2024-01-28

Adaptive Software Engineering: Integrating AI-driven Techniques for Software Development

Islam, Mahmudul

IUB, CSE

Adaptive Software Engineering: Integrating AI-driven Techniques for Software Development

By

Mahmudul Islam , Farhan Khan

Student ID: **2231661 , 2231662**

Summer, 2023

Supervisor:

Mahady Hasan, PhD

Associate Professor

Department of Computer Science & Engineering

Independent University, Bangladesh

January 28, 2024

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

Department of Computer Science & Engineering

Independent University, Bangladesh

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

Firstly, we express profound gratitude to Almighty Allah for guiding us in our academic journey at Independent University Bangladesh. The experienced faculty at IUB provided invaluable support. Dr. Mahady Hasan, our supervisor, played a crucial role in our Master's program. We extend our heartfelt gratitude to our supervisor Dr. Mahady Hasan for his invaluable guidance and feedback during our graduate project. Special thanks to our respected faculty members Ms. Sabrina Alam, Ms. Nujhat Nahar, and Ms. Farzana Sadia for their support and guidance. We also appreciate the collaborative efforts of co-authors and friends at IUB. Lastly, heartfelt thanks to our family, friends, and loved ones for their unwavering support and encouragement.

Mahmudul Islam & Farhan Khan
October, 2023

Letter of Transmittal

Mahady Hasan, PhD, UNSW
Head, and Associate Professor,
Independent University of Bangladesh,
Department of Computer Science and Engineering

Subject: Graduate Project Report Submission Letter, Summer 2023

Dear Sir,
We, Mahmudul Islam and Farhan Khan (ID: 2231661 and 2231662), enrolled in the Summer 2023 Semester Graduate Project Course. Hereby submit our graduate project, reflective of our experiences during the Master of Science program at Independent University Bangladesh. The primary objectives of our project were to acquire practical insights into the software engineering sector, engage in research, and familiarize ourselves with various technology-related departments within the company, encompassing research and development, documentation, software development, and processes. Adapting to evolving technologies throughout our master's program equipped us to apply this knowledge to real-world scenarios. We sincerely hope for the acceptance of our report and appreciate your consideration.

Sincerely,
Mahmudul Islam , Farhan Khan
Email address: mahmud@iub.edu.bd , 2231662@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

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

In today's rapidly evolving technological landscape, software engineering stands as an influential pillar supporting numerous industries and opportunities of modern life. Organizations strive to deliver high-quality software products within timelines. Therefore, the need to adapt to effective methodologies, techniques, and frameworks is becoming increasingly necessary when it comes to effective product development and business perspective. Also, the software industry is expanding and scopes of development are changing due to new technologies and optimizations. Therefore, the constant adaptation of practices is necessary in the most cost-effective ways. This report consists of three research areas related to software engineering addressing software project management, software testing, and software marketing. However, our research shows that many companies lack the necessary tools and factors for adopting successful practices in terms of project management, software testing, and software marketing strategy. For these lacking the software industry is still not able to bloom to its full potential which results in 50 percent [1] failure risks of projects. Therefore, this report focuses on improving the software industry by identifying the causes of failures, techniques, and business aspects. It combines important research areas to define a proper structure and pattern of possible solutions to the current challenges faced by companies in software project management, testing, and software marketing.

1.2 Contribution of the thesis

Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery is an empirical study that investigates COVID-19 impacts on project management and delivery. It identifies influential factors and practices of software engineering that helped software companies to deliver projects on time based on scenarios. Project management is an integral part of Software development lifecycle (SDLC) models which defines the success rate of a software project. Due to improper standards, structure, and poor practices in project management, most projects fail. Therefore, it is important to identify the flaws in the daily project management practices and adopt efficient approaches to ensure team communication, commitment, and on-time project delivery. Currently, software companies tend to ignore these factors and face challenges to deliver projects within due deadline which eventually affects the company's financial stability. We have collected data from different software companies in Bangladesh through surveys and asked them what factors they found responsible for failures and success in project delivery during and after COVID-19 scenario. This helped us to identify the influential factors (Attentional focus, Team stability, Communication, Team maturity, Team capabilities, Infrastructure, Team Commitment, and User involvement) for project management which were adopted based on two different scenarios. This study which is a part of our thesis will contribute to efficient project management during challenging situations like COVID-19. Our another research study was to investigate the role of Artificial Intelligence in Software Testing. This was a systematic review that aims to provide knowledge of the current uses of AI techniques in software testing. Technology is evolving rapidly and new tools, techniques, and approaches are being adopted in software testing where AI is playing a key role. Traditional software testing techniques are time-consuming and costly. Therefore, the capabilities of AI to perform automated testing have been leveraged which is effective, time efficient, and less error-prone. This study identifies that testing activities such as Test Case Generation, Defect Prediction, Test Case Prioritization, etc are successfully automated using AI techniques. We believe this study which is a part of our thesis will help the related stakeholders to know the current status of automated testing and adapt them to their software testing practices. Our another research study was to identify factors that are crucial to forming the business strategy for online-based ride-sharing services. This was a survey on the business perspective approach that technological companies used to market ride-sharing application services to customers. As the software is being developed, with it comes the concern of business as everything needs to have both positive impact socially and economically. Many companies miss out on properly strategizing the business plan for their company which later becomes a big challenge. Hence, we addressed the influential factors of business strategy and 4P's of marketing strategy (Product, Pricing, Place and Promotion) in terms of ride-sharing applications. Two separate surveys were conducted. One is for the ride-sharing companies and another one is for ride-sharing service users. This study helped us to identify where the companies went wrong by mapping their overall marketing strategy with users' needs and satisfaction. As a result, we were able to identify the influential factors that resulted in a failed and successful software marketing strategy. This research study is ride-sharing-based but can be replaced with any software in terms of strategy evaluation method.

1.3 Organization of the thesis

In Chapter 1, the introduction section describes the entire report by offering a concise overview of the various 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. Chapter 3 describes one of our published research works, providing insights into an empirical study on the impact of COVID-19 on the factors influencing on-time software project delivery. This chapter presents the detailed findings from the published paper. In Chapter 4, we describe our another published research work, a systematic review of artificial intelligence in software testing. This section describes the use of AI in software testing, methodologies, and recent trends in this rapidly evolving field. Chapter 5 explores the factors essential for formulating a business strategy for online-based ride-sharing services. It describes the challenges and opportunities posed by this industry and offers valuable insights for stakeholders and decision-makers. Finally, Chapter 6 the conclusion section serves to summarize the collective knowledge presented throughout the report and provides a cohesive understanding of the research outcomes.

1.4 Thesis Project Management

1.4.1 WBS

WBS is an ideal tool for brainstorming and to visualize further scope of basic phases into much more detailed 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 4 phases. The first phase; Initiation and definition focuses on identifying the research gap and extracting the necessary research areas that have not been addressed. As the research idea along with its contribution is evaluated, further hypotheses and research questions are developed using studies. This development allows the authors to formulate the contributions and value the research papers will bring to society in terms of knowledge. Provided that the research paper has significant contributions and importance, literature review is conducted to gain in-depth knowledge of the research areas. The research papers are collected from Google Scholar. In the second phase; Planning, The project management plan is developed regarding further scopes defining timelines and tasks. The knowledge development along with timeline definition helps us to formulate strategy for research processes such as data collection. Further, the research methodology is designed along with defining research tools and techniques required for necessary data analysis

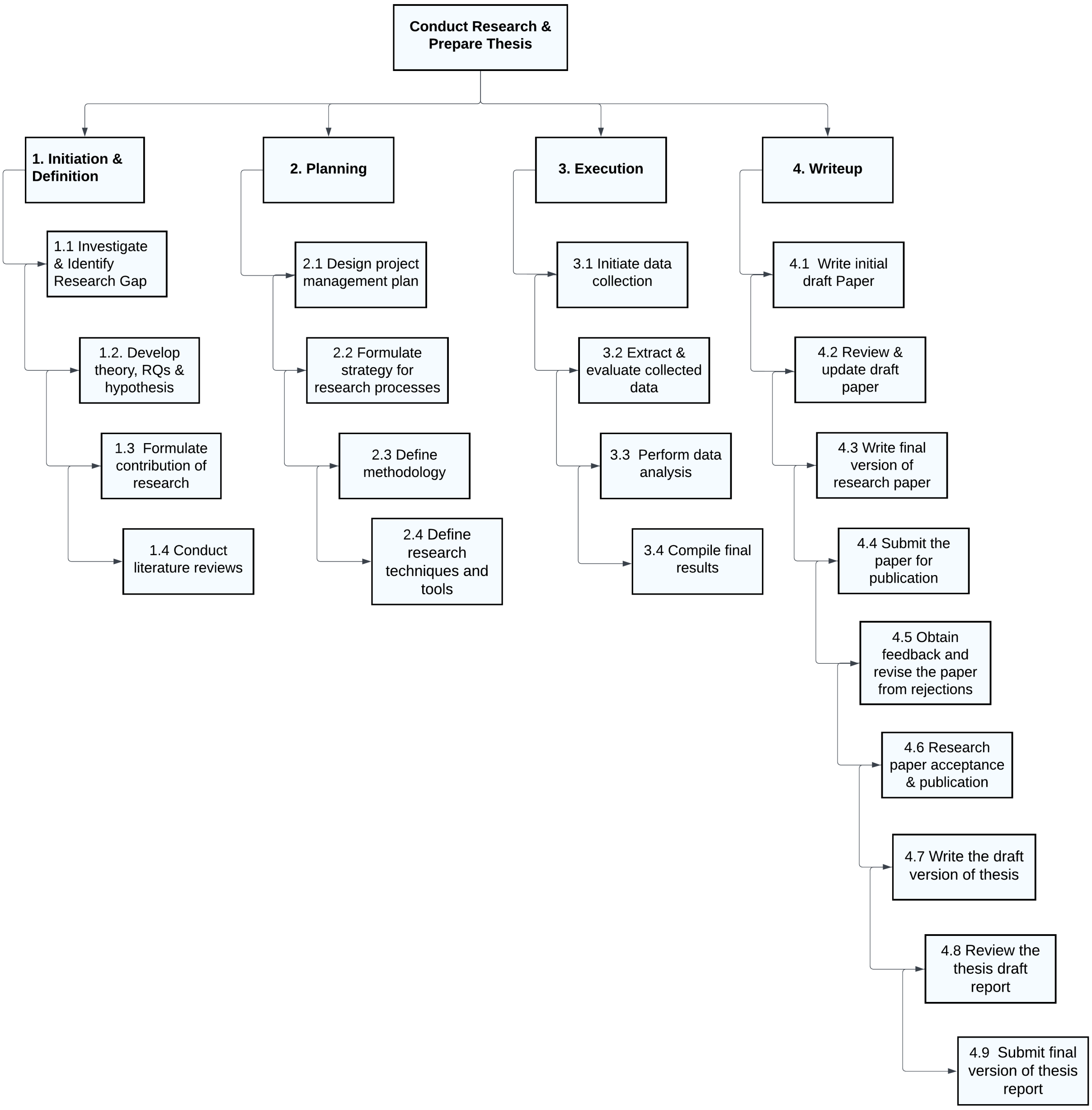


Figure 1.1: Work Breakdown Structure of Thesis

and collection. 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 process. 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 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.

1.4.2 Gantt Chart

Gantt charts are tools that is 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 timeline. In the given chart. We can find the activities related to research work and WBS are defined. Initially starting from identifying research gap, developing theories, selecting methodologies to later on, executing collection and extraction of data to validate results. These results are reviewed with proper references and 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 to acceptance and publication of a research paper. As the research work is completed, overall thesis write-up is is compiled and submitted. This Gantt chart, exactly represents the activities it took over a certain period to develop the 3 research papers to be upto the standards of publishing.

WBS	TASK NAME	START DATE	END DATE	DURATION (WORK DAYS)
	Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery	22-May-2022	27-April-2023	233
1.1	Investigate topic and Identify research gap	22-May-2022	29-May-2022	5
1.2	Develop main theory, research questions & hypothesis	30-May-2022	5-Jun-2022	5
1.3	Formulate contribution of research	6-Jun-2022	12-Jun-2022	5
1.4	Conduct literature reviews	13-Jun-2022	23-Jun-2022	9
2.1	Design project management plan	24-Jun-2022	25-Jun-2022	1
2.2	Formulate strategy for research processes	26-Jun-2022	28-Jun-2022	2
2.3	Define research methodology	29-Jun-2022	4-Jul-2022	4
2.4	Define research techniques and tools	5-Jul-2022	9-Jul-2022	4
3.1	Initiate data collection	10-Jul-2022	4-Aug-2022	19
3.2	Extract and evaluate collected data	4-Aug-2022	8-Aug-2022	3
3.3	Perform data analysis	9-Aug-2022	14-Aug-2022	4
3.4	Compile final results	15-Aug-2022	18-Aug-2022	4
4.1	Write initial draft paper	19-Aug-2022	1-Sep-2022	10
4.2	Review & update draft paper	4-Sep-2022	6-Nov-2022	45
4.3	Write final version of research paper	13-Nov-2022	24-Nov-2022	9
4.4	Submit the paper for publication	25-Nov-2022	25-Nov-2022	1
4.5	Obtain feedback and revise the paper from rejection & resubmit in other venues	26-Nov-2022	22-Feb-2023	63
4.6	Research paper accepted & published	3-Mar-2023	27-Apr-2023	40
	Factors to Form Business Strategy for Online-	24-May-2022	8-Sep-2023	333
1.1	Investigate topic and Identify research gap	24-May-2022	2-Jun-2022	8
1.2	Develop main theory, research questions & hypothesis	2-Jun-2022	7-Jun-2022	4
1.3	Formulate contribution of research	8-Jun-2022	14-Jun-2022	5
1.4	Conduct literature reviews	15-Jun-2022	28-Jun-2022	10
2.1	Design project management plan	29-Jun-2022	29-Jun-2022	1
2.2	Formulate strategy for research processes	30-Jun-2022	2-Jul-2022	2
2.3	Define research methodology	3-Jul-2022	6-Jul-2022	3
2.4	Define research techniques and tools	7-Jul-2022	12-Jul-2022	4
3.1	Initiate data collection	13-Jul-2022	1-Aug-2022	14
3.2	Extract and evaluate collected data	1-Aug-2022	4-Aug-2022	4
3.3	Perform data analysis	7-Aug-2022	14-Aug-2022	5
3.4	Compile final results	14-Aug-2022	18-Aug-2022	4
4.1	Write initial draft paper	21-Aug-2022	1-Sep-2022	9
4.2	Review & update draft paper	1-Sep-2022	1-Sep-2022	1
4.3	Write final version of research paper	2-Sep-2022	4-Sep-2022	1
4.4	Submit the paper for publication	5-Sep-2022	5-Sep-2022	1
4.5	Obtain feedback and revise the paper from rejection & resubmit in other venues	5-Sep-2022	17-Jun-2023	205
4.6	Research paper accepted & published	29-Jun-2023	8-Sep-2023	52
	Artificial Intelligence in Software Testing	1-Feb-2023	14-Oct-2023	165
1.1	Investigate topic and Identify research gap	1-Feb-2023	6-Feb-2023	4
1.2	Develop main theory, research questions & hypothesis	6-Feb-2023	12-Feb-2023	5
1.3	Formulate contribution of research	13-Feb-2023	15-Feb-2023	3
1.4	Conduct literature reviews	15-Feb-2023	12-Mar-2023	18
2.1	Design project management plan	13-Mar-2023	13-Mar-2023	1
2.2	Formulate strategy for research processes	14-Mar-2023	15-Mar-2023	2
2.3	Define research methodology	16-Mar-2023	20-Mar-2023	3
2.4	Define research techniques and tools	21-Mar-2023	23-Mar-2023	3
3.1	Initiate data collection	24-Mar-2023	31-Mar-2023	6
3.2	Extract and evaluate collected data	1-Apr-2023	7-Apr-2023	5
3.3	Perform data analysis	8-Apr-2023	11-Apr-2023	2
3.4	Compile final results	12-Apr-2023	16-Apr-2023	3
4.1	Write initial draft paper	17-Apr-2023	24-Apr-2023	6
4.2	Review & update draft paper	25-Apr-2023	1-May-2023	5
4.3	Write final version of research paper	1-May-2023	4-May-2023	4
4.4	Submit the paper for publication	5-May-2023	5-May-2023	1
4.5	Obtain feedback and revise the paper from rejection & resubmit in other venues	7-May-2023	14-Jun-2023	28
4.6	Research paper accepted & published	14-Jul-2023	14-Oct-2023	66
	Final Thesis Writeup & Report Submission	13-Aug-2023	25-Nov-2023	76
4.7	Write the draft version of thesis	13-Aug-2023	5-Nov-2023	60
4.8	Review the thesis draft report	6-Nov-2023	17-Nov-2023	10
4.9	Submit final version of thesis report	17-Nov-2023	25-Nov-2023	6

