

The Impact of Team Dynamics on Software Quality and Productivity: Evidence from South Africa

Tshipuke Vhangwele¹, Bassey Isong², Adnan Abu-Mahfouz³

^{1,2}Computer Science Department, North-West University, Mafikeng, South Africa

³Council for Industrial Scientific and Industrial Research, Pretoria, South Africa

Received:

December 6, 2025

Revised:

January 21, 2026

Accepted:

February 28, 2026

Published:

March 2, 2026

Corresponding Author:

Author Name*:

Tshipuke Vhangwele

Email*:

tshipukevhangwele@gmail.com

DOI:

10.63158/journalisi.v8i1.1438

© 2026 Journal of Information Systems and Informatics. This open-access article is distributed under a (CC-BY License)



Abstract. Software quality and productivity are influenced not only by technical practices but also by the social dynamics within development teams. This study investigates the combined effect of team dynamics, including trust, communication, collaboration, diversity, and conflict resolution, and software development practices on project outcomes. A mixed-methods design combined regression, Spearman's Rho, and thematic analysis of survey data from 124 South African software professionals. The findings indicate that trust is the strongest positive predictor of software quality and productivity, while communication effectiveness and the use of collaboration tools also improve software outcomes. Equally, unstructured collaboration, excessive planning meetings, and poorly managed communication channels negatively affect performance. Diversity and effective conflict resolution were positively associated with productivity and efficiency. Thematic analysis corroborated these findings, illustrating how unclear communication, low trust, and dysfunctional collaboration lead to delays, rework, and lower quality. The study confirms that successful software outcomes emerge from the alignment of social and technical subsystems, highlighting the critical role of team dynamics in realising the full potential of software development practices. It contributes empirical evidence from an understudied developing-country context and proposes a socio-technical framework to enhance software quality and productivity.

Keywords: Team dynamics, Software development practices, Software quality, Productivity, Socio-technical systems

1. INTRODUCTION

Software development is integrally a socio-technical endeavour, where success depends not only on technical proficiency but also on the interplay of human dynamics, team structures, and development methodologies. Empirical research shows that while individual skill is important, the collective ability of a team to collaborate and communicate, underpinned by psychological safety (PS) and trust that enable open dialogue, risk-taking, and learning, is a central determinant of project success [1], [2], [3]. The industry-wide shift towards Agile and DevOps has intensified focus on human factors.

Agile and DevOps emphasise iterative delivery, continuous feedback, and close collaboration across roles, positioning teamwork, communication, and team culture as critical drivers of software quality and productivity [4], [5]. Although these frameworks provide structured processes, their effectiveness is dependent on how well teams manage human-centric dynamics in practice. Communication, for example, demonstrates a nuanced, inverted U-shaped relationship with productivity: insufficient interaction hinders coordination, while excessive or poorly managed communication creates overload and reduces performance [6], [7], [8]. As a result, modern leadership increasingly adopts shared and facilitative approaches that empower team members while minimising organisational disruption [9].

Team composition and structure strongly influence coordination, knowledge sharing, and productivity. Agile teams, typically small and cross-functional, perform best when limited to nine members or fewer. Studies confirm that smaller teams correlate with higher productivity, fewer defects, and lower rework [10]; stability and reduced turnover further improve velocity and predictability [1], [8]. In DevOps settings, collaboration intensity between Development and Operations teams is a strong predictor of deployment frequency, incident rates, and failure occurrences, regardless of organisational configuration [4]. Similarly, PS and trust are equally important. Teams that promote environments where members feel safe to raise concerns, admit mistakes, and collaborate openly achieve higher quality outputs [2]. A climate of interpersonal safety supports constructive collaboration, while trust, rooted in perceptions of ability, benevolence, and integrity, is particularly essential in distributed teams [2], [11], [12], [13].

Building and sustaining trust relies on supportive contexts, facilitative leadership, organisational backing, empathy, and socio-cultural strategies [2], and emotional intelligence further strengthens trust, reduces stress, and enhances performance [13]. Effective coordination also depends on shared mental models, mutual trust, and structured communication, though communication remains non-linear: too little hampers coordination, while excessive or poorly managed interaction generates overload and friction [1], [6], [7], [8], [14].

Furthermore, development practices directly affect software outcomes. Practices such as Test-Driven Development (TDD) and Testability-Driven Development (TsDD) improve quality, engagement, defect reduction, and maintainability [15], [16], [17], while rapid feedback loops, via DevOps integration or behavioral feedback, improve productivity and quality [4], [16], [18], [19], [20]. Human-related factors, including PS, team capabilities, customer involvement, autonomy, emotional intelligence, and growth mindset, remain central to project outcomes [8], [11], [12], [21], [22], [23]. Diversity, when supported by inclusive practices, enhances creativity, decision-making, and innovation, though geographic and cultural dispersion introduces challenges [24]. Teams must also balance tensions between productivity and quality and between individual and collective needs, where stability, clear roles, and coordinated distributed work remain important determinants of performance [25].

Despite these insights, existing research has limitations. Many of the empirical studies are geographically constrained, focusing on countries such as Portugal or Colombia, etc., which restricts generalizability [3], [11]. Reliance on survey data centers on subjective perceptions instead of objective measures, while cross-sectional designs limit causal interpretation [11]. Furthermore, there is no consensus on defining and measuring productivity and quality, which remain context-dependent. While software development practices are central to success, they require cultural and organisational integration within teams [23], [26]. Most studies examined team dynamics and practices separately, leaving their interaction largely unexplored [7], [27], [28]. These studies, therefore, establish the basis for a new survey in the South African setting, generating context-specific data and enabling comparison with international findings. This is geared towards advancing understanding of the factors shaping software team performance in South Africa. To achieve this, we surveyed software development organisations aimed at

understanding the combined influence of software team dynamics and practices on software outcomes in South African software development ecosystems. The study designed an online survey targeting critical organisations and professions after obtaining ethical clearance.

The contributions of this paper are summarised as follows:

- 1) Provides empirical evidence on the interaction between team dynamics and software development practices in the South African context.
- 2) Identifies which aspects of team dynamics most strongly influence software quality and productivity.
- 3) Examines the role of socio-technical integration, combining human-centric factors with disciplined technical practices.
- 4) Generates context-specific data that allows comparison with international studies, improving generalizability.
- 5) Proposes a framework, offering insights for improving team performance and project outcomes within Agile and DevOps environments.

The remaining parts of the paper are organized as follows: Section 2 presents the related works, Section 3 presents the research methodology followed, while Section 4 presents the results and analysis of the findings. Section 5 discusses the overall findings and their implications, Section 6 presents the threats to validity of the study, Section 7 presents the proposed framework, and Section 8 concludes the paper.

2. RELATED WORKS

This section presents some of the existing empirical studies that highlighted the interplay between team dynamics and development practices. It focuses on aspects directly relevant to understanding their influence on software outcomes, including PS and trust, team structure and composition, communication and collaboration, methodologies and engineering practices, and human-related success factors. Table 1 summarises these studies.

2.1. Psychological safety and trust

PS and interpersonal trust are foundational for team performance. PS is a shared belief that the team is safe for interpersonal risk-taking, enabling members to voice concerns, admit errors, and engage in collaborative problem-solving [3],[12]. Barros et al. [11] reported that high PS supports quality-enhancing behaviours by allowing technical skills and stakeholder interactions to flourish. On the other hand, trust, encompassing perceptions of ability, benevolence, and integrity, is essential in both distributed and co-located teams and is strengthened by supportive environments, facilitative leadership, organisational backing, empathy, and emotional intelligence [2], [11], [13]. Studies have shown that PS indirectly improves project success by enabling team capabilities and customer involvement, while trust and emotional intelligence further improve productivity and quality. Rezvani and Khosravi [13] added that emotional intelligence further reduces stress and indirectly enhances productivity and quality. While most studies rely on self-reported perceptions, the present study contributes by examining these factors within a previously unstudied South African context.

2.2. Team structure and composition

Several studies have shown that team composition, size, and stability strongly influence coordination, knowledge sharing, and productivity. For instance, Strode et al. [1] and Bandaru [10] found that Agile teams, typically small and cross-functional, perform best with six to nine members, achieving higher productivity, fewer defects, and lower rework. Moreover, DevOps structures that promote high collaboration between development and operations teams outperform siloed models, with better on-deployment frequency, incident reduction, and service reliability [4], [5]. Moreover, Scott et al. [8] reported that stability and low turnover further improve velocity and predictability. Lambiase et al. [24] further noted that geographical and cultural dispersion can influence productivity, although in co-located teams these factors are less critical.

2.3. Communication and collaboration dynamics

Communication frequency and structure critically shape team performance. Studies have consistently shown that moderate interaction enhances coordination and knowledge sharing, while excessive or poorly managed communication generates overload and reduces productivity, reflecting an inverted U-shaped relationship [6], [7]. In the same vein, Restrepo-Tamayo et al. [19] showed that leadership, communication, and teamwork

interact to form positive feedback loops that improve productivity. Berrezueta-Guzman et al. [29] also emphasised that accurate self-assessment, transparent communication, and structured conflict resolution support effective collaboration, although their study is limited to student projects. Strode et al. [1] also offered a framework integrating shared leadership, peer feedback, adaptability, and team orientation with trust and communication as coordinating mechanisms, offering empirical grounding for agile team dynamics, though the study does not directly measure software outcomes.

