32
4.6 Practices aim to solve the problems 33

Chapter 1

Introduction

1.1 Overview

In the ever-evolving landscape of software development, Software Process Improvement (SPI) remains a core aspect of ensuring quality, efficiency, and competitiveness [1]. SPI enhances product quality, reduces development costs, and strengthens organizational performance [2]. While international standards such as ISO 9001 and CMMI promote process consistency [6], ISO/IEC 26550 uniquely supports Software and Systems Product Line Engineering (SSPLE)—focusing on reuse, modularity, and scalability [3], [16]. ISO/IEC 26550 provides structured guidelines for managing software and systems development processes. However, implementing such standards poses significant challenges for Small and Medium-sized Enterprises (SMEs), particularly in developing countries like Bangladesh, where resource limitations, skill gaps, and infrastructural constraints are common. As a result, many local firms either ignore these standards or apply them inconsistently, leading to poor software quality, low productivity, and weak process control. In Bangladesh, SMEs dominate the software sector and often develop related products, yet SPLE adoption remains limited due to a lack of awareness, insufficient skills, and the perceived complexity of existing models [32] [33]. Existing SPI frameworks are typically heavyweight and fail to address the flexibility and cost-efficiency needs of smaller organizations [35]. In this context, a lightweight approach should be developed that aligns with international standards while accommodating local realities. Therefore, this study aims to design and propose a lightweight, role-based framework grounded in ISO/IEC 26550, specifically tailored to the resource-constrained environments of Bangladeshi software SMEs [15].

To achieve this, the paper introduces a lightweight PDCA-based SPI model tailored to Bangladeshi SMEs, grounded in ISO/IEC 26550. Using surveys (Cronbach's Alpha = 0.8622) and interviews across 12 SMEs, the model addresses key barriers and emphasizes cost-effective reuse, documentation, and process maturity.

Key Contributions:

• Assesses ISO/IEC 26550 embracing in SMEs over function-based analysis.

• Identifies implementation challenges such as awareness gaps and resource limits.
• Proposes a role-based, iterative PDCA framework for practical SPI. [3]
• Calls for further validation across industries and regions.
This framework offers a scalable path to progress software quality and SME competitiveness in developing markets [3].

1.2 Contribution of the thesis

The main contribution of this research is to increase the productivity of software companies in Bangladesh using a standard process improvement framework. The main focus is to improving the way adaptation ISO/IEC 26550 standard without high cost and complexity. This study offers a lightweight framework specifically to help these SMEs adopt ISO/IEC 26550 in a simple, affordable, and flexible way. ISO/IEC 26550 serves as the foundational standard within the suite of international standards for Software [38]-[39] and Systems Product Line (SSPL) engineering and management [47], [50]. However, many small and medium-sized enterprises (SMEs), particularly in countries like Bangladesh, have yet to fully realize or adopt its potential benefits. The proposed framework is assembled around the well-known PDCA (Plan–Do–Check–Act) cycle and interprets ISO/IEC 26550 into clear, handy steps that match the real-world roles and workflows found in small software companies. This framework actually helps process improvement which increases company's resources and maturity level. To enhancing this research, practical data has been collected from 12 SMEs in Bangladesh through questionnaire-based survey. Not only Bangladesh, others developing country will also be influenced by adopting the SPI practice. The impact of this thesis is principally valuable for legislators, consultants, and researchers focusing on software process improvement in similar socio-economic contexts. The participation of small company will be drastically increased to adopting this tailored framework. Lastly, this research contributes to the broader field of global software development practices. By providing a model and insights that are applicable to SMEs in developing countries, the study contributes to the global discourse on enhancing software development efficiency and quality, thereby improving competitiveness in the international software market.

1.2.1 My Contribution in the thesis

The overall study was collaborative, but my individual contribution focused on specific stages of data collection, analysis, and framework design.

Contribution in Data Collection:
My primary contribution to this research involved conducting empirical data collection and supporting the assessment of software process practices in small and medium-sized enterprises (SMEs) in alignment with ISO/IEC 26550. I was responsible for collecting data from 4 out of the 12 participating software organizations. This included coordinating with organizational representatives, administering structured questionnaires and interviews based on ISO/IEC 26550 process areas, and ensuring the reliability, completeness, and consistency of the collected data.

Contribution in Data Analysis:
During the data analysis phase, I conducted detailed analysis of the collected data from the assigned organizations. This involved data cleaning, categorization, and interpretation to ensure accuracy and consistency. I mapped organizational practices against ISO/IEC 26550 process areas to identify capability gaps and common implementation challenges. In addition, I contributed to cross-company comparative analysis by examining results across multiple organizations to identify recurring challenges and improvement patterns relevant to small and medium-sized software enterprises (SMEs).

Contribution in Proposed Solution:
Based on the analytical and empirical findings, I contributed significantly to the development of the proposed solution by identifying key challenges faced by small and medium-sized enterprises (SMEs) when adopting ISO/IEC 26550. These challenges included limited resources, lack of formal process documentation, and process overhead. I translated these findings into practical and lightweight improvement strategies, such as simplifying documentation requirements, consolidating roles where appropriate, and supporting incremental process adoption. The aim was to ensure that the proposed solution remains practical, feasible, and suitable for resource-constrained SME environments.

Contribution in Proposed Framework:
I contributed to the design and development of the proposed lightweight framework by helping define its structure, phases, and process mapping in alignment with ISO/IEC 26550. My role focused on simplifying standard requirements into practical and incremental steps that small and medium-sized enterprises (SMEs) can realistically adopt. I also supported the development of entry and exit criteria and ensured alignment between empirical findings and framework components. Additionally, I participated in reviewing and refining the framework through iterative discussions and feedback from co-authors to ensure its feasibility for resource-constrained software organizations.

1.3 Organization of the thesis

This thesis is organized into six chapters to guide the reader through the research journey in a logical and easy-to-follow way. Chapter 1 introduces the background, motivation and main contributions of the work. Chapter 2 reviews the existing literature on software process improvement, focusing on standards like ISO/IEC 26550, CMMI, and ISO/IEC 12207, and highlights the common challenges SMEs face in adopting such frameworks. Chapter 3 focuses on project management: it outlines the activity list, the work-breakdown structure (WBS), the Gantt chart and the network diagram used to plan and monitor the research. Chapter 4 presents the design of the lightweight framework itself, explaining its structure, key components, and how it supports SMEs through a simplified PDCA-based approach aligned with ISO/IEC 26550. Chapter 5 reports the findings from small and medium-sized enterprises, combining quantitative

and qualitative results and discussing their economic, social and environmental implications, along with ethical considerations. Finally, Chapter 6 concludes the study by summarizing the key findings and offering suggestions for future research or improvements.

Chapter 2

Literature Review

2.1 Literature Selection Procedure

We used a systematic method to choose the literature for the analysis so that it includes all relevant studies on ISO/IEC 26550, software process improvement (SPI), and the special problems that small and medium-sized organizations face. The process started by finding the right databases. Then, keyword-based searches were used, inclusion and exclusion criteria were applied, and the results were filtered to get to the final pool of studies. This organized method made sure that the review was reliable, clear, and could be repeated.

2.2 Keywords Searched

To reach appropriate publications, combinations of keywords were used:

- "ISO/IEC 26550"
- "Software Process Improvement" OR "SPI"
- "Small and Medium-Sized Enterprises" OR "SMEs"
- "Adoption challenges" AND "Bangladesh"
- "Software product-line engineering"
- "Software development process,"
- "Software quality"
- "CMMI"
- "International organization for standardization"
- "Problems and Challenges of Small and Medium Enterprises in Bangladesh"
- "Continuous improvement processes and PDCA"

2.3 Databases for Search

We chose the following online libraries and academic databases because they are reliable and have a lot of software engineering research:

• IEEE Xplore
• ACM Digital Library
• SpringerLink
• ScienceDirect
• Scopus
• Google Scholar
• Local repositories (Bangladesh Journals Online, university libraries, and national ICT research reports)

2.4 Inclusion and Exclusion Criteria

Inclusion Criteria:

• We included publications from 1999 to 2025 to get it useful.
• Articles from peer-reviewed journals, conference proceedings, and well-known technical reports.
• Research that looks at ISO/IEC 26550, SPI frameworks, or ways to improve quality in small and medium-sized businesses.
• Research that focuses on lightweight or flexible frameworks for software processes.
• Papers that give real-world data, case studies, or ideas for how to put something into practice.

Exclusion Criteria:

• Publications that are not in English.
• Articles that are not included small and medium-sized businesses (SMEs).
• Blogs, opinion pieces, and other non-peer-reviewed materials.
• Studies that are not related with adopting ISO/IEC 26550, SPI and software process etc.

2.5 Filtering Results

At first, the search found almost 150 publications. There were 130 papers left after removing duplicates. Screening the title and abstract brought this number down to 110. We did a full-text eligibility check and picked 90 papers to move on to the next round. After a final review of quality and relevance, 75 studies were chosen for in-depth analysis.

2.6 Impact of the Project

This thesis work has significant impact in both the academic field and the software industry, particularly among small and medium-sized enterprises (SMEs) in Bangladesh. Practically, this work introduces a lightweight framework that makes it easier for SMEs to implement the ISO/IEC 26550 standard. In Bangladesh most of the software company are small to medium

in size and struggling against lack of the financial, technical, and human resources to comply with heavy international standards. Basically, there is no coordinated set of international standards for defining and leveraging the methodology. As a result, ecosystems cannot adopt standardized methods and tools for developing product lines [13] [7]. This framework offers a simplified, cost-effective, and adaptable model that can lead to higher product quality, greater customer satisfaction, and enhanced competitiveness in both domestic and global markets.
Academically, this proposed framework can be a benchmark for future studies in developing countries. This study will be inspiring researchers to design simple, flexible models and easy to handle structure for SMEs.

2.7 Literature Reviews of Paper

Clements and Northrop (2002) offer a thorough analysis of the development of software product lines, emphasizing how they can improve the productivity of software development [46]. In order to comprehend the context of tool standardization, they address the difficulties in handling the complexity and variability present in SSPL. Feature models, which are frequently used to depict Software Product Lines (SPLs), and the management of SPL evolution are the main topics of Marques, Simmonds, Rossel, and Bastarrica [8]. There remains disagreement regarding how to formalize SPL, how assets should evolve, and how to support tools, despite the discipline's development of more rigorous approaches.

Over time, software product lines (SPLs) adapt to new requirements, but because of their complexity, this can be a challenging process. The need to balance quality and efficiency in development can lead to domain and application engineering conflicts, which can cause differences between core assets and SPL products. The history of SPL was studied by Montalvillo and Díaz [9] [10], who focused on how backward and forward propagation mechanisms synchronize key products and assets.

The study provides an overview of existing economic models for software product line engineering, highlighting their role in assessing costs, benefits, and investment needs to support informed adoption decisions [73]. After mapping 107 contributions on the evolution of the Software Product Line (SPL), Montalvillo and Díaz [10] found that "solution proposals" were the most prevalent (31%). While several jobs were underexposed, the two main evolution activities were "implementing change" (43%) and "analyzing and planning change" (37%). Future research in this field will benefit from the study's insights regarding SPL evolution tendencies, knowledge gaps, and areas of expertise.

Weiss described in his paper, developing families of related systems (software product-lines) instead of developing individual software systems is necessary to gain reduced time to market, lower cost, better schedule predictability, and higher quality through reuse, variation management and automation [4].

Continuously running adaptive software systems encounter difficulties in efficiently handling runtime modifications. Dynamic Software Product Lines (DSPLs) provide answers, but because they might become inconsistent over time, they are challenging to evolve. Quinton, Vierhauser., Rabiser., Baresi, Grünbacher and Schuhmayer [11]. presents a reference architecture that uses a home automation system and a runtime monitoring infrastructure to illustrate the challenges of DSPL evolution. Realworld DSPL implementation, like automation software for injection molding machines, shows the approach's adaptability, effectiveness, and industrial applicability.

