Journal of the
Brazilian Computer Society

RESEARCH Open Access

Industrial and OSS developers' profiles: a family of experiments to evaluate a pioneering neuro-linguistic method for preferred representational systems automatic detection

Methanias Colaço Júnior[1,2,3*], Breno Santana Santos[2,3], Manoel Mendonça[1], Daniela Corumba[3] and Mario André de F. Farias[1]

* Correspondence: mjrse@hotmail.com
[1]Software Engineering Laboratory (LES), Federal University of Bahia (UFBA), Salvador, Bahia, Brazil
[2]Postgraduate Program in Computer Science (PROCC), Federal University of Sergipe (UFS), São Cristóvão, Sergipe, Brazil
Full list of author information is available at the end of the article

Abstract

Software projects use mailing lists as the primary tool for collaboration and coordination. Mailing lists can be an important source for extracting behavioral patterns in the software development. A new approach for that is the use of Neurolinguistic theory to determine what is the Preferred Representational cognitive System (PRS) of software engineers in that specific context. Different resources and cognitive channels are used by developers in order to achieve software understanding. An important question on this matter is: What types of representational systems are preferred by software engineers? This paper presents a psychometrically based neurolinguistic method to identify the PRS of software developers. Experimental evaluation of the approach was carried out in three experiments to assess the Preferred Representational System of developers at Industrial and OSS (Apache server and Postgresql) mailing lists. For the OSS projects, the results showed that the PRS scores of the top-committers clearly differ from the general population of the projects. For industry, the experiment showed that the developers indeed have a PRS. Finally, for both scenarios, the qualitative analysis also indicated that the PRS scores obtained are aligned with the developers' profiles, considering that alignment is essential to effective communication within the team and enhances the development process due to a better software comprehension.

Keywords: Open source, Industry, Neuro-linguistic, Mental imagery, Experimental software engineering

Introduction

Developing and maintaining software systems is an arduous task. Large systems are complex and difficult to understand. In order to understand them, the developer must construct a mental model of the software works and structure, i.e., its domain, architecture, and execution flow [1].

In the comprehension process, developers use different resources and representational systems, such as (1) examples, analogies, and code execution; (2) visual descriptions, diagrams, and graphic models of the system; and (3) textual descriptions and source code analyses. Clearly, these resources are complementary and may be combined. However, is there a Context-Specific Preferred Representational System (PRS)? Or, is there a preferred order or combination of the representational systems in the understanding process?

Visual resources, like diagrams and non-conventional visualization metaphors, are being increasingly used in software engineering [2]. Studies show that the way software engineers process those resources impacts on the success of that processing [3], for both text [4] and diagrams [5]. However, we do not know complete studies that evaluate what types of representational systems are preferred by software engineers.

This is a broad question in the sense that different people may have different preferences in different contexts. Actually, the conception that different representational ways for cognition exist is well accepted in the psychology area [6–8]. However, this statement has raised new theories such as Neuro-linguistic, which proposes the use of a PRS in specific contexts [9]. Internal mental processes such as problem-solving, memory, and language consist of visual, auditory, and kinesthetic representations that are engaged when people think about or engage in problems, tasks, or activities. Internal sensory representations are constantly being formed and activated. Whether making conversation, writing about a problem, or reading a book, internal representations have an impact on one's performance. The Preferred Representational System is the one that the person tends to use more than the others to create his/her internal representation.

Bandler and Grinder, Neuro-linguistic Programming (NLP) champions, claim that people say sensory-based words and phrases, or verbal cues, which indicate a context-specific visual, kinesthetic, or auditory processing [9, 10]. These affirmations divide researchers of cognitive psychology area. Some have not found evidences for the declarations [11]; hence, they were criticized by the lack of concept understanding [12]. Meanwhile, others have shown empirical scientific evidence and the need to expand researches [13, 14].

Thus, motivated by the psychometric text analysis presented by Rigby and Hassan [15], we developed a psychometrically based neurolinguistic analysis tool. Our tool, NEUROMINER, uses Linguistic Inquiry and Word Count (LIWC) to classify developers' Preferred Representational Systems (PRS) from mailing lists of their projects. NEUROMINER combines text mining and statistic analysis techniques with NLP sensory-based words in order to classify programmers.

