IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 February 2024, accepted 25 March 2024, date of publication 15 April 2024, date of current version 3 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3388505

RESEARCH ARTICLE

Characterizing Social and Human Factors in Software Development Team Productivity: A System Dynamics Approach

LUZ MARCELA RESTREPO-TAMAYO[1], GLORIA PIEDAD GASCA-HURTADO[1], AND JOHNNY VALENCIA-CALVO[2]

[1]Facultad de Ingenierías, Universidad de Medellín, Medellín 50026, Colombia
[2]Departamento de Ciencias Naturales y Tecnología, Universidad de Aysén, Coyhaique, Aysén 5950000, Chile

Corresponding author: Johnny Valencia Calvo (johnny.valencia@uaysen.cl)

This work was supported by Universidad de Aysén, Chile, with funding from institutional strengthening projects of Chilean National Ministry of Education under Project URY21991.

ABSTRACT Software development projects demand high levels of interaction between work team members. This way, management and decision-making must be supported by analyzing the complex dynamics generated through individual interactions to complete the projects. This complexity can be addressed using system dynamics. This modeling approach studies how the structures and relationships between variables in a system interact to generate behaviors over time. It is used to understand and analyze complex systems and make informed decisions. The first step in modeling is articulating the problem. This step defines the key variables that will be included in the model. Still, the lack of a standardized procedure to select, measure, and propose causal relationships is evident. Subjectivity is often appealed to, but this could lead to inaccurate models and biased results. The challenge intensifies when it comes to qualitative variables. This study introduces a formal methodology to characterize such variables, addressing a gap in the existing literature. The use of systematic mapping and a survey-based study is proposed. The methodology is applied to characterize three social and human factors that influence the productivity of software development teams: communication, leadership, and teamwork. The results captured primary experimental research's proven definitions, measurement mechanisms, and causal relationships. This formalized approach not only fills a significant gap in system dynamics but also lays a foundation for expanding its scope to encompass additional variables. As such, it represents a substantial methodological contribution to the field.

INDEX TERMS Complex systems, human factors, performance analysis, productivity, system dynamics, software development management, qualitative variables characterization, interdisciplinary approaches.

I. INTRODUCTION

Effective resource management, primarily focusing on human resources, is critical in the software industry and the digital economy. This necessity arises from recognizing that software development is fundamentally a social process, with its core centered around individuals [1]. It is executed in teams and demands high levels of communication [2]. Moreover, successful project completion is more compromised by a lack of non-technical skills than technical skills [3], leading to low productivity rates.

Productivity depends on technical and non-technical factors and is a critical aspect of software development project management [4], [5]. However, there is a tendency to focus on measuring technical rather than non-technical factors [6], as the latter entail complexities related to their quantification and control [7], [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Antonio Piccinno.

Within the set of non-technical factors, there are social and human factors that influence the productivity of software development teams [9]. Interpersonal social and individual self-management skills characterize the performance of a work team [10]. That makes management more challenging. Therefore, software development project managers demand tools that facilitate the analysis and understanding of these complexities, making informed and effective decisions more straightforward.

One way to approach decision-making in Software Engineering project management is through simulation [11]. Simulation stands as one of the preeminent techniques within the field of operations research [12]. This invaluable tool serves the purpose of mitigating uncertainties inherent to decision-making processes or evaluating transformation strategies [13]. Its utility becomes evident when dealing with situations where precise mathematical formulations for modeling the target system are either absent or impractical due to the unfeasibility of conducting experiments on the system itself [14].

System Dynamics represents a prominent paradigm within the realm of simulation techniques [11]. This approach enhances the capacity to comprehend and model complex systems, allowing for an exploration of the intricate interplay among multiple variables and their temporal evolution [15]. It constitutes a dynamic, deterministic, and continuous simulation framework [16], with its relevance extending into the contemporary context of Industry 4.0 [17].

Dynamic systems are rarely linear and unidirectional [18]; instead, they are intertwined in nets of interdependence that can be difficult to capture [19]. Such inherent complexity necessitates the adoption of a rigorous methodology to steer the modeling endeavor and facilitates a systematic examination of the underlying factors. Documenting it constitutes a best practice in system conceptualization [20].

The initial phase of the modeling process is articulation of the problem; This step is crucial as it lays the foundation for the entire modeling endeavor [21]. The key variables to be included in the model are defined, an aspect in which a group of experts in the subject of interest usually participate [15]. This approach stems from relying on the firsthand experiences and observations of individuals who engage with the system as the main source of data [22].

Given the nature of the data, system complexity, and information sources, the adoption of qualitative methods for problem identification and dynamic hypothesis development is often advantageous [23]. Although these methods demonstrate economic efficiency, the existing literature highlights certain restrictions that include the subjectivity of researchers, the need to accomplish complex data analysis, the potential difficulty in maintaining anonymity, and limitations in terms of generalization of results [24]. Thus, the lack of a robust methodological approach in the initial phase may lead to inaccurate models and decisions based on unreliable projections.

In this context, a methodology that avoids the weaknesses of the suggested methods and addresses the identification and characterization of the essential factors for constructing system dynamics simulation models is needed. For this reason, the methodology should provide a solid framework for exploring and analyzing complex systems. In addition, it must focus on the visible relationships and the subtle connections that can significantly impact the overall system dynamics. Thus, a systematic approach is needed to guide the selection of variables, the definition of relationships and the identification of measurement tools to ensure that the resulting model is true to reality and facilitates decision-making.

Because of the above, the primary objective of this paper is to introduce a formal methodology to streamline the characterization and analysis of the pivotal elements inherent to system dynamics models. In this way, the main contribution of this work focuses on the first step of modeling: problem articulation. This methodology proposes addressing the complexities and challenges that arise when modeling dynamic systems [25], providing researchers and practitioners with a guide to build accurate and applicable simulation models.

In order to illustrate the methodology implementation for problem articulation, the research team presents a case study characterizing social and human factors that are perceived to influence the productivity of software development teams [9]. Of those social and human factors, communication, leadership, and teamwork constitute a set of essential skills in companies in the digital era [26], [27], [28]. Hence, the application focuses on these three factors.

In software development projects, communication overhead is an aspect that has been extensively studied [29], [30], [31]. Some research indicate that the communication overhead can be calculated as a function of the number of individuals in a team [31], [32], [33]. This characterization facilitates its adaptation and use in simulation models in various scenarios. However, this is not a widespread practice when it comes to soft variables such as leadership and teamwork.

The application of system dynamics in Software Engineering acquires relevance, given that historically, it has focused mainly on the management of technical aspects of projects [34], [35], [36], [37], [38], [39], [40]. At the same time, the social components have received limited attention [41]. Therefore, system dynamics provides a priceless tool to analyze and understand the complexities triggered by the interaction between individuals during the software development process, which is essential for the successful completion of projects.

Considering this particular context and aiming for a methodology that facilitates the proper integration of qualitative variables into a system dynamics simulation model, the following research question is formulated: *In what terms can the social and human factors that are perceived as influencing the productivity of software development teams be described?*

Answering this research question contributes to articulating the problem to be analyzed under system dynamics.

The following sections present an overview of research related to this study (Section II), the research methodology (Section III), the results (Section IV), and their discussion (Section V). Finally, conclusions and suggested avenues for future research are outlined (Section VI).

II. RELATED WORK

A. SYSTEM DYNAMICS

System Dynamics is a simulation paradigm proposed by Forrester in the 1950s [42] to make strategic decisions through the design and improvement of policies [12]. It has been applied principally in engineering, business and computer science [43] and even used to model the software development process [11].

The models under this paradigm exhibit a high degree of abstraction [16] and study the relationships between the structure and behavior of a system [44], considering the causal relationships between its elements and how the system assimilates fortuitous events from feedback [45].

The modeling process using System Dynamics is iterative and consists of five steps: (1) problem articulation, (2) dynamic hypothesis formulation, (3) simulation model formulation, (4) testing and (5) policy design and evaluation [15]. The first step identifies the problem, essential variables, time horizon and reference modes [46]. The objective of the second step is the construction of the dynamic hypothesis, which requires the statement of variables and the underlying causal relationships that condition the behavior of the system of interest [47].

The dynamic hypothesis consists of causal diagrams that outline the positive or negative causal relationships between variables [44]. Feedback is vital because it reflects how some variables affect others in a causal loop and can amplify or dampen changes [48]. These feedback loops allow capturing nonlinear dynamics and cumulative effects in systems. Furthermore, including elements such as response times reflect the inherent delays of the processes [19].

The formulation of the simulation model, which corresponds to the third step, uses Forrester or stock and flow diagrams [44]. These diagrams include stocks, flows, and decision functions that control the flow rates between stocks based on the information available from the system [18]. The mathematical modeling employs differential equations to capture the critical aspect of how variables evolve over time [13].

The fourth step involves testing the structure and behavioral validity [49]. Finally, designing scenarios and analyzing the sensitivity are activities of the fifth step, with the corresponding recommendations and support documents [47].

System Dynamics helps analyze the particularities of the software development process [50]. The software development process is inherently human [1] and involves technical, personal and social skills [51]. Furthermore, it is a process executed through projects [29]. In general, projects are complex, and for this reason, the team performance must include both technical and social controls [52]. Under these characteristics, using a simulation model to support the management of these projects entails incorporating qualitative variables.

Some models address strategic problems [53], [54], meeting deadlines [35], [38], [55], the inspection [56], [57], [58] and agile development [34], [36]. These models include mainly technical factors [41]. Some models incorporate non-technical factors such as communication and motivation [31], [59], [60]. However, given that the list of soft skills required in software development is extensive [26], [61], and is underrepresented [62], an open line of work related to their inclusion is evident.

B. METHODS USED IN PROBLEM ARTICULATION

The literature offers numerous instances showcasing the successful utilization of simulation based on system dynamics [63], [64], [65], [66]. Despite this, there is concern about the lack of transparency of the models to represent the complexity of specific systems, such as those related to project management [67]. That is an aspect that must be addressed from the first modeling step.

Generally, the process begins with describing the problem by interacting with the people involved [21]. Reviewing documents, collecting data, conducting interviews, and direct observation are techniques to obtain the required information, with a particular emphasis on establishing reference modes and explicitly defining the temporal scope [15].

The problem definition must be precise and exhaustive [68]. Thus, determining the exogenous, endogenous and excluded variables [69] and their existing relationships are critical aspects related to the problem articulation [70]. It is possible to use literature review [71], content analysis, focus groups, or interviews [23]. These methods have strengths. However, they also have certain limitations.

The literature review provides a comprehensive overview of existing documentation in a specific field and helps in the early stages of the research process [72]. They excel at identifying knowledge gaps and offer a solid theoretical and conceptual foundation for new research efforts [73]. However, literature reviews may introduce selection bias when choosing which studies to include, and the quality of the literature reviewed may vary. Furthermore, it does not provide primary data but a synthesis of previous research [74].

Narrative analysis comprehensively depicts information and context through a participant's unique perspective on a specific topic [75]. However, this method may introduce subjectivity into the interpretation of findings and may be more challenging to quantify and analyze than quantitative research methods. Furthermore, its implementation requires strong researcher and participant communication skills [76].

Interviews have the advantage of collecting detailed information. They allow researchers to explore individual perspectives and experiences, and questions can be resolved in real time [77]. However, this method can be resource-intensive, especially when there are large samples. They may also be subject to interviewer or respondent bias, and maintaining confidentiality and anonymity can present challenges [72].

On the other hand, the "questionnaire" collects data efficiently from a large sample, making it suitable for obtaining quantitative data. This method makes it easy to standardize questions and answers, simplify the summary of answers, and identify trends [77]. However, it is essential to note that responses obtained through questionnaires can sometimes be superficial or biased, as they may need to capture the full depth of participants' perspectives. Furthermore, questionnaires may not provide the opportunity to explore complex issues in depth, and the response rate may sometimes be low, which could affect the representativeness of the data collected [72].

