Independent University **Bangladesh (IUB)**

IUB Academic Repository

Graduate Thesis 2025

2025

Role of DevOps in the Lifecycle of Software Development in the context of Bangladesh

Nawar, Ramisa

IUB

http://ar.iub.edu.bd/handle/11348/1032

Downloaded from IUB Academic Repository

Role of DevOps in the Lifecycle of Software Development in the context of Bangladesh

By:

Ramisa Nawar
ID: **2120297**

Autumn 2025

Department of Computer Science and Engineering, Independent University, Bangladesh

Supervised by:

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

Attestation

I, Ramisa Nawar (2120297), certify that the report was completed by me and submitted to partial fulfillment of the requirement of a Bachelor's Degree in Computer Science and Engineering from Independent University, Bangladesh (IUB). This project was completed under the supervision of Dr. Saadia Binte Alam. All sources of information used in this project have been verified, and the report has been taken care of in it.

_____ _____

Signature Date

_____

Name

Acknowledgement

I would like to convey my sincere appreciation to Professor, Dr. Saadia Binte Alam, Independent University, Bangladesh. Her steady guidance, thoughtful suggestions, and patience played a pivotal role in the successful completion of the project. Her expertise and mentorship greatly improved the quality and depth of the work, and I am truly appreciative of her unwavering support throughout my academic journey.

Additionally, my gratitude also extends to Ms. Farzana Sadia for her insightful input and consistent support. Her guidance significantly contributed to the execution of this project. I am grateful for the opportunities and resources that contributed to the growth and development of the topic.

I acknowledge and appreciate the contributions of all individuals who showed support, both directly and indirectly, during this endeavor.

Letter of Transmittal

_____, 2025

Dr. Saadia Binte Alam

Associate Professor and Head

Department of Computer Science and Engineering

Independent University, Bangladesh

Subject: Submission of Senior Project Report on “Role of DevOps in the Lifecycle of Software Development in the context of Bangladesh”

Dear Madam, I am honored to submit my senior project report titled “Role of DevOps in the Lifecycle of Software Development in the context of Bangladesh”. This report represents the culmination of my efforts, research, and practical application of various emerging technologies under your esteemed supervision.

Throughout the development of the framework, I incorporated dataset integration strategies, real-world data collection , and data analysis techniques that have significantly enhanced my technical knowledge and collaborative skills. I believe this report provides a comprehensive overview of my work and serves its intended purpose effectively. I would like to extend my heartfelt gratitude for your continuous support, insightful feedback, and invaluable mentorship throughout the project. Your guidance played a vital role in helping me navigate challenges and reach my objectives. I sincerely hope this report meets your expectations and is found to be satisfactory.

Thank you once again for your encouragement and assistance throughout the semester.

Yours sincerely,

Ramisa Nawar (2120297)

Abstract

DevOps (Development and Operations team) is a foundational approach in the recent software engineering scene, integrating it to improve collaboration, automation, and continuous delivery. Its adoption varies across countries with different organizational cultures, resources, and technological infrastructures. In Bangladesh, where the software industry is expanding rapidly along with new technologies, DevOps integration plays a key role. However, there is limited empirical research examining DevOps' role in the Software Development Lifecycle (SDLC). This study addresses this gap by conducting a unified, multi-dimensional investigation across three domains: Critical Success Factors (CSFs), DevOps Capability Maturity (CMMI), and Security integration within DevOps practices (DevSecOps). Data were collected from around 79 professionals, using structured questionnaires derived from global frameworks and models. CSF analysis used a Multiple Linear Regression (MLR) statistical model to quantify the relative association of 19 CSFs with perceived project success, revealing from the exploratory sample that automation-related practices and intra-organizational collaboration showed the largest estimated effect sizes. The maturity assessment used reliability analysis, correlation analysis, and K-means (K = 3) clustering, showing variation across CMMI capabilities: Monitoring and Continuous Development showed higher maturity. Security integration was evaluated using non-parametric methods, including Spearman Rank Correlation, permutation tests, and FDR correction, which demonstrated that security is most effectively integrated at build and pipeline stages. These emphasize the interrelationship between automation, culture, maturity, and security as determinants of DevOps effectiveness in Bangladesh. The study proposes a refined CSF model, a maturity model, and security practices evaluation. The models are tailored according to Bangladesh’s software firm needs.

Contents

1 Introduction..... 10
1.1 Background..... 10
1.2 Problem Statement..... 11
1.3 Outcome of the project..... 18
2 Literature Review..... 20
2.1 Literature Selection Procedure..... 20
2.2 Keywords Searched..... 20
2.3 Databases for Search..... 20
2.4 Existing methods, frameworks, and models..... 20
2.4.1 DevOps Critical Success Factors (CSFs)..... 20
2.4.2 DevOps Capability Maturity Model..... 21
2.4.3 Security in DevOps (DevSecOps)..... 23
2.5 Gap Analysis..... 25
3. Research Methodology..... 26
3.1 Motivation and Purpose..... 26
3.2 Research Approach..... 26
3.3 Sampling and Participants..... 27
3.4 Data Collection..... 29
3.5 Data Analysis..... 29
3.5.1 Critical Success Factors Analysis..... 30
3.5.2 Maturity Model Assessment..... 30
3.5.3 Security Frameworks Evaluation..... 31
3.6 Limitations of the Methodology..... 31
4. Findings and Result Analysis..... 32
4.1 DevOps Critical Success Factors..... 32
4.2 DevOps Maturity Across the Industry..... 34
4.3 Security Integrations in DevOps..... 36
4.4 Patterns Observed in DevOps teams..... 37
5. Proposed Framework..... 39
5.1 DevOps CSF Model..... 39
5.2 DevOps Maturity Assessment Model..... 40
5.3 Security in DevOps Evaluation Model..... 40
6. Conclusion and Future Works..... 42
6.1 The Role of DevOps Across the Software Development Lifecycle (SDLC)..... 42

6.1.1 Mapping Findings to SDLC stages....42
6.2 Limitations....43
6.3 Ethical Considerations....43
6.4 Impact on the Society....43
6.4.1 Sustainability of the Work....44
6.5 Future Plan....44
Bibliography....45

List of Figures

1. Fig. 1.1. DevOps CSFs model in Azad et al. paper.....12
2. Fig. 1.2. DevOps Maturity framework demonstration chart in Teixeira et al. paper..... 12
3. Fig. 1.3.1 Breakdown of CPTM model for DevSecOps in Zhao et al. paper..... 13
4. Fig. 1.3.2 Breakdown of CPTM model for DevSecOps in Zhao et al. paper..... 14
5. Fig. 1.3.3 Breakdown of CPTM model for DevSecOps in Zhao et al. paper..... 15
6. Fig. 1.3.4 Breakdown of CPTM model for DevSecOps in Zhao et al. paper..... 15
7. Fig. 1.3.5 Breakdown of CPTM model for DevSecOps in Zhao et al. paper..... 16
8. Fig. 1.3.6 Breakdown of CPTM model for DevSecOps in Zhao et al. paper..... 16
9. Fig. 1.3.7 Breakdown of CPTM model for DevSecOps in Zhao et al. paper..... 17
10. Fig. 1.3.8 Breakdown of CPTM model for DevSecOps in Zhao et al. paper..... 17
11. Fig. 1.4. CSFs for DevSecOps in de Kock et al. paper..... 18
12. Fig. 5.1. Refined CSF model for DevOps..... 39
13. Fig. 5.2. DevOps Maturity Assessment Model..... 40
14. Fig. 5.3. Security in DevOps Evaluation Model.....41

List of Tables

1. Table 3.1: Structure and Methods of each Questionnaire..... 27
2. Table 3.2: Sample Demography Profile..... 29
3. Table 3.3: Data Types and Analytical Methods.....30
4. Table 4.1: List of DevOps CSFs and categories.....32
5. Table 4.2: Summary of Regression Diagnostics..... 33
6. Table 4.3: Individual CSFs and their Ranks..... 34
7. Table 4.4: Strong Pearson Correlations Among Capability Domains..... 35
8. Table 4.5: Cluster Centroids..... 36
9. Table 4.6: Strongest Correlation with Build Pass Rate (FDR-Significant)..... 37
10. Table 4.7: Qualitative Barriers to Security in DevOps Adoption.....37
11. Table 6.1: Mapping of empirical findings to SDLC stages..... 43

1 Introduction

1.1 Background

Software Engineering practices have been rapidly evolving in recent years, reshaping how digital products are conceived, built, deployed, and maintained. Traditional methodologies have been in use for decades, including the waterfall model and Agile. However, they suffered from lengthy development cycles and often overlooked operational issues. DevOps (Development and Operations) bridges this gap, aligning development and operations for faster, more reliable deployment. Today, DevOps is widely seen as the natural evolution of Agile, extending its principles to the entire software lifecycle. This combination of development and operations has been revolutionary since it fosters improved collaboration among teams and enhances the software delivery lifecycle of an organization. In developing countries like Bangladesh, the IT sector is expanding nationwide, and with it, the significance of DevOps is being amplified. Local software companies here comprise small-to-medium-sized teams and take on diverse projects, and face time constraints and pressure to deliver high-quality products. Teams here frequently encounter challenges related to communication, limited automation, and inconsistent development. For such cases, DevOps provides a promising path for improving productivity, coordination, and strengthening the overall reliability of software systems.

The adoption of DevOps in Bangladesh's software industry alone is an unsatisfactory indication of better improvement, as most companies still struggle to implement it completely. Evidence from the empirical investigations in this work shows that cultural and infrastructural differences significantly shape how DevOps principles are implemented. For instance, the studies on DevOps and the global models and frameworks employed for them demonstrated that although most companies follow DevOps practices, the global frameworks and models are poorly implemented due to cultural and automation differences, and refinement is required for local companies. Furthermore, integration of security into DevOps became vital with increasing software systems; the term DevSecOps has been coined for this purpose and gained traction in Bangladesh, too. Security is a shared responsibility within DevSecOps and embedded throughout the lifecycle. Yet, findings from local companies suggest that security practices often lag, with teams facing constraints such as limited training, a lack of automated tools, and an insufficient awareness of threat modeling.

Given these factors, a unified perspective about critical success factors, maturity, and security within DevOps is necessary to understand its definitive role in Bangladesh's software companies and software development lifecycle (SDLC).

1.2 Problem Statement

Although DevOps has become a globally recognized standard for modern software development, its implementation varies significantly across emerging economies. In Bangladesh, where the software industry is still growing, teams often adopt DevOps practices but in fragmented ways, resulting in inconsistencies between expected and actual outcomes. The absence of appropriate context-specific research limits the ability of organizations to identify which DevOps practices are most impactful, which maturity stages they currently fall under, and how security can be integrated into their workflows. Existing global models, such as DevOps CSFs[1], maturity frameworks[21], and DevSecOps guidelines[14][13], were primarily developed for countries with a high level of process automation and well-established engineering cultures. Applying these frameworks in Bangladesh can lead to inaccurate assessments and ineffective implementation strategies. These are due to a lack of research into the appropriate frameworks and models for local companies, and without an integrated perspective, it becomes difficult to understand how DevOps influences each stage in SDLC in Bangladesh, what conditions promote successful adoption, and how organizations can progress toward more mature and secure practices. The models and frameworks that have been examined in this study are in Figures 1.1, 1.2, 1.3.1, 1.3.2, 1.3.3, 1.3.4, 1.3.5, 1.3.6, 1.3.7, and 1.4.

