Unveiling the Landscape of System Thinking Modeling Tools Use in Software Engineering

Júlia de Souza Borges
julia.s.borges@edu.ufes.br
Ontology Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo
Vitoria, ES, Brazil

Thiago Felippe Neitzke Lahass
thiago.lahass@edu.ufes.br
Ontology Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo
Vitoria, ES, Brazil

Amanda Brito Apolinário
amanda.apolinario@edu.ufes.br
Ontology Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo
Vitoria, ES, Brazil

Paulo Sérgio dos Santos Júnior
paulo.junior@ifes.edu.br
LEDS Research, Department of Informatics, Federal Institute of Education, Science, and Technology of Espírito Santo
Serra, ES, Brazil

Monalessa P. Barcellos
monalessa@inf.ufes.br
Ontology Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo
Vitoria, ES, Brazil

ABSTRACT

Context: Software organizations involve several processes, people, practices, culture, and other factors that affect their behavior. Understanding the organizational environment is crucial for improving processes and products. System Thinking (ST) provides several tools that help in this matter. ST views an organization as a system, comprising elements and interconnections coherently organized in a structure that produces a characteristic set of behaviors. *Objective:* Given ST's successful application in different areas of industry and academia, and its potential to address Software Engineering (SE) problems, we decided to investigate how ST modeling tools have been used in SE. Our goal is to provide a panorama of the use of such tools in SE, identify gaps, and shine a light on research opportunities. *Method:* We carried out a systematic literature mapping, analyzing 700 publications. From these, we identified 10 publications addressing the use of ST modeling tools in SE and conducted an in-depth analysis by investigating the tools that have been used in SE, the ways of and purposes for using these tools, the SE processes that have been supported by them, and the difficulties and benefits that have been perceived when using such tools. *Results:* ST modeling tools have been employed to tackle SE issues in organizational contexts and to model and analyze the results of SE studies. Project Management and Requirements Engineering have been the most common processes in which these tools have been applied. ST modeling tools have helped get a holistic understanding of complex systems and identify strategies to address problematic behaviors that affect SE outcomes. Causal loop diagrams have been the predominant tool. The lack of necessary knowledge has been a challenge for using the tools. *Conclusion*: Using ST tools in SE holds promise. Nevertheless, they also suggest that ST modeling tools have been under-explored in this context. Hence, further investigation into their potential for enhancing SE products and processes is warranted.

KEYWORDS

System Thinking, Software Engineering, Mapping Study, Modeling Tools

1 INTRODUCTION

Characteristics and demands of the modern and digital society have transformed the software development scenario and presented new challenges to software engineers, such as faster deliveries, frequent changes in requirements, lower tolerance to failures, and the need to adapt to contemporary business models [4]. Coping with these challenges is complex and involves various aspects, such as processes, people, tools, policies, and culture. Therefore, a broad and systemic view of the organizational environment is required. System Thinking (ST) is a suitable approach to provide such a view.

In the most common way of analyzing a problem, it is first explored in small parts, at the beginning trying to understand each fragment of the problem and only after that trying to join those fragments together in the joined and more formal representation. ST brings an alternative way in which the interdependence of fragments is understood before analyzing each particular fragment. This interdependence may have a dynamic or structural nature [13]. ST sees an organization as a *system* [1] , consisting of elements (e.g., teams, artifacts, policies) and interconnections (e.g., the relation between the development team, the software artifacts it produces, and the policies that influence their production) coherently organized in a structure that produces a characteristic set of behaviors, often classified as its function or purpose (e.g., the development team produces a software product aiming to accomplish the team function in the organization) [22, 27].

ST has been considered one of the key management competencies for the 21st century. The systemic perspective provided by ST enables us to understand how the system works and, thus, function

[1]Thus, in this paper, when talking about *systems* in the ST context, we are referring to organizational environments (or extracts of them).

more effectively and proactively within it. The more we understand the systemic behaviors, the more we can anticipate them and enhance the system [19]. ST offers a range of tools for visually capturing and communicating about systems (e.g., causal loops diagram, stock & flow diagram). The use of diagramming techniques in the analysis of a system has a long history. Models provide an abstraction of the system, favoring representing and understanding it. In ST, diagramming techniques can generate insights and help identify the actions needed to solve a problem or change behaviors in the system [12]. Here, we refer to these diagramming techniques and tools that support system modeling as *ST modeling tools*.

A few years ago, two of this paper's authors had a successful exploratory experience using ST tools to solve Software Engineering (SE) problems in an organization. ST was very useful in understanding the organization, identifying factors affecting process and product quality, and defining strategies to address undesirable behaviors. As a result, product quality and project estimates were improved [15]. Given this experience and considering the ST potential to help improve SE processes and products, we searched the literature looking for secondary studies addressing ST in SE, aiming to better understand and advance this subject. As we did not find any, we decided to carry out a mapping study to investigate publications addressing the use of ST modeling tools in SE. A mapping study is a secondary study designed to give an overview of a research area through classification and counting contributions about the categories of that classification. It makes a broad study on a topic and aims to identify available evidence about it [23]. The panorama provided by a mapping study allows identifying issues that could be addressed in future research.

In our study, we found 10 papers, containing 9 different proposals. The results revealed that ST modeling tools have been employed to address SE issues in organizational contexts and to model and analyze the results of SE studies. The tools have helped get a holistic view and understand complex systems and identify actions to address problematic behaviors that affect SE outcomes. On the other side, it has been recognized that using the tools requires knowledge and may not be easy. They have been used more often in Project Management and Requirements Engineering context. Causal Loop Diagram has been the predominant tool, depicting cause-and-effect relationships and feedback loops.

This paper contributes to researchers by presenting a panorama about the use of ST modeling tools in SE and pointing out research opportunities that can be further explored. It serves as a starting point for other investigations into the subject as well as a motivation for advances in the research topic. As the paper summarizes evidence and refers to some cases of using ST modeling tools in practical sets, it can also inspire practitioners to use such tools.

This paper is organized as follows: Section 2 provides the background for the paper; Section 3 concerns the research protocol; Section 4 summarizes the obtained results; Section 5 discusses the results; Section 6 regards the study limitations; and Section 7 presents our final considerations.

2 BACKGROUND

A system is a set of elements related together and with the environment [11]. They form a complex and unified whole that has a specific purpose. The interrelation among the elements is crucial to characterize the system, i.e., without such interdependence, we have just a collection of parts, not a system [19]. System Thinking (ST) is based on the System Theory, which has had many elaborations in different disciplines such as biology, cybernetics, social, and management science. According to this theory, it is essential to deal with a system as a whole because the system behaviors and reactions would only be adequately understood by taking into account the whole system [11].

