IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Non Verbal Communication in Software Engineering – an Empirical Study

PAOLO CIANCARINI[1,2], MIRKO FARINA[1], SERGEY MASYAGIN[1], GIANCARLO SUCCI[1], SOFIIA YERMOLAIEVA[1] and NADEZHDA ZAGVOZKINA[1]

[1]Innopolis University, Russia
[2]University of Bologna, Italy

Corresponding author: Giancarlo Succi, g.succi@innopolis.ru

ABSTRACT Communication among humans consists of both verbal and non verbal components. The latter may sometimes express concepts or ideas not conveyable by the former. This is also true in Software Engineering. This paper first analyses theoretically the role of non verbal communication in software development teams, using the framework provided by distributed cognition as a conceptual palette and as a point of reference. Then, it presents an empirical investigation involving 38 IT professionals from Russia, sharing their experiences in communicating and interacting when developing software artifacts. The results of this empirical investigation are consistent with many of the ideas underlying a distributed approach to cognition. In addition, our findings provide valuable insights to make communication more effective in software development teams, while defining a new framework for follow-up studies.

INDEX TERMS Software engineering, software development, non verbal communication, empirical studies.

[1]

I. INTRODUCTION

Sharing concepts and ideas has always been perceived as a key issue in the process of collaborative software development. For example, when a team of software engineers creates a software product, how often have we read about the difficulties involved in agreeing on a common understanding of requirements among team members? The problem was already well-known in the '70s [1]–[3]. Yet, a decade later people were still struggling with the same issues, so much that in his groundbreaking paper Brooks claimed that such problems were the hardest to solve in the production of a software system [4].

Despite, over the years, much attention has been devoted to the problem of effective communication among software developers, a solution to this problem has not been readily forthcoming. As a consequence, researchers still face the same old issues, albeit in different forms. For instance, think about the obstacles observed in tracking team work [5], or about the struggle to crack the meaning of a programming code, which typically occurs during a code review (a widespread practice in software engineering, consisting in the personal assessment of a source code by human reviewers, interacting with the authors of the code [6]. Recently, a concept called *automatic intention mining* has been introduced, in order to classify sentences in developers' dialogues taking place online [7]. The idea is that when working online developers can record all important interactions and discussions, which they can subsequently analyze.

In this paper our working hypothesis is that part of the problems to which we referred above are related to a lack of understanding of the dynamics underlying interaction and communication in software teams.

In this context, it is worth mentioning that most researchers - although fully aware of such issues - attempted to defuse them; for instance, by either: (i) promoting the development of formal methods in requirements acquisition to deal with customers' lack of understanding [8], or (ii) formulating "clear" and "repeteable" definitions of process development models, such as Petri nets [9], [10], to overcome issues in collaboration and workflow management.

A very important step towards a better understanding of the crucial role of effective communication in software development was nevertheless made by proponents of the Agile Movement [11], a recent approach in software development that attempts to develop flexible and effective solutions to

[1]We thank Innopolis University for generously funding this research.

complicated software engineering problems via collaborative efforts between self-organizing, cross-functional teams and their customer(s).

At the core of the Agile Movement lies a set of important "social" values, which can be said to define it. One of the most important of these values is the primacy of '*individuals and interactions over processes and tools*' [12]. This value is associated to some agile principles, which strongly recommend communication. For instance, Principle 4 says that: *Business people and developers must work together daily throughout the project* and Principle 6 says that: *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation*. This idea of design-centered conversation - conceptualised as storytelling - is central in agile development: for example, in a recent paper - inspired by the work of Italian novelist Italo Calvino - we examined and specified this idea in relation to several dimensions (such as Lightness, Exactitude, Quickness, Visibility, Multiplicity, and Consistency) [13].

While researchers investigating Agile methodologies have emphasised the need and importance of individual face-to-face collaborative practices, such an understanding has not yet been fully understood within the boundary of a scientific framework. We only found an early study in which face-to-face collaboration was reported to enable better results than virtual collaboration via network communication [14].

In truth, before the emergence of the Agile Movement in the early 2000s, a few researchers independently pointed out the need for software engineers to learn so-called "soft skills"; that is, they highlighted the importance for developers to focus on how to deal with colleagues at work [15]–[17]. This represented a turning point in the history of the discipline, because it specified an essential step in making software development processes more successful, see for instance Ivanov et al. (2017) [18]. Concurrently, there has been a growing awareness of how distributed cognition is strongly interconnected with knowledge management [19], [20], especially in Information Technology (IT) [21].

For example, Boland et al. (1994) [22] and Richardson (2005) [23], successfully demonstrated how information technology, designed to support knowledge management within the context of distributed cognition, can be a very useful tool in the knowledge dependent environment, in which today's organizations and institutions operate.

And yet, to date, a very limited amount of studies have looked at the essence of the communication process taking place during software development, trying to specify its fundamental components and roadblocks, as well as to highlight how to make the whole process more effective and productive. We are aware that when it comes to communication, the cultural background of the people involved is a crucial factor to take into account. There are indeed some studies addressing this issue: for instance Tam et al. (2020) [24] showed that cultural and social background may have a significant impact on teamwork, while Gandomani et al. (2016) [25] demonstrated how culture and society may affect companies' agile transitions.

Given the preliminary nature of this empirical investigation, which focusses on non-verbal communication during the current Covid-19 pandemic, the analysis is carried out as an observational study. In other words, we did not perform any statistical inference [26].

The contribution of this paper is twofold:
First, it reviews the fundamental principles and mechanisms of verbal and, especially, of non verbal communication, explaining them within the framework offered by the theory of *distributed cognition*.
Then, on such basis, the paper goes on to review empirically how non-verbal communication mechanisms can justify the conditions for positive and negative communications in and by software engineering teams, with a specific focus on the relations that emerge in positive communications.

These findings seem quite relevant in the present circumstances, where a pandemic has forced the whole world to adjust working practices, favouring remote work, in order to comply with social distancing and other protocols required by the authorities. Such a readjustment has determined a redefinition of available communication channels, sometimes resulting in risky misunderstandings. Developing a framework for understanding communication among team members could therefore allow us to formulate appropriate mitigation strategies that could bring about more effective solutions for increasing the quality of software engineering practices.

This paper is thus organized as follows. Section II presents the fundamentals of verbal and non verbal communications and interactions within the framework of a distributed approach to cognition. Section III analyses how different communication channels play a role in software development. Section IV describes the structure of our experimental interviews. Section V presents the results of the experimentation; we use rankings to evaluate what interaction modes are perceived in communications and then we perform a comprehensive textual analyses of the conditions required for effective and easy communications. Section VI discusses our results in a broader context, reflecting on their relevance and applicability for the field, while Section VII focuses on shortcomings and potential issues threatening the validity of our investigation. Finally, Section VIII summarises what we achieved and outlines possible future research directions.

II. DISTRIBUTED COGNITION IN VERBAL AND NON VERBAL COMMUNICATION CHANNELS

According to distributed cognition, the social context and the artifacts present in the environment result in a cognitive system that is distributed among actors engaged in a collaborative activity [27].

The basic idea underlying distributed cognition is that people and artifacts can collaboratively engage in richly scaffolded, environmentally-involving partnerships to enhance their cognitive profiles and to master sophisticated cognitive problems that they would not be able to solve on their own, in the absence of such bi-directional partnerships [28]–[31].

