2024-07-09

Creation of consumer and producer surplus through integration of the software engineering body of knowledge for De-Soft

Dewri, Tusher Godfray

IUB

Creation of consumer and producer surplus through integration of the software engineering body of knowledge for De-Soft.

Tusher Godfray Dewri
Student ID # 1022005

July 09, 2024

Supervisor:
Mahady Hasan, PhD
Associate Professor
Department of Computer Science & Engineering
Independent University, Bangladesh

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

Department of Computer Science & Engineering
Independent University, Bangladesh
July 09, 2024

Attestation

We affirm that this graduate report is the result of our own independent work undertaken during our post-graduate study. We have acknowledged all materials and sources employed in this document. To the best of our knowledge, this report has not been previously submitted for assessment in any other academic unit and we have not engaged in any form of plagiarism. We adhere to international academic standards providing proper citations for the use of others' work within our University project.

Signature Date

Name

Acknowledgement

I do hereby express my deepest thanks and appreciation to Mahady Hasan, PhD, our honorable teacher for giving us his valuable time and sincere guidance during the study period which has inspired us in preparing this report.

I am indebted to Mr. Tahsin (Sr. Engineer) of De-Soft, Mr. Tusher (CEO) of De-Soft, Mr. Reza (IT Manager) of De-Soft, and other officers who provided required information about the production and consumption of software through their long working experience to prepare this report. And also De-Soft's clients who used their software and shared their experiences from which I was able to find out the benefit of consumption of software. Finally, thanks to Almighty GOD and our parents for giving us their blessings.

Application to Submit the Report

July 09, 2024
Mahady Hasan, PhD
Associate Professor
School of Engineering and Computer Science
Independent University, Bangladesh

Subject: Submission of Final Thesis Paper

Dear Sir,
Please accept my thesis paper which covers the following topic:

1. Explain consumer surplus (customers or clients) in the context of software engineering where consumers are able to benefit.
2. Spell out producer surplus (software company) and how they are able to benefit.
3. Describe the process both consumer and producer will benefit and improve day by day if they follow the software engineering body of knowledge.

I tried my level best and covered all the areas according to your suggestions. I therefore pray and hope that you will like my work and accept my project.

Your Obedient Student,
Tusher Godfray Dewri #1022005

Contents

Abstract

Dedicated I focused on one company for the last couple of years and studied this company's (De-Soft) last 14 years journey. I thoroughly studied their requirement gathering and analysis, product line, architecture process, marketing, product selection process, project handling, service and support development, communication, customization, quality maintenance, and business process re engineering. The engineering approach applicability limitation is eye-catching in this sector in our country. Most startup companies are not concerned about the engineering approach. They believe in software development coding and implementation. But now they are starting to practice the engineering approach and converting themselves. Producers' production and surplus benefit and benefit level management engineering now are bold line discussion matter as well as consumer surplus. In this paper, we will discuss more with real industry experience and data how to improve consumer and producer surplus through the integration of the software engineering body of knowledge. Software engineering every single tool's implementation will help us to improve both producer and consumer surplus.

Chapter 1

Introduction

1.0.1 Overview

Ever since Bangladesh's software sector first emerged, companies have faced enduring difficulties in turning a profit from their software ventures. Technical skill is important in software development, but without a solid understanding of economics, it frequently does not translate into profitable endeavors. This disparity affects businesses' capacity to optimize excess from consumers and producers, impeding their long-term viability and profitability.

The use of economics to software industry methods provides a methodical way to deal with these issues. An understanding of consumer behavior and preferences is essential to this integration. Economic frameworks shed light on how much consumers are ready to pay and how much they think software is worth. By utilizing this information, software companies may better customize their products to match the needs of their customers, which will increase consumer surplus—the extra value that customers receive from software above what they pay for.

Additionally, software companies use economics to optimize their pricing methods. By using strategies like price discrimination and dynamic pricing, businesses can maintain their competitiveness in the market while gaining a larger share of the surplus from consumers. In addition to increasing income, this strategic pricing raises the perceived value of software products, retaining customers and broadening the market.

Economic considerations also guide decisions about resource allocation and product creation. Businesses can lessen the risk of investing in unneeded features by giving priority to features that are in line with consumer demand and financial sustainability. Over time, sustainable growth is supported and the return on investment is enhanced by this effective resource allocation.

Additionally, by comprehending the financial effects of software development and use, businesses may better negotiate regulatory environments. By putting in place laws that uphold intellectual property rights, encourage competition, and encourage innovation—all essential elements in guaranteeing the software industry's long-term survival—policymakers may cultivate an atmosphere that is favorable to industrial expansion.

Nonetheless, this report focuses on if the software engineering sector is to grow in Bangladesh and elsewhere, economics must be incorporated into core principles. It enables businesses to better allocate resources, increase consumer and producer surplus, optimize pricing tactics, and successfully handle regulatory obstacles. In addition to increasing profitability, this all-encompassing strategy fosters innovation and sustainable growth, setting up software companies for long-term success in a cutthroat international market.

1.0.2 Contribution of the Thesis

This thesis on consumer and producer surplus in the software industry of Bangladesh makes several significant contributions to both academia and industry practice.

The implementation of economic principles into software development and commercial strategy is one of its main contributions. In the past, the software sector has frequently placed a strong emphasis on technical know-how while under utilizing economic ideas. By offering a systematic framework that enables software companies to make well-informed decisions about product development, pricing strategies, and resource allocation based on economic theory, this study fills this knowledge vacuum. Requirement engineering, quality assurance, software architecture, software marketing, BPR, and project management make up the six design facets of the study. The thesis highlights the significance of economic factors in guaranteeing the long-term sustainability and longevity of software companies. Businesses can reduce the risks brought on by changes in the market and pressure from competitors by using economic principles like strategic pricing and effective resource allocation. In addition to increasing profitability, this proactive strategy fortifies the company's capacity for innovation and adaptation in the face of shifting market conditions.

The thesis also offers policymakers in Bangladesh insightful analysis and helpful policy recommendations. It supports laws that uphold favorable regulations for the software

sector, such as those that safeguard intellectual property rights, encourage honest competition, and foster innovation. Policymakers can promote growth and competitiveness in the industry and so aid in the nation's overall economic development by cultivating such an environment.

Through the unique application of economic theories to the Bangladeshi software sector, this thesis advances academic understanding. It provides case studies and empirical analysis that show how economic principles can be applied in real-world situations to improve business outcomes for the industry. The thesis is a useful tool for academics, professionals, and legislators who want to comprehend the economics of the software industry and encourage its long-term expansion.

In summary, this thesis adds more than just theoretical understanding; it also offers real-world applications that have the potential to greatly improve the operational effectiveness, financial success, and long-term viability of software companies in Bangladesh. By incorporating economic principles into decision-making procedures, businesses can better handle complexity, seize market opportunities, and support the nation's competitive and flourishing software industry ecosystem.

1.0.3 Organization of the Thesis

In Chapter 1, the introduction section describes the entire report by offering a concise overview of the research projects undertaken. Moving on to Chapter 2, an analysis of relevant literature is presented. This comprehensive review serves to establish the theoretical framework and contextual background for the next chapters. The methodology section of Chapter 3 explains how we gathered our data on several fronts and how we examined the results. In Chapter 4, the implementation scenario is described. It shows the challenges and opportunities posed by this industry and offers valuable insights for stakeholders and decision-makers. Finally, Chapter 5, the conclusion section serves to summarize the collective knowledge presented throughout the report and provides a cohesive understanding of the research outcomes.

1.1 Thesis Project Management

1.1.1 WBS

WBS is a great tool for edition and for expanding the scope of fundamental stages. into sections that are far more in-depth. The diagram below illustrates how a top-down method was employed to further break down the smaller activities that are required to finish. the stages. This structure has been used throughout the development of research articles in general. Figure 1.1 makes it clear that the high-level research project and strategy are broken down into four stages. Finding the research gap and identifying the essential but unexplored study fields are the main goals of the first phase, initiation and definition.In order to obtain a thorough understanding of the study areas, a literature review is carried out if the research paper has noteworthy contributions and significance. The study articles are gathered from Google Scholar. The second stage, which is planning, the development of the project management plan takes into account additional scopes that specify deadlines and tasks. The formulation of a timetable and the advancement of knowledge aid in the formulation of research methods, including data gathering. Additionally, the design of the study methodology includes specifying the instruments and procedures needed for the essential data analysis.Now moving on to the third phase; Execution, after the methods are identified, we execute the research by initiating data collection and extraction using Python, scale sets and excel. This provides us the ability to visualize the extracted data for analysis and compile results into meaningful outcomes. Finally, in the fourth 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. As the final version is developed the research papers are submitted for publication. Later following external reviews by publishers, more flaws are detected with which when corrections are done the paper becomes publication-worthy. Given the process is followed properly eventually the research papers are published. The thesis write-up is then initiated and reviewed by the department. After the review process is completed, the final thesis version is submitted to the department. To summarize, the WBS determines the process and sub-processes necessary for the completion of a thesis. In terms of chapters 3,4 and 5, we have followed these phases to complete the research work.

