Systemic thinking and software development processes

Paolo Ciancarini
Università di Bologna
Bologna, Italy
paolo.ciancarini@unibo.it

Mirko Farina
Innopolis University
Innopolis, Russia
m.farina@innopolis.ru

Giancarlo Succi
Università di Bologna
Bologna, Italy
g.succi@unibo.it

Ananga Thapaliya
Innopolis University
Innopolis, Russia
a.thapaliya@innopolis.ru

***Abstract*—The mind is undoubtedly the most critical resource for software engineers, making understanding how it functions a crucial scientific endeavor that can enhance the efficiency of their work. This paper posits that software engineers who work in a collaborative environment tend to think systematically, and that Agile methodologies, which are widely adopted and considered effective, foster systemic thinking.**

This study, based on an online survey with over 100 responses, was conducted as an observational study [1]. Our findings, which align with our initial hypothesis, suggest that software engineers working in a supportive environment tend to adopt systemic thinking and that this type of thinking is emphasized even more in Agile environments.

As an observational study, this research lays the groundwork for future confirmatory studies that can use appropriate inferential methods to determine the validity of our hypotheses.

1. Introduction

Since our early programming experiences with computers, we have been taught that to solve complex software engineering problems, we need to divide them into smaller sub-problems, address each sub-problem separately, find a solution, and then combine these solutions into a comprehensive solution. We have also learned that if a sub-problem is complex, we should apply this approach recursively [2]. This strategy is known as *divide and conquer* [3][1].

This works well for the so-called "tame" problems, that is, problems easy to decompose into subproblems, which can require a lot of effort, but which are not intrinsically complex [4]. The sad story is that there are also "wicked problems," those that cannot be easily decomposed into sub-problems, whose different facets are all interdependent with one another, and whose solutions require comprehensive global thinking at all their facets. For such problems, different approaches are required. The reality is that most software problems are wicked, almost all, apart from trivial ones, which do not deserve the attention of managers, researchers, and software engineers [5]. Still, software engineering textbooks and manuals of work mostly operate tackling problems as "tame problems" and developers and managers often blindly think that the concept of divide and conquer is the "right" way to go, ignoring that it now almost five decades that it has been evidenced that the typical divide and conquer models (waterfall, V-shape and incremental) feature a high amount of risks and uncertainty [6].

1) This type of approach has been found to work in the context of a strong autocratic decision structure but not in problems with respect to corporates and communities [7].
2) It expects the Project Manager to be rational, understand the problem as a whole, and anticipate the consequences of the decisions he/she makes. Recent research has shown that human beings are not as rational as it was thought to be. This has spawned a whole new field related to behavioral studies [8].
3) This approach fails when the steps cannot be followed in a sequence. for e.g. when the requirements for a project change rapidly or that development has to be speeded up, waterfall, V-shape or incremental falls short [7].
4) These approach fails when understanding the problem as a whole and anticipating the consequence of decisions is not possible [8].

This paper focuses mostly on this last point. It is common that software engineers use traditional "divide and conquer" approaches to solve wicked software development problems by first following the mandated approach and then, at the end, when they see that they cannot reach the desired target in the prescribed "clean" way, hacking solutions, considering systems as a whole and hammering the software left and right until it satisfies the final acceptance test of the customer. This latest stage is often done (at least partially) secretly, with a bit of shame, and it is often named "*cowboy coding*" [9].

In the last quarter of a century, the concept of agile methods has emerged. They are "formally" rooted in the concept of lean management but what interests us now is that they somehow depart from the concept of divide and conquer.

1. It is interesting to note that the term "divide and conquer" is sometimes incorrectly associated with the Roman strategy of managing conquered populations, which was to instill internal conflicts and wars among different groups, creating complexity in local politics, and ultimately achieving overall peace through local stability. This oversimplified explanation of the Roman strategy, called "divide et impera," has little to do with the current "divide and conquer" concept.

They aim at a comprehensive understanding of the system to build through luring and partially imprecise communication mechanisms, like prototypes or system metaphors, but, as observed elsewhere, the lack of precision of the detail contributes to a better perception of the system as a whole. They are not cowboy coding, as they are well-prescribed methodologies, still somehow perceived with skepticism and they are tried to be used more as an ad-hoc solution and as a set of practices instead of as a comprehensive way to tackle the essence of software production.

Systemic thinking is a particular model of work of the mind in which solutions to problems are sought to analyze the relationships among all the entities that "have some stake" in the problem and not by branching problems and solutions in the hope to reconcile everything in the end like in a big puzzle. Our key investigation is on the role of such systemic thinking in the development of software and on the connection that systemic thinking has with agile methods.

Our idea is that software development is an intrinsically wicked problem requiring systemic thinking, thus methodologies supporting systemic thinking are ontologically more suitable to it, and that systemic thinking should therefore take into consideration not only when crafting software development processes but also educating software engineers.