2.4. Methodologies and practices

Studies have also shown that disciplined development practices translate team potential into measurable improvements in software quality and productivity. Lindsjrn et al. [30] stated that high-quality agile processes, measured as Teamwork Quality (TWQ), which includes communication, coordination, mutual support, balanced contributions, effort, and cohesion, positively impact team performance, learning, and satisfaction. Moreover, practices such as TDD [15], [16] and TsDD [17] improve maintainability, reduce defects, and enhance engagement, though overemphasis on passing tests may compromise overall design quality. Also, whether via DevOps integration or behavioural feedback mechanisms, further accelerate problem detection and resolution [18], [19], [20].

2.5. Human-related critical success factors

Beyond structural and methodological aspects, factors such as PS, team capabilities, customer involvement, team autonomy, emotional intelligence, and growth mindset are central to team outcomes [2], [11], [12], [15], [31]. These factors enable teams to coordinate effectively, adapt to change, and maintain performance under varying project conditions. Alami et al. [12] also showed that PS is socially and human-centric, emphasising leadership, collective decision-making, and openness as drivers of effective collaboration.

2.6. Diversity and cultural dispersion

Diversity improves innovation and decision-making when supported by inclusive practices, though cultural and geographic dispersion introduces challenges related to hierarchy perception, communication, and long-term orientation [24]. Effective management of diverse teams requires shared norms, hybrid team cultures, and active mitigation of time zone and cultural differences. These factors outline how team

dynamics and collaboration influence software outcomes in distributed or culturally varied contexts.

2.7. Core tensions and trade-offs

Software development inherently involves trade-offs, particularly between quality and productivity. Investments in quality may initially slow delivery but improve long-term outcomes, whereas a focus on rapid throughput can increase technical debt. Sustaining high performance requires balancing individual and collective contributions, clarifying roles, and maintaining team stability [1], [25]. These trade-offs interact with team dynamics and development practices, reinforcing the need for coordinated management strategies to optimise both software quality and productivity.

Table 1. Summary of related empirical studies

Study/ Year	Context/ Sample	Methods	Main Constructs	Main Findings	Focus	Limitations
[1] 2023	Co-located agile teams, single product	Literature review, focus groups, case studies	Shared leadership, peer feedback, adaptability, team orientation, mental models, trust, communication	Identifies behavioural markers linking agile practices to team effectiveness	Team dynamics: indirect links to productivity & quality	Limited to co-located, non-safety-critical teams
[10] 2024	Scrum teams in one enterprise (n=22, 12, 6)	Empirical observation, metrics, poll	Team size, climate, diversity, leadership, support	Smaller teams (6) performed best (90% on-time, low rework); communication issues in larger teams	Productivity, quality	Limited to one enterprise, small sample
[11] 2024	Agile teams in Portugal	Survey + PLS analysis	Team capability, customer involvement, PS, autonomy	Capability & involvement drive success; PS indirect via TC & CI; autonomy moderates effects	ASD project success	Limited generalisability (Portugal only), survey design
[24] 2024	25 GitHub communities, 140 surveyed	Statistical modelling, survey	Cultural & geographical dispersion, socio-technical factors	Dispersion affects productivity (positive/negative); communication mitigates risks	Productivity	Limited dataset, productivity proxy (commits)
[31] 2022	Agile teams in NA & Europe (25 devs, 5 teams)	Daily surveys, interviews	Individual vs. team productivity, collaboration, meetings	Individual productivity predicts team productivity; remote work complicates assessment	Productivity	Small sample, self-reported, COVID-19 context
[4] 2025	104 DevOps professionals worldwide	Survey + statistical analysis	Team formation type, collaboration, performance metrics	DevOps adoption improves performance; high-collab teams outperform others	Lead time, deployment frequency,	Self-reported, context variability, cross-sectional

Study/ Year	Context/ Sample	Methods	Main Constructs	Main Findings	Focus	Limitations
					recovery, incidents	
[12] 2023	Agile teams (N=365)	Mixed methods: survey + interviews	Leadership, decision- making, autonomy, openness, PS	Leadership integrity, no- blame, openness fosters PS; technical factors insufficient	PS, communication, learning	Correlational, ASD scope, construct subjectivity
[29] 2024	Student software projects	GitLab commits, survey	Contribution alignment, leadership, communication, conflict resolution	Accurate self-assessment, structured conflict resolution improve performance	Academic performance	Limited to student teams, self- reported bias
[18] 2024	130 students, 15 Scrum projects (7 experimental, 8 control)	Comparative case study; JIRA plugin (ProDynamics); surveys; correlation analysis	Behavioural feedback (communication, emotions, meetings, productivity); performance measures	Automated feedback improved Development Performance and satisfaction; no significant change in Team Performance; 77% of users reported positive impact	Development performance, team performance	Student sample, self-reported data, aggregated emotional data, unmeasured prior experience
[19] 2024	Software teams, industry professionals	Systematic mapping + survey	Communication, leadership, teamwork	Human Factors form reinforcing loops boosting productivity & quality; weak measurement practices.	Productivity, quality	Measurement validity issues, survey generalisability
[25] 2021	Large Australian software development company / organisational participants	Ethnography, participant observation, interviews, rich pictures	Entanglement of software quality, productivity, time, infrastructure, and human practices (post- human perspective)	Software quality is dynamically linked with productivity, infrastructure, and human practice; investing in quality enhances long- term productivity; mentoring and expertise development are critical	Software quality, developer productivity, mentoring effectiveness	Single- organisation study; findings may be context- specific; limited generalisability

As shown in Table 1, recent studies show the central influence of human and socio-technical factors on agile software outcomes. Evidence indicates that PS, leadership integrity, customer involvement, and collaboration are strong predictors of success, while technical practices alone are insufficient [1], [11], [12], [19]. In the same vein, productivity and quality also tend to improve when teams are small, cohesive, and supported by clear communication structures [10], [24], [31]. Although much of the evidence is from industry settings in Europe, North America, and selected enterprise contexts, studies involving students show similar patterns, suggesting that these dynamics emerge early in team

formation [18], [29]. In nations such as South Africa, agile adoption is increasing but may be constrained by organisational maturity, skills shortages, and infrastructure gaps. These findings point to the value of strengthening collaborative capability, PS, and measurement practices. The literature further indicates that local empirical work should consider socio-cultural diversity, remote collaboration norms, and hybrid team structures, which may moderate performance outcomes [24], [31].

3. RESEARCH METHODOLOGY

This section presents the approaches used in achieving the results presented in this study. The workflow is shown in Figure 1 and discussed as follows.

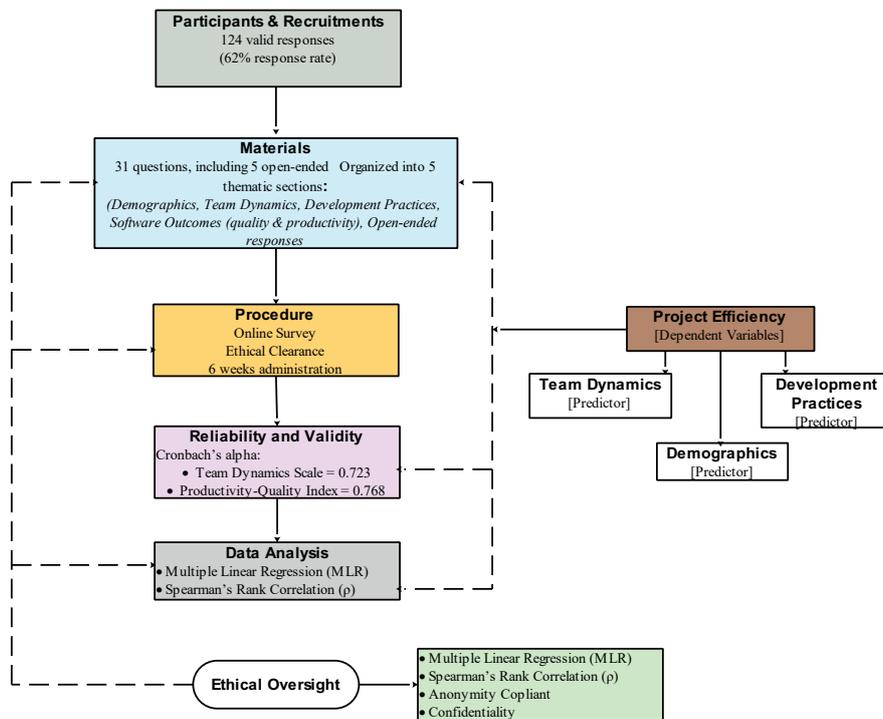


Figure 1. Methodology workflow

3.1. Participants and recruitment

The survey targeted software development professionals working in South Africa, chosen because their work involves structured team interactions and software practices relevant to examining the influence of team dynamics and development methodologies on software outcomes. Participants were recruited through multiple channels: LinkedIn

professional networks, online developer communities, and targeted emails to industry professionals. To avoid overrepresentation, invitations were randomised across subgroups. A total of 124 valid responses were received from the anticipated 200, yielding a 62% response rate. Participants included software developers, project managers, analysts, and designers, with experience ranging from less than five years to over ten years. Organisations represented varied from small teams within large corporations to independent software development companies, allowing a wide range of experiences to be captured.