Research on distributed cognition [32], [33] is therefore typically motivated by the observation that many important cognitive activities are often deeply entrenched in concrete community practices. Such cognitive activities are then intrinsically socio-cultural and -often- even materially distributed phenomena [?]. This means that to understand the reasons and the effectiveness of such activities we need to:

- contextualize them in the social and in the cultural environments where they occur, and
- consider the collateral elements of such environments, including their physical and virtual structure.

This claim should not be too surprising. Think, for instance, about the pivotal role that the so-called "setting" has in psychotherapy sessions [34]–[36]. Likewise, there has been a century long discussion on how to create an effective and cognitively enhancing childhood classroom layout [37], [38].

The crucial point underlying research on distributed cognition is that we should not take mental processes alone as the *explanans* for our cognitive activities [39]–[41]. This idea can be partly found in Gregory Bateson's seminal research [42], as well as in Ashby (1961) [43], which inspired Hutchins' seminal work in the field (1996) [30], and in Donald (2018) [44]. Because, on Hutchins' view, we cannot take the mental alone as the marker of the cognitive, it follows that we should push for the development of larger cognitive systems [45], encompassing the active participation of multiple actors interacting in a context where a range of artifacts are available to perform sophisticated cognitive activities.

To date, the distributed approach to cognition has been successfully used to study collaborative work practices in a number of different fields, including: airplane cockpits [46], air traffic control [47], ship navigation [48], software teams [27], call centers [49], and control centers [50].

Most of these studies highlighted the complex interdependency among people and technological artifacts in their collaborative activities. Specifically, they demonstrated that cognitive processes can take place in technical, materially and temporally distributed systems [51], [52]. As such, they showed that cognition is not necessarily brain-bound, and that cognitive abilities may well emerge in complex collaborative techno-social environments, where humans and technologies collude to solve complex tasks [39], [53].

It is possible to use the concepts and ideas underlying the theory of distributed cognition [33], [45], [54] to study hybrid couplings between humans and pieces of technological wetware. Distributed cognition, however, can also be observed and applied to a larger class of phenomena that typically involves humans interacting with each other. For example, the analysis of cognitive ecologies and material culture [52], [53], investigations over the nature of human agency [51], or the study of human memories [55], [56].

Given its reach, breath, and scope we believe that distributed cognition is a useful framework for explaining group dynamics and performances within agile teams. We are not first in using this approach in software engineering research; for instance, Mangalaraj et al. (2014) [57] used distributed cognition in order to examine the impact of design patterns and collaborating pairs on software design.

In this paper we take the theoretical framework afforded by distributed cognition to explore how verbal and non verbal communication channels and the related coordinating mechanisms used by social actors, may affect the development and quality of software engineering.

Before we get to look at how distributed cognition can help us achieving this goal, let us briefly summarise how such an approach has been applied to the study of language and gestures, which -on our view- represent two paradigmatic cases of verbal and non-verbal communication channels.

Distributed cognition typically describes language as a tool that enables cognitive enhancements. Clark famously argued that the role of language (and material symbols more generally) is: 'to provide a new kind of thought-enabling cognitive niche ... an animal-built physical structure that transforms one or more problem spaces in ways that (when successful) aid thinking and reasoning about some target domain or domains' [58]. Crucially, 'these physical structures combine with appropriate culturally transmitted practices to enhance problem-solving, and (in the most dramatic cases) to make possible whole new forms of thought and reason' (see also [59]).

The idea is therefore that language can be understood as a structuring activity unfolding in real-time across multiple time-scales. This activity involves individuals in social processes and arises when actors involved perform their actions in ensembles; that is, by coordinating their lives, both behaviorally and cognitively [60].

This understanding of language as a cognitive enhancing tool has been particularly influential in modern culture. Terrence Deacon (1998) [61], for instance, developed a theory of language that asserts that our capacity for abstract thought, logical reasoning, and compositionally structured, counterfactual thinking is ultimately dependent on public language.

However, language - when used in a verbal communication channel - is not the only means available to us for expressing our thoughts. Gestures (a non-verbal communication channel) sometimes represent our thoughts and emotions more effectively than language [62].

Think about when you are in a foreign country and you are unable to communicate in the country's language. Suppose that you ask a random person in the street some information (e.g. the location of a restaurant, which you know is nearby). Imagine that the person you stopped does not know any language apart from her own, the country's language, which you can neither speak nor understand. Now, suppose that the speaker understands your request (by comprehending the name of that particular restaurant you are looking for). This is not an unusual situation. However, she cannot translate her thoughts into comprehensible words for you. Yet, she points her finger towards the place you are looking for and by pointing her finger she tells you the direction you need to follow in order to reach that place. Following the direction of her finger you eventually reach the sought location (perhaps, if you are in Italy, she will also add some gesture to comment on the quality of the food of that restaurant...).

Gestures thus represent a magnificent tool in our cognitive arsenal that allows us to materialize and express our thoughts but also to augment comprehension and learning capabilities [62]–[64]. For this reason, it is not surprising that distributed cognition has also been applied to the study of gestures [65], [66].

Consider the following example as a good illustration of this point. The example is supposed to show how gestures can become vehicles of mental states in inter-subjective interactions [67].

The example involves an expert scientist who interacts with a novice. The aim of this interaction is to teach the novice how to correctly derive the meaning of complex Functional Magnetic Resonance Imaging (fMRI) scans. Alač and Hutchins [67] show how the gestures performed by the expert scientist become instrumental, so constitutive, in the process of teaching the novice scientist how to interpret the meaning of those scans. This is therefore a case where the distributed partnership between two users, which turns out to be gesture-involving, determines the acquisition of a new skill (hence of a cognitive ability) that one of the persons involved could have neither learned nor mastered on her own. In this sense then, gestures can become crucial in scaffolding new cognitive competences but can also facilitate learning in significant ways. It has been also demonstrated that students, mirroring teachers' gestures may learn concepts more effectively than those who do not [68].

Having introduced the theoretical framework underlying distributed cognition and showed how such a framework can inform research on verbal and non-verbal communication, we next turn to analyse a variety of communication channels in software development.

III. DISTRIBUTED COGNITION AND COMMUNICATION CHANNELS IN SOFTWARE DEVELOPMENT

Software engineers and computer programmers must interact, hence communicate, when developing sophisticated software programs. This is because developing a software is usually a mammoth task, often requiring several people with different expertise and a lot of time spent together and effort shared collaboratively.

When developing a software program, programmers and engineers must not only produce and comment the text of the program (the source code) together, but also test it and reach consensus on its outputs or effects, in order to analyze the risks of its usage. Moreover, before writing any code, the developers must consider and understand the requirements of the software and this requires a lot of communication and interactions not only among developers but also with stakeholders, who often do not have a technical background. In an agile context this problem of requirements consensual understanding has been studied using distributed cognition [69].

Modern software programs are produced following practices that are quite structured and typically include more than just (co)reading or (co)writing some code. For instance, drawing or even sketching diagrams is a widely used tool to visualise, share, and improve -through discussion- the architecture of a given code [70].