The "focus group" leverages the experience and knowledge of professionals in a particular field [78]. It is valuable in areas with limited prior research, as it can offer insights that are difficult to obtain otherwise. It is a method characterized by being economical and quick to execute [79]. However, challenges associated with selecting experts and their availability may pose difficulties in assembling the group. Furthermore, it may be susceptible to individual bias of participating experts, and not all research topics may have easily identifiable experts [80].

The problem articulation is essential in system dynamics modeling [81], and the literature reports using one or more of those methods. However, a conspicuous gap exists in the form of a standardized methodology for the formal delineation and explication of variables and their interactions. Table 1 presents some studies based on system dynamics where this deficiency is evident.

TABLE 1. Methods for problem articulation.

Method	References
Narrative explanation	[30], [33], [58], [82]–[86]
Literature review	[31], [87]–[89]
Literature review, interview	[32], [35], [90]
Observation, interviews, metrics, literature reviews	[56]
Literature review, interview, industrial data	[57]
Interviews, group meetings, archival data	[37]
Data projects, project documentation	[34]
Expert judgment	[38]
Conceptualization from literature	[91]
Conceptualization from literature, experience - observation of the process	[36]
Literature review, expert opinion survey	[92]
Exploratory data analysis	[60]
Literature review, interview, survey	[93]
Literature review, panel of experts	[94]
Survey	[95]

Exploring which variables to incorporate into the model and how they are causally related to others is a task that should be linked to identifying instruments to measure them. Doing this allows, in the model's practical application, the collection of data to be consistent with the purpose of the analysis. Mainly, when dealing with qualitative variables related to soft skills, it is essential to ensure the reliability and validity of the construct [96], avoiding subjective measurements that affect the relevance of the model's results [97]. Thus, it is pertinent to inquire how to measure them [98]. That is an aspect that could be included in a formal methodology to characterize this type of variable.

III. PROPOSED METHODOLOGY

The experimental research's primary objective is to establish cause-effect relationships with a high level of reliability [78]. This type of study examines how a change in one independent variable influences another dependent variable under controlled conditions [99]. However, conducting experiments to examine all possible causal relationships between the variables of interest involves a process that demands significant resources. Therefore, it is desirable to explore such relationships through information documented in published research, which can be achieved through a literature review [24], [74], [75].

The literature review regarding the investigation of cause-effect relationships between variables can provide insights into how these are measured. Thus, this method addresses two fundamental requirements for the problem articulation [71]. A structured protocol to follow a literature review is the systematic mapping study [100]. However, to avoid bias in selecting articles, it is necessary to validate the applicability and use of the identified measures. In this context, a survey becomes a valuable tool [101].

Consequently, this paper proposes a mixed research approach involving a systematic mapping study and a survey-based study to address the research question comprehensively (Fig. 1). The complementary use of these two methods offers a formal methodology that avoids subjectivities in the problem articulation and, at the same time, strengthens the formulation of dynamic hypotheses.

Systematic mapping is a protocol for obtaining a literature review by identifying, analyzing, and interpreting information from primary studies related to a specific topic [100]. The protocol consists of three phases: 1) planning the review, 2) conducting the review, and finally, 3) writing the report with the analysis and presentation of results [102]. In that way, this study conducts systematic mapping to identify the characteristics inherent in each of the three social and human factors of interest. Fundamentally, the aim is to find definitions of the factors, relationships with other factors, and metrics or instruments to measure them.

Upon obtaining the results of the systematic mapping, the data related to metrics or instruments will go through a validation process through a survey-based study, to establish comparisons with the practices of software development teams in the industry. This method allows obtaining relevant

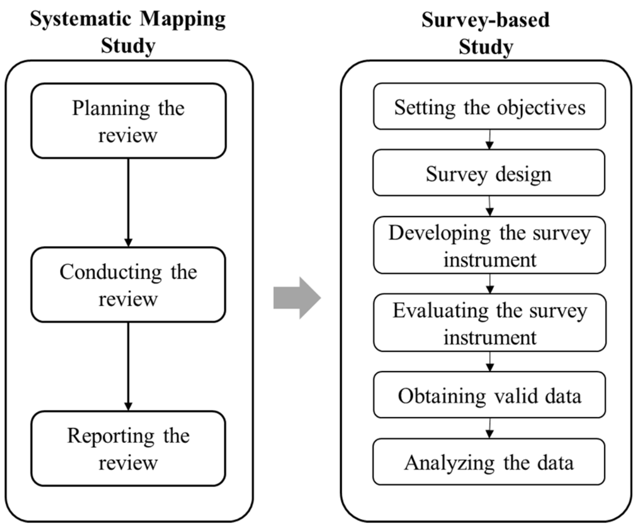


FIGURE 1. Methodology to define key variables in system dynamics models.

information from a population from a sample [103]. Before application, the design requires experts on the subject to validate the questions and analyze their wording. In addition, the final version must have been tested on a pilot sample and adjusted according to identified opportunities for improvement [96].

A. SYSTEMATIC MAPPING STUDY

The first phase is planning the review, which includes defining the research question and specifying the protocol elements according to the purpose of the research. In this case, the objective of the systematic mapping is to summarize the scientific production in Software Engineering that addresses definitions of the factors of interest, relationships with productivity, and metrics or instruments to measure them. Therefore, the research question presented in the Introduction is subdivided into the following questions, which will guide the systematic mapping:

RQ1: What are the definitions reported in the Software Engineering literature of the social and human factors of interest?

RQ2: What are the factors or variables with which the social and human factors of interest are linked, as the scientific literature indicates? Is productivity included?

RQ3: What metrics or instruments for measuring the social and human factors of interest have been mentioned in the Software Engineering literature?

The mapping protocol defined in this study considers the following elements:

- **Specialized databases**: IEEE Xplore, ACM, Science Direct, and Springer since they are electronic sources of relevance in Software Engineering [23].
  The general structure of the search string is *context AND non-technical factor AND (measure? OR metric? OR asses?) AND team AND productivity*. Therefore, the search string for each social and human factor in the context of Software Engineering is as follows.
  Communication: *("software develop?" OR "software engineer?") AND communication AND (measure? OR metric? OR asses?) AND team AND productivity*
  Leadership: *("software develop?" OR "software engineer?") AND leadership AND (measure? OR metric? OR asses?) AND team AND productivity*
  Teamwork: *("software develop?" OR "software engineer?") AND (teamwork OR "team cohesion") AND (measure? OR metric? OR asses?) AND team AND productivity*
  To use the search string in ACM, select the "Anywhere" filter, and the document type is "Research Article". In Science Direct, select "Research articles" and filter "Computer Science" as the subject area. In Springer, select the discipline "Computer Science". Finally, in IEEE, use the 'All metadata' filter.
- **Time**: Studies related to Industry 4.0 started to be published in 2012 [104], [105]. Thus, this mapping includes studies published between 2012 and 2022.
- **Exclusion criteria**: Studies with unavailable files (C1) that are books or book chapters (C2), that are literature reviews (C3), that are not written in English (C4), that are not related to the measurement of factors of interest in Software Engineering (C5), that are not related to Software Engineering (C6), that are repeated (C7) or that are not scientific articles, such as supplements, call to events, business news, interviews, prefaces, among others (C8).
- **Inclusion criteria**: Studies written in English that measure communication, leadership, or teamwork in the Software Engineering context.
- **Classification procedure**: Read each study's title, keywords, abstract, and conclusions captured in each database. Write the title of each article that meets the inclusion criteria in an Excel database. That allows the identification of repeated studies.
- **Quality assessment**: Experts in Software Engineering and external to the research evaluate the quality of the preselected studies with five Yes/No questions. Each 'Yes' answer receives a score of 1, while each 'No' answer receives a score of 0. The questions are QQ1: Is the objective of the study related to the systematic mapping research questions? QQ2: Do the authors specify the context (academic/industry) under which they conducted the study? QQ3: Do the authors specify the study method they followed to obtain the results they present? QQ4: Are the limitations of the study discussed? QQ5: Do the data support the conclusions? Studies with average scores of 4 and 5 move on to data extraction.
- **Data extraction strategy**: Read each study completely to extract general data (title, authors, year of publication, keywords, DOI, and journal), data related to the research methodology (research question or hypothesis, context,

data collection method, size of the sample), and finally, the definition of non-technical factors and their metrics. Store data in a spreadsheet.

- **Synthesis of data extraction**: The synthesis uses a qualitative meta-summary, which includes extracting the findings, editing them, grouping them into common themes, abstracting the findings, and calculating frequencies [106], [107].

The research team and independent experts critically evaluated the mapping protocol to conclude the planning phase. The review by the research team involved conducting a pilot search test in each database to verify the consistency between the results and the study's objective. The independent experts reviewed the congruence between the research question, the search string, the inclusion and exclusion criteria, and the data to be extracted.

Upon concluding the initial phase, "Planning the review," the protocol must be executed as outlined. That encompasses the subsequent phase, "Conducting the review." Ultimately, the third phase involves consolidating the findings and presenting them alongside the respective analyses.

B. SURVEY-BASED STUDY

The research team executed the survey-based study according to Kitchenham and Pfleeger's procedure [108]. The following subsections detail the first five phases of the survey-based study (Fig. 1).

1) SETTING THE OBJECTIVES

The goal of this survey-based study is to get information on the use of the metrics and instruments identified through prior systematic mapping. In addition, acquire other measurement mechanisms that the software industry uses.

Following the above, the hypothesis proposed here is linked to a potential deficiency in the awareness among members of software development teams regarding the measurement of social and human factors within their work teams. These factors are relevant, but it is still necessary to establish a formal link in the management and performance of the team. In this way, this survey constitutes progress in the discussion that must take place within work teams to include these factors in decision-making.

2) SURVEY DESIGN

Following a cross-sectional approach in which a population is analyzed at a specific time, each participant completes the survey only once. Moreover, the decision to employ a self-administered online questionnaire elicits more authentic and thoughtful responses, ensures confidentiality, and provides participants the flexibility to respond in their preferred environments.

3) DEVELOPING THE SURVEY INSTRUMENT

The survey has four sections: 1) general data, 2) questions on communication, 3) questions on leadership, and 4) questions on teamwork. The general data section is comprised of 17 questions inquiring about respondents' location, age, profession, experience, and company and team characteristics (adapted from [109]). Before starting with the general data section, the survey presents the study's objective. The text states that the research team analyzes the responses in an aggregate manner and that it guarantees the anonymity and confidentiality of the responses, with strict adherence to the laws that regulate statistical secrecy and the protection of personal data.

The sections related to the social and human factors contain Yes/No questions, open questions, and multiple-response questions. Each section presents the definition of the factor and asks: Considering the previous definition, is this factor measured in your team? If Yes, it requests to mention the metrics the work team uses to measure it. If the answer is No, it requests to indicate which metrics or instruments identified in the systematic mapping could be used. In this way, the survey includes 26 questions, but everyone must answer 23 (Fig. 2).

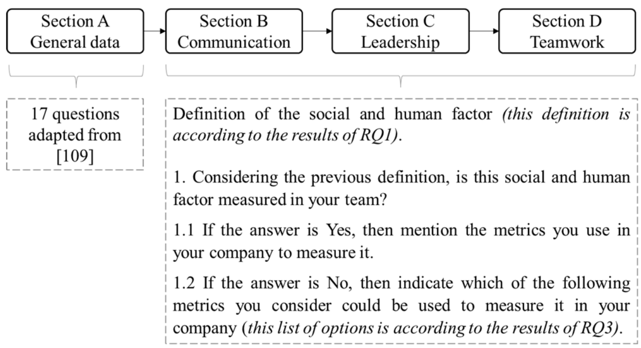


FIGURE 2. Survey structure.

An expert in qualitative research, who is independent of the research project, must review the survey's first version. The assessment encompasses the overall survey design, the alignment of questions with research objectives, the use of neutral and non-suggestive language, the logical organization of questions, the clarity of instructions, and all aspects related to ethics and confidentiality.

4) EVALUATING THE SURVEY INSTRUMENT

The survey is submitted to the judgment of experts in Software Engineering to study the content validity. Each question had to be rated 1 (not relevant and should be eliminated), 2 (relevant but should be modified), or 3 (relevant and should be included). The evaluation format has a space for experts to include comments if necessary. With these data, the team research calculates the Content Validity Coefficient (CVC) of each question, retaining only those with a value greater than 0.7 [110].