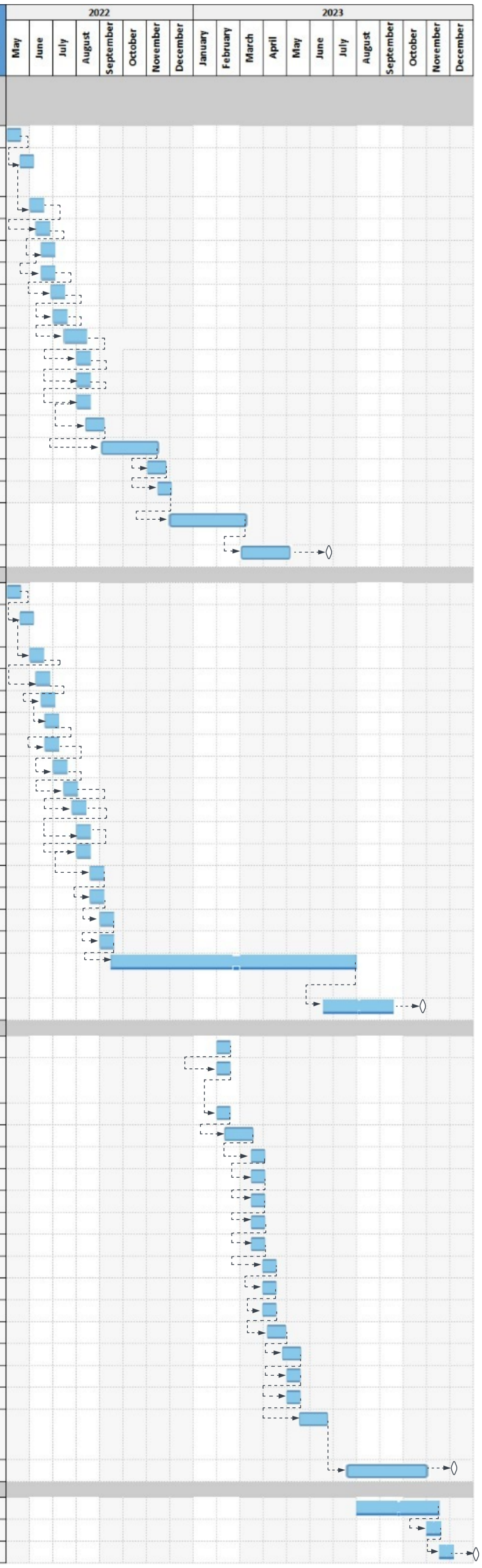


Figure 1.2: Gantt Chart

Chapter 2

Literature Review

2.1 Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery: An Empirical Study

[2] investigated the critical success factors (CSFs) of the agile software development model. They found six different CSFs where organizational factors were most influential and project, organizational, people and process factors have a direct impact on performance expectancy factors. [3] conducted a case study among students to identify both success and failure factors that impact project Implementation using Agile methods. They found that there were differences in non-functional tasks while employing different methods. [4] surveyed two software-intensive companies through semi-structured interviews, concentrating on five-month cooperation for operationalizing process metrics using Agile software development. [5] reviewed previous studies. They selected 38 papers out of 2422 papers for this study and extracted 25 common motivators and 14 common demotivators factors. They classified Motivator and demotivator factors into four classes such as people, organization, technical, and process. [6] evaluated the agile success factors. MCDM technique was put out as a potential solution with a priority of 0.5633, Scrum was chosen as the agile process's most effective success element. [7] adopted a qualitative research method through which five semi-structured interviews were conducted. They cross-mapped obtained results. Most of the failure factors had a direct impact on the project that led to delays on-time delivery of software projects. [8] found that frequent request changes made by clients have a negative effect on on-time delivery and cost. They proposed a cost and time estimation model for ASD that can be followed for successful on-time software delivery. [9] reviewed the previous studies related to agile project management. They divided project success factors into 6 main categories management, process, project, organizational, people, and technical factors. [10] performed a

systematic literature review to understand and list newly developed Requirement Prioritization techniques. [11] investigated how the implementation of a hybrid methodology will enhance software project success in terms of product delivery on time. They found hybrid models are flexible and adaptable. [12] explain the adoption of agile methodology and suggest the product can be delivered more efficiently by expansive testing before releasing the software code to production. [13] explained the adoption of agile software development. They surveyed 50 respondents from several software companies and found the impacts, issues and outcomes of adopting agile methodology. [14] examine the factors that impact the agile software development model. They found 24 critical factors and classified them into four categories. This study [15] discussed agile adoption in Saudi Arabia. They identified five motivator factors for agile adoption. [16] explained the benefits of agile methods. The notable benefits are on-time delivery, customer satisfaction. [17] did a systematic literature review on different SDLC models and discussed pros and cons of each methods.

2.2 Artificial Intelligence in Software Testing: A Systematic Review

They [18] proposed a deep learning model to rank test cases. In this work, they consider historical records of test case executions and based on that deep learning model rank test cases. They [19] conducted an empirical study on continuous integration testing. They found the strategy of reward function of Reinforcement learning improves the existing test case prioritization practices. They [20] developed a deep reinforcement learning technique for performing black box testing on android apps. Their developed technique outperforms existing techniques in terms of fault identification. They [21] proposed a deep learning-based approach for prioritizing test cases from the interaction of humans with software applications. They showed that test case prioritization can perform successfully from human interactions using their proposed model. They [22] presented an approach to generate input for the graphical user interface of software applications by only capturing screenshots of applications. They [23] proposed a machine learning-based approach to predict metamorphic relations of scientific software using graph kernels. They concluded that features extracted from graphs help to achieve a good result. They [24] presented an approach to automate test oracle mechanism using machine learning. Their proposed approach captures historical usage data and based on that generates an oracle. They [25] detected metamorphic relations using graph kernels and support vector machines (SVM). They [26] analyzed software defect prediction using machine learning algorithms. They found that linear classifier performs well compared to other algorithms. They [27] proposed an improved CNN model to predict software defects and their proposed model outperformed existing models.

2.3 Factors to Form Business Strategy for Online-Based Ride-Sharing Services

Moslehpour, M. et al. [28] conducted a research study to investigate How the relationship between social media marketing activity and purchase intention involves trust and brand impression. Lampe, I. et al. [29] conducted a study, where they sought to comprehend a ride-sharing service Topokeni's social media marketing tactics. The findings of this study demonstrated that social media had emerged as a viable alternative for attracting new customers and tenants. Islam, M.T. et al. [30] conducted a comparative study to evaluate factors that influence customers' perception of ride-hailing services and to understand the competitive market position of ride-hailing companies based on their existing performance. Abdullah, O.N. et al.[31] did a literature review from different sources regarding the growth of pathao in a competitive market. They summarized a business model of pathao and how to use a time-to-time market strategy. G.W.Chelliah et al.[32] analyzed Grab's marketing services and suggested improvement of Grab's marketing strategies. Zailani, N.F.I. et al. [33] investigated the influential factors of ride-sharing application services in Malaysia.

2.4 Social Media Marketing Activity Influences Customer Behavior Outcomes

Izogo, E.E. and Mpinganjira, M. [34] conducted a research study to find how social media content motivates customers. They used structural equation modeling and ANOVA for the analysis. They found inspirational content has a positive effect on customer motivation. Godey, B. et al. [35] investigated how social media marketing activities influence customers' behavior toward a specific brand. They also used structural equation modeling for the analysis. Upadhyay, Y. et al.[36] investigated social media marketing's effect on customer behavior using a structural model. They found that brand equity has a partial effect on customer behavior. Haudi et al. [37] investigated the effect of social media marketing activities on product brand. To identify the factors involved, they adopted a quantitative research method where they surveyed 400 customers. Later, they measured the impact on brand outcomes such as trust, honesty, and loyalty. Ebrahim [38] investigated the impact of social media marketing activities on product brand. For this study, the author surveyed 287 users which suggests that social media marketing attributes directly or indirectly influence product brand. Chen et al. [39] proposed that social media marketing activities have indirect effects on satisfaction through social identification and perceived value. To identify the factors involved, they surveyed 502 users who use social media for online marketing. Kang and Park [40] measured the impact of factors on customer purchase intentions. They have also identified keywords and factors which should be prioritized in social media marketing. Structural equation modeling has been used for analysis. The findings show that brand trust and attitude significantly influence customer purchase intention. In this study [41], the impact of COVID-19 pandemic on consumers' purchase intentions has been measured. A combination of questionnaires and ANOVA analysis were used. Dastane [42] examines the effect of social media marketing on online purchase intention and examines how customer relationship management acts as an important mediator influencing these two factors.

2.5 Software Requirement Classification using Machine Learning

Using text vectorization and machine learning techniques the database can be updated. In this case, they have applied Term Frequency - Inverse Document Frequency. They have assessed the performance of the applied models with the help of dissimilar performance criteria such as accuracy, recollection, etc. They also have found that machine learning algorithms do not perform well while the data set is imbalanced[43]. The user requirement has been classified into a small data set which comprises the user feedback. The best pacifier has been applied toward and the performance I could receive came to the best of 87%. At first, the data cleaning process has been applied which means the entire information has been chopped off into small useful phrases. In the second phase, more words are emitted that are less important And the BERT classifier outmatches the rest of the classifier with More than 80% of accuracy[44].

The unpredictabilities in SRS papers are accurately defined and expressed by Prof. Brown, T. et al. Identifying the difference in the SRS paper is a difficult job and has influenced the researchers for it. The main techniques that are used to recognize the distinction in NK SRS documents are analysis and examinations. The suggested implementation includes information based structure that combines connotation and linguistic analysis of needs[45]. Abid, M., Hasan et al. identify some evaluation matrix which identifies the effectiveness of our proposed algorithms for specifying real software requirements. Thus automating the requirementing process by and reducing development costs and the possibility of human errors[46].

Better requirements prioritisation methods must be picked which will benefit software developers, proposed by Talele P. et al. The task is to use various algorithms to allocate and filter the software requirements..The suggested planning will aim to draw out the attributes that will be then used to train the model with the help of machine learning(ML) algorithms. And it will analyse and sort out the software requirements thus advancing development terms and evaluating[47]. A well structured analysis of twenty - four machine learning - based methods for recognizing and allocating non - functional requirements was described by Binkhonain, M. et al. This was managed by three main investigated questions of how the machine learning(ML) algorithms will be assessed. The review of the paper detects that although machine learning based approaches have the possibility in terms of categorization and recognition of non - functional requirements; however, they have faced several open challenges which can and will take effect to their output and applied and experimental application[48].

Atoum L. et al predict an application's user experience on UX before creating a prototype available for use. This study gathers multiple UX specialists to create a standard dataset user metrics which relies on documented software needs. The consequences of the study show that the dataset has a minimal standard deviation error and a high Cronbach Alpha. Conclusion was made based on this that the new standard dataset has a probability and could be useful to make estimations of UX almost instantly unaccompanied by the need of personalised UX assessment[49]. The TF - ML, Three - Family ML, machine learning, is used to develop the machine learning model which can foresee the risks from the dataset filled with the RS, requirement risks. Ten different TF - ML algorithms were applied. As a result, the following study suggests using Credal Decision Tree, CDT, for software requirement risk prediction. This study can also be chosen for creating and maintaining the main model for the prediction of the risk for software requirement[50].

The dataset is processed into three different types: the dataset is converted into csv file, second they applied different categorised algorithms and finally sorted out the unique stratification features. The data is then processed using NLTK and SP (Stanford Parser) to remove commas, periods, hyphens and all other punctuations. The classifier is a variation of both non - functional requirements and UC (University of Cincinnati) dataset of 33% and thus creating a hybrid dataset. This results in a NFs binary classifier where all the quality attributes are found as a dataset. By applying the best 200 informative attributes, accuracy was achieved of more than 70%.[51].

Hugo Vilamizar et al in 2021 described that ML, machine learning, is a swoting of computer instructions of code that improve axiomatically over occurrences. They have found dissimilar sorts of benefactions and recognized the standard distinctions that are exceptionally applicable to the context of machine learning or ML[52]. The authors evaluated various strategies from previous papers for requirements extraction and ordering using which they proposed a model for requirements prioritisation. Then requirements are analysed using ML algorithms and users are assigned. Feedback is obtained through ML models and output requirements are extracted and the algorithm is further trained[53].

Chapter 3

Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery: An Empirical Study

Islam, M., Khan, F., Hasan, M., Sadia, F., & Hasan, M. Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery: An Empirical Study. 18th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), April 24-25, 2023 Prague, Czech Republic

One of the main reasons for software project failure is not choosing the proper methodologies. There are many Software Development Life Cycle (SDLC) models such as Waterfall, Agile, Incremental, Prototype, Spiral, etc. Software companies follow different SDLC models according to their needs and project requirements. The issues with traditional development are solved by the agile model. Therefore, software and IT companies are adopting agile approaches, which help them to accomplish their goals. Cost overrun and on-time delivery is a major issue in the software industry and the success of a project depends on them [54]. Effort estimation is crucial for any software project since it helps to deliver on time with the defined constraints [55]. This study investigates the impact of the COVID-19 pandemic on the crucial factors that impact on-time software project delivery in different SDLC models and how the level of impact of these factors shifted with respect to different demographic information such as company size, developers' years of experience and different SDLC models used software professionals.

Covid 19 pandemic has impacted everything including the software development paradigm. It was a challenge for the team to interact properly although online platforms such as google meet, zoom, and other platforms made the interaction possible but it was not quite smooth. Several factors impact on-time delivery [56] and during the pandemic, these factors might be changed to different degrees. We investigated the research gap and found that no study has yet been conducted that describes how the important factors of on-time delivery have changed over time, especially during COVID-19 in different SDLC models. This has motivated us to conduct this research study and measure the impact of COVID-19 on-time software project delivery in different SDLC models. Previous studies were mostly agile model related therefore we cover different SDLC models in this study. In another previous study, the success factors are stated [57] for agile software development but it is not determined how these factors influence on-time software delivery.

The research objective of this study is to find the impact of COVID-19 on the factors that impact on-time delivery in different SDLC models like Agile, Incremental, Prototype, Waterfall and also the relation of these factors with different demographic information such as company size, developers' years of experience and the use of SDLC models by software professionals. We followed a quantitative approach and obtained 72 data through questionnaires from 11 different software companies in Bangladesh. Finally, calculated statistical parameters to extract the correlation among factors from the obtained data.

The main contributions of this paper are:

- An overview of the impact of COVID-19 on-time software project delivery in different SDLC models.
- Identification of the top factors which are responsible for on-time software project delivery in different SDLC models before and during COVID-19.
- Identification the change of impact level of important factors for on-time software project delivery with respect to different demographic information.

The rest of this paper is structured as follows. Related works have been described in Section 2. In Section 3 research survey design has been explained. Detailed survey results have been presented in section 4. Validity threats have been discussed in section 5. Finally, conclusions have been drawn in section 6.

3.1 Research Design

This research work is designed through the study of various previous research papers. This helped us to identify the research gap and design the research questionnaires for this study. Previous studies focused particularly on agile software development. After identifying the research gap, in this study, we explored how different factors affect on-time software delivery in different SDLC models over the time, especially during COVID-19.

3.1.1 Purpose of the Study

The purpose of this study is to analyze and quantify the effects of the COVID-19 pandemic and other variables on the timely completion of software projects in different SDLC models. There are multiple factors that influence software delivery on time, and these factors have taken on different forms during the COVID-19 pandemic which is still unknown specifically in detail. This has motivated us to carry out this research and determine the impact level of COVID-19 on the timely delivery of software projects and compare these factors with before COVID-19 scenario. In this study, we focused on the following research questions (RQs) and these questions have been adopted from this [56] study, we have modified the questions with the concept of COVID-19.

- **RQ1**: How much does each of the following factors influence on-time software project delivery [Before and During COVID-19] ?
- **RQ2**: Do any other factors or certain types of circumstances, need to be added to ensure on-time software project delivery?

There were 25 questions under RQ1 adopted from [56, 57, 2]. These 25 questions were related to different factors such as Attentional Focus, Bugs, Collocation, Communication, Executive Support, Insufficient Testing, Lack of Code Quality, Organizational Alignment, Organizational Politics, Organizational Stability, Poor Documentation, Project Criticality, Project Newness, Project Size, Refinement Quality, Regular Delivery, Task Dependencies, Team Capabilities, Team Commitment, Team Maturity, Team Stability, Technical Dependencies, Infrastructure, and User Involvement. After analyzing the responses from software companies we have identified the change of factors for on-time software project delivery during and before COVID-19. We also identified that the impact level of factors for on-time software project delivery changes with respect to different parameters. Along with research questions, demographic questions (DQs) have also been asked in this study. The IT professionals have been asked the following demographic questions in this study.

- **DQ1**: Which of the following best describes your role at your company?
- **DQ2**: How many years of work experience do you have in the software development industry?
- **DQ3**: What is the size of your company?
- **DQ4**: Which software development life cycle model has been practiced or followed by you ?