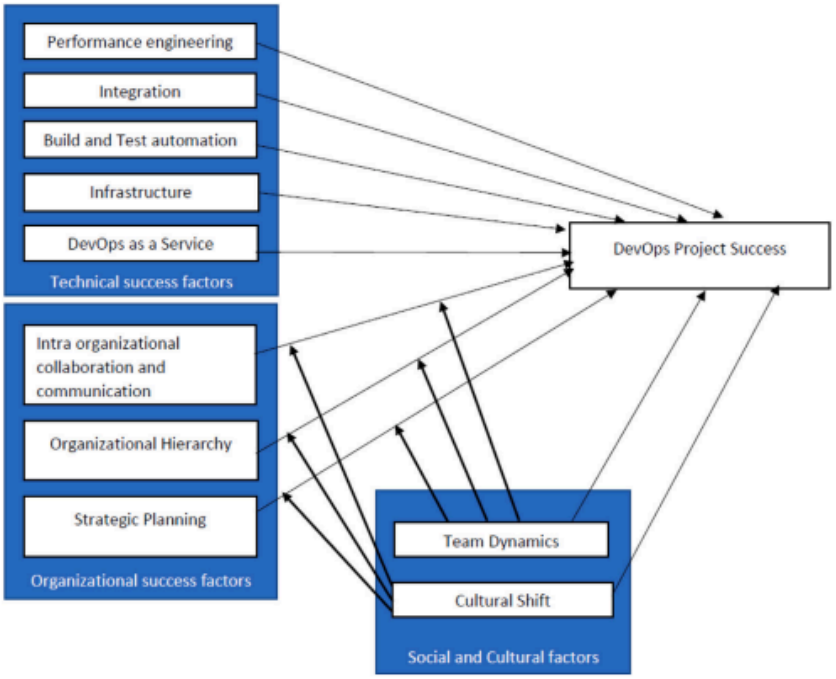

Fig. 1.1. DevOps CSFs model in Azad et al. paper

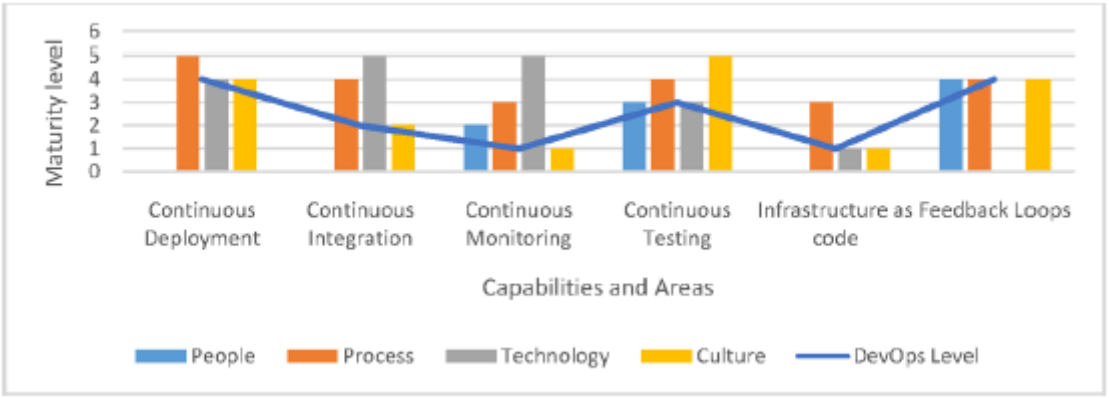

Fig. 1.2. DevOps Maturity framework demonstration chart in Teixeira et al. paper

Fig. 1.3.1 Breakdown of CPTM model for DevSecOps in Zhao et al. paper

Fig. 1.3.2 Breakdown of CPTM model for DevSecOps in Zhao et al. paper

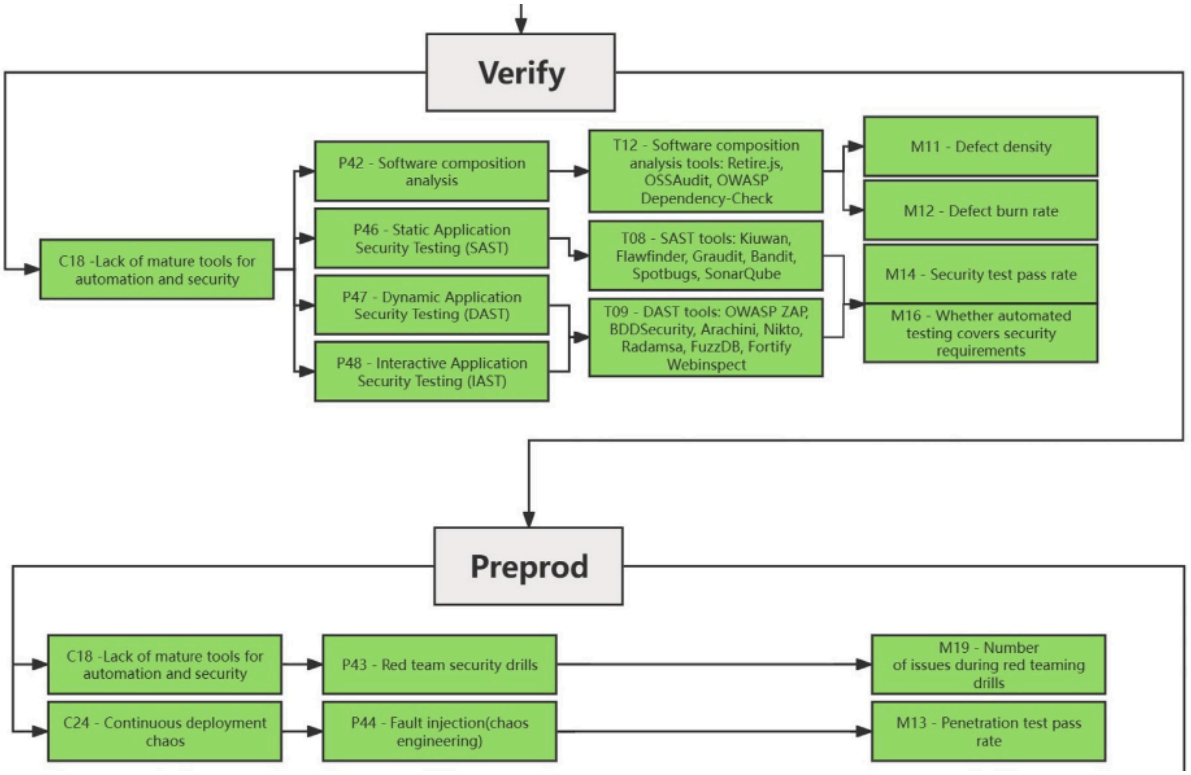

Fig. 1.3.3 Breakdown of CPTM model for DevSecOps in Zhao et al. paper

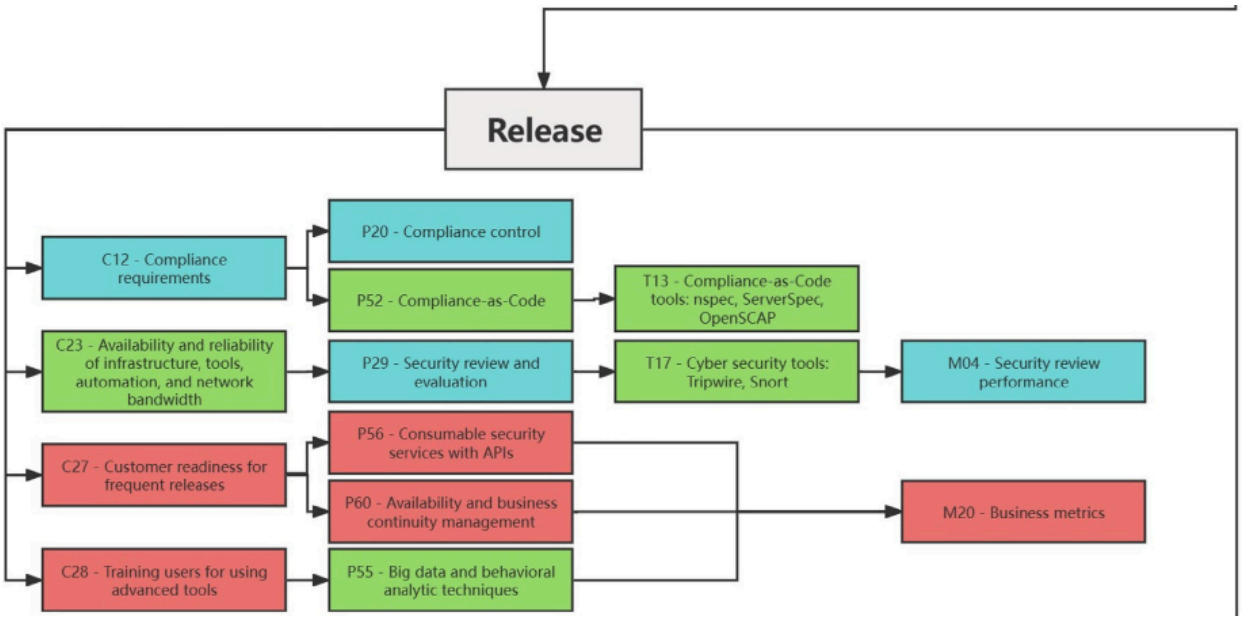

Fig. 1.3.4 Breakdown of CPTM model for DevSecOps in Zhao et al. paper

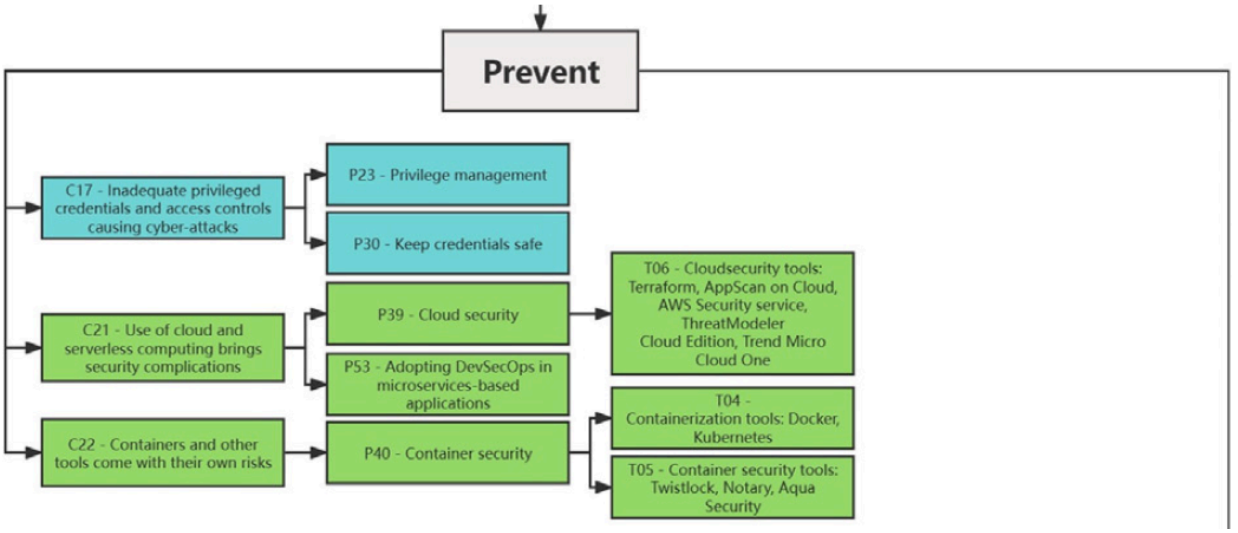