ST has been used in industry and academia for years [22, 26, 27]. It provides a set of tools to model systems and understand their different elements and behaviors. Next, we provide a brief description of the tools relevant to this paper.

Causal Loops Diagram (CLD) allows for representing the dynamics of a system by means of the system borders, relevant variables, their causal relationships, and *Feedback Loops*. A positive causal relationship means that two variables change in the same direction (e.g., increasing the number of bad design decisions increases the number of defects), while a negative causal relationship means that two variables change in opposite directions (e.g., increasing test efficacy decreases the number of defects).

Feedback Loops are mechanisms that change the variables of the system. There are two main types. A balancing loop is an equilibrant structure in the system and is a source of stability and resistance to change. A reinforcing loop compounds change in one direction with even more change[22, 27].

A beneficial effect of using CLD is that it helps identify archetypes. An *Archetype* is a common structure of the system that produces a characteristic pattern of behavior. For example, the archetype *Shifting the Burden* occurs when a problem symptom is solved by applying a symptomatic solution, which diverts from the fundamental solution [18]. *Fix that Fails*, in turn, occurs when an effective fix in the short-term creates side effects, a "fail", for the long-term behavior in the system [18]. Each archetype has a corresponding modeling pattern. Therefore, by analyzing a CLD is possible to identify archetypes by looking for their modeling patterns.

Figure 1 illustrates a fragment of CLD (adapted from [10]) created to a particular organization. The CLD contains three balancing loops, one reinforcing loop, and the *Shift the Burden* archetype (identified by the elements represented in blue). In this scenario, *Misunderstood requirements* contribute to increasing the number of *Defects in software artifacts*, which makes the organization mobilize (and often overload) the development team to fix defects by performing *New urgent development activities*, which decreases the number of *Defects in software artifacts*. These urgent activities are performed as fast as possible, aiming not to delay other activities. Thus, they do not properly follow good software quality practices. Moreover, they contribute to increasing the project cost and time (*Late and over-budget project*). *Defects in software artifacts* increase the need to use *Software quality techniques* that, when used, lead to fewer *Defects in software artifacts*. The archetype *Shifting the Burden* is composed of two balancing feedback loops and one reinforcing feedback loop. The balancing feedback loops (between *New urgent development activities* and *Defects in software artifacts*, and between *Defects in software artifacts* and *Software quality techniques*) mean that the involved variables influence each other in a balanced and stable way (e.g., higher/lower the number of *Defects in software artifacts*, more/less *New urgent development activities* are performed). In the reinforcing feedback loop, *New urgent development activities* are a symptomatic solution that leads to *Defects fixed through rework*, a side effect, because once urgent development activities fix the defects in software artifacts, the organization believes that the problem was solved. This, in turn, decreases the need for using *Software quality techniques*, which is a more fundamental solution. As a result, software artifacts continue to be produced with defects, overloading the development team with new urgent development activities [10].

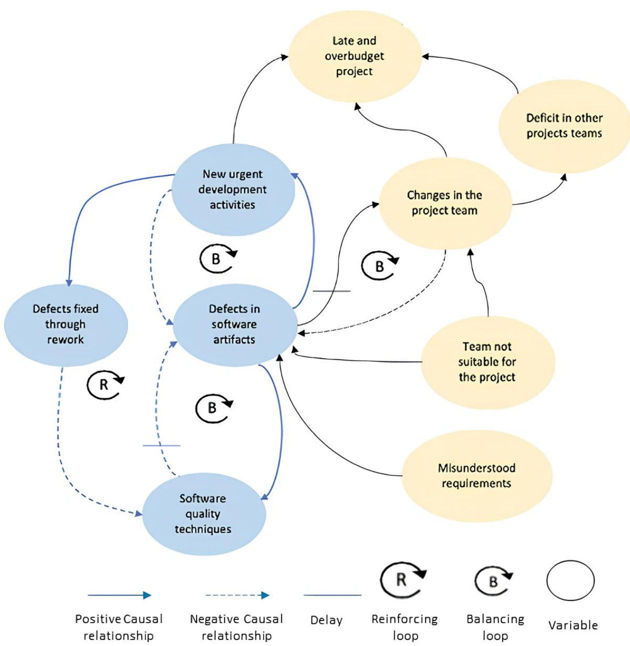

Figure 1: Example of CLD [10].

While CLD is useful for representing interactions between variables, it has limitations in capturing the structure of resource accumulation and flows in the system. *Stock & Flow Diagram* can be used for this purpose. Stocks represent accumulations that characterize the system's state, providing information for decision-making and actions. They impart inertia and memory to systems, creating delays by accumulating the difference between resource inflow and outflow. Flows represent the rates of resource inflow and outflow in the system, directly influencing the temporal evolution of stock variables [27].

Figure 2 illustrates a fragment of a Stock & Flow diagram representing the software development process (adapted from [8]). The software development process is conceptualized as transforming an initial stock of requirements that need to be developed (*Original Work to Do*) into a stock of developed requirements (not shown in the figure). The flows show how requirements move from the initial to the final stock.

Influence Diagrams serve as a means to identify and capture the significant relationships or influences among the elements of

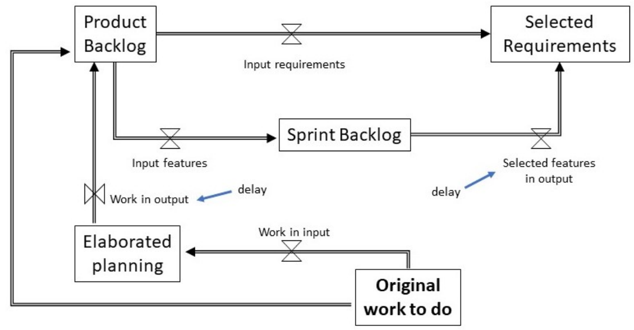

Figure 2: Example of Stock & Flow [8].

a system. They follow the information-action-consequence paradigm. Actions taken based on gathered information lead to consequences. These consequences, in turn, generate more information and actions[5].

Figure 3 represents a fragment of an Influence Diagram created to identify factors influencing the software quality (adapted from [2]). The fragment illustrated in the figure focuses on requirements. The *Amount of requirements changes* (changes made in the requirements specifications) and the *Amount of review* (reviewing requirements specifications) influence the *Amount of rework*, which is the effort spent on reworking both new and inadequate requirements. This variable directly influences the *Level of inadequate requirements*, which refers to the quality of requirements specifications. On the other hand, the *Level of schedule pressure*, which is the effect of the project falling behind the schedule, influences directly the *Level of inadequate requirements* and the *Amount of reviews* [2].