3.2. Materials

Firstly, a literature review of related studies [1]-[10] informed the survey design, helping identify relevant constructs and formats while avoiding potential biases. The survey comprised 26 main questions and 5 sub-questions, organized into five sections: demographics (gender, role, experience, organization type), team dynamics (communication, trust, leadership, conflict resolution), software development practices (methodologies, collaboration tools, technical debt management), software outcomes (perceptions of software quality, productivity, and project success), and open-ended responses for contextual insights. A combination of Likert scales, multiple-choice, and open-ended items captured both measurable constructs and qualitative insights. The survey was piloted and evaluated by experts to ensure clarity and relevance.

3.3. Procedure

Following ethics clearance from the North-West University (NWU) Research Ethics Committee, the self-administered online survey was hosted on Google Forms. Invitations were distributed across diverse subgroups representing various roles, experience levels, and organisational settings. Participation was voluntary, and an introductory message explained the study's purpose, confidentiality, and anonymity. No personal identifiers were collected. The survey remained open for approximately six weeks, and respondents typically completed it within 15–20 minutes.

3.4. Reliability and Validity

The survey's internal consistency was assessed using Cronbach's alpha. The team dynamics effectiveness scale, comprising four items: communication effectiveness, trust, diversity, and conflict resolution, achieved an alpha of 0.723. A two-item magnitude index

measuring the impact of team dynamics on productivity and quality produced an alpha of 0.768 with an inter-item correlation of 0.623. These results confirm sufficient reliability for subsequent statistical analyses.

3.5. Data preparation and analysis

In this study, to ensure precision and consistency, the collected data were coded and cleaned. In this case, categorical variables were transformed into dummy variables to enable their inclusion in regression-based analyses, ordinal variables were appropriately scaled to preserve their ranked nature while allowing meaningful statistical comparison, and missing data were handled using listwise deletion to retain only complete cases. In addition, to analyse the data collected, multiple methods were used, including multiple linear regression (MLR), Spearman's Rank Correlation, etc. The MLR statistical method was used to examine how team dynamics and development practices impact software outcomes. The dependent variable was Project Efficiency, and independent variables included demographic factors (gender, role, experience, organisational size), team attributes (trust, communication, conflict resolution, leadership style, team size), and development practices (methodology, quality and productivity metrics, collaboration patterns, technical debt management). The regression model is expressed as shown in Equation 1.

$$PROJEFF = \alpha + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon \quad (1)$$

Where *PROJEFF* is the project efficiency score, $X_1 \dots X_k$ represent the predictor variables, $\beta_1 \dots \beta_k$ are the estimated coefficients, α is the intercept, and ε is the error term.

The Spearman's Rank Correlation uses Spearman's Rho to assess linear relationships among variables, accommodating mixed data types (categorical and ordinal). It was calculated as shown in Equation 1.

$$\rho = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2-1)} \quad (2)$$

Where ρ = Spearman's rank correlation coefficient, d_i = difference between the two ranks of each observation, and n number of observations.

In addition, qualitative data were gathered through open-ended questions included in the survey. ATLAS.ti was used to facilitate the organisation of data and the synthesis of findings. The initial step was the generation of the word clouds that offered a visualisation of the most common codes and hence, the identification of notable concepts among the respondents. These codes served as the basis for identifying themes relevant to the study. The second step involved conducting a thematic analysis in ATLAS.ti to systematically examine and develop themes, following the approach described by Braun and Clarke. This analysis helped identify recurring patterns related to team dynamics, development practices, and their influence on software quality and productivity. The qualitative insights were then cross-validated with the quantitative results to strengthen and enrich the interpretation of the findings.

3.6. Ethical considerations

Before this study commenced, ethics approval was obtained from the NWU Research Ethics Committee. To this end, participation was voluntary, responses were anonymous, and confidentiality was maintained in accordance with the Protection of Personal Information Act (POPIA).

4. RESULTS AND DISCUSSION

In this study, we combined quantitative analyses, including statistical inference, regression, and correlation, with qualitative thematic analysis to investigate the relationships between team dynamics, software development practices, and software outcomes. The results are presented in five sections: descriptive participant data, institutional characteristics, empirical team and practice factors, quantitative analyses, and qualitative insights.

4.1. Occupational roles

Figure 2 (a) shows that most respondents were male, about 64%, with females making up 34%, while 2% preferred not to disclose gender. Figure 2 (b), which shows the current roles of the respondents, shows that software developers formed the majority, representing over 60% of the sample. Notably, several respondents held overlapping roles, for instance, project managers who also identified as engineers or business

analysts. This overlap highlights the multifunctional nature of software work and reflects the complexity of team structures in practice.

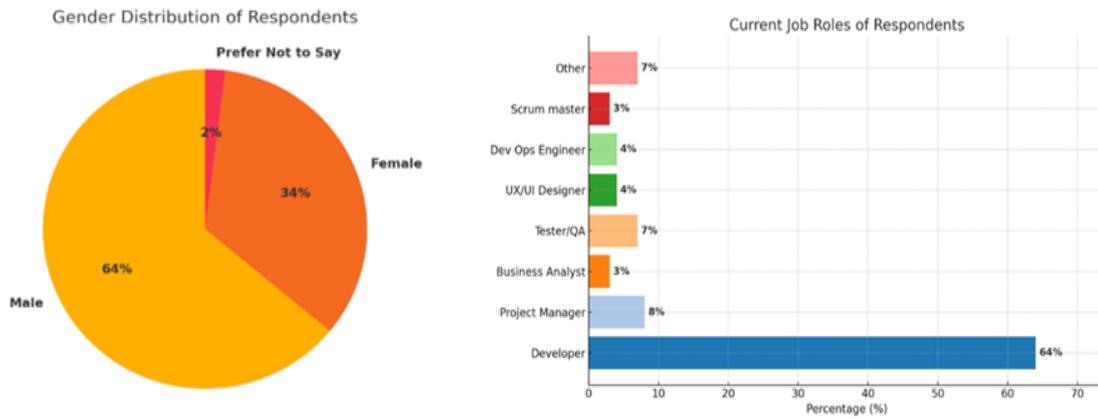


Figure 2. (a) Gender distribution, (b) Distribution of respondents by current job roles

4.2. Work Experience

Most participants had between one and six years of professional experience. As shown in *Fig. 3*, specifically, 39% reported fewer than three years, while slightly over 20% had between three and six years. A minority reported less than one year or more than ten years of experience. These results suggest a predominance of early- to mid-career professionals in the sample, offering insights into perspectives from those actively shaping current industry practices. The inclusion of both novice and more seasoned practitioners strengthens the credibility and transferability of the findings.



Figure 3. Distribution of years of experience

4.3. Institutional Factors

As shown in Figure 4, nearly half of the respondents worked in large organisations with more than 500 employees. However, within these organisations, teams were relatively small: 38% consisted of 6-10 members, while 27% worked in teams of 1-5. This indicates that software development often occurs in compact, cross-functional groups embedded within large institutions. Such a structure underscores the importance of studying how smaller team dynamics contribute to productivity and quality, even in resource-rich environments.

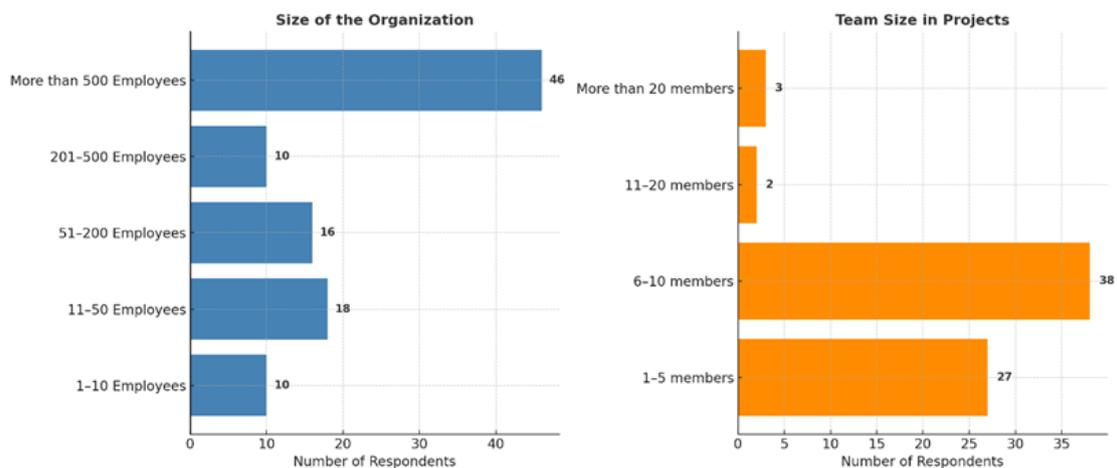


Figure 4. Distribution by team and organisational Size

4.4. Empirical Factors

4.4.1. Principal leadership style and teamwork

Figure 5 (a) revealed that 55% of developers worked under democratic leadership, which tends to promote participation and shared decision-making. In contrast, 27% reported authoritarian leadership, while smaller groups experienced laissez-faire approaches characterised by independence and autonomy. Figure 5 (b) also showed that teamwork arrangements reflected similar diversity: 58% operated in hybrid formats, 23% worked fully remotely, and 19% were entirely on-site. These patterns highlight the central role of leadership and work arrangements in shaping collaboration and project execution.