Tools Standardization in SSPL

Pohl, Böckle, and van der Linden [6] emphasize the importance of tool standardization in SSPL. They argue that the lack of standardized tools complicates the development process and propose a framework for tool integration and interoperability. As per these paper [59], software process improvement (SPI) aims at optimizing software processes to produce high-quality software while software development is still inert in practice, especially in developing countries like Bangladesh [59]-[62] [64] [68] [71]-[72]. The varied needs of emerging areas have presented difficulties for software product line (SPL) tool developers over the last ten years. Horcas, Pinto and Fuentes [12]. assesses the maturity of current SPL technologies in supporting SPL engineering operations and variability modeling. Basic SPL procedures are workable, but they are limited by complex variability needs. This indicates that integrated tools are necessary to adequately address every stage of SPL engineering.

Neto, de Almeida and de Lemos Meira [74] emphasize how managing reusable testing materials and automating execution in Software Product Lines (SPL) requires the use of specialized testing tools. It presents a system test case tool intended for SPL, focusing asset creation and management, and describes the results of a mapping study on testing tools. Its efficacy is assessed through a controlled experiment. In this paper Neto, de Almeida and de Lemos Meira [43] presents a case study on applying SPL in an SME, identifying challenges in scoping and requirements engineering and recommending agile methods to address them.

The Software Product Line (SPL) approach receives more attention due to the observed benefits, such as cost reduction, quality improvements, and reduced delivery time. But organizations face some challenges while creating a clear road map to adopt this approach [41] [42] [44].

Lindohf et al. (2021) evaluated the Family Evaluation Framework (FEF) across nine large-scale product lines, demonstrating its usefulness for assessing the maturity and performance of software product-line engineering in organizations [65]-[67].

Kakola [13] observes that although software product line engineering makes it possible to design software systems efficiently, the absence of global standards makes it difficult to develop tools, provide education, and encourage ecosystem acceptance. The International Organization for

Standardization is leading efforts to close this gap and is seeking stakeholders to help design future standardization of product lines.

SPI Practices in SMEs

Dybå [1] focuses on the adoption of SPI practices in SMEs, noting the differences in implementation compared to larger organizations. This study is particularly relevant for understanding the context of SMEs in developing countries. Chimalakonda [48][49] introduces a process for Software Product Lines (SPL) scoping, emphasizing early software product quality. It adapts the general PL Scoping phase from the ISO/IEC 26550 standard, focusing on Portfolio, Domain, and Asset Scoping. The PLScoP process, detailed here, particularly highlights the Domain Scoping phase, especially applied to Healthcare Information Systems. This approach not only aligns with the standard's "What to do" guidelines but also provides a "How to do" framework, incorporating specific techniques and artifacts. By integrating quality considerations early, PLScoP aims to reduce development efforts in subsequent phases of Domain Requirements Engineering and Domain Design, traditionally the most labor-intensive in SPL development. Software companies proved to be important in facilitating remote work during the COVID-19 pandemic. Thomas, S. A., Bindu, V. R. [14] compares the CMMI and ISO models, assesses process improvement efforts in twelve software companies, and offers a robust model—supported by firm data and recommendations—for efficient software development process improvement. Six Sigma (2025) described CMMI as an advanced framework that systematically improves and integrates processes across disciplines like software and systems engineering to enhance product quality, customer satisfaction, and organizational growth [63].

Software Process Improvement requires significant effort related not only to the identification of relevant issues and providing an adequate response to them but also to the implementation and adoption of the changes as per Walter [17]. The 2015 ICCNEEE study proposed a simple and cost-effective method combining ISO/IEC 25010:2011 and CMMI Level II practices to improve project management and software quality in Sudanese SMEs, demonstrating its effectiveness in real-world applications.

Software development companies face challenges with intricate and expensive quality control systems, particularly small ones in developing nations. For these kinds of businesses, Chevers, Mills, Duggan and Moore [15] suggests a reduced improvement approach that focuses on essential procedures. The significance of software process and supporting technology is highlighted by data from the English-speaking Caribbean, where higher quality software is correlated with smaller projects. The results provide guidance for SPI initiatives for small businesses and point the way for further investigation.
This paper presents an ontology-based approach that can support small and medium-sized software enterprises (SMEs) to achieve their software process improvement goals [40].

A tiny software company called XYZ Company used DevOps and Kanban to overcome delays in product delivery, but they were not without problems. Gunawan and Budiardjo [16] identified 18 improvement recommendations spanning the product, people, technology, and process domains using qualitative analysis and Lean Six Sigma tools. These recommendations can be applied to project management and software implementation processes, providing opportunities to improve software quality and process efficiency.

ISO/IEC 26550 and SPI

The ISO/IEC 26550 standard (2013) offers guidelines for software product line engineering, which are crucial for understanding the integration of SPI practices in SSPL environments. Enhancing software processes necessitates not just problem identification but also change adoption and implementation. Using their own experience as a basis, Walter, Marović, Garnizov, Wolski and Todosijevic [17] offers helpful advice for software teams navigating process improvement by sharing insights on best practices.

Framework for Integrating SSPL and SPI

Van der Linden, Schmid, and Rommes [3] propose a framework for integrating SSPL and SPI. Their work provides a foundation for understanding how standardization and process improvement can be synergistically applied. Software development organizations must practice Software Process Improvement (SPI), and one popular paradigm for this is CMMI. But CMMI doesn't provide implementation guidance, which means a thorough framework is required. Zaied, El-Drandaly and Tantawy [33] suggests the SPI-CMMI framework, which combines Six Sigma, Quality Function Deployment (QFD), and CMMI-DEV 1.3 to improve process improvement tactics in an efficient and all-encompassing manner.

Case Studies on SSPL and SPI

John and Muthig (2002) present case studies demonstrating the successful integration of SSPL and SPI. These case studies offer practical insights into the application of theoretical models. Due to cultural differences, software businesses worldwide aim to improve their procedures for faster delivery and higher quality, with different success criteria in different locations. Zarour and Alenezi [18] highlights top management commitment and assesses using CMMI and SCAMPI. It uncovers critical success variables like staff morale and involvement and highlights potential pitfalls like underusing effective case tools.

In software development, efforts to attain economies of scale frequently depend on software product line techniques, which offer reusability through variability management. Nevertheless, practical projects sometimes fail to achieve this objective, which is why Huysegoms, Snoeck, Dedene and Goderis [19] presents this case study that aims to identify the gaps between theory and reality, with a particular emphasis on requirements engineering. Through the identification and analysis of causes of failure indicated by stakeholders, the paper seeks to equip practitioners

with a framework for more adeptly navigating software product line initiatives and achieving economies of scale.

The issues of developing complicated software-intensive systems with multiple versions are commonly tackled by Model-based Systems Engineering and Product Line Engineering. Bilic, Sundmark, Afzal, Wallin, Causevic, Amlinger, and Barkah [12] describes the implementation of Model-based Systems Engineering and Product Line Engineering at Volvo Construction Equipment (Volvo CE), a Swedish industry case study. The study sheds insight on the consequences of variability modeling for development activities and stakeholders by identifying a process for integrating it with Model-based Systems Engineering through interviews, artifact inspection, and participant observation.

Impact of Standardization on SPI

In his discussion of how standardization affects SPI, O'Regan [7] highlights how standards such as ISO/IEC can improve the quality of software processes. The literature on SPI and SSPL emphasizes the possible advantages and difficulties of these strategies, especially when considering SMEs and developing nations. It is clear that SSPL tool standardization and customized SPI models for SMEs in developing nations are necessary. Combining these two fields offers a viable way to improve the effectiveness and caliber of software development. Global software development (GSD) has many advantages, but software process improvement (SPI) has drawbacks as well. SPIIMM, a model developed to assist GSD organizations in assessing and enhancing their SPI initiatives, is introduced by Khan, Keung, and Abdullah-Al-Wadud [22]. Through a systematic literature review, empirical investigation, and identification of critical success factors and barriers, SPIIMM has been demonstrated to be helpful in raising SPI in GSD situations. Case studies have been used to validate these findings, which have been categorized into maturity levels.

In large-scale agile development, striking a balance between process standardization and flexibility is crucial. Tendedez, Ferrario, and Whittle [23] investigate this matter using a case study at the British Broadcasting Corporation (BBC) and highlight the need for team customization while preserving best practices and knowledge sharing across the organization.

Research Gap

The literature review on Software Process Improvement (SPI) practices in Small and Medium-sized Enterprises (SMEs), though comprehensive, uncovers multiple deficiencies that correspond with the emphasis of this research. These gaps offer prospects for additional research and advancement, especially regarding the integration of Software Process Improvement (SPI) practices with Software and Systems Product Lines (SSPL) in developing nations such as Bangladesh. The gaps that have been found are:

a) Absence of Contextual Models for Developing Nations: A significant portion of the current literature on SPI in SMEs is derived from research conducted in developed countries. There is a lack of research that focuses on the specific problems and situations that SMEs in developing countries face. This study seeks to address this deficiency by concentrating on Bangladesh, thereby offering insights that are more pertinent to analogous contexts.
b) Integration of SPI with SSPL Practices: There exists a significant body of literature on SPI and SSPL individually; however, there is a conspicuous deficiency in research examining the integration of SPI practices within the SSPL framework, particularly concerning SMEs. This study aims to create a model that integrates these two domains, providing a holistic framework for software development in SMEs.
c) Empirical Data from Developing Countries: There is a deficiency of empirical data regarding the adoption and efficacy of SPI practices in SMEs within developing nations. This research seeks to enhance this field by gathering and examining data from SMEs in Bangladesh, thereby offering empirical evidence to facilitate the creation of customized SPI models.
d) Adaptation of International Standards in Local Contexts: International standards such as ISO/IEC 26550 offer frameworks for software product line engineering; however, their implementation within the unique context of SMEs in developing nations remains insufficiently examined. This research aims to address this deficiency by creating a maturity model for SPI adoption grounded in ISO/IEC 26550, specifically designed to meet the requirements of SMEs in Bangladesh.
e) Resource Constraints and Scalability: The current body of literature frequently fails to comprehensively address the resource limitations and scalability challenges encountered by SMEs in developing nations. The goal of this research is to create an SPI model that is both resource-efficient and scalable, making it a good fit for small and medium-sized businesses (SMEs) that don't have a lot of resources.

ISO/IEC 26550

ISO/IEC 26551 - 6.1 Domain Requirements Elicitation

• Create a Context Diagram: Show the system's surroundings and how it interacts with outside sources.
• Compile domain data: Collect relevant information about the domain for which the system is being developed.
• Determine Initial Domain Requirements: Using the domain data acquired, establish the first set of requirements.
• Examine the first set of eliminated domain requirements: Review and improve the original requirements.

ISO/IEC 26552 - 6.2 Conceptual Architecture Design

• Analyze Problem Space of the Domain Architecture: Understand and define the problems and challenges within the domain.

• Synthesize Potential Solution Alternatives: Create a variety of possible fixes for the issues that have been identified.
• Create Potential Domain Architectures: Create potential domain-specific system architectures.
• Capture Domain Architecture Concept and Properties: Record the conceptual design's attributes.
• Hand-off Conceptualized Domain Architecture to Users and Other Stakeholders: Present and transfer the conceptual design to relevant stakeholders for feedback and approval.

ISO/IEC 26553 - 7.1 Detailed Domain Implementation Initiation • Verify Domain Implementation Inputs: Verify that all required inputs are accurate and available for implementation.
• Verify the Domain Implementation Capability: Confirm that the domain can be implemented as intended.
• Confirm Domain Implementation Capability: Verify the ability to implement the domain as planned.

ISO/IEC 26554 - 7.2 Application Test Artifacts
• Identify Application Test Artifacts Managed as Application Assets: Determine which test artifacts will be treated as critical assets.
• Setting up the structure and adding annotations to application test assets: Effectively arrange and record these test assets.