3.1.2 Research Method

This research study was conducted on different IT professionals such as Project Manager, Software Developer, Software Architect, QA Engineer, DevOps from 11 different software companies in Bangladesh. An extensive literature review was performed before conducting this research study which helped us to get a solid idea about the topic and identify research gap. In this study, quantitative method in 4 point likert scale format has been used to collect data [58]. Python has been used in data analysis step. We studied the previous literature [59] and found that Net Promoter Score (NPS) is a metric which is used to measure how likely people suggest a thing to others. In this study, NPS along with mode and mean have been calculated. The methodology in Figure 1, represents the flow of research method that was developed in four stages: Research Design, Data Collection, Data Analysis and Final Result.

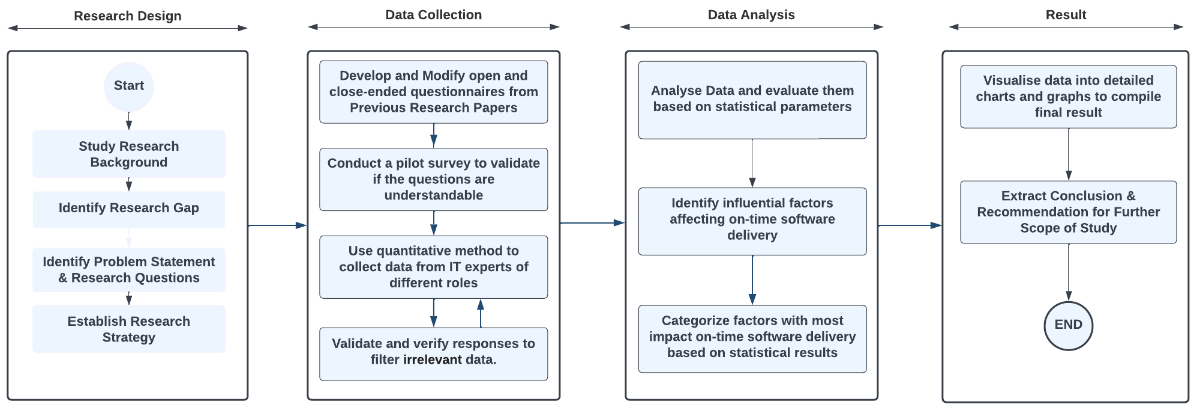


Figure 3.1: Research Methodology

3.1.3 Data Collection Method

In this study, the criteria for selecting participants were based on purposive sampling methods where IT professionals who were involved in the software project deliverables were included in the sample's inclusion criteria. The individuals who possessed primary features for selection criteria were filtered and addressed with proper instructions. The questionnaires were organized in two sections that addressed the demographics and factors impacting the timeliness of the IT team's software project deliveries before and during Covid-19 pandemic. In this study, There were a total of 26 questions adopted from previous literature papers [56, 57, 2] where they identified the factors for on-time project delivery. In this study we have used those questions but modified with the concept of COVID-19 and identified the change among factors due to COVID-19.

Most of the questions were designed as close-ended in a 4-point Likert scale format such as no impact, small, moderate, and large impact. The factors were presented in a random order to the survey participants to reduce ordering bias and a short description was included to explain the factors to participants. Moving forward, after the design of questionnaires and selection of participants, a pilot test was conducted before the initiation of the data collection process. It ensured that the interview questions were understandable to the IT professionals. Therefore, after the successful identification of misinterpretations and corrections of questions in the pilot test, quantitative data collection was initiated via google form. We were then able to collect, organize, characterize, and evaluate textual data. The key benefits of using these techniques are the enhanced ability to simplify and manage data without affecting context and obtain deeper insights that are expected from a study.

3.1.4 Data Analysis Method

after the completion of the data collection phase, an in-depth data analysis was performed using Python. Then, different statistical parameters such as NPS, Mode, Mean were calculated. NPS calculation allows to determine the most impactful factors while the calculation of Mode help to identify the central tendency. The factors were assessed from the obtained data where respondents revealed their approach to software deliverables before and during COVID-19 and their correlation with on-time software project delivery. As a result, taking these factors into account improved the generalizability of the findings.

3.2 Result and Analysis

In this study, 72 responses have been received from different roles such as Software Developer, Software Architect, Project Manager, QA Engineer, DevOps, etc from 11 different software companies in Bangladesh. In this section, we will describe the findings of this study regarding the impact of COVID-19 on-time software project delivery and the change of impact level of factors with respect to different parameters.

3.2.1 Demographics

Role: The survey participants were asked questions regarding their role in the software company. question DQ1 make sure that the participants are in the target population. Figure 2 depicts the role of the respondents in the software company. People working in different roles at software companies participated in this survey. The majority of the respondents were Software Developers and Project Managers. Software Architects, QA Engineers, and DevOps also participated in this survey as all of them play a role in on-time software delivery.

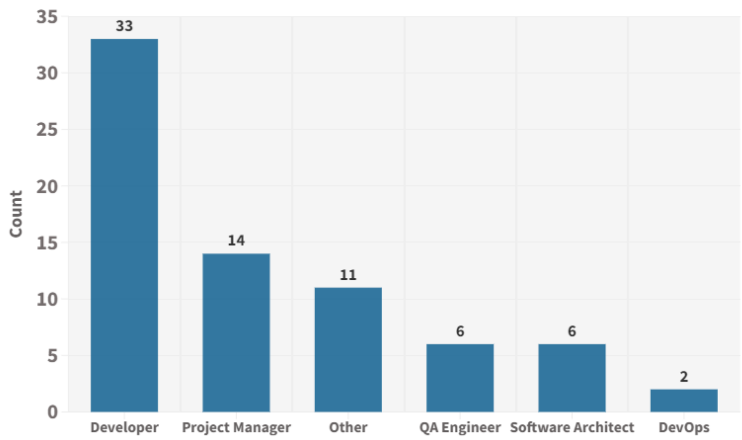


Figure 3.2: Survey Respondents' Role in the Software Company

Work Experience in Software Industry: Work experience in the software industry is important demographic information for our study. Software companies impose their employees to follow any specific SDLC model during the software development while a freelancer or a self-taught developer may not follow any specific SDLC model. We asked question DQ2 which reveals the work experience of participants. Results are shown in figure 3, most of the participants in this study have a software industry experience of one to five years while the least of participants have ten to twenty years of industry experience.

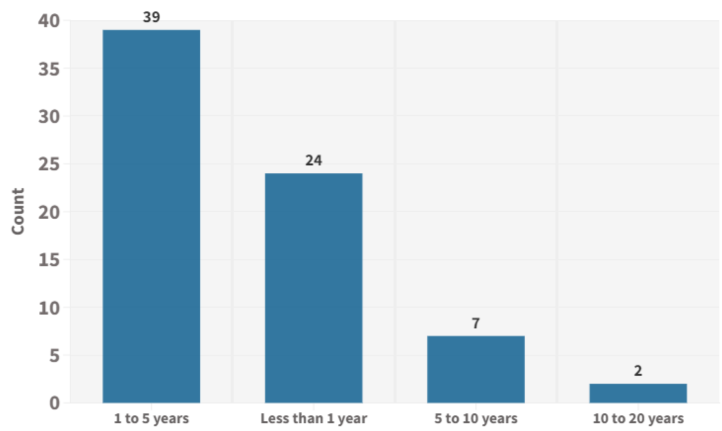


Figure 3.3: Survey Respondents' Work Experience in the Software Industry

Company Size: Question DQ3 had been asked to the participants to know the software company size. In this study, we find that the level of impact of factors for on-time software project delivery changes with respect to company size which has been described in detail later. Figure 4 shows the company size of the survey respondents. In this survey, most of the respondents belong to 20-50 size of software companies which is almost 28% of the respondents. 18% of the respondents in this survey belong to 100-500 size of software companies. 15% of the respondents are working in the medium size software companies where the company size is 50-100. Software professionals from large software companies where the number of employees over 500 also participated in this survey, they constitute 14% of the total respondents.

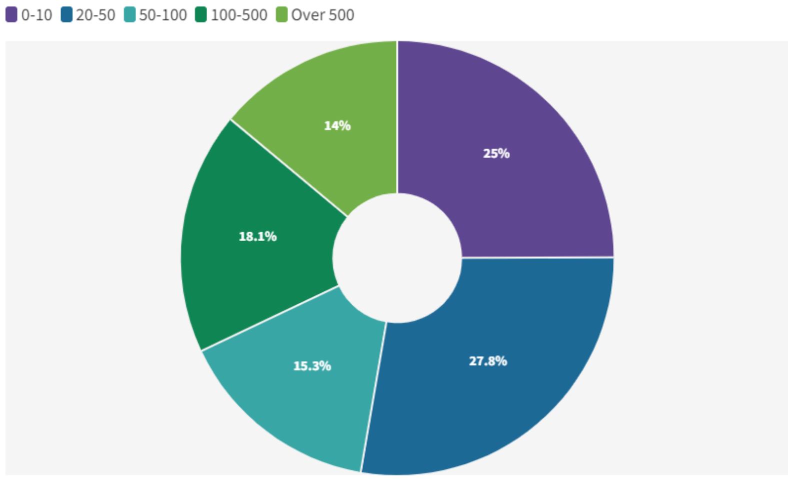


Figure 3.4: Survey Respondents' Company Size

Use of Software Development Life Cycle (SDLC) models: We asked participants question DQ4 to know about their followed SDLC models during software developments. Participants in this study responded that they follow mainly four SDLC models Agile, Incremental, Waterfall and Prototype Models. Results shown in figure 5 describe the use of SDLC models by survey participants. The Agile model is followed by most of the participants while the Prototype model is practiced or followed by the least of the participants during software development.

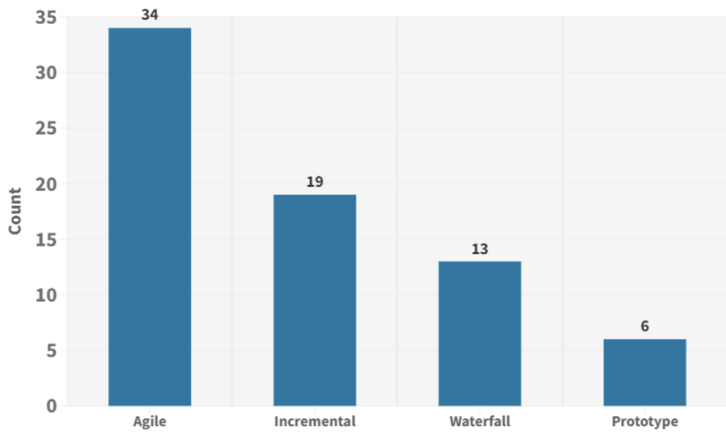


Figure 3.5: Use of SDLC Models by Survey Respondents

3.2.2 Impact of COVID-19 On-Time Software Project Delivery

In this research survey, we identified the top factors which influenced on-time software delivery before and during COVID-19. Data have been collected on a 4-point Likert scale.

In this study, the collected data have been mapped using table 1. Quantitative analysis has been applied to the survey data based on the mapping. We identified the top factors that impact on-time software project delivery during and before COVID-19. We also identified changes in the level of impact of factors with respect to different parameters such as respondents' company size and software industry experience. Net Promoter Score (NPS) has been used to identify the top factors [59]. Mode and Mean are also calculated.

NPS = (Count of Promoters - Count of Detractors) / Total Participants

Table 3.1: Likert Scale Data Mapping

Text Rating	Quantitative Rating	Type
No Impact	1	Detractors
Small Impact	3	Neutral
Medium Impact	5	Neutral
Large Impact	7	Promoters

Influential Factors for On-Time Software Project Delivery During COVID-19: Figure 6 represents identified top 5 factors during COVID-19 that have the highest NPS which means they have the highest impact or influence on-time software project delivery during COVID-19 according to survey participants. The mode and mean of those factors have also been calculated. Attentional focus was the most influential factor for on-time software project delivery during COVID-19. Team Stability, Communication, Team Maturity, and User Involvement were also significant factors for on-time software project delivery respectively. The identified factors are described below.

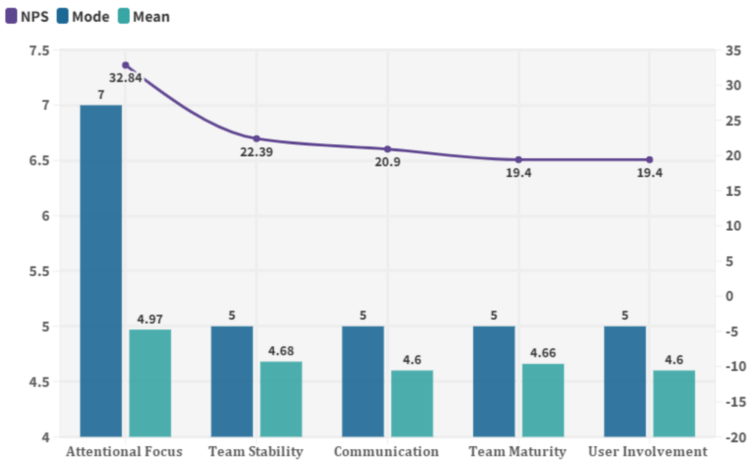


Figure 3.6: Top Five Factors for On-Time Software Project Delivery During COVID-19

Attentional Focus: In software development, Attentional Focus is an essential key element to accomplish the assigned task. Attentional Focus is to concentrate dedicatedly on a specific task or increased focus on a specific task which helps to achieve the task. Our findings in figure 6 from this study indicate that the most important factor impacting on-time software project delivery is Attentional Focus. During COVID-19 most of the software companies started to practice working from home. As day-to-day physical interaction reduced significantly, software professionals started to depend more on online meetings. Also, setting up a workplace at home had also become tough for some of them, there were a lot of distractions at home. So the amount of context switching increased for everyone and this started to reduce concentration or Attentional Focus. So this impacted the delivery time highly.

Team Stability: In software development, Team Stability is a major factor for a company's success as well as to deliver software projects timely to the clients [60]. Results of this study are presented in figure 6 where team stability has been deemed as the second most important factor. This is also quite intuitive. During COVID-19, the software industry observed a substantial resignation. People started to leave office jobs, move to family and switched to new remote jobs. Also, many people took long-term sick leave due to health conditions of their own or the family. Therefore, overall Team Stability has deteriorated and negatively impacted on-time delivery. Similar argument is applicable to Communication and User Involvement which have been perceived as the third and fifth significant factors respectively for on-time software project delivery during COVID-19. Communication and User Involvement are severely affected by COVID-19.

Communication: In software development, proper communication is a major criteria to understand the user requirements and deliver quality software products[61]. During COVID-19 overall team communication was disrupted and the clients could not connect to the software vendors as there was a restriction on mutual contact and everyone had to maintain social distancing. Eventually, this communication gap led to late software delivery. Figure 6 shows that Communication is perceived as the third most important factor for on-time software project delivery during COVID-19.

Team Maturity: Team Maturity or Group Maturity is significant for the efficiency of the software development team [62]. When a team works together for a long time, mutual understanding among the team members grows which helps them to achieve state of art in their work. During COVID-19 people in the software industry switched their job to remote jobs and the team had broken which ultimately impacted the Team Maturity. Figure 6 demonstrates that during COVID-19, Team Maturity was considered as the fourth most crucial factor for timely software project delivery.

User Involvement: In software development, User Involvement is a key factor for successful project completion [63]. User Involvement helps to understand the user requirements and deliver proper software products which eventually leads to user satisfaction. During COVID-19 user involvement decreased as a result the delivery of the software project was delayed. Figure 6 shows that User Involvement is found as the fifth most important factor for on-time software project delivery during COVID-19.

Influential Factors for On-Time Software Project Delivery Before COVID-19: In this survey, we also investigated the influential factors which were responsible for on-time software project delivery before COVID-19. The higher NPS value of a factor means the higher impact of that factor on-time software project delivery according to survey participants who were Software Developers, Software Architects, Project Managers, QA Engineers, and DevOps. Figure 7 represents the top five factors that have the highest impact on-time software project delivery before COVID-19, they are Team Capabilities, Infrastructure, Team Commitment, Team Stability, and Team Maturity. Team Stability and Team Maturity are found as common important factors both before and during COVID-19. The identified factors for on-time software project delivery before COVID-19 are described below except Team Stability and Team Maturity as they have already been described.