Finally, a pilot test with the final version of the survey, in which Software Engineering professionals participate, makes it easier to estimate the average time required to complete it and identify additional aspects to adjust.

5) OBTAINING VALID DATA

The participants are related to the second author; thus, the non-probability convenience sampling approach was used in this study [111]. The researchers selected this approach because the software developer community is diverse and extensive. As a community spanning a diversified range of work backgrounds and specializations, with many collaborating with multiple companies, it is practical and feasible to select participants willing and available to contribute their experiences and perspectives. Non-probability convenience sampling has been used in Software Engineering [108], [112], and provides valuable information about the diversity of roles and contexts in the software development domain despite the inherent limitations of non-probability selection.

Following the procedure of Mendes et al. [113], the second author invited people to share the survey with others who were part of software development teams and who were interested in supporting the research. In this way, the research team also applied non-probabilistic snowball sampling [111]. The use of these two non-probability sampling schemes allows for increasing the number of people surveyed and reduces the bias that could occur when only convenience sampling is used.

IV. RESULTS

A. RESULTS OF THE SYSTEMATIC MAPPING

The execution of the search string in the identified databases yielded an initial capture of 7080 studies. The application of the established inclusion and exclusion criteria and the review of the general elements of each document led to the selection of 40 studies (Fig. 3).

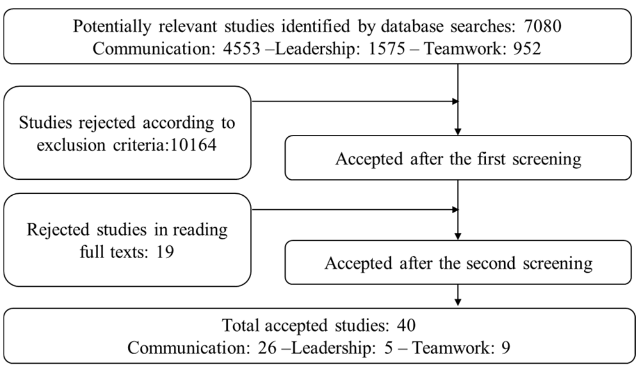


FIGURE 3. Number of captured studies.

Three highly qualified professional experts with postgraduate degrees and working in the academic sector evaluated the quality of the 40 studies. One of them specializes in gamification, and the remaining two have extensive experience leading software development teams. The three selected people know and use the systematic mapping protocol. The average evaluation was within the acceptance range in all cases. As a result, all studies were maintained.

Table 2 shows the number of studies captured per year and factor. Through these results, it is possible to identify that interest in these factors has increased over the last six years.

TABLE 2. Studies per year per factor.

Year	Communication	Leadership	Teamwork	Total
2012	-	1	-	1
2013	-	-	1	1
2015	2	-	-	2
2017	2	-	-	2
2018	3	-	-	3
2019	5	1	1	7
2020	2	1	2	5
2021	6	-	3	9
2022	6	2	2	10
Total	26	5	9	40

Of the 40 included studies, 21 did not detail a specific development methodology. On the other hand, 18 studies were related to agile methodologies. One study was developed under a hybrid methodology. It is important to note that, of the 40 studies, 26 took place in the industrial setting, while 10 were conducted in the academic context; in addition, one of the studies was conducted in both contexts.

According to the typology of research methods by Kothari [72], 24 of the 40 studies are Field Research. In comparison, the remaining 16 are related to Library Research. Field Research used principally interviews and surveys. Concerning Library Research techniques, analysis of historical research records predominates by taking data from repositories such as Jazz, Github, or job portals.

Communication is the social and human factor that has the most studies captured. This result corroborates the importance of this factor for research in Software Engineering since it is closely related to the success of the projects. The limited number of studies on the other two factors represents a research opportunity.

RQ1: What are the definitions reported in the Software Engineering literature of the social and human factors of interest?

The factor definition unifies criteria and facilitates the design of metrics for the management of software development teams. According to this premise, the definitions of each factor reported in the primary studies captured are in Table 3.

These definitions suggest the close relationship that exists between the three social and human factors in the context of Software Engineering. In addition, they reflect their importance at the team level. In this way, those who manage teams can design strategies to promote these factors once they have measures that allow them to identify the team's level.

RQ2. What are the factors or variables with which the social and human factors of interest are linked, as the scientific literature indicates? Is productivity included?

Some primary studies investigated the relationships between the factor and other variables of interest for software development. Table 4 presents the variables that explain each social and human factor. In this case, each factor is considered as a dependent variable.

Teamwork is one of the factors that explains communication, and in turn, communication is one of the factors

TABLE 3. Definitions for each factor.

Social and human factor	Definition
Communication	Communication and collaboration capture how people and teams communicate and work together [114]. Team member's build on each other's ideas and work to develop a joint understanding [115]. Closed-loop communication is defined as 'the exchange of information between a sender and a receiver irrespective of the medium' [116].
Leadership	Ability to influence and guide, motivate and inspire other members of a team, organization or community, including project, business, practice and technology leadership [117]. "Team leadership is defined as the 'Ability to direct and coordinate the activities of other team members, assess team effectiveness, assign tasks, develop team knowledge, skills, and abilities, motivate team members, plan and organize, and establish a positive atmosphere" [116].
Teamwork	Ability to work well with others during conversations, projects, meetings or other collaborations [117]. Is a shared bond that drives team members to stay together and to want to work together [118]. Team cohesion is defined as "the tendency for a group to stick together and remain united in the pursuit of its goals and objectives" [119]. Is a set of interdependent individuals who view themselves as a group and perform a task defined by the organization [120].

TABLE 4. Factor analyzed as a dependent variable.

Social and human factor	Variables on which factor depends
Communication	Motivation [121]. Interaction frequency and team characteristics [122]. Agile enterprise architecture [123]. Distance Geographical, distance organizational, distance psychological and distance cognitive [124]. Team politeness and Team sentiment [125].
Leadership	Idealized behavior, intellectual stimulation, inspirational motivation, individualized consideration [126].
Teamwork	Development approach applied [118]. Team processes and Team cohesion [127]. Interpersonal skills, management skills, heterogeneity, team maturity, experience in the organization, role and goal clarity, and formal leadership [120]. Shared leadership, team orientation, redundancy, adaptability, peer feedback, shared mental models, mutual trust, communication [116].

that explains teamwork. Therefore, these results show the interrelationship that exists between both factors.

On the other hand, motivation is a factor that explains leadership and communication. However, motivation is a factor that is outside the scope of this study and, therefore, constitutes future work.

The social and human factors are independent variables in some primary studies. Considering this, Table 5 presents the variables which each factor explains.

Based on these results, the three factors promote the success of work teams. Therefore, it is relevant for project managers to include them in their management models to analyze and improve team performance.

TABLE 5. Factor analyzed as an independent variable.

Social and human factor	Variables where factor is explanatory
Communication	Productivity [114], [128]. Knowledge diffusion [128], [129]. Job satisfaction [130]. Customer satisfaction [121]. Team performance and Teamwork quality [131]. Software team effectiveness [119]. Project success [132]. Well-being [133]. Agile development performance [123]. Software quality [128], [134]. Effective collaboration [115]. Talent's individual performance [135], developer retention [136]. Teamwork effectiveness [116].
Leadership	Intrinsic motivation [126]. Teamwork effectiveness [116].
Teamwork	Performance [127], [131], [137]. Software team effectiveness [119].

TABLE 6. Metrics / instruments for measuring communication.

Description	Type
Communication density [129]	M
Number of times a day a person communicates by messaging with colleagues. Number of times per day that a person communicates face-to-face with colleagues [130]	M
Number of messages sent through institutional platforms per day [141]	M
Number of comments provided by each developer. Percentage of active or inactive time [136]	M
Number of meetings in a time. Time between meetings [128]	M
Network metrics that show who is connected to whom and how [135]	M
Perceived effort to communicate with another person [124]	M
Flow distance: interviews to understand the communication flow [121], [142]	I
Questionnaire on communication efficiency and effectiveness [123]	I
Teamwork Quality [131], [137]	I

RQ3: What metrics or instruments for measuring the social and human factors of interest have been mentioned in the Software Engineering literature?

The measurement of non-technical factors in Software Engineering is a challenge [138], as evidenced by the limited literature that addresses metrics for such purposes. Table 6 presents the metrics (M) or instruments (I) reported in the literature to measure communication.

In the leadership case, the systematic mapping captured two instruments: MultiFactor Leadership Questionnaire [126], [139], and Questionnaire on leader traits [116], [120].

In the teamwork case, the systematic mapping captured four instruments: Group Environment Questionnaire [118], [140], Questionnaire on the teamwork process [116], [127], Teamwork Quality [131], [137], and Teamwork Process Antecedents [120].

The three factors have instruments that satisfy the characteristics of validity and reliability, and therefore, team managers can use them. At this point, it is necessary to review its availability and interpretability. If there are limitations of this type, an alternative is to use metrics.

Only metrics for communication are evident, and this may be related to the level of investigative maturity that the factor has. In this way, there is an open line of work to identify metrics for the other factors.

The results related to metrics or instruments to measure communication, leadership, and teamwork correspond to the inputs for the survey-based study.

B. RESULTS OF THE SURVEY-BASED STUDY

An expert in qualitative research reviewed the survey's first version, and she suggested minor adjustments. With respect to expert judgment, three people active in Software Engineering research participated in content validity. The CVC of the survey was 0.916, none of the questions had a coefficient less than 0.7, and the wording of some questions was adjusted according to their recommendations.

Nine professionals who work in software development participated in the pilot test. They suggested adjusting the conditionals of the form and specifying that the study context is the software industry.

The survey's final version was hosted on Google Forms to facilitate its dissemination. The second author sent the description of the study's purpose, data processing considerations, and the form link. The survey was available between May 1st and July 31st, 2023.

Seventy-five people responded, 63 of whom reside in Colombia. The remaining 12 individuals reside in various locations: three in Chile, two in Spain, two in Canada, one in the United States, one in Guatemala, one in Mexico, one in Bolivia and one in the United Arab Emirates. Of the respondents, 72% were men, while the remaining 28% were women. The ages of the participants ranged from 21 to 62 years. Regarding their educational level, 26 people indicated having an undergraduate degree, 32 mentioned having completed specialization studies, 13 had a master's degree, and 4 had a doctorate. Regarding development methodologies, 46 participants indicated that they work under agile methodologies, 4 work under traditional methods, 21 use hybrid methodologies and four have developed their own methodologies. It is relevant to highlight that 63 respondents have five or more years of experience in software development.

After completing the survey, 10 of the 75 responded that they measure the three factors in their work teams. Furthermore, 14 people reported measuring one factor, and eight measures two factors. The remaining 43 mentioned not measuring any of the factors. Table 7 details the number of responses related to the people who measure each factor in their work teams.

Results related to measuring communication.

Those who responded that their teams measure communication indicated that they do so by collecting and analyzing data from collaborative platforms. Slack or Microsoft Teams are tools that serve this purpose.

TABLE 7. General results on factor measurement.

Social and human factor	Yes	No
Communication	17 (22.7%)	58 (77.3%)
Leadership	23 (30.7%)	52 (69.3%)
Teamwork	20 (26.7%)	55 (73.3%)

TABLE 8. Preferences of metrics/instruments to measure communication.

Metric/instrument	Frequency
Teamwork Quality	27
Number of meetings in a time	25
Questionnaire on communication efficiency and effectiveness	23
Number of times a day a person communicates by messaging with colleagues	19
Flow distance: interviews to understand the communication flow	17
Number of messages sent through institutional platforms per day	16
Perceived effort to communicate with another person	13
Number of comments provided by each developer	13
Time between meetings	12
Number of times per day that a person communicates face-to-face with colleagues	11
Communication density	9

From an organizational point of view, measurement is done through the number of posts, the number of employees who open and read organizational emails, and the number of employees who participate in meetings, training, and onboarding activities. The results also suggest tracking emails, messages, and calls. Additionally, utilize project log analysis, performance evaluations, and retrospective meetings.