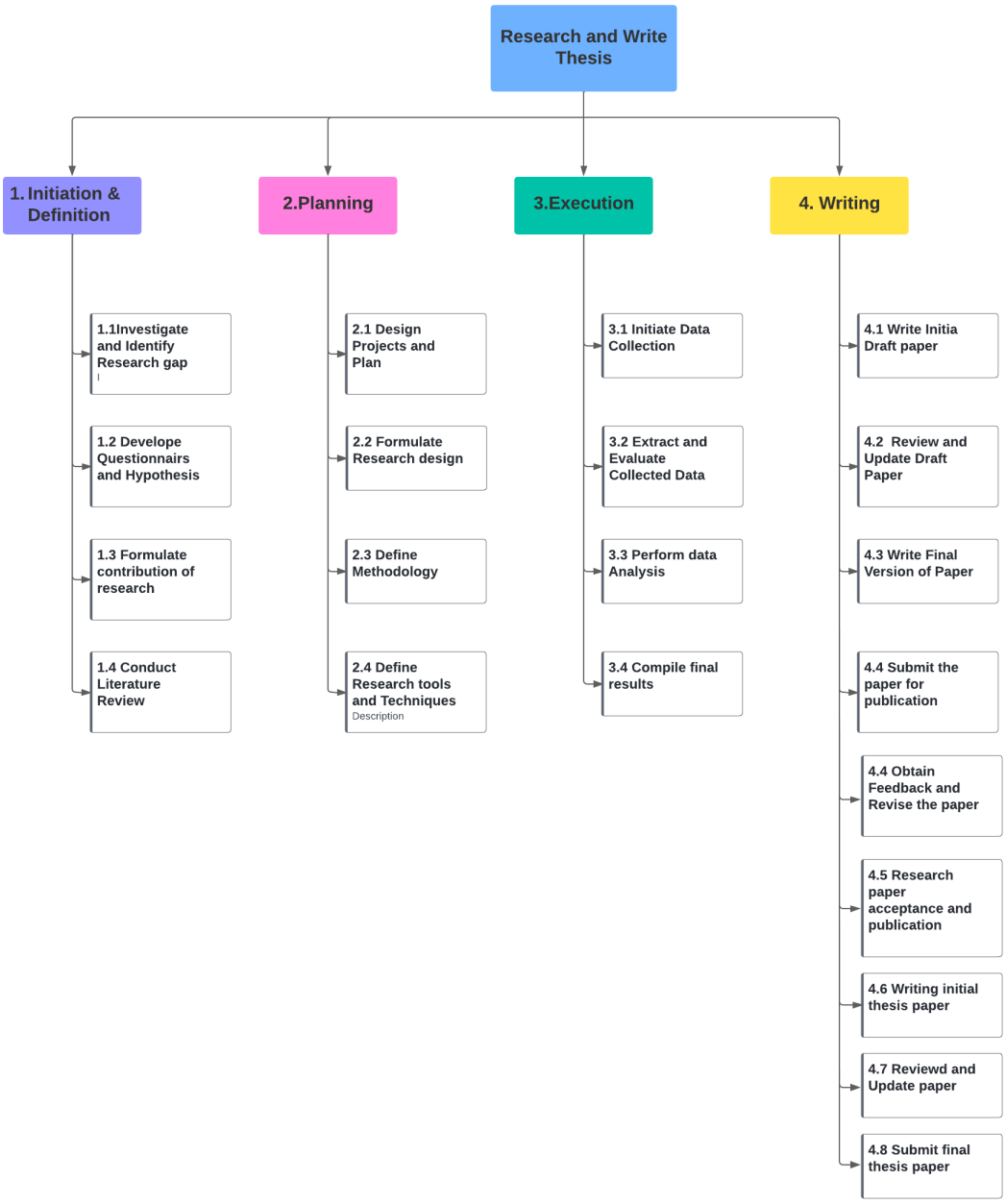
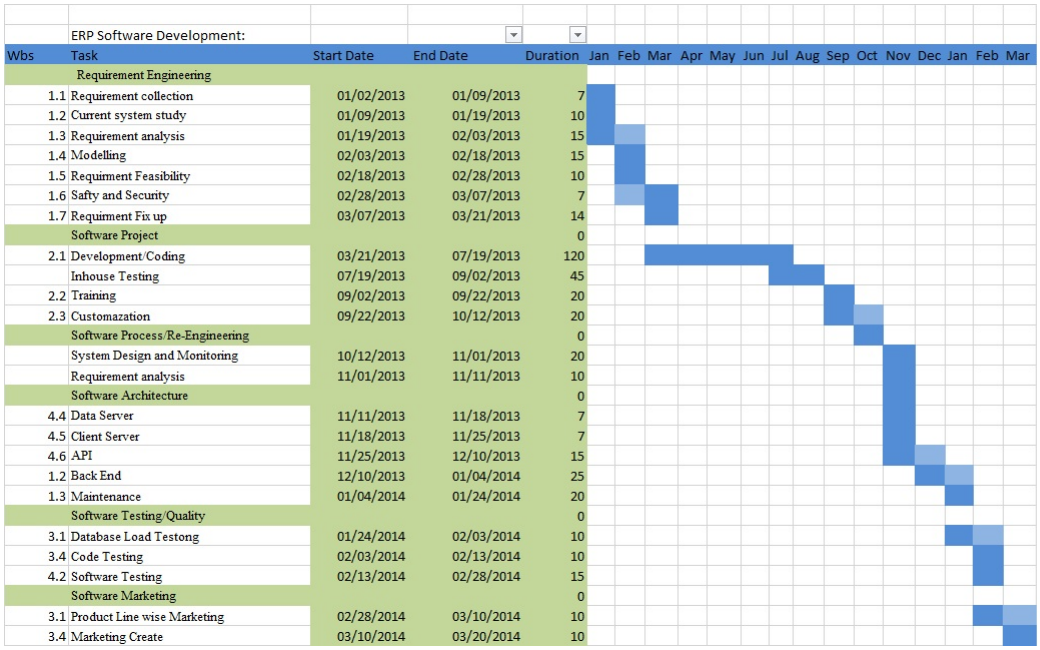


Figure 1.1: Work Breakdown Structure of Thesis

1.1.2 Gantt Chart

Gantt charts are tools that are used to schedule projects and research tasks and plan accordingly. It is a representation of the activities and days it takes to complete them. As you can see below, we have a compiled overview of the total days, each event needs to be completed in a sequential manner. The tasks are dependent on each other as one defines the next task's outcome. The Gantt chart defines the necessary tasks required to complete research papers within a certain timeline. The chart consists of elements such as Task list, Timeline and bars. These elements are being used to schedule research

timelines. In the given chart. We can find the activities related to research work and WBS are defined. Initially starting from identifying research gaps, developing theories, selecting methodologies to later on, executing collection and extraction of data to validate results. These results are reviewed with proper references and the research paper is made eligible for publication. In most cases, a research paper is hardly accepted at first submission, there are always scope for flaws and revisions which are not detected initially. As the papers are reviewed, certain flaws are modified which eventually result in acceptance and publication of a research paper. As the research work is completed, overall thesis write-up is compiled and submitted.



	ERP Software Development:			
Wbs	Task	Start Date	End Date	Duration
	Requirement Engineering			
1.1	Requirement collection	01/02/2013	01/09/2013	7
1.2	Current system study	01/09/2013	01/19/2013	10
1.3	Requirement analysis	01/19/2013	02/03/2013	15
1.4	Modelling	02/03/2013	02/18/2013	15
1.5	Requirment Feasibility	02/18/2013	02/28/2013	10
1.6	Safty and Security	02/28/2013	03/07/2013	7
1.7	Requirment Fix up	03/07/2013	03/21/2013	14
	Software Project			0
2.1	Development/Coding	03/21/2013	07/19/2013	120
	Inhouse Testing	07/19/2013	09/02/2013	45
2.2	Training	09/02/2013	09/22/2013	20
2.3	Customazation	09/22/2013	10/12/2013	20
	Software Process/Re-Engineering			0
	System Design and Monitoring	10/12/2013	11/01/2013	20
	Requirement analysis	11/01/2013	11/11/2013	10
	Software Architecture			0
4.4	Data Server	11/11/2013	11/18/2013	7
4.5	Client Server	11/18/2013	11/25/2013	7
4.6	API	11/25/2013	12/10/2013	15
1.2	Back End	12/10/2013	01/04/2014	25
1.3	Maintenance	01/04/2014	01/24/2014	20
	Software Testing/Quality			0
3.1	Database Load Testong	01/24/2014	02/03/2014	10
3.4	Code Testing	02/03/2014	02/13/2014	10
4.2	Software Testing	02/13/2014	02/28/2014	15
	Software Marketing			0
3.1	Product Line wise Marketing	02/28/2014	03/10/2014	10
3.4	Marketing Create	03/10/2014	03/20/2014	10

Figure 1.2: Gantt Chart

Chapter 2

Literature Review

This study systematically maps and analyzes 38 publications on sustainable software systems development, identifying 29 approaches since 2000. The analysis covers the evolution of these approaches, their influence on each other, supported requirements engineering activities, and provided evidence. It highlights an increase in publications discussing a multi-dimensional concept of sustainability but notes a lack of industrial application and evaluates the need for integrating these approaches into agile development processes. The study also emphasizes the importance of developing supporting software tools for practitioners.[1]

This paper introduces a Data Lake based metadata framework, which utilizes the concept of blueprints to characterize the data sources and the data itself to facilitate process mining tasks. The applicability and effectiveness of the proposed framework is validated through a real-world smart manufacturing case-study, namely a poultry meat production factory, which offers operational support and business workflow analysis.[2]

The results highlight the critical value of a thorough QA framework that supports rich learning opportunities, creates equitable chances, and improves skill acquisition. For students to receive a topnotch education and efficient service, the incorporation of QA metrics into LMS systems becomes essential. This work emphasizes the importance of ongoing research projects to look into the use and effectiveness of the suggested model in the real world[3]

This research paper presents a hybrid approach to overcome the challenges related to inadequate or insufficient client involvement and understanding during the software requirements phase. The aim of this study is to investigate the factors that contribute to this challenge and propose a solution that combines traditional and agile methodologies. To accomplish this, a survey was conducted to collect responses from industry professionals in the software development sector.[4]

In this paper, we have proposed a process model which will be followed additionally with typical software project management. By adopting this proposed process model will help the software companies to make profit of software business. This process model will build core and reusable software components based on targeted market segment and will integrate those components while developing software. Thus, software can be built with minimal customization and software companies can save their resources.[5]

Outcome Based Education aka OBE is one of the de-facto standards for modern educational system. An outcome is a culminating demonstration of learning that the students should be able to do at the end of a course, and in process at end of the degree program. Therefore, OBE is an approach to education in which decisions about the curriculum are driven by the exit learning outcomes that the students should perform in their professional life.[6]

This research implies that Project management core knowledge areas are known to all. Project Management Body of Knowledge (PMBOK) [6] has prepared these knowledge areas considering all types of projects in mind.In this paper, we have proposed additional knowledge area for software project management named Marketing Analysis. We strongly believe that adopting the activities of the proposed knowledge area software firms would be able manage their software project more efficiently and effectively. [7]

This research intends to propose solutions to the problems related with AI-based software testing solutions. We identified researchable issues and gathered insights into the current state of software companies in Bangladesh through company interviews and surveys. [8]