We have approached this issue with an on online survey. The limit of surveys for generalizing the results are well known [10]. For this reason we treat this as a first observational study, on the base of which further and deeper inferences can be drawn [1].

Specifically, we formulate our ideas into the following research questions for which we aim to draw for the first observations:

RQ_1: Do software engineers have a high level of systemic thinking in general?

RQ_2: Is there a difference between the level of system thinking of developers following traditional development approaches vs. the ones following agile approaches?

Our research approach is centered on a questionnaire crafted to determine the level of systemic thinking of software engineers on the "Systems Thinking Scale" [11], similar to what has been done in other knowledge-intensive disciplines, such as medicine. More than 100 developers and software engineers worldwide have participated in it.

We argue that in this study we provide a unique, original, and relevant contribution to the field as we assess the level of systemic thinking employed by software engineers in general and when specific approaches (like agile methods) are employed, paving the way to explain very important phenomena that have been observed so far and to craft effective software engineering experts and practices for the future.

Concretely, we claim that our results are extremely relevant, as they go to the root of the problems of software development and engineering and they provide a coherent and fundamental explanation of two phenomena that have emerged in the last half of a century:

1) the structural effectiveness of agile methods, but also the challenges of applying agile methods in larger organizations and in communicating to other, non-agile, parts of software companies,
2) the high number of musicians, artists in general, and even philosophers that have been observed among the best software developers.

This paper is organized as follows. Section 2 introduces the concepts of systemic thinking and the current approaches to measuring the level of system thinking of individuals and teams. Section 3 presents the structure of our experiment and describes the analysis procedures we performed, Section 4 details its results, and Section 5 discusses them. Section 6 discusses the validity of the results. Eventually, 7 draws some conclusions and outlines our future research.

2. Background on Systemic Thinking

2.1. Holism for Systemic Theory

The term "holism" was coined by Jan Smuts in his 1926 book *Holism and Evolution* [12]. Holism is a way of thinking that conceptualizes things and systems as aggregate and hence argues that we should see them as wholes, not merely as a collection of parts [13].

Aspects of holistic thinking have been present in ancient, medieval, Renaissance, and modern philosophies as well as in many religions (such as Buddhism) [14].

Parmenides, who can probably be considered the father of holism, argued that at some basic level, the world is a changeless unity. According to him, "All is One. Nor is it divisible, wherefore it is wholly continuous.... It is complete on every side like the mass of a rounded sphere." see [15]. In other words, he claimed that an understanding of the world requires thinking of it as an entangled unit, as a system of strongly interconnected entities.

In a similar vein, Pythagoras, who is generally credited as the first person to have used the word "cosmos" to refer to the universe as a whole [16], developed a full-blown holistic mathematical theory that envisaged numbers as a basic form of matter. According to him, all things have numbers in them, and hence all the objects of the universe — including human societies — are arranged in harmonious mathematical relationships with one another [17].

Hippocrates also created a holistic medicine that emphasized the importance of understanding the patient's health, and her mind, as well as the need for harmony between the individual, the social and the natural environment [18]

In a well-known passage from Plato's *Charmides* (155a8–58e5), Socrates discusses the holistic principles underlying medical practice to provide an understanding of the soul *anima mundi*, hence developing what is nowadays known as his epistemological holism [19]. Plotinus and subsequently other Neoplatonists (such as *Marsilio Ficino*) re-interpreted and re-elaborated the concept of *anima mundi* in important ways [20].

In the seventeenth century, another philosopher -Baruch Spinoza- defended a holistic (almost pantheistic) philosophy, according to which all the differences and divisions we experience in the world are just aspects of a mightier underlying single substance, which he identified with God or Nature [21].

Hegel, too, held almost mystical visions about the unity of knowledge and of all things related to it. Proof of that is that he based his holistic philosophy of the concepts of unity of nature and unity of state. Nature, on Hegel's view, consists of one timeless, unified, rational and spiritual reality [22]. Likewise, the state is an invisible higher reality to which single participating individuals defer and out of which they derive their strength and identity [23].

Building on these ideas, Karl Marx and other modern collectivist political thinkers emphasized the existence of a higher collective reality (the class, the unity, the group) at the expense of the individual [24].

In a similar vein, the rise of Gestalt psychology [25] can be interpreted as yet another attempt of humans to distance themselves from excessive individualism leading to alienation and fragmentation and instead exhibit a unitary appreciation of human beings' inter-dependency with one another and with the environment [26]

Quantum non-locality [9], resting on the non-local entanglement of particles, can be understood as a kind of relational holism in physics [27].

Generalizing this brief historical excursus, we can certainly assert that holism represents one of the key principles/ideas underlying systemic thinking.