Finally, one response indicated the use of simple questionnaires to understand the perception of each team member's communication. Another response indicated measuring communication through the number of interactions, the average size of interactions, and cross-cutting interactions.

Those who responded that their teams do not measure communication indicated that instruments such as the Teamwork Quality (TWQ) test or the Communication Effectiveness and Efficiency Questionnaire could be used. Metrics such as the number of meetings (face-to-face or virtual) in a time window or the number of times a day that a person communicates by messaging with colleagues can also be considered. The frequency of possible use received by each of the metrics and instruments reported in Table 6 related to communication is detailed in Table 8.

According to this result, there is no significant preference for the use of instruments over the use of metrics. If the team decides to use metrics, it is preferable to use several metrics simultaneously to ensure that all perspectives of the factor are covered.

Results related to measuring leadership.

Of the three factors, leadership had the highest frequency of people reporting, measuring it in software development teams (Table 7). The investigation into the measurement of this factor yielded results that involve both metrics and surveys. The set of metrics includes the number of initiatives a person leads or the number of team members.

Leadership is also analyzed based on the ability to plan and manage tasks, which can be analyzed with the support of platforms such as Asana. Some responses indicated measuring leadership through average team performance and satisfaction with personal projection.

In a complementary manner, some answers were related to analyzing the leader's immersion in the operational and strategic aspects of the business, the evaluation based on the fulfillment of monthly objectives, and the review of the goals achieved compared to those proposed for a given period.

The use of surveys that address relational dynamics and intra-group interactions was reported to determine whether leadership aimed at continuous improvement is exercised and perceived. Also reported were questionnaires aimed at assessing the leader's characteristic traits and annual surveys that capture the level of success and trust in leaders. Companies can define and evaluate their leadership principles, as Amazon does.

Alternative mechanisms to evaluate this factor were also reported, for example, through analyzing the Organizational Climate using the Employee Net Promoter Score (eNPS). These surveys explore happiness in each project or through performance evaluations.

Of a total of 52 respondents who stated that leadership is not evaluated in their work teams, the MultiFactor Leadership Questionnaire received the support of 28 participants for its use. At the same time, the Questionnaire on leader traits obtained 27.

Results related to measuring teamwork. In the area of teamwork, mention is made of an instrument designed explicitly in an organization, which evaluates from the attitude of respect towards others to the clarity in the definition of tasks. It also highlights the possibility of analyzing teamwork through retrospective meetings. These meetings, which are held periodically, allow an in-depth analysis of positive and negative aspects, and identify actions to be implemented in the short term to achieve substantial improvements.

In an additional approach, metrics revolving around several key aspects were reported. These include the quantification of shared interactions, the recording of cross interactions per shared module, the correlation between the number of modules and the number of developers involved, the measurement of the impact of people in the achievement of objectives and results, the evaluation of the volume of projects executed by each work cell, the tracking of deliverables per team and the evaluation of the degree of fulfilment of the team in their tasks and responsibilities.

Concerning the use of instruments that could be used to measure teamwork, 10 of the 55 people who responded that this factor is not measured in their teams agree that it is possible to use any of the four instruments reported. The frequencies according to the responses obtained are presented in Table 9.

TABLE 9. Preferences of metrics/instruments to measure teamwork.

Metric/instrument	Frequency
Questionnaire on the teamwork process	32
Teamwork Process Antecedents	31
Teamwork Quality	30
Group Environment Questionnaire	27

V. DISCUSSION

A. DEFINITION OF FACTORS

Defining variables avoids misunderstandings and confusing interpretations in the modeling process. It ensures that everyone involved in the project understands the variables and avoids including unnecessary or irrelevant variables in the model's objectives [143]. In addition, it contributes to the validation and verification process by starting from a base conceptualization, which in turn facilitates the analysis [12]. In the specific context of Software Engineering, the results in Table 3 can be enriched with the findings of other studies that have conducted literature reviews and provided definitions for various social and human factors [144] or soft skills [145].

B. RELATIONSHIPS BETWEEN FACTORS

Considering the results on the relationships between variables shown in Tables 4 and 5, it becomes clear that both teamwork and leadership influence the productivity of the software development team. Furthermore, it is evident that communication influences team productivity and software quality [128], [134].

Teamwork and communication are intertwined in the context of software development teams [116]. The tasks to be executed in projects are often interdependent and require the collaboration of multiple individuals with diverse skills and roles [146]. Effective communication enables team members to understand their shared responsibilities, goals, and challenges [147]. Constant and open communication facilitates early problem identification and conflict resolution. It also promotes a collaborative work environment where ideas can flow freely, knowledge can be shared, and creative solutions can be explored [148].

Leadership plays a role in fostering intrinsic motivation. That is possible because transformational leadership is associated with motivating, inspiring and stimulating critical thinking [126], which drives creativity [149].

That highlights the intricate net of relationships between communication, leadership, and teamwork in software development. As these connections are explored, it becomes clear that success in this industry is not based solely on technical competence but on the ability of teams to collaborate, communicate effectively, and be inspired by a leadership that fosters creativity. This understanding suggests considering a holistic approach in which these factors are mutually intertwined, and the search for practical solutions in Software Engineering becomes a balance between technical excellence and team dynamics.

C. MEASUREMENT OF FACTORS

Measuring qualitative variables introduces complexities in both research and data analysis. Unlike quantitative variables that are easily quantifiable and directly amenable to statistical analysis through numerical representation, qualitative variables encapsulate characteristics that defy simple numerical quantification [150].

The challenge with qualitative variables lies in their subjective nature, heavily reliant on individual perception and interpretation [151]. The absence of numerical values hinders the use of conventional statistical methods, thereby constraining the capacity to achieve precise and objective measurements. Furthermore, variations in how researchers interpret and code qualitative responses introduce an additional layer of subjectivity to the data analysis process [152]. Addressing these difficulties, therefore, requires specialized qualitative approaches and analysis techniques that consider the richness of qualitative information without attempting to force it into a purely numerical structure.

A broad spectrum of human factors in Software Engineering has generated research interest [145]. The measurement of this type of factor is usually based on surveys and questionnaires. However, these methods pose challenges in obtaining consistent and reliable data [153]. This challenge also permeates simulation under system dynamics.

One of the good practices to follow during simulation model conceptualization is to ensure, in principle, that each variable can be measured [20]. Therefore, the results related to RQ3 support this purpose.

The captured literature offers a set of 10 metrics and three instruments intended to analyze communication in software development teams, as detailed in Table 6. However, the survey reveals that only 22.7% of participants reported implementing these metrics in their work teams, as Table 7 illustrates. By making a comparison between the metrics identified in the systematic mapping and those used by the respondents, a distinctive pattern emerges: the metrics that stand out are those that focus on quantifying the number of interactions through messages [130], [141], comments [136] and face-to-face conversations [141]. Additionally, three respondents also mentioned this approach as a viable alternative for assessing communication in their work context.

According to the results of RQ3, leadership, and teamwork are measured through psychometric instruments. Regarding leadership, the Multifactor Leadership Questionnaire is often used. It is a test widely used in various organizational contexts [154]. However, it is protected by copyright, which may limit its availability at the team level. On the other hand, some of the answers provided by the 23 participants who indicated measuring this factor were related to evaluating leadership through identifying and evaluating traits and qualities considered essential in a leader. Although this approach coincides with a second option reported in Table 6, it is based on individual perceptions, introducing subjectivity into the measurement process. That approach is related to the use of behavioral markers. It is a mechanism adapted in Software Engineering from disciplines such as aviation and medicine, where observable behaviors that affect individual or team performance are identified [155].

The systematic mapping allowed to identify four potential instruments to analyze teamwork. It was considered a viable option to measure it by at least half of the people who do not do it in their work teams.

Psychometric tests are frequently employed to examine such factors, and their utilization tends to be more prevalent at the organizational rather than at the team level, because interpreting their results necessitates the expertise of a professional [156]. In the organizational context, they are usually applied at specific moments, such as the selection process, to assess fit to profile [157]. This situation would justify the results shown in Table 7, where it is evident that, for each factor, more than 69% of those surveyed said they did not measure it.

This finding highlights a scenario in which the assessment and monitoring of these social and human factors has yet to become widespread. For most respondents, including these factors in the management schemes of software development teams should be more evident. While they play a crucial role in the success of software development teams [6], they are often overlooked or underestimated compared to the technical aspects.

Under traditional methodologies, it is possible to track communication and leadership through collaborative tools. However, cascade or sequential models often focus on hierarchical structures and sequential processes [158], which can restrict the breadth and depth of monitoring these factors.

In communication, traditional methodologies often rely on written documentation and regular meetings. While these practices can help keep a formal record of communication, they can overlook aspects such as informal conversations and real-time interactions. Thus, through platforms such as Slack or Microsoft Teams, it is possible to identify message frequency, response times, communication patterns, and recurring themes.

In terms of leadership, traditional methodologies often rely on a rigid, hierarchical leadership structure, making it challenging to assess transformational leadership and team influence accurately. While collaborative tools can provide a record of the leader's actions, they may need to fully capture the quality of leader-collaborator relationships or their ability to inspire and stimulate creativity.

Organizations are adopting agile methodologies and more flexible collaborative tools to address these constraints. These approaches enable more fluid communication, foster innovation by providing room for experimentation and continuous feedback, and value leadership that adapts to the team's changing needs. However, the choice of methodology and collaborative tools must be aligned with the organization's goals and the culture it wishes to promote, recognizing the importance of maintaining a balance between tradition and adapting to an ever-changing business environment.

Under agile methodologies, the retrospective is a propitious space to socialize the problems that arose due to the absence of social and human factors. The retrospective is a regular meeting at the end of each iteration to encourage the team to reflect on their work and constantly search for improvements in their work practices [159]. This process, supported by tools such as Kanban board or Starfish [160], stands out for the entire team's participation, deep reflection and collaborative orientation [161]. Formalizing an instrument to measure social and human factors in the retrospective is a line of future work.

The metrics and instruments to measure non-technical factors like communication, leadership, and teamwork are input to include them in the decision-making process. A manager team can utilize these tools to assess how the team performs in relation to each factor. After that, it is possible to design an intervention strategy to improve it and, consequently, improve the team's productivity. That constitutes a practical implication of the results presented in this study.

D. DYNAMIC HYPOTHESIS

Fig. 4 shows the causal diagram derived from the defined relationships. It consists of a balancing loop called ''B1: productivity and social and human factors'' and a reinforcing loop called ''R1: interaction between team members''.

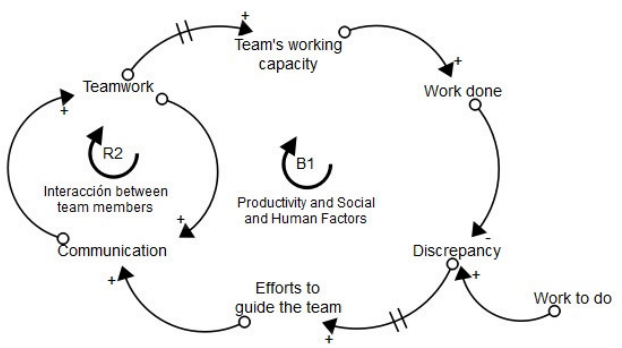


FIGURE 4. Proposed dynamic hypothesis.

The productivity cycle (B1) is aligned with the goal-seeking archetype, in which an attempt is made to bring the system toward an ideal state [48]. Estimating a software development project includes estimating the effort, time, and resources required for its completion [159]. In this context, ''work to do'' represents the total estimated amount of work on the project. At the same time, ''team's working capacity'' indicates the work the team expects to accomplish in each iteration.

Tracking project progress can be done through ''work done'' behavior [162]. It is helpful to analyze the discrepancy between ''work done - estimated'' and ''work done''. Increasing ''work done'' reduces this disparity, while a growing gap requires intervention. This intervention is aligned with the efforts to guide the team, linked to leadership, defined as the 'Ability to direct and coordinate the activities of other team members, assess team effectiveness, assign tasks, develop team knowledge, skills, and abilities, motivate team members, plan and organize, and establish a positive atmosphere [116], according to Table 3.