This study focuses on the usability in video games which is crucial to ensure that game systems comply with usability standards.The study involved conducting polls, heuristic scoring studies, and speaking with several game developers. Twelve usability heuristics were created based on the Nielsen usability technique and the stages of game software development.[9]

This study focuses on the innovators will be to improve the business processes as well as study the business practice to learn the concealed requirements of the customer. This paper is an attempt to analyze the existing proposals on the SWEBOK contents, structure, and make a proposal on what kind of contents it should have, and how it should be structured so that the innovators as well as in-house and software contractors can have proper guidance and can benefit from it.[10]

This study introduces a question-based framework designed to raise awareness among

software professionals about the sustainability effects of software systems. Recognizing the need for a paradigm shift in the industry, the framework aims to empower requirements engineers to lead discussions on sustainability. An evaluation study conducted with computer science students demonstrated that the framework is applicable across various system types and effectively facilitates conversations on the potential sustainability impacts of software systems.[11]

This study addresses the increasing necessity of software security by focusing on early identification of security threats in the software development process. It introduces the STORE methodology, aimed at systematic security requirement elicitation based on threat analysis, and identifies key points for effective attack analysis. The methodology's effectiveness is validated through a case study of an ERP system and compared with existing techniques SQUARE and MOSRE, demonstrating that STORE provides more organized and efficient elicitation of security requirements, thereby aiding requirement engineers in developing secure and high-quality software.[12]

In this study a survey was conducted to analyze the selection of requirements elicitation techniques in various software project contexts among 328 Ukrainian IT and non-IT practitioners, mainly business analysts and product owners. The seven most used techniques were identified, including interviews, document analysis, and brainstorming. Using the Chi-Square Test, significant relationships were found between project context attributes and elicitation techniques, such as company type and document analysis. Despite limitations, including the survey's focus on Ukraine and the non-exhaustive list of techniques, the findings offer practical guidance for selecting elicitation techniques and suggest areas for future research, such as analyzing other business analysis tasks and employing factor analysis.[13]

This review paper aims to provide an in-depth understanding of the current state of software testing using AI. The review will examine the various approaches, techniques, and tools used in this area and assess their effectiveness. After studying all the selected papers, we find that various testing tasks can be automated successfully using AI (Machine Learning and Deep Learning) such as Test Case Generation, Defect Prediction, Test Case Prioritization Metamorphic Testing, Android Testing, Test Case Validation, and White Box Testing. This study also finds that the integration of AI in software testing is making software testing activities easier along with better performance. This literature review paper provides a thorough analysis of the impact AI can have on the software testing process.[14]

This study investigates the role of explain ability in achieving transparency and building trust in complex software systems. Through a study with 107 participants, it iden-

tifies explain ability as a crucial factor not only for transparency but also for other non-functional requirements (NFRs). While explanations can enhance transparency and trust, they can also negatively impact usability, termed the "double-edged sword effect." To reconcile this, the study explores the relationship between explain ability and usability, offering recommendations for integrating user-centered design (UCD) techniques to align with user needs. The study highlights challenges in handling NFRs and suggests lightweight UCD activities to support requirements engineers in the elicitation, analysis, and design phases.[15]

This study investigates the impact of COVID-19 on the factors influencing on-time software project delivery across various Software Development Life Cycle (SDLC) models, such as Agile, Incremental, Waterfall, and Prototype models. The methodology involved a quantitative survey of software professionals from 11 companies in Bangladesh, using data analysis techniques including Python, Net Promoter Score, mode, and mean calculations. The study found significant changes in key factors affecting project delivery, with attentions focus, team stability, communication, team maturity, and user involvement emerging as crucial during COVID-19, while team capabilities, infrastructure, and team commitment were more critical before the pandemic.[16]

This study examines strategies to overcome slow growth in software firms by addressing the impact of high project diversity. The methodology involves analyzing market requirements, understanding client willingness to pay, and utilizing decision metrics to predict profitability using neural networks, MCDM, and clustering. The findings highlight that high project diversity limits economies of scale, hinders innovation, restrains creativity, and complicates management, with the proposed decision-making framework aiming to reduce project diversity, enhance reuse, and improve overall efficiency and profitability.[17]

This study investigates the impact of remote work on IT organizations in Bangladesh through a survey of 250 employees, focusing on tools and productivity, work-life balance, and career growth. The methodology involved a dynamically branching questionnaire distributed to 11 IT companies, with data analyzed using Google Data Studio and categorized into three main areas.Key findings include a positive correlation between specialized tools and productivity, improved work-life balance with remote work, and inconclusive evidence on career growth.[18]

The study provides a detailed exploration of Agile methodologies in project management, comparing them with traditional methods and highlighting their implementation challenges and successes. The methodology includes a systematic literature review using academic databases and grey literature sources, with a focus on studies published from

2000 onwards. The findings emphasize the importance of addressing unresolved challenges in Agile implementation and suggest areas for future research to enhance project management practices.[19]

This study explores the integration of traditional project management and agile methodologies for sustainable project management through a systematic literature review. The study analyzes the benefits, challenges, and implementation strategies of a hybrid approach in software project development, highlighting the need for adapting project management practices to meet evolving industry demands. Challenges may have included sourcing relevant literature, interpreting open-ended responses, and synthesizing diverse findings to draw meaningful conclusions.[20]

This study proposes a new process model for incremental asset building in software project management, aiming to increase profitability and sustainability of software companies by building reusable core components based on market segments.The methodology involves a process model with steps like market analysis, project selection, and building/integrating reusable components, described through 10 elements including purpose, procedures, and standards.The novelty lies in integrating market analysis and product line engineering into project management, with findings suggesting that spreading development costs across multiple clients through reusable components can address the problem of low profitability in traditional customized software development.[21]

This study conducted a Systematic Literature Review (SLR) on Business Process Re-engineering (BPR) in the logistics industry, analyzing 408 articles and selecting 22 relevant ones for classification. The methodology involved data collection, analysis, and classification of articles into positive BPR results, negative results, and results in different industries. The study identified key challenges in logistics, emphasized the need for novel tools and methods, and highlighted the importance of evaluating BPR strategies for improved performance and competitiveness. The novelty lies in the comprehensive review of BPR literature in logistics, providing insights into the current state of research and identifying areas for future investigation. [22]

This study examines the impact of Scrum, an Agile methodology framework, on software engineering management through empirical analysis, including literature reviews, case studies, and surveys. Implemented within a software development unit, the research identifies how Scrum artifacts, events, and values enhance team capabilities and address SEM challenges compared to traditional approaches. The study highlights the reasons for Scrum's popularity, the empirical foundation of its principles, and suggests areas for future research in optimizing Agile practices.[23]

This study addresses the need for re-engineering the software development process in a Mexican technology services company, particularly focusing on the risks small- and medium-sized enterprises (SMEs) face in meeting delivery times and project requirements. Using the Hammer and Champ methodology, which involves six stages including analysis and implementation through Business Process Model and Notation (BPMN 2.0), the research demonstrates significant cost savings, time reduction, and improved resource distribution. The paper provides a step-by-step guide and simulation of process re-engineering, showing substantial potential improvements and client satisfaction before operational changes are made.[24]

This study examines cloud data portability, interoperability, and the software architectures of SQL and NoSQL databases. It highlights the differences between SQL, designed for organized data and horizontal expansion, and NoSQL, which supports horizontal scalability for unstructured data. The study reviews various paradigms used by cloud platforms (IaaS, PaaS, SaaS, DBaaS) and compares SQL (e.g., Oracle RDBMS) with NoSQL (e.g., MongoDB) on aspects like scale, performance, availability, consistency, and shading. Findings suggest NoSQL databases are preferable for big data analytics, while SQL databases are more suitable for online transaction processing (OLTP).[25]

This study presents a systematic mapping study on architectural tactics in software architecture literature. Analyzing 91 primary studies from 552 initially identified, it finds that: (i) there is a lack of rigor in characterizing and defining architectural tactics; (ii) most proposed tactics do not adhere to the original definition; and (iii) there is limited industrial evidence of their usage. The study organizes existing literature, highlights research opportunities, and advocates for more systematic definitions and descriptions of architectural tactics to better support the design and analysis of software architectures for quality attributes.[26]

This study introduces the 123-SGR architecture designed to enhance Serious Exergames (SEGs) for physical rehabilitation by incorporating three key functionalities: flexibility/equivalence, complementary, and monitoring. The architecture was applied to adapt an existing SEG, I Blue It, using new interaction devices like MANO-BD, Pressure Belt, and Oximeter for multi modal interaction. Evaluation by health and technology professionals confirmed the potential utility of I Blue It in respiratory rehabilitation, noting the importance of multi modal functionalities. The 123-SGR architecture facilitates the development of flexible, information-rich, and safe SEGs for physical rehabilitation, with future work focusing on integrating AI reasoning and addressing the usability of multiple input devices.[27]

This study describes the implementation and impact of the Architectural Kara work-

shop in a Software Architecture course, designed to teach students how to design, document, and evaluate software architectures through project-based group assignments. Feedback collected via a survey showed that the workshop enhanced skills in identifying non-functional requirements, elaborating assumptions, and collaboration. Despite the potential bias in feedback, the workshop was found to improve conceptual knowledge and collaborative skills. Observations and lessons learned from the workshop are shared, with future iterations planned to include more discussion and design time. Instructors are encouraged to adopt this workshop to enhance instructional activities in Software Architecture courses.[28]

This study explores the nascent field of Quantum Software Architecture (QSA) within Quantum Software Engineering (QSE), emphasizing the need for research in developing architectures for hybrid applications combining quantum and classical components. It provides an overview of quantum computing to make the topic accessible to a wider computer science audience and identifies initial research challenges and opportunities in QSA. The discussions highlight the need for systematic literature reviews, empirical evaluations, and surveys to further develop this preliminary research into a robust field of study.[29]

This study explores the recent trends and current state of AI in software testing by examining and analyzing 20 selected articles out of an initial 90. The study finds that AI techniques, including Machine Learning and Deep Learning, effectively automate tasks such as Test Case Generation and Defect Prediction, improving overall testing performance. It highlights the novelty of AI integration in simplifying testing processes, identifies challenges such as data quality, and suggests future research to further enhance AI-driven testing methodologies.[30]