2.2. Systemic Theory and Thinking

Systemic theory was first proposed, in the early '40s of XX century, by von Bertalanffy [28], an Austrian biologist known -among other things- for his important contributions to the laws of thermodynamics. However, in the early '50s, Systemic theory became very influential also in cybernetics, where it was applied and further developed by world-renowned cyberneticist Ross Ashby [29]. At the beginning of the '70s, the Systemic theory became pervasive among other disciplines, and psychologists (such as Bateson [30] as well as philosophers (such as Laszlo [31] started using its principles as a conceptual palette for novel insightful investigations. More recently, the Systemic theory has been successfully and profitably applied to organizational development as well as to management theory [32], [33].

In essence, the Systemic theory draws from a set of different disciplines (ranging from engineering, computer science, and cognitive science, to management, philosophy, psychology, and even biology) and attempts to provide a discipline-agnostic approach for dealing with complex and sophisticated problems [34]. Such an approach typically allows its users to understand the structure and properties of a given system in terms of the relationships of interdependence occurring among its different components [35].

The goal of the Systemic theory is therefore to comprehend systems "from a system of systems point of view" [36]p.2. This means that it typically attempts to formulate research frameworks and paradigms in which "physical, technological, biological, economic, environmental, social, cognitive, and metaphysical systems can be observed, studied, thought, modeled, simulated, and intervened" [36] p.2.

Systemic theory thus describes reality and the world as a network of complex structures, whether physical, virtual, or entirely numerical [37]. Yet, such descriptions require a fundamental (even if not detailed) understanding of the different areas of science that study them. The act of applying systemic theory to understand a given situation is called systemic thinking.

If indeed systemic thinking studies systems as aggregate then the reader may legitimately ask, how such an act of comprehension and understanding can be scientifically measured. Software engineers who practice systems thinking focus on understanding the behavior and interactions of the overall system, but their primary goal is to use technology to manage systems that have complex cause-and-effect relationships. They often use systems thinking before the system is even created, which limits their ability to learn from observing the system in action. In engineering, the concept of systems thinking emphasizes the importance of interactions and interfaces, as they contribute to the emergence of the system.

2.3. Measurements of Systemic Thinking

Metrics for systems thinking are used to measure and evaluate an individual's or group's ability to understand and analyze complex systems. These metrics typically involve a set of criteria or standards that are used to assess an individual's or group's performance in terms of their ability to think systematically.

The key characteristics of systemic thinking are typically measured by first identifying the cognitive styles, where the cognitive style describes how a person organizes and processes the information that s/he receives, for instance when making decisions some people concentrate on the individual components of the problem being scrutinized while others take the information as a whole and analyze the problem in a global perspective. Such various styles of thinking are believed to relate to problem-solving scenarios and to aid in determining how successful decision-makers will be [38].

Researchers have developed different metrics or scales to try to measure and objectively classify systemic thinking ([39] and [40]).

The Systems Thinking Scale (STS) [41] is probably one of the better-known examples. The Systems Thinking Scale is a 20-item instrument that uses a 5-point Likert-type scale (such as 0= Never; 1= Seldom; 2= Some of the time; 3= Often; 4= Most of the Time) in the attempt to compute and sum up responses to different items of a psychometric test devised for measuring systems thinking [41] [2].

In this study, we took this scale as a source of inspiration for our own work, in which we assess the systemic thinking skills of software engineers working in a software firm

2. More details can be found at https://case.edu/nursing/sites/case.edu.nursing/files/2018-04/STS_Manual.pdf, last accessed Jan 2023

to calculate the STS scores for four process frameworks, namely *well structured agile*, *mixed* (agile + traditional), *traditional* (waterfall), and *ad hoc* (eg. cowboy).

There is ample evidence suggesting that the 20-item STS test is valid [11]. It takes about 10 minutes to complete and it is simple to administer [42], [43], [41]. It has been consistently shown that such a scale possesses good reliability [42] while having the potential to:

- increase our understanding of the psychological mechanisms of systems thinking [43];
- and boost our understanding of human social capital [42].

3. Experiment and Analysis

3.1. Definition of the research questions

In the definition of our research problem, we have used the so-called Goal, Question, Metrics (GQM) paradigm [44]. The GQM is a systematic approach to fit and integrate objectives into particular models of cycles, products, and quality perceptions obtaining at the end the detail of a measurement system that provides some numeric insights into a problem.

Using GQM a particular target is characterized by goals, which are distilled into questions, and then metrics are introduced that can rate the information got as an answer to these questions. The calculated data may be analyzed by answering the questions to recognize whether the targets are achieved. Accordingly, metrics are characterized from a top-down viewpoint and examined and explained from the bottom-up perspective by using GQM. Despite the fact that the GQM approach is mostly used as a process measurement system, it is also convincing for resources, assets, and research questions [45], [46], [47].

In our case, first, a goal was identified, then it was distilled into a few questions that help to separate the problem into its multiple elements (in our case the questions were already validated as we were using standard systemic scale questions) [11]. The questions were then distilled into metrics which allow the questions to be answered. Metrics are empirical as the empirical metric relied on the measured variable alone which are the systemic scores that range from 0 to 80.