As leadership improves, communication intensifies because effective leaders understand its importance in achieving goals and keeping their teams focused and engaged [126]. Quality communication between team members and the Project manager positively influences the team's results [128]. In this way, leaders who strive to improve communication often achieve more effective leadership and more satisfied teams.

Teamwork and communication are inherently intertwined. Effective communication enables successful teamwork [127], while effective teamwork, in turn, fosters smoother collaborative communication [118]. These elements form a positive feedback loop (R1). Both are essential for the achievement of objectives and the success of any team.

Finally, improvements in teamwork present increases in productivity [120]. Consequently, teamwork maintains a positive causal relationship with the ability to work as a team, thus increasing the work done.

E. COMMENTS ON THE PROPOSED METHODOLOGY

The literature review is a method used in system dynamics to articulate the problem (Table 1). However, a repeatable procedure to characterize the variables of interest is not exhibited. This aspect is covered by applying systematic mapping, through which it is possible to identify relevant research related to a question or topic of interest [102].

Systematic mapping collects definitions, relationships between variables, and metrics or instruments related to each variable of interest. However, subjecting the results related to the measurement of these variables to a review by professionals working in the field provides a more complete perspective. Doing this evaluation through a survey encourages reflection on the measurement of these variables and their importance. Additionally, the survey avoids the problems of transcription, tabulation, and analysis of data obtained through interviews.

This proposal represents a new perspective for system dynamics modeling because it formalizes a methodology to characterize the key variables to be included in a simulation model. This aspect has yet to be reported in the literature. Finally, this proposed methodology shows adaptability for other work teams, not only those related to software development, and can be further extended to address more complex challenges.

F. LIMITATIONS AND THREATS TO THE VALIDITY OF THE STUDY

This study has certain limitations to consider. First, the survey is only available in Spanish, restricting its applicability to Spanish-speaking countries. The need for future English translations and exploration of this topic in English-speaking contexts is recognized. Moreover, the findings are specific to the context of Software Engineering and should not be prematurely generalized to other disciplines. However, considering the relevance of this study's contributions, broader applicability may be possible in the future.

Internal threats related to article selection bias are identified. However, this threat was addressed by clearly defining the inclusion and exclusion criteria. These elements are part of the systematic mapping protocol and were reviewed by experts external to the research.

The non-probabilistic convenience sampling can introduce biases in the sample since the selection of participants is influenced by personal connections and references, which can condition their responses. Considering this risk, convenience sampling was leveraged by non-probabilistic snowball sampling to expand the sample and capture responses from people with whom there is no direct link.

One of the external threats identified is the small number of studies available in the literature to define, measure, and relate the variables that were interesting to analyze in the research. That is an aspect covered by the proposed methodology by allowing the participants to complement the information captured in the systematic mapping review through a survey of closed and open questions.

Another external threat identified is related to participants' limited knowledge of psychometric tests. However, strategies were implemented to address this potential bias. A brief description of each psychometric test included in the survey was included. Additionally, accessibility was facilitated to complete the survey from anywhere, without time restrictions, and open communication channels were provided to the researchers to address concerns.

These considerations underscore the importance of interpreting the results within their context and recognizing areas of improvement for future research.

VI. CONCLUSION AND FUTURE WORK

The problem articulation corresponds to the first step of system dynamics modeling. The problem is defined and limited in this stage regarding variables and time. It is usual to use qualitative methods due to the importance of resorting to the experience and knowledge of those who are part of the system of interest. However, a standardized procedure needs to be improved to define and describe the variables and their interactions formally.

In response to this gap, this article presented a methodology that addresses this need in the simulation modeling process based on system dynamics. The mixed approach, consisting of a systematic mapping and a survey-based study (Fig. 1), aims to provide a solid and effective structure for characterizing the factors included in the model.

By presenting this methodology and emphasizing its applicability within the Software Engineering context, an effort is made to contribute to advancing research and development in system dynamics-based simulation. Additionally, a valuable tool is provided for practitioners and academics to construct more precise, dependable, and relevant simulation models. This methodology can be used to analyze non-technical factors in another context by adjusting the first keyword of the general structure of the search string.

This study contributes to empirical research in Software Engineering because it characterizes non-technical factors relevant to team management. As a result of the application of the methodology, the research team captured the definitions of communication, leadership, and teamwork reported in the literature. Moreover, it has elucidated the relationships between these factors and their measurement mechanisms. Asking Software Engineering professionals about the use of the metrics or instruments identified at the team level expanded the spectrum of possibilities. Characterizing these factors strengthens the approach of the dynamic hypothesis.

Key findings include:

- The methodology proposed in this study formalizes the articulation of problems in system dynamics. This methodology includes the consecutive use of two methods: (i) systematic mapping study and (ii) survey-based study.
- It enables the detailed portrayal of non-technical factors by addressing (i) the definition of the factor, (ii) its relationship with other factors, and (iii) measurement mechanisms.
- The characterization of non-technical factors, following the proposed methodology, strengthens the formulation of dynamic hypothesis.
- The list of metrics and instruments to measure communication in software development teams expands the possibilities of analysis from dimensions other than ''communication overhead''.
- The instruments reported in this research to measure non-technical factors promote their inclusion in management schemes of software development teams.

The proposed methodology can be applied to characterize other social and human factors such as commitment, autonomy or emotional intelligence, and even non-technical factors such as adaptability or problem-solving. This is achieved by applying the search string's general structure as outlined in the systematic mapping protocol, which specifies the non-technical factor to be included.

By adjusting the first two terms of the search string according to the research interest, this methodology can be extended to other contexts and include other factors. That expands the spectrum of application possibilities of the methodology proposed in this study.

Social and human factors are inherent to the work of software development teams. In this way, the characterization of communication, leadership, and teamwork presented in this study is applicable to both teams that use traditional and agile methodologies. However, the research team identifies a line of future work, adapting the simulation model parameters according to the characteristics of each development methodology. This aspect will be part of the third step of the system dynamics simulation process.

Finally, another aspect that constitutes future work following the system dynamics modeling process is the formulation of the simulation model. This model should be part of a simulation framework that supports informed decision-making in the planning stage of software development projects, incorporating social and human factors.

ACKNOWLEDGMENT

The outcomes of this work were realized within the framework of the binational cooperation project between Universidad de Medellín and Universidad de Aysén, alongside the Department of Natural Sciences and Technology.

REFERENCES

[1] C. Z. Kirilo, J. M. Abe, L. A. de Lima, L. C. Machi Lozano, M. Nogueira, C. C. de Oliveira, and K. Nakamatsu, ''Organizational climate assessment using the paraconsistent decision method,'' *Proc. Comput. Sci.*, vol. 131, pp. 608–618, Nov. 2018, doi: 10.1016/j.procs.2018.04.303.

[2] J. F. Dooley, *Software Development, Design and Coding*. Galesburg, IL, USA: Apress, 2017, doi: 10.1007/978-1-4842-3153-1.

[3] K. Richter, R. R. Dumke, and F. Group, *Modeling, Evaluating and Predicting IT Human Resources Performance*. Boca Raton, FL, USA: CRC Press, 2015.

[4] S. Wagner and M. Ruhe, ''A systematic review of productivity factors in software development,'' in *Proc. 2nd Int. Workshop Software Productiv. Anal. Cost Estimation (SPACE)*, Beijing, China, 2008.

[5] A. Trendowicz and J. Munch, ''Chapter 6 factors influencing software development productivity-state-of-the-art and industrial experiences,'' *Advances in Computers*, vol. 77, no. 9, pp. 185–241, 2009.

[6] C. Sadowski and T. Zimmermann, *Rethinking Productivity in Software Engineering*. Cham, Switzerland: Springer, 2019.

[7] E. D. Canedo and G. A. Santos, ''Factors affecting software development productivity: An empirical study,'' in *Proc. 33rd Brazilian Symp. Softw. Eng.*, Sep. 2019, pp. 307–316, doi: 10.1145/3350768.3352491.

[8] L. F. Capretz, F. Ahmed, and F. Q. B. da Silva, ''Soft sides of software,'' *Inf. Softw. Technol.*, vol. 92, pp. 92–94, Dec. 2017, doi: 10.1016/j.infsof.2017.07.011.

[9] 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.

[10] M. K. Shaikh, ''How to form a software engineering capstone team?'' *Heliyon*, vol. 7, no. 4, Apr. 2021, Art. no. e06629, doi: 10.1016/j.heliyon.2021.e06629.

[11] J. A. García-García, J. G. Enríquez, M. Ruiz, C. Arévalo, and A. Jiménez-Ramírez, ''Software process simulation modeling: Systematic literature review,'' *Comput. Standards Interface*, vol. 70, Jun. 2020, Art. no. 103425, doi: 10.1016/j.csi.2020.103425.

[12] A. M. Law, *Simulation Modeling and Analysis*, 5th ed. New York, NY, USA: McGraw-Hill, 2015.

[13] L. G. Birta and G. Arbez, *Modelling and Simulation Exploring Dynamic System Behaviour*. Springer, 2019.

[14] F. Hillier and G. Lieberman, *Introduction a La Investigacion De Operaciones, 9°*. New York, NY, USA: McGraw-Hill, 2010.

[15] J. D. Sterman, *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York, NY, USA: McGraw-Hill, 2000.

[16] A. Sarmiento-Vásquez and E. López-Sandoval, ''Una comparación cualitativa de la dinámica de sistemas, La simulación de eventos discretos Y La simulación basada en agentes,'' *Ingeniería Ind.*, vol. 1, no. 35, p. 27, Dec. 2017, doi: 10.26439/ing.ind2017.n035.1789.

[17] W. de Paula Ferreira, F. Armellini, and L. A. De Santa-Eulalia, ''Simulation in industry 4.0: A state-of-the-art review,'' *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106868, doi: 10.1016/j.cie.2020.106868.

[18] J. W. Forrester, *Industrial Dynamics*. Cambridge, MA, USA: MIT Press, 1961.

[19] P. M. Senge, *La Quinta Disciplina: El Arte Y La Prectica De La Organizacition Abierta Al Aprendizaje*, 2nd ed. Mountain View, CA, USA: Granica, 2010.

[20] I. J. Martinez-Moyano and G. P. Richardson, ''Best practices in system dynamics modeling,'' *Syst. Dyn. Rev.*, vol. 29, no. 2, pp. 102–123, Apr. 2013, doi: 10.1002/sdr.1495.

[21] J. Duggan, *System Dynamics Modeling with R*. Cham, Switzerland: Springer, 2016.

[22] J. W. Forrester, ''Policies, decisions and information sources for modeling,'' *Eur. J. Oper. Res.*, vol. 59, no. 1, pp. 42–63, May 1992.

[23] L. F. Luna-Reyes and D. L. Andersen, ''Collecting and analyzing qualitative data for system dynamics: Methods and models,'' *Syst. Dyn. Rev.*, vol. 19, no. 4, pp. 271–296, Dec. 2003, doi: 10.1002/sdr.280.

[24] K. Mwita, ''Strengths and weaknesses of qualitative research in social science studies,'' *Int. J. Res. Bus. Social Sci.*, vol. 11, no. 6, pp. 618–625, Sep. 2022, doi: 10.20525/ijrbs.v11i6.1920.

[25] S. Elsawah, S. A. Pierce, S. H. Hamilton, H. van Delden, D. Haase, A. Elmahdi, and A. J. Jakeman, ''An overview of the system dynamics process for integrated modelling of socio-ecological systems: Lessons on good modelling practice from five case studies,'' *Environ. Model. Softw.*, vol. 93, pp. 127–145, Jul. 2017, doi: 10.1016/j.envsoft.2017.03.001.

[26] *Talent Radar How the Best Companies Get the Skills They Need to Thrive in the Digital Era*, Infosys, Karnataka, India, 2019.