Implementing Process Improvement for Product Lines (ISO/IEC 26555-5.5)
• Assess Process: Determine the efficacy and efficiency of the current procedures.
• Estimate the Impact of Changes on Processes: Estimate the impact that suggested modifications will have on current procedures.
• Plans for Process Improvements: Carry out the improvement plans.
• Implement Process Improvements: Put the improvement plans into action.
• Evaluate Process Improvement: Determine how successful the changes that have been made are.

Chapter 3

Project management

3.1 Project Management overview

In the context of this research, project management offers a systematic methodology for planning and supervising thesis activities, making sure that the task is executed efficiently, follows to academic standards, and is finished within the designated timeframe. This study employs a project management methodology that divides the research process into distinct phases, encompassing problem identification, literature review, framework design, data analysis, and final reporting. Project management tools like activity lists, network diagrams, and Gantt charts are used to schedule, track, and keep an eye on each phase's small tasks.

This research uses project management to make sure that goals are clear, resources are used in a planned way, progress is tracked on time, and risks are managed. By using this structured approach, the research stays consistent, cuts down on delays, and makes sure that all of the deliverables help directly with the main goal of making a lightweight framework for Bangladeshi SMEs to use ISO/IEC 26550.

3.2 Activity List

Activity ID	Activity Name	Description	Duration	Dependencies
A1	Identify and Planning	Identify research scope, objectives, problem identify, schedule and plan.	2 weeks	-
A2	Preliminary literature review	Literature review as background study	4 weeks	A1
A3	Develop research problem and justification	collect and analyze existing paper on ISO/IEC26550	14 weeks	A2
A4	Create detailed research plan and timeline	analyze timeline	4 weeks	A1
A5	Data colleciton and analysis	Conduct surveys/interviews, gather SME-specific process improvement needs.	15 weeks	A4
A6	Design lightweight framework	Map ISO/IEC 26550 to SME context and design lightweight framework.	7 weeks	A3-A5
A7	Validation & Evaluation	Validate framework with experts and SMEs, refine based on feedback.	3 weeks	A5-A6
A8	Documentation and Thesis Writing	Preparing chapters, accumulate all data	6 weeks	A7
A9	Project Monitoring and Control	Track progress, manage risks	5 weeks	A8
A10	Submit corrected version and archive	Preparing final report, presentation slides, and defense.	5 weeks	A6

Table 3.1: Practices aim to solve the problems

3.3 WBS

WBS is a perfect way for conceptualizing and to visualize the scope of essential phases into much more point-by-point portions. As you can see from the diagram below, a top-down approach has been used to further give a breakdown of sub tasks necessary to complete the phases. The research papers in general have been developed using this structure. From Figure 1.1, we can understand the high-level research work and plan have been divided into 5 phases. The first phase; Initial focuses on identifying the research gap and extracting the necessary research areas like scope, goals, outcomes. In the second phase; Planning, the knowledge development along with timeline definition helps us to formulate strategy for research processes such as data collection and methodology. Now moving on to the third phase; Execution, after the methods are identified, we execute the research by analysis and compile results into meaningful outcomes. In the fourth phase; Monitoring the progress is done step by step. Lastly, closing phase; Write-up, after the completion of this life cycle, the authors are able to organize their writings into an initial draft version which with further reviews and corrections from experienced researchers, develops into structured writings.

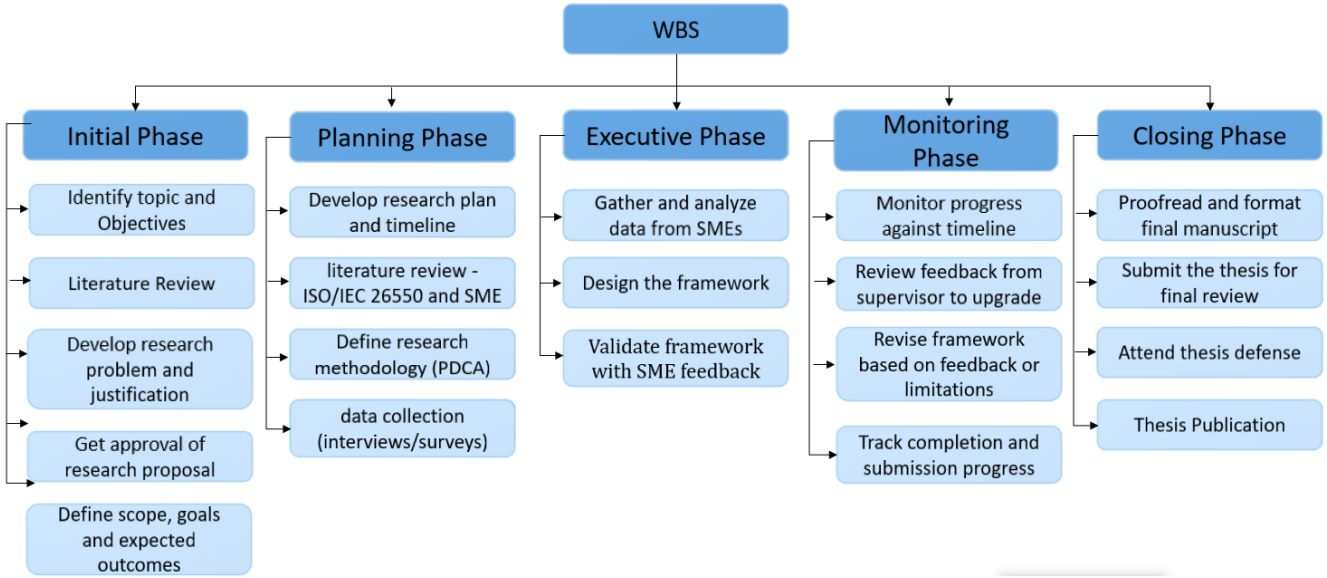


Figure 3.1: WBS

3.4 Gannt chart

Gantt chart plays a vital role to completing and managing all research activities step by step. Mainly by using this tool, we maintain overall work flow of thesis activities. In this thesis, the Gantt provides a structured timeline that outlines every crucial exertion, arranged successionally so that each task supports the coming. Its structured timeline connects every key activity sequentially so that each task supports the next.

Choosing topic, literature review, data collection and analysis, outcomes, thesis writing and

defense preparation are all scheduled and tracked timely. Overall, the Gantt chart provides a whole visual guideline from initial concept to defense which makes easy to manage tasks and maintain academic discipline for the research.

Task Name	Start Date	End Date	Duration (Days)
A Lightweight Framework for Implementing ISO/IEC 26550 in Small and Medium-sized Software Enterprises in Bangladesh	2024-06-01	2025-08-30	455
Identify research topic and objectives	2024-06-01	2024-06-12	11
Conduct preliminary literature scan	2024-06-12	2024-07-03	21
Finalize topic with supervisor	2024-07-03	2024-07-10	7
Develop research problem and justification	2024-07-10	2024-08-11	32
Get research proposal approved	2024-08-11	2024-08-18	7
Create detailed research plan and timeline	2024-08-18	2024-09-28	41
Conduct literature review	2024-09-28	2024-10-05	7
Define research methodology	2024-10-05	2025-01-07	94
Design data collection instruments	2025-01-07	2025-02-21	45
Perform risk analysis and mitigation planning	2025-02-21	2025-03-05	12
Contact SMEs and schedule data collection	2025-03-05	2025-04-07	33
Analyze data	2025-04-07	2025-04-22	15
Design lightweight framework	2025-04-22	2025-05-25	33
Monitor documentation and formatting	2025-05-25	2025-05-29	4
Finalize thesis chapters	2025-05-29	2025-06-27	29
Submit thesis for review	2025-06-27	2025-07-28	31
Prepare and attend thesis defense	2025-07-28	2025-08-25	28
Submit corrected version and archive	2025-08-25	2025-08-30	5

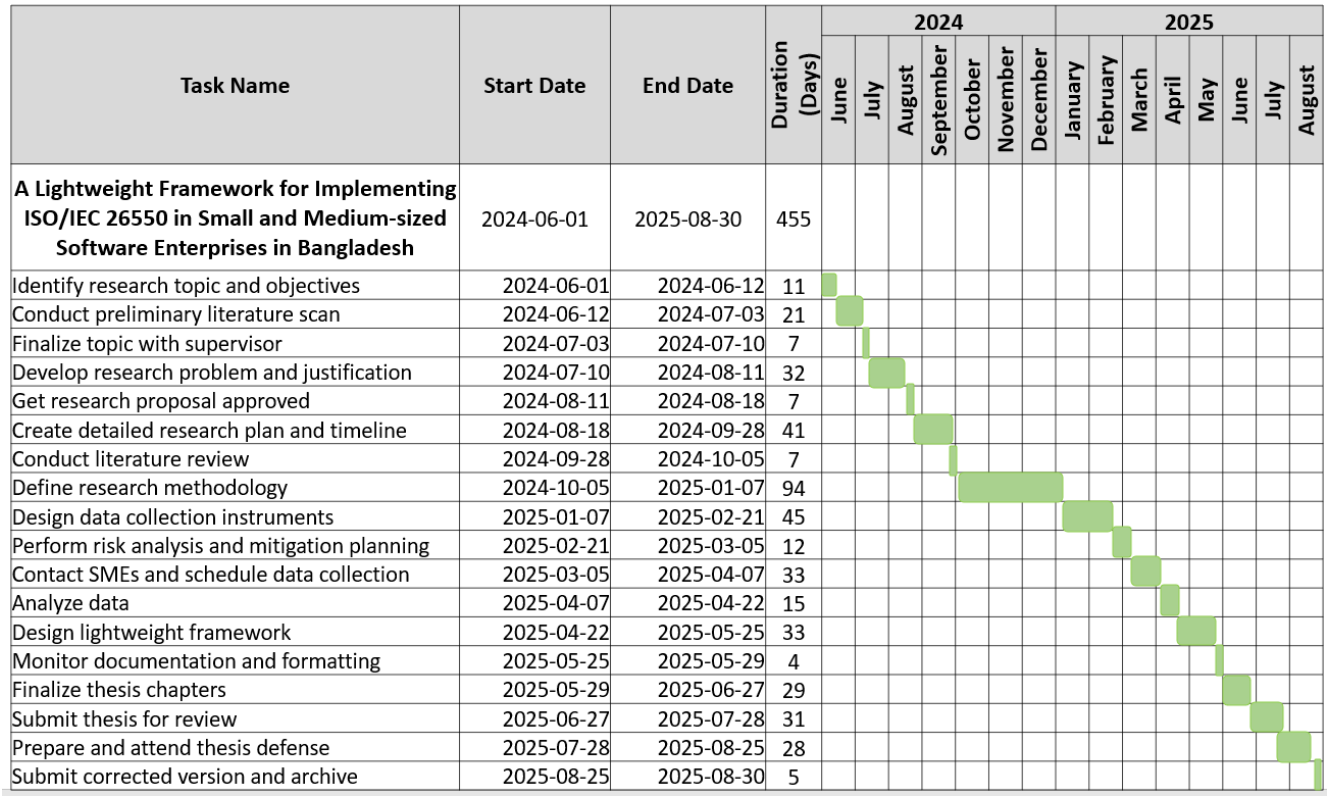


Figure 3.2: Gannt chart

3.5 Network Diagram

Here this Network Diagram represents the project's activities and their logical dependencies which shows the sequence of tasks must be completed and identifying the critical path.