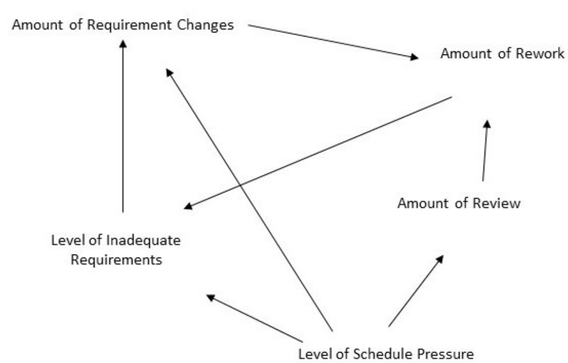

Figure 3: Example of Influence Diagram [2].

Last, *Simulators* represent simulation models or virtual environments. Analogous to flight simulators that enable pilots to safely develop their skills before applying them in the real world, simulators allow experimenting and testing strategies, structures, and decision rules in a controlled context [27].

3 RESEARCH PROTOCOL

Considering the successful use of ST in several fields and the need to gather knowledge about its use in SE to provide advances that contribute to SE processes and product improvement, we conducted a mapping study to investigate how ST modeling tools have been used in SE. Given that our focus is on the use of ST modeling tools in SE, works discussing ST in a broad sense, such as those discussing ST as a "way of thinking" holistically (e.g., [25] and [7]) are out of our scope. Moreover, we are interested in works clearly addressing SE. Thus, the ones discussing ST in other contexts (e.g., [1]) are not the focus of this study. To ensure methodological integrity and replicability, we followed the process outlined in [20], which consists of three phases, namely: planning, execution, and reporting.

The **goal** of this study is to investigate the use of ST modeling tools in SE. To achieve this goal, we defined the **research questions** presented in Table 1.

The **search string** utilized in this study consists of two groups of terms connected by the AND operator. The first group incorporates terms related to System Thinking joined by the OR operator to accommodate synonyms. The second group confines the study within the realm of Software Engineering. To formulate the string, we conducted tests using various terms, logical connectors, and combinations and selected the string that yielded better results in terms of both publication quantity and relevance. More restrictive strings (e.g., including only the terms "system thinking" AND "software engineering") omitted significant publications discovered in the preliminary literature review that preceded the study and were employed as control publications. More comprehensive search strings yielded a surplus of publications outside the scope of our interest. We used two papers([24, 30]) identified during the informal literature review as control publications. The search string used in this study is as follows: *((("system* thinking" OR "thinking system" OR "systems thinking" OR " system* map" OR "system* theory") AND ("software engineering" OR "software development" OR "software requirement" OR "software implementation" OR "software test" OR "software process" OR "agile development" OR (software AND "project management") OR (software AND "requirement* engineering")))*. In the second group, we included terms referring to some SE processes (e.g., implementation, test) to make the string more comprehensive. For that, we ran tests including several processes. Many of them did not lead to any new relevant publication and, thus, were not added to the string.

Scopus digital library served as the **source** for publications. It is a significant database of peer-reviewed literature that indexes papers from various sources, including IEEE, ACM, and Science Direct.

Publication selection was performed in five steps. Step 1 (S1) *Preliminary Selection and Cataloging*: publications are selected by applying the search string in the source search mechanism. (S2) *Duplications Removal*: duplicate publications are eliminated. (S3) *Selection of Relevant Publications – First Filter*: title, abstract, and keywords of the publications selected in (S2) are analyzed considering the following inclusion criterion and the exclusion criteria: (IC1) the publication approaches the use of ST modeling tools in the SE context; (EC1) the publication does not approach the use of ST modeling tools in the SE context; (EC2) the publication does not have an abstract; (EC3) the publication was published only as an abstract; (EC4) the publication is a secondary study, tertiary study, editorial, summary of keynote, tutorial or the proceedings of a scientific event; (S4) *Selection of Relevant Publications – Second Filter*: the full text of the publications selected in S3 is analyzed considering the criteria presented before and also the following ones: (EC5) the publication is not written in English; (EC6) the publication is an older version of another publication already considered; (EC7) it was not possible to have access to the full text of the publication; and (EC8) the publication was considered a violation of publishing principles according to the publication vehicle [2]. (S5) *Snowballing:* as suggested in [20], papers cited in publications selected in S4 were analyzed by following S2, S3, and S4. Publications containing results pertinent to the research topic were incorporated into the study.

We used the Parsifal [3] tool to support publication selection. To consolidate information and facilitate **data extraction**, the publications obtained in the selection steps were cataloged and recorded in spreadsheets. Each publication was assigned an id, and details such as title, authors, year, and publication venue were recorded. Data from publications returned in S4 were extracted and organized in a data extraction table tailored to address the research questions.

Publication selection and data extraction were carried out by the first three authors. Each publication was reviewed by at least two authors. Data extraction was reviewed by the fifth author and discussed until a consensus. Quantitative data was tabulated and used in graphs. The fourth and fifth authors reviewed the results.

Once the data was validated, the authors carried out data **interpretation and analysis**. Discordances were discussed and resolved. Finally, the five authors performed qualitative analysis considering the findings, their relation to the research questions, and the study purpose.

4 DATA EXTRACTION AND SYNTHESIS

Searches were conducted for the last time in March 2024 and the study considered papers published until 2023. The process followed and the number of publications selected at each stage are presented in Figure 4.

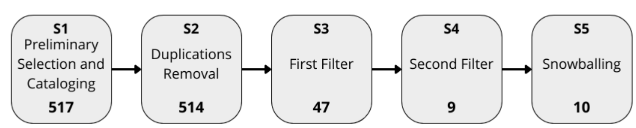

Figure 4: Publications selection.

In the 1st step (S1), 517 publications were identified. In the 2nd step (S2), duplicated publications were removed, resulting in 514 publications. In the 3rd step (S3), we applied criteria EC1, EC2, EC3, and EC4 considering the title, abstract, and keywords. This reduced

[2] EC8 was added to the set of criteria during the study execution because we found a publication in that situation
[3] https://parsif.al/

Table 1: Research questions and their rationale

ID	Research Questions	Rationale
RQ1	When and where have the papers been published?	Provide an understanding of when and where (journal/conference/workshop) publications addressing the use of ST modeling tools in SE have been published, to analyze the maturity of the research topic.
RQ2	Which types of research have been done?	Identify the research types that have been conducted, according to the classification defined in [28], namely: Evaluation Research; Proposal of Solution; Validation Research; Philosophical Paper; Opinion Paper; and Personal Experience Paper. This information will help analyze the maturity of the research topic and the use of ST in practical settings.
RQ3	Which ST modeling tools have been used?	Investigate which ST modeling tools have been used in SE, verify which ones have been predominant, and identify gaps and opportunities to apply ST modeling tools.
RQ4	How have ST modeling tools been applied in SE?	Investigate how ST modeling tools have been used in SE (e.g., if they are used in combination, in the context of any approach or process, etc.) and the purposes for using them.
RQ5	Which SE processes have been supported by the use of ST modeling tools?	Investigate the SE processes (e.g., requirements engineering, coding, testing) that have been supported by ST modeling tools, verify the range of addressed processes and if there has been a predominance of any of them.
RQ6	Which difficulties have been reported on the use of ST modeling tools in SE?	Identify challenges faced in the use of ST modeling tools in SE to reflect on research directions to address them.
RQ7	Which benefits have been perceived in the use of ST modeling tools in SE?	Identify the advantages of using ST modeling tools in SE and verify which ones have been more prominent.