[27] M. Hernandez-de-Menendez, R. Morales-Menendez, C. A. Escobar, and M. McGovern, ''Competencies for Industry 4.0,'' *Int. J. Interact. Des. Manuf. (IJIDeM)*, vol. 14, no. 4, pp. 1511–1524, Dec. 2020, doi: 10.1007/s12008-020-00716-2.

[28] L. M. Kipper, S. Iepsen, A. J. Dal Forno, R. Frozza, L. Furstenau, J. Agnes, and D. Cossul, ''Scientific mapping to identify competencies required by Industry 4.0,'' *Technol. Soc.*, vol. 64, Feb. 2021, Art. no. 101454, doi: 10.1016/j.techsoc.2020.101454.

[29] B. Hughes and M. Cotterell, *Software Proiect Management,*. New York, NY, USA: McGraw-Hill, 2009.

[30] J. Collofello, D. Houston, I. Rus, A. Chauhan, D. M. Sycamore, and D. Smith-Daniels, ''A system dynamics software process simulator for staffing policies decision support,'' in *Proc. 31st Hawaii Int. Conf. Syst. Sci.*, 1998, pp. 103–111, doi: 10.1109/hicss.1998.654764.

[31] T. K. Abdel-Hamid and S. Madnick, ''Software productivity: Potential, actual, and perceived,'' *Syst. Dyn. Rev.*, vol. 5, no. 2, pp. 93–113, Jun. 1989, doi: 10.1002/sdr.4260050202.

[32] T. K. Abdel-Hamid, ''The dynamics of software project staffing: A system dynamics based simulation approach,'' *IEEE Trans. Softw. Eng.*, vol. 15, no. 2, pp. 109–119, Jul. 1989, doi: 10.1109/32.21738.

[33] R. Madachy and D. Tarbet, ''Case studies in software process modeling with system dynamics,'' *Softw. Process Improv. Pract.*, vol. 5, nos. 2–3, p. 133, 2000.

[34] L. Cao, B. Ramesh, and T. Abdel-Hamid, ''Modeling dynamics in agile software development,'' *ACM Trans. Manage. Inf. Syst.*, vol. 1, no. 1, pp. 1–26, Dec. 2010, doi: 10.1145/1877725.1877730.

[35] C. Andersson, L. Karlsson, J. Nedstam, M. Host, and B. I. Nilsson, ''Understanding software processes through system dynamics simulation: A case study,'' in *Proc. 9th Annu. IEEE Int. Conf. Workshop Eng. Comput.-Based Syst.*, Sep. 2002, pp. 41–48, doi: 10.1109/ECBS.2002.999821.

[36] P. C. Das and U. R. Dhar, ''A system dynamics approach towards software development project—A case study,'' *Int. Adv. Res. J. Sci. Eng. Technol.*, vol. 3, no. 4, pp. 120–123, 2016.

[37] H. Rahmandad and D. M. Weiss, ''Dynamics of concurrent software development,'' *Syst. Dyn. Rev.*, vol. 25, no. 3, pp. 224–249, Jul. 2009.

[38] M. T. I. Trammell, A. Moulton, and S. E. Madnick, ''Effects of funding fluctuations on software development: A system dynamics analysis,'' *Eng. Manage. J.*, vol. 28, no. 2, pp. 71–85, Apr. 2016.

[39] J. S. Aguilar-Ruiz, J. C. Riquelme, D. Rodríguez, and I. Ramos, ''Generation of management rules through system dynamics and evolutionary computation,'' *Lect. Notes Comput. Sci.*, vol. 2559, pp. 615–628, 2002.

[40] A. G. Rodrigues and T. M. Williams, ''System dynamics in project management: Assessing the impacts of client behaviour on project performance,'' *J. Oper. Res. Soc.*, vol. 49, no. 1, pp. 2–15, Jan. 1998, doi: 10.1057/palgrave.jors.2600490.

[41] E. F. Franco, K. Hirama, and M. M. Carvalho, ''Applying system dynamics approach in software and information system projects: A mapping study,'' *Inf. Softw. Technol.*, vol. 93, pp. 58–73, Jan. 2018, doi: 10.1016/j.infsof.2017.08.013.

[42] E. Balas, ''IFORS' operational research Hall of fame,'' *Int. Trans. Oper. Res.*, vol. 13, no. 2, pp. 169–174, Mar. 2006, doi: 10.1111/j.1475-3995.2006.00541.x.

[43] A. P. Galvão Scheidegger, T. Fernandes Pereira, M. L. Moura de Oliveira, A. Banerjee, and J. A. Barra Montevechi, ''An introductory guide for hybrid simulation modelers on the primary simulation methods in industrial engineering identified through a systematic review of the literature,'' *Comput. Ind. Eng.*, vol. 124, pp. 474–492, Oct. 2018, doi: 10.1016/j.cie.2018.07.046.

[44] J. Aracil, *Dinámica de Sistemas*, 4th ed. Madrid, España: Isdefe, 2016.
[45] C. Cadenas and W. Guaita, *Dinámica De Sistemas: Una Metodologia Para La Construccion De Modelos De Toma De Decisiones En Sectores Agroindustriales*. Bogotá, Colombia: Editorial Politécnico Grancolombiano, 2020.
[46] S. Albin and J. W. Forrester, *Building a System Dynamics Model Part 1: Conceptualization*. Cambridge, MA, USA: Massachusetts Inst. Technol., 1997, p. 35.
[47] M. Schaffernicht, *Indagacion De Situaciones Dinamicas Mediante La Dinamica De Sistemas*. San Francisco, CA, USA: Fundamentos, 2006.
[48] E. Pruyt, *System Dynamics Models for Big Issues: Triple Jump Towards Real-World Complexity*. The Netherlands: TU Delft Library, 2013.
[49] Y. Barlas, "Model validation in system dynamics," in *Proc. Int. Syst. Dyn. Conf.*, 1994, pp. 1–10.
[50] S. Ferreira and M. Faezipour, "An analysis of processes, risks, and best practices for use in developing systems engineering process simulators," *Proc. Comput. Sci.*, vol. 8, pp. 87–92, Jun. 2012, doi: 10.1016/j.procs.2012.01.018.
[51] C. Sallati, J. D. A. Bertazzi, and K. Schützer, "Professional skills in the product development process: The contribution of learning environments to professional skills in the Industry 4.0 scenario," *Proc. CIRP*, vol. 84, pp. 203–208, Oct. 2019, doi: 10.1016/j.procir.2019.03.214.
[52] I. Pavez, H. Gómez, C. Liu, and V. A. González, "Measuring project team performance: A review and conceptualization," *Int. J. Project Manage.*, vol. 40, no. 8, pp. 951–971, Nov. 2022.
[53] A. Rodrigues and J. Bowers, "System dynamics in project management: A comparative analysis with traditional methods," *Syst. Dyn. Rev.*, vol. 12, no. 2, pp. 121–139, 1996.
[54] A. G. Rodrigues and T. M. Williams, "System dynamics in software project management: Towards the development of a formal integrated framework," *Eur. J. Inf. Syst.*, vol. 6, no. 1, pp. 51–66, Mar. 1997, doi: 10.1057/palgrave.ejis.3000256.
[55] D. Pfahl, A. Al-Emran, and G. Ruhe, "A system dynamics simulation model for analyzing the stability of software release plans," *Softw. Process Improv. Pract.*, vol. 12, pp. 475–490, Sep. 2007.
[56] J. D. Tvedt and J. S. Gollofello, "Evaluating the effectiveness of process improvements on software development cycle time via system dynamics modelling," in *Proc. 19th Annu. Int. Comput. Softw. Appl. Conf. (COMPSAC95)*, 1995, pp. 318–325, doi: 10.1109/CMPSAC.1995.524796.
[57] R. J. Madachy, "System dynamics modeling of an inspection-based process," in *Proc. IEEE 18th Int. Conf. Softw. Eng.*, Dec. 1996, pp. 376–386, doi: 10.1109/icse.1996.493432.
[58] Z. Sahaf, V. Garousi, D. Pfahl, R. Irving, and Y. Amannejad, "When to automate software testing? Decision support based on system dynamics: An industrial case study," in *Proc. Int. Conf. Softw. Syst. Process*, May 2014, pp. 149–158, doi: 10.1145/2600821.2600832.
[59] I. Fatema, "Agile software develoment teamwork productivity—A system dynamcs approach to analyse the productivity influence factors," M.S. thesis, Ind. Inform. Tech., Univ. Dhaka, Dhaka, Bangladesh, 2019.
[60] F. Kortum, J. Klünder, and K. Schneider, "Characterizing relationships for system dynamics models supported by exploratory data analysis—A conceptualizing approach about the meeting diversity in student software projects," in *Proc. Int. Conf. Softw. Eng. Knowl. Eng.*, Jul. 2017, pp. 608–614, doi: 10.18293/seke2017-143.
[61] M. Hirudayaraj, R. Baker, F. Baker, and M. Eastman, "Soft skills for entry-level engineers: What employers want," *Educ. Sci.*, vol. 11, no. 10, p. 641, Oct. 2021, doi: 10.3390/educsci11100641.
[62] J. Whittle, M. A. Ferrario, W. Simm, and W. Hussain, "A case for human values in software engineering," *IEEE Softw.*, vol. 38, no. 1, pp. 106–113, Jan. 2021, doi: 10.1109/MS.2019.2956701.
[63] M. Kunc, M. J. Mortenson, and R. Vidgen, "A computational literature review of the field of system dynamics from 1974 to 2017," *J. Simul.*, vol. 12, no. 2, pp. 115–127, Apr. 2018.
[64] N. Kedir, N. Siraj, and A. R. Fayek, "Application of system dynamics in construction engineering and management: Content analysis and systematic literature review," *Adv. Civil Eng.*, vol. 2023, pp. 1–22, May 2023.
[65] W. B. Fontoura and G. M. Ribeiro, "System dynamics for sustainable transportation policies: A systematic literature review," *Urbe Revista Brasileira de Gestão Urbana*, vol. 13, pp. 1–15, Aug. 2021, doi: 10.1590/2175-3369.013.e20200259.
[66] O. O. Oladimeji, H. Keathley-Herring, and J. A. Cross, "System dynamics applications in performance measurement research: A systematic literature review," *Int. J. Productiv. Perform. Manage.*, vol. 69, no. 7, pp. 1541–1578, Apr. 2020, doi: 10.1108/ijppm-12-2018-0453.
[67] D. Rumeser and M. Emsley, "Key challenges of system dynamics implementation in project management," *Proc. Social Behav. Sci.*, vol. 230, pp. 22–30, Sep. 2016, doi: 10.1016/j.sbspro.2016.09.004.
[68] B. Oyo, D. Williams, and E. Barendsen, "Integrating action research and system dynamics: Towards a generic process design for participative modelling," in *Proc. 42nd Annu. Hawaii Int. Conf. Syst. Sci. HICS*, 2009, pp. 1–11.
[69] M. Zolfagharian, R. Akbari, and H. Fartookzadeh, "Theory of knowledge in system dynamics models," *Found. Sci.*, vol. 19, no. 2, pp. 189–207, Jun. 2014, doi: 10.1007/s10699-013-9328-9.
[70] B. Bala, F. Mohamed, and K. Mohd, *System Dynamics. Modelling and Simulation*. Cham, Switzerland: Springer, 2017.
[71] M. G. Oladokun and C. O. Aigbavboa, "The system dynamics modelling method," in *Simulation-Based Analysis of Energy and Carbon Emissions in the Housing Sector* (Green Energy and Technology). Cham, Switzerland: Springer, 2018, pp. 77–95, doi: 10.1007/978-3-319-75346-1_4.
[72] C. R. Kothari, *Research Methodology—Methods and Techniques*. Chennai, India: New Age Int., Ltd, 2004.
[73] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*. Malden, MA, USA: Blackwell Publishing, 2008, doi: 10.1002/9780470754887.
[74] U. Flick, *An Introduction To Qualitative Research*. Newbury Park, CA, USA: Sage, 2009.
[75] R. Yin, *Qualitative Research From Start To Finish*. New York, NY, USA: Guilford Press, 2016.
[76] J. Carver, C. Seaman, and R. Jeffery, *Using Qualitative Methods in Software Engineering*. Lund, Sweden: Lund Institute of Technology, 2004.
[77] R. Kumar, *Research Methodology A Step-by-step Guide for Beginners*. Newbury Park, CA, USA: Sage, 2011.
[78] C. A. M. Álvarez, *Metodología de la investigación cuantitativa y cualitativa. Guía didáctica*. Neiva, Colombia: Universidad Surcolombiana, 2011.
[79] R. Widdows, T. A. Hensler, and M. H. Wyncott, "The focus group interview: A method for assessing users' evaluation of library service," *College Res. Libraries*, vol. 52, no. 4, pp. 352–359, Jul. 1991, doi: 10.5860/crl_52_04_352.
[80] G. Maruyama and C. S. Ryan, *Research Methods in Social Relations*, 8th ed. Malden, MA, USA: Wiley, 2014.
[81] J. Morecroft, "System dynamics," in *Systems Approaches to Making Change: A Practical Guide*, M. Reynolds and S. Holwell, Eds. London, U.K.: Springer, 2020, pp. 25–88, doi: 10.1007/978-1-4471-7472-1_2.
[82] A. Powell, K. Mander, and D. Brown, "Strategies for lifecycle concurrency and iteration—A system dynamics approach," *J. Syst. Softw.*, vol. 46, nos. 2–3, pp. 151–161, Apr. 1999.
[83] P. Hsia, C.-T. Hsu, and D. C. Kung, "Brooks' law revisited: A system dynamics approach," in *Proc. 23rd Annu. Int. Comput. Softw. Appl. Conf.*, 1999, pp. 370–375.
[84] J. M. Lyneis, K. G. Cooper, and S. A. Els, "Strategic management of complex projects: A case study using system dynamics," *Syst. Dyn. Rev.*, vol. 17, no. 3, pp. 237–260, Sep. 2001, doi: 10.1002/sdr.213.
[85] M. Wu and H. Yan, "Simulation in software engineering with system dynamics: A case study," *J. Softw.*, vol. 4, no. 10, pp. 1127–1135, Dec. 2009, doi: 10.4304/jsw.4.10.1127-1135.
[86] K. Hiekata, M. T. Khatun, and M. A. Chavy-Macdonald, "System dynamics modeling to manage performance based on scope change for software development projects," in *Proc. 26th ISTE Int. Conf. Transdisciplinary Eng.*, Tokyo, Japan, 2019, pp. 675–684, doi: 10.3233/ATDE190177.
[87] M. Farshchi, Y. Jusoh, and A. Murad, "Impact of personnel factors on the recovery of delayed software projects: A system dynamics approach," *Comput. Sci. Inf. Syst.*, vol. 9, no. 2, pp. 627–652, 2012, doi: 10.2298/csis110525003f.
[88] M. Alefari, M. Almanei, and K. Salonitis, "A system dynamics model of employees' performance," *Sustainability*, vol. 12, no. 16, p. 6511, Aug. 2020, doi: 10.3390/su12166511.
[89] X. Zhang, Y. Tan, and Z. Yang, "Analysis of impact of requirement change on product development progress based on system dynamics," *IEEE Access*, vol. 9, pp. 445–457, 2021, doi: 10.1109/ACCESS.2020.3046753.
[90] G. Valença and C. Alves, "Strategies to foster software ecosystem partnerships—A system dynamics analysis," in *Enterprise Information Systems* (Lecture Notes in Business Information Processing). Porto, Portugal, 2018, pp. 189–211, doi: 10.1007/978-3-319-93375-7_10.
[91] S. Matalonga, M. Solari, and T. S. Feliu, "An empirically validated simulation for understanding the relationship between process conformance and technology skills," *Softw. Quality J.*, vol. 22, no. 4, pp. 593–609, Dec. 2014.