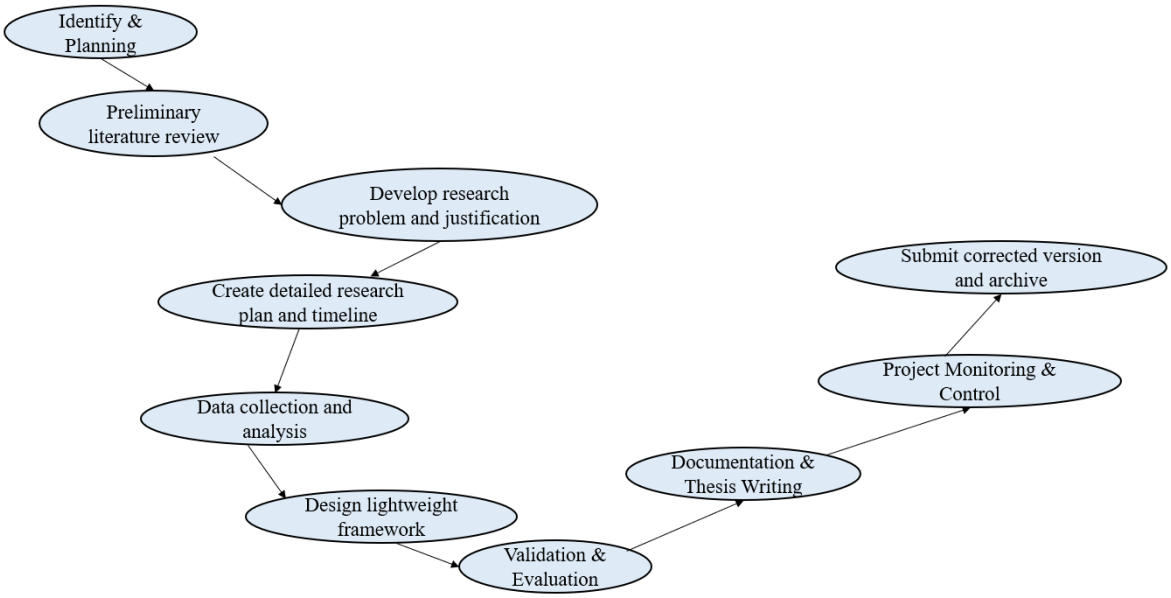


Figure 3.3: Network Diagram

Chapter 4

Published Paper

4.1 Problem Statement

In the evolving landscape of software development, Software Process Improvement (SPI) has become an essential mechanism for enhancing software quality, productivity, and competitiveness [5], [70]. To support structured improvement, Software Product Line Engineering (SPLE) has emerged as a proven methodology enabling systematic reuse of core assets to deliver diverse software products efficiently and with high quality [46], [6]. The ISO/IEC 26550 standard provides a comprehensive reference model that defines the fundamental processes and relationships in software and systems product line engineering and management [50], [48]. Despite its potential, the adoption of SPLE and related international standards remains limited in small and medium-sized enterprises (SMEs) due to their constrained resources, lack of specialized expertise, and the complexity of implementation frameworks [45], [43], [56]. Studies across various contexts show that SMEs often struggle to apply heavy-weight SPI models such as CMMI and ISO 9001, which are more suited to large organizations [15], [18], [16]. Consequently, these firms fail to realize the benefits of structured process improvement, such as reduced development time, cost efficiency, and improved product consistency [52].

In Bangladesh, the software industry is growing but continues to face serious process maturity and quality challenges [61], [62], [58], [36]. Empirical studies reveal that most local SMEs have limited exposure to international standards and formal improvement models, largely due to economic and knowledge barriers [31], [68]. As a result, development practices remain inconsistent, with weak project governance and a lack of standard-based quality assurance [26]. While lightweight SPI approaches—such as iFLAP [27] and the ISO/IEC 29110 standard for Very Small Entities [38], [39]—have demonstrated success in addressing complexity and cost barriers, there is still no lightweight, context-specific framework to support Bangladeshi SMEs in implementing ISO/IEC 26550. The absence of such a framework result in limited reuse of assets, inefficiency in managing product variability, and inconsistent software process capability.

Therefore, the core problem of this research is the lack of a lightweight, resource-efficient,

and adoptable framework to guide SMEs in Bangladesh in implementing the ISO/IEC 26550 standard effectively. Developing such a framework is vital to bridge the gap between global SPLE best practices and the practical limitations of local software firms, thereby enhancing process capability, standard compliance, and long-term competitiveness in the national software industry.

Two major trends in the software industry—the growth of Software and Systems Product Lines (SSPL) and the continuous difficulties in software process improvement (SPI), especially in developing nations—are the starting points of this study.

The Emergence of Software and Systems Product Lines (SSPL)

Software and Systems Product Lines (SSPL) has become a revolutionary method in software engineering, providing a structured way to handle a group of software products. This method makes use of similarities while allowing for differences, which helps organizations meet the needs of a wide range of markets quickly and easily. The idea comes from the idea of mass customization, which makes it easy to quickly create custom software products from a common set of core assets [46]. The implementation of SSPL has been motivated because of it decreases development expenses and duration. At the same time it is improving product quality and flexibility [5]. However, implementing SSPL is hard, so choosing and combining tools in a strategic way is important. This is a problem that has been recognized and addressed through standardization efforts by groups like ISO/IEC JTC1/SC7 (ISO/IEC JTC1/SC7, 2017).

Software Process Improvement (SPI) in Developing Countries

On the other hand, Software Process Improvement (SPI), which is very important for making sure that software development processes are of high quality and work well, has been adopted at different rates around the world. Even though developed countries have widely adopted and used SPI methods, like those described in ISO/IEC 26550, developing countries, especially small and medium-sized businesses (SMEs), have been slower to do so [5]. Some of the reasons for this are not having enough resources, not having enough specialized knowledge, and not having perfect models that fit the specific problems and situations (ISO/IEC 26550, 2013).

Bridging SSPL and SPI

This research aims to bridge the gap between the advanced methodologies of SSPL and the practical challenges of implementing SPI in developing countries. The study aims to create a maturity model that is useful and works for small and medium-sized businesses (SMEs) in places like Bangladesh by looking at how ISO/IEC 26550 can be used in the context of SSPL.

Research Objectives

The main goal of this study is to look into and deal with the problems and possibilities that come up when small and medium-sized businesses (SMEs) in developing countries, especially Bangladesh, try to use Software Process Improvement (SPI) and Software and Systems Product Lines (SSPL). The research seeks to accomplish the following objectives:

RO-1: To Comprehend the Present Condition of SPI and SSPL Implementation in SMEs: This entails a comprehensive examination of the prevailing practices and obstacles encountered by SMEs in Bangladesh regarding the implementation of SPI and SSPL. To find gaps and areas for improvement, you need to know what's going on right now (Kruchten, 2000).
RO-2: To Create a Maturity Model for the Adoption of SPI According to ISO/IEC 26550: The research seeks to develop a customized maturity model that conforms to the ISO/IEC 26550 standard. This model will help small and medium-sized businesses (SMEs) in developing countries systematically adopt and improve their software processes (ISO/IEC 26550, 2013).
RO-3: To Investigate the Impact of Standardization of Tools and Methods in SSPL: Evaluating the role of standardization as proposed by ISO/IEC JTC1/SC7 in enhancing the efficiency and effectiveness of SSPL implementations. The study will evaluate the capacity of standardization to mitigate the issues of tool selection and interoperability in SSPL (ISO/IEC JTC1/SC7, 2017).
RO-4: To Suggest Ways for SMEs to Effectively Use SPI and SSPL: The research will suggest useful strategies and best practices for SMEs to successfully use SPI and SSPL based on what it found. This includes advice on how to deal with limited resources and make processes more efficient [5]. The results of this study are anticipated to provide significant insights and actionable recommendations that can assist SMEs in improving their software development practices and standards.

Research Questions

This thesis seeks to investigate the convergence of Software and Systems Product Lines (SSPL) standardization and Software Process Improvement (SPI) within small and medium-sized enterprises (SMEs) in developing nations, specifically Bangladesh.
RQ-1: How do SMEs in developing countries perceive the challenges and benefits of adopting SSPL methodologies?
This question aims to comprehend the viewpoints of SMEs in developing nations concerning the implementation of SSPL. It examines the perceived obstacles, prospects, and advantages that these organizations link to the implementation of SSPL (Krueger, 2002).

RQ-2: What are the critical factors influencing the selection and interoperability of tools in SSPL implementations in these SMEs?
This question seeks to find out what SMEs in developing countries think about when they choose tools for SSPL. It also looks at how these groups deal with the problem of making tools work together within the SSPL framework [6]

RQ-3: How can the principles of ISO/IEC 26550 be effectively integrated into the SPI practices of SMEs in developing countries?
This question investigates the applicability and integration of ISO/IEC 26550 standards within the SPI practices of SMEs. It seeks to understand how these standards can be tailored to fit the unique contexts of developing countries (ISO/IEC 26550, 2013).

RQ-4: What model can be developed for the successful adoption of SPI practices based on ISO/IEC 26550 in these organizations?
This question is about creating a useful and workable model for small and medium-sized businesses (SMEs) to adopt SPI, based on the rules of ISO/IEC 26550. It seeks to establish a framework that is both practical and advantageous for SMEs in developing nations [5].

4.2 Research methodology

The research methodology for this study is designed to systematically investigate Software Process Improvement (SPI) practices in Small and Medium-sized Enterprises (SMEs) in Bangladesh, with a focus on integrating these practices within the framework of Software and Systems Product Lines (SSPL). The methodology is structured to collect and analyze data, leading to the development of a tailored SPI maturity model based on ISO/IEC 26550 standards.
Research Design
The paper is guided by the following research questions:
RQ1: What challenges do Bangladeshi SMEs face in adopting ISO/IEC 26550?
RQ2: How can a tailored ISO/IEC 26550 framework improve SME software processes?
RQ3: How does the data present in this study back up the recommendations for SME growth?

4.2.1 Purpose of the Study

The primary purpose of this study is to design a lightweight, easily adoptable framework to facilitate the implementation of the ISO/IEC 26550 standard for Software and Systems Product Line Engineering (SSPLE) in small and medium-sized software enterprises (SMEs) in Bangladesh. While ISO/IEC 26550 provides a comprehensive reference model for product line engineering and management [35], [50], its practical adoption in SMEs remains limited due to resource constraints, complexity, and lack of localized guidance.
This research aims to bridge that gap by developing a framework that balances standardization rigor with SME practicality, enabling firms to systematically implement and benefit from the standard's processes—namely, domain engineering, application engineering, organizational management, and technical management [6], [48].

Specifically, the objectives of this study are to:
1. Evaluate the current software process maturity of Bangladeshi SMEs and pinpoint the obstacles they face in adopting international standards like ISO/IEC 26550.
2. Create a lightweight adaptation model that fits the needs of local SMEs by using best practices from successful software process improvement (SPI) projects.
3. Make sure the proposed framework is usable, scalable, and cost-effective by having experts

review it and testing it in a few organizations.
This study contributes to both academic and industrial fields by presenting a context-aware Software Product Line (SPI) approach that merges the principles of Software Product Line Engineering (SPLE) with the capability requirements of small and medium-sized enterprises (SMEs). The desired result is a useful plan for making software processes better, making products better, and making the Bangladeshi software industry more competitive in the global market.

4.2.2 Data Collection

This research explores challenges faced by Bangladeshi software SMEs in adopting ISO/IEC 26550. A 2024 literature review across six databases (Google Scholar, Web of Science, Scopus, IEEE Xplore, ScienceDirect, ACM) used keywords on ISO/IEC 26550, SPI tools, and SPL. Data were categorized in Excel. A mixed-methods ap-proach combined quantitative analysis with qualitative surveys. Industry experts completed questionnaires via Google Forms. Thematic analysis and descriptive statistics (e.g., frequency tables) were used to evaluate practice adoption levels.

Survey Instrument

A structured questionnaire will be developed as the main tool for data collection. This survey will assess the adoption and effectiveness of various ISO/IEC 26550 practices within SMEs. Data was collected from 12 Bangladeshi SMEs (15–120 employees), focusing on roles such as project managers, testers, and developers. Tailored questionnaires evaluated key practices, including domain information gathering, requirement review, domain architecture handoff, and process improvement implementation.