As an illustration of this point, consider a team of programmers enacting the collaboration rules specified by Scrum, which is an agile, iterative and incremental process model, for developing, managing, and delivering software products [71]. As a software process model, Scrum defines minimal sets of roles (Product Owner, Scrum Master, Developer) and of collaboration ceremonies (sprint, sprint planning, daily meeting, sprint review, sprint retrospective), which require a lot of interaction among the people included in the development team.

Before any code is written, the team must receive from the Product Owner (the individual or the sponsor that pays or is responsible for the development of the product) a set of instructions or requirements that the end product must fulfil.

Such requirements (or instructions) are usually put in a special narrative form, which is called the *user story*. The user story is therefore a boundary object, basically a short sentence expressed in natural language proposed by the owner and written from the perspective of an end user, that has the function of influencing the functionality of the system being developed.

User stories are often collected in repositories, in the form of index cards or post-it notes; a repository of user stories is similar to a whiteboard, which facilitates autonomous retrieval. User stories, once received and accepted by the team, are broken down into tasks, which are then estimated by the developers. One way of estimating a user story is performed through the so-called *planning poker game*; that is, a combination of verbal and non-verbal (card-based) communication acts [72].

User stories, however, are not necessarily finished products when they are collected in repositories - quite the opposite. They can be subsequently modified; that is, rewritten, expanded or reduced, based on enhancing interactive communication taking place between team members and customers and stakeholders. This typically takes the forms of notes, attachments, and acceptance criteria, which may be written in typical structured formats (such as Given-When-Then or in bullet-points). A user story is completed when all acceptance criteria are met.

The brief analysis conducted above has an important goal in the economy of this paper. It is supposed to highlight the fact that we need a powerful theoretical model to make sense and properly account for the many cognitively enhancing actions performed by members of a team of software developers adopting, for instance, the Scrum collaboration model.

An ideal candidate model to analyze the sort of interactions in communication going on in such cases is - we believe - the framework offered by distributed cognition, which we introduced above.

Distributed cognition explains collaborative work practices as distributed and cognitively enhancing activities taking place among team members [73], [74]. As such it seems like a very powerful tool in our arsenal to make sense of the many potential cognitively-enhancing interactions that happens at several stages (both in presence and online) in the development of software programs.

TABLE 1. Examples of communication and interaction artifacts in software development, classified as verbal/non-verbal and in-presence/on-line.

	in presence	on line
verbal	conversation user story shared board shared video	shared document (eg., googledoc) text chat audio/video-chat
non verbal	body language clothes tattoos gestures facial expressions gazes	emoticons avatars fonts and shortcuts sound and action words

With respect to this latter point, it is worth mentioning that Agile methods recommend that cooperation should take place in the same work-space, possibly in the same room, i.e. in presence. Face-to-face communication and related interactions are seen as instrumental in bringing about results (end products) that are typically qualitatively better, richer, and more meaningful (hence, less likely to be disrupted by potential impediments to interactions, see section III) than those arising from online communication.

It is for this reason that we list in Table 1 a number of communication and interaction artifacts (verbal and non-verbal) developed or delivered online.

We notice that, starting March 2020, the COVID-19 pandemic has forced nearly all developing teams worldwide to find new ways of cooperating on line. The interviews that we refer to in the next section were conducted at the beginning of the pandemic, so the distinction between in presence/online was not emphasized.

Instead, we put emphasis on the distinct modalities of non verbal interaction, which involve the use of voice, body language, time awareness, control of space, face movements and gestures. In fact, most non-verbal artifacts can be classified in the six categories shown in Table 2, and we use this classification to analyze the answers given by our interviewers.

IV. INTERVIEW STUDY

In order to analyse the role that non-verbal communication may have in software development, we conducted a set of semi-structured interviews, which were based on a pre-compiled questionnaire that all the interviewees had to answer. The questionnaire containing all the questions we asked during the interviews is appended to this manuscript see Tab.7.

TABLE 2. Modalities of non-verbal interaction

Mode	**Definition**
Vocalics	non-verbal uses of the voice showing emotion and providing cues as to how the message should be interpreted
Kinesics	body motion communication such as facial expressions and gestures
Chronemics	time orientation, understanding and organization; use of and reaction to time pressures; awareness of time; wearing or not wearing a watch; arriving, starting, and ending late or on time
Proxemics	the human use of space; interpersonal distance; haptics
Oculesics	eye movements, eye behavior, gaze, and eye-related nonverbal communication
(non verbal) Synchrony	face-to-face interaction and mimicry [75]

Designing an effective questionnaire for an interview about interaction and communication is no easy task, since the way questions are defined, organized, and laid down may significantly influence the answers. Therefore, we followed the rules defined in [76]–[79] to minimize biases (such as redundancy and replication) in responses.

All interviews were conducted online and recorded on audio and then processed manually in a spreadsheet form. All recordings were performed only after we obtained written consensus by the participants involved.

The interviews were conducted between March and April 2020. The sample group considered in our study consisted of 38 people working in the IT industry of Russian and Tatar ethnicity, both males and females, aged between 22 and 35, with considerable working experience. More precisely, the participants in our study worked as managers, software developers, or data scientists in various tech companies based in Innopolis, a city in the Russian Federation. Figure 1 shows the distribution of our experimental sample: 63% of interviewees worked as software developers, 24% worked as managers, and 13% worked as Data Science engineers.

We also collected information about the particular development methodology used in the participants' teams (see Table 3), where applicable. The vast majority of participants worked in accordance with the principles of Scrum (45%) or other Agile (21%) methodologies.

TABLE 3. Statistics on methodologies applied in participants' teams.

Methodology	**% of participants**
Scrum	**45**
Generic agile	**21**
No methodology	**16**
Combinations	**8**
Kanban	**5**
Iterative	**2.5**
No-name ad-hoc	**2.5**

Following the best practices in the discipline [80], we

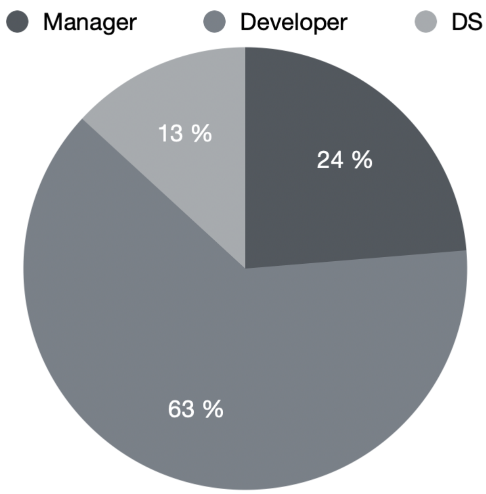

FIGURE 1. Statistics on participants' job position.

then performed a content analysis of the outcomes; first, using ranks to determine the perceived modes of non verbal communications and subsequently performing a thorough textual analysis to determine modes and factors that made communications to be perceived as easy and effective (Section V) [81].

V. OVERALL ANALYSIS OF THE RESULTS

As a starting point for our analysis, we would like to emphasize two things:

- we analysed the perceived modes of non verbal communications;
- we also focused on two different types of perception:
  - self-perception; that is, how every person perceives her/his own communication,
  - the perception of the communication arising from the interaction with others.

Consequently, we split our analysis into three major categories:

- those modes perceived by the person performing the communication;
- those modes perceived by the person towards which the communication is directed;
- those modes perceived by either the person performing the communication or by the person towards which the communication is directed.