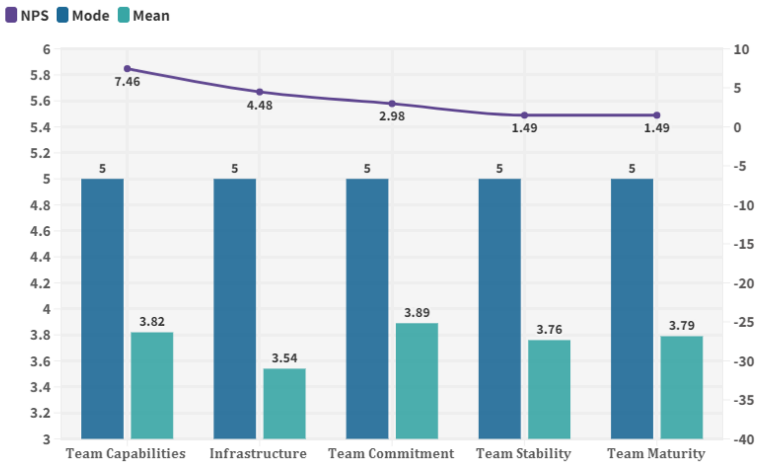


Figure 3.7: Top Five Factors for On-Time Software Project Delivery Before Covid-19

Team Capabilities: In software development, Team Capabilities is the expertise or skills of the software development team's members which is a crucial factor for successful project delivery [64]. In this survey we find Team Capabilities was the most important factor for on-time software project delivery before COVID-19. Results are shown in figure 7.

Infrastructure: Infrastructure is essential for software development, testing, and maintenance. Unavailability of Infrastructure leads to late delivery of software projects to the clients[65]. Figure 7 shows that Infrastructure was a major factor for on-time software project delivery before COVID-19.

Team Commitment: Team Commitment is the dedication of the team member to the timely delivery of the product and their concentration on attaining the team's goal. Team Commitment is an important factor for on-time delivery [66]. In this survey, Team commitment is also found as a major factor for on-time delivery of the software project before COVID-19 as shown in figure 7.

Comparison of factors before & during COVID-19: A comparison of factors before and during COVID-19 has been shown in table 2. The comparison in table 2 shows that during COVID-19 the most important factor for on-time project delivery was Attentional Focus while before COVID-19 Team Capabilities was the most important factor. Team Stability and Team Maturity were important for on-time software project delivery both before and during COVID-19. The results are shown in Table 2 and it is clear that there is a change of factors that are responsible for on-time software project delivery during and before COVID-19.

Before COVID-19 most important factors were Team Capabilities, Infrastructure and Team Commitment. Whereas during COVID-19, these factors did not play any significant role. One reason behind this can be, companies could recruit remote workers who have better capability. Apart from that, work from home enabled people to choose working hours freely which actually increased the overall commitment of the team. We see two factors like Team Stability and Team Maturity Present in both cases. This indicates that these two factors are not susceptible to external environments. Whatever the condition is, these two factors play a vital role in on-time software project delivery.

Table 3.2: Comparison of factors before and during COVID-19

Top 5 factors during COVID-19				Top 5 factors before COVID-19			
Factors	Mode	Mean	NPS	Factors	Mode	Mean	NPS
Attentional Focus	7	4.97	32.84	Team Capabilities	5	3.82	7.46
Team Stability	5	4.68	22.39	Infrastructure	5	3.54	4.48
Communication	5	4.6	20.9	Team Commitment	5	3.89	2.98
Team Maturity	5	4.66	19.4	Team Stability	5	3.76	1.49
User Involvement	5	4.58	19.4	Team Maturity	5	3.79	1.49

Change of Impact Level of Factors with Respect to Employee Experience: We have already identified the top five factors for on-time software project delivery both before and during COVID-19. In the second level of analysis, we investigated how the impact level of these identified factors changes with respect to different demographic information such as employee experience, company size and different SDLC models used by software professionals. The Mode has almost same value for factors and NPS has already been calculated to determine the factors. In the second level of analysis, The Mean value of these factors with respect to employee experience and company size has been calculated and presented in figure 8,9,10, and 11 respectively.

Figure 8 depicts change of impact level of identified top factors with respect to software professionals' experience during COVID-19. These factors do not have the same level of impact on the different levels of experienced people. In figure 8, results show that more experienced software professionals (5-10 years of experience) voted for Attentional Focus more. This is quite intuitive as senior members are more involved in meeting with clients, clearing junior members' obstacles, etc. So their context-switching frequency is high, and as a result, they can not focus on a single task. Therefore, Attentional Focus is the most important factor for them. For fresh graduates or software professionals (0-1 years of experience) all the factors have a similar type of impact. However, their preference for Communication factor is higher than the other two groups. This indicates that they are dependent on clear communication more than the senior or mid-level employees.

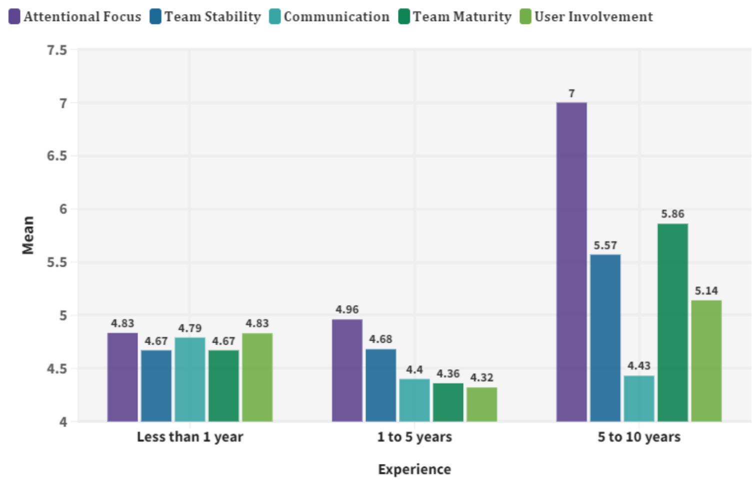


Figure 3.8: Change of Impact Level of Factors with Respect to Professionals' Experience

In figure 9, two patterns or trends of factors with respect to software professionals' experience have been presented. There are two figures side by side in figure 9 where the left figure shows that Attentional Focus and Team Stability are increasing as experience increases. This means that these two factors were more significant during COVID-19 for experienced software professionals compared to less experienced software professionals for delivering software projects timely. On the other hand, the right figure shows that Communication, Team Maturity, and User Involvement have dropped for 1 to 5 years of experienced software professionals but increased again for 5 to 10 years of experienced software professionals. Overall, all the factors were found more significant for experienced (5 to 10 years of experience) software professionals except Communication during COVID-19.

Change of Impact Level of Factors with Respect to Company Size: In figure 10, results show that more emphasis is placed on Attentional Focus in large software companies. However, Team Stability becomes more crucial in mid-sized (100-500) companies. It is quite understandable, as mid-sized companies many things are people dependent. So, the movement of people impacts the delivery. However, most of the big companies are process-oriented which helps them to reduce dependency on individuals. Figure 11 shows that the trend of all the factors upward for large software companies (Over 500).

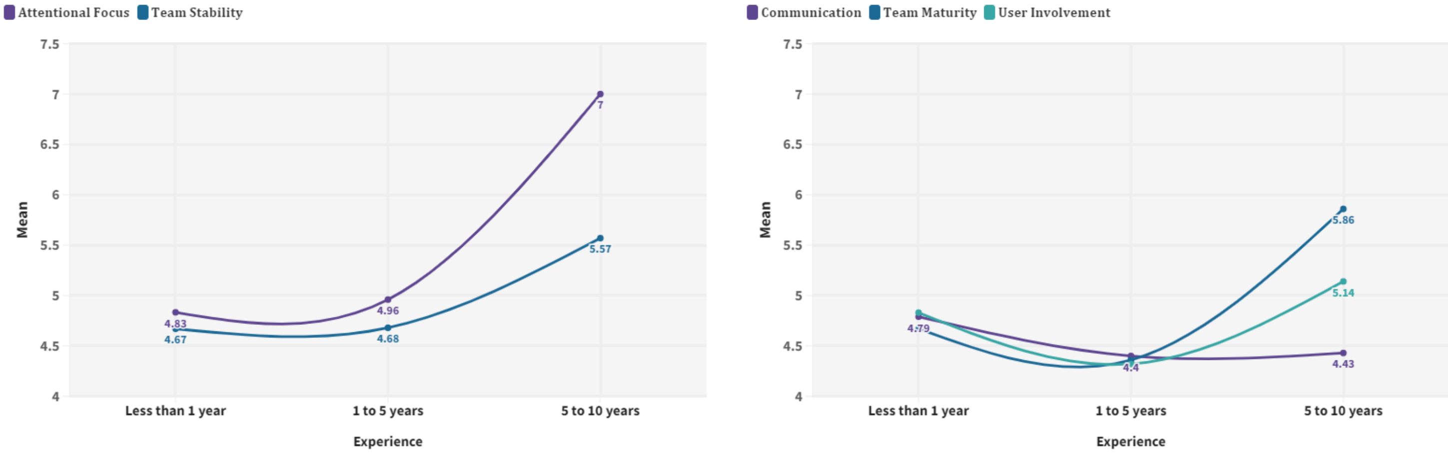


Figure 3.9: Change of Impact Level of Factors with Respect to Professionals' Experience

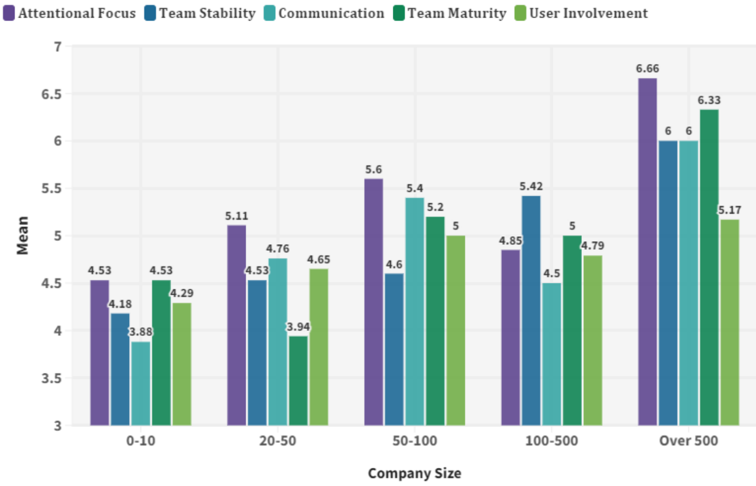


Figure 3.10: Change of Impact Level of Factors with Respect to Company Size

Change of Impact Level of Factors with Respect to different SDLC Models: In this study, we find that the impact level of factors is not the same in different SDLC models. Figure 12 shows that Attentional Focus and Team Maturity were most significant for Agile practitioners during COVID-19. This is very understandable as due to the pandemic people worked remotely and the main challenge was to avoid distraction in the home environment and to focus on the work. At the same time, Team Maturity was also a crucial factor and this is also intuitive as people worked from home therefore, understanding among team members was important to achieve task completion. For Waterfall practitioners, User Involvement was less significant and this is perceptible as in the waterfall model requirements are defined at the beginning, and in the whole development process, users do not get involvement.

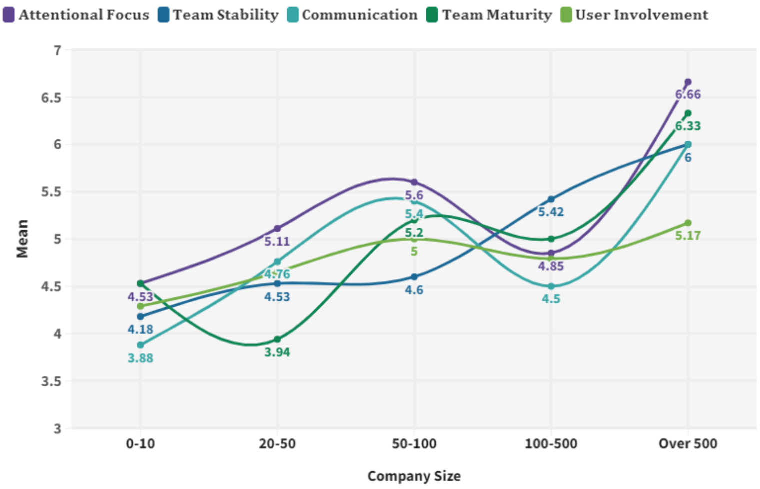


Figure 3.11: Change of Impact Level of Factors with Respect to Company Size

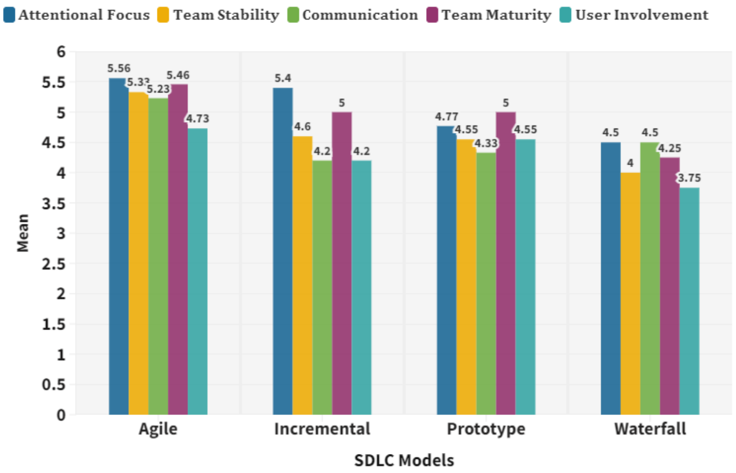


Figure 3.12: Change of Impact Level of Factors with Respect to different SDLC

3.3 Threats to Validity

In this study, there were around two internal threats to validity. Firstly, there was a possibility that the participants following different SDLC models may not have sufficient knowledge to complete the survey questionnaires. Therefore, to prevent introducing bias, we presented survey questionnaires in a random order addressing all common factors of SDLC models which allowed respondents to answer questions according to their comfort zone and understanding. The accuracy of the responses provided by survey respondents creates a second threat to internal validity as each respondent may have different perceptions. Since most of the information in the results is based on respondents' perceptions. Therefore, there can be flaws and misinterpretations of questions resulting in improper answers. Also, given that this survey focused on before and after covid-19 impacts, there is a possibility that participants may recall events differently than they actually happened during covid-19. Hence, there is a chance that the outcome or result may not accurately reflect reality. However, much of the information required for the study could only be gathered through surveys. Therefore, the questions designed were verified and validated by the participants alike technical people (Project manager, Software developer, etc.). They answered close-ended questions avoiding confusion and the data were collected through convenience sampling. This discards the possibility of uncertainty, error, and misinterpretation that's why we believe surveys are a valid approach if correct methods are followed.

The study sample might not be representative of all technical personnel with different roles in the industry. We collected a wide sample of data from various technical personnel that belong to certain phases of SDLC starting from requirement gathering, system design, development, testing, and Infrastructure management. However, different people have different perceptions and they cannot be generalized. But as we have addressed all roles related to each phase of software development and since the members with those roles are experienced and knowledgeable, it can be accredited that the majority will agree to their responses and opinions, therefore proving there is no threat to validity.

There can be many possibilities where the survey respondents may misinterpret the questions of a survey. Therefore, we created and conducted surveys accordingly. Our survey questionnaires were designed and modified from previous literature and verified through peer reviews and research advisors' feedback. Following this, we simulated a test run with 10 participants and identified some vital flaws which were resolved and validated. The survey was categorized into 3 sections which were explained to the respondents without bias and responses were recorded according to the respondents' perspectives.

Chapter 4

Artificial Intelligence in Software Testing: A Systematic Review

Islam, M., Khan, F., Alam, S., & Hasan, M. Artificial Intelligence in Software Testing: A Systematic Review. IEEE Region 10 Technical Conference (TENCON 2023), Thailand.

Software testing has a crucial role in software engineering as it is essential for ensuring the quality, performance, security, and reliability of software systems. By conducting testing, developers can identify and rectify any bugs, or defects in the software, improving its overall functionality and making sure that the software satisfies customer needs and expectations. AI is a vast area , so in this paper we mainly investigate the subarea of AI which are Machine Learning (ML) and Deep Learning (DL) techniques in software testing. The field of software testing currently faces a number of challenges. As software systems grow increasingly complex, it becomes more challenging to manually test all possible scenarios. Also, traditional test automation approaches are time-consuming and complex to implement. Apart from that, keeping pace with agile development is also a challenge as it requires rapid testing. AI has the potential to address these challenges by offering optimized and effective testing strategies. The aim of this study is to gain a thorough understanding of the current state of the field of software testing automation through the use of AI. This review will examine the various methods, techniques, and tools utilized in this domain and evaluate their efficiency. The motivation for this systematic literature review stems from the potential benefits that AI can offer in the field of software testing. AI has the potential to automate the testing process and optimize testing strategies, making software testing more efficient, effective, and accessible. Moreover, AI can address the shortage of skilled testers and help keep pace with the rapid development cycles of agile development methodologies. There are several challenges in software testing that can be solved using AI. Some of these issues include manually generating test cases, test optimization, test results analysis, etc.

The following research questions have been investigated in this research study.

RQ1: Does manual testing have drawbacks?

RQ2: Can integration of AI (ML or DL techniques) in software testing help to overcome the drawbacks of manual testing?