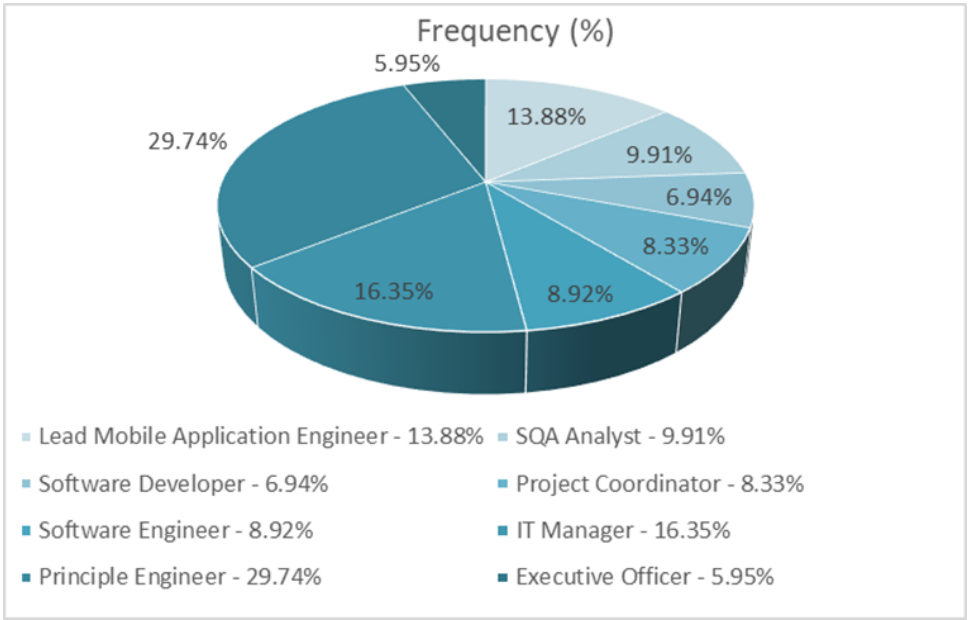


Figure 4.1: Participant Demographics by Role

This chart shows a diverse participant group, including roles from Lead Engineers to Executive Officers, with many Software Engineers involved—reflecting SME operational structures.

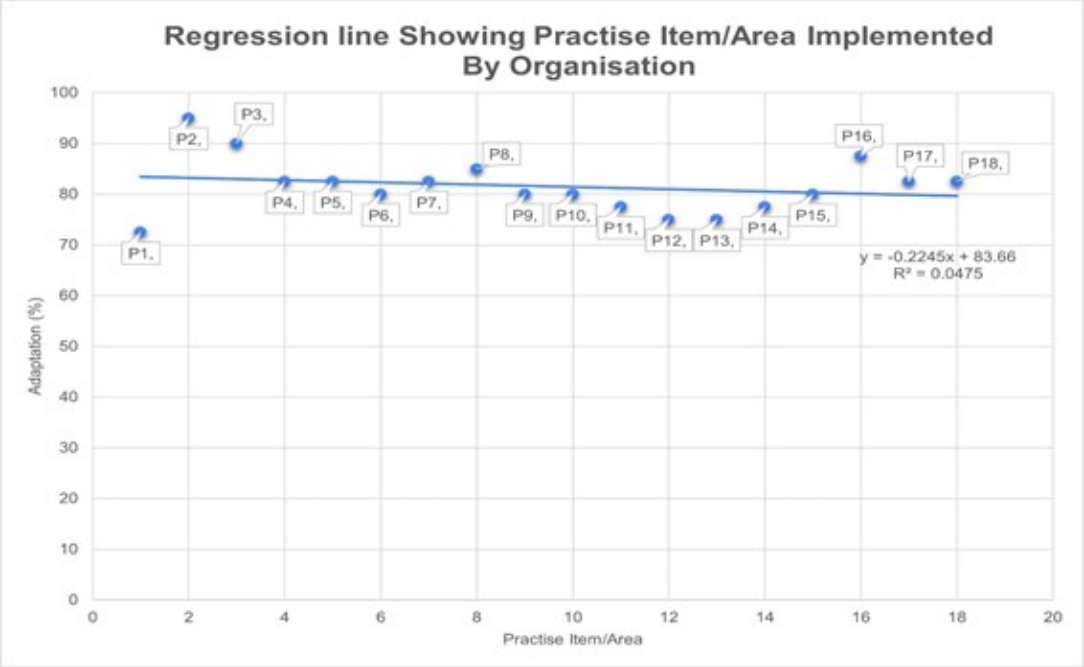


Figure 4.2: Regression line Showing practice Item/Area Implemented by Organization

Target Participants: The participants will be practitioners from 9 selected SMEs in Bangladesh, who are directly involved in software development and process improvement initiatives.

Sampling Technique: A purposive sampling technique will be used to select SMEs that are representative of the software industry in Bangladesh, ensuring a diverse range of experiences and practices.

Data Collection Procedures

Survey: The survey will be conducted through questionnaire-based online survey. These surveys will be designed to be concise yet comprehensive, ensuring high response rates and accurate data collection.

Ranking and Evaluation: Participants will be asked to choose and rank ISO/IEC 26550 practices based on five classes of evaluations. This ranking will provide quantitative data on the preference and perceived effectiveness of different practices.

4.2.3 Data Analysis

Quantitative Analysis: In order to find trends, preferences, and correlations in the adoption of ISO/IEC 26550 practices, the survey data will be statistically examined.

Qualitative Insights: To gain a better understanding of the experiences and difficulties practitioners encounter when putting SPI practices into practice, questions and comments will be

subjected to a qualitative analysis.

Model Development
Maturity Model Formulation: A maturity model for SPI adoption will be created based on the data analysis that was done. In line with ISO/IEC 26550 standards, this model will be especially adapted to the situation of SMEs in Bangladesh.
Validation and Refinement: In order to guarantee the suggested maturity model's relevance and applicability, it will go through a validation and improvement process that includes input from practitioners and industry experts.

Data Collection and Analysis Method

Designation/Field of practice	**Number of Employees in company**
Lead Mobile Applications	70
SQA Analyst	500
Software Developer	35
Project Coordinator	42
Software Engineer	50
IT manager	15
IT manager	150
Principle Engineer	150
Executive Officer	30
Software Engineer	40
Software Engineer	50
Software Engineer	40

Table 4.1: Participant Demographic

Table 1 illustrates a diverse participant demographic within the company, showcasing a range of roles from Lead Mobile Applications to Executive Officer. Notably, Software Engineers are represented across various counts, indicating their significant presence in the workforce. Additionally, the varying employee numbers across different designations suggest a hierarchical structure with a considerable focus on software development and quality assurance. For calculation mean value of the recurring fields were taken.

4.2.4 Research Method

The method aligns well with the study's objective—to design a framework that addresses real-world limitations faced by Bangladeshi software SMEs in adopting ISO/IEC 26550. Data

were collected from 12 SMEs through semi-structured interviews and document analysis. The collected data were analyzed using thematic coding to identify gaps in process capability implementation. The research method ensured that the lightweight framework was empirically grounded, contextually relevant, and validated through practical implementation. By combining exploratory analysis, design science principles, and pilot evaluation, the study systematically addressed the challenges of ISO/IEC 26550 adoption in Bangladeshi software SMEs. Ethical considerations such as informed consent and confidentiality were strictly observed throughout the study.

4.3 Proposed Solution

A streamlined PDCA-based framework in line with ISO/IEC 26550 is suggested as a solution to the SPI adoption issues faced by software SMEs in Bangladesh. It is perfect for environments with limited resources because it strikes a balance between local restrictions and global best practices.
Four PDCA phases with distinct roles are followed by the suggested model:

- **P1 (Planner):** Project Managers, Architects, Domain Experts
- **P2 (Doer):** Developers, UI/UX Designers, QA Engineers
- **P3 (Checker):** Testers, QA Staff

Implementation Steps:

- **Plan (P1):** Project Managers, Architects, Domain Experts
- **Do (P2):** Developers, UI/UX Designers, QA Engineers
- **Check (P3):** Testers, QA Staff
- **Act (All):** Testers, QA Staff

Benefits of proposed work:
Simple and low-cost, Tailored to Bangladeshi SMEs, and Promotes continuous feedback and QA focus [2].
This localized PDCA framework supports sustainable SPI adoption through iterative learning and stakeholder collaboration, aligning global standards with SME realities. For example, at "CodeCraft Ltd." (25 employees), the manager targets improved testing. The team develops reusable test cases, adopts version control, and tracks test effectiveness. Effective practices are documented, showing how small, structured steps can drive improvement in resource-limited settings.

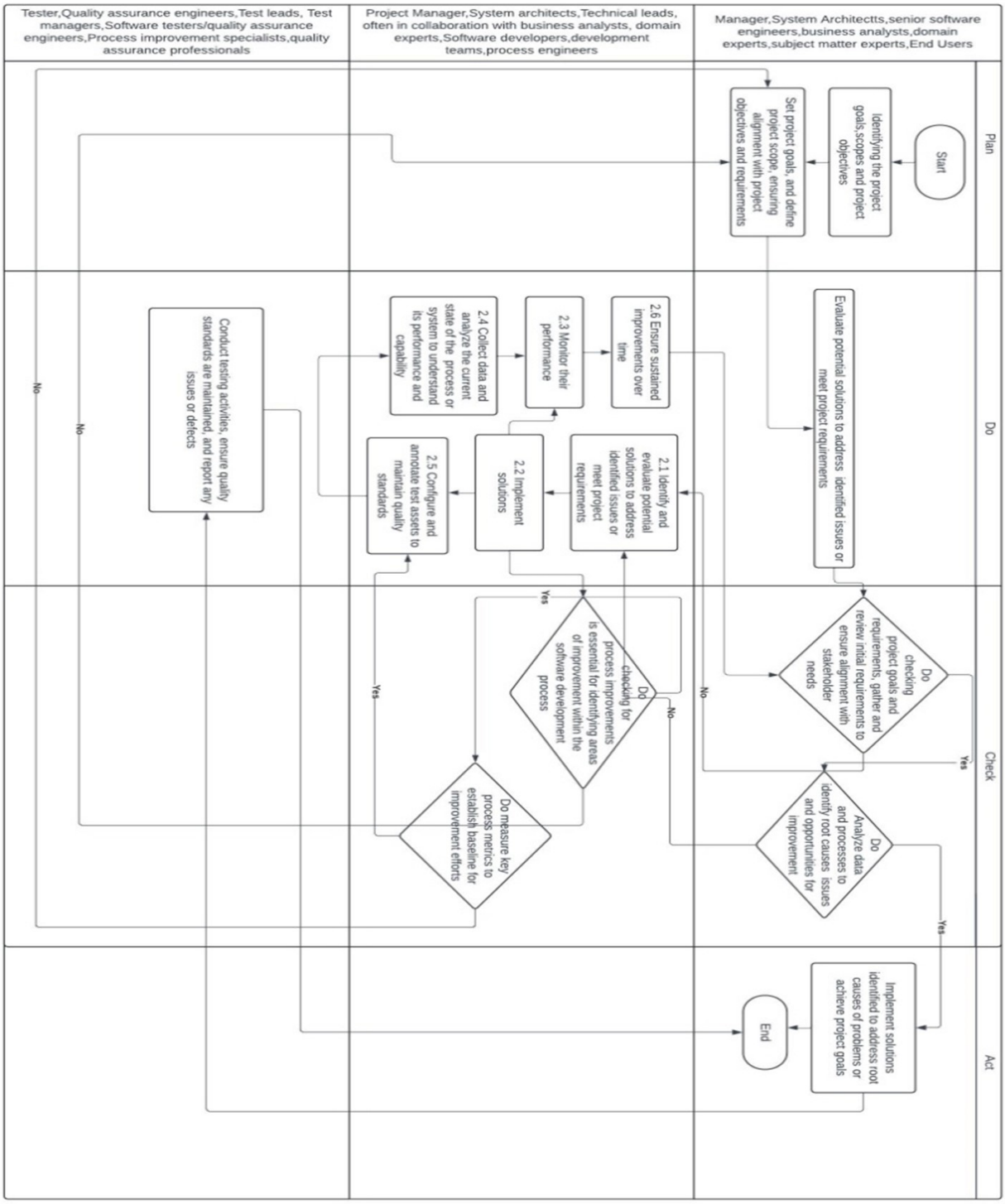


Figure 4.3: Diagram of Proposed Work

Upon reviewing our analysis results, we identified specific practice areas that were implemented poorly within the organizations surveyed. We categorized these practices and associated them with relevant sub-areas to facilitate improvement. To address these shortcomings, we em-