The GQM framework is not static but may require refinement. In this way, it is important that the metrics help to evaluate the elements as well as the reliability of the whole model, which is to make the measures developed. In addition, the growth of metrics gives a sign of how to measure them. It is possible to evaluate the calculation of solid and stable elements in an unbiased way, while casual and unstable elements should be abstractly evaluated.

Specifically, our GQM is:

1) **Goal:** To apply the systemic thinking scale and measure systemic thinking of software engineers in software development teams.
2) **Questions:** a) Do software engineers have a high level of systemic thinking in general? b) What is the level of systemic thinking in software development teams involved in different process frameworks (well-structured agile, traditional framework, mix of agile and traditional and ad hoc process)?
3) **Metrics:** The measurement system based on the STS [41], structured in 20 items with the corresponding scores organized in the Likert Scale.

3.2. Structure of the experimentation

In this study, we assess the level of systemic thinking of software engineers working in any software firm. We used an already validated 20-item Systemic Thinking Scale (STS) as a survey that we believe can measure the systemic thinking capacity of software engineers based on past studies [41] as well as face validity [11]. We designed the online form according the to the best standards in the discipline [48], [49], [50], [51] The questions consist of demographic questions and questions from the systemic thinking scale layed out according to the recommended standards [52], [53]. The survey was available for 1 month period. In this 1 month period, a survey was filled out by different individuals involved in the software engineering domain such as developers and analysts. We stopped receiving answers after 1 month, and by this time period, we have already received 100 responses.

The responses in the study were analyzed using basic statistical techniques. This study was primarily observational, which is common in software engineering research [54], [55]. However, it can be difficult to find participants for a study like this, especially when the focus is on a small group of software managers and engineers working in a software company. Therefore, we cannot make statistical inferences from our results. Despite this, we believe our results are significant because they show a trend of ideas that could be further explored and potentially lead to solid conclusions with additional analysis.

3.3. Subjects involved in the experiments and their profile

The survey based on the systemic thinking scale was forwarded to our professional networks and published in various verified LinkedIn and Facebook software engineering groups. Our aim was to get around 100 answers.

In terms of gender, there were 80 (about 75%) males, 18 (17%) females, and 9 (8%) preferred not saying anything about their gender as we can see in the figure 1 with an average age of 33.

Among the participant 45 (about 42%) people completed a bachelor's degree, 36 (34%) completed a master's degree, 18 (17%) a Ph.D. and 8 (7%) people were without a degree (Figure 2).

These individuals were from different countries such as Nepal, the USA, Brazil, India, and many more as shown in table 1.

The processes in place were various: 50 (about 47%) reported to use agile methods, 34 (32%) a mix of agile and

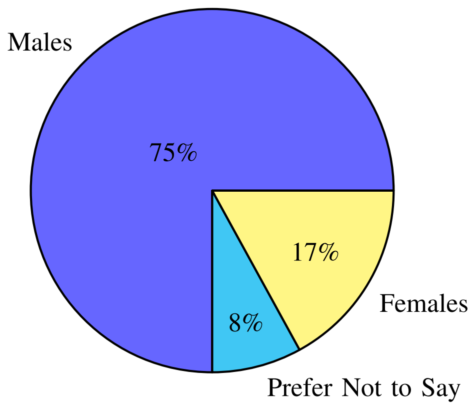

Figure 1: Gender distribution that filled out the questionnaire

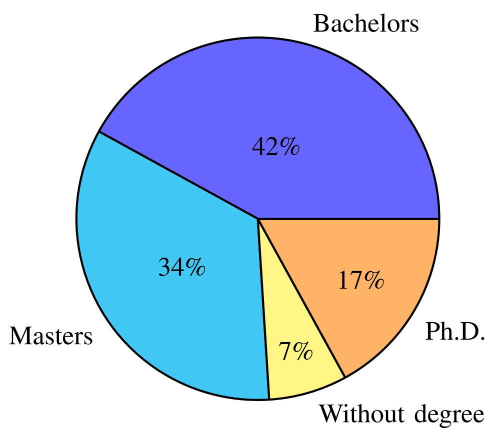

Figure 2: Degree distribution that filled out the questionnaire

traditional, 10 (9%) traditional, such as waterfall, and 13 (12%) an ad-hoc approach without a process (Figure 3).

In terms of country, there were 3 (about 3%) individuals from France, 6 (6%) from India, 25 (23%) from Italy, 4(4%) from Kazakhstan, 5(5%) from Nepal, 2(2%) from Nigeria, 29(27%) from Russia, 2(2%) from the USA, and 15(14%)

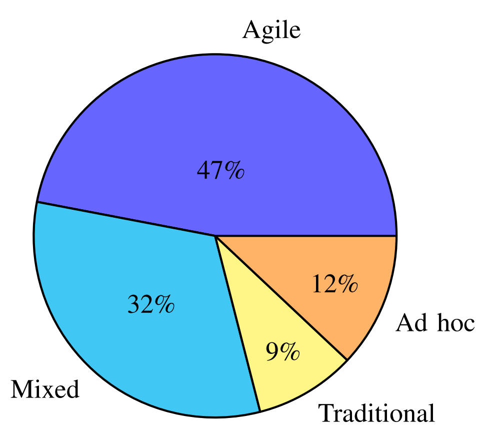

Figure 3: Process framework distribution from the survey

from other countries. In total, there were 16(15%) individuals who did not specify their country of origin (Figure 4).