Fig. 1.3.5 Breakdown of CPTM model for DevSecOps in Zhao et al. paper

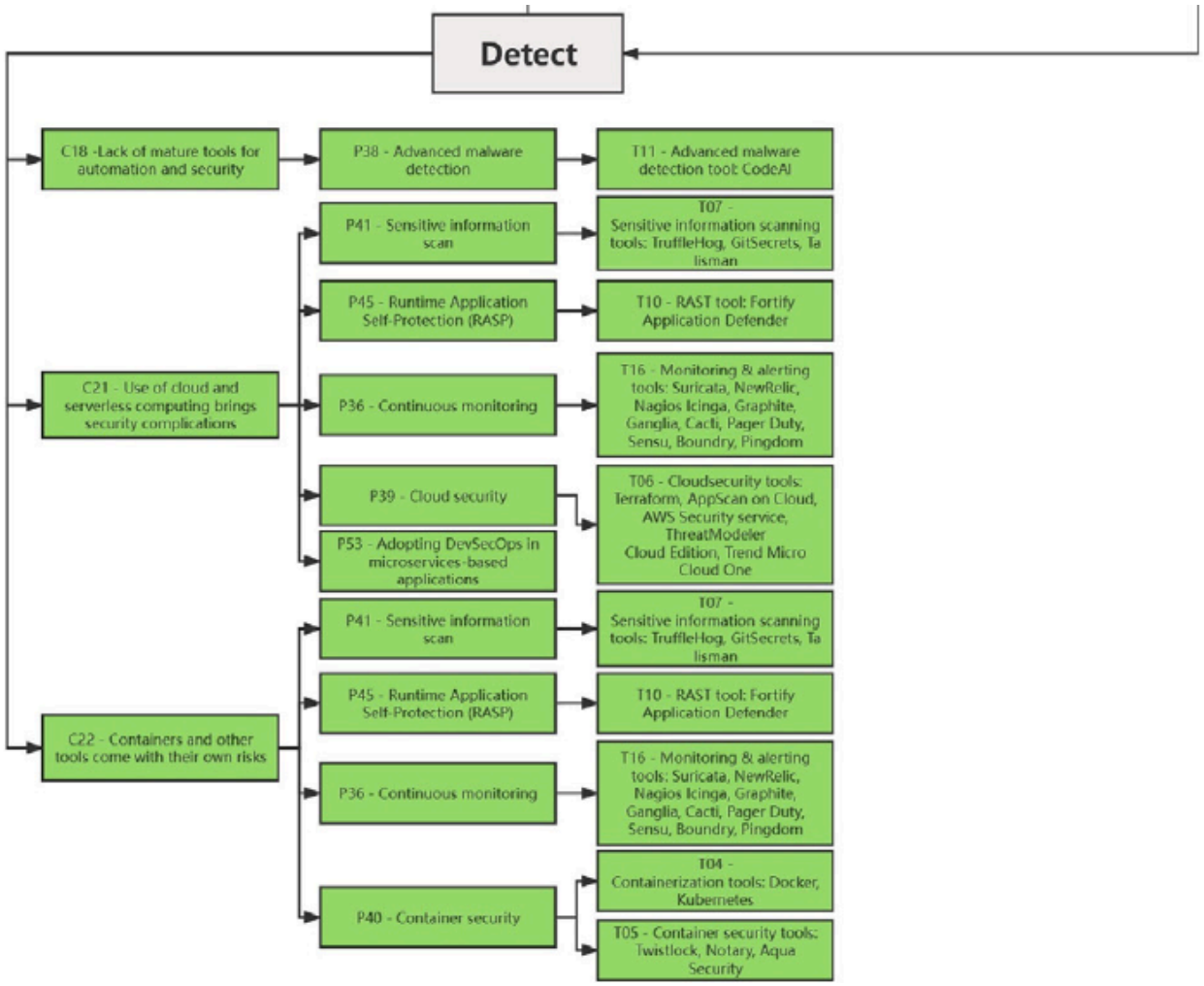

Fig. 1.3.6 Breakdown of CPTM model for DevSecOps in Zhao et al. paper

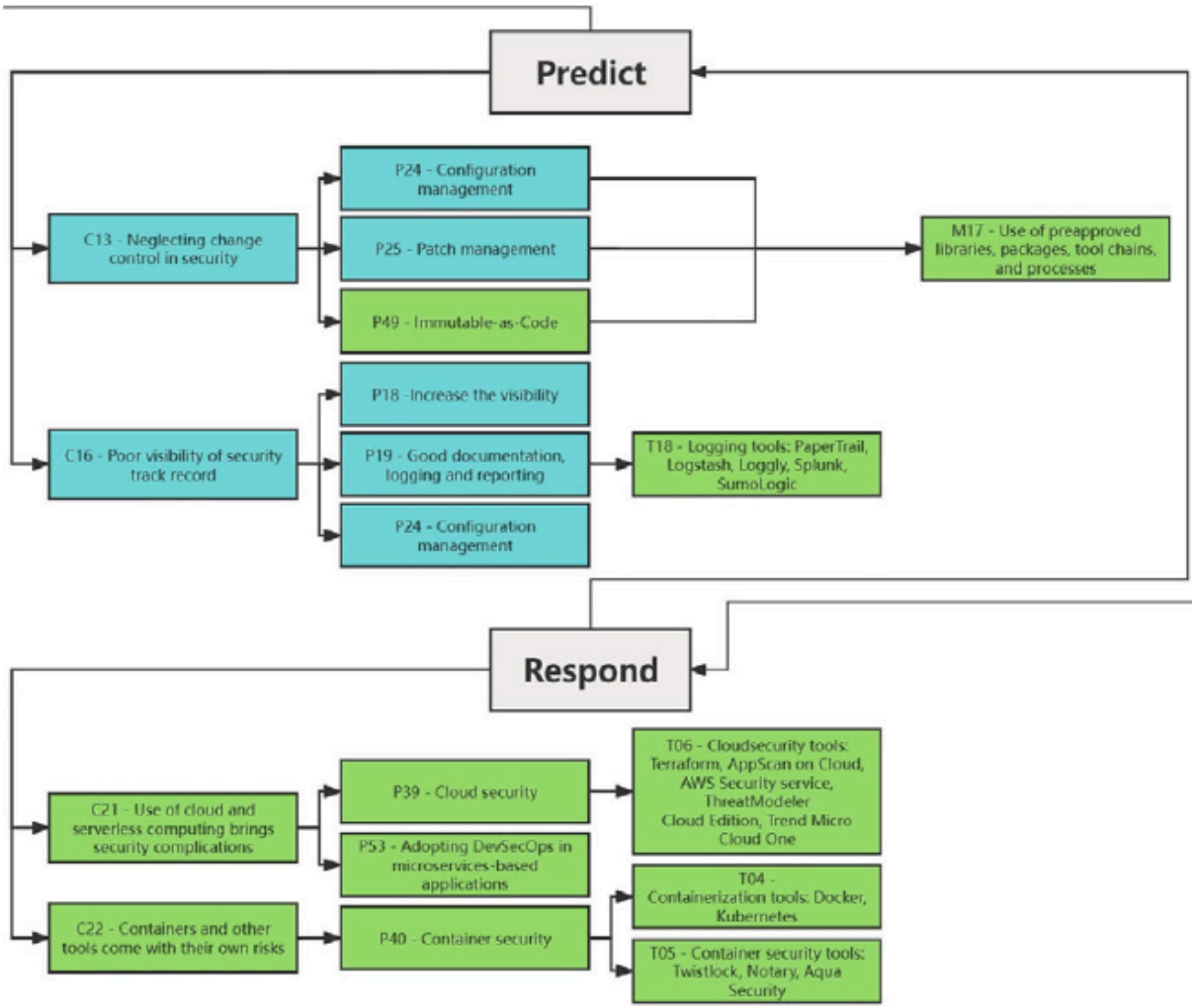

Fig. 1.3.7 Breakdown of CPTM model for DevSecOps in Zhao et al. paper

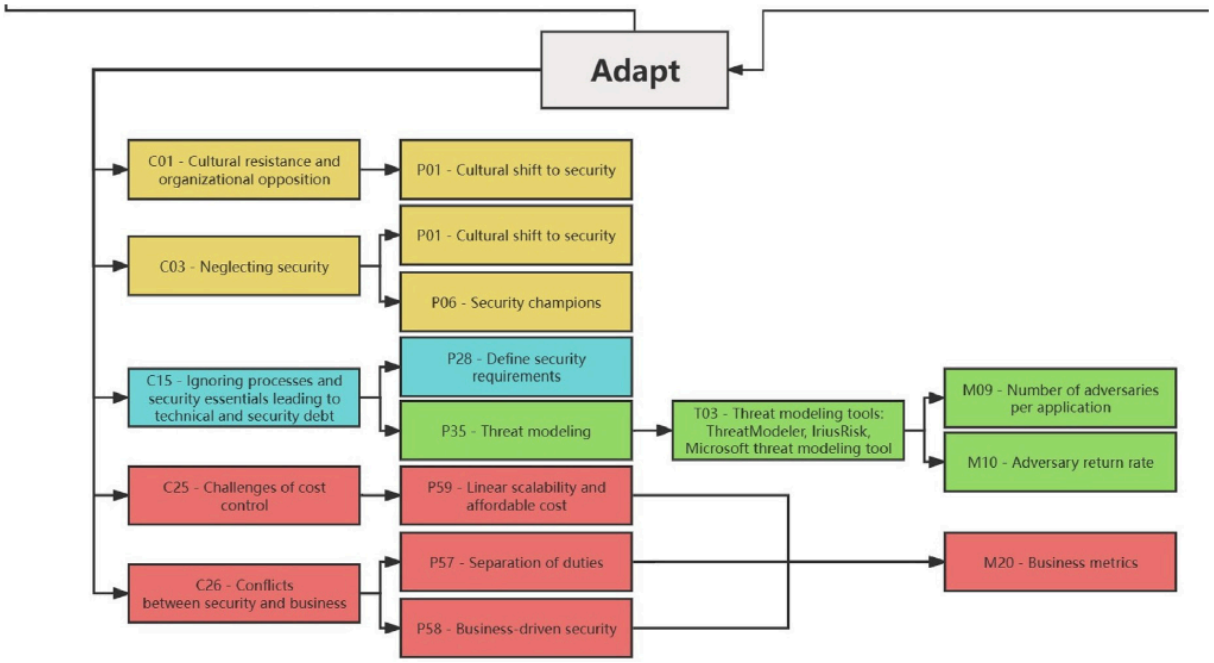

Fig. 1.3.8 Breakdown of CPTM model for DevSecOps in Zhao et al. paper

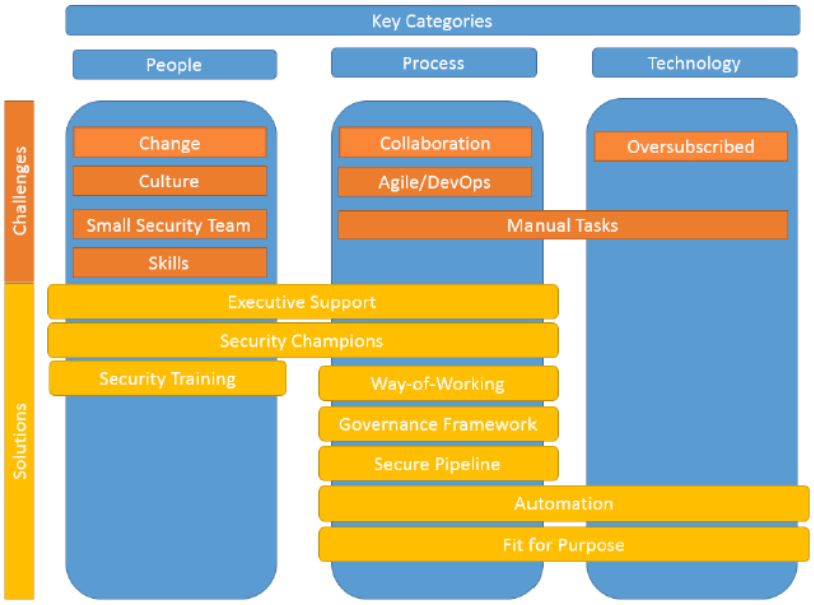

Fig. 1.4. CSFs for DevSecOps in de Kock et al. paper

The work here is exploratory and will identify preliminary associations and context-specific signals about DevOps practice in Bangladesh.