RQ3: What software testing tasks can be automated by AI (ML or DL)?

RQ4: What techniques do researchers use to assess AI (ML or DL) when used in software testing?

In this research study, 40 articles have been screened from different research libraries but through a gradual filtering process, only 20 articles were found suitable for the study. We have structured the paper in the following way. Related works have been discussed in section 2 while the background of software testing and AI have been presented in section 3. Systematic review and the results have been presented in section 4 and 5 consecutively. In the end, conclusion is presented in section 6.

4.1 Software Testing & Artificial Intelligence

Software Testing is a process to evaluate the software and identify defects [67]. It is crucial for software to work or perform as per requirements but it is natural having bugs or defects in software. The bugs can be generated during development, bug fixing, feature addition, code refactoring, and even during software maintenance [68]. Therefore, it is obvious for the development team to test the software under different scenarios before releasing it to the client. There are different strategies and techniques for software testing. Based on the nature of the software it would be decided which software testing technique should be used [69]. Software testing techniques are very tedious and automation comes here to ease the process. How AI can automate software testing and why it is getting more acceptance than any other technique will be discussed in this section. AI is a broad area that encompasses various subareas, and ML is one of the most prominent and widely applied subareas within AI. In this paper, we discuss software testing using Machine Learning (ML). We also focus on software testing using Deep Learning (DL).

4.1.1 Software Testing Using Machine Learning

Machine Learning (ML) is a process where machines learn from data using algorithms and can further predict or make decisions based on the data [70]. The data-centric learning approach has made Machine Learning powerful and widely accepted in different areas including the software industry. Figure 1 shows the general approach to apply Machine Learning algorithms in software testing. There are different activities in software testing like bug detection, generating test data, test case generation, test optimization, API testing, etc [71].

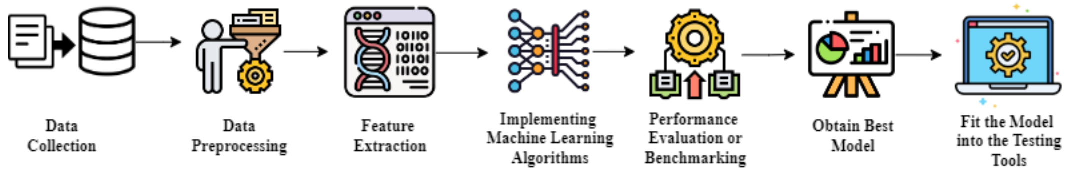


Figure 4.1: A General Approach to Apply ML Techniques in Software Testing

Bug Prediction using ML: Bug prediction can be performed using machine learning. ML algorithms analyze software code and predict the likelihood of future bugs in the code. For performing bug prediction, ML models need to be trained on historical data from past software projects to identify patterns. Once the model is trained, then it can predict the likelihood of bugs occurring in new code [72]. They [73] used supervised ML algorithms to predict software faults based on historical data.

Test Case Generation using ML: In software development, Test case generation from the requirement specifications document is one of the biggest challenges in software testing. Software test cases can be generated using ML. ML model needs to be trained on a set of data where a set of software features are considered as input and the corresponding test cases as output. Finally, the model uses training data to generate new test cases [74].

Test Case Prioritization using ML: Test case prioritization can be performed using machine learning. ML algorithms determine the most critical test cases to execute based on the likelihood of failure and the potential effect on the system. For prioritizing test cases, a machine learning model needs to be trained on a set of labeled data, where a set of software features are considered as input and the corresponding priority level of each test case as output. Finally, the model uses this training data to prioritize new test cases based on their predicted priority level [75].

4.2 Systematic Review

A review is a systematic study that helps to identify the existing work, research question improvement scope, and existing empirical studies [76]. In this study, 20 research papers have been reviewed from the past 7 years which have been collected from 6 different databases such as ScienceDirect, IEEE, SCITEPRESS, ACM, Wiley Online Library and MDPI.

4.2.1 Eligibility Criteria

Eligibility criteria for selecting articles for a systematic literature review include relevance to the research questions, publication time frame, language, publisher, and study design [77]. For this systematic review, after filtering we have selected 20 articles out of

40 articles from the last 7 years. We only selected articles relevant to software testing using machine learning.

4.2.2 Search String Strategy

After lots of searching in the research databases and google scholar, we found many articles about software testing using machine learning but only the most relevant articles were selected. In this process, we have used the advanced search string strategy [78]. In this search string strategy, Boolean operators (AND, OR, NOT) have been used to combine and exclude keywords in the search query.

Our search string was *[("Software Testing" AND "Machine Learning") AND ("Testing Automation Technique" OR "Deep Learning" OR "Black-box Testing" OR "Integration Testing" OR "Metamorphic Testing" OR "White Box Testing") NOT ("Manual Testing" OR "Adhoc testing")].*

Apart from the search string strategy, one can use titles, keywords, or abstracts to find out relevant publications. The purpose of this study is to review the effective application of AI (Machine Learning, Deep Learning) in software testing by developing research questions, collecting and selecting proper relatable studies through filtering methods. By examining the existing literature and answering the research questions, we aim to provide the best current practices for software testing.

4.2.3 Data Screening and Analysis

Each paper examines different aspects of applications of ML, DL techniques in software testing. In most studies, the authors compare different ML and DL models based on their performance and identify the best results they could generate from those models based on the subject criteria and expected outcome. For the collection process, we have purposively identified 40 research papers that are related to ML, DL, and software testing by searching keywords in google scholar. We also used a backward snowballing method where we checked the references of the selected papers and identified 20 papers. After the collection of papers, we started the screening process where by reading the title we would be able to differentiate whether the topic being addressed is relatable or not. Table I shows lists of inclusion and exclusion criteria in detail during paper selection for this study.

The paper selection process involved sorting based on eligibility criteria with a focus on the automation of software testing using machine learning and deep learning techniques. After filtering, 20 research papers were selected for the literature review study. The search strategy consisted of 5 stages: identification of the research topic, screening, selection of eligible papers, and final inclusion of research articles.

Table 4.1: Inclusion and Exclusion Criteria

Area	**Criteria for Inclusion**	**Criteria for Exclusion**
Article type	Research article, SLR	Poster, Book
Searched keywords	Software testing, Machine learning, testing automation technique, Test data generation, Blackbox testing, Whitebox testing	Keywords other than ones in "Inclusion criteria"
Interest of area	Software testing, Software Engineering, Artificial Intelligence	Area excluding "Inclusion criteria"
Language	English	Languages except English
Time period	2015 -2022	Before 2015

Table 4.2: Selected Research Studies According to the Publisher

Publisher Name	# Research Articles
IEEE	8
ACM	6
MDPI	2
Wiley Online Library	1
Science Direct	2
SCITEPRESS	1

4.2.4 Data Extraction

Data extraction means the process of retrieving relevant data from various sources for a specific purpose, such as a literature review [79]. In the context of software testing using machine learning and deep learning, data extraction may involve searching through academic journals, conference proceedings, and other sources to gather information on the latest developments and trends in software testing using ML and DL. This information can then be used to summarise a comprehensive review of the current state of the field, identify gaps in existing knowledge, and provide insights into future directions for research and practice. Table II shows the details of the selected number of studies and their publishers.

4.3 Results

This section provides insights into state-of-the-art techniques and their effectiveness in improving the quality and efficiency of software testing using machine learning and deep learning. The review aims to provide a comprehensive synopsis of the existing research in this domain by analyzing a number of studies. We have selected 20 studies for the study. The details findings of these selected studies have been presented in table III. We also investigated the answer of the research questions from the relevant research papers.

RQ1: Does manual testing have drawbacks?

Manual testing has several drawbacks. Some of the drawbacks of manual testing are it is time-consuming, it does not cover all possible scenarios and use cases, it is costly, it is susceptible to human errors and it can not reproduce test cases accurately [90]. Machine Learning and Deep Learning can help to overcome the mentioned drawbacks of manual testing. ML and DL can automate the testing process. By leveraging the power of algorithms, more accurate testing can be performed [91].

RQ2: Can integration of AI (ML or DL techniques) in software testing help to overcome the drawbacks of manual testing?

Integration of ML and DL techniques in software testing can help to overcome the drawbacks of manual testing by improving the efficiency, accuracy, and effectiveness of the testing process. ML and DL algorithms can be trained to automate repetitive testing tasks, which reduces the required effort for manual testing. This improves the efficiency of the testing process and enables faster testing. ML and DL algorithms can also analyze large amounts of data which help to identify defects in the software system. Identification of the defects improve the accuracy of the testing. Apart from that , ML algorithms can generate test cases using historical data or existing code, optimize the testing by prioritizing test cases [86].

RQ3: What software testing tasks can be automated by AI (ML or DL) ?

Machine Learning and Deep Learning techniques can automate different types of software testing tasks such as test results analysis, test case prioritization, defect prediction, test execution, test case evaluation, test case refinement, testing cost estimation, test oracle construction, identification of metamorphic relations, and test case generation [92]. Table IV shows testing activities that can be automated by machine learning

RQ4: What techniques do researchers use to assess AI (ML or DL) when used in software testing?

Researchers consider different performance matrices to assess ML algorithms when used in software testing. The performance matrices are cross-validation, accuracy, precision, recall, receiver operating characteristic (ROC) curve, area under the curve (AUC), and f1 score [92]. The column total represents the number of papers where these testing activities have been automated by machine learning.

Table 4.3: Summary of the Selected Studies

SL	Source	Year	Findings
1	[20]	2022	Authors proposed an approach utilizing Deep Reinforcement Learning (RL) for automating the exploration of Android apps. Authors developed a tool called ARES along with FATE that integrates with ARES.
2	[80]	2022	This paper analyzed ML frameworks in the context of software automation & evaluated the performance of testing tools considering various factors. Accuracy , scope are important factors to determine the effectiveness .
3	[81]	2022	This study investigates the efficacy of machine learning, data mining, and deep learning methodologies in predicting software faults.
4	[82]	2022	This paper introduces Keeper, a novel testing tool. It creates pseudo-inverse functions for ML APIs.
5	[18]	2021	This study presents DeepOrder, a DL based techniques. DeepOrder considers various factors such as test case duration.
6	[19]	2021	This study investigated reward function and reward strategy . Proposed strategies showed promising results.
7	[83]	2021	This paper introduces Deep GUI. Deep GUI utilizes deep learning techniques to create a model of valid GUI interactions.
8	[83]	2021	This study finds that most ML libraries lack a high-quality unit test suite. Also, discover recurring trends in the unexamined code.
9	[84]	2021	This study presents a DL approach to predict the validity of test inputs for RESTful APIs. It achieved 97% accuracy for the new APIs.
10	[21]	2019	This paper introduces Humanoid, a DL based approach for generating GUI test inputs by leveraging knowledge gained from human interactions.
11	[25]	2019	This study finds equivalent mutants are effective for augmenting data and improving the detection rate of metamorphic relations.
12	[27]	2019	This study introduces an enhanced CNN model specifically designed to improve the learning of semantic representations from source-code.
13	[85]	2019	This study used three supervised ML algorithms for predicting software bugs. The developed models effectively worked for various scenarios.
14	[86]	2019	This study finds ML algorithms have predominantly been employed in different areas of software testing. Test case generation, evaluation, test oracle construction.
15	[87]	2018	This paper describes a tool that generates test data for programs. The tool operates by clustering input data from a corpus folder and creating generative models for each cluster
16	[88]	2018	This paper introduces a methodology called DaOBML, which offers tool support to enhance the quality of environmental models that generate complex artifacts like images or plots.
17	[89]	2017	This study introduces DeepXplore, an innovative whitebox system designed to systematically test DL systems and detect faulty behaviors. DeepXplore can solve joint optimization problems.

Table 4.4: Testing Activities Automated by ML & DL

Software Testing Activity	**Total**
Test Case Generation	4
Defect Prediction	3
Test Case Prioritization	3
Metamorphic Testing	2
Android Testing	2
Test Case Validation	1
White Box Testing	1

Precision: Precision is a statistical measure that quantifies the ratio of true positive instances out of the total positive predictions made. [93].

Recall: Recall is a statistical indicator utilized to quantify the fraction of true positive outcomes within the entirety of actual positive instances [20].

ML and DL algorithms have shown promising results to automate software testing tasks. Some of the promising algorithms are Neural networks, Decision Tree, Support vector machines, and Random Forest, .

Chapter 5

Factors to Form Business Strategy for Online-Based Ride-Sharing Services

Islam, M., Khan, F., Nahar, N., & Hasan, M. Factors to Form Business Strategy for Online-Based Ride-Sharing Services. Second World Conference on Information Systems for Business Management (ISBM), 2023, Springer.

Ride-sharing services are becoming increasingly popular worldwide but the global impact of COVID-19 has been a major setback for ride-sharing companies as they had to experience a negative demand across all regions amid the pandemic. However, the emerging service sector is projected to grow by 7.5% in 2025 from USD 6.68 billion in 2017 [94] which will boost the economy of the transport sector worldwide. The uprising of this market has been due to the marketing strategies adopted which address the lack of proper commute facilities and attract customers by providing better facilities and services at a reasonable price [95]. Bangladesh has a serious traffic jam problem and ride-sharing services are playing an important role to reduce it to some extent. Numerous ride-sharing services exist in Bangladesh such as Uber, Pathao, Sohoz-ride, Obhai, Obon, Amarbike, Texiwala, Chalu, Gariwala, Jobike, etc [96]. From our survey, figure 3 shows that most of the customers belong to Uber and Pathao. People are getting more interested to use ride-sharing services since the internet and mobile usage are very accessible [97].

Ride-sharing companies were badly impacted by the COVID-19 pandemic. This has motivated us to go for this research study and survey the impact of COVID-19 on ride-sharing companies and the change in customers' perception of it. In this study, we found that ride-sharing companies had to change their marketing strategy during the pandemic since the customer had health concerns and even discount offers could not attract customers during the pandemic. In this study, the following research questions have been focused on.

RQ1: Which of the psychological factors influenced your need to use ride-sharing services [Before & During COVID-19 Pandemic]?

RQ2: In the epidemic situation, ride-sharing companies offered some sort of extra services (e.g. door to door groceries, medicines delivery). Do you find it helpful in this epidemic situation like COVID-19]?

RQ3: When a promo or a discount is offered, are you more intending to use a ride-sharing service than usual?

RQ4: What more facilities would you recommend adding to a ride-sharing service for customers in the future?

In this research study, a mixed method has been followed which is the combination of both qualitative and quantitative methods. Significant insights have been derived through statistical analysis.

The main contributions of this paper are:

1. Identification of important factors to form business strategies for ride-sharing services
2. An overview of the impact of COVID-19 and customers' perception on the ride-sharing services

The rest of this paper is organized as follows. The related works have been discussed in section 2. In section 3, the research design has been presented. In section 4, the survey results have been presented. Finally, in section 5, the conclusion has been drawn.

5.1 RESEARCH DESIGN

This research study is designed after a rigorous study of different research articles from which we understood the background of the research area and identified the research gaps. Following this, questions were developed addressing the topic. In this study, we extracted the relevant data and analyzed the implementation of software marketing strategies of ride-sharing companies. When given an opportunity, and time to market proper marketing strategies play a vital role in the ride-sharing service industry. Hence, if a market strategy fails it can have a high impact on a ride-sharing company since it is a very competitive market.

5.1.1 Research Method

The research study was conducted on ride-sharing companies and customers, with a combination of both literature reviews of previous research articles and the results of our conducted survey. We used a blended approach of both qualitative and quantitative methods to collect data. The survey was conducted using convenience and simple random sampling where a sample of 108 respondents was selected randomly and conveniently for conducting the customer survey using a structured questionnaire. Also, a few semi-structured interviews were conducted where 4 other experts belonging to the marketing department of the ride-sharing companies participated, from whom we extracted useful information that permitted adaptability and improvisation. The research methodology, which is presented in Figure 1, was developed based on three stages: research design, data collection & analysis and results.

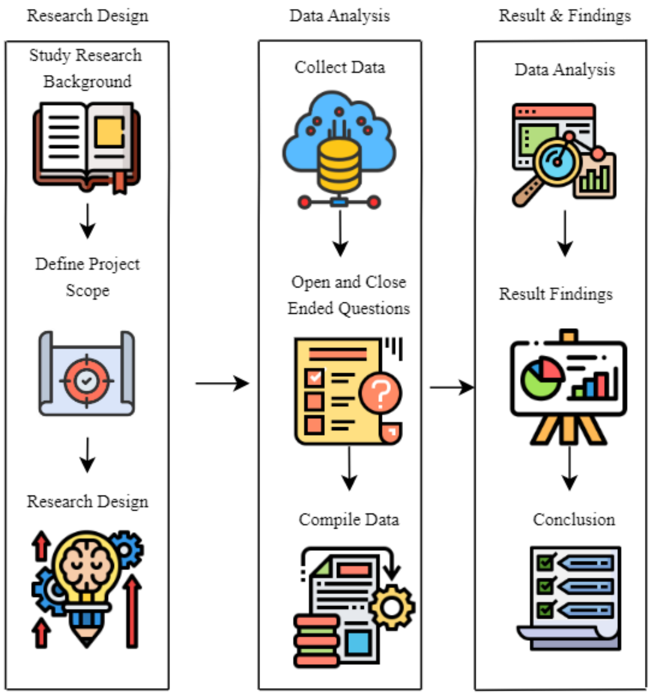


Figure 5.1: Research Methodology

5.1.2 Data Collection and Analysis Method

The surveys were distributed using google forms to the stakeholders; one was for marketing teams of ride-sharing companies and another was for customers who are ride-sharing service users. Also, for data analysis, Python has been used to extract obtained statistical data and to evaluate the variables. Later, we extracted influential factors from collected data affecting companies' software marketing strategies and customers' perceptions of those strategies. However, while gathering data we also faced difficulties with having a stable amount of respondents which restricted our flow of work.