This study investigates the prevalence of OWASP Top 10 2017 vulnerabilities in web applications from five sectors of the Bangladesh Government using automated tools like Burp Suite, ZAP, and Netscape. Comprehensive black-box testing revealed that the Services and Transportation sectors are the most vulnerable, while data comparisons highlighted each tool's effectiveness in identifying security issues. Future work aims to include more testing tools, expand testing to a broader range of websites using OWASP Top 10 2021 guidelines, and integrate white-box and gray-box testing for deeper vulnerability insights and enhanced software development practices.[31]

This study evaluates three testing tools—Jest, Enzyme, and Cypress—specifically designed for React-based single-page applications. The primary focus is on analyzing the execution time performance and developer-friendliness of these tools within modern development practices like Agile and CI/CD. The study concludes that Enzyme offers the best

balance of integration ease and efficiency for automated front-end testing. However, the study's analysis was limited to literature reviews and work experience due to constraints such as the COVID-19 situation, and it suggests future work to include empirical surveys and additional factors like resource usage for a more comprehensive evaluation.[32]

This study reviews the application of Large Language Models (LLMs) in software testing, covering their use in tasks like unit test case generation, test oracle generation, system test input generation, program debugging, and program repair. It identifies challenges such as achieving high test coverage and addressing the test oracle problem, while noting that LLMs are typically used in limited testing life cycle phases. The study highlights opportunities for research in unaddressed testing stages and advanced prompt design techniques. It aims to guide future research by identifying gaps and potential exploration areas, aiding researchers and practitioners in enhancing software testing practices with LLMs.[33]

This study investigates the development of AI-based software for social media marketing across three countries Romania, France, and Italy focusing on different AI modules like image analysis, sentiment analysis, and audience analysis. The study reveals that cross-cultural factors significantly influence the awareness and adoption of machine learning in professional settings, while factors like gender and job position do not. Data from digital agencies indicate that country of residence, professional experience, and frequency of AI software use are critical determinants. The research underscores the importance of tailoring AI software to cultural and professional profiles and identifies potential development challenges and future research directions to optimize AI capabilities for social media marketing.[34]

This study investigates the relationship between innovation and competition within industries, aiming to identify factors influencing innovation and overall industry welfare. Through surveys and analysis of models and case studies, the study explores how competition impacts innovation across various sectors, emphasizing higher productivity in manufacturing and services under competitive conditions. The findings suggest that effective policy interventions can mitigate challenges and sustain innovation and competition, particularly beneficial for small and medium-sized internet-based firms. The paper advocates for regulatory measures to support innovation while improving product market outcomes, offering practical implications for policymakers and industry stakeholders alike.[35]

This study explores the impact of marketing-mix elements and after-sales service on brand equity dimensions (awareness, perceived quality, loyalty) in Bangladesh's software sector using a developed model. Conducted as a descriptive survey with 70 responses, it

employed SPSS and Amos for correlation, regression, and structural equation modeling analysis. Findings reveal significant relationships between select marketing-mix factors and brand equity dimensions, with perceived quality acting as a mediator. The study underscores the importance of continuous brand equity monitoring and suggests future research directions for refining the model and expanding the scope beyond Bangladesh's context.[36]

This study aims to provide a comprehensive guide for successful Enterprise Resource Planning (ERP) deployment, crucial for enhancing organizational efficiency and meeting customer demands. Through six interviews and a methodology model diagram, it addresses implementation challenges such as effective communication and shared responsibilities. The guidebook offers practical solutions to ensure a smooth ERP deployment process, empowering businesses and their clients with insights and resources for maximizing the benefits of ERP systems.[37]

This study examines the agile implementation approach in Software-as-a-Service (SaaS), using a qualitative case study of a B2B start-up's adoption of HubSpot marketing automation (MA). It highlights iterative adaptations in SaaS features and organizational routines to better align with organizational goals over four years. The research contributes by proposing a framework for managing agile implementation processes, emphasizing processes like sales lead management, content marketing, and customer intelligence as pivotal for leveraging MA software in start-up firms.[38]

Chapter 3

Methodology

The relationship between software production and consumption is covered in this paper, along with how these activities result in surplus for both producers and consumers. Through the integration of software engineering expertise, it investigates methods to enhance both forms of excess. In order to gather information from software developers and users, the methodology entails a thorough literature search and a two-layered survey approach. Literature Search A comprehensive literature search was conducted to identify relevant studies and theories related to the production and consumption of software and their impact on consumer and producer surplus. The search utilized several databases, including ScienceDirect, IOPscience, IEEExplore, Repository, E-Repository, ResearchGate, SSRN, JIMF, SpringerLink, MDPI, AgEconsearch, AJOL, and Policy Commons. The relevant papers were identified by searching the metadata (title, abstract, keywords) using specific keyword search queries. The keywords were derived from the research questions and included: Requirement Engineering Project Process Architecture Quality Assurance, Software Marketing. In order to comprehend the dynamics of consumer and producer excess in Bangladesh's software industry, this thesis employs a two-layered survey strategy. Examining how producers and consumers interact in the market with an emphasis on how each party achieves surplus through various pricing methods is the goal.[34]

3.1 Survey Design

The purpose of the study was to gather information from two different groups: users and manufacturers of software. Businesses and individual users who buy software products and services make up the customers, while software corporations and developers make up the producers. The poll was designed to gather information about purchase patterns, price strategies, and perceived value from both viewpoints. The research was consists of few interviews from different organizations in Bangladesh where we have analyzed the

data in the respective of other studies[34]

3.2 Two-Layered Approach

1. Producer study: This section of the study focused on the cost structures and pricing strategies of software producers. Questions regarding production costs, pricing strategies, and profit margins were posed to producers. Among the key topics of investigation were: -

The price of creating and maintaining software.

- Pricing techniques applied while offering software to various market niches.

- Elements affecting the determination of prices.

- The belief about the value and readiness to pay that customers have for their items.

2. Consumer Survey: This layer focused on software users, seeking to comprehend their software-related purchase patterns and the benefits they receive from using software. Customers were questioned regarding their purchasing histories, willingness to pay, and degree of pleasure. The cost of software goods and services was one of the main research topics.

- The thought put into the usefulness and worth of the purchased program.

- Elements affecting the judgments they make about purchases.

- Contentment with the cost in light of the advantages obtained.

3.3 Data Collection

A representative sample from the producer and consumer groups was guaranteed by the use of online questionnaires and interviews in the administration of the surveys. To promote high response rates, the online surveys were made to be succinct and easy to use. In order to obtain more in-depth understanding and qualitative data, interviews were held with significant players in the software sector.To classify and compare features, data taken from each article were entered into an Excel computer worksheet. Utilizing Excel sheets, the dataset was computed, followed by quantitative analysis and the calculation of descriptive statistics, the majority of which were frequency tables for the categorical variables. In order to address the research questions that served as the project's compass, the study summarized the results from the chosen papers. [9] The relationship between

production costs and consumer pricing was investigated using the collected data. The primary objective of the analysis was to identify the excess generated at both ends: The producer surplus is the difference between a producer's asking and selling prices for software. This surplus demonstrates the producers' profit margin.In our another paper we have designed our paper in this manner. [9]

The gap between what customers are prepared to pay and what they actually pay is known as the "consumer surplus." The extra value that customers obtain over and above the cost is reflected in this surplus.

3.3.1 Consumer Surplus

Consumer surplus is the difference between the maximum price a consumer is willing to pay and the actual price they pay. If a consumer is willing to pay more than the current asking price, they derive more benefit from the purchased product than the amount paid.

For this study, I focused on De-soft, a company I have studied extensively over the last couple of years. De-soft's journey over the past 14 years was analyzed, including their requirement gathering and analysis, product line architecture, process, marketing, product selection process, project handling, service and support, development, communication, customization, quality maintenance, and business process re-engineering.

De-soft has sold their products to more than 120 companies in a scattered manner, initially without following any product line architecture or engineering process. Currently, they have 40+ regular clients who provide them with regular service charges.

Main Product: Club-ERP

- **Base Price**: BDT 20,00,000.00

- **Sale Price**: BDT 10,00,000.00

- **Market Price**: BDT 18,00,000.00

- **Number of Clients**: 18

De-soft has been able to decrease the product price to increase consumer surplus. By following a software product line architecture and implementing a single product multiple client concept, they have achieved significant success.

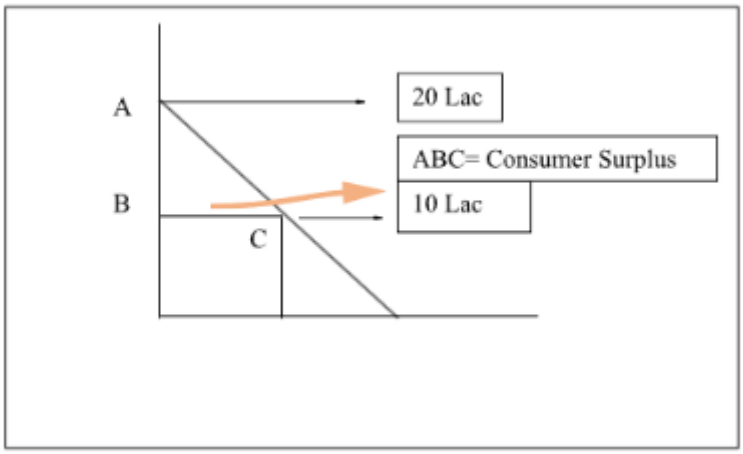


Figure 3.1: Consumer Surplus

Another Product: Garment HR and Payroll

- **Base Price**: BDT 4,00,000.00

- **Sale Price**: BDT 2,50,000.00

- **Market Price**: BDT 1,50,000.00

- **Number of Clients**: 8

In this case, De-soft has not been able to reduce the price below the market price, resulting in lower consumer satisfaction and difficulty in selling the product.