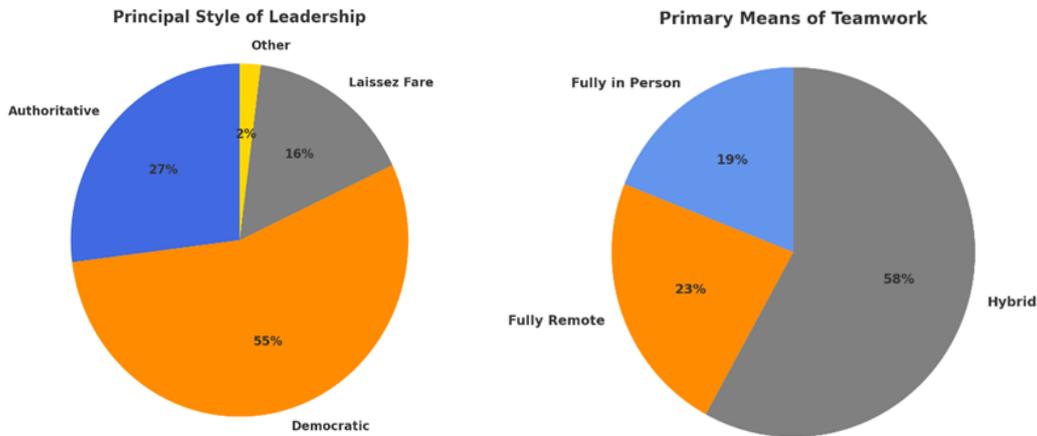


Figure 5. (a) Leadership style, (b) Teamwork style

4.4.2. Frequency of communication and address of technical debt

As shown in Figure 6, daily communication was common, reported by 60% of respondents, while 25% communicated several times per week. In addition, planning meetings were also viewed as essential for coordination: 34% attended daily, 32% weekly, and 23% bi-weekly, whereas only 12% participated monthly or less. This indicates that frequent communication and structured planning remain integral to managing software projects. However, communication was also tied to addressing technical debt, where inconsistencies in information flow influenced how quickly and effectively issues were resolved.

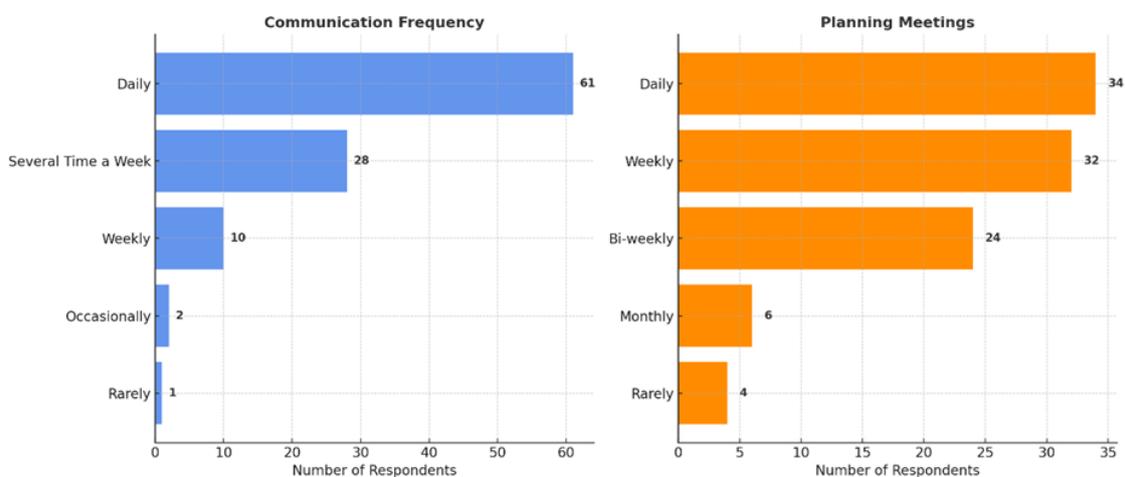


Figure 6. Communication frequency and meeting planning

4.4.3. Communication channels and challenges

Like other international counterparts, our analysis revealed that communication is critical to team performance [24], [31]. Figure 7 shows that Microsoft Teams was the most frequently used communication tool with 40%, followed by Slack (32%) and Zoom (26%). Email use was minimal, about 2%, indicating its limited utility for real-time coordination. Despite the availability of tools, several challenges persisted: lack of time (43%) and lack of clarity (37%) were the most significant barriers, followed by language and cultural issues (15%), time zone differences (3%), and unclear roles (2%). Such challenges illustrate the strain between technical measures and human communication practices.

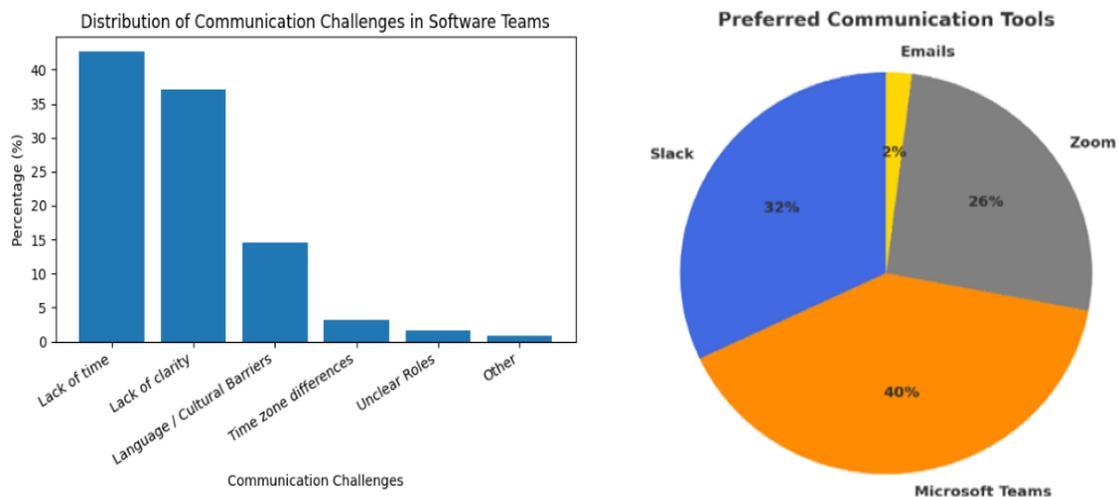


Figure 7. Communication channels and challenges distribution

4.4.4. Effectiveness of team dynamics aspects and software development practices

As shown in Figure 8, software quality and productivity were reported as being strongly influenced by project management methodology, trust, conflict resolution, and the diversity of skills, backgrounds, and problem-solving approaches within teams. In contrast, challenges related to collaboration hindrances were not found to have a definitive impact on productivity or software quality. Certain aspects of team dynamics appeared to exert only limited or negligible effects on these outcomes. These findings suggest that, while team dynamics and development practices play a critical role, additional factors beyond these dimensions also contribute to variations in software quality and productivity.