the number of papers to 47, representing a reduction of approximately 91% from the previous step. In the 4th step (S4), we applied all the inclusion and exclusion criteria considering the full text. As a result, nine relevant papers were identified. Finally, in the 5th step, we performed a backward snowballing by checking the references of the nine selected publications. In this step, we analyzed 183 publications and selected a new one, which in total added up to 10 publications. The publication found during snowballing referred to a proposal addressed by a publication previously selected in S4. We did not exclude it by using EC6 because the publications are complementary. When extracting data about the proposal addressed in these publications, we considered both of them. Thus, although we have selected 10 publications, they present nine different proposals. The selected proposals and respective publications are presented in Table 2. Next, we present the data synthesis for each research question. Given the identification of fewer than 10 proposals, we refrained from presenting percentage values in tables and graphs. Moreover, for simplification reasons, when we refer to percentages in the text, we rounded the values to integer numbers. Details about the study, including the research protocol, selected publications, and extracted data are available in the study package [9].

Publication year and type (RQ1): Figure 5 illustrates the distribution of publications over the years and their vehicle. In this question, we considered the publications individually (i.e., 10 publications). Conferences have been the predominant publication venue, comprising 8 publications (80%), while journals accounted for 2 publications (20%).

Research type (RQ2): From this question forward, we considered [13], and [14] as a single publication (#5), resulting, thus, in the nine proposals listed in Table 2. Table 3 shows the number of publications considering their research type according to the classification proposed in [28]. All the nine selected publications propose a solution to a problem and argue its relevance, i.e., they are classified as *Proposal of Solution*. Five of them (around 55%) performed

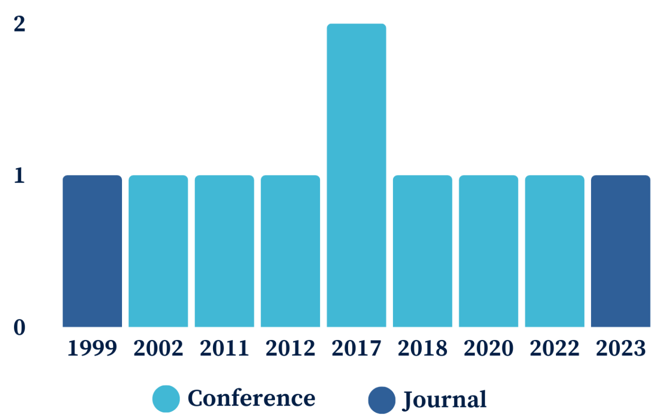

Figure 5: Publication year and vehicle.

some kind of evaluation. From these, three (around 33%) were evaluated in practice (i.e., also classified as *Evaluation Research*) and two (around 22%) evaluated characteristics of the solution not yet implemented in practical settings (i.e., *Validation Research*).

ST Modeling Tools (RQ3): Six ST modeling tools were identified in the selected publications. *Causal Loop Diagram* was used in eight of the nine selected publications, being the predominant tool (approximately 89%). It is followed by *Feedback Loops*, which were utilized in five publications. *Simulator* was used in four publications, *Influence Diagram* in two, while *Stock & Flow Diagram* and *Archetypes* were applied in one publication. Seven publications (around 78%) combined different modeling tools. Figure 6 illustrates these results.

Ways and purposes for using ST Modeling Tools (RQ4): The tools have been used in different ways and with several purposes, as it is shown in Table 4. The use of ST modeling tools to

Table 2: Selected publications.

ID	Brief Description	Vehicle	Ref.
#1	Discusses the use of ST as a problem-solving technique in Requirements Engineering. A model-based process simulation framework that uses causal loop diagrams and feedback loops is proposed to help understand the Requirements Engineering process effectiveness and evolution over time.	Software Quality Journal	[29]
#2	Discusses the use of a simulator (simulation model) based on causal loops and influence diagrams to analyze variables involved in software development processes (e.g., requirements elicitation, modeling). The model visualizes relations between software process variables and can be used by project managers when planning projects. The model was used in a case study to analyze the relationship between effort, lead time, and product quality.	International Conference and Workshop on the Engineering of Computer-Based Systems	[2]
#3	Presents a study in which ST tools (causal loop diagram, stock & flow diagram, and simulation) were used to analyze the dynamic behavior of the adoption of Kanban and Scrum, versus the Waterfall approach. A model based on the relationships between system variables was built to describe the behavior of the different approaches under similar starting conditions and enable the assessment of the relative benefits of each of them.	International Conference on Agile Development	[8]
#4	Discusses the need to analyze the domain when developing software to avoid the gap between the system and its supporting software. A causal loop diagram is defined to represent the topological model of functional characteristics.	International Conference on Evaluation of Novel Approaches to Software Engineering	[3]
#5	Presents a study that investigated factors that influence productivity in agile teams in Bangladesh. The authors identified the factors through interviews and used influence and causal loop diagrams to represent the results, showing the complex interrelated structure of different factors affecting agile teamwork productivity.	Asia-Pacific Software Engineering Conference; International Conference on Software Engineering Advances	[13], [14]
#6	Presents a study in which causal loop diagrams and feedback loops are used to provide a holistic view of the software development process before applying either Agile or Lean methods. The software development process was mapped as a collection of feedback loops.	International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management	[6]
#7	Proposes a ST-based process that uses causal loop diagrams and archetypes to support organizations in the transition from traditional to agile software development. The process resulted from a participative case study in a Brazilian organization that performs software projects in partnership with a European organization.	Brazilian Symposium on Software Engineering	[24]
#8	Establishes a causal loop diagram for adapting complex software testing situations. The diagram serves as a simulation model that aims to contribute to increasing software quality by helping software developers and engineers make decisions considering the pressure of software releases schedule and the constraints of testing costs. Simulations were conducted to validate the model.	International Conference on Computation, Communication and Engineering	[16]
#9	Presents 5P, a model for software project measurement that aims to achieve optimization and global control. By using a ST approach, the proposed model enables global optimization by aligning project objectives, product outcomes, and stakeholders' needs. The model was evaluated in a case study.	Software Computing Journal	[30]

Table 3: Research Type.