There are indeed means of non-verbal communications that remain unperceived rationally and occur completely unconsciously and/or subliminally [82]. We are not referring to them in this study. This does **not** mean that such channels of communication are not important; rather that they will be the focus of further studies. It is also worth noting that in our analyses some non verbal communication channels showed very low replies; this could be explained by the fact that some people are not aware of their presence and such channels are experienced at the unconscious or subliminal level.

Moreover, communications can carry feelings and emotions that indeed play a central role on how such communication is interpreted [83], [84]. Therefore, we consider three cases:

- first, any kind of communication no matter whether with neutral, positive, or negative feelings or emotions;
- then, communications carrying positive feelings and emotions only; and
- finally, communication carrying negative feelings and emotions only.

We would like to notice that the term *positive* and *negative* refer to the perception of the person producing or receiving non verbal messages. It must therefore be noted that perception in the sense used in this study is -to some extent, at least-subjective in character. Nevertheless, it is normal practise in our field to study it as such [85].

Finally, we turned our attention to two different classes or attributes of communications; namely communications that are perceived as effective and communications that are perceived as easy [86]. In these cases we performed a textual analysis of the answers to determine the explicit modes and the factors that play a roles in such situations [81].

A. OVERALL MODES OF NON VERBAL COMMUNICATION

Our first analysis concerned what modes of non-verbal communication are perceived overall (Table 4).

TABLE 4. Rank of communication modes being perceived

Mode used	by me	by others	by any
Vocalics	74%	58%	79%
Kinesics	50%	34%	55%
Chronemics	13%	16%	18%
Proxemics	0%	3%	3%
Oculesics	0%	3%	3%
Synchrony	5%	0%	5%

It is evident that vocalics (79%) and kinesics (55%) are very well perceived, as they are intentionally performed: the percentage of self-perception (74% for vocalics and 50% for kinesics) is even higher than the level of perception by others (58% and 34% respectively). The presence of Chronemics was non trivial as well (18%); however, with a slightly higher presence of perception by others, meaning that other people are more receptive to messages about, for instance, timing of events than the people sending such messages. Proxemics, oculesics, and synchrony were almost not perceived.

B. WHAT HAPPENS WHEN COMMUNICATION IS PERCEIVED AS POSITIVE

Our analysis refers to what is going on when communication is perceived as positive (Table 5). There are two potentially important provisos in analysing our data:

- the concept of positive communication is loosely defined, as previously mentioned. However, we think that it adequately characterizes the situation in which a person feels comfortable in sharing her/his ideas and, again, as mentioned earlier on, it is a common practice in the field [85];

- here we are analysing the simultaneous presence of two elements, positive communication and a specific non verbal communication mechanism; this does not imply *per se* any form of causation; however, this provides a strong indication of some sort of role played by the specific non verbal communication mechanism on making communication positive, a point which -as we shall see- will become central in the economy of our paper.

TABLE 5. When the communication is considered positive

What is being used	by me	by others	by any
Vocalics	16%	16%	26%
Kinesics	21%	8%	24%
Chronemics	3%	5%	5%
Proxemics	0%	3%	3%
Oculesics	0%	0%	0%
Synchrony	3%	0%	3%

What one can observe is that now non verbal communication patterns are less perceived overall – with respect to the top categories we have a reduction of about two thirds for vocalics and chronemics and one half for kinesics. Our hypothesis is that people involved in *happy* interactions get so emotionally involved that they cannot recall exactly what is happening.

Moreover, we still have the predominance of vocalics and kinesics, but their relationship changes. Before, vocalics was dominating. Now, kinesics is getting very close to vocalics, and, actually, in self-perceived communication scores higher than vocalics. A possible explanation for this result is that in such easy interactions people feel free to move around, to interact, to express themselves with explicit body signals. In other words, they do not need to use a very static mechanism, such as their voices, to make their point or express their ideas.

C. WHAT HAPPENS WHEN COMMUNICATION IS PERCEIVED AS NEGATIVE

The last piece of analysis relates to what happens when communication is perceived as *negative* (Table 6). Once again, we would like to notice that the same consideration we made for the term "positive" earlier on, ought to be made here for the term "negative."

TABLE 6. When the communication is considered negative

What is being used	by me	by others	by any
Vocalics	5%	8%	11%
Kinesics	11%	5%	11%
Chronemics	8%	8%	16%
Proxemics	0%	3%	3%
Oculesics	0%	3%	3%
Synchrony	5%	0%	5%

In this case, we note a degradation in the perception of non verbal communication mechanisms. However, we still have the same three most used modes of perception, albeit in a different ranking – chronemics becomes the most relevant overall (16%), and it is followed by vocalics and kinesics.

The reduction of non verbal communication channels could be attributed to the fact that when people become tense, they enter a defensive mode, in light on which they are less likely to express themselves freely. The perception of time, though, becomes more important, perhaps because people would like to stop such communication or they feel as if they are wasting their time by participating in it. Moreover, people tend to use kinesics more than any other mode of perception, perhaps for the need to communicate their uneasiness somehow, especially in situations where a frank verbal expression of their feelings would not be permissible or would determine undesired consequences.

D. COMPONENTS UNDERLYING EFFECTIVE COMMUNICATION

To find common factors that could highlight effective communication patterns, we gathered answers to the following question: "Do you think communication is effective during these meetings?" (Question 14 in Table 7)

Six responses highlighted that the effectiveness of the meetings depended on chronemics:

- It annoys me when people do not spend time on workflows: for instance, when they make jokes or waste their time producing memes.
- If 15 minutes for the meeting is enough, then the meeting is effective.
- The meeting was effective because I could not find a way to spend less time for it.
- A meeting is effective when it is over quickly.
- It is not very convenient to schedule a meeting for just 5 minutes.
- If I have other important deadlines, then I am not fully focused on the discussion.

Then, **two** responses displayed a reference to the vocalics and **one** response showed a reference to the kinesics.

Overall, it appears that effectiveness is linked strongly to chronemics. Taking into account what we noted above, we could argue that effective interactions are centered onto the ability to exhibit time management and that when communication becomes negative the time factor become more peculiar. We are not arguing that if communication is positive a meeting ought lasting longer than it should; however, if the situation is difficult, it appears essential in software teams to show a very effective ability to manage time. Therefore, vocalics and kinesics appears relevant more as a collateral factor to keep the environment positive.

E. COMPONENTS UNDERLYING EASY COMMUNICATION

Factors leading to easy communication were retrieved from answers to the following question “During meetings is it easy for you to communicate and express your ideas?”. (Question 13 in Table 7)

Twelve responses contained a reference to vocalics:

- Easy, because communication is frequent and is led by one person.
- balance in listening and speaking is indeed important.
- I was asked questions and I knew how to answer them. There is no pressure on me.
- When one speaks, the others are silent and listen.
- I notice changes in the participants’ facial expressions
- It is hard to talk about salary delays. On the contrary, it is easier to write an email about it.
- It is easy to communicate when everyone is laughing. If there is a note of fun in the meeting, then everyone is relaxed.
- People are calm. They communicate from a position of equality and they are all friendly.
- No one raises their voice.
- I feel relaxed and smile a lot.
- Usually everything is easy, everyone said something, someone made a joke, and we continued working.