5.2 RESULT AND ANALYSIS

Customers that utilize ride-sharing services like Uber, Pathao, OBhai, etc. provided 108 responses. Conversely, we gathered information from four ride-sharing companies Pathao, Shuttle, Uber, and Obhai. The interviewees' positions were marketing strategist, marketing executive, and business analyst. In this section, firstly current software marketing strategy of ride-sharing companies in Bangladesh has been described which is obtained from our survey data. Secondly, an analysis of the sample's demographic data has been presented. Finally, Important factors to form business strategies, customers' perceptions & impact of COVID-19 have been presented using the survey results.

5.2.1 Software Marketing Strategy of Ride Sharing Companies

Product Strategy of Ride-Sharing Companies: There are many ride-sharing companies in Bangladesh but Uber and Pathao have captured the majority of customers [98]. Our collected survey data also show the same result in figure 3 that 92% of riders have been captured by Uber and Pathao. Uber and Pathao both are offering their services through a mobile application. Uber offers mainly ride services but there are also different categories in ride services such as Uber Intercity, UberX, Uber Premier, Uber XL for different target markets. Pathao also has different products such as PathaoBikes to avoid traffic jams and arrive faster to the destination, PathaoCar, PathaoParcel, PathaoFood, PathaoCourier, and PathaoShop targeting different customer bases.

Pricing Strategy of Ride-Sharing Companies: Ride-sharing companies offer their services to customers at different prices based on the area and supply and demand factors in Bangladesh [99]. In this survey research, we found that 33 respondents out of 108 which is almost 31% of total respondents consider affordable fare while choosing a ride-sharing service shown in figure 1. Companies should consider affordable fare during the formation of the business strategies.

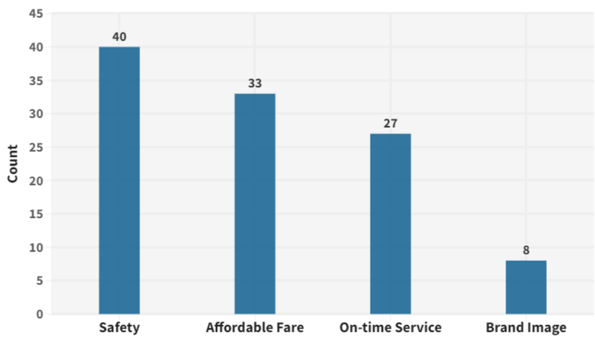


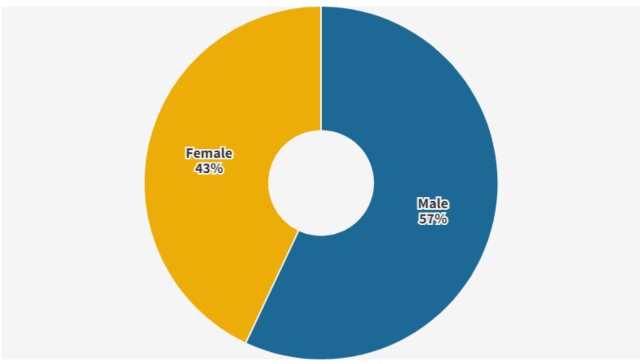
Figure 5.2: Factors while Choosing Ride-Sharing Services

Place and Distribution Strategy of Ride-Sharing Companies: All ride-sharing companies channel their products through mobile applications [30] and the services offered by ride-sharing companies are growing due to the easy accessibility of the internet. Customers can purchase food from their favorite eateries through UberEats, PathaoFood, etc, and enjoy rides wherever they are by simply tapping their preferred rides. Using a single app, customers can track their taxis, parcel, food, and others.

Promotional Strategy of Ride-Sharing Companies: Ride-Sharing companies utilize a variety of tactics to market their brand and attract customers, including advertising, offering free rides, and discount coupons to customers and these are current practices of software marketing strategies [100]. This study finds the discount offer is one of the key marketing strategies, presented in figure 5 that 73% of riders said they tend to use ride-sharing services while the discount is offered. In addition to this, sometimes companies give free rides which is an effective marketing strategy for the business as customers share and promote this kind of experience.

5.2.2 Demographics

Gender and Age Group: The gender breakdown of the participants in this survey is presented in figure 2 (left side figure). The result shows that 43% are female and 57% are male. The findings thus demonstrate that men utilize ride-sharing services more than women but at the same time, a large portion of users are women. Figure 2 (right side figure) depicts the breakdown of survey participants' age groups and demonstrates that 70 participants which is 66% of the total number of participants are between the ages of 23 and 26. The results show that youth are the main users of ride-sharing services.



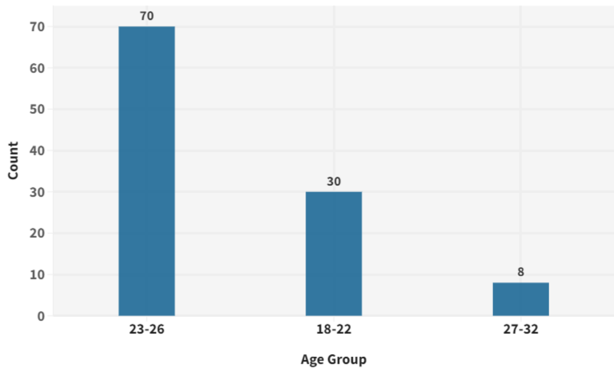


Figure 5.3: Survey Participants' Gender and Age Group Breakdown

Ride Sharing Company Preference: In this survey, we find that most of the riders use Uber and Pathao. The obtained result from the collected data is presented in figure 3. Figure 3 shows that 68 (63%) riders use Uber while 30 (28%) riders use Pathao. Obhai, Shuttle, and Sohoj have 4%, 3%, and 2% respectively.

In this research survey, we also investigated how customers' perceptions changed during COVID-19 and its impact on ride-sharing companies.

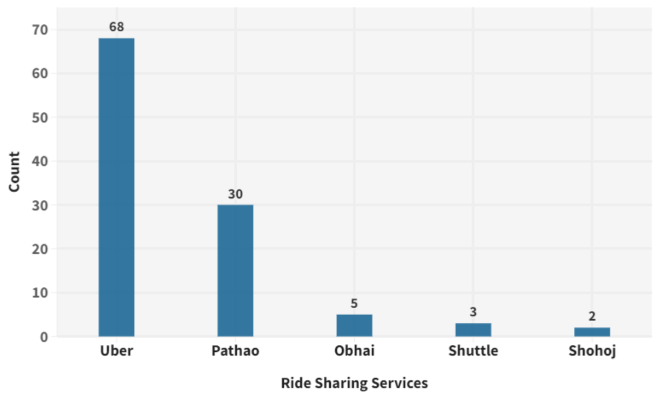


Figure 5.4: Customers' Ride Sharing Company Preference

5.2.3 Factors to form Business Strategies, Customers' Perceptions & Impact of COVID-19

We have already mentioned a few key factors in figure 1 above that companies should consider while forming business strategies which are extracted from our survey data. A few more factors will be presented in this section along with customers' perceptions & impact of COVID-19 extracted from the survey data. We asked riders the following Research Questions (RQs) in this survey.

RQ1:Which of the psychological factors influenced your need to use ride-sharing services [Before and During COVID-19] ?

To obtain an overall understanding of the factors which influenced consumers to use ride-sharing services, we asked RQ1 for both scenarios before and during COVID-19. The obtained result from survey data is presented in figure 4 (left side figure) which shows before COVID-19 the most influential factor was avoiding traffic jams and unavailability of public transport while during COVID-19 the most influential factor was health concern. During COVID-19 ride-sharing companies campaigned for COVID-19 health awareness and they did not allow their riders without masks [101]. Our survey result in figure 4 shows that this marketing strategy worked very well for the companies since the highest percentage of the riders said they choose ride-sharing services during COVID-19 for health concerns. The companies should consider the factors presented in figure 4 during the formation of business strategies.

RQ2: In the epidemic situation, ride-sharing companies offered some sort of extra services (for example: door to door groceries, medicines delivery). Do you find it helpful in this epidemic situation like COVID-19] ?

During the COVID-19, ride-sharing companies offered extra services like medicine delivery, door-to-door groceries, etc. In this survey, we find riders have a positive perception of the services. Figure 5 (left side figure) shows 44% respondents of our survey have a positive perception and 16% have a strong positive perception that these services

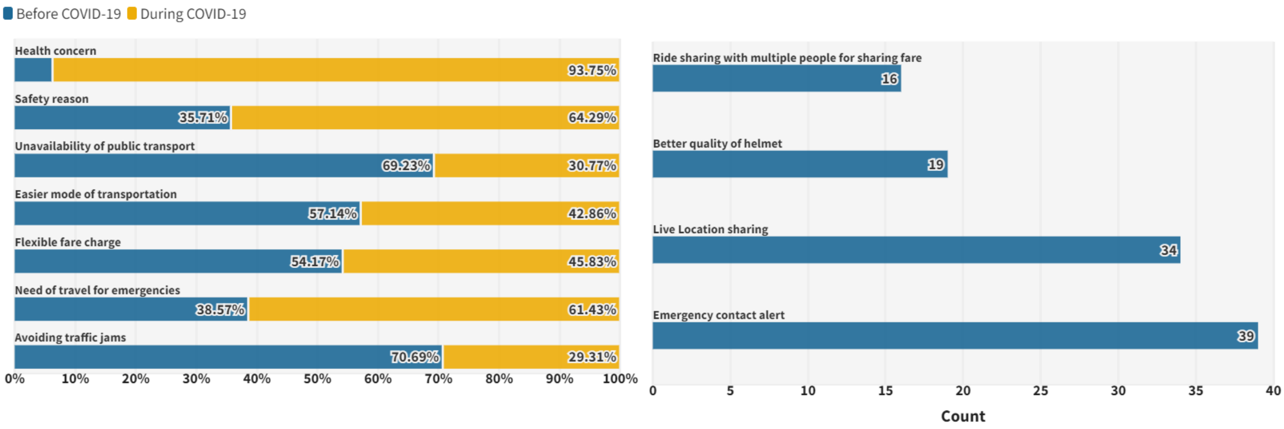


Figure 5.5: Influential factors to Use Ride-sharing Services Before and During COVID-19 **and** Recommended features by customers

are helpful. Therefore, ride-sharing companies should keep continuing to provide these services and consider them in their business strategies.

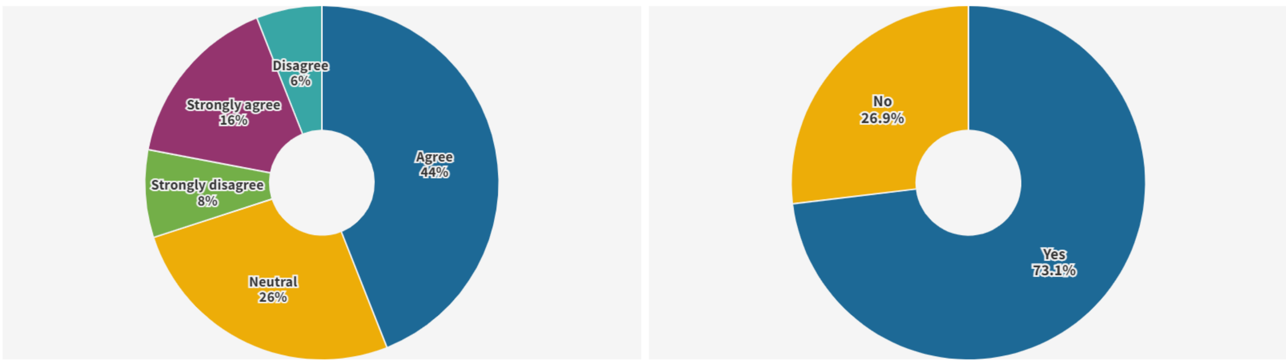


Figure 5.6: Customers' Perception about Extra Services **and** Intention to Use Ride Sharing Services During Discount Offer

RQ3: When a promo or a discount is offered, are you more intending to use a ride-sharing service than usual?

Customers' have a strong intention to use ride-sharing services while promo or discount is offered [102]. Most of the ride-sharing companies have adopted this marketing strategy, our survey results also prove this. Figure 5 (right side figure) depicts that 79 (73%) respondents of this survey said that they intend to use ride-sharing services while promo or discount is offered. Therefore, companies should pay extra attention to promo or discount offers and include them in their business strategies since it is one of the key factors to succeed in ride-sharing software marketing.

RQ4: What more facilities would you recommend adding to a ride-sharing service for you (customer) in the future?

We also asked the riders what kind of features or facilities they would like to have in

Table 5.1: Influential Factors to form the Business Strategies According to Ride-sharing Company Professional

Company Professionals' Perspective				
Influential Factors on Marketing strategies	**Cmp 1**	**Cmp 2**	**Cmp 3**	**Cmp 4**
Covid 19 had negative impacts	✓	✓	✓	✓
Customer retention low	✓	✓	✓	✓
Difficulties in the promotion of the services	✓	✓	✓	✓
Address minority groups			✓	✓
Filter riders based on KPI performance	✓			
Provide new features	✓	✓	✓	✓
Improve training program of drivers	✓	✓		

the ride-sharing services. The results in figure 4 (right side figure) shows that 39 (36%) respondents asked for emergency contact alert feature in the ride-sharing app while 34 respondents (32%) for live Location sharing feature, 19 respondents (18%) asked for a better quality of helmet, and 16 respondents (14%) asked for ride-sharing feature with multiple people so that they can share the fare and commute at a lower cost. Ride-sharing companies should look into these features presented in figure 4 which are recommended by customers in our survey. The suggested features by the riders could be considered as important factors for the companies to form their business strategies. The companies could integrate the suggested features into their existing ride-sharing services and use them in their ride-sharing software marketing to promote their product.

We have surveyed ride-sharing service users and interviewed ride-sharing service marketing and business experts.

From the survey and interview, we have extracted important factors to form the business strategies for ride-sharing services which are listed in Tables 1 and 2 respectively. Table 1 shows that company professionals' perceived COVID-19, difficulties in the promotion, filtering rider's based on KPI, providing new features, and improve training program for the drivers as important factors for their marketing strategy. Due to confidentiality, in table 1, we did not mention the companies' names directly like Uber, Pathao rather we mentioned them as Cmp 1, Cmp 2. In table 1, Cmp 1, Cmp 2, Cmp 3, and Cmp 4 represent Company 1, Company 2, Company 3, and Company 4 respectively.

Table 2 shows that promo and discounts, new features in the ride-sharing application, avoiding traffic jam, health and safety measure, and availability of ride-sharing services during emergency travel are perceived as positive influential factors by users. These perceived factors encourage users to use ride-sharing services. On the other hand, lack of proper amenities, high fares in some cases, poor customer service, COVID-19, and offline rider's service discourage the usage of ride-sharing services.

Table 5.2: Influential factors to form Business Strategies according to ride-sharing service Users

Customers' Perspective	
Negative influential factors that discourage usage of ride sharing services	Positive influential factors that encourage usage of ride sharing services
Decrement in the usage of ride sharing services due to pandemic	Promo offer and discounts
Lack of proper amenities	New features addition in the ride-sharing application
Poor customer service	Arriving at the destination on time by avoiding traffic jam
High fares	Providing health and safety measure during pandemic
Poor customer service	Availability of vehicle during emergency

Chapter 6

Conclusion

The study "Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery: An Empirical Study" investigates the adaptive nature of software companies during the challenges posed by the COVID-19 pandemic. Despite numerous obstacles, the study reveals how these companies managed to deliver software projects on time, showcasing the resilience and adaptability of the software development process in the face of unprecedented circumstances. The second research study "Artificial Intelligence in Software Testing: A Systematic Review" focuses on the integration of AI-driven techniques for software development. This systematic review explores the role of artificial intelligence in software testing. It shows the increasing utilization of AI techniques in software engineering, particularly in testing processes. The study highlights the justification for incorporating AI-driven approaches. It also emphasizes the widespread adoption of AI techniques due to enhanced computer hardware capabilities and the abundance of available data. The third research study "Factors to Form Business Strategy for Online-Based Ride-Sharing Services" examines the marketing perspective within the context of Bangladesh. This study investigates the factors influencing the success of online-based ride-sharing services. The study emphasizes the significance of continuous user feedback and its integration into the software development process. It suggests that incorporating user insights enhances the success rate of the software. It also provides valuable insights for forming business strategies in the competitive ride-sharing industry. These three research studies collectively contribute to the theme of adaptive software engineering by addressing challenges posed by external factors, integrating AI-driven techniques in software testing, and exploring business strategies with a focus on user feedback and integration in the context of online-based ride-sharing services in Bangladesh.