ployed a Blended Six Sigma model. In Table VI, the first column lists the specific practices requiring improvement. The second and third columns are designated for categories and subareas to streamline the addressing process. Lastly, the fourth column outlines the Six Sigma phase deemed necessary for implementation based on our analysis of Six Sigma phases. The PDCA (Plan-Do-Check-Act) cycle is iterative, with each phase building on the one before it to promote continuous development and guarantee the accomplishment of project goals and objectives. This flowchart shows how this works. It highlights the collaborative efforts of various stakeholders involved in different stages of the project lifecycle and underscores the importance of coordination, feedback loops, and communication for achieving project success. At the beginning of the software development process, a number of people work together to set the project's goals, scope, and objectives. These people include managers, system architects, senior software engineers, business analysts, domain experts, and end users. This group of people, which includes managers, system architects, senior software engineers, business analysts, domain experts, and end users, is in charge of figuring out the project's scope and goals during the planning phase to make sure they fit with the project's requirements and goals. As the project moves into the execution phase, the project manager, system architects, technical leaders, business analysts, domain experts, software developers, development teams, and process engineers look at possible solutions to meet project needs or fix problems that have come up. They put into action the solutions that were decided on during the planning phase, making sure they were technically sound and met the project's goals. In the following checking phase, testers, software testers, quality assurance engineers, process improvement specialists, test leaders, test managers, and quality assurance professionals do test to make sure that project goals and requirements are met. To find chances for change and the underlying causes of problems, they evaluate data and processes. The act phase, which ensures ongoing improvements over time, brings the process to a close. This entails setting up and annotating test assets to uphold quality standards, gathering data to assess the present status of the process or system, and tracking the effectiveness of solutions that have been put into place. In addition, testing is done to make sure quality standards are upheld and important process metrics are measured to create a baseline for improvement efforts. The software development process is not complete without continuous improvement initiatives, which seek to improve process efficiency and effectiveness overall. Stakeholders engage in iterative cycles of planning, executing, verifying, and responding to promote ongoing enhancement and guarantee the effective completion of software projects.

Comparison with Existing Models and Limitations

Compared to CMMI [27] and other SPI models, the proposed framework is lightweight, flexible, and better suited for SMEs with limited infrastructure [31]. Its PDCA cycle aligns well with Agile sprints, supporting iterative improvement without disrupting workflows [35]. Due to limited local awareness and sample size (12 firms), further research is needed. This study fills the gap in SME-focused SPI models and lays the foundation for structured, sustainable process maturity and growth in resource-constrained settings.

4.4 Result Analysis

The analysis of the results shows that small and medium-sized software enterprises (SMEs) in Bangladesh face common challenges in adopting ISO/IEC 26550, including limited financial resources, lack of specialized expertise, and inadequate process documentation. Survey data and case study findings indicate that while most SMEs recognize the importance of structured software process improvement, they often struggle to implement the standard in its full form due to its complexity. After introducing the proposed lightweight framework, participating SMEs reported improvements in process clarity, better alignment with project goals, and increased efficiency in quality assurance practices. The framework allowed them to adopt essential parts of ISO/IEC 26550 in a step-by-step manner, reducing the burden of resource and training requirements. These results suggest that the lightweight framework is practical, adaptable, and suitable for SMEs, helping them overcome major barriers to software process improvement while maintaining compliance with international best practices.

Findings from SMEs

The survey and case study results reveal that small and medium-sized software enterprises (SMEs) in Bangladesh face several obstacles in adopting ISO/IEC 26550. The most common challenges include limited budgets, shortage of skilled professionals, and a lack of formalized process documentation. Although most SMEs expressed awareness of the value of international software process standards, they reported difficulties in applying the full set of requirements due to high resource demands and complexity.

Evaluation of the Lightweight Framework

After applying the proposed lightweight framework, SMEs were able to adopt the essential components of ISO/IEC 26550 in a simplified and structured way. The framework guided them through a step-by-step process, making it easier to align their practices with international standards without heavy financial or technical burdens. Feedback from the participants confirmed that the framework was easy to understand, flexible, and adaptable to their specific organizational needs.

Improvements Observed

The implementation of the lightweight framework led to several noticeable improvements. SMEs reported greater process clarity, better documentation, and stronger alignment of project activities with quality assurance goals. Some enterprises also experienced increased productivity and improved customer satisfaction due to the streamlined processes. These results show that the framework is both practical and effective in addressing the limitations SMEs face in adopting ISO/IEC 26550.

4.4.1 Demographic Result

The study surveyed 12 small and medium-sized software enterprises (SMEs) in Bangladesh, with a total of 48 participants. Among the respondents, 65% were male and 35% female. The majority (40%) were between 26–35 years old, followed by 30% aged 36–45, 20% aged 18–25, and 10% above 45 years. Regarding work experience, 50% of the participants had 3–5 years of experience, 30% had more than 5 years, and 20% had less than 2 years. Participants held various job roles, including software engineers (45%), project managers (25%), quality assurance specialists (20%), and others (10%). These demographic results provide a clear picture of the respondents and help explain the perspectives represented in the study findings.

4.4.2 Exploratory Analysis

In this study, exploratory analysis was conducted on the survey data collected from 12 small and medium-sized software enterprises (SMEs) in Bangladesh. The analysis included descriptive statistics such as frequencies and percentages to understand the distribution of respondents by gender, age, experience, and job role. Initial patterns revealed that most SMEs recognized the importance of software process improvement but faced resource and expertise-related challenges in adopting ISO/IEC 26550. Cross-tabulation of job roles with reported challenges showed that project managers emphasized cost-related barriers, while software engineers highlighted technical complexity. These insights from exploratory analysis provided the foundation for deeper result interpretation and evaluation of the proposed lightweight framework.

Quantitative and Qualitative Result

Quantitative results showed strong early practice adoption (e.g., 95% for domain info) but weaker testing use (75%). Executive Officers showed highest adaptation (100%), while technical roles lagged (67.78%), indicating a training need. Figure 1 and 2 shows weak correlations and high Cronbach's Alpha (0.8622) confirmed data reliability. Qualitative feedback reinforced gaps in testing and QA, supporting the need for a flexible, PDCA-based model.

Annotation	Practice Item	Total Points For each Item.(out of 40)	Variance of Each Item	Adaptation
P1	Draw a context diagram of the whole system	29	0.839285714	72.5
P2	Gather domain information	38	0.214285714	95
P3	Identidy initial domain requirements	36	1.142857143	90
P4	Review the initial elicited domain requirements	33	0.982142857	82.5
P5	Analyze problem space of the domain architecture	33	0.410714286	82.5
P6	Synthesize potential solution alternatives	32	0.857142857	80
P7	Formulate potential domain architectures	33	1.267857143	82.5
P8	Capture domain architecture concept and properties	34	0.785714286	85
P9	Hand-off conceptualized domain architecture to user and other stakeholders	32	0.571428571	80
P10	Confirm inputs for domain implementation	32	0.857142857	80
P11	Confirm domain implementation capability	31	1.267857143	77.5
P12	Identify application test artifacts managed as application assets	30	0.785714286	75
P13	Structure configuration and annotation for application test assets	30	1.071428571	75
P14	Assess process	31	0.410714286	77.5
P15	Estimate the impact of changes on processes	32	0.571428571	80
P16	Plans for process improvements	35	0.267857143	87.5
P17	Plans for process improvements	33	1.267857143	82.5
P18	Evaluate process improvement	33	0.696428571	82.5

Table 4.2: Practice Items and their adaptation rate

Table 2 outlines various practice items along with their corresponding total points, variances, and adaptation percentages. Each practice item represents a specific task or activity within the domain. Notably, the variance of each item indicates the degree of variability or consistency in performance across assessments.

Items such as "Gather domain information" and "Identify initial domain requirements" demonstrate high levels of adaptation, with percentages reaching 95% and 90% respectively, suggesting strong proficiency in these areas. Conversely, "Identify application test artifacts managed as application assets" and "Structure configuration and annotation for application test assets" exhibit lower adaptation percentages at 75%, indicating potential areas for improvement or further focus.

The table's variance values show how scores for each practice item vary or disperse across various tests. Greater performance variability is indicated by higher variance values, which imply that some tasks might be more difficult or interpretable than others.

Practice items such as "Identify initial domain requirements" and "Draw a context diagram of the whole system" show comparatively low variances, suggesting more consistent performance across assessments. Conversely, items like "Identify domain implementation capability" and "Implement process improvements" show higher variance values, suggesting that evaluators or different contexts may give more varied scores when completing these tasks.

To ensure more consistent and dependable performance across assessments, it is essential to comprehend the variance of each practice item in order to pinpoint areas that may require further instruction or clarification.

Field of Practise	Total Points for each practice field out of 90	Variance in practisce field	Adaptation (%)
Lead Mobile Applications	61	1.075163399	67.78
SQA Analyst	75	0.617647059	83.33
Software Developer	78	0.23529411	86.67
Project Coordinator	67	0.683006536	74.44
Principle Engineer	67	1.035947712	74.44
Executive officer	90	0.0000000	100
IT manager	74	0.692810458	82.22
Software engineer	75	0.382352941	83.33

Table 4.3: Adaptation in field of practice

Table 4.3 shows how quickly practice items are adapted in different fields of practice. There is a certain percentage that represents the adaptation rate for each field, such as Lead Mobile Applications, SQA Analyst, Software Developer, Project Coordinator, Principal Engineer, Executive Officer, IT Manager, and Software Engineer. This percentage shows how good or ready people in each field are at using or adopting certain practices. For example, Executive Officers have a 100% adaptation rate, which means that they have fully adopted all of the practice items. On the other hand, Lead Mobile Applications have a 67.78% adaptation rate, which means that there are some areas where practices could be improved or more fully integrated. The table is a way to compare how well different fields of practice adapt, which can help improve performance and strategy in each area.

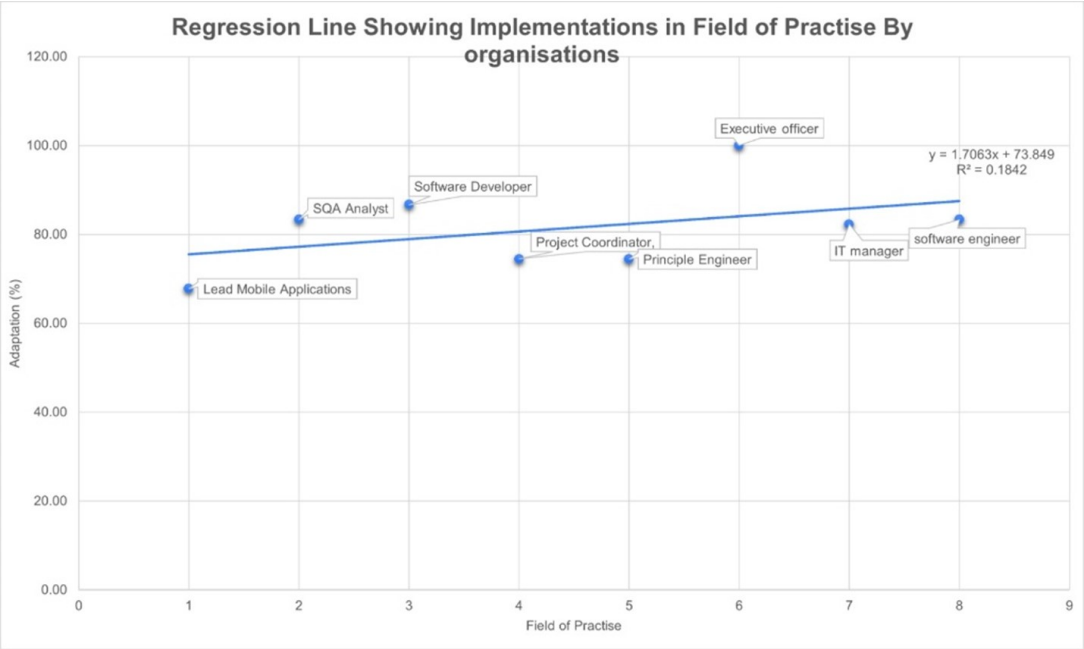


Figure 4.4: Regression Line Showing Implementations in Field of Practice By Organizations

of Item (k)	**18**
Summed Variance of Each Item	14.2678
Variance of Total Score for each participant	76.8392
Cronbach's Alpha	.862216

Table 4.4: Cronbach's alpha

The variance values show how much performance varies or spreads out across different assessments in each practice field. A lower variance means that the scores in that practice field are pretty stable across evaluations, which means that the performance level is more stable and predictable. For instance, the fields of Software Developer, Software Engineer, SQA Analyst, and IT Manager have relatively low variance values, which means that the tests in these fields tend to give the same results each time. Conversely, higher variance values imply greater variability in performance across assessments within a particular practice field. For instance, fields like Lead Mobile Applications, Principal Engineer, and Project Coordinator have higher variance values, suggesting that the assessments within these fields may produce more diverse scores among evaluators or across different contexts. This variability could stem from various factors such as the complexity of tasks within the field, differing interpretations of assessment criteria, or varying levels of expertise among individuals evaluated.