1.3 Outcome of the project

This project presents a comprehensive and unified analysis of DevOps and its role across the software development lifecycle in Bangladesh. We drew our empirical findings from Critical Success Factors (CSFs), Maturity Assessment, and Security Integration analysis surrounding DevOps, and these elements have been synthesized into a single purpose: to understand how DevOps influences the planning, development, testing, deployment, operations, and security stages within Bangladesh's software industry.

Through this approach, the project contributes to:

- An understanding of DevOps practices in Bangladesh, through CSFs analysis starting from organizational, cultural, to technical dimensions.
- A contextually refined interpretation of maturity in DevOps
- A perspective on Security in DevOps (DevSecOps) integration
- Conceptual models that illustrate how DevOps interacts with the different stages of the SDLC within local organizations.

This work ultimately provides insights for industry professionals and researchers seeking to improve DevOps adoption, enhance the maturity of processes, and integrate security for efficient software development practices in Bangladesh.

2 Literature Review

2.1 Literature Selection Procedure

The literature selection process in this study followed a pre-planned systematic approach to include credible and relevant works related to DevOps, DevOps Maturity, and DevSecOps practices. The priority was given to peer-reviewed papers and studies published within the last decade. The literature was screened based on the scope of this project and methodological relevance.

2.2 Keywords Searched

A set of targeted combinations of keywords was searched to capture the breadth of DevOps research. The primary keywords used include: “DevOps”, “DevOps Adoption”, “DevOps Critical Success Factors”, “DevOps Maturity Model”, “Software Development Lifecycle AND DevOps”, “DevOps in Developing Countries”, “DevOps Security”, “DevSecOps”, and “DevSecOps Practices”.

2.3 Databases for Search

The sources were collected from a variety of academic databases. The primary databases included: Google Scholar, ResearchGate, ScienceDirect, and IEEE Xplore.

2.4 Existing methods, frameworks, and models

2.4.1 DevOps Critical Success Factors (CSFs)

Several studies have explored the need for identification and prioritization of Critical Success Factors (CSFs) for DevOps. One systematic literature review, for instance [1], examined 38 relevant papers and identified 10 CSFs in three broad categories: technical, organizational, and social/cultural, synthesizing a framework for the CSFs that led to DevOps project success. Then, the same author attempted to validate the CSFs of DevOps projects found in the SLR as reported by professionals and collected qualitative data by surveying 15 professionals experienced in DevOps with an open-ended survey questionnaire. The existing 10 CSF model was validated by the respondents of the survey through open-ended, qualitative answers[2]. Likewise, a study conducted a systematic literature review as well as interviews, proposing a

conceptual model that groups DevOps CSFs into four areas: collaborative culture, DevOps practices, proficient DevOps team, and metrics measurement[62]. In a different approach[20], a paper developed a taxonomy of DevOps success factors using the CAMS principles framework. The research involved a 4-phase methodology including literature review, factor mapping, empirical verification, and ranking using the PROMETHEE II method. The study identified 16 success factors, mapped their logical relationships, and created a holistic taxonomy. Another paper[17] attempted to find the CSFs of continuous integration, continuous delivery and continuous deployment processes (CI/CD- processes) and conducted 3-phase research which included a systematic literature review to find potential CSFs, classification of the potential CSFs and a multiple case study to validate, recognize and understand the importance of the CSFs in relation to the success of CI/CD- processes. Through a systematic literature review, another paper identified critical success factors for continuous practices and grouped them to create an initial framework. This led to confirmation and classifications of some factors that have an impact, namely, coordination, collaboration challenges, and organizational structure[16]. Another study[19] focused on CSFs for DevOps, including automation tools, organizational support, and collaborative culture. The integration of security (DevSecOps) was highlighted here as challenging due to poor collaboration and delayed security measures. In addition to these CSF-focused studies, one paper took a SWOT analysis approach coupled with an Analytic Hierarchy Process (AHP) to comprehensively explore DevOps implementation and provide a strategic framework to guide organizations. The factors were categorized into strengths (organizational culture, collaborative tools, automation capabilities), weaknesses (resistance to change, skill gaps), opportunities (market trends, technological advancements), and threats (cybersecurity risks, integration challenges)[4]. Another study shifted focus to human dimensions influencing DevOps adoption[18]. Through systematic mapping, they identified 59 distinct human factors, such as communication, leadership, teamwork, and organizational culture, and proposed a theoretical framework to explain their interrelationships and influence on DevOps success.

2.4.2 DevOps Capability Maturity Model

The research on DevOps maturity expanded rapidly in the last decade, and authors reviewed existing models and also developed conceptual maturity models. One paper proposed a structured DevOps

maturity model grounded in SLR and interviews from experts; the model mapped core capabilities and areas onto CMMI-like maturity levels and was validated qualitatively with professional guidance[21]. Another paper carried this agenda forward by systematically surveying and benchmarking available DevOps maturity models, demonstrating that most proposals reused CMMI-style layerings but lacked broad empirical validation[23]. A comparative work by another study[29] highlighted that different models often emphasize distinct dimensions and that maturity labels are not directly comparable. In a paper[37], the SLR catalogued fifteen DevOps maturity models and found that only a minority had undergone empirical validation. A historical survey situated DevOps maturity models within the wider lineage of process improvement models (Shewhart, Nolan, CMMI), arguing that modern practice favors more adaptable, domain-specific forms rather than one canonical framework[24]. A paper further stated the competence and maturity model for software development organizations, emphasizing interdependencies among capabilities and providing an assessment artifact validated for limited organizations[35]. A second body of work provided empirical grounding for DevOps practices and maturity claims. A study analyzed the HELENA2 dataset (n > 1000) to quantify relationships between practice adoption frequency and maturity indicators: automation, collaboration, and continuous feedback correlated strongly with advanced maturity[31]. Another paper developed the Lean-Agile DevOps Maturity Framework and validated it using telemetry from three large enterprises; the study linked higher maturity scores with improved DORA-style outcomes such as deployment frequency and MTTR reductions[42]. The study[33] combined multiorganization case studies and simulation to show performance gains from mature DevOps adoption but acknowledged limited external validity from small sample sizes. Notably, several papers reported that high maturity does not automatically guarantee improved business performance; a study[38] emphasized that maturity models must be complemented by organizational changes, governance, and strategic alignment to deliver measurable benefits. Another study introduced a GQM-based survey to structure DevOps assessment through goals, questions, and metrics and to visualize results through radar plots[39]. Then, a paper[40] proposed a GMQ metrics model for DevOps that operationalized 11 metrics and validated them in enterprise case studies. A paper performed a systematic mapping of DevOps-CMMI integration efforts and found that while conceptual compatibility exists, practical integration mechanisms and operationalized measurement approaches are still scarce[41]. Another paper[25] reviewed methods used in maturity model development, noting the dominance of custom, pragmatic approaches and the increasing adoption of

Design Science Research, but also a lack of methodological standardization across studies. Several papers developed focused maturity models for particular capabilities or needs. The study[28] presented SPIM-TA, a five-level maturity framework specifically for automated testing in global software development, validated in two case studies. A paper proposed an integrated maturity model that explicitly included security and AI capabilities on top of a DevOps foundation, arguing for open, extendable architectures that can evolve as new domains[26]. Multiple studies showed that maturity is not purely technical. A paper[30] examined the transition from Agile to DevOps through a qualitative field study and argued that soft skills, role evolution, and “organizational smartness” are central to maturity advancement. Another study applied a diagnostic questionnaire in public organizations and highlighted cultural knowledge and structural barriers as primary constraints to progress[36]. A study surveyed industrial practices for performance engineering within DevOps and showed that despite tool availability, performance practices were infrequently integrated, revealing organizational and skills barriers[34].

2.4.3 Security in DevOps (DevSecOps)

Each academic paper primarily focused on defining DevSecOps and outlining the dimensions. The paper[14] identified definitions, challenges, practices, tools, and metrics through a multivocal literature review, culminating in the CPTM (Challenge-Practice-Tool-Metric) framework, which conceptualized DevSecOps as a lifestyle model. Similarly, a study[13] emphasized organizational, cultural, and technological factors, identifying critical success factors (CSFs) under people, process, and technology dimensions. Expanding this, a paper[43] introduced a hybrid methodology combining interpretive structure modeling and fuzzy TOPSIS to identify and prioritize DevSecOps challenges. The findings showed dominance of standards and automation gaps as central obstacles. Complementarily, another paper proposed FOBICS, a metric framework for qualifying DevSecOps performance, bridging the gap between qualitative models and measurable outcomes. Beyond technical integration, several studies showed that social and cultural factors underpin successful adoption. A paper characterized DevSecOps as a cultural transformation rather than a purely technical shift, emphasizing collaboration, knowledge sharing, and inclusion[60]. However, they noted insufficient empirical studies exploring how such cultural shifts occur. Another paper[58] showed similar findings through an SLR, identifying 21 challenges in adoption, chiefly human, process, and tool-related, and revealing the paucity of developer-centered automation tools. Then, a paper further conceptualized an

integrated adoption framework emphasizing cultural alignment, strategic leadership, and iterative feedback loops[51]. Similarly, a study[56] explored automation of compliance with security standards (IEC 62443-4-1), confirming that up to 60% of activities could be automated but acknowledging persistent gaps in generalizability. Contemporary research began integrating advanced technologies to enhance security automation. A study reviewed AI-driven security methods in DevSecOps, categorizing approaches such as anomaly detection, NLP-based policy enforcement, and explainable AI. While AI promised scalability, the authors noted the absence of standardized benchmarks and real-world validations[45]. Another paper extended this line of inquiry by merging LLMs with Security Chaos Engineering to automate attack defence tree generation, demonstrating improved early threat detection[44]. Yet, both studies revealed that empirical validation and deployment scalability remain pressing issues. Two papers[53][61] reviewed automation and continuous security controls, reinforcing the necessity of “shift-left” security practices and emphasizing the need for sociotechnical alignment. Another paper categorized compliance in DevSecOps into initiation, management, and technical enforcement, revealing automation and metric development as underexplored areas[59]. Similarly, a paper[49] proposed a generalized DevSecOps pipeline integrating continuous testing and security validation, validated through a real-world application. It demonstrated the practical benefits of continuous testing but also acknowledged tool limitations and the need for broader validation. Complementing these quantitative advances, two studies’ models, the FOBICS model[46] and the conceptual model[50], provided structured measurement frameworks emphasizing automation, collaboration, and monitoring. Another paper examined secure SDLC practices in Bangladesh IT firms[52], identifying major barriers such as resource limitations, inconsistent tool use, and lack of awareness. They proposed cost-effective automation strategies, showing the disparity between global best practices and local implementation capacity. Likewise, a paper[47] analyzed DevSecOps in SMEs using mixed methods, combining TAM and DOI models. Findings revealed that despite positive attitudes towards adoption, SMEs face substantial resource and automation barriers. Several works advanced theoretical models of secure DevOps. A paper[57] applied the founded theory to derive a hypothetical framework distinguishing functional and non-functional security concerns, addressing professional perspectives. Another paper presented a broad SLR of DevOps concepts and challenges, reaffirming the socio-technical nature of DevSecOps and the lack of unified definitions across research[54]. A study proposed the ADOC model[53], emphasizing open-source security integration in cloud environments, while a study[50] synthesized CI/CD security practices into a continuous

integration model, stressing automation and shared responsibility.

2.5 Gap Analysis

When viewed from the lens of the software development lifecycle (SDLC), there are several gaps in the literature, especially in the context of developing countries.

DevOps CSFs largely rely on qualitative reviews[1], offering limited quantitative validation. There is also a lack of multicollinearity and specific context prioritization. Literature on the DevOps maturity model is similarly fragmented[21]; most models are weakly validated and lack justification for reliability or maturity thresholds, limiting their application in environments with different organizational and automation capabilities. In DevSecOps, existing works are dominated by conceptual frameworks[14][13], with few empirically grounded analyses linking security practices to security outcomes or integrating other organizational dimensions into the measurement. Empirical studies in general remain scarce across these three domains.