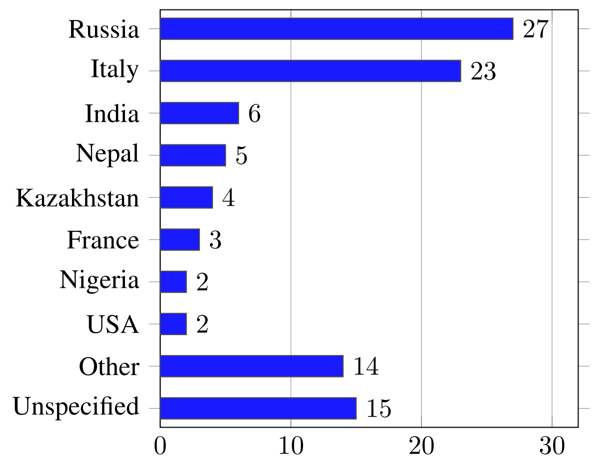

Figure 4: Distribution of Respondents across Countries

The summary of demographic information is in Table 1.

Demography	**Data**
Gender	80 Males, 18 Females, 9 Prefer Not to Say
Average Age	33
Average Working Experience	10
Education	45 BS, 36 MS, 18 PhD, 8 Others
Countries	Nepal, Russia, Nigeria, USA, Brazil, France, Bulgaria Syria, Italy, Albania, Kazakhstan, Philippines, Canada, India, Luxembourg, Turkey, and Greece.
Roles	Developer, Analyst, Scrum Master, Project Manager, Agile Coach, Professor, CTO, Designer, QA Specialist
Process	Agile (50), Mixed (34) Traditional (10), and Ad hoc (13)

TABLE 1: Summary of the Demography

4. Results

As mentioned, responses to the items on the Systems Thinking Scale are provided by users a 5-points Likert-type scale: 0=Never, 1=Seldom, 2=Sometime, 3=Often, and 4=Always.

Item scores are summed to provide an STS score. Scores can range from 0 – 80. There are no reverse-coded items.

In Table 2 are the mean scores and ranges obtained on the STS for software development teams involved in various process frameworks.

Process	**Average**	**Stdev**	**Score**
Agile	2.78	0.17	55.66
Mixed	2.75	0.29	54.95
Traditional	2.74	0.32	54.75
Ad hoc	2.49	0.29	49.89

TABLE 2: Systemic Thinking Scores

First of all, we notice that all scores are well above the middle level, 40, so we can observe that software engineers in general adopt a medium-to-high level of systemic thinking.

As regards to the second research question, we found that the team and individuals that are involved with a well-structured agile framework scored an average of 2.78 (a total of 55.66) evidencing a quite high level of systemic thinking, the highest in the group. The concept that agility promotes systemic thinking appears to be supported by the fact that the second highest group was of those applying a mixed agile and non-agile process (2.75 average for a total of 54.95), then followed by traditional processes (2.74 average for a total of 54.75).

Then, another idea that appears in the results is that still having a well-defined process promotes anyway systemic thinking over just working ad-hoc, which scores 2.49 for a total of 49.89. This could be because an organizational structure forces anyway people to think at the "big picture" over just working chaotically [56].

We have also performed the t-test to measure whether the differences in results could be regarded as significant. Indeed, as anticipated in the introduction and discussed later in Section 6, this is just an observational study [1]. Still, measuring the result of the t-test could give an idea of the strength of the effects that we have analyzed [3]. On such premises, we can look at the results (Table 3):

- agile and mixed show a probability of having their means of STS not different from the one of ad-hoc of 1.1%, well below the typical threshold of 5%;
- traditional show a probability of having its mean of STS not different to ad-hoc of 7.1%
- traditional and mixed are almost indistinguishable one another in terms of STS, as previously detected in another study [60].

	Mixed	**Traditional**	**Ad hoc**
Agile	0.619	0.529	0.011
Mixed	-	0.941	0.011
Traditional	-	-	0.071

TABLE 3: Probability of identical means of systemic thinking scores in different processes computed using the paired sample t-test

5. Discussion

The results of our observational empirical investigation lead us to the following conclusions about the two research questions.

Regarding $\mathbf{RQ_1}$, our findings indicate that software engineers, regardless of the process they follow, score significantly above the mean (40) in the Systems Thinking Scale (STS). This suggests that they tend to adopt a holistic, integrated, and dynamic approach to problem-solving and decision-making. They focus on understanding the interrelationships and feedback loops within a system and are proactive and collaborative in their approach [61], [62]. However, it is important to note that a higher score in the STS does not necessarily mean that the engineers are more effective or successful.