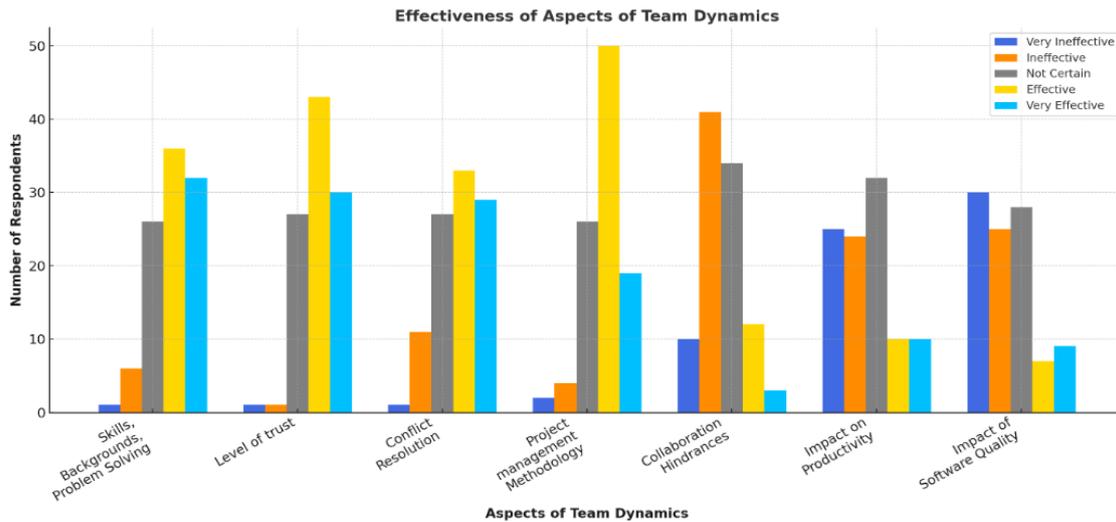


Figure 8. Software quality vs productivity trade-offs

4.4.5. Software development methods and collaboration

Figure 9 shows that, in the realm of software development in South Africa, Agile methodologies dominate with 70%, reflecting their prominence in modern development. Smaller proportions reported using Waterfall, DevOps, or hybrid methods. Moreover, collaboration practices were diverse: 31% used collaboration mainly for brainstorming, 13% for knowledge sharing, 17% for code reviews, 12% for pair programming, 9% for debugging and stand-ups, and 10% reported no collaboration. These findings reveal a varied but pragmatic approach to collaboration, often tailored to specific project needs.

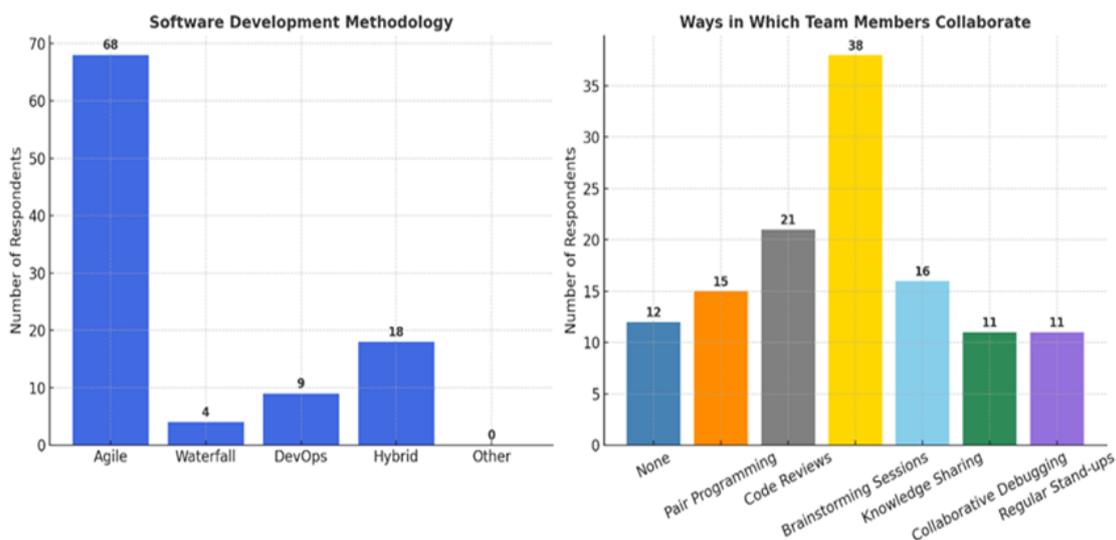


Figure 9. Development methodologies and practices

4.4.6. Collaboration and technical debt

Software development in South Africa is gaining momentum in the same way as developed nations are growing. The analysis revealed that software development tools Trello (32%) and Jira (31%) were the most widely used collaboration tools, followed by Confluence (20%), while Miro and GitHub accounted for only 2% each. Likewise, 10% of respondents reported not using any collaboration tools. In terms of technical debt, more than 40% addressed it occasionally, 27% regularly, and 20% rarely, with fewer than 10% never addressing it. This variability shows that while tools support structured collaboration, the extent to which teams systematically address technical debt remains inconsistent.

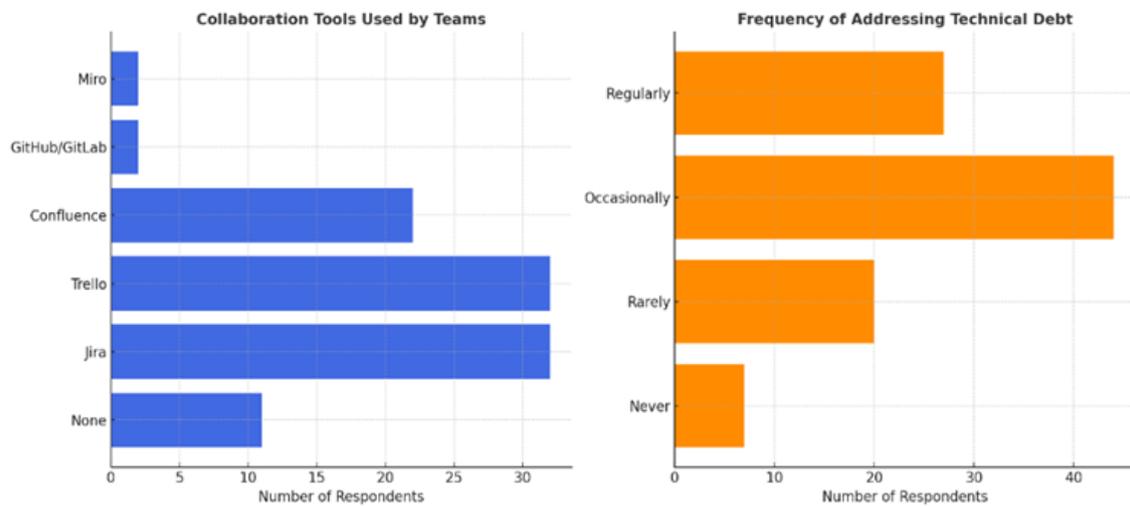


Figure 10. Collaborations and technical depth

4.5. Qualitative and quantitative analysis

The quantitative analysis based on the MLR, as shown as presented in Table 2, revealed that trust among team members was the most influential factor in determining software quality and productivity. Teams with higher levels of trust consistently achieved better outcomes, underscoring the importance of PS and mutual reliability. Communication effectiveness and the use of collaboration tools also contributed positively to project performance, while unstructured or excessive collaboration and overly frequent planning meetings were associated with reduced efficiency. These findings highlight that the quality and structure of interactions within a team matter more than the mere frequency or volume of collaborative activities. In the same vein, Spearman's Rho correlation further reinforced these insights, showing that team diversity strengthened conflict resolution,

productivity, and overall efficiency. Effective conflict management and coordinated collaboration emerged as key enablers of successful software outcomes. Conversely, poorly structured communication, collaboration challenges, and inappropriate methodology choices were linked to lower performance. Qualitative findings supported this pattern, with participants describing delays, rework, and reduced quality arising from unclear communication, low trust, and dysfunctional collaboration. Together, these results demonstrate that software outcomes depend not only on technical practices but also on the social and relational dynamics of the team.

Table 2. MLR results

Regression Model Parameters			Unstandardized Coefficients		Standardized Coefficients	t	Sig.
Model	Variable	Description	B	Std. Error	Beta		
20	(Constant)		1.704	.510		3.340	.001
X ₁	JOBR	Current Job Roles	.039	.027	.104	1.447	.015
X ₂	ORGSZ	Size of Organisation	.050	.043	.085	1.160	.025
X ₃	COMCH	Communication Channels	-.188	.082	-.177	-2.295	.024
X ₄	COMEFF	Communication Effectiveness	.175	.083	.162	2.120	.036
X ₅	TMTRST	Team members' Level of trust	.376	.085	.362	4.407	.000
X ₆	COLLB	Collaboration in Software Development	-1.218	.315	-.426	-3.871	.000
X ₇	COTOOL	Collaboration tools	.180	.074	.227	2.431	.017
X ₈	PLNMTG	Meetings Planning	-.131	.055	-.173	-2.390	.019
X ₉	TMCOLB	Collaboration Rating	.145	.053	.325	2.733	.007

a. Dependent Variable: The effectiveness of project management methodology in delivering software outcomes.

Source: Data Analysis, 2025

The qualitative findings reflected these quantitative patterns. As shown in Table 3, communication challenges, reported by over half the respondents, included unclear requirements, poor information flow, and role misalignment, often resulting in delays and rework. Low trust led to disengagement, reluctance to raise concerns, and duplicated effort, underscoring the importance of PS. Dysfunctional collaboration, such as siloed work and unilateral decision-making, further hindered team cohesion and output quality. Together, these results demonstrate that software outcomes depend not only on technical practices but critically on team dynamics, particularly trust, communication effectiveness, and well-structured collaboration.