These gaps indicate the absence of a context-specific, empirically validated, and unified framework that connects success factors, maturity models, and security practice evaluation. This motivates this study, which uses these three dimensions to understand the role of DevOps in SDLC, combining CSF, maturity, and security analyses in a single national context.

3. Research Methodology

3.1 Motivation and Purpose

In this project, the role of DevOps in SDLC in Bangladesh is examined by observing three major analytical dimensions:

- Identifying the Critical Success Factors (CSFs) that influence DevOps project success.
- Assessing the maturity of the DevOps capability framework across local organizations.
- Evaluating how security practices are integrated into DevOps workflows.

Each of these dimensions has been studied individually in prior global studies, and their combined research within a single country's context is limited. Since these dimensions are related yet analytically distinct, a multidimensional empirical design is justified and will be essential in the future.

3.2 Research Approach

This study adopted an exploratory and empirical approach, combining quantitative study data with both qualitative and quantitative methods for each dimension: Critical Success Factors (CSFs), Maturity model, and Security frameworks. This approach is currently appropriate as research on DevOps adoption and implementation in Bangladesh is underexplored, and existing international frameworks and models[1][21][14][13] have not been jointly evaluated in a single national context. Rigid parametric assumptions may not hold in small sample studies; hence, exploratory designs allow studies to proceed without normally distributed responses or variances[12]. Three separate questionnaires for each dimension were prepared using 10-point Likert scale questions for the questions that would be later used as items to do analyses on for each respective dimension. These Likert scale items were used for quantitative purposes. Table 3.1 summarizes the structure.

Purpose	**Source**	**Items**	**Scale**	**Analytical Use**
DevOps CSFs Questionnaire	Azad et al. (Critical Success Factors model)	19 items	10-point Likert	Multiple Linear Regression (MLR) to identify high-impact CSFs

DevOps Maturity Questionnaire	Teixeira et al. (Maturity Model)	24 items	10-point Likert	Reliability (Cronbach α), Pearson Correlation, and K-means clustering
Security in DevOps Questionnaire	Zhao et al. (CPTM framework) + de Kock et al. (security CSFs)	18 + 3 security outcome items	10-point Likert + ordinary outcome metrics	Spearman Correlation, permutation testing, bootstrapping, and FDR

Table 3.1: Structure and Methods of each Questionnaire

In an attempt to integrate qualitative measures, open-ended responses from the security practices survey were coded thematically to identify contextual barriers, organizational enablers, and practitioner perspectives.

As shown in Table 3.1, each questionnaire was prepared using established models and frameworks to use them as measurable components for analyses:

- CSFs questionnaire was based on Azad et al's model[1], 19 CSFs across organizational, cultural, and technical factors were included.
- The maturity model questionnaire was based on Teixiera et al's maturity model[21], using their six capability domains and four areas, and arranged them into 24 Likert items per domain.
- Security in DevOps (DevSecOps) questionnaire was based on the CPTM framework[14] and the security CSFs model[13], adapting 18 practices and enablers from them, plus three security outcome metrics.

All the Likert items were pilot tested for clarity and adjusted where necessary.

3.3 Sampling and Participants

A convenience sampling strategy was used for three of the dimensions' purposes, as experts from the software industry were specifically approached, which resulted in a limited number of samples.

Participants were recruited through professional networks and company contacts. Study-wise sample overview is as follows:

- CSFs analysis: 23 DevOps professionals.
- Maturity model assessment: 31 professionals, including DevOps engineers, release managers, and project managers.
- Security practices evaluation: 25 professionals, including DevOps engineers, SREs, Developers, and Media Op Engineer.

Table 3.2 shows the demographic characteristics in each of the questionnaires.

Demographic Variable	**Categories**	**CSFs (N=23)**	**Maturity (N=31)**	**Security (N=25)**
Job Title	DevOps Engineer	60.87 %	61.29 %	44.00 %
	Software Engineer	26.09 %	19.35 %	16.00 %
	Cloud Engineer	4.35 %	3.23 %	12.00 %
	DevOps Architect	4.35 %	3.23 %	12.00 %
	Team Lead	4.35 %	3.23 %	4.00 %
	SRE	-	3.23 %	4.00 %
	Media Operations Engineer	-	3.23 %	8.00 %
	Release Manager	-	3.23 %	-
Experience	< 1 year	21.74 %	12.90 %	24.00 %
	1 - 3 years	34.78 %	38.71 %	28.00 %
	4 - 6 years	26.09 %	29.03 %	28.00 %
	7 - 10 years	13.04 %	16.13 %	16.00 %
	> 10 years	4.35 %	3.23 %	4.00 %

Company Size	10 - 50	8.70 %	12.90 %	28.00 %
	51 - 200	17.39 %	35.48 %	32.00 %
	201 - 500	17.39 %	12.90 %	4.00 %
	500+	56.52 %	38.71 %	36.00 %

Table 3.2: Sample Demography Profile

Most participants have 1-3 years of experience, and the majority of them in each questionnaire are from large firms (500+ employees). Small-Medium-Large software firm employees participated, which helped in determining the overall outcome of this study. The sample sizes are appropriate and sufficient for the exploratory analysis of frameworks and model validation of this study.

3.4 Data Collection

The data were collected via Google Form, and confidentiality and voluntary participation were explicitly stated.

- CSFs: 5 sections, the first section for demographic purposes, and the rest were 19 CSF items. Responses were collected over two weeks.
- Maturity: 6 capability domains and 4 area sections with 24 items per domain. Responses were collected over three weeks.
- Security: 18 CPTM and CSFs practices along with 3 security outcomes, and an open-ended section. Responses were collected over three weeks.

Expert professionals in the DevOps field participated voluntarily.

3.5 Data Analysis

The responses for CSFs analysis, Maturity model assessment, and Security frameworks evaluation were analyzed in Microsoft Excel's Analysis ToolPak and Python. Table 3.2 shows the methods applied to the collected data.

Data Type	Description	Questionnaire	Analytical Method
Quantitative (Likert)	0 - 10 range	All	MLR, Pearson Correlation, K-means clustering, Cronbach α
Outcomes	Build Pass Rate (%), Time to Fix, and Incident Frequency	Security in DevOps	Spearman Correlation, permutation tests
Qualitative text	Open-ended responses to security barriers	Security in DevOps	Manual thematic coding

Table 3.3: Data Types and Analytical Methods

This type of combination is justified, according to Wasserman[6], for non-parametric methods.

3.5.1 Critical Success Factors Analysis

The analysis used Multiple Linear Regression (MLR) as the statistical model since, in exploratory conditions such as this, MLR is robust to slight violations. DevOps project success (Y – the dependent variable) was regressed against 19 identified CSFs (X – the independent variables where k =19). One of the questions - “Will increased process automation contribute to DevOps project success?” - was defined as the dependent variable (Y) based on perceived project success through automation. This MLR model is fitted using matrices, the coefficients/parameters are then estimated using the method of least squares, and their properties (i.e., R^2, adjusted R^2, and Variance inflation factors) are calculated. Variance inflation Factors (ViF) were computed to assess multicollinearity; the maximum ViF among the predictors was 6.01, which is acceptable multicollinearity. Finally, inferences are made regarding the importance of each regressor using two-tailed t-tests with exploratory $\alpha = 0.20$. This threshold ($\alpha = 0.20$) detects preliminary signals in low-sample conditions.

3.5.2 Maturity Model Assessment

Multiple analytical process was applied following a sequence. Basic descriptive statistics were applied to find capability-level maturity, followed by Cronbach’s α in order to evaluate the internal consistency of each capability scale. Then, Pearson Correlation was calculated to analyze inter-capability relationships. K-means clustering (K=3) was employed with normalized capability scores to detect maturity segments, and lastly, CMMI-Level Mapping was done based on centroid

profiles and threshold rules.

3.5.3 Security Frameworks Evaluation

Since the sample size here was small, non-parametric methods were the best fit. Spearman's rank correlation was done for association analysis between the 18 security practices derived from the CPTM framework and the security CSFs model (Q8-Q25) and security outcomes (Q26-Q28). To maintain robustness with N=25, the following were done:

- Permutation tests (10,000 iterations) to produce distribution-free p-values
- Bootstrap confidence intervals (10,000 resamples)
- False Discovery Rate (FDR) correction for multiple hypotheses
- Thematic coding of open-ended responses into categories such as cultural alignment, time and resource constraints, and cross-functional skill development

Bootstrapping[27] repeatedly resamples with replacements to generate confidence intervals and estimate variability in Spearman Correlation. Permutation testing[32] avoids distributional assumptions and computes p-values for Spearman Correlations. Benjamini-Hochberg False Discovery Rate (FDR) controls type-I error, as there are 54 hypotheses here. As recommended by Braun et al.[15], The open-ended responses were manually thematically coded.

3.6 Limitations of the Methodology

While effective for exploratory-based work, the methodological design faced some limitations. Small sample sizes limit statistical generalizability, and a cross-sectional design prevents causal inference. There was an overfitting risk in CSFs regression, low reliability values in the maturity model, and zero-inflated incident data, which limited security outcome conclusiveness. These generally limit a study’s rigor; however, given the small sample sizes and other constraints, results should be considered exploratory and indicative, not conclusive.

4. Findings and Result Analysis

The findings of this study involve the analysis of three empirical components: DevOps Critical Success Factors (CSFs), DevOps Maturity framework assessment, and evaluation of DevOps security practices. A unified understanding of how DevOps functions around SDLC in Bangladesh's organizations can be synthesized from these analyses.

4.1 DevOps Critical Success Factors

The analysis here is based on the Multiple Linear Regression (MLR) applied to CSF items. The MLR model was fitted using 23 questionnaire responses and the 19 CSFs items. Coefficient estimates and standardized effect sizes were calculated to rank the CSFs by importance.

The 19 CSFs were categorized into the original three groups[1]: Organizational, Social and Cultural, and Technical factors. Table 4.1 lists the CSFs per category.

Category	**Variables with Descriptions**
Organizational Success Factors	X_1- Frequency of DevOps Events; X_2-Frequency of Meetings; X_3-Clearly defined roles and cultural alignment; X_4-Leadership roles
Social and Cultural Success Factors	X_5-Team interaction; X_6-Sharing of ideas, feedback and continuous learning; X_7- Cultural inclusiveness and diversity; X_8-Conflict resolution
Technical Success Factors	X_9-Adoption of performance engineering practices; X_{10}-Adoption of lightweight performance engineering practices; X_{11}-Adoption of complex but thorough performance engineering practices; X_{12}-Smooth integration of performance engineering practices with DevOps tools; X_{13}-Mitigating challenges during CI/CD pipeline; X_{14}- Increased build automation; X_{15}-Increased test automation; X_{16}-Importance of infrastructure for software process integration; X_{17}-Importance of infrastructure in continuous software development and deployment; X_{18}-Increased infrastructure provisioning; X_{19}-Importance of microservices

Table 4.1: List of DevOps CSFs and categories

The regression model included all 19 predictors (X_1-X_{19}). Two-tailed t-tests were run on the calculated individual coefficients to confirm their significance with the null hypothesis (H0: $\beta i = 0$), and the alternative hypothesis (H1: $\beta i \neq 0$). The hypothesis testing showed that only five factors, X_1, X_6, X_8, X_{10}, and X_{12}, were statistically significant at an exploratory α=0.20 threshold. The rest of the calculated coefficients were not significant because of the failure to reject $\beta i = 0$ in the tests. Table 4.2 shows the regression diagnostics, including SSE, SSR, R^2, adjusted R^2, and F-statistics.