NEUROMINER was used in three experiments which analyzed top committers and subjects of two large-scale OSS projects (Apache Server and Postgresql), as well as industry developers of closed-source projects. For OSS projects, the results showed that the measured PRS scores can indeed differentiate top committers from the general population. For industry, the developers indeed have a PRS. Finally, for both scenarios, the qualitative analysis also indicated that the PRS scores obtained are aligned with the developers' profiles.

The rest of this paper is organized as follows. The next section presents the cognitive and learning styles. The Neuro-linguistic programming section introduces NLP. The Text mining basis section reports text mining definitions used throughout the article. The Linguistic Inquiry and Word Count for Neuro-linguistic section describes our approach to LIWC and to mining software development mailing lists. In The family of experiments section, we detail the experimental evaluation of our approach. Related works is the seventh section. Finally, the Conclusion and future work section closes the paper.

Cognitive and learning styles

In the scientific community focused on cognitive research, it is widely accepted that the way people choose, or tend to choose, to learn has an impact on the learning performance [16]. There is a wide range of definitions, theories, models, interpretations, and measures dealing with the learning process. Among them, two items have led to several valuable insights in many research fields: learning styles [16] and cognitive styles [17]. This article does not aim to theorize on these concepts or even create new definitions for them. It will just use them to build the necessary theoretical background for the work presented here.

The terms learning style and cognitive style have been defined in different ways by different researchers. Allport [18] described cognitive style as a common habit or a notably personal way of solving, thinking, noticing, and recalling problems. Garity [19] noticed that a cognitive style has been used to define the cognitive process of thinking, noticing, and recalling. Cognitive style is how subjects process information and prefer to learn. Badenoch [20], in his study about "personality type, preference of learning style and instructional strategies", claims that the learning style theory intends to investigate the learning process and product, in order to understand the interactions in the learning environment. In his opinion, the type of cognitive personality, however, is a classification of the theory of the learning style. Hartley [21] defines "cognitive styles" as the ways that the subjects lead their cognitive tasks and "learning styles" as the ways that the subjects lead their learning tasks.

In order to cope with these concepts, avoiding those many and sometimes confusing definitions, this paper considers that learning styles are ways that each individual uses to process information and to understand some subject, and cognitive styles are common ways in which individuals process information, transform it in internalized knowledge, and recall it when necessary. Each individual uses learning styles to understand and comprehend a subject; however, in order to process and transform the learned information in readily recoverable knowledge, or even to better assimilate and optimize the subject, cognitive styles are used. In summary, there is tenuous difference between learning and apprehending, between learning and cognition, between styles of "acquiring" knowledge and styles of "using, optimizing, and transforming" knowledge.

We emphasize that learning styles have been cited to support every context involving representation systems, just as this work is based only on cognitive styles, which are used by individuals to process information based on their perception of reality and previous experiences (see Neuro-linguistic programming section); therefore, the dimensions of learning styles are outside the scope of this work.

Representative dimensions and measurements of the cognitive styles

In the 70's, the Cognitive Psychology researchers intensified discussions about how to measure the subjects' intellectual abilities. Hunt et al. [22] proposed to use laboratory tests to investigate the construction and usage of such human abilities. Estes [23] proposed tests to measure cognitive abilities and as a mean to find ways to improve cognitive performance. Underwood [24] proposed to apply tests to detect the differences between subjects and to use this information as the basis for nomethetics, a psychology/psychiatry theory that analyzes the influence of a patient's cognition about a disease in his self-healing process.

Cognitive style measurement is value based. This so called ability-factor model initially relied heavily on measuring how much as opposed to how come [25]. However, from the cognitive style point of view, it is almost useless to measure "how many hours one spent on a task" and not measuring "how the task was done". Style measurement should focus on "how it was done" as opposed to "how much was done". For that, Lohman and Bosma [25] propose the following principles:

(a) Apply tasks in which the individual differences are clearly reflected, mostly in measurements of "how it was done", i.e., of the way the tasks that each one solves are evaluated according to different strategic solutions taken;
(b) Have some guidance in order to make clear the inferences about the strategies from the answers given for each task. Even when facing different ways to solve some tasks, the ways one finds to solve them will be associated with the dependent measures to each subject, such as speed of reasoning and reflex;
(c) Have a Measurement Model that captures both the subject profile, his strategies for problem resolution and correlated measures [26]. In order to guarantee consistency, this measurement model needs to be validated and packaged for reuse;
(d) It is necessary that the Measurement Model supports association and relationship analysis between different strategies.