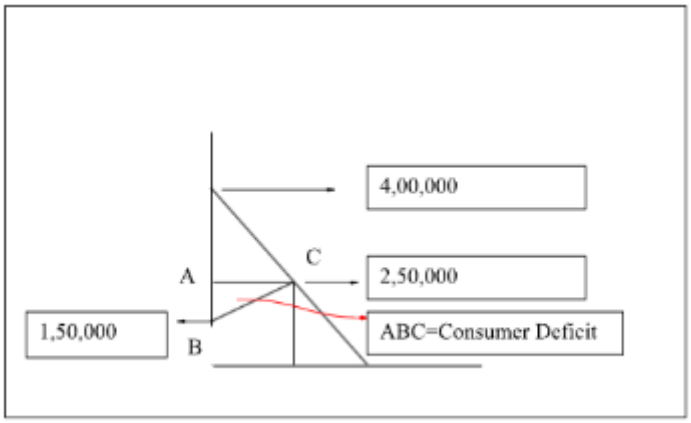


Figure 3.2: Consumer Deficit

3.3.2 Producer Surplus

Producer surplus is the difference between the amount that a producer receives for a good and the minimum amount they would be willing to accept. It measures the benefit that the producer receives for selling the goods in the market.

Main Product: Club-ERP

- **Base Price**: BDT 20,00,000.00

- **Sale Price**: BDT 10,00,000.00

- **Market Price**: BDT 18,00,000.00

- **Production and Customization Price**: BDT 2,00,000.00

- **Producer Surplus**: BDT 8,00,000.00

De-soft's main product, Club-ERP, demonstrates a significant producer surplus. By reducing the sale price from the base price and market price, De-soft has managed to create a competitive advantage while maintaining a substantial producer surplus. This strategy allows them to attract more clients and sustain their business operations effectively.

In summary, the data collection results highlight the importance of strategic pricing and product line architecture in enhancing both consumer and producer surplus. De-soft's experience with their products, particularly Club-ERP, illustrates how integrating software engineering principles can lead to increased market success and sustainability.The surveys revealed that producers often sell software at a higher price than their production cost, generating a significant producer surplus. Simultaneously, consumers purchase software at prices lower than their perceived value, resulting in a consumer surplus. This two-layered approach highlights a win-win situation where both producers and consumers achieve a surplus, contributing to a healthy and sustainable software market in Bangladesh.

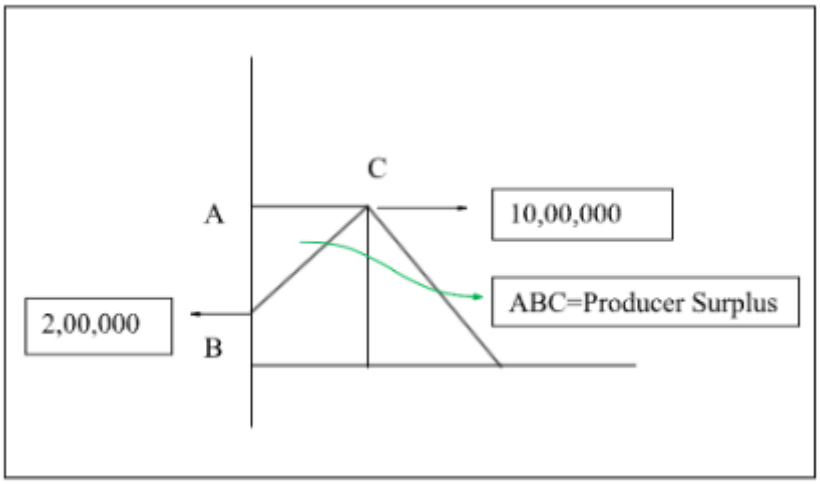


Figure 3.3: Producer Surplus

The methodology provides a comprehensive understanding of the economic interactions in the software industry, offering valuable insights for optimizing pricing strategies and enhancing market efficiency. By analyzing the surplus from both producer and consumer perspectives, the thesis contributes to a balanced view of the software industry's economic dynamics.

Chapter 4

Result Analysis

4.1 Balancing Consumer and Producer Surplus

In the integration of software engineering principles with microeconomics concepts, the focus is on achieving a balance where both the producer and consumer are satisfied. The real-time experience with De-soft, a software company in Bangladesh, illustrates how strategic implementation of software engineering practices can lead to such a balance. Here is an analysis of how this balance can be achieved, with specific attention to pricing strategies and quality assurance.

4.1.1 Consumer Surplus: Enhanced Satisfaction at Lower Prices

Consumers derive surplus when they are able to purchase products at a price lower than what they are willing to pay. De-soft's approach of pricing their products competitively, while maintaining high quality, has led to significant consumer surplus.

For example, the Garment HR and Payroll software:

Base Price: BDT 4,00,000

Sale Price: BDT 2,50,000

Market Price: BDT 1,50,000

They offering the product at BDT 2,50,000, consumers gain a surplus of BDT 1,00,000, as they pay less than the market price for a high-quality product. Although this pricing strategy does not maximize consumer surplus to the same extent as other products, it still makes the product accessible and increases customer satisfaction.

4.1.2 Producer Surplus: Maximizing Profit Margins

Producer surplus occurs when the sale price of a product exceeds the production cost. De-soft's effective cost management and strategic pricing have led to substantial producer surplus.

For instance, the Point of Sales (POS) software:

Production and Customization Cost: BDT 50,000

Sale Price: BDT 2,00,000

The producer surplus here is BDT 1,50,000, achieved by keeping production costs low and setting a competitive sale price. This surplus allows De-soft to reinvest in further product development and quality improvement, creating a sustainable business model.

The producer surplus here is BDT 8,00,000, achieved by keeping production costs low and setting a competitive sale price. This surplus allows De-soft to reinvest in further product development and quality improvement, creating a sustainable business model.

4.2 Strategic Insights for Balancing Surplus

1. Requirement Engineering: Properly gathering and analyzing requirements ensures that products meet consumer needs and are delivered on time. This reduces development costs and delays, increasing both consumer and producer surplus.

4.2.1 Analysis of Product Release Timelines

The table below (Table 1) illustrates the estimated and actual time taken for the deployment of various products developed by De-soft. This analysis highlights the discrepancies between the planned timelines and the actual time required for product release.

Product	Estimated Time (month)	Actual Required Time (month)
ERP	12	14
Education	2	8
Garments	2	8
HRM	3	10
Distribution	2	7
Billing & Invoicing	2	12
POS	2	2

Table 4.1: Estimated and Actual Time for Various Products

4.3 Key Observations

1. **ERP**:

Estimated Time: 12 months

Actual Time: 14 months

Analysis: Although the ERP project exceeded the estimated time by 2 months, the deviation was relatively small, suggesting a better alignment of requirement gathering and project management practices compared to other projects.

2. **Education, Garments, and Billing Invoicing**: These projects show substantial time overruns (actual time being 4 to 6 times the estimated time), indicating severe issues in requirement estimation and project scope management.

3. **POS**: **Estimated Time**: 2 months **Actual Time**: 2 months **Analysis**: The POS project met its estimated timeline, showcasing an instance where accurate estimation and effective project management were achieved.

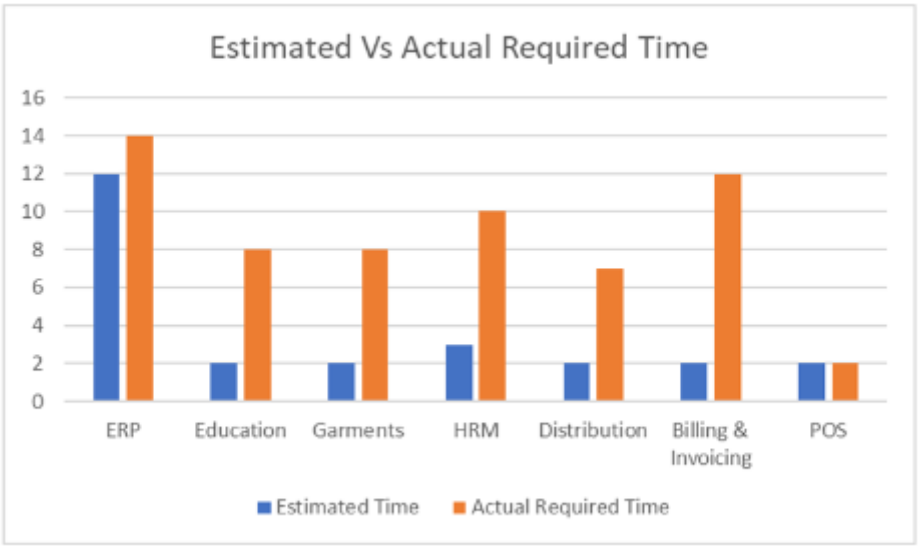


Figure 4.1: Graphical Representation of Estimated and Actual Time

- **Case Study**: For the ERP development, thorough requirement analysis and risk management led to the project being completed within 14 months, close to the estimated 12 months. This adherence to timelines increased consumer trust and satisfaction, while controlled costs enhanced producer surplus.

2. **Assurance of Quality**: Superior products boost customer willingness to pay and decrease the need for rework. Quality assurance procedures guarantee that the software satisfies industry standards, which raises customer satisfaction and lowers producers' long-term expenses. As an illustration, De-soft's monthly maintenance and customisation fees are warranted by the superior quality of their output, which generates steady income and a rise in producer surplus.

3. **Architecture for Software**: Scalability and cost savings are possible when a software product line architecture is put into practice. With this strategy, producers can offer more clients at lower prices, increasing surplus for both consumers and producers.

- **Case Study**: The advantages of a scalable software architecture are illustrated by the effective implementation of Club-ERP to several clients at a reasonable cost.