3. For completeness, we have also to consider that several studies of this kind published also in highly reputable venues have performed inferential tests and claimed generalizability of the results, still without fulfilling the required hypotheses [57], [58], [59].

Regarding $\mathbf{RQ_2}$, our results show that agile processes tend to be more effective in promoting systemic thinking compared to traditional and ad hoc processes. The results were ranked from highest to lowest in terms of STS scores: agile, mixed of agile and traditional, traditional, and finally ad hoc. This suggests that even traditional processes, which are more rigid and structured, can foster better systemic thinking than ad hoc processes, which are more haphazard [63]. This could be due to the fact that traditional processes provide clear guidelines and expectations for team members to follow, whereas ad hoc approaches may lack structure and direction.

Finally, it is important to note that the STS is just one tool for assessing systemic thinking and these results should be considered in conjunction with other factors that may influence a team's ability to think systemically. Also, the sample size and representativeness of the teams in this study is low, so further research is necessary to confirm these findings and to explore the factors that contribute to systemic thinking in software development teams.

6. Validity of the Findings

Our research, like all empirical investigations, is susceptible to the possibility of inaccurate conclusions. In this specific study, our approach is mainly observational, meaning that we aim to gather initial insights for future inferences rather than making inferences ourselves [54], [55]. Thus, the discussion of validity in our work focuses mainly on the reliability of our observations.

Regarding the external validity of our observations, it is worth noting that the number of participants in this study is limited and not randomly selected, as they primarily come from Russia and Italy. Gathering professionals who are willing to complete research questionnaires can be challenging, especially in this case due to the small and specialized community of managers and developers working in a software firm. Despite these limitations, we surveyed more than one hundred individuals from a large population, so we believe that our approximation of the general population is not biased to the extent that it negates our hypotheses.

In terms of internal validity, the questionnaire and online survey processes were designed to minimize the risk of researcher and respondent biases, using the best practices in the field, as previously discussed. While it is nearly impossible to eliminate biases entirely, we adopted a process called observed triangulation to reduce the occurrence of the experimenter-expectancy effect. The survey was based on literature with substantial scientific support and multiple team members analyzed the results to increase the credibility and validity of our findings. We also used a standardized data analysis process for qualitative data.

It is possible that other biases may impact our research findings, such as the confirmation bias. This type of bias involves only seeking information that supports the original hypothesis. In our case, we started the study with a focus on systemic thinking and sought instances of this concept in the practice of software engineers. However, to minimize this bias, we thoroughly and carefully evaluated the responses and impressions received from our survey participants. We made an effort to mitigate the occurrence of confirmation bias.

7. Conclusions

In this exploratory study the level of systems thinking of software engineers working using different development processes was assessed using the Systemic Thinking Scale.

The results showed that software engineers generally possess a medium to high level of systemic thinking, with teams and individuals using well-structured agile frameworks scoring the highest on average, followed by those who use a combination of agile and traditional processes and those who only use traditional processes. The study suggests that having a well-defined process can enhance systemic thinking compared to ad-hoc methods.

These findings underscore the potential benefits of well-structured agile processes and may account for their success. They also indicate that any reasonable process is likely to be better than no process at all.

However, it is important to note that this study has limitations, such as its small sample size and observational nature, which should be considered when interpreting the results. Further research with larger sample sizes and in a more controlled setting could validate these findings.

In addition, future studies could focus on strategies for enhancing systemic thinking skills and capabilities among development teams, as well as investigating the relationship between process frameworks and systemic thinking in greater detail. This may include considering factors such as team size, composition, and organizational culture that may affect a team's ability to systematically tackle complex problems.

References

[1] P. R. Rosenbaum, *Observational Studies*, ser. Springer Series in Statistics. Springer, 2002.

[2] E. A. Posner, K. E. Spier, and A. Vermeule, "Divide and conquer," *Journal of Legal Analysis*, vol. 2, no. 2, pp. 417–471, 2010.

[3] D. Maisto, F. Donnarumma, and G. Pezzulo, "Divide et impera: subgoaling reduces the complexity of probabilistic inference and problem solving," *Journal of the Royal Society Interface*, vol. 12, no. 104, p. 20141335, 2015.

[4] H. W. J. Rittel and M. M. Webber, "Dilemmas in a general theory of planning," *Policy Sciences*, vol. 4, no. 2, pp. 155–169, June 1973. [Online]. Available: http://dx.doi.org/doi:10.1007/BF01405730

[5] A. Janes and G. Succi, *Lean Software Development in Action*. Heidelberg, Germany: Springer, 2014.

[6] T. E. Bell and T. A. Thayer, "Software requirements: Are they really a problem?" in *Proceedings of the 2nd international conference on Software engineering*, 1976, pp. 61–68.