Then, **seven** responses referred to chronemics all in negative terms (e.g., “Long meetings are tiring”), **three** to kinesics all in positive terms (e.g., “I’m relaxed”), **one** each to synchrony, appearance, and environmental.

Overall, it appears that an easy (perceived) communication is very similar to a positive communication with a clear reference to vocalics and kinesics in positive terms and with references to chronemics in negative terms.

F. SUMMARY

In analysing these results we found that the predominant non verbal communication modes perceived in a communication refer to vocalics, kinesics, and chronemics. Vocalics and kinesics are typically perceived in positive communications. On the contrary, chronemics emerges in negative communications, which is also when the perception of non verbal communications strongly diminishes. Moreover, meetings perceived as effective appear to be strongly connected to the ability to show mastery of chronemics.

Most of the reflections that we can make on such analysis appear aligned with our original intuition, apart from one. During negative communications it appears that people tend to loose the perception of non verbal messages. In such situations, probably the presence of tools or mechanisms to mediate could be advantageous. After all, it seems that there is not any additional information rationally perceived via non verbal communications, and so that all the remaining non verbal modes could just convey negative feelings and emotions. This indeed would raise a point of singularity on several claims always found in software engineering papers and books, asserting that in person communication is necessarily better, as argued in the Agile Manifesto “individuals and interactions over processes and tools” [12]. This issue has been partly discussed, in the early ’90s, by proponents of computer mediated communication [87]–[89] and, more recently by [90], [91]. However, it has remained largely unexplored in Software Engineering.

VI. DISCUSSION

We established that effective or successful meetings are those in which there is a dominance of Vocalics and Kinesics. So, positive (hence enhancing) interactions seem to take place when there is fruitful integration of verbal and non verbal communication channels. We also found that when the perception of non-verbal communication degrades due to Chronemics, negative communication may arise.

This is an important result in the economy of our paper, as it suggests that non-verbal communication channels (such as a gesture) play a crucially important role not only in the establishment of successful meetings but also in the formulation of positive (possibly mind-enhancing, productivity-boosting) interactions among individuals. In other words, our findings seem to suggest that the non-verbal factor is instrumental, therefore constitutive, of successful interactions in software development. This is consistent with the principles of distributed cognition we introduced earlier on.

In addition, these results may be used to better comprehend how people behave during a meeting, as well as to understand how they may perceive the meeting’s dynamics and whether certain actions ought be taken in order to improve a meeting’s effectiveness.

For example, the presence of the Kinesics factor may indicate that everything is fine with the meeting and that everyone is comfortable; hence, that the meeting is likely to be a productive one. The presence of the Chronemics factor (e.g., people looking at their watches), on the contrary, may indicate that the meeting is problematic, that there could be issues with communication or perhaps a degree of uneasiness, due to various possible factors (e.g., lack of mutual understanding, lack of leadership etc).

We believe that our findings are significant, hence worthy of publication, because they can be used to inform or -at least- partly predict the easiness and productivity of a meeting, as well as to describe problems arising in collaborative interactions. In other words, these results are important because they can allow us to formulate mitigation strategies that could maximise workflow management. In addition, these findings seem to be also crucial for Agile Teams, inasmuch as they demonstrate that curbing or limiting non-verbal communication may cause work issues and reduce work effectiveness alongside performances.

The discussion above provides further evidence for the merit of distributed cognition as a tool and as a framework to make sense of collaborative interactions. In cases in which communication among team members is difficult or negative, distributed cognition creates the conditions for the construction of richly collaborative partnerships, which may positively contribute to overcoming such issues. In cases in which communication among team members is already positive and effective, distributed cognition triggers and enables a modality of comprehension of problems, goals, tasks, situations, and issues that is ontologically and epistemically richer than the one experienced by the single team member on its own.

VII. LIMITATIONS AND THREATS TO VALIDITY

As usual, some shortcomings may affect the overall validity of our study. We discuss some of such limitations briefly below.

Firstly, as mentioned in the introduction, this work is an observational study carried out without performing inferential analysis. Our work also awaits replication. To corroborate and extend our findings and preliminary conclusions, further analysis could employ clustered observational studies and inferential statistics and also consider the specific development methods, application domains, team structures, and underlying organizational and individual cultural factors potentially affecting communication in software engineering teams [26], [92], [93]. This is a major issue in all scientific research and this is why we make our data openly available [94], [95].

Secondly, our study is based on semi-structured interviews, which is the only way to conduct an analysis at this stage. Indeed, semi-structured interviews provide subjective viewpoints, and this makes their interpretation partly problematic. Thirdly, our sample (38 people of Russian cultural background) albeit consistent, needs to be expanded to achieve a higher level of representativeness. Our participants mostly come from the city of Innopolis (a city in the Russian Federation), where there are a few large companies ($\geq$ 250 employees) and several small or medium enterprises. About half of the employees work in large companies, a quarter in small (10 – 49), and the rest are split between medium size (50 – 249) and micro-sized ($<$ 10) ones. This structure and population are similar to several other parts of the world including the Silicon Valley, the areas around Sophia Antipolis (France), London, Shenzhen, etc. The companies in Innopolis typically employ male software engineers, who hold a graduate degree in a STEM field and have more than 5 years of professional experience. The participating companies are a reasonable representation of the overall population of the developers and software engineers present in the Innopolis Special Economic Zone; we assume that the industrial community in this area is similar to communities in other areas around the world. Still, as mentioned on the first point above, future work with professional communities distributed on a wider area could provide more reliable confirmations for our findings and contribute to increase the degree of universality of our results [96]. Fourthly, a control group could be introduced to ensure that the design of our experiment is sound. Nonetheless, we must notice and acknowledge that empirical works in software engineering are quite challenging, and that we still managed to collect a relatively large sample of responses, offering quite uniform and consistent answers, hence good for statistical purposes.

Having briefly described a number of potential limitations or shortcomings affecting our study, we next focus on a series of potential threats that may call into question the validity of our results.

According to [97], at least three types of biases may occur at any time in case study research, thereby affecting its reliability. These are:

- Respondent bias - when the participants respond inaccurately or falsely to the questions asked. This type of bias typically occurs in research involving participant self-report, such as structured interviews or surveys.
- Researcher bias - when researchers, under the influence of previous knowledge and/or assumptions intentionally or unintentionally, mislead the research they carry out.
- Reactivity - when the researcher physical presence influences the respondents' answers.

[98] explicitly suggests adopting a number of techniques to avoid the occurrence of such biases in case study research. The techniques that [98] encourage to deploy include:

1) Prolonged Involvement
2) Member checking
3) Peer debriefing
4) Audit trail
5) Observer Triangulation

We achieved *Prolonged Involvement* by cultivating a friendly and relaxed atmosphere, where participants had no time pressure and were free to express themselves as it pleased them. Data and results were returned to participants to check for accuracy and resonance with their own experiences (this contributed to achieve *Member Checking*). To comply with *Peer Debriefing* we sent a cover letter describing the research to the participants involved in our study. This letter was sent before the interview took place, alongside other important information (such as the purpose of the study, the reason for the interview, the interviewer's role in the study, the subject of the questions, and the approximate duration of the interview). To accomplish *Audit Trial*, hence to specify and describe how we collected and analyzed the data; the research topic, the rationale for the interview as well as its agenda were clearly and transparently mentioned at the beginning of each interview. Anonymity was also ensured for all the participants involved in our study. We achieved *Observer Triangulation*, as suggested by [99], by having two researchers supervising each interview [100].