Research Type	Publications
Proposal of Solution	#1, #4, #5, #6
Proposal of Solution & Evaluation Research	#2, #7, #9
Proposal of Solution & Validation Research	#3, #8

run simulations is the only answer that appeared more than once (publications #1, #2, #3, and #8). Moreover, ST modeling tools have been applied to address SE problems in organizational contexts and also to model and analyze SE studies (#5 and #6).

Supported SE Processes (RQ5): The processes that ST modeling tools have addressed more often are Project Management and Requirements Engineering. ST modeling tools were applied considering the Software Development process as a whole in two publications. Software Measurement and Software Testing were

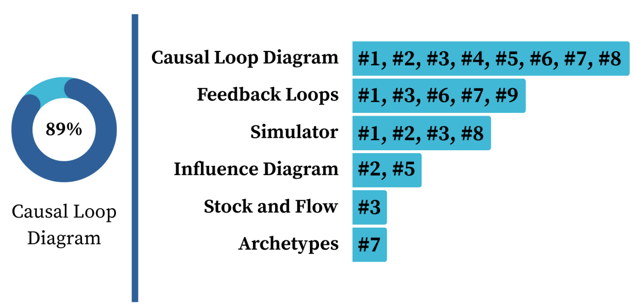

Figure 6: ST modeling tools.

Table 4: Use of ST Modeling Tools.

ID	How	For What
#1	Causal loop diagram and feedback loops were used in a model-based process simulation framework.	Illustrate and understand interactions among factors that affect the Requirements Engineering process.
#2	Causal loop and influence diagrams were used as a basis for simulations.	Analyze the relation between effort, lead time, and product quality and support project planning.
#3	Causal loop diagram, feedback loops, and stock & flow diagram were used as a basis for a simulation model.	Model the internal behavior of software development approaches (Kanban & Scrum and Waterfall), identify factors impacting error rate in the different approaches, and compare them.
#4	Causal loop diagram was used with mathematical abstraction in a topological model.	Understand the software features and the relation among them.
#5	Causal loop diagram and influence diagrams modeled survey results.	Examine the dynamics existing within team and organizational resources and understand relationships among factors that affect agile team productivity.
#6	Causal loop diagram was used as a collection of feedback loops mapping the software process.	Provide a holistic view of the software development process and understand factors affecting it negatively.
#7	Causal loop diagrams, feedback loops, and archetypes were combined with GUT matrix and ontologies in a systematic improvement process.	Identify undesirable behaviors and their causes to help identify leverage points to implement agile practices.
#8	Causal loop diagram was used in a model representing complex software testing.	Adapt complex testing situations using simulations and help improve software testing planning.
#9	Three feedback loops were used in a model to support software project measurement.	Align project objectives, deliverables, and stakeholders' needs and aid in software measurement.

the targets of ST modeling tools in one publication each. Table 5 summarizes these results. Some of the identified processes appear explicitly in our search string. This could lead to questions about its influence on these results. We must clarify that, as we explained in Section 3, we tested terms referring to several other processes, and the final results did not change.

Table 5: Supported SE Processes.

Processes	Publications
Project Management	#2, #3, #5, #7
Requirements Engineering	#1, #2, #4
Software Development	#3, #6
Software Testing	#8
Software Measurement	#9

Challenges (RQ6): Table 6 summarizes the identified challenges. Some publications present explicitly the challenges faced when using the tools. For example, in #7 it is said *"it involves a lot of tacit knowledge and judgment. Besides knowledge about System Thinking tools [...], it is necessary to have organizational knowledge to apply them"*. Other publications are not very explicit. In these cases, we analyzed the text to extract the challenges. After extracting the challenges, we classified the equivalent ones in the same category. In several publications, the challenges are not mentioned at all. These cases were considered in the "not informed" category. As can be noticed in Table 6 , the main challenges are knowledge-related and highlight that to properly use ST modeling tools, it is necessary to know the tools, the organization, and the domain of interest.

Benefits (RQ7): Table 7 shows the identified benefits. We adopted the same data extraction procedure followed in RQ6. The main benefit indicates that ST modeling tools help understand and improve

Table 6: Challenges in the use of ST Modeling Tools.

Challenges	Publications
Not Informed	#1, #5, #6, #9
It demands knowledge of the ST modeling tools	#2, #3, #4, #7
It demands knowledge of the organization or domain of interest	#2, #3, #4, #7
Dealing with complexity and high volume of variables	#3
Reducing bias when making assumptions to run simulations	#3
Identifying the cause-and-effect relationships	#4
It may be hard to use the tools in a large scope	#7
It demands effort and is time-consuming	#7
Obtaining relevant parameter values in simulations	#8

software processes. It is also pointed out their support in understanding the relationship among the software process variables, representing and understanding complex systems. Moreover, the tools help identify problematic behaviors and actions to address them.

5 DATA ANALYSIS AND DISCUSSION

In this section, we provide additional information about the analyzed approaches and discuss the results.

Analyzing ***when and where the studies were published*** **(RQ1)**, we noticed that although we have found a limited number of papers, the topic has been under investigation for approximately a decade. Most of the papers (80%) were published in conferences, which usually require less mature works than journals. The findings related to ***research type*** **(RQ2)** indicate that all the selected publications

Table 7: Benefits of using ST Modeling Tools.

Benefits	Publications
Help understand and improve processes	#1, #2, #3, #9
Help represent and understand complex systems	#4, #5, #8
Help understand the relationships among software process variables	#2, #3, #6
Help identify problematic behaviors	#6, #7
Help identify actions/strategies to address problematic behaviors	#6, #7
Help understand complex relations in long chains of cause and effect	#2
Help improve communication among different stakeholders	#1
Serve as an input for automated transformation	#4
Help understand the organization's behaviors	#7
Help identify leverage points	#7
Provide a dynamic perspective of the organization	#7
Avoid complex mathematical derivations	#8

propose solutions for SE problems. 55% incorporate some kind of evaluation, but only 33% were used in practical environments. This may be a sign of difficulty in applying ST modeling tools in the industry, which reinforces that research on this topic is not yet mature and there appears to be a gap between theory and practice that should be further explored.