Statistic	**Value**
SSE	1.470
SSR	31.960
R^2	0.956
Adjusted R^2	0.678
MSE (s^2)	0.490
F-value	3.433
p-value	0.169

Table 4.2: Summary of Regression Diagnostics

The high R^2 (0.956) indicates strong exploratory power, though the adjusted R^2 (0.678) suggests potential overfitting risk. X_6 (“Sharing of ideas, feedback and continuous learning”) had the strongest standardized effect, and only one factor (X_8) was significant at α=0.05; however, five were significant at exploratory α=0.20. This suggests that most CSFs exert weak effects due to multicollinearity, sample size, and varied organizational maturity. From the standardized coefficients ($|\beta_i^*|$), the independent variables can be clustered into categories according to the CSF model by Azad[1], and their average standardized coefficients can be calculated and ranked according to their importance to DevOps project success, shown in Table 4.3.

CSF	Average β_i^*	Rank
Performance Engineering	0.400	8
Integration	0.516	4
Build and Test Automation	0.480	6
Infrastructure	0.536	3
DevOps as a Service	0.035	10
Intra-organizational Collaboration and Communication	1.051	1
Organizational Hierarchy	0.240	9
Strategic Planning	0.401	7
Team Dynamics	0.778	2
Cultural Shift	0.504	3

Table 4.3: Individual CSFs and their Ranks

4.2 DevOps Maturity Across the Industry

The maturity model for DevOps presented in[21] has been assessed here using capability scores, reliability measures, and clustering-based segmentation. Six capabilities were analyzed: Continuous Integration (CI), Continuous Development (CD), Continuous Testing, Infrastructure as Code (IaC), Continuous Monitoring, and Feedback Loops. The mean values are the capability maturity scores for N=31. Monitoring and CD have the highest maturity scores (8.94 and 8.71), while CI and Testing show the lowest maturity (7.38 and 7.56) and reliability (α=0.43 and α=-0.24) calculated through Cronbach's alpha. Though the α for CI and Testing are very low, item-total internal relations were inspected to check whether specific items were misaligned or reversed; hence, this was treated as formative rather than reflective and retained with caution. Pearson Correlation was used to show inter-capability relationships. Table 4.4 lists the pairs with strong correlations.

Capability Pair	r	Significance
Testing - IaC	0.716	P < 0.01
CI - Testing	0.670	P < 0.01
CI - Feedback	0.622	P < 0.01
CI - CD	0.606	p< 0.01

Table 4.4: Strong Pearson Correlations Among Capability Domains

Testing - IaC has the highest correlation, showing that organizations use IaC extensively in automated testing environments. Almost all capability pairs, except IaC-Monitoring (r=0.315, p>0.05), showed positive r, which highlights the co-evolution of DevOps practices. K-means clustering (K=3) was integrated to segment maturity into three groups: the cluster centroids, which were chosen based on the lowest, middle, and highest scoring respondents per capability. The capability scores were normalized,

$$x^{'} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where X is capability. Normalization is required as Euclidean distance depends on scale[22]. Then, Squared Euclidean distances were computed for each respondent per centroid,

$$D^2(X, C) = \sum_{i=1}^{6} ((x_i - c_i)^2$$

where C is the centroid. Afterwards, cluster labels were assigned using the minimum distance rule. The centroids were finally updated, pitting cluster labels against each capability, and they are shown in Table 4.5.

Capability	Cluster 1 (Developing)	Cluster 2 (Maturing)	Cluster 3 (Optimizing)

CI	4.76	7.48	8.69
CD	7.68	8.32	9.60
Testing	5.5	7.28	8.92
IaC	6.11	7.52	8.98
Monitoring	8.14	8.45	9.77
Feedback Loop	6.66	7.67	8.78

Table 4.5: Cluster Centroids

The clustered centroids were then mapped to Capability Maturity Model Integration (CMMI) style maturity levels in order to interpret these outcomes with recognized maturity levels. Most are in mid-to-high maturity levels (3-5).

4.3 Security Integrations in DevOps

The CPTM framework[14] and security CSFs[13] were evaluated here through the analyses. Since the responses amount to 25, the analysis relied on non-parametric techniques such as Spearman Rank Correlations, 10,000 iterations permutation tests, bootstrap confidence intervals, and Benjamini-Hochberg False Discovery Rate (FDR) correction. Participants reported three security performance indicators: Build Pass Rate, Time to fix high/critical vulnerabilities, and Security Incident Frequency over the past year. The correlation between Security practices and outcomes was measured using Spearman's Rank Correlation. Table 4.6 summarises the strongest correlations.

Practice	**Description**	**p (Spearman)**	**p_perm**	**FDR_Significant**
Q16	Automation used to help decide which vulnerabilities are most urgent (CPTM)	0.755	< 0.001	Yes
Q15	Incident Response Playbooks and Exercises (CPTM)	0.741	< 0.001	Yes

Q12	Release blocked if there are open high/critical security findings (CPTM)	0.673	< 0.001	Yes
Q14	Centralized logging and security monitoring (CPTM)	0.676	0.001	Yes
Q25	Automation at Scale (Environment, Pipelines) (CSF)	0.648	< 0.001	Yes

Table 4.6: Strongest Correlation with Build Pass Rate (FDR-Significant)

Build Pass Rate emerges as the most stable and reliable indicator among the three security outcomes; these five practices in Table 4.6 represent automation-heavy and monitoring-intensive DevSecOps capabilities.

Open-ended responses were coded into themes. Table 4.7 shows the frequency distribution.

Theme	**Frequency (%)**
Time and Resource Constraints	50.0%
Cultural Alignment/Shared Responsibility	37.5%
Cross-Functional Skill Development	37.5%

Table 4.7: Qualitative Barriers to Security in DevOps Adoption

These indicate that many organizations struggle with limited staff and a lack of shared ownership between DevOps and security teams. There are also significant skill gaps that can affect the implementation of DevSecOps practices.

4.4 Patterns Observed in DevOps teams

Across the three empirical analyses, CSFs, Maturity Assessment, and Security in DevOps evaluation, several patterns can be observed.

DevOps adoption in Bangladesh's software companies heavily relies on automation. Both CSF analysis (Section 4.1) and Matuity assessment (Section 4.2) indicate that factors tied to automation play a major role in DevOps success. Similarly, Security in DevOps evaluation (Section 4.3) showed that the most statistically reliable correlations were tied to automated security practices. Organizational and Social-Cultural factors remain inconsistent, though in CSF analysis, it emerged as the strongest standardized predictor; the qualitative themes from the Security evaluation (Table 4.7) show persistent challenges with shared responsibility and communication.

5. Proposed Framework

The analyses of CSF, Maturity, and Security (CPTM+CSF) practices shaped the following models and framework for the entire study.

5.1 DevOps CSF Model

The hypothesized model in Figure 5.1 shows that the rank of each CSF, ergo its importance, is pictorially represented by the thickness of the arrow. The thickness of the arrows was determined by the ranking in Table 4.3. This refined model is a modification of the synthesized CSF model proposed by Azad [1]. It is clear that Intra-organizational collaboration and communication proved to be the most important CSF, as per calculations, hence it has the thickest arrow, whereas DevOps as a service has the thinnest arrow since it proved to be the least important CSF according to the calculations.

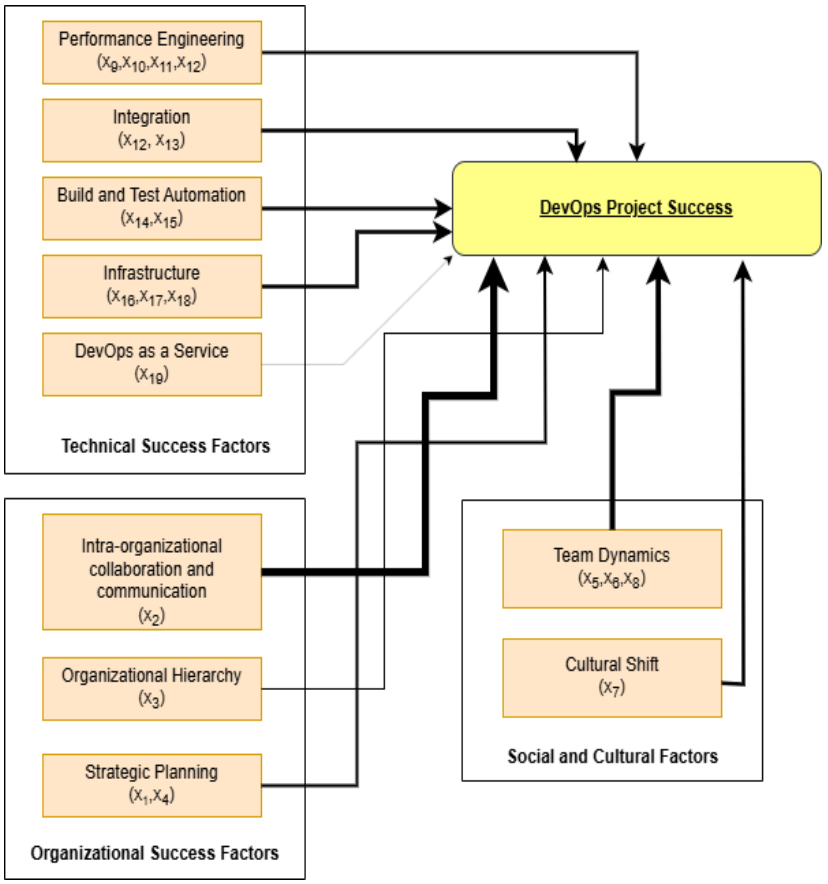

Fig. 5.1. Refined CSF model for DevOps

5.2 DevOps Maturity Assessment Model

The model retains the original six capability domains[21] and has been refined based on the empirical findings. The placements and arrows are data-driven, although it has been designed in traditional CMMI-grid style. The CMMI levels are color-coded to signify the importance, red being low and green being high. Each of the elements belongs to their respective capabilities are in the grid, and capability maturity levels are assigned from the K-means cluster centroids (Table 4.5) and mapped to CMMI tiers. The colored grids reflect the maturity levels they fall under, with most organizations in higher tiers for CD and Monitoring. At the bottom, the four areas that influence the core capability domains are connected to them.

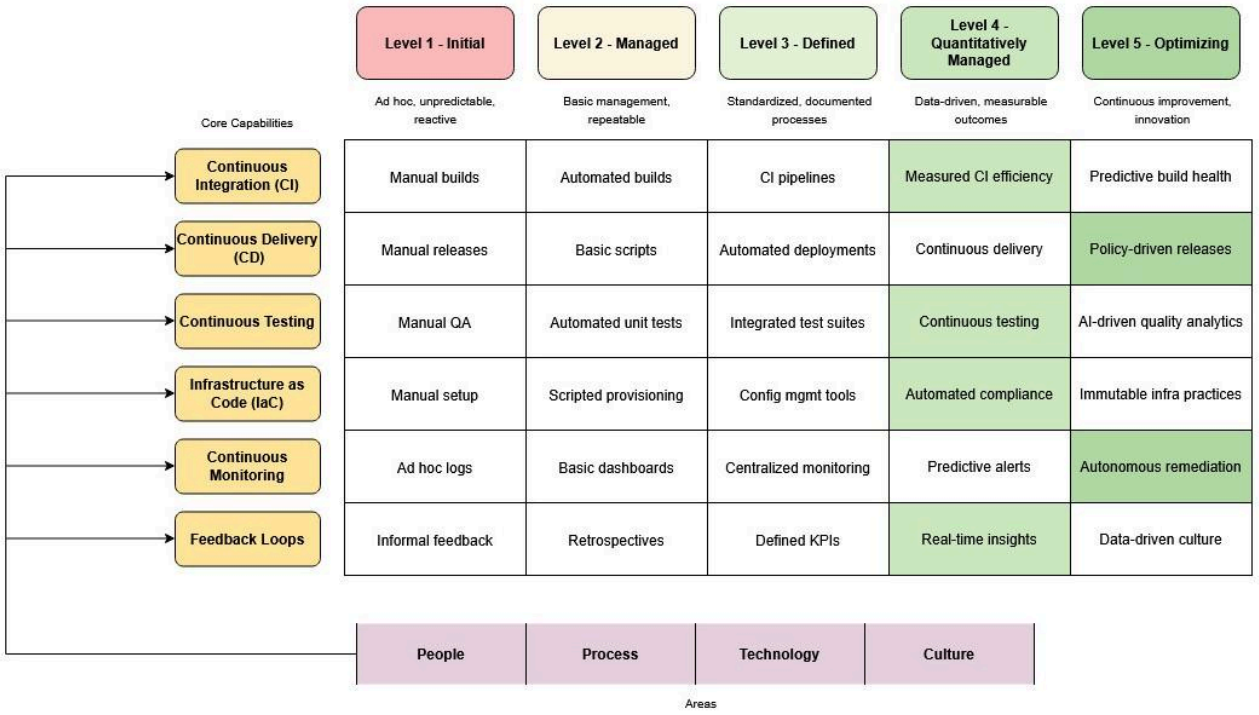

Fig. 5.2. DevOps Maturity Assessment Model

5.3 Security in DevOps Evaluation Model

The model in Figure 5.3 has been designed keeping the qualitative theme analysis of the constraints faced by Bangladesh's software firms in mind. This is a dual-ring model with the validated factors: the