[92] D. Mishra and B. Mahanty, "A study of software development project cost, schedule and quality by outsourcing to low cost destination," *J. Enterprise Inf. Manage.*, vol. 29, no. 3, pp. 454–478, Apr. 2016, doi: 10.1108/jeim-08-2014-0080.

[93] I. Fatema and K. Sakib, "Using qualitative system dynamics in the development of an agile teamwork productivity model," *Int. J. Adv. Softw.*, vol. 11, nos. 1–2, pp. 170–185, 2018.

[94] M. Maghsoudi and N. Nezafati, "Navigating the acceptance of implementing business intelligence in organizations: A system dynamics approach," *Telematics Informat. Rep.*, vol. 11, Sep. 2023, Art. no. 100070, doi: 10.1016/j.teler.2023.100070.

[95] E. Ginters et al., "Approach to the development of a conceptual model of a sociotechnical system based on digital technologies," in *Proc. 6th Eur. Conf. Electr. Eng. Comput. Sci. (ELECS)*, Dec. 2022, pp. 36–46, doi: 10.1109/ELECS55825.2022.00014.

[96] A. M. S. Rodríguez, "Diseño Y validación de instrumentos de medición," *Diá-logos*, vol. 8, no. 14, pp. 19–40, 2014.

[97] S. Roy and P. K. J. Mohapatra, "Methodological problems in the formulation and validation of system dynamics models incorporating soft variables," in *Proc. Syst. Dyn. Soc., 21st Int. Conf.*, 1994, pp. 1–15.

[98] G. Coyle, "Qualitative and quantitative modelling in system dynamics: Some research questions," *Syst. Dyn. Rev.*, vol. 16, no. 3, pp. 225–244, 2000.

[99] B. Durakovic, "Design of experiments application, concepts, examples: State of the art," *Periodicals Eng. Natural Sci. (PEN)*, vol. 5, no. 3, pp. 421–439, Dec. 2017, doi: 10.21533/pen.v5i3.145.

[100] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele, Staffs, U.K., 2007, doi: 10.1145/1134285.1134500.

[101] P. Cazau, *Introducción a la Investigación en Ciencias Sociales*, 3th ed. Buenos Aires, Argentina: Rundinuskín, 2006.

[102] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, Apr. 2007, doi: 10.1016/j.jss.2006.07.009.

[103] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer, 2012, doi: 10.1007/978-3-642-29044-2.

[104] P. K. Muhuri, A. K. Shukla, and A. Abraham, "Industry 4.0: A bibliometric analysis and detailed overview," *Eng. Appl. Artif. Intell.*, vol. 78, pp. 218–235, Feb. 2019, doi: 10.1016/j.engappai.2018.11.007.

[105] M. J. Cobo, B. Jürgens, V. Herrero-Solana, M. A. Martínez, and E. Herrera-Viedma, "Industry 4.0: A perspective based on bibliometric analysis," *Proc. Comput. Sci.*, vol. 139, pp. 364–371, Aug. 2018, doi: 10.1016/j.procs.2018.10.278.

[106] D. S. Cruzes and T. Dyb, "Research synthesis in software engineering: A tertiary study," *Inf. Softw. Technol.*, vol. 53, no. 5, pp. 440–455, May 2011, doi: 10.1016/j.infsof.2011.01.004.

[107] X. Huang, H. Zhang, X. Zhou, M. Ali Babar, and S. Yang, "Synthesizing qualitative research in software engineering: A critical review," in *Proc. IEEE/ACM 40th Int. Conf. Softw. Eng. (ICSE)*, May 2018, pp. 1207–1218.

[108] B. A. Kitchenham and S. L. Pfleeger, "Personal opinion surveys," in *Guide to Advanced Empirical Software Engineering*. London, U.K.: Springer, 2008, pp. 63–92, doi: 10.1007/978-1-84800-044-5_3.

[109] L. Machuca-Villegas, G. P. Gasca-Hurtado, S. M. Puente, and L. M. R. Tamayo, "An instrument for measuring perception about social and human factors that influence software development productivity," *J. Universal Comput. Sci.*, vol. 27, no. 2, pp. 111–134, Feb. 2021, doi: 10.3897/jucs.65102.

[110] I. Pedrosa, J. Suárez-Álvarez, and E. García-Cueto, "Evidencias sobre la validez de contenido: Avances teóricos Y métodos para su estimación [Content validity evidences: Theoretical advances and estimation Methods]," *Acción Psicológica*, vol. 10, no. 2, p. 3, Jun. 2014, doi: 10.5944/ap.10.2.11820.

[111] R. W. Cooksey, *Illustrating Statistical Procedures: Finding Meaning in Quantitative Data*, 3th ed. Singapore: Springer, 2020, doi: 10.1007/978-981-15-2537-7.

[112] F. G. de Oliveira Neto, R. Torkar, R. Feldt, L. Gren, C. A. Furia, and Z. Huang, "Evolution of statistical analysis in empirical software engineering research: Current state and steps forward," *J. Syst. Softw.*, vol. 156, pp. 246–267, Oct. 2019, doi: 10.1016/j.jss.2019.07.002.

[113] F. Mendes, E. Mendes, N. Salleh, and M. Oivo, "Insights on the relationship between decision-making style and personality in software engineering," *Inf. Softw. Technol.*, vol. 136, Aug. 2021, Art. no. 106586, doi: 10.1016/j.infsof.2021.106586.

[114] N. Forsgren, M.-A. Storey, C. Maddila, T. Zimmermann, B. Houck, and J. Butler, "The space of developer productivity," *Commun. ACM*, vol. 64, no. 6, pp. 46–53, Jun. 2021, doi: 10.1145/3453928.

[115] J. M. Carroll, H. Jiang, and M. Borge, "Distributed collaborative homework activities in a problem-based usability engineering course," *Educ. Inf. Technol.*, vol. 20, no. 3, pp. 589–617, Sep. 2015, doi: 10.1007/s10639-013-9304-6.

[116] D. Strode, T. Dingsøyr, and Y. Lindsjorn, "A teamwork effectiveness model for agile software development," *Empirical Softw. Eng.*, vol. 27, no. 2, pp. 1–50, Mar. 2022, doi: 10.1007/s10664-021-10115-0.

[117] M. Galster, A. Mitrovic, S. Malinen, and J. Holland, "What soft skills does the software industry *Really* want? An exploratory study of software positions in New Zealand," in *Proc. 16th ACM IEEE Int. Symp. Empirical Softw. Eng. Meas.*, Sep. 2022, pp. 272–282, doi: 10.1145/3544902.3546247.

[118] R. Wlodarski, J.-R. Falleri, and C. Parvéry, "Assessment of a hybrid software development process for student projects: A controlled experiment," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng. Softw. Eng. Educ. Training (ICSE-SEET)*, May 2021, pp. 289–299, doi: 10.1109/icse-seet52601.2021.00039.

[119] N. Assavakamhaenghan, W. Tanaphantaruk, P. Suwanworaboon, M. Choetkiertikul, and S. Tuarob, "Quantifying effectiveness of team recommendation for collaborative software development," *Automated Softw. Eng.*, vol. 29, no. 2, pp. 1–48, Nov. 2022, doi: 10.1007/s10515-022-00357-7.

[120] G. Marsicano, F. Q. B. da Silva, C. B. Seaman, and B. G. Adaid-Castro, "The teamwork process antecedents (TPA) questionnaire: Developing and validating a comprehensive measure for assessing antecedents of teamwork process quality," *Empirical Softw. Eng.*, vol. 25, no. 5, pp. 3928–3976, Sep. 2020.

[121] F. Kortum, O. Karras, J. Klünder, and K. Schneider, "Towards a better understanding of team-driven dynamics in agile software projects: A characterization and visualization support in JIRA," in *Product-Focused Software Process Improvement*. Barcelona, Spain, 2019, pp. 725–740, doi: 10.1007/978-3-030-35333-9_56.

[122] E. Bjarnason, B. G. Bern, and L. Svedberg, *Inter-Team Communication in Large-scale Co-located Software Engineering: A Case Study*, vol. 27. Cham, Switzerland: Springer, 2022, doi: 10.1007/s10664-021-10027-z.