Finally, given the nature of the data and the structure of our investigation, we did not perform inferential statistics. However, we did perform an observational analysis, following the best recommendations and disciplinary norms adopted in Software Engineering [101]–[103].

VIII. CONCLUSIONS

In this paper we analyzed both verbal and non-verbal interactions in software development, exploiting an approach inspired by the principles of distributed cognition. As we have seen above, distributed cognition is an approach used in the cognitive sciences to model and describe collaborative partnerships and working interactions.

In this article, we focused especially on non-verbal interactions, which have not been studied in depth. In fact, we found very little research work published on non verbal interactions of software developers [104].

The role of distributed cognition in our research is absolutely crucial. This theory is instrumental to generate a uniform and consistent analysis of people's behaviors and activities as well as of their mutual and reciprocal interactions when engaged in complex tasks (such as software development), which require continuous interactions and/or persistent communications.

We made explicit and studied a number of modalities of non-verbal interaction. Given the nature of our research, based on semi-structured interviews, we collected developers' responses to our research questions. Crucially, such responses were obtained from people who met, mostly physically, with other people at conferences and seminars over many years. Admittedly, in today's workplace, as reshaped by the Covid-19 pandemic, non-verbal communication (e.g.,sending a Telegram message full of emoticons or an immediate response via Slack) is replacing body language or glances launched through arch eyebrows. This suggests the need for future work to focus, more and more, on non-verbal communication among software developers cooperating via Internet and applications like Zoom or MS Teams.

We hope that our work will contribute to improve the channels and tools used for communication and collaboration in software development, and more importantly, make aware ourselves, our co-workers, and our students who build software in teams that non-verbal interactions are crucially important for their work, just like verbal communication. In addition, we also wish that this study will raise greater awareness of how different communications channels may improve knowledge management within organizations taking advantage of the insights offered by the paradigm of distributed cognition.

In sum, we believe that software developers must improve their communication skills because their world is going to be less and less dependent on interactions with computers alone, and increasingly more focused on online communication with other humans *via* computers.

• • •

REFERENCES

[1] D. T. Ross and K. E. Schoman, "Structured analysis for requirements definition," IEEE Transactions on Software Engineering, vol. SE-3, no. 1, pp. 6–15, 1977.

[2] P. Zave, "A comprehensive approach to requirements problems," in COMPSAC 79. Proceedings. Computer Software and The IEEE Computer Society's Third International Applications Conference, 1979., 1979, pp. 117–122.

[3] K. T. Orr, "Structured Requirements Definition in the 80s," in Proceedings of the ACM 1980 Annual Conference, ser. ACM '80. New York, NY, USA: Association for Computing Machinery, 1980, p. 350–354.

[4] F. P. Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering," IEEE Computer, vol. 20, no. 4, p. 10–19, 1987.

[5] E. Salas, N. J. Cooke, and M. A. Rosen, "On teams, teamwork, and team performance: Discoveries and developments," Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 50, no. 3, pp. 540–547, 2008.

[6] M. Beller, A. Bacchelli, A. Zaidman, and E. Juergens, "Modern code reviews in open-source projects: Which problems do they fix?" in Proceedings of the 11th Working Conference on Mining Software Repositories, ser. MSR 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 202–211.

[7] Q. Huang, X. Xia, D. Lo, and G. Murphy, "Automating intention mining," IEEE Transactions on Software Engineering, vol. 46, no. 10, pp. 1098–1119, 2020.

[8] T.-L. Tran and J. Sherif, "Quality function deployment (qfd): an effective technique for requirements acquisition and reuse," in Proc. of Software Engineering Standards Symposium, 1995, pp. 191–200.

[9] B. Krä and Luqi, "Petri net-based models of software engineering processes," in Twenty-Third Annual Hawaii International Conference on System Sciences, vol. 2, 1990, pp. 104–112 vol.2.

[10] G. Denaro and M. Pezzè, Petri Nets and Software Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 439–466.

[11] M. Fowler and J. Highsmith, "The agile manifesto. software development, 9 (8), 28-35," 2001.

[12] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for agile software development," 2001. [Online]. Available: http://www.agilemanifesto.org/

[13] P. Ciancarini, S. Masyagin, and G. Succi, "Software design as story telling: reflecting on the work of Italo Calvino," in Onward! 2020: Proc. of the 2020 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, 2020, pp. 195–208.

[14] H. P. Andres, "A comparison of face-to-face and virtual software development teams," Team Performance Management: An International Journal, vol. 8, no. 1/2, pp. 39–48, 2002.

[15] V. Bellotti and S. Bly, "Walking away from the desktop computer: Distributed collaboration and mobility in a product design team," in Proc. ACM Conference on Computer Supported Cooperative Work, ser. CSCW '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 209–218.

[16] S. Jarzabek and R. Huang, "The case for user-centered case tools," Commun. ACM, vol. 41, no. 8, p. 93–99, Aug. 1998.

[17] R. E. Kraut and L. A. Streeter, "Coordination in software development," Commun. ACM, vol. 38, no. 3, p. 69–81, Mar. 1995.

[18] V. Ivanov, A. Rogers, G. Succi, J. Yi, and V. Zorin, "What Do Software Engineers Care about? Gaps between Research and Practice," in Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ser. ESEC/FSE 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 890–895.

[19] S. Rinkus, K. A. Johnson-Throop, and J. Zhang, "Designing a knowledge management system for distributed activities: a human centered approach," AMIA Annual Symposium proceedings, pp. 559–563, 2003.

[20] S. Gasson and J. Waters, "Social and distributed cognition in knowledge management systems," in 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07), 2007, pp. 199–199.

[21] C. Ferran and R. S. Koussa, "Distributed cognition supported by information technology can help solve the knowledge management bottleneck," Academy of Business Research Journal, pp. 32–54, 2012.

[22] R. J. Boland Jr, R. V. Tenkasi, and D. Te'Eni, "Designing information technology to support distributed cognition," Organization science, vol. 5, no. 3, pp. 456–475, 1994.

[23] S. M. Richardson, "Knowledge management and the design of distributed cognition systems," in Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005, pp. 242a–242a.

[24] C. Tam, E. J. da Costa Moura, T. Oliveira, and J. Varajão, "The factors influencing the success of on-going agile software development projects,"

International Journal of Project Management, vol. 38, no. 3, pp. 165–176, 2020.

[25] T. J. Gandomani and M. Z. Nafchi, "Agile transition and adoption human-related challenges and issues: A grounded theory approach," Computers in Human Behavior, vol. 62, pp. 257–266, 2016.

[26] P. Rosenbaum, Observational Studies, ser. Springer Series in Statistics. Springer, 2002. [Online]. Available: https://books.google.co.uk/books?id=K0OglGXtpGMC

[27] N. V. Flor and E. L. Hutchins, "Analyzing Distributed Cognition in Software Teams: A case study of team programming during perfective software maintenance," in Empirical studies of programmers: Fourth workshop, J. Koenemann-Belliveau, T. Moher, and S. Robertson, Eds. Ablex, 1991, pp. 36–64.

[28] A. Newen, L. D. Bruin, and S. Gallagher, Eds., The Oxford Handbook of 4E Cognition. Oxford University Press, Sep. 2018.