Concerning the employed ***ST modeling tools*** **(RQ3)**, CLD emerged as the most used tool (eight out of nine proposals used it). This suggests a concern with getting a holistic view of the organization (or domain of interest) by understanding how different variables affect each other. Four of these proposals also used feedback loops, which facilitate understanding changes in the variables and, thus, support a deeper analysis of the system. Given that feedback loops are often used in the CLD context, it seems that there is room to make the most of using this tool when using CLD. Analogously, although CLD has been the predominant tool, only one work (#7) used archetypes. This tool helps identify patterns of behavior and define proper strategies to address them. Once CLD is built, it is possible to look at the diagram and identify modeling patterns. For instance, if "Shifting the Burden" is identified in the CLD built for an organization, it reveals that three variables are related to each other in such a way that an undesirable behavior has been treated by applying a symptomatic solution, which diverts attention away from the proper solution. Thus, the organization can deviate from the symptomatic solution and take action towards the proper one. CLD has also been used as a basis for building simulation models to make predictions. The predominance of CLD suggests that the approaches have preferred the use of tools that enable a global vision of the system by means of its variables. This certainly contributes to understanding and analyzing SE environments. Moreover, CLD is usually easier to build than other tools (e.g., Stock & Flow). However, the use of other tools could be further explored.

By analyzing ***how ST modeling tools have been applied*** **(RQ4)**, it is possible to notice that most of the publications (seven out of nine) combined different tools. This is indeed a good strategy because the tools can be used in a complementary way, providing different perspectives of the system. Thus, applying different tools helps to observe and understand behaviors that may not be fully revealed by any single tool alone. ST modeling tools have been used in different ways and for diverse purposes. There is a predominance of combining models aiming to run simulations (four out of nine publications). This was in some way expected because some diagrams (e.g., CLD, stock & flow) model the system variables, flows, and the relationship among them, providing useful information and models to run simulations that support making predictions. Moreover, simulations help automatize tasks and, thus, the interest in using them may be higher than in using only the diagrams. Apart from this commonality, ways of and reasons for using ST modeling tools have been quite diverse (see Table 4). This diversity can be understood as evidence of the wide range of applications ST modeling tools have, suggesting that they have the potential to address various SE problems.

Another aspect that reinforces this perception and deserves attention is the use of the tools to deal with SE issues in different contexts. Some works have applied the tools to address SE problems in specific organizations. For example, in #7, CLD, feedback loops and archetypes were used combined with GUT matrix and ontologies to help understand the organization, identify undesirable behaviors, their causes, and leverage points where agile practices should be implemented to produce better results. Other works, in turn, instead of focusing the use of the tools on any particular organization, apply them to model and analyze SE aspects in general. For example, the authors of #5 conducted interviews in Bangladesh to investigate factors that influence productivity in agile teams and used CLD and influence diagram to model and analyze data. In #3, the tools were used to analyze Kanban & Scrum and Waterfall approaches behavior. CLD was used to identify cause-and-effect relationships and stock & flow diagrams represented the approaches in terms of stocks, inflows, and outflows. The models were used to build a simulation model to compare the approaches, helping identify their distinctions and determine pros and cons. Therefore, using ST tools can be a powerful approach to better understand not only specific organizations but also broader aspects of SE and explore a wide range of applications.

As for ***SE processes supported by ST modeling tools*** **(RQ5)**, Project Management and Requirements Engineering processes have been predominant. We found this result quite understandable because these processes have a wide spectrum of variables and involve many different aspects that should be considered when addressing problems. For example, Project Management works as an umbrella process covering the entire development process as well as support processes (e.g., reuse, quality assurance) – aspects related to all these processes in some way affect project management results. Requirements Engineering also involves multiple variables of the organizational environment that influence the quality of requirements (e.g., several stakeholders, communication aspects, domain understanding, and skills involved in developing requirements). Hence, a holistic and systemic view of the system favors understanding these processes and addressing problems. Three of the four publications addressing Project Management used simulations (this is not a surprise because simulations help make predictions). Other processes have also been the target of using ST modeling tools, such as Measurement and Test. In the former, feedback loops helped align goals, stakeholders' needs, and deliverables. This alignment is crucial to plan the metrics to be used. In the latter, CLD and simulations were used to model complex test situations and support test planning.

Given that SE involves a lot of processes, we believe that the use of ST modeling tools also should be explored in other processes. For instance, CLD and archetypes could help organizations identify and treat undesirable behaviors in the coding process (e.g., behaviors that cause defects injection and code smells). Moreover, given that ST modeling tools help provide a holistic view, they could be further explored in the context of many processes at the same time. For example, they could be used to represent the relationships among social (e.g., collaboration, friendship, network, willingness, happiness) and technical (e.g., defects, design choices) aspects to investigate how they influence software quality and team productivity, among others.

Regarding ***challenges faced in using ST modeling tools*** **(RQ6)**, the need for knowledge of the tools and of the organization or domain of interest are the ones more prominent (identified in four of nine publications). These challenges suggest that there is still a gap between theory and practice. In the literature, there are several books, papers and other materials providing knowledge about ST tools. However, the results may indicate that these knowledge sources have not properly reached the interested parties or they have not been enough. Hence, it would be relevant to investigate the causes of this gap and take action to minimize it. Moreover, to effectively use ST modeling tools, it is crucial to know not only the tools but also the domain or organization to be modeled. The lack of such knowledge hampers the holistic understanding of the application context and suggests that people have had difficulty getting a broader view of the system. In fact, many people often perform their tasks focused solely on the immediate context, without concern with the global perspective. Therefore, it is necessary to amplify software engineers' perspectives of how an organization (or domain of interest) works. Six other challenges were cited only once. Some of them are related to specific tools used in the publications. For example, referring to simulations, #3 points out that reducing bias is not trivial, while #8 says that obtaining relevant parameters may not be easy. Four out of nine publications did not mention any challenge. We believe that this probably does not mean difficulties have not been experienced, but that the authors did not discuss them in the paper.

Finally, considering the ***benefits of using ST modeling tools in SE*** **(RQ7)**, four out of nine publications indicated that the tools have helped understand and improve processes. Three publications went a little deeper pointing out the understanding of the processes variables, which certainly contributes to understanding the processes. Other three publications recognized the aid of ST modeling tools in understanding complex systems, such as software features (#4), teams (#5), and complex testing situations (#8). The predominance of benefits referring to process can be justified by the fact that many publications used the tools in the context of a process (e.g., Requirements Engineering (#1) and Measurement (#9)) and, thus, the process is turned into the system being analyzed. Having said that, we could understand the three most cited benefits as a more generic one "help understand the system", which is a benefit ST aims to provide, regardless of the application area [22] [17].

When considering these benefits and the main identified challenges, we observe that they all are in some way related to knowledge. On one hand, the challenges highlight that knowing the application context is not easy. On the other hand, the benefits point out that ST modeling tools help get knowledge of the application context. Therefore, the models built by using ST tools help fill the gap of knowledge about the application context. Nevertheless, prior knowledge to build the models is still needed. By helping understand the system, ST tools enable the identification of undesirable behaviors and the definition of corrective actions. These benefits were perceived in two publications. One of them, also indicated that ST tools are useful in identifying leverage points so that corrective actions can focus on them and provide better results. Other benefits appeared only once. Even so, they reveal relevant advantages of using ST tools, such as improving communication among stakeholders and offering a dynamic perspective of organizations. Some benefits are related to specific tools. For example, simulators can be an alternative approach to avoid complex mathematical derivations, while CLD helps understand long chains of cause and effect and, when used to build simulators, serve as input to automation.