Table 3. Summary of qualitative analysis

Aspect	Quotes	Quantitative Findings
Communication	<p>"Communication was not clear during the project phase, resulting in delayed delivery and rework."</p> <p>"Team requirements communication from business analysts was inadequate, impacting the final product."</p> <p>"Updating the code without notifying everyone impacted version control and caused bugs."</p>	<p>Negative correlation between communication frequency and software quality ($\rho = -0.258, p < 0.01$). Negative correlation between communication challenges and productivity ($\rho = -0.329, p < 0.01$).</p> <p>Positive correlation between communication effectiveness and project efficiency ($\rho = 0.388, p < 0.01$).</p>
Trust	<p>"A development team's lack of trust and communication led to inefficiencies, as no one felt safe enough to raise blockers or concerns." "People felt hesitant to speak up due to fear of criticism, which delayed important decisions." "Trust issues led to redundant testing as team members doubted each other's work."</p>	<p>Strong positive correlation between trust and project efficiency ($\rho = 0.493, p < 0.01$). Trust was the strongest positive predictor ($B = 0.376, \beta = 0.362, p < 0.001$).</p>
Collaboration	<p>"In one project, we were developing a web-based system, but struggled because everyone worked in silos." "Taking decisions alone without involving others severely impacted the cohesion of our solution." "Collaboration was poor; no brainstorming sessions were conducted, and decisions were rushed individually."</p>	<p>Collaboration was negatively associated with outcomes ($B = -1.218, \beta = -0.426, p < 0.001$). Positive correlation between collaboration hindrances and reduced software quality/productivity ($\rho = 0.322, p < 0.01$).</p>

4.6. Discussion

The descriptive findings outlined the demographic and organisational profile of participants. Most were software developers (62.1%), predominantly male (64%), working in small teams of fewer than ten members despite being part of larger organisations. This reflects patterns in the South African software industry, where women account for less than 24% of the workforce, and aligns with the JCSE-IITPSA [32] survey showing developer shortages. Agile methodologies dominated (70%), consistent with global shifts toward adaptive approaches. Although Microsoft Teams, Slack, and Zoom supported interaction, challenges such as lack of time (43%) and clarity (37%) indicate that tool availability does not guarantee communication quality. Similar issues were reported by Ruvimova et al. [31], who found that meetings and collaboration dynamics variably affect productivity.

The regression results showed trust ($B = 0.376$, $\beta = 0.362$, $p < 0.001$) as the strongest predictor of software outcomes, supporting Strode et al. [1] and Alami et al. [12] on the centrality of trust in agile teams. Communication effectiveness ($B = 0.175$, $p = 0.036$) enhanced outcomes, while multiple communication channels ($B = -0.188$, $p = 0.024$) reduced them, echoing Restrepo-Tamayo et al. [19] on the importance of structured interaction. Unexpectedly, collaboration was negatively associated with outcomes ($B = -1.218$, $p < 0.001$), suggesting coordination overhead when poorly managed. This contrasts with Korkmaz and Aydin [4], who linked high collaboration to better performance, highlighting that outcomes depend on workflow clarity. Collaboration tools ($B = 0.180$, $p = 0.017$) improved results, supporting Berrezueta-Guzman et al. [29] on the role of tool integration. Planning meetings ($B = -0.131$, $p = 0.019$) also reduced outcomes, consistent with Ruvimova et al. [31]. Correlations strengthened these findings. Diversity correlated positively with conflict resolution ($\rho = 0.499$), productivity ($\rho = 0.240$), and efficiency ($\rho = 0.327$). Conflict resolution correlated with productivity ($\rho = 0.276$) and efficiency ($\rho = 0.336$), consistent with Berrezueta-Guzman et al. [29]. Trust correlated strongly with efficiency ($\rho = 0.493$), and communication effectiveness ($\rho = 0.388$) supported Barros et al. [11] on the role of PS. By contrast, communication frequency ($\rho = -0.258$) and methodology choice ($\rho = -0.293$) were negatively related to quality, reflecting the risks of fragmented communication and misaligned practices, as also noted by Ruvimova et al. [31] and Bandaru [10].

Furthermore, the qualitative themes, as shown in Table 3, explained these statistical patterns. Communication issues were linked to unclear requirements and fragmented information flows, supporting Restrepo-Tamayo et al. [19]. Trust issues reflected disengagement and reluctance to share knowledge, aligning with Strode et al. [1] and Alami et al. [12]. Ineffective collaboration was described as siloed work and unilateral decisions, supporting the regression finding of negative effects and echoing practices identified by Kortum et al. [18].

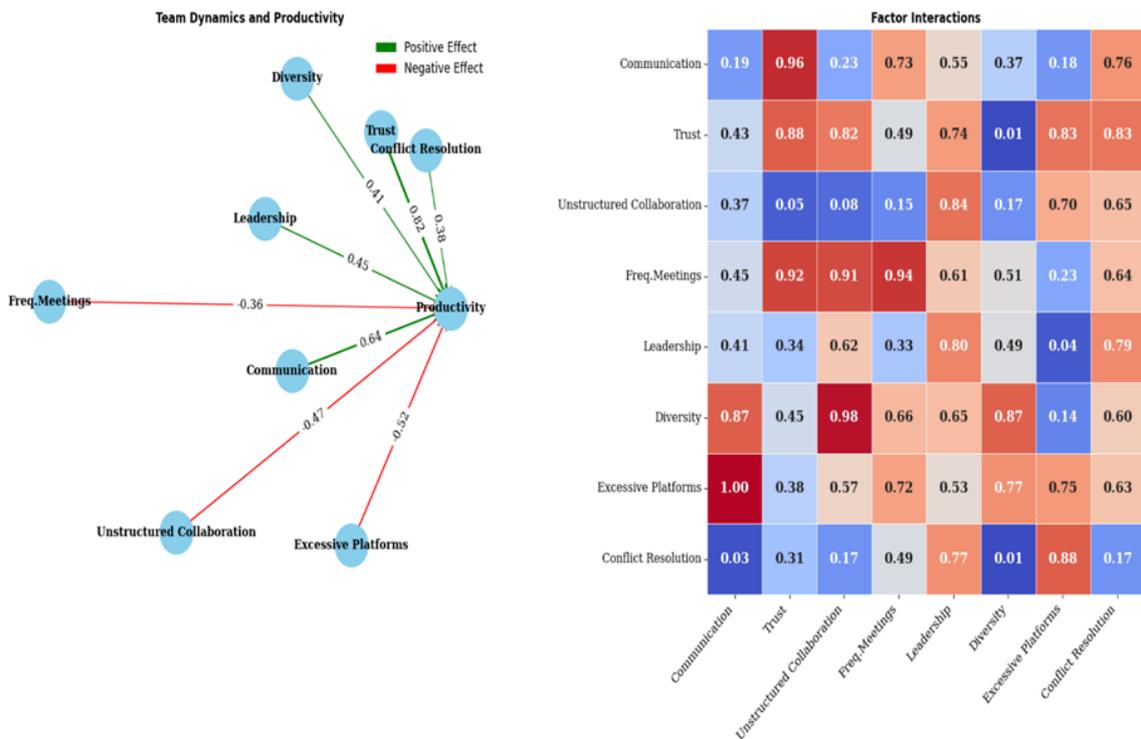


Figure 11. Team dynamics and productivity relationships

Table 4. Comparison with existing studies

Study/Year	Setting / Sample	Methods	Main Constructs	Main Findings	Focus	Limitations / Contribution
[1] 2023	Co-located agile teams, single product	Literature review, focus groups, case studies	Shared leadership, peer feedback, adaptability, team orientation, mental models, trust, communication	Identifies behavioural markers linking agile practices to team effectiveness	Team dynamics: indirect links to productivity & quality	Limited to co-located, non-safety-critical teams
[10] 2024	Scrum teams in one enterprise	Observation, metrics, poll	Team size, climate, diversity, leadership, support	Smaller teams more effective; communication	Productivity, quality	Limited to one enterprise; small sample

				issues in larger teams		
[11] 2024	Agile teams in Portugal	Survey + PLS analysis	Team capability, customer involvement, PS, autonomy	Capability & involvement drive success; autonomy moderates effects	ASD project success	Limited generalisability; survey-based
[24] 2024	25 GitHub communities (n=140)	Statistical modelling, survey	Cultural & geographical dispersion, socio-technical factors	Dispersion affects productivity; communication mitigates risks	Productivity	Limited dataset; productivity proxy
[31] 2022	Agile teams in NA & Europe (25 devs, 5 teams)	Daily surveys, interviews	Individual vs. team productivity, collaboration, meetings	Individual productivity predicts team productivity; remote work complicates assessment	Productivity	Small sample; self-reported
[4] 2025	104 DevOps professionals worldwide	Survey + statistical analysis	Team formation type, collaboration, performance metrics	High-collab DevOps teams outperform others	Lead time, deployment, recovery	Self-reported; cross-sectional
[19] 2024	Software teams, industry professionals	Mapping + survey	Communication, leadership, teamwork	Human factors reinforce productivity & quality; weak measurement practices	Productivity, quality	Measurement validity issues
This Study (2025)	South African software professionals (n=124)	Survey + regression + correlation + qualitative analysis	Team dynamics (trust, communication, leadership, conflict resolution), software development practices (methodology, tools), demographics	Trust strongest predictor; communication effectiveness positive; excessive platforms, unstructured collaboration, frequent meetings negative; diversity and conflict resolution positive	Productivity, quality	Combines quantitative and qualitative evidence, confirming trust, safety, and communication quality as enablers while revealing negative effects of over-collaboration and misapplied methods

Figure 11 illustrates these relationships, while Table 4 situates them within prior studies. Trust consistently emerged as the strongest enabler, alongside communication effectiveness and diversity, while excessive collaboration, uncoordinated tools, and poorly structured meetings acted as barriers [1], [4], [10], [11], [19], [24], [31]. Demographic factors moderated outcomes, and leadership style and methodology choice showed context-dependent effects. These findings extend prior research by demonstrating that technical processes and social dynamics are interdependent. Trust, PS, communication effectiveness, diversity, and conflict resolution were enablers, while excessive communication, unstructured collaboration, and misapplied methodologies undermined outcomes. Sustained improvement in software productivity and quality requires optimising both dimensions together.