[7] I. H. Sarker, F. Faruque, U. Hossen, and A. Rahman, "A survey of software development process models in software engineering," *International Journal of Software Engineering and Its Applications*, vol. 9, no. 11, pp. 55–70, 2015.

[8] B. Acharya and P. K. Sahu, "Software development life cycle models: A review paper," *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. 11, pp. 169–176, 2020.

[9] J. S. Bell, "On the problem of hidden variables in quantum mechanics," *Reviews of Modern Physics*, vol. 38, no. 3, p. 447, 1966.

[10] A. N. Ghazi, K. Petersen, S. S. V. R. Reddy, and H. Nekkanti, "Survey research in software engineering: Problems and mitigation strategies," *IEEE Access*, vol. 7, pp. 24 703–24 718, 2018.

[11] M. A. Dolansky, S. M. Moore, P. A. Palmieri, and M. K. Singh, "Development and validation of the systems thinking scale," *Journal of General Internal Medicine*, vol. 35, no. 8, pp. 2314–2320, 2020.

[12] J. C. Smuts, *Holism and evolution*. Macmillan, 1926.

[13] S. Y. Auyang, *Foundations of complex-system theories: in economics, evolutionary biology, and statistical physics*. Cambridge University Press, 1998.

[14] J. Freeman, "Towards a definition of holism," *British Journal of General Practice*, vol. 55, no. 511, pp. 154–155, 2005.

[15] D. Gallop, "Parmenides of elea: fragments," *Philosophical Review*, vol. 96, no. 3, 1987.

[16] K. R. Popper, "Back to the pre-socratics: The presidential address," in *Proceedings of the Aristotelian Society*, vol. 59. JSTOR, 1958, pp. 1–24.

[17] J. Fauvel, R. Flood, and R. J. Wilson, *Music and mathematics: From Pythagoras to fractals*. Oxford University Press on Demand, 2006.

[18] G. Korobili and K. Stefou, "Plato's charmides on philosophy as holistic medical practice," in *Holism in Ancient Medicine and Its Reception*. Brill, 2020, pp. 201–219.

[19] C. C. Taylor, "Plato's epistemology," *The Oxford Handbook of Plato*, pp. 165–90, 2008.

[20] C. Thumiger, *Holism in Ancient Medicine and Its Reception*. Brill, 2020.

[21] G. Deleuze, *Spinoza: practical philosophy*. City Lights Books, 1988.

[22] G. W. F. Hegel, J. Burbidge, and L. Dickey, *The Cambridge Companion to Hegel*. Cambridge University Press, 1993.

[23] G. W. F. Hegel, *Hegel: Elements of the philosophy of right*. Cambridge University Press, 1991.

[24] K. Marx, W. Adams, T. Ball *et al.*, *The Cambridge Companion to Marx*. Cambridge University Press, 1991, vol. 1.

[25] W. Köhler, "Gestalt psychology today." *American psychologist*, vol. 14, no. 12, p. 727, 1959.

[26] I. Rock and S. Palmer, "The legacy of gestalt psychology," *Scientific American*, vol. 263, no. 6, pp. 84–91, 1990.

[27] J. S. Bell and J. S. Bell, *Speakable and unspeakable in quantum mechanics: Collected papers on quantum philosophy*. Cambridge university press, 2004.

[28] L. Von Bertalanffy, "General system theory: Foundations, development, applications," Georges Braziller, Inc., Tech. Rep., 1993.

[29] W. R. Ashby, *An introduction to cybernetics*. Chapman & Hall Ltd, 1961.

[30] G. Bateson, *Mind and nature: A necessary unity*. Bantam Books New York, 1979, vol. 255.

[31] E. Lasklo, *Introduction to systems phylosophy: Toward a new paradigm of contemporary thought*. Gordon & Breach, 1972.

[32] E. H. Schein, "Organizational psychology then and now: Some observations," *Annu. Rev. Organ. Psychol. Organ. Behav.*, vol. 2, no. 1, pp. 1–19, 2015.

[33] P. M. Senge, *The fifth discipline: The art and practice of the learning organization*. Currency, 2006.

[34] H. Lipson, "Principles of modularity, regularity, and hierarchy for scalable systems," *Journal of Biological Physics and Chemistry*, vol. 7, no. 4, p. 125, 2007.

[35] H. T. Odum, *Ecological and general systems: an introduction to systems ecology*. Univ. Press of Colorado, 1994.

[36] L. Gallón, "Systemic thinking," *Quality Education*, pp. 830–840, 2020.

[37] T. B. Sheridan, "The system perspective on human factors in aviation," in *Human factors in aviation*. Elsevier, 2010, pp. 23–63.

[38] E. Sadler-Smith and B. Badger, "Cognitive style, learning and innovation," *Technology analysis & strategic Management*, vol. 10, no. 2, pp. 247–266, 1998.

[39] S. Taylor, J. Calvo-Amodio, and J. Well, "A method for measuring systems thinking learning," *Systems*, vol. 8, no. 2, p. 11, 2020.