6.0.1 Sustainability

Sustainability of Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery: The research project on the "Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery: An Empirical Study" deals with on-time software project delivery. The study addresses the challenges brought about by COVID-19 on-time software project delivery. Using a quantitative approach the research systematically explores crucial factors across various Software Development Life Cycle (SDLC) models. It provides comprehensive insights applicable to diverse project management methodologies. The identification of key factors such as Attentional Focus, Team Stability, Communication, Team Maturity, and User Involvement during the pandemic, and the recognition of their shifting significance before COVID-19, offer actionable insights for project management strategies. Implementing identified factors for on-time project delivery can lead to efficiency gains and resource optimization in the software company. Adapting these factors will help companies to manage and perform better project delivery which will eventually help to have sustainable project management. Sustainable practices or guidelines involve adapting project management strategies to align with changing conditions, such as those arising from COVID-19. The challenge of sustainability lies in how companies behave when situations arise. Our study offers guidelines on how companies can adapt to such abnormal situations like COVID-19. This analysis will also work as a guideline to evolving project handling based on scenarios.

Sustainability of Artificial Intelligence (AI) in Software Testing: The sustainability of Artificial Intelligence (AI) in software testing is evident in its capacity to address the challenges posed by the growing complexity of software systems. As outlined in our study, the systematic review study highlights the relevance of AI in software testing. It offers a comprehensive analysis of recent trends and the current state of the field. The identified AI techniques, including Machine Learning (ML) and Deep Learning (DL), show a versatility that aligns with the evolving demands of software testing tasks. Practical implications, such as the successful automation of various testing activities show the tangible benefits that AI brings to software testing processes. The positive trend in the adoption of AI-driven techniques further shows its sustainability. It reflects the industry's recognition of AI as a valuable tool for enhancing efficiency.
Cost analysis: The cost of AI-powered testing varies but it is generally lower than the cost of manual testing. This is because AI tools can automate many of the tasks that would otherwise be done by manual testers. According to this research study [103], AI has the potential to automate 70% of software testing tasks and can save up to 12 billion dollars per year for an organization. Furthermore, another study [104] shows that AI-based software testing can reduce 40% of testing costs. These findings show the cost-effectiveness of AI-driven software testing, making it a sustainable choice for organizations.

Sustainability of Factors to Form Business Strategy for Online-Based Ride-

Sharing Services: The sustainability of the research project "Factors to Form Business Strategy for Online-Based Ride-Sharing Services" is perceivable in its relevance to the ride-sharing industry. By addressing critical social, economic, sustainability, and environmental challenges, the study provides a comprehensive resource for stakeholders navigating the evolving landscape of ride-sharing services. The research's sustainability is supported by the important findings of this study for both consumers and ride-sharing companies. The exploration of software marketing strategies during the COVID-19 pandemic contributes to the research's impact and adaptability. The identified factors influencing the ride-sharing business, such as safety prioritization and effective promotional marketing, offer practical insights that are pertinent for the industry seeking to enhance customer retention. The study's acknowledgment of consumer needs, including features like emergency contact alerts and live location sharing highlights its responsiveness to evolving market demands and reinforces its sustainability. The positive consumer perception of extra services during COVID-19 strengthens the relevance of the research by showing the industry's adaptability in the face of external disruptions.

6.0.2 Feasibility

Feasibility of Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery: The feasibility of the study "Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery: An Empirical Study" is perceivable in its focused objective, systematic methodology, and insightful findings. The identification of Attentional Focus, Team Stability, Communication, Team Maturity, and User Involvement as crucial factors during COVID-19, compared to the pre-pandemic period, highlights the study's ability to capture and analyze shifting dynamics within the software development landscape. The willingness of software companies to adopt and integrate the identified factors into their project management practices will make the software development process more feasible. Further training and support to teams need to be given to adapt new strategies for successful implementation of practices. Only then it will be feasible. The findings contribute to the feasibility by offering actionable insights for software development teams and project managers facing dynamic challenges in the ever-evolving field of software engineering.

Feasibility of AI in Software Testing: AI in software testing is feasible as it offers efficiency through rapid automation of repetitive tasks and contributes to faster time-to-market. Its adaptability to complex systems using machine learning addresses challenges posed by intricate software architectures. It enhances test coverage for more reliable software. AI's data-driven testing capabilities proactively identify potential issues related to software testing. Though the initial implementation of AI in software testing would take more resources but long long-term cost savings, minimizing errors, and prevention of post-release bug fixes would be significant. The industry's adoption of AI-driven testing tools and frameworks highlights the technology's acceptance and feasibility. Challenges related to initial investments and tool selection still exist. However, the overall trend shows the growing recognition of AI's potential to revolutionize software testing processes efficiently.

Feasibility of Factors to Form Business Strategy for Online-Based Ride-Sharing Services: The feasibility of the study on "Factors to Form Business Strategy for Online-Based Ride-Sharing Services" is clear in its exploration of critical elements within the ride-sharing industry, considering social, economic, sustainability, and environmental challenges. The study's objective to investigate important factors for ride-sharing services shows a clear research goal that contributes to the feasibility of the research. The positive perception of extra services during COVID-19 indicates the responsiveness of ride-sharing companies to customer needs and emphasizes the feasibility of the research by showing real-world impacts. Apart from that, user acceptance and market adoption of suggested improvements have been made in this research, and adapting to changes has made the strategies of ride-sharing companies feasible.
In summary, each research study presents opportunities for sustainability and feasibility. However, the actual impact and feasibility depend on factors such as technological advancements, market acceptance, and the willingness of industry stakeholders to adopt recommended practices.

6.0.3 Social and Environmental Impact

The research topics On-time software project delivery, Use of AI techniques in software testing and Factors to form business strategy have a vital social and environmental impact. The main goal of these research studies is to help the software industry to identify the flaws and utilize more efficient techniques, practices, and business decisions to overcome the problems socially and ensure the environmental impacts are at a minimum. The following factors are being used in terms of research topics to define the social impact. AI techniques in Software testing can contribute to the automated way of testing software with more accuracy to detect bugs. This can help increase productivity and reduce time-consuming tasks that become financial burdens. This satisfies stakeholder goals.

Social and Environmental Impact of "Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery": The research contributes to understanding how the COVID-19 pandemic has influenced factors affecting on-time software project delivery. It potentially informs strategies for mitigating disruptions in the software development sector which is a social impact of this research. On the other hand, though the primary focus is to identify the factors for on-time project delivery but the study indirectly contributes to environmental sustainability by promoting more efficient and remote work practices. It reduces the need for physical commuting and office spaces which is an environmental impact.

Social and Environmental Impact of Artificial Intelligence in Software Testing: This research study addresses the evolving landscape of software testing through a systematic review. It impacts the workforce by highlighting the need for skilled engineers with knowledge of AI for software testing that is a social impact. On the other hand, the adoption of AI in software testing can lead to more energy-efficient testing processes and contribute to environmental sustainability by optimizing resource utilization which has an environmental impact.

Social and Environmental Impact of Factors to Form Business Strategy for Online-Based Ride-Sharing Services: This study explores factors influencing the business strategy of ride-sharing services. The social impact of this study lies in providing insights into consumer preferences and expectations during and post-COVID-19 which promotes better service adaptation. On the other hand, ride-sharing services contribute to environmental sustainability by reducing traffic congestion. It aligns with the broader environmental goals for sustainable transportation solutions.

6.0.4 Ethics

Ethical Considerations in Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery: In this research, we were aware of ethical considerations in data collection and ensured privacy and consent from participants. We were transparent in data collection and no personal data of the survey participants were disclosed. We presented data carefully in the result section of the study so that it complies with the ethical issues. We did not disclose any personal data even if it contains significant information. We made sure of the privacy of data throughout the study.

Ethical Considerations in Artificial Intelligence in Software Testing: While implementing AI in software testing it must be ensured that ethical AI practices are adopted. Data transparency needs to be ensured along with fairness, and accountability in the development and deployment of AI-driven software testing tools. In the review, we tried to encourage responsible AI adoption and mitigate potential biases in testing algorithms.

Ethical Considerations in Factors to Form Business Strategy for Online-Based Ride-Sharing Services: In this study, we considered ethical issues. Thus we ensured the privacy of survey participants and maintained transparency in data analysis. We did not even mention the company name rather we denoted them as Cmp1, Cmp2, etc in order to preserve the privacy of the company. We did not disclose any participant's data nor a company's data which make sure the research ethics for this study.

6.0.5 Project Summary

The main goal of this graduate project report is to examine important areas of modern technological environments. In the context of software engineering, the research on COVID-19's impact on software project delivery aims to comprehend how different influential factors affect the dynamics of on-time project delivery based on scenario. Concurrently, the investigation of artificial intelligence in software testing seeks to explore cutting-edge methodologies and developments, highlighting the evolving capabilities AI has to integrate automation of software testing. Furthermore, from business perspective, the emphasis is on identifying critical variables and effective marketing tactics in the face of the COVID-19 epidemic that Ride-sharing companies used to attract customers. The key findings from these studies are mentioned next. The study on the impact of COVID-19 on software project delivery revealed vital factors influencing on-time project delivery, showing a significant shift before and during the pandemic. Also, AI in software testing highlighted successful automation across various testing tasks, with a focus on Test Case Generation, Test Case Prioritization, and Defect Prediction. Concurrently, safety emerged as a critical factor during the COVID-19 pandemic for customers using ride-sharing services for which ride-sharing companies adopted different promotional marketing strategies which played a vital role in attracting customers. This integrated graduate project report offers a comprehensive understanding of how software development, marketing, and testing techniques are evolving, and how the subtle impact of external influences like the COVID-19 epidemic are affecting the area. These insights help us understand consumer behavior better, automate software testing processes, and modify project management techniques to respond to shifting conditions. It serves as a valuable resource for practitioners, policymakers, and researchers navigating the dynamic intersection of technology and societal challenges.

6.0.6 Future Work

Future work of our first work titled "Impact of COVID-19 on the Factors Influencing On-Time Software Project Delivery" could extend to a broader geographical scope to assess the global impact of COVID-19 on software project delivery. Also, exploring more specific strategies that software development teams adopted around the world in response to pandemic-related challenges could provide more insights. Future research for the second study titled "Artificial Intelligence in Software Testing" could investigate ethical issues associated with the use of AI in software testing. Also investigating the integration of AI with emerging technologies, such as DevOps practices could provide more insights and understanding of AI's role in modern software development. Future work for the third research article titled "Factors to Form Business Strategy for Online-Based Ride-Sharing Services" may be going for a longitudinal study to track the evolving business strategies of ride-sharing services beyond the COVID-19 time period. Also, exploring the impact of regulatory changes on the industry along with the assessment of the long-term sustainability of safety measures and additional services introduced during COVID-19 could be a valuable area of investigation.

Bibliography

[1] A. H. A. Zahid, M. W. Haider, M. S. Farooq, A. Abid, and A. Ali, “A critical analysis of software failure causes from project management perspectives,” 2018.

[2] T. B. Chiyangwa and E. Mnkandla, “Modelling the critical success factors of agile software development projects in south africa,” *South African Journal of Information Management*, vol. 19, no. 1, pp. 1–8, 2017.

[3] S. Dhir, D. Kumar, and V. Singh, “Success and failure factors that impact on project implementation using agile software development methodology,” in *Software engineering*, pp. 647–654, Springer, 2019.

[4] P. Ram, P. Rodriguez, M. Oivo, and S. Martínez-Fernández, “Success factors for effective process metrics operationalization in agile software development: A multiple case study,” in *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, pp. 14–23, IEEE, 2019.

[5] S. A. K. Ghayyur, S. Ahmed, M. Ali, A. Razzaq, N. Ahmed, and A. Naseem, “A systematic literature review of success factors and barriers of agile software development,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 278–291, 2018.

[6] A. Srivastava, D. Mehrotra, P. Kapur, and A. G. Aggarwal, “Analytical evaluation of agile success factors influencing quality in software industry,” *International Journal of System Assurance Engineering and Management*, vol. 11, no. 2, pp. 247–257, 2020.

[7] R. Marques, G. Costa, M. Silva, and P. Gonçalves, “A survey of failures in the software development process,” 2017.

[8] S. A. Butt and T. Jamal, “Frequent change request from user to handle cost on project in agile model,” *Proc. of Asia Pacific Journal of Multidisciplinary Research*, vol. 5, no. 2, pp. 26–42, 2017.

[9] T. Bergmann and W. Karwowski, "Agile project management and project success: A literature review," in *International Conference on Applied Human Factors and Ergonomics*, pp. 405–414, Springer, 2018.

[10] F. Hujainah, R. B. A. Bakar, M. A. Abdulgabber, and K. Z. Zamli, "Software requirements prioritisation: a systematic literature review on significance, stakeholders, techniques and challenges," *IEEE Access*, vol. 6, pp. 71497–71523, 2018.

[11] H. Edison, X. Wang, and K. Conboy, "Comparing methods for large-scale agile software development: A systematic literature review," *IEEE Transactions on Software Engineering*, 2021.

[12] M. A. Akbar, J. Sang, A. A. Khan, S. Mahmood, S. F. Qadri, H. Hu, H. Xiang, *et al.*, "Success factors influencing requirements change management process in global software development," *Journal of Computer Languages*, vol. 51, pp. 112–130, 2019.

[13] N. Saleem, "Empirical analysis of critical success factors for project management in global software development," in *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*, pp. 68–71, IEEE, 2019.

[14] E. Van Kelle, J. Visser, A. Plaat, and P. van der Wijst, "An empirical study into social success factors for agile software development," in *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, pp. 77–80, IEEE, 2015.

[15] S. Hussain, Z. Fangwei, A. F. Siddiqi, Z. Ali, and M. S. Shabbir, "Structural equation model for evaluating factors affecting quality of social infrastructure projects," *Sustainability*, vol. 10, no. 5, p. 1415, 2018.

[16] T. Kamal, Q. Zhang, M. A. Akbar, M. Shafiq, A. Gumaei, and A. Alsanad, "Identification and prioritization of agile requirements change management success factors in the domain of global software development," *IEEE Access*, vol. 8, pp. 44714–44726, 2020.

[17] G. Arcos-Medina and D. Mauricio, "Aspects of software quality applied to the process of agile software development: a systematic literature review," *International Journal of System Assurance Engineering and Management*, vol. 10, no. 5, pp. 867–897, 2019.

[18] A. Sharif, D. Marijan, and M. Liaaen, "Deeporder: Deep learning for test case prioritization in continuous integration testing," in *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 525–534, IEEE, 2021.

[19] Y. Yang, Z. Li, L. He, and R. Zhao, "A systematic study of reward for reinforcement learning based continuous integration testing," *Journal of Systems and Software*, vol. 170, p. 110787, 2020.

[20] A. Romdhana, A. Merlo, M. Ceccato, and P. Tonella, "Deep reinforcement learning for black-box testing of android apps," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 31, no. 4, pp. 1–29, 2022.

[21] Y. Li, Z. Yang, Y. Guo, and X. Chen, "Humanoid: A deep learning-based approach to automated black-box android app testing," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 1070–1073, IEEE, 2019.

[22] F. YazdaniBanafsheDaragh and S. Malek, "Deep gui: Black-box gui input generation with deep learning," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 905–916, IEEE, 2021.

[23] U. Kanewala, J. M. Bieman, and A. Ben-Hur, "Predicting metamorphic relations for testing scientific software: a machine learning approach using graph kernels," *Software testing, verification and reliability*, vol. 26, no. 3, pp. 245–269, 2016.

[24] R. Braga, P. S. Neto, R. Rabêlo, J. Santiago, and M. Souza, "A machine learning approach to generate test oracles," in *Proceedings of the XXXII Brazilian Symposium on Software Engineering*, pp. 142–151, 2018.

[25] A. Nair, K. Meinke, and S. Eldh, "Leveraging mutants for automatic prediction of metamorphic relations using machine learning," in *Proceedings of the 3rd ACM SIGSOFT International Workshop on Machine Learning Techniques for Software Quality Evaluation*, pp. 1–6, 2019.

[26] P. D. Singh and A. Chug, "Software defect prediction analysis using machine learning algorithms," in *2017 7th international conference on cloud computing, data science & engineering-confluence*, pp. 775–781, IEEE, 2017.