[29] M. Farina, "Embodied cognition: dimensions, domains and applications," Adaptive Behavior, vol. (to appear), pp. 1–16, 2020.

[30] E. Hutchins and T. Klausen, "Distributed cognition in an airline cockpit," Cognition and communication at work, pp. 15–34, 1996.

[31] J. Kiverstein, M. Farina, and A. Clark, The extended mind thesis. Oxford University Press, 2013.

[32] E. Hutchins, "How a cockpit remembers its speeds," Cognitive science, vol. 19, no. 3, pp. 265–288, 1995.

[33] J. Hollan, E. Hutchins, and D. Kirsh, "Distributed cognition: toward a new foundation for human-computer interaction research," ACM Transactions on Computer-Human Interaction (TOCHI), vol. 7, no. 2, pp. 174–196, 2000.

[34] N. Zivlak-Radulovic, M. Miskovic, V. Popovic, D. Zoric, and V. Banjac, "The necessity of an interdisciplinary approach in the treatment of personality disorder-munchausen syndrome," European Psychiatry, vol. 30, no. S1, pp. 1–1, 2015.

[35] M. Jordan, Nature and therapy: Understanding counselling and psychotherapy in outdoor spaces. Routledge, 2014.

[36] L. E. Beutler and A. J. Consoli, "Matching the therapist's interpersonal stance to clients' characteristics: Contributions from systematic eclectic psychotherapy." Psychotherapy: Theory, Research, Practice, Training, vol. 30, no. 3, p. 417, 1993.

[37] A. King, "Enhancing peer interaction and learning in the classroom through reciprocal questioning," American Educational Research Journal, vol. 27, no. 4, pp. 664–687, 1990.

[38] T. Luo and F. Gao, "Enhancing classroom learning experience by providing structures to microblogging-based activities," Journal of Information Technology Education: Innovations in Practice, vol. 11, 2012.

[39] A. Clark and D. Chalmers, "The extended mind," analysis, vol. 58, no. 1, pp. 7–19, 1998.

[40] A. Clark et al., Supersizing the mind: Embodiment, action, and cognitive extension. OUP USA, 2008.

[41] E. Hutchins, "Material anchors for conceptual blends," Journal of pragmatics, vol. 37, no. 10, pp. 1555–1577, 2005.

[42] G. Bateson, "The logical categories of learning and communication," Steps to an Ecology of Mind, pp. 279–308, 1972.

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

[44] M. Donald, "The evolutionary origins of human cultural memory," Handbook of culture and memory, pp. 19–40, 2018.

[45] J. Sutton, "Distributed cognition: Domains and dimensions," Pragmatics & Cognition, vol. 14, no. 2, pp. 235–247, 2006.

[46] A. Becvar, J. Hollan, and E. Hutchins, "Representational gestures as cognitive artifacts for developing theories in a scientific laboratory," in Resources, Co-Evolution and Artifacts. Springer, 2008, pp. 117–143.

[47] C. A. Halverson, "Inside the cognitive workplace: New technology and air traffic control." Ph.D. dissertation, UCSD, 1996.

[48] E. Hutchins, "The technology of team navigation," in Intellectual teamwork. Psychology Press, 2014, pp. 205–234.

[49] C. A. Halverson, "Activity theory and distributed cognition: Or what does cscw need to do with theories?" Computer Supported Cooperative Work (CSCW), vol. 11, no. 1-2, pp. 243–267, 2002.

[50] Y. Wærn, C. Garbis, and H. Artman, "Co-ordination for distributed cognition in control rooms," in 3rd European Conference on Cognitive Science. Siena, Italy, 1999, pp. 31–36.

[51] L. Malafouris, How things shape the mind. MIT press, 2013.

[52] C. Knappett, An archaeology of interaction: network perspectives on material culture and society. Oxford University Press Oxford, 2011.

[53] E. Hutchins, "Cognitive ecology," Topics in cognitive science, vol. 2, no. 4, pp. 705–715, 2010.

[54] J. Sutton, "Material agency, skills and history: Distributed cognition and the archaeology of memory," in Material agency. Springer, 2008, pp. 37–55.

[55] J. Sutton, C. B. Harris, P. G. Keil, and A. J. Barnier, "The psychology of memory, extended cognition, and socially distributed remembering," Phenomenology and the cognitive sciences, vol. 9, no. 4, pp. 521–560, 2010.

[56] K. Michaelian and J. Sutton, "Distributed cognition and memory research: History and current directions," Review of philosophy and psychology, vol. 4, no. 1, pp. 1–24, 2013.

[57] G. Mangalaraj, S. Nerur, R. Mahapatra, and K. H. Price, "Distributed Cognition in Software Design: An Experimental Investigation of the Role of Design Patterns and Collaboration," MIS Quarterly, vol. 38, no. 1, pp. 249–274, 2014.

[58] A. Clark, "Language, embodiment, and the cognitive niche," Trends in cognitive sciences, vol. 10, no. 8, pp. 370–374, 2006.

[59] D. C. Dennett, Elbow room: The varieties of free will worth wanting. mit Press, 2015.

[60] D. Spurrett and S. J. Cowley, "How to do things without words: Infants, utterance-activity and distributed cognition," Language Sciences, vol. 26, no. 5, pp. 443–466, 2004.

[61] T. W. Deacon, The symbolic species: The co-evolution of language and the brain. WW Norton and Company, 1998, no. 202.

[62] A. Jamalian and B. Tversky, "Gestures alter thinking about time," in Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 34, no. 34, 2012.

[63] S. Kang and B. Tversky, "From hands to minds: Gestures promote understanding," Cognitive Research: Principles and Implications, vol. 1, no. 1, pp. 1–15, 2016.

[64] M. Chu and S. Kita, "The nature of gestures' beneficial role in spatial problem solving." Journal of Experimental Psychology: General, vol. 140, no. 1, p. 102, 2011.

[65] D. McNeill, "Gesture, gaze, and ground," in International workshop on machine learning for multimodal interaction. Springer, 2005, pp. 1–14.

[66] Z. Radman, The hand, an organ of the mind: What the manual tells the mental. MIT Press, 2013.

[67] M. Alač and E. Hutchins, "I see what you are saying: Action as cognition in fmri brain mapping practice," Journal of cognition and culture, vol. 4, no. 3-4, pp. 629–661, 2004.

[68] S. W. Cook and S. Goldin-Meadow, "The role of gesture in learning: Do children use their hands to change their minds?" Journal of cognition and development, vol. 7, no. 2, pp. 211–232, 2006.

[69] H. Sharp, H. Robinson, J. Segal, and D. Furniss, "The Role of Story Cards and the Wall in XP teams: a distributed cognition perspective," in Agile Conference. IEEE, 2006, pp. 56–75.

[70] H. Cervantes and R. Kazman, Designing software architectures: a practical approach. Addison-Wesley Professional, 2016.

[71] L. Rising and N. S. Janoff, "The scrum software development process for small teams," IEEE software, vol. 17, no. 4, pp. 26–32, 2000.