4.6.1. Validity Threats

In this study, several potential validity threats may affect interpretation. The sample, though diverse in roles and organisational settings, was limited to South African software professionals and may not fully represent other regions or industries, which could restrict its external validity. Practices like self-reported survey responses could introduce possible social desirability and recall biases, particularly regarding communication practices, trust levels, and collaboration behaviours. Thus, the cross-sectional design prevents causal inferences, since the observed relationships represent associations at a single point in time. Measurement validity could also be affected by using Likert scales and constructed indices for complex constructs such as team dynamics and software outcomes, despite expert review and pilot testing. Additionally, while qualitative data added contextual depth, interpretation may be influenced by the researcher's subjectivity, even though systematic coding was applied to mitigate bias. However, despite these limitations, the study was conducted carefully, and no limitation undermines the validity of the findings.

4.6.2. Proposed Framework

This section presents the proposed conceptual Socio-Technical Best Practices Framework (ST-BPF), which integrates team dynamics (social subsystem) and software development practices (technical subsystem) to improve software quality and

productivity. Grounded in Socio-Technical Systems (STS) theory [32], the framework guides teams and organisations by translating empirical insights into a structured model that emphasises the alignment of social and technical dimensions for joint optimisation. The ST-BPF comprises four progressive layers: Input, Subsystems (Social and Technical), Interaction, and Output.

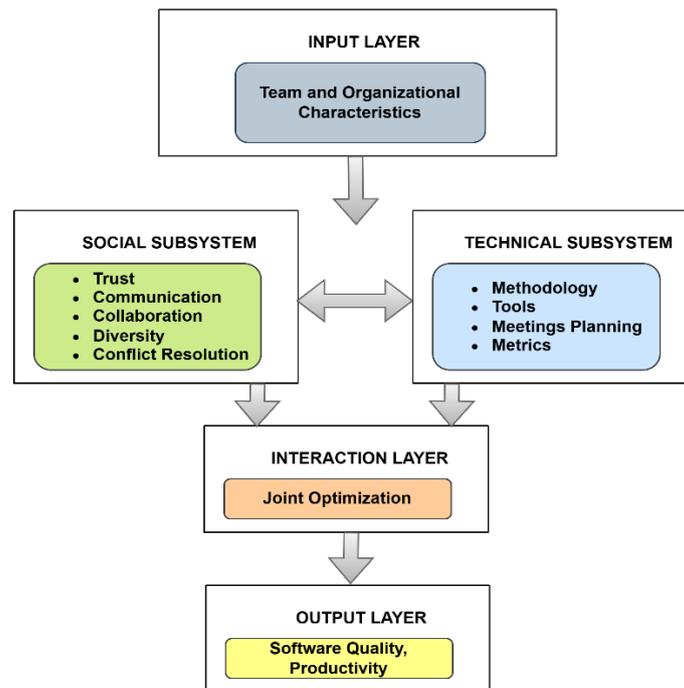


Figure 12. ST-BPF Framework

1) Input

This layer establishes foundational conditions that influence both social and technical aspects of software development. Important factors include team, organisational culture, leadership structure, and project complexity. These elements shape the environment in which teams operate, defining constraints and opportunities that affect the application of social and technical measures without directly determining outcomes.

2) Social subsystem

This layer captures human and relational factors central to team effectiveness: trust, communication, collaboration, diversity, and conflict resolution.

- a) Trust refers to mutual reliability and PS.
- b) Communication involves clarity, timeliness, and bidirectional information flow.

- c) Collaboration entails cooperative effort, shared responsibility, and knowledge exchange.
- d) Diversity reflects the variety of perspectives and skills that enhance creativity and problem-solving.
- e) Conflict resolution addresses disagreements constructively, reducing disruption and sustaining cohesion.

3) Technical subsystem

The Technical Subsystem represents the structured processes and tools that enable development activities, including methodologies, tools, planning meetings, and performance metrics.

- a) Methodologies indicate the overarching development approach and its adaptability to team and project needs.
- b) Tools are technological enablers supporting development, collaboration, and delivery.
- c) Planning meetings provides formal coordination and alignment opportunities.
- d) Metrics serve as indicators to monitor progress, quality, and productivity.

The bi-directional relationship between the Social and Technical Subsystems, illustrated in Figure 12, highlights their interdependence: changes in one subsystem influence the other, reinforcing the need for alignment.

4) Interaction

This layer operationalises the principle of joint optimisation, actively managing the alignment between social and technical factors. This ensures that processes and tools meet the needs while team dynamics support effective technical practice. Achieving this alignment requires continuous reflection and adjustment in response to outcomes and project settings.

5) Output

The layer defines the ultimate objectives of the framework: software quality and productivity.

- a) Software quality includes dimensions such as reliability, maintainability, defect density, and user satisfaction.

- b) Productivity refers to the efficiency and effectiveness of delivering working software on time, within budget, and with optimal resource use.

Both outcomes are shaped by the interaction of the subsystems. Cohesive teams with strong trust and collaboration can identify and resolve issues early, improving quality, while well-designed processes and tools reduce waste and accelerate delivery. Moreover, optimal results arise when both subsystems are strong and jointly optimised; weaknesses in one undermine the other and compromise outcomes.

6) Practical application

The proposed ST-BPF serves as a diagnostic and improvement tool. Teams experiencing quality or productivity challenges can trace potential causes across both subsystems, identifying misalignments such as unclear communication channels or inadequate defect-tracking tools. This dual perspective supports integrated interventions, combining technical improvements with social initiatives. The framework is intentionally adaptable, allowing teams to tailor methodologies, tools, or practices to their context while maintaining alignment between social and technical efforts.

Table 5. Framework applicability

Step	Objective	Main Activities
1. Assess Contextual Inputs	Define boundary conditions for analysis.	Identify team size, culture, leadership, and project complexity.
2. Evaluate Social and Technical Subsystems	Identify strengths and weaknesses in each subsystem.	Assess trust, communication, collaboration, methodology fit, and tool usage.
3. Diagnose Misalignments in the Interaction Layer	Analyse the interface between subsystems to uncover misalignments.	Compare social and technical findings to identify mismatches.
4. Implement Integrated Improvements	Apply joint optimisation interventions.	Pair technical adjustments with social initiatives.

5. CONCLUSION

This paper presented a study on how team dynamics and software development practices interact to influence software quality and productivity. An empirical investigation was conducted through an online survey of software development professionals in South Africa. The results show that software outcomes are significantly influenced by the alignment of social and technical factors. Particularly, quantitative analyses revealed that trust and effective communication consistently enhanced productivity and efficiency, while excessive or poorly structured collaboration and communication channels introduced coordination overhead and fragmentation. Thematic analysis strengthened these findings, indicating that unclear requirements, low trust, and dysfunctional collaboration directly hinder software delivery. These results confirm that software development success arises not solely from technical practices but from the interaction between social and technical subsystems. The most critical aspects of team dynamics identified were trust, communication, diversity, and conflict resolution. Trust emerged as the strongest positive predictor of efficiency and quality, while effective communication improved coordination and shortened delivery cycles. Diversity of skills and backgrounds supported conflict resolution and productivity, whereas unstructured collaboration and unresolved conflicts were associated with reduced performance. These findings demonstrate that constructive team interactions are essential for maximising the benefits of software development practices. Based on these empirical insights, the study developed a conceptual ST-BPF combining important social and technical factors into a model to improve software quality and productivity.

This study has some limitations. The sampling was limited to software professionals in South Africa, which may affect generalizability to other cultural or organisational settings. In addition, reliance on self-reported survey data introduces the possibility of response bias. Also, the proposed ST-BPF has not been validated to assess its effectiveness in the software development ecosystems. Future research could focus on longitudinal studies to track how team dynamics and practices evolve over project phases, particularly in the context of AI integration, and conduct comparative studies across different organisational and cultural settings to enhance generalizability. Additionally, the ST-BPF will be validated in a real-world software development setting to evaluate its strengths and limitations in supporting productivity and software quality.