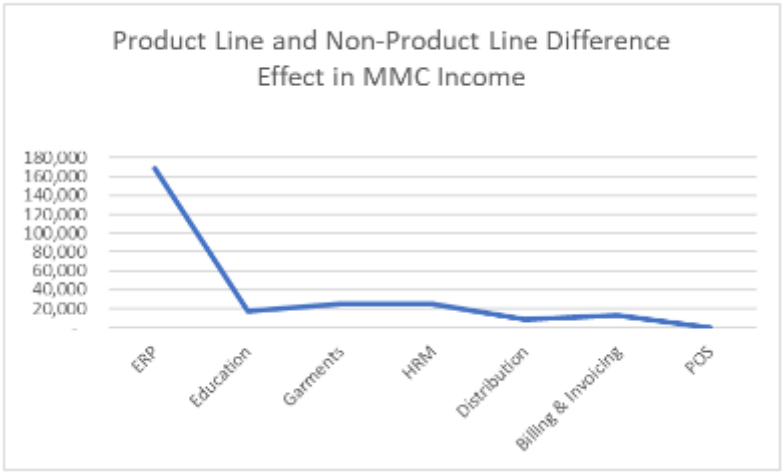


Figure 4.2: Relation Between Product Line and Non-Product line in MMC income

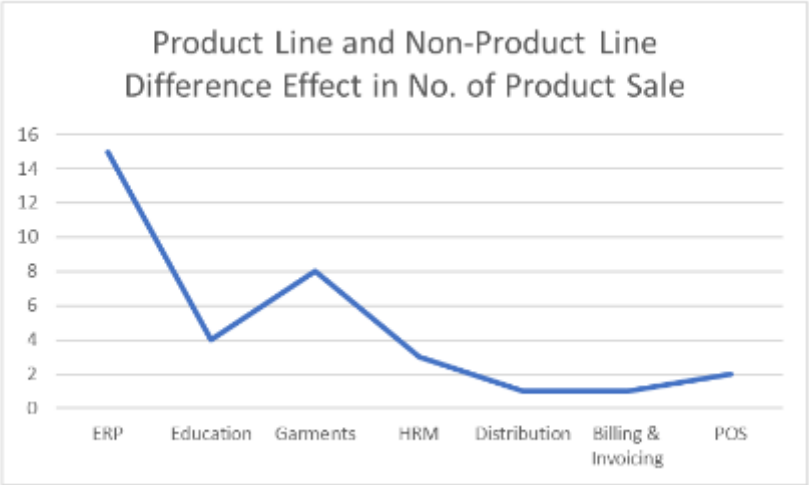


Figure 4.3: Relation Between Product Line and Non-Product line in No. of Product Sale

A more organized product line architecture has replaced the non-product line method in De-soft's approach to product development. The quantity of products sold as well as the revenue from Monthly Maintenance Charges (MMC) have both been significantly impacted by this change. The links shown in Figures 5 and 6 are further explained in the analysis that follows.

The difference between the MMC income from items developed with a product line approach and those developed without one is seen in Figure 5.

The difference between the quantity of products sold through a non-product line method and one using a product line strategy is depicted in Figure 6.

The Said Company's Product List 1. **ERP-Enterprise Resource Planning** (Desktop, Web and Mobile App)

2. Total Integrated Central Accounts
3. Budget/Assumption
4. Point of Sales (POS)
5. Human Resource Management System (HRMS)
6. Cash Card (Debit Card System in Own Premises)
7. SMS Integration with Software
8. E-mail Integration with Software
9. Fixed Asset Management System
10. Fixed Asset Maintenance System

11. MRR Base Payment System

12. Smart Kitchen Display KOT Print System

13. Life Donor Member Display System

14. Executive Committee Members In/out Board

15. Membership Management System

16. Banquet Hall Booking System

17. Cineplex Management System

18. Bakery Production, Cost, Stock Sales System

19. Inter Department Sales Transfer System

20. Event Ticket Sales Integration

21. Housie Sheet Sales Integration

22. Rest House Room Booking System

23. Employees Performance Management System

24. Member Access Control System with Visual Information

25. Members Movement Dash Board

26. Affiliated Club Member Movement Register

27. Online Payment Gateway Integrate with Accounts

28. Web Integration

29. Apps Integration

30. Food Order Apps

31. Voting Solutions

32. School ERP (Integrated General Ledger, Fixed Asset, Fees Management, System Generated Bank Money Receipt, Sales, Inventory, Result Management for Both Bangla English Medium, SMS Management, Library Management, HRM, Payroll Attendance, Student Attendance by Card Punch)

33. School Management System (Integrated General Ledger, Fixed Asset, Fees Management, Sales, Inventory)

34. ERP for Game Zone

35. Fixed Asset Management System

36. Fixed Asset Maintenance Management and Administration

37. Cineplex Management with Seat Plan

38. Bakery Production, Recipe, Inventory Sales System

39. Restaurant Management System

40. Integrated Accounts with Sales Inventory

41. Daily Basis Cash System

42. MRR Base Bank Payment Voucher and Cheque Print

43. Cash Card Solutions (Debit Card System in Own Premises) Human Resource Information System (HRIS)

44. Supplier Chain Management System

45. Loan Calculation

46. Tax Calculation

47. Factory Wages (Daily or Working Base)

48. Sales Distribution System

49. Point of Sales

50. Library Management System

51. Voting Software Solution (OMR AND OCR Technology)

52. Club App (Both Android iPhone)

53. Restaurant Waiter App

54. Member App

55. School Tuition Fees App

56. Online Payment Gate

57. Election App with Live Voting Result A lengthy product line suggests that they put in a lot of effort but are unsuccessful. They located a project and went to work. However, this work lacks the study base and product line engineering background.

4. **Project Management**: Well-managed projects guarantee timely and cost-effective completion. This lowers expenses and delays while raising producer surplus and enhancing customer satisfaction.

- **Case Study**: Projects like POS, which were finished on schedule, demonstrate the

value of effective project management in obtaining timely delivery and cost control.

5. **Marketing**: Demand and product awareness are raised by strategic marketing. More customers can be drawn in by a potent marketing plan, which will raise sales volume and producer surplus.

- **Observation**: De-soft's clientele was restricted by their dispersed marketing initiatives. A targeted marketing approach could improve market share and profitability.

6. **Business Process Re-Engineering (BPR)**: The necessity for expensive re-engineering is decreased by careful demand research and planning. Producers can meet consumer expectations and sustain profitability by minimizing re-engineering expenditures.

- **Case Study**: Higher expenses and unsatisfied customers resulted from a billing software project's failure to foresee the need for re-engineering. These problems might have been avoided with proper BPR, protecting surplus for both parties.

The significant need for better project estimates and management procedures at De-soft is shown by the comparison between estimated and actual deployment times. Through an emphasis on precise requirement engineering, risk mitigation, and effective project management techniques, the organization may more effectively align projected and real deadlines, thereby mitigating delays and overspending. This alignment is essential for optimizing producer and consumer surplus, guaranteeing prompt delivery of high-quality goods that satisfy customers, and preserving the business's profitability..

Improving Consumer and Producer Surplus through the Integration of Software Engineering Body of Knowledge

My real-world experience working with De-soft and their clients over the past few years has allowed me to pinpoint a number of crucial areas where both producers and consumers stand to gain a great deal. A strong body of knowledge in software engineering can be integrated to improve surplus for both producers and consumers. The study's conclusions and outcomes are covered in detail in the parts that follow.

4.3.1 Quality Over Price: The Consumer Perspective

Observation: Consumers consistently prioritize quality over price. While price remains a concern, the willingness to pay more for better quality indicates that consumers value reliable and high-performing software solutions.

Implications**:**

Consumer Satisfaction: High-quality software products lead to greater consumer satisfaction. This satisfaction translates into repeat business, positive word-of-mouth, and long-term loyalty.

Consumer Surplus: When consumers receive a high-quality product at a price lower than their maximum willingness to pay, they experience a higher consumer surplus. For instance, De-soft's Club-ERP software, with a market price of BDT 18,00,000 but sold at BDT 10,00,000, provides consumers with a surplus of BDT 8,00,000.

4.3.2 The Role of Producers

: Enhancing Delivery and Satisfaction **Observation**: The delivery of quality products and services is critical for consumer satisfaction, which in turn affects producer success. Market competition drives producers to improve their offerings continuously.

Implications:

Producer Surplus: Producers benefit from cost-effective production and higher sales prices. For example, De-soft's Club-ERP software has a production and customization cost of BDT 2,00,000 but is sold at BDT 10,00,000, yielding a producer surplus of BDT 8,00,000.

Market Competitiveness: Effective integration of software engineering principles helps producers maintain competitiveness. This includes thorough requirement analysis, efficient project management, and robust quality assurance processes.

4.3.3 Requirement Engineering

: Foundation for Success Observation: Many companies in Bangladesh, including De-soft, initially started software development using agile methods without solid requirement engineering, leading to delays and cost overruns.

Case Study: **HRM Software**: Estimated completion time was 3 months, but actual time required was 10 months, with costs increasing more than threefold. **ERP Software**: With comprehensive requirement engineering, the estimated time was 12 months, and actual completion time was 14 months.

Implications: Accurate Estimations: Proper requirement engineering ensures ac-

curate time and cost estimations, reducing delays and budget overruns. **Consumer Satisfaction**: On-time project completion enhances consumer satisfaction as they can start benefiting from the software without delays.

4.3.4 Software Architecture: Moving from Legacy to Modern Approaches

Observation: De-soft transitioned from a legacy approach to a more structured software engineering process, resulting in better product outcomes.

Implications: Product Line Efficiency: Structured product line architecture leads to more efficient development and maintenance processes, boosting both MMC income and sales. **Scalability and Reliability**: Modern software architecture enhances scalability and reliability, which are crucial for long-term consumer and producer surplus.

4.3.5 Quality Assurance

: **A Pillar for Consumer and Producer Surplus Observation**: Quality is a primary concern for consumers, who are willing to pay more for reliable and efficient software. **Implications**:

Reduced Rework and Wastage: High-quality software minimizes the need for rework and reduces wastage, which translates into cost savings for producers.

Enhanced Productivity: Quality software improves productivity and reduces unplanned work for consumers, adding to their surplus.

Regular Maintenance Income: For De-soft, quality assurance has led to a steady income stream through monthly maintenance charges and customization fees, benefiting both consumers and producers.

4.3.6 Marketing Strategies: Expanding Reach and Impact Observation

: Traditional business expansion methods have limited the market reach of many software firms, including De-soft, which lacks a dedicated marketing team.

Implications: **Increased Market Penetration**: Effective marketing strategies and

a dedicated team can significantly increase market penetration and client acquisition. **Sustainable Growth**: Good marketing policies and methods ensure sustainable growth, contributing to both consumer and producer surplus.

Business Process Re-Engineering (BPR): Managing Change and Adaptation Observation: Software re-engineering is common but challenging, often leading to increased costs and customer dissatisfaction when not managed properly.

Case Study: Billing Software: Initially completed within the estimated duration, but massive requirement changes later necessitated re-engineering, resulting in higher costs and customer dissatisfaction.