[123] Y. I. Alzoubi, A. Q. Gill, and B. Moulton, "A measurement model to analyze the effect of agile enterprise architecture on geographically distributed agile development," *J. Softw. Eng. Res. Develop.*, vol. 6, no. 1, pp. 1–15, Dec. 2018, doi: 10.1186/s40411-018-0048-2.

[124] E. Bjarnason, H. Sharp, and B. Regnell, "Improving requirements-test alignment by prescribing practices that mitigate communication gaps," *Empirical Softw. Eng.*, vol. 24, no. 4, pp. 2364–2409, Aug. 2019, doi: 10.1007/s10664-019-09698-6.

[125] M. Ortu, G. Destefanis, S. Counsell, S. Swift, R. Tonelli, and M. Marchesi, "How diverse is your team? Investigating gender and nationality diversity in GitHub teams," *J. Softw. Eng. Res. Develop.*, vol. 5, no. 1, pp. 1–45, Dec. 2017, doi: 10.1186/s40411-017-0044-y.

[126] Y. Li, C.-H. Tan, and H.-H. Teo, "Leadership characteristics and developers' motivation in open source software development," *Inf. Manage.*, vol. 49, no. 5, pp. 257–267, Jul. 2012, doi: 10.1016/j.im.2012.05.005.

[127] S. Wood, G. Michaelides, and C. Thomson, "Successful extreme programming: Fidelity to the methodology or good teamworking?" *Inf. Softw. Technol.*, vol. 55, no. 4, pp. 660–672, Apr. 2013, doi: 10.1016/j.infsof.2012.10.002.

[128] J. B. Awotunde, F. E. Ayo, R. O. Ogundokun, O. E. Matiluko, and E. A. Adeniyi, *Investigating the Roles of Effective Communication Among Stakeholders in Collaborative Software Development Projects*, vol. 12254. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-58817-5_24.

[129] S. A. Licorish and S. G. MacDonell, "Communication and personality profiles of global software developers," *Inf. Softw. Technol.*, vol. 64, pp. 113–131, Aug. 2015, doi: 10.1016/j.infsof.2015.02.004.

[130] A. Tarasov, *Assessing Job Satisfaction of Software Engineers Using GQM Approach*, vol. 11771. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-29852-4_10.

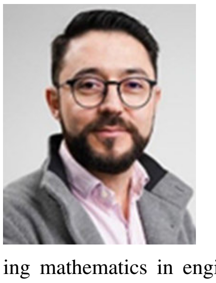
[131] A. Y. Aksekili and C. J. Stettina, *Women in Agile: The Impact of Organizational Support for Women's Advancement on Teamwork Quality and Performance in Agile Software Development Teams* (Lecture Notes in Computer Science), vol. 408, 2021, pp. 3–23.

[132] R. Jolak, "Software engineering whispers: The effect of textual vs. Graphical software design descriptions on software design communication," *Empirical Software Eng.*, vol. 25, no. 6, pp. 1–18, 2020.

[133] D. Russo, P. H. P. Hanel, S. Altnickel, and N. van Berkel, "Predictors of well-being and productivity among software professionals during the COVID-19 pandemic—A longitudinal study," *Empirical Softw. Eng.*, vol. 26, no. 4, pp. 1–17, Jul. 2021.
[134] S. Datta, "How does developer interaction relate to software quality? An examination of product development data," *Empirical Softw. Eng.*, vol. 23, no. 3, pp. 1–15, 2018, doi: 10.1007/s10664-017-9534-0.
[135] A. Grigoriev and O. Mondrus, "Managing academic performance by optimal resource allocation," *Scientometrics*, vol. 127, no. 5, pp. 2433–2453, May 2022, doi: 10.1007/s11192-022-04342-5.
[136] E. Constantinou and T. Mens, "An empirical comparison of developer retention in the RubyGems and npm software ecosystems," *Innov. Syst. Softw. Eng.*, vol. 13, nos. 2–3, pp. 101–115, Sep. 2017, doi: 10.1007/s11334-017-0303-4.
[137] P. Ciancarini, M. Missiroli, and S. Zani, *Empirical Evaluation of Agile Teamwork*, vol. 1439. Cham, Switzerland: Springer, 2021.
[138] U. M. Devadas and Y. Y. Dharmapala, "Soft skills evaluation in the information technology and business process management industry in Sri Lanka: Skills, methods and problems," *Int. J. Econ. Bus. Hum. Behav.*, vol. 2, no. 3, pp. 1–13, 2021, doi: 10.5281/zenodo.5280309.
[139] J. Weichbrodt et al., *Understanding Leadership in Agile Software Development Teams: Who and How?*, vol. 445. Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-031-08169-9_7.
[140] R. Wlodarski, A. Poniszewska-Maranda, and J.-R. Falleri, "Comparative case study of plan-driven and agile approaches in student computing projects," in *Proc. Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2020, pp. 1–6, doi: 10.23919/SoftCOM50211.2020.9238196.
[141] M. Kuutila, M. Mäntylä, M. Claes, M. Elovainio, and B. Adams, "Individual differences limit predicting well-being and productivity using software repositories: A longitudinal industrial study," *Empirical Softw. Eng.*, vol. 26, no. 5, p. 88, Sep. 2021, doi: 10.1007/s10664-021-09977-1.
[142] J. Klünder, F. Kortum, T. Ziehm, and K. Schneider, "Helping teams to help themselves: An industrial case study on interdependencies during sprints," in *Human-Centered Software Engineering*. Cham, Switzerland: Springer, 2019, pp. 31–50.
[143] J. Banks, *Handbook of Simulation Principles, Methodology, Advances, Applications, and Practice*. Hoboken, NJ, USA: Wiley, 1998.
[144] L. Machuca-Villegas, G. P. Gasca-Hurtado, L. M. Restrepo-Tamayo, and S. M. Puente, "Social and human factor classification of influence in productivity in software development teams," in *EuroSPI Systems, Software and Services Process Improvement*. Düsseldorf, Germany, 2020, pp. 717–729, doi: 10.1007/978-3-030-56441-4_54.
[145] G. Matturro, F. Raschetti, and C. Fontán, "A systematic mapping study on soft skills in software engineering," *J. Univers. Comput. Sci.*, vol. 25, no. 1, 2019, doi: 10.3217/jucs-025-01-0016.
[146] *SWEBOK V3.0 Guide to the Software Engineering Body of Knowledge*. Los Ángeles, CA, USA: IEEE Computer Society Press, 2014.
[147] F. Fagerholm, M. Ikonen, P. Kettunen, J. Münch, V. Roto, and P. Abrahamsson, "Performance alignment work: How software developers experience the continuous adaptation of team performance in lean and agile environments," *Inf. Softw. Technol.*, vol. 64, pp. 132–147, Aug. 2015, doi: 10.1016/j.infsof.2015.01.010.
[148] J. F. DeFranco and P. A. Laplante, "Review and analysis of software development team communication research," *IEEE Trans. Prof. Commun.*, vol. 60, no. 2, pp. 165–182, Jun. 2017, doi: 10.1109/TPC.2017.2656626.
[149] F. Q. B. da Silva, C. V. F. Monteiro, I. E. dos Santos, and L. F. Capretz, "How software development group leaders influence team members' innovative behavior," *IEEE Softw.*, vol. 33, no. 5, pp. 106–109, Sep. 2016, doi: 10.1109/MS.2016.120.
[150] W. Hasselbring and S. Giesecke, *Research Methods in Software Engineering*. Oldenburg, Germany: GITO Verlag, 2006.
[151] C. Ugwu and H. E. Val, "Qualitative research," *IDOSR Journal Comput. Appl. Sci.*, vol. 8, no. 1, pp. 20–35, 2023.
[152] A. Cropley, *Introduction to Qualitative Research Methods: A Practice-Oriented Introduction*. Timisoara, Romania: Intaglio Publications, 2022, doi: 10.1016/b978-0-12-107750-1.50012-1.
[153] L. M. Restrepo-Tamayo and G. P. Gasca-Hurtado, "Human aspects in software development: A systematic mapping study," in *Collaboration Technologies and Social Computing* (Lecture Notes in Computer Science). Santiago, Chile, 2022, pp. 1–22, doi: 10.1007/978-3-031-20218-6_1.
[154] M. J. Tejeda, T. A. Scandura, and R. Pillai, "The MLQ revisited: Psychometric properties and recommendations," *Leadership Quart.*, vol. 12, no. 1, pp. 31–52, Mar. 2001.
[155] L. L. Bender, *Measurement of Non-technical Skills of Software Development Teams*. Fargo, North Dakota: North Dakota State University, 2014.
[156] R. Feldt, R. Torkar, L. Angelis, and M. Samuelsson, "Towards individualized software engineering: Empirical studies should collect psychometrics," in *Proc. Int. Workshop Cooperat. Human Aspects Softw. Eng.*, May 2008, pp. 49–52, doi: 10.1145/1370114.1370127.
[157] D. H. Kluemper and P. A. Rosen, "Future employment selection methods: Evaluating social networking web sites," *J. Managerial Psychol.*, vol. 24, no. 6, pp. 567–580, Aug. 2009, doi: 10.1108/02683940910974134.
[158] A. P. Murray, *The Complete Software Project Manager*. Hoboken, NJ, USA: Wiley, 2016.
[159] A. Villafiorita, *Introduction To Software Project Management*. Boca Raton, FL, USA: CRC Press, 2014.
[160] E. Derby and D. Larsen, *Agile Retrospectives Making Good Teams Great*. Raleigh, NC, USA: The Pragmatic Bookshelf, 2006.
[161] Y. Andriyani, R. Hoda, and R. Amor, "Reflection in agile retrospectives," in *Proc. 18th Int. Conf. Agile Softw. Develop.*, Cologne, Germany, 2017, pp. 3–19, doi: 10.1007/978-3-319-57633-6_1.
[162] K. G. Cooper, "The rework cycle: Why projects are mismanaged," *Proj. Manag. Action*, vol. 24, no. 1, pp. 17–21, 1993.

LUZ MARCELA RESTREPO-TAMAYO received the Graduate degree in industrial engineering from the University of Antioquia, in 2009, and the master's degree in statistics from the National University of Colombia, in 2014. She is currently pursuing the Ph.D. degree in engineering with the University of Medellín. Since 2016, she has been an Associate Professor with the Faculty of Engineering, University of Medellín. Her research interest includes improving the productivity and quality of processes through data analysis. She is also working on the study of the productivity of software development teams and considering non-technical factors based on system dynamics.

GLORIA PIEDAD GASCA-HURTADO received the Ph.D. degree from the Department of Languages, Computer Systems and Software Engineering, Faculty of Informatics, Polytechnic University of Madrid, Spain. She is currently a Professor with the Faculty of Engineering, University of Medellín. In addition, she has a background in systems engineering and a specialization in systems auditing. Her research interests include educational innovation and digital transformation in engineering education, with a specific focus on gamification and enabling technologies of the fourth industrial revolution, training in promising practices for software development in teams and agile methodologies, the treatment of social and human factors in software engineering, and the productivity of software development teams. At the curricular level, the research professor has experience in leading and participating in curricular innovation processes that incorporate active learning strategies.



JOHNNY VALENCIA-CALVO received the degree in electronics engineering and the M.Sc. degree in industrial automation from Universidad Nacional de Colombia, in 2010 and 2012, respectively, and the Ph.D. degree in computer sciences and decision, in 2016.

He is currently an Assistant Professor with the University of Aysén, specializes in dynamic analysis, nonlinear dynamics, and system dynamics, with a focus on modeling, simulation, and applying mathematics in engineering. His professional expertise encompasses systems dynamics, complex systems modeling, project management, and administration. His research interests include dynamic analysis, numerical methods, system dynamics, underpinned by strong skills in modeling, simulation, experimentation, and applied mathematics. He has led projects aimed at promoting science, technology, and innovation, forming, and integrating teams around these themes. His current research interests include investigates remote sensing, analysis, modeling, simulation based on system dynamics, decision-making, knowledge management, and smart lands. He aims to establish the Data, Modeling, and Simulation Center, Patagonia, Chile, to support decision-making through a digital ecosystem, enhancing public policy design with technology, and science-based solutions.