Practice	**Sub-areas**	**Six-Sigma phase**	**Action phase(Plan/ Do/Check/Act**
P1	requirement analysis and system architecture development	Define	Plan
P2	requirement analysis and domain knowledge acquisition areas	Define	Plan
P3	requirement analysis area	Define	Plan
P4	requirement analysis and validation area	Measure	Check
P5	system architecture and design area & SSO	Define	Plan
P6	system design and architecture development areas	Analyze	Planning and execution
P7	system architecture and design area	Define	Planning and design
P8	system architecture and design area	Define	Planning and design
P9	system architecture and communication with stakeholders areas	Control	Planning and execution
P10	planning and requirement validation areas	Measure	Planning and execution
P11	technical feasibility analysis and system architecture development	Measure	Planning and execution
P12	quality assurance and testing areas	Measure	Planning
P13	quality assurance and testing areas	Control	Do
P14	process improvement and optimization	Analyze	Check
P15	change management and project planning areas	Measure	Planning
P16	process management and quality assurance	Define	Planning
P17	process improvement areas	Improve	Planning
P18	quality assurance and process improvement areas	Control	Check

Table 4.5: Practice, sub-areas, six sigma phases and action phase

Practice	Aims to
P1	to identify and define the problem, goals, scope, and project objectives
P2	understanding the problem, identifying project goals, and defining project scope
P3	defining the problem, project goals, scope, and stakeholders' requirements
P4	defining project goals and requirements, gathering and reviewing initial requirements to ensure they align with stakeholder needs and project objectives
P5	understanding the problem, setting project goals, and defining project scope, ensuring that the system architecture aligns with the project objectives and requirements
P6	Identifying and evaluating potential solutions to address identified issues or meet project requirements
P7	understanding the problem, identifying project goals, and defining project scope
P8	understanding the problem and setting project goals and scope
P9	implementing solutions, monitoring their performance, and ensuring that the improvements are sustained over time
P10	identifying and defining critical inputs and outputs of the process
P11	collecting data and analyzing the current state of the process or system to understand its performance and capability
P12	Identifying test artifacts as assets helps in understanding the testing process's effectiveness and efficiency
P13	Configuring and annotating test assets properly contribute to maintaining the quality standards of the software product and its ongoing development
P14	analyzing data and processes to identify root causes of issues and opportunities for improvement. Assessing the process helps in understanding current performan
P15	Identifying and measuring key process metrics to establish a baseline for improvement efforts
P16	Planning for process improvements is essential for identifying areas of improvement within the software development process
P17	Define, Measure, and Analyze phases and focuses on implementing solutions identified to address the root causes of the problem or to achieve the project goals
P18	monitoring and controlling the improved processes to ensure that they continue to meet the desired performance standards and objectives

Table 4.6: Practices aim to solve the problems

Table 4.6 link ISO/IEC 26550 practices to artefacts (e.g., use cases, test plans) and PDCA/Six Sigma stages, reinforcing iterative SPI.

RQ1: Key challenges include limited awareness, weak QA practices (75%), resource gaps (variance 1.27), and low unified process use (65%) [33][17].

RQ2: A tailored PDCA-based framework [35] offers a cost-effective, role-defined, and feedback-driven SPI path aligned with SME needs.

RQ3: Strong core practice adoption (e.g., 95% domain info, 86.67% developers) and high reliability (Cronbach's = 0.8622) [19] support the framework's fit. Tables 5–6 tie practices to Six Sigma phases, showing practical applicability.

Chapter 5

Sustainability

5.1 Sustainability Overview

In improving software processes for small and medium-sized enterprises (SMEs), sustainability means that the lightweight ISO/IEC 26550 framework is not just technically effective, but also affordable, socially acceptable, environmentally responsible, and ethically fair. Considering these aspects helps the framework stay practical, useful, and adaptable for Bangladeshi SMEs, even as industry needs change and resources remain limited.

5.2 Economical Sustainability

Basically, economic sustainability focuses on the cost-effectiveness and financial feasibility. To implementing the framework in SMEs, they often face resource constraints. This proposed lightweight framework will minimize adoption costs in following ways:

- It will reduce the cost of extra training and expert consultants.
- It will carry simple documentation and compliance processes.
- This framework is designed in a way that any organization do not need major restructuring to adopt ISO/IEC 26550 practices.

This ensures that SMEs can improve software process capability without facing prohibitive financial burdens, making the framework sustainable in the long run.

In the present situation in Bangladesh, we can say SMEs in Bangladesh regularly struggle with high upfront cost and limited financing access to adopting process improvements. This framework will minimize the need for external consultants, complex documentation, and restructuring needs of ongoing organization's policy.

5.3 Social and Environmental Sustainability

Social Sustainability

Practicing this framework contributes in social sustainability in many ways like it improve team skills of local software engineers and managers, process transparency and reputation of small company. On the other hand, this framework helps to build professional growth, improves teamwork, and encourages knowledge sharing across SME teams. Furthermore, practicing this international standard strengthens the reliability of Bangladeshi software firms and employment stability improving in global markets.

Environmental Sustainability

This research indirectly contributes on environment. By practicing this framework, organizations will concern about efficient documentation, process automation, and reduced redundancy. The framework minimizes unnecessary paper usage and resource wastage. Moreover, by adopting this practice, SMEs can reduce their carbon footprint and align it with environmentally responsible business models.

5.4 Ethics

The proposed framework addresses ethical sustainability by:

- It helps to report honest process maturity and compliance.
- It helps to protect data privacy during surveys, interviews, and case studies with SMEs.
- It helps to use responsible resources and discourage manipulative practices where organizations might falsely claim standard compliance.
- The framework helps to build trust and credibility in both organization and with external stakeholders.

By take part economic, social, environmental, and ethical dimensions, this lightweight ISO/IEC 26550 framework offers a complete and sustainable solution for SME software process improvement in Bangladesh.

Chapter 6

Conclusion

6.1 Project Summary

This project contributes to multiple sustainability goals: Generally, SPI for small and medium software firms is definitely a topic where new chance and challenges could be addressed. Most of the software firms either failed to adopt standard or cannot afford it. The main objective of this thesis is building a framework for those software companies who are struggling continuously against low productivity and quality. Our analysis focused on understanding how specific practice areas were tailored to suit the needs of small software firms in Bangladesh. Our research is ISO-based SPI initiatives in SMEs by integrating phases of the Plan-Do-Check-Act (PDCA) cycle. The ultimate goal is to propose a framework that facilitates cost-effective, efficient, and comparatively high productivity. The proposed lightweight framework of ISO/IEC 26550, focusing on planning, monitoring, validation, and continual improvement, while maintaining all standards.

Key contributions of this thesis include:

• Key barriers to adopt ISO/IEC 26550 in the Bangladeshi software sector

• Validation of the framework through expert review and SME feedback to ensure feasibility and relevance

• Flexibility, cost-efficiency, and minimal documentation for easy adoption

6.2 Future Work

Very few companies are practicing software process improvement framework in their project management and software development life cycle. But small and medium sized software companies are totally avoiding this standard. For this reason, they are facing many problems in later. This framework is specially structured for SME in Bangladesh. After this, future work will be implementing the proposed framework within actual SME settings throughout Bangladesh. Organization should adopt this framework for enhancing process quality, team productivity, and product delivery over time.

Bibliography

[1] Dyba, T. (2005). An empirical investigation of the key factors for success in software process improvement. IEEE Transactions on Software Engineering, 31(5), 410-424.

[2] Beebe, N. H. (2018). A complete bibliography of publications in the Journal of Software Maintenance and Evolution: Research and Practice.

[3] Van der Linden, F. J., Schmid, K., & Rommes, E. (2007). Software product lines in action: The best industrial practice in product line engineering. Springer Science & Business Media.

[4] Weiss, D. M., & Lai, C. T. R. (1999). Software product-line engineering: A family-based software development process. Addison-Wesley Longman Publishing Co., Inc.

[5] Zahran, S. (1999). Software process improvement: Practical guidelines for business success. Addison Wesley Longman, UK.

[6] Pohl, K., Bockle, G., & Van Der Linden, F. (2005). Software product line engineering: Foundations, principles, and techniques (Vol. 1). Heidelberg: Springer.

[7] O'Regan, G. (2002). A practical approach to software quality. Springer Science & Business Media.

[8] Marques, M., Simmonds, J., Rossel, P. O., & Bastarrica, M. C. (2019). Software product line evolution: A systematic literature review. Information and Software Technology, 105, 190-208.

[9] Montalvillo, L., & D´ıaz, O. (2022). Evolution in software product lines: An overview. Handbook of Re-Engineering Software Intensive Systems into Software Product Lines, 495-512.

[10] Montalvillo, L., & D´ıaz, O. (2016). Requirement-driven evolution in software product lines: A systematic mapping study. Journal of Systems and Software, 122, 110-143.

[11] Quinton, C., Vierhauser, M., Rabiser, R., Baresi, L., Gru¨nbacher, P., & Schuhmayer, C. (2021). Evolution in dynamic software product lines. Journal of Software: Evolution and Process, 33(2), e2293.

[12] Horcas, J. M., Pinto, M., & Fuentes, L. (2023). Empirical analysis of the tool support for software product lines. Software and Systems Modeling, 22(1), 377-414.

[13] Kakola, T. (2010, January). Standards initiatives for software product line engineering and management within the international organization for standardization. In 2010 43rd Hawaii International Conference on System Sciences (pp. 1-10). IEEE.

[14] Thomas, S. A., & Bindu, V. R. (2022, August). Sustainment and management of software process improvement initiatives. In AIP Conference Proceedings (Vol. 2520, No. 1). AIP Publishing.

[15] Chevers, D., Mills, A. M., Duggan, E., & Moore, S. (2020). A software process improvement model for small firms in developing countries. In Novel theories and applications of global information resource management (pp. 47-80). IGI Global.

[16] Gunawan, F., & Budiardjo, E. K. (2021, January). A quest of software process improvements in DevOps and Kanban: A case study in small software company. In Proceedings of the 2021 4th International Confer- ence on Software Engineering and Information Management (pp. 39-45).

[17] Walter, B., Marovic´, B., Garnizov, I., Wolski, M., & Todosijevic, A. (2022, August). Two case studies on implementing best practices for software process improvement. In European Conference on Software Process Improvement (pp. 259-270). Cham: Springer International Publishing.