Implications: **Cost Management**: Effective BPR minimizes the need for costly re-engineering, preserving producer surplus. **Customer Retention**: Proper management of BPR ensures customer satisfaction and retention, avoiding the loss of clients due to dissatisfaction.

The integration of software engineering principles with microeconomic strategies can create a balanced ecosystem where both producers and consumers are happy. By focusing on requirement engineering, quality assurance, scalable software architecture, effective project management, strategic marketing, and proper business process re-engineering, software companies like De-soft can achieve significant consumer and producer surplus. This balanced approach not only enhances customer satisfaction but also ensures sustainable profitability for the producer. Including a large database of information about software engineering greatly increases surplus for both producers and consumers. Software firms can improve their competitiveness, profitability, and customer happiness by concentrating on quality assurance, precise requirement engineering, successful marketing tactics, and appropriate business process re-engineering. The transition of De-soft from an antiquated method to a methodical engineering procedure provides an insightful case study that highlights the advantages of these approaches.

Chapter 5

Conclusion

This thesis on consumer and producer surplus in the software industry of Bangladesh makes several significant contributions to both academia and industry practice. Its primary contribution lies in the integration of economic principles into software development and commercial strategy. Historically, the software sector has emphasized technical expertise while often neglecting economic theories. By providing a systematic framework that enables software companies to make well-informed decisions regarding product development, pricing strategies, and resource allocation based on economic theory, this study addresses this critical knowledge gap. The study encompasses six design facets: requirement engineering, quality assurance, software architecture, software marketing, business process reengineering (BPR), and project management. The thesis shows the importance of economic factors in ensuring the long-term sustainability and viability of software companies. Businesses can mitigate risks associated with market fluctuations and competitive pressures by applying economic principles such as strategic pricing and efficient resource allocation. This proactive approach not only enhances profitability but also strengthens a company's capacity for innovation and adaptability in response to changing market conditions. Moreover, this thesis offers valuable insights and practical policy recommendations for policymakers in Bangladesh. It advocates for regulations that create a favorable environment for the software industry, including the protection of intellectual property rights, the promotion of fair competition, and the encouragement of innovation. By fostering such an environment, policymakers can drive growth and competitiveness in the industry, contributing to the overall economic development of the nation. Academically, this thesis advances understanding through the unique application of economic theories to the Bangladeshi software sector. It provides case studies and empirical analyses demonstrating how economic principles can be practically applied to enhance business outcomes within the industry. By bridging the gap between theory and practice, the thesis serves as a valuable resource for scholars, practitioners, and policymakers interested in the economics of the software industry and its sustainable growth. In summary, this thesis offers more than theoretical insights; it presents practical applica-

tions that have the potential to significantly enhance the operational efficiency, financial success, and long-term viability of software companies in Bangladesh. By incorporating economic principles into decision-making processes, businesses can better navigate complexities, seize market opportunities, and support the development of a competitive and thriving software industry ecosystem in the nation. This integrated approach promises to not only improve individual business outcomes but also contribute to the broader economic progress of Bangladesh.

5.0.1 Sustainability

The sustainability of this project on consumer and producer surplus in the software industry of Bangladesh is multi-faceted, involving economic, social, and environmental dimensions. Each aspect is crucial in ensuring the long-term viability and positive impact of the project on the software industry and the broader national economy.

Economic Sustainability

Framework for Decision-Making: By integrating economic principles into software development and commercial strategies, the project provides a robust framework for making informed decisions. This helps software companies optimize product development, pricing strategies, and resource allocation, leading to enhanced profitability and market resilience.

Risk Mitigation: The application of strategic pricing and efficient resource allocation reduces risks associated with market fluctuations and competitive pressures. This proactive approach ensures that companies can maintain their financial health and continue to invest in innovation and growth.

Policy Recommendations:The project's recommendations for favorable regulations, such as the protection of intellectual property rights and the promotion of fair competition, create a supportive environment for the software industry. These policies are essential for sustaining economic growth and competitiveness in the sector.

Social Sustainability

Skill Development: Emphasizing the importance of requirement engineering, quality assurance, software architecture, and project management fosters a culture of continuous learning and skill development. This helps build a skilled workforce capable of driving the industry's growth and innovation.

Industry Growth: By supporting policies that encourage innovation and fair com-

petition, the project contributes to creating job opportunities and promoting social mobility within the software sector. This, in turn, supports the broader socio-economic development of Bangladesh.

Collaboration and Knowledge Sharing: The project bridges the gap between academic theory and industry practice, facilitating collaboration between scholars, practitioners, and policymakers. This knowledge-sharing ecosystem is vital for the sustainable development of the software industry.

5.0.2 Feasibility

This project aims to investigate consumer and producer surplus in the software industry of Bangladesh by integrating economic principles into software development and commercial strategy. The focus areas include requirement engineering, quality assurance, software architecture, software marketing, business process reengineering (BPR), and project management. The feasibility of this project will be assessed based on several criteria, including economic viability, technical feasibility, market demand, and policy support.

Economic Feasibility

Cost-Benefit Analysis: The implementation of economic principles into software development can lead to cost savings through strategic pricing and efficient resource allocation. Potential increase in profitability for software companies due to informed decision-making and improved operational efficiency. Initial investment required for training and developing the economic framework can be offset by long-term gains.

Funding and Resources: Availability of funding from academic institutions, industry stakeholders, and government grants. Allocation of resources, including skilled personnel and technological infrastructure, is feasible within the context of Bangladesh's growing IT sector.

Technical Feasibility

Integration of Economic Principles: The project's framework aligns well with existing software development processes and can be integrated without significant disruptions. Utilization of case studies and empirical analysis ensures that the proposed economic models are grounded in real-world applications.

Implementation Challenges: Potential resistance to change from traditional software development practices to a more economically-driven approach.Need for continuous

training and development programs to equip professionals with the necessary economic knowledge.

Market Demand

Industry Need: The software industry in Bangladesh is evolving, with an increasing need for strategic decision-making to sustain growth and competitiveness. Adoption of economic principles can address current gaps in resource allocation, pricing strategies, and long-term planning.

Competitor Analysis: Limited integration of economic principles in the current market presents an opportunity for early adopters to gain a competitive edge. Potential to set a precedent in the industry, encouraging other companies to follow suit.

5.0.3 Social and Environmental Impact

The social impact of integrating economic principles into the software industry of Bangladesh is multifaceted, addressing various dimensions such as employment, skill development, and societal welfare.

Social Impact

The social impact of integrating economic principles into the software industry of Bangladesh is multifaceted, addressing various dimensions such as employment, skill development, and societal welfare.

Employment Generation: The adoption of economic strategies in the software industry is likely to drive business growth, leading to increased job creation. As software companies become more profitable and expand, they will require a larger workforce, thus contributing to the reduction of unemployment rates in Bangladesh.

Skill Development and Education: The emphasis on economic principles within the software industry will necessitate a workforce equipped with both technical and economic knowledge. This demand will encourage educational institutions to offer specialized programs, thereby enhancing the skill sets of future professionals. Additionally, current employees will benefit from continuous learning opportunities and professional development, fostering a culture of lifelong learning.

Societal Welfare: A thriving software industry can have a ripple effect on societal welfare. Increased employment and higher incomes will improve the standard of living for many individuals and families. Moreover, the industry's growth can stimulate other sectors of the economy, creating a more vibrant and diverse economic landscape.

Gender Equality: The expansion of the software industry presents opportunities for promoting gender equality. By encouraging the inclusion of women in the workforce and supporting initiatives that target gender disparity, the industry can contribute to a more equitable society.

Innovation and Entrepreneurship: By applying economic principles, the industry will foster an environment that encourages innovation and entrepreneurship. This can lead to the development of new products and services, further enhancing the industry's contribution to social progress and economic development.

Environmental Impact

While the software industry is generally perceived as having a lower environmental footprint compared to traditional manufacturing industries, it is not without its environmental impacts. The thesis indirectly addresses these impacts through the principles it advocates.

Resource Efficiency: The application of economic principles such as efficient resource allocation can lead to more sustainable use of resources within the software industry. Companies can optimize their operations to minimize waste, reduce energy consumption, and use resources more judiciously, thereby lowering their environmental footprint.

Sustainable Practices: As companies adopt strategic pricing and resource allocation, there is potential for increased investment in sustainable practices. This could include adopting green technologies, implementing energy-efficient systems, and reducing the carbon footprint of operations. These measures contribute to the broader goal of environmental sustainability.

Digital Transformation: The growth of the software industry can drive digital transformation across various sectors, leading to more efficient and less resource-intensive processes. For instance, digital solutions can replace paper-based systems, reducing deforestation and paper waste. Moreover, the promotion of remote work can decrease the carbon emissions associated with commuting.

Environmental Awareness: By incorporating economic principles into business strategies, companies can also prioritize corporate social responsibility (CSR) initiatives that focus on environmental sustainability. This could involve educating employees and stakeholders about environmental issues and encouraging practices that support ecological balance.

Waste Management: The software industry produces electronic waste (e-waste) as hardware becomes obsolete. Economic principles can guide companies in developing effective e-waste management strategies, promoting recycling and the safe disposal of electronic components.

5.0.4 Ethics

The research presented in this thesis on consumer and producer surplus in the software industry of Bangladesh adheres to stringent ethical standards to ensure the integrity, reliability, and societal benefit of its findings. Key ethical considerations addressed in this study include.

Informed Consent and Confidentiality: When conducting case studies and empirical analyses, it is essential to obtain informed consent from all participants, ensuring they are fully aware of the study's purpose, procedures, and any potential risks or benefits. Participants' privacy and confidentiality must be safeguarded, with data anonymized to prevent the identification of individuals or organizations.

Honesty and Transparency: The research process, including data collection, analysis, and reporting, must be conducted with utmost honesty and transparency. Any potential conflicts of interest should be disclosed, and the limitations of the study should be clearly stated to provide a balanced and accurate representation of the findings.

Equity and Fairness: The research and its applications must promote equity and fairness within the software industry. This includes advocating for policies that ensure fair competition, protect intellectual property rights, and provide equal opportunities for businesses of all sizes to thrive.