Based on the panorama provided by the study results and the limited number of identified publications, we noticed that the use of ST modeling tools in SE is a relatively unexplored and promising subject. In summary, CLD has been the tool used more often. Even so, its combination with other tools (e.g., feedback loops and archetypes) to get a holistic view of the system and understand its behaviors has been neglected. Although some processes have benefited from the use of ST modeling tools (particularly Project Management and Requirements Engineering) many other processes could take advantage of such tools. The tools have been used in different ways and for several purposes, suggesting that a wide range of applications is possible. We believe that the benefits provided by the use of ST modeling tools should be taken as a motivation to address the challenges of using them and advance the topic.

6 LIMITATIONS OF THE STUDY

Any study has inherent limitations and researchers face challenges (e.g., how to consistently apply selection criteria, and classify and interpret data) that must be considered with the results. Next, we discuss some of these aspects in our study.

The subjectivity embedded in publication selection and data extraction is a limitation of our study. To deal with it, the activities were performed by the three first authors and reviewed by the fifth author. First, the researchers selected a set of publications and analyzed them until reached a consensus on the results. After that, the publications were divided among the three first authors and all of them played "executor" and "reviewer" roles. Each publication was analyzed against the criteria by one researcher (executor) and reviewed by the other two researchers (reviewers), who independently executed the same procedures, and the outcomes were compared. To assess agreement, we used the kappa coefficient [21], which measures the agreement degree between two researchers in qualitative evaluation. Given that publication selection was made by three researchers in different combinations of executer and reviewer, we calculated the kappa coefficient for each combination of two researchers. The lower value of kappa was 0.83. According to Landis and Koch [21], this means perfect agreement. Any discrepancies in publication selection were addressed through discussion and resolution. Data extraction was also performed by the three first authors playing executor and reviewer roles. In the end, extracted data was reviewed by the fifth author, refined, and discussed with all the other authors until a consensus.

Another limitation refers to the search string. The absence of some terms may have led to miss relevant publications. To mitigate this, we ran several tests and compared the results until we reached the defined string. Some terms that at first seemed to be relevant, after being tested were not included in the search string. For example, by adding terms referring to system dynamics, only three new papers were added and none of them passed the publication selection filters.

The use of a single source is also a limitation. To reduce this, we conducted backward snowballing. However, we did not perform forward snowballing (i.e., look for relevant publications by analyzing the ones citing the publications selected in the study) and this has to be considered as a limitation that can cause relevant publications not to be captured. Moreover, we decided not to search any specific conference proceedings, journals, or grey literature. Thus, we have worked exclusively with publications indexed by the selected electronic database and the ones identified during snowballing. The exclusion of these other sources makes the review more repeatable, but possibly some valuable publications may have been left out of our analysis.

The classification of data in categories in some research questions also represents a limitation. Some of the categories were based on classifications previously proposed in the literature (e.g., research type [28]). Others were established during data extraction (e.g., challenges, benefits), based on data provided by the selected publications. Determining the categories and how publications data fit them involves a lot of judgment. Therefore, different results could be obtained by other researchers.

Finally, another limitation regards data interpretation, which involves specific knowledge and judgment, potentially leading different researchers to get different conclusions. To minimize this threat, data was represented in spreadsheets, tables, and graphs, and interpretation was performed by the authors iteratively, considering the research questions. In this way, complementary interpretations were combined and different interpretations were discussed. Moreover, when writing this paper, we went back to the data and results and analyzed them once more, looking for additional information or new perceptions that we could have missed.

7 FINAL CONSIDERATIONS

Systems Thinking involves viewing a system (e.g., an organization, a process) not only from an individual subjective perspective or interest but also from a broader meta-perspective. It is about stepping back from the details to grasp the big picture. When applying ST, it is necessary to think of the causes, effects, and controls at a higher abstraction level. For that, ST provides us with a set of tools. As the world becomes ever more tightly interwoven globally and as the pace of change continues to increase, we will all need to become increasingly "system-wise" [19].

ST has been successfully used in several areas for years. In this paper, we investigated its use in the SE context by focusing on the use of ST modeling tools. A total of 700 publications were considered, and nine proposals were selected. Seven research questions were defined to investigate the following facets: (i) distribution of the selected publications over the years and the type of vehicle; (ii) research type; (iii) ST modeling tools used in SE; (iv) ways of and purposes for using these tools; (v) supported SE processes; (vi) difficulties; and (vii) benefits of using ST modeling tools.

The study contributes by providing an overview of research related to the topic. In summary, ST modeling tools have been employed to tackle SE issues in organizational contexts and to model and analyze SE general aspects addressed in SE studies. Project Management and Requirements Engineering have been the most common processes in which these tools have been applied. ST modeling tools have helped get a holistic view of complex systems, understand and improve processes, identify problematic behaviors that affect SE outcomes, and define strategies to address them. CLD has been the predominant tool. The lack of the necessary knowledge for using the tools, which involves knowledge of the tool and application context (organization or domain of interest) has been considered a challenge.

We believe that the results indicate that using ST tools in SE holds promise. However, they also suggest that the tools have been under-explored and there is room for further investigation into their potential of contributing to SE products and processes improvement. Therefore, it is necessary to carry out studies aiming at advancing ST in SE and providing evidence of the effects of using ST modeling tools to enhance SE processes and products.

As future work, a broader investigation into the use of ST in SE considering not only modeling tools could complement this study. Moreover, comparing ST use in SE with its use in other areas would help inspire new applications and advances in the research topic. Considering this study's results, particularly the fact that lack of knowledge has been pointed out as the main difficulty in using ST modeling tools, we have driven efforts to define guidelines and develop tools to help software engineers use some ST tools (e.g., CLD and archetypes). Our goal is to take a step forward to provide advances to bridge the gap between ST modeling tools and the knowledge necessary to use them in SE.

ARTIFACT AVAILABILITY

The study package, containing the research protocol, selected publications, and extracted data is available at [9].

ACKNOWLEDGMENTS

This research is supported by the Coordination for the Improvement of Higher Education Personnel - CAPES Brazil (Finance Code 001) and the Espírito Santo Research and Innovation Support Foundation - FAPES (Processes 2023-5L1FC, 2022-NGKM5, 2021-GL60J, and T.O. 1022/2022).