[72] J. Lopez-Martinez, A. Ramirez-Noriega, R. Juarez-Ramirez, G. Licea, and Y. Martinez-Ramirez, "Analysis of Planning Poker Factors between University and Enterprise," in 5th Int. Conf. in Software Engineering Research and Innovation (CONISOFT). IEEE Computer Society, 2017, pp. 54–60. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CONISOFT.2017.00014

[73] P. Barthelmess and K. M. Anderson, "A view of software development environments based on Activity Theory," Computer Supported Cooperative Work (CSCW), vol. 11, no. 1-2, pp. 13–37, 2002.

[74] P. Chita, "Agile Software Development – Adoption and Maturity: An Activity Theory Perspective," in Agile Processes in Software Engineering and Extreme Programming, ser. LNBIP, J. Garbajosa, X. Wang, and A. Aguiar, Eds., vol. 314. Springer, 2018, pp. 160–176.

[75] M. Riehle, J. Kempkensteffen, and T. Lincoln, "Quantifying Facial Expression Synchrony in Face-To-Face Dyadic Interactions: Temporal Dynamics of Simultaneously Recorded Facial EMG Signals," J Nonverbal Behavior, vol. 41, pp. 85–102, 2017.

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

[77] 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.

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

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

[80] G. Rugg and M. Petre, A gentle guide to research methods. Maidenhead: McGraw-Hill/Open Univ. Press, 2007, oCLC: 255616300.

[81] C. P. Chai, "Text mining in survey data," Survey Practice, vol. 12, no. 1, 3 2019.

[82] A. Pentland, "The impact of unconscious communication," Gallup Management Journal, September 2009, retrieved on the 20th November 2020. [Online]. Available: https://news.gallup.com/businessjournal/122711/impact-unconscious-communication.aspx

[83] D. L. Roter, R. M. Frankel, J. A. Hall, and D. Sluyter, "The expression of emotion through nonverbal behavior in medical visits. Mechanisms and outcomes," Journal of general internal medicine, vol. 21 Suppl 1, no. Suppl 1, pp. S28–S34, 01 2006. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/16405706

[84] J. L. Tracy, D. Randles, and C. M. Steckler, "The nonverbal communication of emotions," Current Opinion in Behavioral Sciences, vol. 3, pp. 25 – 30, 2015, social behavior.

[85] D. A. Sauter, "The Nonverbal Communication of Positive Emotions: An Emotion Family Approach," Emot Rev, vol. 9, no. 3, pp. 222–234, Jul 2017.

[86] B. Eunson, Communicating in the 21st Century. John Wiley & Sons, Australia, 2008.

[87] J. W. Chesebro and D. G. Bonsall, Computer-Mediated Communication: Human Relationships in a Computerized World. University of Alabama Press, 1989.

[88] H. Clark and S. E. Brennan, "Grounding in communication," in Perspectives on socially shared cognition, 1991.

[89] J. M. Metz, "Computer-mediated communication: Literature review of a new context," Journal of Organizational Computing and Electronic Commerce, vol. 2, pp. 31–49, 04 1994.

[90] H. Park and S. McKilligan, "A systematic literature review for human-computer interaction and design thinking process integration," in Design, User Experience, and Usability: Theory and Practice, A. Marcus and W. Wang, Eds. Cham: Springer International Publishing, 2018, pp. 725–740.

[91] A. Meier and L. Reinecke, "Computer-Mediated Communication, Social Media, and Mental Health: A Conceptual and Empirical Meta-Review," Communication Research, 10 2020.

[92] S. W. Raudenbush, "Statistical analysis and optimal design for cluster randomized trials." Psychological methods, vol. 2, no. 2, p. 173, 1997.

[93] L. C. Page, M. A. Lenard, and L. Keele, "The design of clustered observational studies in education," AERA Open, vol. 6, no. 3, p. 2332858420954401, 2020. [Online]. Available: https://doi.org/10.1177/2332858420954401

[94] L. V. Hedges and E. C. Hedberg, "Intraclass correlation values for planning group-randomized trials in education," Educational Evaluation and Policy Analysis, vol. 29, no. 1, pp. 60–87, 2007.

[95] G. J. P. van Breukelen and M. J. J. M. Candel, "Efficient design of cluster randomized trials with treatment-dependent costs and treatment-dependent unknown variances," Statistics in medicine, vol. 37, no. 21, pp. 3027–3046, 09 2018.

[96] J. Henrich, S. J. Heine, and A. Norenzayan, "The weirdest people in the world?" Behavioral and brain sciences, vol. 33, no. 2-3, pp. 61–83, 2010.

[97] A. Riege, "Validity and reliability tests in case study research: a literature review with "hands-on" applications for each research phase," Qualitative Market Research, vol. 6, no. 2, pp. 75–86, 2003.

[98] C. Robson, Real World Research - A Resource for Social Scientists and Practitioner-Researchers, 2nd ed. Blackwell Publishing, 2002.

[99] L. S. Connaway and R. R. Powell, Basic research methods for librarians. ABC-CLIO, 2010.

[100] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," Empir Software Eng, 2009.

[101] M. V. Zelkowitz and D. Wallace, "Experimental validation in software engineering," Information and Software Technology, vol. 39, no. 11, pp. 735–743, 1997, evaluation and Assessment in Software Engineering. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584997000256

[102] J. Carver, J. Van Voorhis, and V. Basili, "Understanding the impact of assumptions on experimental validity," in Proceedings of the 2004 International Symposium on Empirical Software Engineering, ser. ISESE '04. USA: IEEE Computer Society, 2004, p. 251–260.

[103] N. Juristo and A. M. Moreno, Basics of Software Engineering Experimentation, 1st ed. Springer Publishing Company, Incorporated, 2010.

[104] S. Morrison-Smith and J. Ruiz, "Challenges and barriers in virtual teams: a literature review," SN Applied Sciences, vol. 2, no. 6, p. 1096, 2020. [Online]. Available: https://doi.org/10.1007/s42452-020-2801-5

APPENDIX. QUESTIONNAIRE

TABLE 7: Questionnaire used in the study

	Questions
1	Do you work online or offline?
2	How do you receive information about the tasks you need to perform in order to do your job?
3	What form do these tasks usually have?
4	Where do you store information about such tasks?
5	How do you measure the progress on your tasks?
6	Do you normally understand the tasks you are given?
7	When you don't, which communication channel/s do you use to comprehend them?
8	How often do you adopt this/these communication channel/s?
9	How often do you meet with people in your team?
10	Describe how such meetings usually go (where they are held, how many people attend them, what is their atmosphere, how people sit and communicate etc....)
11	During these meetings, do you listen or talk more? Do you ever get distracted?
12	At these meetings, does everyone communicate for the same amount of time? Or is there any imbalance in the way communication is implemented among team members?
13	During such meetings, is it easy for you to communicate and express your ideas?
14	Do you think communication is effective during these meetings?
15	How do you know that communication is easy and effective?
16	When communication is easy and effective: • How do other people behave? • Does it matter to you how they behave? • What is your body posture? • Do you look into someone's eyes?
17	How often do people look others in the eyes? • What does it mean to you when this happens? • (if applicable) How does this affect your communication style?
18	Do you use gestures or specific body postures to convey a certain message or to make communication easier and/or more effective? If so, how?
19	Does it ever happen that you and another person touch each other (accidentally or intentionally?) during a meeting
20	Do you use Agile Methodologies at all? If so, which one exactly?
21	Do you work in a team? If so, how many members does your team have?