In general, effective software development requires managing human dynamics with the same thoroughness and intentionality as technological components.

ACKNOWLEDGMENTS

This research was supported by the UDSC, the Department of Computer Science at the North-West University Mafikeng campus, and the Council for Scientific and Industrial Research (CSIR), Pretoria, South Africa.

REFERENCES

- [1] D. Strode, T. Dingsøy, and Y. Lindsjorn, "A teamwork effectiveness model for agile software development," *Empir. Softw. Eng.*, vol. 27, Art. no. 56, Mar. 2022. doi: 10.1007/s10664-021-10115-0.
- [2] S. Tyagi, R. Sibal, and B. Suri, "Empirically developed framework for building trust in distributed agile teams," *Inf. Softw. Technol.*, vol. 145, May 2022, Art. no. 106828. doi: 10.1016/j.infsof.2022.106828.
- [3] L. Machuca-Villegas, G. P. Gasca-Hurtado, S. M. Puente, and L. M. R. Tamayo, "Perceptions of the human and social factors that influence the productivity of software development teams in Colombia: A statistical analysis," *J. Syst. Softw.*, vol. 192, Oct. 2022, Art. no. 111408. doi: 10.1016/j.jss.2022.111408.
- [4] H. E. Korkmaz and M. N. Aydin, "An empirical study on performance comparisons of different types of DevOps team formations," *Front. Comput. Sci.*, vol. 7, Art. no. 1554299, Sep. 2025. doi: 10.3389/fcomp.2025.1554299.
- [5] N. Azad and S. Hyrynsalmi, "DevOps critical success factors – A systematic literature review," *Inf. Softw. Technol.*, vol. 157, May 2023, Art. no. 107150. doi: 10.1016/j.infsof.2023.107150.
- [6] S. Pal, A. Nair, and Z. Zuo, "Collaborative dynamics in open source software development: Unveiling the influence of team interaction and the role of project manager," *J. Oper. Manag.*, vol. 70, no. 7, pp. 1076–1099, Oct. 2024. doi: 10.1002/joom.1324.

- [7] S. Vijayaraghavan, J. Song, T. Guan, S. Choi, and S. Kulkarni, "Influence of Communication Among Shared Developers on the Productivity of Open Source Software Projects," arXiv preprint arXiv:2203.15459, Mar. 2022.
- [8] E. Scott, K. N. Charkie, and D. Pfahl, "Productivity, Turnover, and Team Stability of Agile Teams in Open-Source Software Projects," in Proc. 46th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA), IEEE, Aug. 2020, pp. 124–131. doi: 10.1109/SEAA51224.2020.00029.
- [9] K. Constantino, S. Zhou, M. Souza, E. Figueiredo, and C. Kästner, "Understanding Collaborative Software Development: An Interview Study," 2020, doi: 10.1145/3372787. (*Catatan: entri awal hanya mencantumkan DOI pendek; periksa proceedings lengkap jika perlu.*)
- [10] S. P. Bandaru, "Agile Methodologies in Software Development: Increasing Team Productivity," ISARJST, 2024.
- [11] L. Barros, C. Tam, and J. Varajão, "Agile software development projects – Unveiling the human-related critical success factors," *Inf. Softw. Technol.*, vol. 170, Jun. 2024, Art. no. 107432. doi: 10.1016/j.infsof.2024.107432.
- [12] A. Alami, M. Zahedi, and O. Krancher, "Antecedents of psychological safety in agile software development teams," *Inf. Softw. Technol.*, vol. 162, Oct. 2023, Art. no. 107267. doi: 10.1016/j.infsof.2023.107267.
- [13] A. Rezvani and P. Khosravi, "Emotional intelligence: The key to mitigating stress and fostering trust among software developers working on information system projects," *Int. J. Inf. Manage.*, vol. 48, pp. 139–150, Oct. 2019. doi: 10.1016/j.ijinfomgt.2019.02.007.
- [14] M. Herrmann, M. Obaidi, and J. Klünder, "Different and similar perceptions of communication among software developers," *Inf. Softw. Technol.*, vol. 181, May 2025, Art. no. 107698. doi: 10.1016/j.infsof.2025.107698.
- [15] Md. S. Rahman, A. Kumar Saha, U. Chakraborty, H. T. Sujana, and S. M. A. Shafi, "Evaluating the impact of Test-Driven Development on Software Quality Enhancement," *Int. J. Math. Sci. Comput.*, vol. 10, no. 3, pp. 51–76, Sep. 2024. doi: 10.5815/ijmsc.2024.03.05.
- [16] W. Ren and S. Barrett, "Test-driven development, engagement in activity, and maintainability: An empirical study," *IET Softw.*, vol. 17, no. 4, pp. 509–525, Aug. 2023. doi: 10.1049/sfw2.12135.

- [17] S. Parsa, M. Zakeri-Nasrabadi, and B. Turhan, "Testability-driven development: An improvement to the TDD efficiency," *Comput. Stand. Interfaces*, vol. 91, Jan. 2025, Art. no. 103877. doi: 10.1016/j.csi.2024.103877.
- [18] F. Kortum, J. Klünder, and K. Schneider, "Behavior-driven dynamics in agile development: The effect of fast feedback on teams," in *Proc. 2019 IEEE/ACM Int. Conf. Softw. Syst. Processes (ICSSP 2019)*, May 2019, pp. 34–43. doi: 10.1109/ICSSP.2019.00015.
- [19] L. M. Restrepo-Tamayo, G. P. Gasca-Hurtado, and J. Valencia-Calvo, "Characterising social and human factors in software development team productivity: A system dynamics approach," *IEEE Access*, vol. 12, pp. 59739–59755, 2024. doi: 10.1109/ACCESS.2024.3388505.
- [20] A. B. Farid, Y. M. Helmy, and M. M. Bahloul, "Enhancing Lean Software Development by using DevOps Practices," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 7, 2017.
- [21] E. Wassouf-Jr, P. Fukuda, and A. Fontao, "Investigating the Developer Experience of LGBTQIAPN+ People in Agile Teams," in *Proc. Int. Conf. Softw. Eng. (ICSE)*, IEEE Computer Society, 2025, pp. 43–54. doi: 10.1109/ICSE-SEIS66351.2025.00010.
- [22] L. Betti, L. Gallo, J. Wachs, and F. Battiston, "The dynamics of leadership and success in software development teams," *Nat. Commun.*, vol. 16, Art. no. 3956, 2025.
- [23] E. Klotins et al., "Use of agile practices in start-up companies," *E-Informatica Softw. Eng. J.*, vol. 15, no. 1, pp. 47–64, 2021. doi: 10.37190/e-Inf210103.
- [24] S. Lambiase et al., "An empirical investigation into the influence of software communities' cultural and geographical dispersion on productivity," *J. Syst. Softw.*, vol. 208, Feb. 2024, Art. no. 111878. doi: 10.1016/j.jss.2023.111878.
- [25] J. Prior and J. Leaney, "Software quality and its entanglements in practice," *Ethnogr. Pract. Ind. Conf. Proc.*, vol. 2020, no. 1, pp. 163–176, Oct. 2020. doi: 10.1111/epic.12024.
- [26] M. C. Teffo, K. Sigama, and F. Kanobe, "A contextualised model of the use of agile technique in South African software development teams," *S. Afr. J. Inf. Manage.*, vol. 25, no. 1, Apr. 2023. doi: 10.4102/sajim.v25i1.1617.
- [27] I. G. Ndukwe, S. A. Licorish, A. Tahir, and S. G. MacDonell, "How have views on software quality differed over time? Research and practice viewpoints," *J. Syst. Softw.*, vol. 195, Jan. 2023, Art. no. 111524. doi: 10.1016/j.jss.2022.111524.
- [28] S. A. Licorish, "Understanding the effect of agile practice quality on software product quality," *IEEE Trans. Softw. Eng.*, vol. 51, no. 2, pp. 650–662, 2025. doi: 10.1109/TSE.2025.3532502.

- [29] S. Berrezueta-Guzman, I. Parmacli, M. K. Habib, S. Krusche, and S. Wagner, "Assessing teamwork dynamics in software development projects," in Proc. 2025 IEEE Global Eng. Educ. Conf. (EDUCON), Apr. 2025, pp. 1–5, IEEE. doi: 10.1109/EDUCON62633.2025.11016305.
- [30] Y. Lindsjørn, D. I. K. Sjøberg, T. Dingsøyr, G. R. Bergersen, and T. Dybå, "Teamwork quality and project success in software development: A survey of agile development teams," *J. Syst. Softw.*, vol. 122, pp. 274–286, Dec. 2016. doi: 10.1016/j.jss.2016.09.028.
- [31] A. Ruvimova et al., "An exploratory study of productivity perceptions in software teams," in Proc. Int. Conf. Softw. Eng. (ICSE), IEEE Computer Society, 2022, pp. 99–111. doi: 10.1145/3510003.3510081.
- [32] H. Abbas and G. Katina, "Socio-technical theory," *Trist & Bamforth*, vol. 1, no. 2, pp. 1–16, 2023.