REFERENCES

[1] John F Affeldt. 1999. The application of system dynamics (SD) simulation to enterprise management. In *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 2*. 1496–1500. https://doi.org/10.1109/WSC.1999.816885

[2] Carina Andersson, Lena Karlsson, Josef Nedstam, Martin Host, and Bertil I Nilsson. 2002. Understanding software processes through system dynamics simulation: a case study. In *Proceedings Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*. IEEE, 41–48. https://doi.org/10.1109/ECBS.2002.999821

[3] Erika Asnina, Janis Osis, and Asnate Jansone. 2012. System thinking for formal analysis of domain functioning in the computation independent model. *ENASE 2012 - Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering* 2 (2012), 232–240. https://doi.org/10.5220/0004090602320240

[4] Monalessa Perini Barcellos. 2020. Towards a Framework for Continuous Software Engineering. In *Proceedings of the 34th Brazilian Symposium on Software Engineering* (Natal, Brazil) *(SBES '20)*. Association for Computing Machinery, New York, NY, USA, 626–631. https://doi.org/10.1145/3422392.3422469

[5] Stuart Burge. 2015. The systems thinking tool box. *Burge, Hughes, Walsh* (2015), 1–5.

[6] Phoebe Mae Ching and Jose Edgar Mutuc. 2018. Evaluating agile and lean software development methods from a system dynamics perspective. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. IEEE, 1–6. https://doi.org/10.1109/HNICEM.2018.8666338

[7] Paolo Ciancarini., Mirko Farina., Artem Kruglov., Giancarlo Succi., and Ananga Thapaliya. 2023. A Reflection on the Use of Systemic Thinking in Software Development. In *Proceedings of the 18th International Conference on Evaluation of Novel Approaches to Software Engineering - ENASE*. INSTICC, SciTePress, 521–529. https://doi.org/10.5220/0011963300003464

[8] Luisanna Cocco, Katiuscia Mannaro, Giulio Concas, and Michele Marchesi. 2011. Simulating kanban and scrum vs. waterfall with system dynamics. *Lecture Notes in Business Information Processing* 77, 117–131. https://doi.org/10.1007/978-3-642-20677-1_9

[9] Julia de Souza Borges, Thiago Felippe Neitzke Lahass, Paulo Sérgio Santos Júnior, Amanda Apolinario, and Monalessa P. Barcellos. 2024. Supplementary material of the study "Unveiling the Landscape of System Thinking Modeling Tools Use in Software Engineering". (2024). https://doi.org/10.6084/m9.figshare.25647963.v3

[10] Paulo Sérgio dos Santos Júnior, Monalessa Perini Barcellos, and Rodrigo Fernandes Calhau. 2022. First step climbing the Stairway to Heaven Model-Results from a Case Study in Industry. *Journal of Software Engineering Research and Development* 10 (2022), 5–1. https://doi.org/10.5753/jserd.2021.1992

[11] Manfred Drack and Wilfried Apfalter. 2007. Is Paul A. Weiss' and Ludwig von Bertalanffy's system thinking still valid today? *Systems Research and Behavioral Science: The Official Journal of the International Federation for Systems Research* 24, 5 (2007), 537–546. https://doi.org/10.1002/sres.855

[12] M Fakhimi, D Robertson, and T Boness. [n.d.]. THE BASIC PRINCIPLES OF SYSTEMS THINKING AND SYSTEM DYNAMICS. ([n. d.]). https://doi.org/10.36819/SW21.003

[13] Israt Fatema and Kazi Sakib. 2017. Factors Influencing Productivity of Agile Software Development Teamwork: A Qualitative System Dynamics Approach, In 24th Asia-Pacific Software Engineering Conference (APSEC). *Proceedings - Asia-Pacific Software Engineering Conference, APSEC* 2017-December, 737–742. https://doi.org/10.1109/APSEC.2017.95

[14] Israt Fatema and Kazi Muheymin Us Sakib. 2017. Analyse agile software development teamwork productivity using qualitative system dynamics approach. *ICSEA 2017* (2017), 71.

[15] Omitted for sake of anonimization. [n.d.].

[16] Tao Huang and Chih-Chiang Fang. 2022. Applying System Dynamics Approach for Optimizing Software Release Decisions. In *2022 2nd International Conference on Computation, Communication and Engineering (ICCCE)*. IEEE, 54–57. https://doi.org/10.1109/ICCCE55785.2022.10036236

[17] P. John Sterman. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World with CD-ROM*. McGraw-Hill Education.

[18] DH Kim. 1993. System Archetypes I: Diagnosing Systemic Issues and Designing High-Leverage Interventions.

[19] D.H. Kim. 2018. *Introduction to System Thinking*. Pegasus Communications. 1–21 pages.

[20] B Kitchenham, L Madeyski, and D Budgen. 2022. Supplementary material for SEGRESS: Software engineering guidelines for REporting secondary studies. *IEEE Transactions on Software Engineering* (2022). https://doi.org/10.1109/TSE.2022.3174092

[21] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.

[22] Donella H Meadows. 2008. *Thinking in systems: A primer*. chelsea green publishing.

[23] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology* 64 (2015). https://doi.org/10.1016/j.infsof.2015.03.007

[24] Paulo Sérgio dos Santos, Monalessa P Barcellos, and Rodrigo Fernandes Calhau. 2020. Am i going to heaven? first step climbing the stairway to heaven model results from a case study in industry. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. 309–318. https://doi.org/10.1145/3422392.3422406

[25] Mali Senapathi and Meghann L. Drury-Grogan. 2021. Systems Thinking Approach to Implementing Kanban: A case study. *Journal of Software: Evolution and Process* 33, 4 (2021), e2322. https://doi.org/10.1002/smr.2322 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/smr.2322 e2322 JSME-20-0112.R1.

[26] John Sterman. 1994. Learning in and about complex systems. Systems Dynamics Review, 10, 1994. , 291–330 pages.

[27] J. Sterman. 2010. *Business Dynamics: Systems Thinking And Modeling For The Complex World*. Tata McGraw Hill Education Private Limited. https://doi.org/10.1057/palgrave.jors.2601336

[28] Roel Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements engineering* 11 (2006), 102–107. https://doi.org/10.1007/s00766-005-0021-6

[29] DW Williams, Tracy Hall, and Michael Kennedy. 1999. A framework for improving the requirements engineering process management. *Software Quality Journal* 8 (1999), 133–147. https://doi.org/10.1023/A:1008956910828

[30] Zhimin Zhao, ShouXi Deng, Yaofei Ma, Qiang Gao, and Na Zhao. 2023. Software project measurement based on the 5P model. *Soft Computing* 28 (2023), 1–23. https://doi.org/10.1007/s00500-023-09175-9