Policy Implications and Societal Impact: The policy recommendations provided must be developed with careful consideration of their broader societal impact. Policies should aim to create a regulatory environment that not only supports the growth of the software industry but also aligns with the broader economic and social goals of Bangladesh, such as reducing inequality and promoting sustainable development.

5.0.5 Project Summary

This thesis explores consumer and producer surplus in the Bangladeshi software industry, integrating economic principles into software development and commercial strategy. Traditionally, the sector has prioritized technical skills over economic theories. The study

provides a framework for informed decision-making in product development, pricing, and resource allocation, focusing on six design facets: requirement engineering, quality assurance, software architecture, software marketing, business process re-engineering (BPR), and project management. Key findings emphasize the importance of economic factors for the sustainability and profitability of software companies. Applying strategic pricing and efficient resource allocation helps mitigate market risks and fosters innovation. The thesis offers policy recommendations for Bangladeshi policymakers to protect intellectual property, promote fair competition, and encourage innovation, aiding in industry growth and national economic development. By applying economic theories to real-world scenarios, the thesis bridges the gap between theory and practice, providing valuable insights for scholars, practitioners, and policymakers. Overall, it presents practical applications to enhance the efficiency, financial success, and long-term viability of software companies in Bangladesh, contributing to a thriving industry ecosystem.

5.0.6 Future Work

Future research can expand upon this thesis by conducting longitudinal studies to assess the sustained impacts of implementing economic principles in various software companies across Bangladesh. Additionally, exploring sector-specific applications, such as cyber security and artificial intelligence, could reveal tailored economic strategies for different sub-sectors. Evaluating policy recommendations through collaboration with policymakers and industry stakeholders will be crucial to understanding their real-world impact and refining them for optimal industry support. Further integration of economic principles with emerging technologies like AI and blockchain could enhance decision-making processes, while developing educational programs would help bridge theoretical knowledge with practical industry needs. Comparative studies with other developing and developed countries' software industries could provide valuable insights and benchmarks for elevating Bangladesh's global competitiveness.

Bibliography

[1] D. Gobov and I. Huchenko, "Requirement elicitation techniques for software projects in ukrainian it: an exploratory study," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pp. 673–681, IEEE, 2020.

[2] A. Ahamed, N. Sadman, T. A. Khan, M. I. Hannan, F. Sadia, and M. Hasan, "Automated testing: Testing top 10 owasp vulnerabilities of government web applications in bangladesh," *ICSEA 2022*, p. 56, 2022.

[3] M. T. Hasana, F. Akterb, S. Alamc, and M. Hasand, "Quality assurance model for lms to facilitate education and equality,"

[4] M. T. Hasan, N. M. A. Bakar, N. Nahar, M. Hasan, and M. Rokonuzzaman, "A hybrid approach to overcome requirements challenges in the software industry," 2023.

[5] F. Sadia, M. Hasan, N. Nahar, and M. Rokonuzzaman, "A new process model of incremental asset building for software project management," in *2021 IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 86–90, IEEE, 2021.

[6] M. M. Syeed, A. Shihavuddin, M. F. Uddin, M. Hasan, and R. H. Khan, "Outcome based education (obe): Defining the process and practice for engineering education," *IEEE Access*, vol. 10, pp. 119170–119192, 2022.

[7] S. Tahsin, A. Munim, M. Hasan, N. Nahar, and M. Rokonuzzaman, "Market analysis as a possible activity of software project management," in *2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 180–183, IEEE, 2018.

[8] S. A. M. TOUSHIF HOSSAIN, M. Sakib, R. Hossain, M. N. Nahar, and M. Hasan, "Ai based solutions for software testing in bangladeshi software companies,"

[9] S. R. Akthar, M. R. Islam, N. Islam, and F. Sadia, "Enhancing game usability: A framework for small-to-medium-sized game development businesses," 2023.

[10] L. N. Raha, A. W. Hossain, T. Faiyaz, M. Hasan, N. Nahar, and M. Rokonuzzaman, "A guide for building the knowledgebase for software entrepreneurs, firms, and professional students," in *2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 165–171, IEEE, 2018.

[11] P. Bambazek, I. Groher, and N. Seyff, "Requirements engineering for sustainable software systems: a systematic mapping study," *Requirements Engineering*, vol. 28, no. 3, pp. 481–505, 2023.

[12] L. Duboc, B. Penzenstadler, J. Porras, S. Akinli Kocak, S. Betz, R. Chitchyan, O. Leifler, N. Seyff, and C. C. Venters, "Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow," *Requirements Engineering*, vol. 25, pp. 469–492, 2020.

[13] L. Chazette and K. Schneider, "Explainability as a non-functional requirement: challenges and recommendations," *Requirements Engineering*, vol. 25, no. 4, pp. 493–514, 2020.

[14] M. Islam, F. Khan, S. Alam, and M. Hasan, "Artificial intelligence in software testing: A systematic review," in *TENCON 2023-2023 IEEE Region 10 Conference (TENCON)*, pp. 524–529, IEEE, 2023.

[15] M. T. J. Ansari, D. Pandey, and M. Alenezi, "Store: Security threat oriented requirements engineering methodology," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 191–203, 2022.

[16] A. Ahamed, T. A. Khan, N. Sadman, M. I. Hannan, N. Nahar, and M. Hasan, "The rise of remote project management-a new norm?: A survey on it organizations in bangladesh.," in *ICSOFT*, pp. 518–525, 2023.

[17] M. Islam, F. Khan, M. Hasan, F. Sadia, and M. Hasan, "Impact of covid-19 on the factors influencing on-time software project delivery: An empirical study.," in *ENASE*, pp. 554–561, 2023.

[18] J. Leong, K. May Yee, O. Baitsegi, L. Palanisamy, and R. K. Ramasamy, "Hybrid project management between traditional software development lifecycle and agile based product development for future sustainability," *Sustainability*, vol. 15, no. 2, p. 1121, 2023.

[19] E. C. Daraojimba, C. N. Nwasike, A. O. Adegbite, C. A. Ezeigweneme, and J. O. Gidiagba, "Comprehensive review of agile methodologies in project management," *Computer Science & IT Research Journal*, vol. 5, no. 1, pp. 190–218, 2024.

[20] S. R. Akthar, F. Sadia, and M. Hasan, "Strategies to overcome slow growth in software firms: Project diversity perspective," tech. rep., EasyChair, 2023.

[21] N. Li and H. Nazif, "Systematic literature review on business process re-engineering approaches in logistics," *Kybernetes*, vol. 51, no. 10, pp. 3009–3024, 2022.

[22] K. Bhavsar, V. Shah, and S. Gopalan, "Scrum: An agile process reengineering in software engineering," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 3, pp. 840–848, 2020.

[23] L. P. Lopez-Arredondo, C. B. Perez, J. Villavicencio-Navarro, K. E. Mercado, M. Encinas, and P. Inzunza-Mejia, "Reengineering of the software development process in a technology services company," *Business Process Management Journal*, vol. 26, no. 2, pp. 655–674, 2020.

[24] R. Choudhary and N. Riaz, "A business process re-engineering approach to transform business process simulation to bpmn model," *Plos one*, vol. 18, no. 3, p. e0277217, 2023.

[25] T. Yue, W. Mauerer, S. Ali, and D. Taibi, "Challenges and opportunities in quantum software architecture," *Software Architecture: Research Roadmaps from the Community*, pp. 1–23, 2023.

[26] U. Nasir, "Using architectural kata in software architecture course: An experience report," in *Proceedings of the 5th European Conference on Software Engineering Education*, pp. 215–219, 2023.

[27] W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy, and B. Luo, "Sql and nosql database software architecture performance analysis and assessments—a systematic literature review," *Big Data and Cognitive Computing*, vol. 7, no. 2, p. 97, 2023.

[28] G. Márquez, H. Astudillo, and R. Kazman, "Architectural tactics in software architecture: A systematic mapping study," *Journal of Systems and Software*, vol. 197, p. 111558, 2023.

[29] C. Dias, J. T. C. Nery, M. d. S. Hounsell, and A. B. Leal, "A multimodal software architecture for serious exergames and its use in respiratory rehabilitation," *Sensors*, vol. 23, no. 21, p. 8870, 2023.

[30] M. M. Hasan, M. A. Rahman, M. S. Chowdhury, M. H. Rahman, K. H. Abdulle, F. Sadia, and M. Hasan, "Testing react single page web application using automated testing tools.," in *ENASE*, pp. 469–476, 2022.

[31] M. T. Hasan, S. N. Mahal, N. M. A. Bakar, M. M. Hasan, N. Islam, F. Sadia, and M. Hasan, "An unified testing process framework for small and medium size tech enterprise with incorporation of cmmi-svc to improve maturity of software testing process.," in *ENASE*, pp. 327–334, 2022.

[32] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, and Q. Wang, "Software testing with large language models: Survey, landscape, and vision," *IEEE Transactions on Software Engineering*, 2024.

[33] L. Lavazza *et al.*, "Icsea 2022: the seventeenth international conference on software engineering advances," 2022.

[34] S. R. Akthar, M. S. Khan, F. Sadia, and M. Hasan, "Optimizing b2b relationships with post-covid sales and marketing automation through erp implementation: A value-driven approach in bangladesh.," in *ICSBT*, pp. 209–215, 2023.

[35] N. M. A. Bakar, M. Hasan, and M. Rokonuzzaman, "Enhancing product market payoff in small and medium internet-based firms: A survey-based analysis of innovation and competition factors," 2023.

[36] J. Mero, M. Leinonen, H. Makkonen, and H. Karjaluoto, "Agile logic for saas implementation: Capitalizing on marketing automation software in a start-up," *Journal of business research*, vol. 145, pp. 583–594, 2022.

[37] A. Capatina, M. Kachour, J. Lichy, A. Micu, A.-E. Micu, and F. Codignola, "Matching the future capabilities of an artificial intelligence-based software for social media marketing with potential users' expectations," *Technological Forecasting and Social Change*, vol. 151, p. 119794, 2020.

[38] M. T. Hasan, M. Akter, S. I. Shuhrid, T. A. Khan, F. Sadia, and M. Hasan, "Investigation the influence of marketing-mix efforts on brand equity in the bangladesh software industry.," in *ICSBT*, pp. 152–159, 2022.