inner ring represents the CPTM security practices (Q8-Q17), and the outer ring represents the security CSFs (Q18-Q25). Each item is shaded based on its Spearman Correlation strength with Build Pass Rate at the center; darker green indicates a stronger positive association, while lightest green indicates a weak association. At the center are the three security outcomes, with Build Pass Rate as the most stabilized one, while MTTR and Incident Frequency reflect their weaker correlation. At the bottom, the qualitative barrier themes are displayed with their observed frequencies.

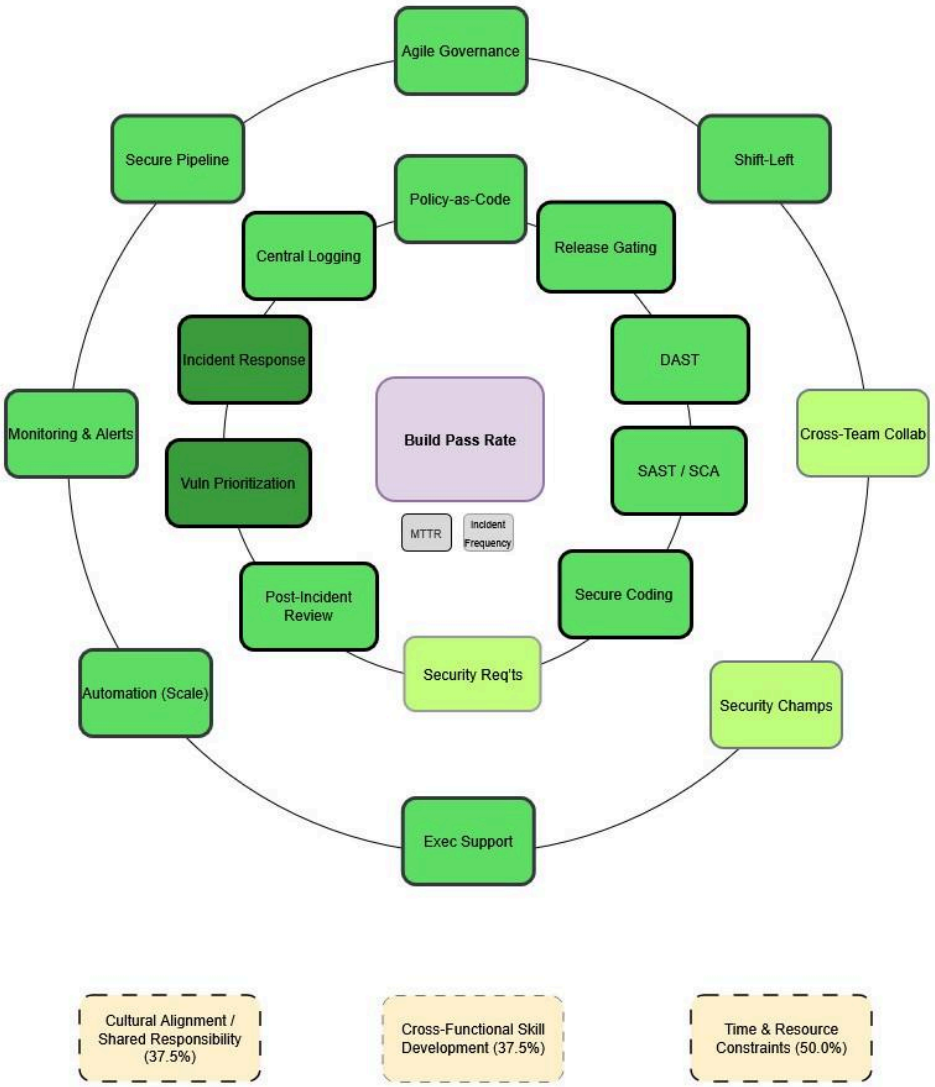

Fig. 5.3. Security in DevOps Evaluation Model

6. Conclusion and Future Works

6.1 The Role of DevOps Across the Software Development Lifecycle (SDLC)

This study examined DevOps through three dimensions: Critical Success Factors (CSFs), capability maturity, and security integration, to understand its role across the Software Development Lifecycle (SDLC) in Bangladesh. The overall findings show that DevOps practices in local firms are driven primarily by automation, which influences the planning, development, testing, deployment, and monitoring stages. CSF analysis showed the importance of cultural collaboration and continuous learning, highlighting how organizational and social factors shape DevOps functions at early planning and development phases. The maturity model assessment revealed uneven adoption across core capabilities, with Monitoring and Continuous Development being more advanced than Continuous Integration and Testing. Security evaluation showed that DevSecOps practices are growing but unevenly integrated, with strong alignment at the build and pipeline stages but weaker adoption in incident management and time management. Together, these outcomes demonstrate that DevOps in Bangladesh contributes meaningfully to shorter cycles and improved coordination across SDLC.

6.1.1 Mapping Findings to SDLC stages

The methods and analyses in Chapters 3 and 4 examined CSF effects, DevOps maturity, and DevSecOps correlations. The results are synthesized here to show how DevOps influences Planning, Development, Testing, Deployment, and Operations within software teams in Bangladesh. Table 6.1 shows the mapping to the quantitative and qualitative evidence.

SDLC Stage	**Evidence**	**Source**
Planning	Strong influence of collaboration and automation (higher standardized coefficients for X_2 and X_3)	Table 4.3
Development	High maturity observed in Continuous Development; use of Infrastructer-as-Code (IaC)	Table 4.5 and 4.4
Testing	Build and Test automation ranked among the strongest CSFs; lower reliability in Testing	Table 4.3, 4.5, and Section 4.2
Deployment	Build Pass Rate significantly correlated with	Table 4.6

	pipeline automation and security gating	
Operations/Maintenance	Continuous Monitoring shows the highest maturity across teams; qualitative themes highlight reliance on monitoring, alerting, and rapid response practices	Table 4.5 and 4.7

Table 6.1: Mapping of empirical findings to SDLC stages

6.2 Limitations

The sample sizes for the three questionnaires were modest (N = 23, N=31, N=25), which limits statistical power and generalizability. The cross-sectional approach can also restrict the ability to infer causal relationships, especially between the DevOps practices and outcomes. In CSF analysis, the MLR model showed risks of overfitting. For the maturity model assessment, Cronbach's alpha is low in certain capabilities, which shows limited internal consistency. However, this is typical for formative constructs. In security evaluation, zero-inflated incident outcome data affected the stability of correlations through Time to Fix and Incident Frequency.

These limitations are expected in exploratory research within an under-explored context, but should be addressed in future studies.

6.3 Ethical Considerations

The study followed ethical standards for human-focused software engineering studies. Participation in all of the questionnaires was voluntary, and respondents were informed of the purpose and their right to withdraw at any point. Personal identifiable details were not collected, and all responses were kept confidential and used strictly for this study’s analysis only. The study also avoided any organizationally sensitive disclosures, particularly for security practices.

6.4 Impact on the Society

The study contributes to the growing DevOps adoption in Bangladesh’s software companies and has practical implications, too. By providing insights into critical success factors, maturity levels, and security integration techniques, the research helps organizations benchmark their current practices and identify areas for improvement.

Increased DevOps adoption can support faster software delivery, improved reliability, and better systems. These would benefit the broader society that relies on digital services with each passing day.

6.4.1 Sustainability of the Work

The use of global frameworks and models and their adoption into Bangladesh's firms allows for systematic process enhancement. The proposed data-driven framework as a whole can be reused or extended in future organizational assessments, reducing the need for repeated model development.

6.5 Future Plan

Future work should expand the scope and depth of DevOps research in Bangladesh. Larger and more diverse samples would strengthen the empirical validation of the findings. Incorporating structural equation modeling (SEM) to analyze complex relationships between factors, maturity, and security practices would be possible with a larger dataset. The proposed models (CSF, Maturity grid, and dual-ring Security model) can he further validated through industrial case studies.

Bibliography

1. Azad, Nasreen, and Sami Hyrynsalmi. "DevOps critical success factors—A systematic literature review." *Information and Software Technology* 157 (2023): 107150.
2. Azad, Nasreen. "Understanding DevOps critical success factors and organizational practices." *Proceedings of the 5th International Workshop on Software-intensive Business: Towards Sustainable Software Business*. 2022.
3. Azad, Nasreen, and Sami Hyrynsalmi. "What are the critical success factors of DevOps projects? a systematic literature review." *International Conference on Software Business*. Springer, Cham, 2021.
4. Noorani, Noor Mohammed, et al. "Factor prioritization for effectively implementing DevOps in software development organizations: a SWOT-AHP approach." *Axioms* 11.10 (2022): 498.
5. Azad, Nasreen, and Sami Hyrynsalmi. "Devops challenges in organizations: Through professional lens." *International Conference on Software Business*. Cham: Springer International Publishing, 2022.
6. Wasserman, Larry. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.
7. Royce, Winston W. "Managing the development of large software systems: concepts and techniques." *Proceedings of the 9th international conference on Software Engineering*. 1987.
8. Rockart, John F. "Chief executives define their own data needs." *Harvard business review* 57.2 (1979): 81-93.
9. Braun, Michael T., and Frederick L. Oswald. "Exploratory regression analysis: A tool for selecting models and determining predictor importance." *Behavior research methods* 43.2 (2011): 331-339.
10. Beck, Kent, et al. "Manifesto for agile software development." (2001).
11. Chow, Tsun, and Dac-Buu Cao. "A survey study of critical success factors in agile software projects." *Journal of systems and software* 81.6 (2008): 961-971.
12. Montgomery, Douglas C. *Design and analysis of experiments*. John wiley & sons, 2017.
13. de Kock, Jacques, and Jacques Ophoff. "Critical success factors for integrating security into a DevOps environment." *15th Dewald Roode Workshop on Information Systems Security*

Research. IFIP Working Group 8.11/11.13, 2023.