[18] Zarour, M., & Alenezi, M. (2021). Software process improvement initiative in the medium-sized IT organization: A case study. Indonesian Journal of Electrical Engineering and Computer Science, 23(1), 500- 509.

[19] Huysegoms, T., Snoeck, M., Dedene, G., & Goderis, A. (2013). A case study on variability management in software product lines: Identifying why real-life projects fail. International Journal of Information Systems and Project Management, 1(1), 37-48.

[20] Akthar, S. R. (2024). Adaptive Software Engineering: Optimizing Deployment Strategies and Mitigating Risk Factors for Success from ERP and Game Development (Doctoral dissertation, IUB, CSE).

[21] Bilic, D., Sundmark, D., Afzal, W., Wallin, P., Causevic, A., Amlinger, C., & Barkah, D. (2020, February). Towards a model-driven product line engineering process: An industrial case study, in Proceedings of the 13th Innovations in Software Engineering Conference (pp. 1-11).

[22] Khan, A. A., Keung, J. W., & Abdullah-Al-Wadud, M. (2017). SPIIMM: Toward a model for software process improvement implementation and management in global software development. IEEE Access, 5, 13720- 13741.

[23] Tendedez, H., Ferrario, M. A. M., & Whittle, J. (2018). Software development and CSCW: Standardization and flexibility in large-scale agile development. Proceedings of the ACM on Human-Computer Interaction, 2(CSCW), 1-23.

[24] Taber, K. (2018). The use of Cronbach's alpha when developing and reporting research instruments in science education. Research in Science Education, 48, 1-24.

[25] Smetkowska, M., & Mrugalska, B. (2018). Using Six Sigma DMAIC to improve the quality of the production process: A case study. Procedia Social and Behavioral Sciences, 238, 590-596.

[26] Lesum, S. A., Akthar, S. R., Islam, M. R., Sadia, F., & Hasan, M. (2024). Project Governance to Improve the Performance of Software Projects by Mitigating the Software Risk Factors: The Moderating Role of Project Leadership. Procedia Computer Science, 239, 1863-1870.

[27] Pettersson, F., Ivarsson, M., Gorschek, T., & O¨ hman, P. (2008). A practitioner's guide to lightweight software process assessment and improvement planning. Journal of Systems and Software, 81(6), 972–995.

[28] Naughton, E., Moran, R., Kharub, M., Sa, J. C., and McDermott, O. (2024). A structured model for continuous improvement methodology deployment and sustainment: A case study. Heliyon, 10(21), e40034.

[29] Six Sigma. (2025). Capability Maturity Model Integration (CMMI). Retrieved from https://www.6sigma.us/process-improvement/capability- maturity-model-integration-cmmi/ (Accessed: 2025-04-19).

[30] ASQ. (2025). PDCA Cycle. Retrieved from https://asq.org/quality- resources/pdcacycle (Accessed: 2025-04-19).

[31] Thakur, O. A., Alam, M. K., Masum, M. H. (2022). The Problems and Challenges of Small and Medium Enterprises in Bangladesh. Journal of Business and Development Studies (JBDS), 1(1), 53-71.

[32] Friman, S. (2023). Idea management software to facilitate continuous improvement processes and PDCA (Doctoral dissertation, Technische Universität ¨t Wien).

[33] Zaied, A. N. H., El-Drandaly, K. A., & Tantawy, A. S. A. A proposed Framework for Software Process Improvement: Extending CMMI-DEV model Using Six Sigma and Quality Function Deployment Techniques.

[34] Akthar, S. R., Islam, M. R., Hasan, M. B., Siddiqua, M. S., Haque, S. I., Saad, J. A., ... & Hasan, M. Overcoming Obstacles in Model-Driven Engineering: Lessons from the Software Industry.

[35] Ka¨ko¨la¨, T. (2013). A revision of the International Standard ISO/IEC 26550:2013 Software and systems engineering – Reference model for product line engineering and management.

[36] Chakraborty, P., Hasan, K., Iqbal, A., Uddin, G., & Shahriyar, R. (2024). A comparative study of software development practices in Bangladesh, an emerging country. International Journal of Software Engineering, Technology and Applications, 2(2), 149-187.

[37] Mahdi, M. N., Walshe, R., Farrell, S., & Pandit, H. J. (2024). Comprehensive Review and Future Research Directions on ICT Standardisation. Information, 15(11), 691.

[38] Laporte, C. Y., O'Connor, R. V., & Paucar, L. H. G. (2015, April). The implementation of ISO/IEC 29110 software engineering standards and guides in very small entities. In International Conference on Evaluation of Novel Approaches to Software Engineering (pp. 162-179). Cham: Springer International Publishing.

[39] Wongsai, N., Siddoo, V., & Wetprasit, R. (2015, October). Factors of influence in software process improvement: An ISO/IEC 29110 for very-small entities. In 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE) (pp. 12-17). IEEE.

[40] Bibi, S., Gerogiannis, V. C., Kakarontzas, G., & Stamelos, I. (2014, August). Ontology based bayesian software process improvenent. In 2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA) (pp. 568-575). IEEE.

[41] Gökalp, E., Çalişkanbaş, B., & Koçyiğit, A. (2024). A reference model for software product line capability assessment. Journal of Software: Evolution and Process, 36(4), e2567.

[42] Kim, J. A. (2022). A case study of domain engineering in software product line engineering. Journal of Logistics, Informatics and Service Science, 9(1), 97-115.

[43] da Silva, I. F., Neto, P. A. D. M. S., O'Leary, P., de Almeida, E. S., & de Lemos Meira, S. R. (2014). Software product line scoping and requirements engineering in a small and medium-sized enterprise: An industrial case study. Journal of Systems and Software, 88, 189-206.

[44] Stallinger, F., Neumann, R., Schossleitner, R., & Kriener, S. (2011, May). Migrating towards evolving software product lines: Challenges of an SME in a core customer-driven industrial systems engineering context. In Proceedings of the 2nd International Workshop on Product Line Approaches in Software Engineering (pp. 20-24).

[45] Knauber, P., Muthig, D., Schmid, K., & Widen, T. (2000). Applying product line concepts in small and medium-sized companies. IEEE Software, 17(5), 88-95.

[46] Clements, P., & Northrop, L. (2002). Software product lines (Vol. 1). Boston: Addison-Wesley.

[47] Carlos Herrera, J., Losavio, F., & Ordaz, O. (2016). QuaDRA: Quality-oriented Design of Reference Architecture for Software Product Lines based on ISO/IEC 26550. Revista Antioqueña de las Ciencias Computacionales, 6(1).

[48] Chimalakonda, S., & Lee, D. H. (2021). A family of standards for software and systems product lines. Computer standards & interfaces, 78, 103537.

[49] Chimalakonda, S., & Lee, D. H. (2016). On the evolution of software and systems product line standards. ACM SIGSOFT Software Engineering Notes, 41(3), 27-30.

[50] ISO/IEC 26550. (2013). Software and systems engineering—Reference model for product line engineering and management.

[51] Derras, M., Deruelle, L., Douin, J. M., Levy, N., Losavio, F., & Reiner, V. (2018, July). Reference Architecture Design: a practical approach. In 13th International Conference on Software Technologies (ICSOFT) (Vol. 1, pp. 599-606). SCITEPRESS SCIENCE AN D TECHNOLOGY PUBLICATIONS.

[52] Metzger, A., & Pohl, K. (2014). Software product line engineering and variability management: achievements and challenges. Future of software engineering proceedings, 70-84.

[53] Gökalp, E., Çalişkanbaş, B., & Koçyiğit, A. (2024). A reference model for software product line capability assessment. Journal of Software: Evolution and Process, 36(4), e2567.

[54] Pohl, K., & Metzger, A. (2018). Software product lines. In The Essence of Software Engineering (pp. 185-201). Cham: Springer International Publishing.

[55] OliveiraJr, E., & Benavides, D. (2022). Principles of software product lines. In UML-Based Software Product Line Engineering with SMarty (pp. 3-26). Cham: Springer International Publishing.

[56] Ihme, T., Pikkarainen, M., Teppola, S., Kääriäinen, J., & Biot, O. (2014). Challenges and industry practices for managing software variability in small and medium sized enterprises. Empirical Software Engineering, 19(4), 1144-1168.

[57] Takahashi, M. (2020). Problems of Small and Mid-Sized Enterprises in Japan'S Software Industry. Eurasian Journal of Economics and Finance, 8(4), 274-278

[58] Ashrafuzzaman, M. D., Chowdhury, M. S. H., Akhtaruzzaman, M. D., & Sarwar, H. (2014). An assessment of software development practices of SMEs in Bangladesh. In International Conference on Advances in Computer Science and Electronics Engineering-CSEE (Vol. 2014).

[59] Oyshi, F., Bonik, A., Shin, M. O. H., Rashid, J., Akter, M., Hasan, M., & Sadia, F. (2023). A Novel model to adapt CMMI Level 2 by Assessing the Local SMEs of Bangladesh. Procedia Computer Science, 219, 2043-2050.

[60] Haque, S. (2005). Adoption and up gradation of CMMI: prospect of software industry of Bangladesh (Doctoral dissertation, BRAC University).

[61] Wong, B., & Hasan, S. (2006). Software process improvements in bagladesh. In International Conference on Software Engineering Research and Practice. CSREA Press.

[62] Begum, Z., Khan, M. S. A., Hafiz, M., Islam, M. S., & Shoyaib, M. (2010). Software development standard and software engineering practice: A case study of Bangladesh. arXiv preprint arXiv:1008.3321.

[63] Solyman, A. M., Ibrahim, O. A., & Elhag, A. A. M. (2015, September). Project management and software quality control method for small and medium enterprise. In 2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE) (pp. 123-128). IEEE.

[64] Keshta, Ismail. (2019). Approaches to Software Process Improvement: A State-of-the-Art Review. Journal of Software. 519-529. 10.17706/jsw.14.11.519-529.

[65] K Pohl, G Böckle, FJ van Der Linden (2005). Software product line engineering: foundations, principles and techniques, Springer.

[66] Robert Lindohf, Jacob Krüger, Erik Herzog, Thorsten Berger (2021). Software product-line evaluation in the large, Empirical Software Engineering, 26, 30(2021).

[67] Krüger J., Mahmood W., Berger T. (2020). Promote-pl: A round-trip engineering process model for adopting and evolving product lines. International conference on systems and software product line engineering

[68] Kim, J.A., Yang, J.S. (2017). Software Product Line Lifecycle Management-Integration Engineering and Management Process. In: Park, J., Chen, SC., Raymond Choo, KK. (eds) Advanced Multimedia and Ubiquitous Engineering. FutureTech MUE 2017. Lecture Notes in Electrical Engineering, vol 448. Springer, Singapore. https://doi.org/10.1007/978-981-10-5041-1822017

[69] Pushpo, F. H., & Uddin, M. K. Prospect and Opportunities for ISO Standards in Bangladesh.

[70] Unterkalmsteiner, M., Gorschek, T., Islam, A. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2014). A conceptual framework for SPI evaluation. Journal of Software: Evolution and Process, 26(2), 251-279.

[71] Fuggetta, A., & Di Nitto, E. (2014). Software process. In Future of Software Engineering Proceedings (pp. 1-12).

[72] Bayona-Oré, S., Chamilco, J., & Perez, D. (2019). Applying CMMI best practices to improve processes. In MATEC Web of Conferences (Vol. 292, p. 01065). EDP Sciences.

[73] Walter, B., Wolski, M., Stanisavljević, Ž., & Todosijević, A. (2019, August). Designing a maturity model for a distributed software organization. An experience report. In European Conference On Software Process Improvement (pp. 123-135). Cham: Springer International Publishing.

[74] Ali, M. S., Babar, M. A., & Schmid, K. (2009, August). A comparative survey of economic models for software product lines. In 2009 35Th euromicro conference on software engineering and advanced applications (pp. 275-278). IEEE.

[75] Schuhmayer, C. (2021). Evolution in dynamic software product lines. Journal of Software: Evolution and Process, 33(2), e2293.