[27] C. Pan, M. Lu, B. Xu, and H. Gao, "An improved cnn model for within-project software defect prediction," *Applied Sciences*, vol. 9, no. 10, p. 2138, 2019.

[28] M. Moslehpour, T. Ismail, B. Purba, and W.-K. Wong, "What makes go-jek go in indonesia? the influences of social media marketing activities on purchase intention," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 17, no. 1, pp. 89–103, 2021.

[29] I. Lampe, R. Alatas, N. Orynka, and G. B. R. Saputra, "Local online courier and ride-hailing service social media marketing," *Komunikator*, vol. 13, no. 1, pp. 66–77, 2021.

[30] M. T. Islam, T. M. H. Arif, and O. G. Alvi, "A comparative study on the performance of existing ride-hailing companies in bangladesh: Management and customer service perspective," *Himalayan Journal of Economics and Business Management*, vol. 3, no. 2, 2022.

[31] O. N. Abdullah, F. A. Likhon, A. A. Parvez, and I. Ishtiaque, "Pathao: A tech start-up that shook bangladesh," *North South Business Review*, p. 11.

[32] G. W. Widyatama, S. Chelliah, Y. Kai, Y. Yingxing, Y. C. Tien, W. C. Mey, and L. G. Sin, "Grab marketing strategy, research & development," *International journal of Tourism and hospitality in Asia Pasific (IJTHAP)*, vol. 3, no. 2, pp. 97–104, 2020.

[33] N. F. I. Zailani, A. Albattat, A. H. Sulaiman, I. A. A. Nazari, and N. F. N. Nasirman, "Factors influencing consumer perception on ride-sharing application services: a case study of grab car," *Psychology and Education*, vol. 57, no. 9, pp. 2490–2495, 2020.

[34] E. E. Izogo and M. Mpinganjira, "Behavioral consequences of customer inspiration: the role of social media inspirational content and cultural orientation," *Journal of Research in Interactive Marketing*, vol. 14, no. 4, pp. 431–459, 2020.

[35] B. Godey, A. Manthiou, D. Pederzoli, J. Rokka, G. Aiello, R. Donvito, and R. Singh, "Social media marketing efforts of luxury brands: Influence on brand equity and consumer behavior," *Journal of business research*, vol. 69, no. 12, pp. 5833–5841, 2016.

[36] Y. Upadhyay, J. Paul, and R. Baber, "Effect of online social media marketing efforts on customer response," *Journal of Consumer Behaviour*, vol. 21, no. 3, pp. 554–571, 2022.

[37] W. Handayani, Y. T. Suyoto, T. Prasetio, E. Pitaloka, H. Wijoyo, H. Yonata, I. R. Koho, Y. Cahyono, *et al.*, "The effect of social media marketing on brand trust, brand equity and brand loyalty.," *International Journal of Data & Network Science*, vol. 6, no. 3, 2022.

[38] S.-C. Chen and C.-P. Lin, "Understanding the effect of social media marketing activities: The mediation of social identification, perceived value, and satisfaction," *Technological forecasting and social change*, vol. 140, pp. 22–32, 2019.

[39] M. Y. Kang and B. Park, "Sustainable corporate social media marketing based on message structural features: Firm size plays a significant role as a moderator," *Sustainability*, vol. 10, no. 4, p. 1167, 2018.

[40] A. N. Mason, J. Narcum, and K. Mason, "Social media marketing gains importance after covid-19," *Cogent Business & Management*, vol. 8, no. 1, p. 1870797, 2021.

[41] D. O. Dastane, "Impact of digital marketing on online purchase intention: Mediation effect of customer relationship management," *Journal of Asian Business Strategy, DOI*, vol. 10, pp. 142–158, 2020.

[42] A. Wibowo, S.-C. Chen, U. Wiangin, Y. Ma, and A. Ruangkanjanases, "Customer behavior as an outcome of social media marketing: The role of social media marketing activity and customer experience," *Sustainability*, vol. 13, no. 1, p. 189, 2020.

[43] E. Dias Canedo and B. Cordeiro Mendes, "Software requirements classification using machine learning algorithms," *Entropy*, vol. 22, no. 9, p. 1057, 2020.

[44] R. R. Mekala, A. Irfan, E. C. Groen, A. Porter, and M. Lindvall, "Classifying user requirements from online feedback in small dataset environments using deep learning," in *2021 IEEE 29th International requirements engineering conference (RE)*, pp. 139–149, IEEE, 2021.

[45] T. Brown, "Machine learning (ml) framework for identifying inconsistency in software requirement documents (srd),"

[46] M. M. H. Abid, R. N. Tanna, M. R. Noyon, J. H. Masud, T. Akter, *et al.*, *Automation requirements engineering using machine learning*. PhD thesis, Brac University, 2021.

[47] P. Talele and R. Phalnikar, "Software requirements classification and prioritisation using machine learning," in *Machine Learning for Predictive Analysis: Proceedings of ICTIS 2020*, pp. 257–267, Springer, 2021.

[48] M. Binkhonain and L. Zhao, "A review of machine learning algorithms for identification and classification of non-functional requirements," *Expert Systems with Applications: X*, vol. 1, p. 100001, 2019.

[49] I. Atoum, J. Almalki, S. M. Alshahrani, and W. Al Shehri, "Towards measuring user experience based on software requirements," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, 2021.

[50] B. Khan, R. Naseem, I. Alam, I. Khan, H. Alasmary, and T. Rahman, "Analysis of tree-family machine learning techniques for risk prediction in software requirements," *IEEE Access*, vol. 10, pp. 98220–98231, 2022.

[51] Z. Kurtanović and W. Maalej, "Automatically classifying functional and non-functional requirements using supervised machine learning," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 490–495, Ieee, 2017.

[52] H. Villamizar, T. Escovedo, and M. Kalinowski, "Requirements engineering for machine learning: A systematic mapping study," in *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 29–36, IEEE, 2021.

[53] I. Morales-Ramirez, D. Munante, F. Kifetew, A. Perini, A. Susi, and A. Siena, "Exploiting user feedback in tool-supported multi-criteria requirements prioritization," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pp. 424–429, IEEE, 2017.

[54] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *Journal of systems and software*, vol. 81, no. 6, pp. 961–971, 2008.

[55] M. Jørgensen, "A review of studies on expert estimation of software development effort," *Journal of Systems and Software*, vol. 70, no. 1-2, pp. 37–60, 2004.

[56] E. Kula, E. Greuter, A. Van Deursen, and G. Gousios, "Factors affecting on-time delivery in large-scale agile software development," *IEEE Transactions on Software Engineering*, vol. 48, no. 9, pp. 3573–3592, 2021.

[57] C. Tam, E. J. da Costa Moura, T. Oliveira, and J. Varajão, "The factors influencing the success of on-going agile software development projects," *International Journal of Project Management*, vol. 38, no. 3, pp. 165–176, 2020.

[58] S. E. Harpe, "How to analyze likert and other rating scale data," *Currents in pharmacy teaching and learning*, vol. 7, no. 6, pp. 836–850, 2015.

[59] M. A. Laitinen, "Net promoter score as indicator of library customers' perception," *Journal of Library Administration*, vol. 58, no. 4, pp. 394–406, 2018.

[60] D. Dönmez, G. Grote, and S. Brusoni, "Routine interdependencies as a source of stability and flexibility. a study of agile software development teams," *Information and Organization*, vol. 26, no. 3, pp. 63–83, 2016.

[61] A. Rasheed, B. Zafar, T. Shehryar, N. A. Aslam, M. Sajid, N. Ali, S. H. Dar, and S. Khalid, "Requirement engineering challenges in agile software development," *Mathematical Problems in Engineering*, vol. 2021, 2021.

[62] S. L. Ramírez-Mora, H. Oktaba, and J. Patlán Pérez, "Group maturity, team efficiency, and team effectiveness in software development: A case study in a cmmi-dev level 5 organization," *Journal of Software: Evolution and Process*, vol. 32, no. 4, p. e2232, 2020.

[63] M. Bano, D. Zowghi, and F. da Rimini, "User involvement in software development: the good, the bad, and the ugly," *IEEE Software*, vol. 35, no. 6, pp. 8–11, 2018.

[64] S. D. Vishnubhotla, E. Mendes, and L. Lundberg, "An insight into the capabilities of professionals and teams in agile software development: A systematic literature review," in *Proceedings of the 2018 7th International Conference on Software and Computer Applications*, pp. 10–19, 2018.

[65] D. E. Harter and S. A. Slaughter, "Quality improvement and infrastructure activity costs in software development: A longitudinal analysis," *Management Science*, vol. 49, no. 6, pp. 784–800, 2003.

[66] N. B. Moe, T. Dingsøyr, and T. Dybå, "Overcoming barriers to self-management in software teams," *IEEE software*, vol. 26, no. 6, pp. 20–26, 2009.

[67] G. J. Myers, T. Badgett, T. M. Thomas, and C. Sandler, *The art of software testing*, vol. 2. Wiley Online Library, 2004.

[68] L. Tan, C. Liu, Z. Li, X. Wang, Y. Zhou, and C. Zhai, "Bug characteristics in open source software," *Empirical software engineering*, vol. 19, pp. 1665–1705, 2014.

[69] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad, "Software testing techniques: A literature review," in *2016 6th international conference on information and communication technology for the Muslim world (ICT4M)*, pp. 177–182, IEEE, 2016.

[70] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[71] Z.-H. Zhou, *Machine learning*. Springer Nature, 2021.

[72] M. Efendioglu, A. Sen, and Y. Koroglu, "Bug prediction of systemc models using machine learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 3, pp. 419–429, 2018.

[73] A. Hammouri, M. Hammad, M. Alnabhan, and F. Alsarayrah, "Software bug prediction using machine learning approach," *International journal of advanced computer science and applications*, vol. 9, no. 2, 2018.

[74] D. Zhang, “Machine learning in value-based software test data generation,” in *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’06)*, pp. 732–736, IEEE, 2006.

[75] B. Kitchenham and P. Brereton, “A systematic review of systematic review process research in software engineering,” *Information and software technology*, vol. 55, no. 12, pp. 2049–2075, 2013.

[76] D. Papaioannou, A. Sutton, and A. Booth, “Systematic approaches to a successful literature review,” *Systematic approaches to a successful literature review*, pp. 1–336, 2016.

[77] H. A. Mohamed Shaffril, S. F. Samsuddin, and A. Abu Samah, “The abc of systematic literature review: the basic methodological guidance for beginners,” *Quality & Quantity*, vol. 55, pp. 1319–1346, 2021.

[78] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, *et al.*, “The prisma 2020 statement: an updated guideline for reporting systematic reviews,” *International journal of surgery*, vol. 88, p. 105906, 2021.

[79] S. R. Jonnalagadda, P. Goyal, and M. D. Huffman, “Automating data extraction in systematic reviews: a systematic review,” *Systematic reviews*, vol. 4, no. 1, pp. 1–16, 2015.

[80] S. Fatima, B. Mansoor, L. Ovais, S. A. Sadruddin, and S. A. Hashmi, “Automated testing with machine learning frameworks: A critical analysis,” *Engineering Proceedings*, vol. 20, no. 1, p. 12, 2022.

[81] I. Batool and T. A. Khan, “Software fault prediction using data mining, machine learning and deep learning techniques: A systematic literature review,” *Computers and Electrical Engineering*, vol. 100, p. 107886, 2022.

[82] C. Wan, S. Liu, S. Xie, Y. Liu, H. Hoffmann, M. Maire, and S. Lu, “Automated testing of software that uses machine learning apis,” in *Proceedings of the 44th International Conference on Software Engineering*, pp. 212–224, 2022.

[83] S. Wang, N. Shrestha, A. K. Subburaman, J. Wang, M. Wei, and N. Nagappan, “Automatic unit test generation for machine learning libraries: How far are we?,” in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 1548–1560, IEEE, 2021.

[84] A. G. Mirabella, A. Martin-Lopez, S. Segura, L. Valencia-Cabrera, and A. Ruiz-Cortés, “Deep learning-based prediction of test input validity for restful apis,” in

2021 IEEE/ACM Third International Workshop on Deep Learning for Testing and Testing for Deep Learning (DeepTest), pp. 9–16, IEEE, 2021.

[85] S. D. Immaculate, M. F. Begam, and M. Floramary, "Software bug prediction using supervised machine learning algorithms," in *2019 International conference on data science and communication (IconDSC)*, pp. 1–7, IEEE, 2019.

[86] V. H. Durelli, R. S. Durelli, S. S. Borges, A. T. Endo, M. M. Eler, D. R. Dias, and M. P. Guimarães, "Machine learning applied to software testing: A systematic mapping study," *IEEE Transactions on Reliability*, vol. 68, no. 3, pp. 1189–1212, 2019.

[87] C. Paduraru and M.-C. Melemciuc, "An automatic test data generation tool using machine learning.," in *ICSOFT*, pp. 506–515, 2018.

[88] V. A. de Santiago, L. A. R. da Silva, and P. R. de Andrade Neto, "Testing environmental models supported by machine learning," in *Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing*, pp. 3–12, 2018.

[89] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *proceedings of the 26th Symposium on Operating Systems Principles*, pp. 1–18, 2017.

[90] A. Leitner, I. Ciupa, B. Meyer, and M. Howard, "Reconciling manual and automated testing: The autotest experience," in *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pp. 261a–261a, IEEE, 2007.

[91] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine learning testing: Survey, landscapes and horizons," *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 1–36, 2020.

[92] Q. Song, Y. Guo, and M. Shepperd, "A comprehensive investigation of the role of imbalanced learning for software defect prediction," *IEEE Transactions on Software Engineering*, vol. 45, no. 12, pp. 1253–1269, 2018.

[93] N. Tatbul, T. J. Lee, S. Zdonik, M. Alam, and J. Gottschlich, "Precision and recall for time series," *Advances in neural information processing systems*, vol. 31, 2018.

[94] W.-L. Shiau, H. Chen, K. Chen, Y.-H. Liu, and F. T. C. Tan, "A cross-cultural perspective on the blended service quality for ride-sharing continuance," *Journal of Global Information Management (JGIM)*, vol. 29, no. 6, pp. 1–25, 2021.

[95] E. Yunus, D. Susilo, S. Riyadi, M. Indrasari, and T. D. Putranto, “The effectiveness marketing strategy for ride-sharing transportation: Intersecting social media, technology, and innovation,” *Entrepreneurship and Sustainability Issues*, vol. 7, no. 2, p. 1424, 2019.

[96] S. Islam, E. Huda, F. Nasrin, and M. Freelanch Researcher, “Ride-sharing service in bangladesh: Contemporary states and prospects,” *International Journal of Business and Management*, vol. 14, no. 9, pp. 65–75, 2019.

[97] C.-M. Chen, D. Shallcross, Y.-C. Shih, Y.-C. Wu, S.-P. Kuo, Y.-Y. Hsi, Y. Holderby, and W. Chou, “Smart ride share with flexible route matching,” in *13th International Conference on Advanced Communication Technology (ICACT2011)*, pp. 1506–1510, IEEE, 2011.

[98] N. Kumar, N. Jafarinaimi, and M. Bin Morshed, “Uber in bangladesh: The tangled web of mobility and justice,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. CSCW, pp. 1–21, 2018.

[99] M. N. Sakib, “The ride-sharing services in bangladesh: Current status, prospects, and challenges,” *European Journal of Business and Management ISSN*, pp. 2222–1905, 2019.

[100] S. Ahmed, M. M. Choudhury, E. Ahmed, U. Y. Chowdhury, and A. Al Asheq, “Passenger satisfaction and loyalty for app-based ride-sharing services: through the tunnel of perceived quality and value for money,” *The TQM Journal*, vol. 33, no. 6, pp. 1411–1425, 2020.

[101] M. N. Al-Ameen, T. Tamanna, S. Nandy, M. M. Ahsan, P. Chandra, and S. I. Ahmed, “We don’t give a second thought before providing our information: understanding users’ perceptions of information collection by apps in urban bangladesh,” in *Proceedings of the 3rd ACM SIGCAS Conference on Computing and Sustainable Societies*, pp. 32–43, 2020.

[102] H. Rasheed Gaber and A. M. Elsamadicy, “What drives customers to continue using ride-sharing apps during the covid-19 pandemic? the case of uber in egypt,” *Cogent Business & Management*, vol. 8, no. 1, p. 1944009, 2021.

[103] J. Rymer and K. Appian, “The forrester wave™: Low-code development platforms for ad&d pros, q4 2017,” *Cambridge, MA: Forrester Research*, 2017.

[104] G. Uğur-Tuncer, C. Davutoglu, and N. M. Durakbasa, “Intelligent test automation for improved software quality assurance,” in *Digital Conversion on the Way to Industry 4.0: Selected Papers from ISPR2020, September 24-26, 2020 Online-Turkey*, pp. 614–624, Springer, 2021.