14. Zhao, Xiaofan, et al. “Identifying the Primary Dimensions of DevSecOps: A Multi-Vocal Literature Review.” *Journal of Systems and Software*, vol. 214, Aug. 2024, p. 112063. *DOI.org (Crossref)*, https://doi.org/10.1016/j.jss.2024.112063.
15. Braun, Virginia, and Victoria Clarke. "Using thematic analysis in psychology." *Qualitative research in psychology* 3.2 (2006): 77-101.
16. Khan, Saad Ullah, et al. "Factors influencing vendor organizations in the selection of DevOps for global software development: an exploratory study using a systematic literature review." *Cognition, Technology & Work* 25.4 (2023): 411-426.
17. Van Belzen, Michiel, Jos Trienekens, and Rob Kusters. "Validation and Clarification of Critical Success Factors of DevOps Processes." *Proceedings of the 26th International Conference on Enterprise Information Systems, Angers, France: SCITEPRESS–Science and Technology Publications*. 2024.
18. Pérez-Sánchez, Juanjo, et al. "A theory on human factors in DevOps adoption." *Computer Standards & Interfaces* 92 (2025): 103907.
19. Leite, Leonardo, et al. "The organization of software teams in the quest for continuous delivery: A grounded theory approach." *Information and Software Technology* 139 (2021): 106672.
20. Rafi, Saima, et al. "Decision-Making Taxonomy of DevOps Success Factors Using Preference Ranking Organization Method of Enrichment Evaluation." *Mathematical Problems in Engineering* 2022.1 (2022): 2600160.
21. Teixeira, Daniel, et al. “A Maturity Model for DevOps.” *International Journal of Agile Systems and Management*, vol. 13, no. 4, Dec. 2020, pp. 464–511. *Zotero*, https://doi.org/10.1504/IJASM.2020.112343.
22. Tan, P. N., et al. "Classification: Basic concepts and techniques." *Introduction to Data Mining* (2019): 113-192.
23. College of Computer and Information Sciences, Prince Sultan University , Rafha Street, Riyadh, Saudi Arabia, et al. “A Research on DevOps Maturity Models.” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 3, Sept. 2019, pp. 4854–62. *DOI.org (Crossref)*, https://doi.org/10.35940/ijrte.C6888.098319.
24. Cusick, James J. *A Survey of Maturity Models from Nolon to DevOps and Their Applications in*

Process Improvement. *Zotero*.

25. Pereira, Rúben, and João Serrano. “A Review of Methods Used on IT Maturity Models Development: A Systematic Literature Review and a Critical Analysis.” *Journal of Information Technology*, vol. 35, no. 2, June 2020, pp. 161–78. *DOI.org (Crossref)*, https://doi.org/10.1177/0268396219886874.
26. Seixas, Nuno, and Marco Vieira. *An Integrated Maturity Model for Modern Software Development: Addressing Security, Artificial Intelligence and Governance in DevOps*. *Zotero*.
27. Tibshirani, Robert J., and Bradley Efron. "An introduction to the bootstrap." *Monographs on statistics and applied probability* 57.1 (1993): 1-436.
28. Abrar, Muhammad Faisal, et al. “Developing SPIM-TA: A Maturity-Level Framework for Systematic Process Improvement in Software Testing Automation.” *Ain Shams Engineering Journal*, vol. 16, no. 8, Aug. 2025, p. 103472. *DOI.org (Crossref)*, https://doi.org/10.1016/j.asej.2025.103472.
29. Gasparait, Monika, and Saulius Ragaišis. *Comparison of Devops Maturity Models*. *Zotero*.
30. Hemon, Aymeric, et al. “Conceptualizing the Transition from Agile to DevOps: A Maturity Model for a Smarter IS Function.” *Smart Working, Living and Organising*, edited by Amany Elbanna et al., vol. 533, Springer International Publishing, 2019, pp. 209–23. IFIP Advances in Information and Communication Technology. *DOI.org (Crossref)*, https://doi.org/10.1007/978-3-030-04315-5_15.
31. Kumar, Ankur, et al. *Assessment of DevOps Maturity in Software Development Organisations: An Practitioners Perspective*. 2022. *Zotero*.
32. Good, P. "Permutation Tests-A Practical Guide to Resampling Methods for Testing Hypotheses." (2000).
33. Alamin, Zumhur, et al. “Evolving DevOps Practices in Modern Software Engineering: Trends, Challenges, and Impacts on Quality and Delivery Performance.” *Journix: Journal of Informatics and Computing*, vol. 1, no. 1, Mar. 2025, pp. 21–29. *DOI.org (Crossref)*, https://doi.org/10.63866/journix.v1i1.4.
34. Bezemer, Cor-Paul, et al. “How Is Performance Addressed in DevOps?” *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering* [Mumbai India], 2019, pp. 45–50. *DOI.org (Crossref)*, https://doi.org/10.1145/3297663.3309672.
35. De Feijter, Rico, et al. “DevOps Competences and Maturity for Software Producing

Organizations." *Enterprise, Business-Process and Information Systems Modeling*, edited by Jens Gulden et al., vol. 318, Springer International Publishing, 2018, pp. 244–59. Lecture Notes in Business Information Processing. *DOI.org (Crossref)*, https://doi.org/10.1007/978-3-319-91704-7_16.

36. Gimenez, Paulo J A, and Gleison Santos. *DevOps Maturity Diagnosis – A Case Study in Two Public Organizations*. 2020. *Zotero*.
37. Jayakody, J. A. V. M. K., and W. M. J. I. Wijayanayake. "DevOps Maturity; A Systematic Literature Review." *2024 International Research Conference on Smart Computing and Systems Engineering (SCSE)* [Colombo, Sri Lanka], 2024, pp. 1–6. *DOI.org (Crossref)*, https://doi.org/10.1109/SCSE61872.2024.10550493.
38. Marnewick, Carl, and Josef Langerman. "DevOps and Organizational Performance: The Fallacy of Chasing Maturity." *IEEE Software*, vol. 38, no. 5, Sept. 2021, pp. 48–55. *DOI.org (Crossref)*, https://doi.org/10.1109/MS.2020.3023298.
39. Neubrand, Thomas, and Thorsten Haendler. "Development of a GQM-Based Technique for Assessing DevOps Maturity:" *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* [Budapest, Hungary], 2020, pp. 117–29. *DOI.org (Crossref)*, https://doi.org/10.5220/0010177801170129.
40. Orozco Garcés, Carlos Eduardo, et al. "Towards a Metrics Model for Devops Adoption: Case Study of an Enterprise Application." SSRN, 2023. *DOI.org (Crossref)*, https://doi.org/10.2139/ssrn.4536719.
41. Rajab, Rania, and Mouhib Alnoukari. "DevOps Integration With Capability Model Maturity Integration: A Systematic Mapping Review." *IEEE Access*, vol. 13, 2025, pp. 31829–41. *DOI.org (Crossref)*, https://doi.org/10.1109/ACCESS.2025.3542630.
42. Sethupathy, Utham Kumar Anugula. "Navigating Continuous Improvement: An In-Depth Analysis of Lean-Agile and DevOps Maturity Models." *Journal of Software Engineering and Applications*, vol. 18, no. 09, 2025, pp. 317–35. *DOI.org (Crossref)*, https://doi.org/10.4236/jsea.2025.189019.
43. Akbar, Muhammad Azeem, et al. "Toward Successful DevSecOps in Software Development Organizations: A Decision-Making Framework." *Information and Software Technology*, vol. 147, July 2022, p. 106894. *DOI.org (Crossref)*, https://doi.org/10.1016/j.infsof.2022.106894.
44. Bedoya, Martin. *Enhancing DevSecOps Practice with Large Language Models and Security*

Chaos Engineering. *Zotero*.

45. Binbeshr, Farid, and Muhammad Imam. *Comparative Analysis of AI-Driven Security Approaches in DevSecOps: Challenges, Solutions, and Future Directions*. 2025. *Zotero*.
46. Caniglia, Alessandro, et al. "FOBICS: Assessing Project Security Level through a Metrics Framework That Evaluates DevSecOps Performance." *Information and Software Technology*, vol. 178, Feb. 2025, p. 107605. *DOI.org (Crossref)*, https://doi.org/10.1016/j.infsof.2024.107605.
47. Cheenepalli, Jayaprakashreddy, et al. *Advancing DevSecOps in SMEs: Challenges and Best Practices for Secure CI/CD Pipelines*. *Zotero*.
48. Farooq, Muhammad Shoaib, and Usman Ali. *Harnessing the Potential of Blockchain in DevOps: A Framework for Distributed Integration and Development*. *Zotero*.
49. Feio, Clarisse, et al. "An Empirical Study of DevSecOps Focused on Continuous Security Testing." *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* [Vienna, Austria], 2024, pp. 610–17. *DOI.org (Crossref)*, https://doi.org/10.1109/EuroSPW61312.2024.00074.
50. Gbenle, Peter, et al. *A DevSecOps-Centered Conceptual Model for Continuous Integration and Secure Deployment in Software Development Lifecycles*. *Zotero*.
51. Gupta, Akanksha. "An Integrated Framework for DevSecOps Adoption." *International Journal of Computer Trends and Technology*, vol. 70, no. 6, July 2022, pp. 19–23. *DOI.org (Crossref)*, https://doi.org/10.14445/22312803/IJCTT-V70I6P102.
52. Ibrahim, Fatima, et al. "Bridging the Security Gap in SDLC: Challenges and Solutions for Local IT Firms in Bangladesh." *EPJ Web of Conferences*, edited by H. Liu et al., vol. 328, 2025, p. 01046. *DOI.org (Crossref)*, https://doi.org/10.1051/epjconf/202532801046.
53. Kumar, Rakesh, and Rinkaj Goyal. "Modeling Continuous Security: A Conceptual Model for Automated DevSecOps Using Open-Source Software over Cloud (ADOC)." *Computers & Security*, vol. 97, Oct. 2020, p. 101967. *DOI.org (Crossref)*, https://doi.org/10.1016/j.cose.2020.101967.
54. Leite, Leonardo, et al. "A Survey of DevOps Concepts and Challenges." *ACM Computing Surveys*, vol. 52, no. 6, Nov. 2020, pp. 1–35. *DOI.org (Crossref)*, https://doi.org/10.1145/3359981.
55. Moyo, Sibonile, and Ernest Mnkandla. "A Novel Lightweight Solo Software Development

Methodology With Optimum Security Practices." *IEEE Access*, vol. 8, 2020, pp. 33735–47. *DOI.org (Crossref)*, https://doi.org/10.1109/ACCESS.2020.2971000.

56. Moyon, Fabiola, et al. "Integration of Security Standards into DevOps Pipeline." figshare, 2020, p. 1391549 Bytes. *DOI.org (Datacite)*, https://doi.org/10.6084/M9.FIGSHARE.11294534.
57. Rafi, Saima, et al. "Towards a Hypothetical Framework to Secure DevOps Adoption: Grounded Theory Approach." *Proceedings of the Evaluation and Assessment in Software Engineering* [Trondheim Norway], 2020, pp. 457–62. *DOI.org (Crossref)*, https://doi.org/10.1145/3383219.3383285.
58. Rajapakse, Roshan N., et al. "Challenges and Solutions When Adopting DevSecOps: A Systematic Review." *Information and Software Technology*, vol. 141, Jan. 2022, p. 106700. *DOI.org (Crossref)*, https://doi.org/10.1016/j.infsof.2021.106700.
59. Ramaj, Xhesika, et al. "Holding on to Compliance While Adopting DevSecOps: An SLR." *Electronics*, vol. 11, no. 22, Nov. 2022, p. 3707. *DOI.org (Crossref)*, https://doi.org/10.3390/electronics11223707.
60. Sánchez-Gordón, Mary, and Ricardo Colomo-Palacios. "Security as Culture: A Systematic Literature Review of DevSecOps." *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* [Seoul Republic of Korea], 2020, pp. 266–69. *DOI.org (Crossref)*, https://doi.org/10.1145/3387940.3392233.
61. Sinan, Maysa, et al. "Integrating Security Controls in DevSecOps: Challenges, Solutions, and Future Research Directions." *Journal of Software: Evolution and Process*, vol. 37, no. 6, June 2025, p. e70029. *DOI.org (Crossref)*, https://doi.org/10.1002/smr.70029.
62. Jayakody, Vihara, and Janaka Wijayanayake. "Critical success factors for DevOps adoption in information systems development." *International Journal of Information Systems and Project Management* 11.3 (2023): 60-82.