[40] L. Cabrera, J. Sokolow, and D. Cabrera, "Developing and validating a measurement of systems thinking: The systems thinking and metacognitive inventory (stmi)," *Routledge handbook of systems thinking*, pp. 1–42, 2021.

[41] A. Halpin and R. Kurthakoti, "Can systemic thinking be measured? introducing the systemic thinking scale (sts)," in *Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference*, vol. 42, 2015.

[42] M. L. Washington, R. Kurthakoti, A. L. Halpin, and S. Byrd, "Assessing systemic thinking in undergraduates: An exploratory study using a total enterprise business simulation," in *Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference*, vol. 41, 2014.

[43] F. Robinson-Bryant, "Developing a systems thinking integration approach for robust learning in undergraduate engineering courses," in *2018 ASEE Annual Conference & Exposition*, 2018.

[44] V. R. Basili, G. Caldiera, and H. D. Rombach, "Goal, question metric paradigm. encyclopedia of software engineering, vol. 1," 1994.

[45] L. C. Briand, C. M. Differding, and H. D. Rombach, "Practical guidelines for measurement-based process improvement," *Software Process: Improvement and Practice*, vol. 2, no. 4, pp. 253–280, 1996.

[46] D. R. van Solingen and E. W. Berghout, *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. McGraw-Hill, 1999.

[47] R. van Solingen, E. Berghout, and F. van Latum, "Interrupts: just a minute never is," *IEEE software*, vol. 15, no. 5, pp. 97–103, 1998.

[48] A. Furnham, "Response bias, social desirability and dissimulation," *Personality and individual differences*, vol. 7, no. 3, pp. 385–400, 1986.

[49] P. Podsakoff, S. MacKenzie, J. Lee, and N. Podsakoff, "Common method biases in behavioral research: A critical review of the literature and recommended remedies," *Journal of Applied Psychology*, vol. 88, no. 5, pp. 879–903, 10 2003.

[50] T. G. Bond and C. M. Fox, *Applying the Rasch model: Fundamental measurement in the human sciences*. Psychology Press, 2013.

[51] D. L. Vannette and J. A. Krosnick, *Answering Questions: A Comparison of Survey Satisficing and Mindlessness*. John Wiley & Sons, Ltd, 2014, pp. 312–327.

[52] P. M. Bentler, D. N. Jackson, and S. Messick, "Identification of content and style: A two-dimensional interpretation of acquiescence." *Psychological Bulletin*, vol. 76, no. 3, pp. 186–204, 1971. [Online]. Available: https://doi.org/10.1037/h0031474

[53] N. M. Bradburn and S. Sudman, *The Current Status of Questionnaire Research*. Wiley, 2004, ch. 2, pp. 27–40.

[54] N. Saarimäki, "Methodological issues in observational studies," *SIGSOFT Softw. Eng. Notes*, vol. 44, no. 3, p. 24, oct 2020.

[55] P. Ciancarini, A. V. Kruglov, W. Pedrycz, D. Salikhov, and G. Succi, "Issues in the adoption of the scaled agile framework," in *44th IEEE/ACM International Conference on Software Engineering: Software Engineering in Practice, ICSE (SEIP) 2022, Pittsburgh, PA, USA, May 22-24, 2022*. IEEE, 2022, pp. 175–184.

[56] F. Amagoh, *Systems and Complexity Theories of Organizations*. Cham: Springer International Publishing, 2016, pp. 1–7.

[57] P. Eada, "Experiment to evaluate an innovative test framework: Automation of non-functional testing," 2015.

[58] A. Waqqas, "Communication medium and communication effectiveness in software development virtual teams," Ph.D. dissertation, University of Phoenix, 2013.

[59] J. Pernstål, *Towards Managing the Interaction between Manufacturing and Development Organizations in Automotive Software Development*. Chalmers Tekniska Hogskola (Sweden), 2013.

[60] V. Ivanov, S. Masyagin, A. Rogers, G. Succi, A. Tormasov, J. Yi, and V. Zorin, "Comparison of agile, quasi-agile and traditional methodologies," in *Proceedings of 6th International Conference in Software Engineering for Defence Applications, SEDA 2018, Rome, Italy, June 7-8, 2018*, ser. Advances in Intelligent Systems and Computing, vol. 925. Springer, 2018, pp. 128–137.

[61] M. Nagahi, N. U. I. Hossain, R. Jaradat, V. Dayarathna, C. Keating, S. Goerger, and M. Hamilton, "Classification of individual managers' systems thinking skills based on different organizational ownership structures," *Systems Research and Behavioral Science*, vol. 39, no. 2, pp. 258–273, 2022.

[62] H. T. Bui and E. Galanou, "Translation of systems thinking to organizational goals: A systematic review," *Journal of General Management*, vol. 47, no. 4, pp. 233–245, 2022.

[63] M. C. Jackson, *Systems thinking: Creative holism for managers*. John Wiley & Sons, Inc., 2016.