When trying to measure a cognitive style, the strategies assessed may not be classified as belonging to a category of a specific style. For instance, the fact that a Software Engineer prefers to use diagrams in a situation in which he/she needs to comprehend a class, does not imply that he/she has visual preference in general. Not all strategies represent a style with the same clarity [25].

The Cognitive Style Analysis (CSA) champions, Riding and Cheema [17], describe that cognitive styles have two fundamental and independent bipolar dimensions: the wholist–analytical (WA) and the verbalizer–imagery (VI). The wholist–analytical dimension of cognitive style means the habitual way in which an individual processes and organizes information. Some individuals process and organize information into its component parts (analytics), others retain a global or overall view of information (wholists).

The verbalizer–imagery dimension describes the common ways in which the subject represents the information in memory while he/she thinks. For Riding [27], verbalizers convert the information they read, see, or listen to, into words or verbal associations. Imagers, on the other hand, convert the information they read, see, or listen into spontaneous and frequent mental pictures.

According to Riding [28], the validity of the cognitive model proposed in the Cognitive Styles Analysis approach is supported by the evidence that the WA and VI dimensions are independent from each other, separated and independent from the intelligence, but they interact with personality and are related to behaviors such as learning performance, learning preferences, subject preferences, and social behavior.

The CSA approach is the basis for the work of Fleming [29], which is one of the main influences on our work. Fleming's work proposes a questionnaire to cluster subjects according to their main sensorial abilities in creating learning styles. Fleming [29] was influenced by neurolinguistics [9], correlating words or expressions used by subjects with their preferred ways of representing information in memory.

Based on neurolinguistics, our work considers that the words or expressions used by subjects are related with their preferred ways of representing information in memory. This paper proposes a method specifically conceived to capture the cognitive preferred representational system (PRS) by software engineers. The PRS is the way a person, in specific contexts, prefers to use to communicate and learn [6–8].

In the next section, the relevant concepts on Neurolinguistics and PRS will be presented.

Neuro-linguistic programming

History and some concepts

Neuro-linguistic programming (NLP), created in the 70's, consists of a set of techniques in which the neurological processes, behavioral patterns, and a person's language are used and organized to achieve better communication and personal development. The term NLP is broadly adopted in education, management, and training fields. However, although evidences of NLP have been published as model for comprehension and learning [30], few academic works exist on the subject.

NLP claims that people are intrinsically creative and capable, acting according to how they understand and represent the world, instead of how the world is. Literature constantly cites Korzybski's statement [31] "the map is not the territory", a reference to individual understanding that everyone has—mental model—according to his/her experience, beliefs, culture, knowledge, and values.

For Tosey and Mathison [13], NLP scientific research group members, NLP is presented as an epistemological perspective, with scientific principles which are not usually presented. The first works published by Bandler and Grinder [9, 10] were based on the models of Fritz Perl, Gestalt founder, Virginia Satir, researcher in family therapy, and Milton Erickson, doctor in medicine, master in psychology, and hypnotherapist recognized worldwide. As a consequence, the epistemological view of NLP presents a roadmap to develop the necessary scientific basis to support its beliefs. The research reported in this paper explores this path by scientifically characterizing the use of preferred representational systems for cognition.

This representational system (or internal representation) is highly dependent on context (i.e., it varies with the situation) [12]. This way, some people, in specific contexts, may prefer to use one or more basic systems to communicate and learn [6–8]. Most authors in the area recognize the following basic systems [6–8, 10, 12]:

(a) Visual, that involves internal image creation and the use of seen or observed things, including pictures, diagrams, demonstrations, displays, handouts, films, and flip charts;
(b) Auditory, that involves sound reminders and information transferred through listening; and
(c) Kinesthetic, that involves internal feelings of touch, emotions, and physical experience (holding and doing practical hands-on experiences).

We use all of our senses all of the time and, depending on the circumstances, we may focus on one or more of them—for instance, when listening to a favorite piece of music, we may close our eyes to more fully listen and to experience certain feelings. In order to see things more clearly, we might need to close our eyes and visualize the situation, person, or place.

So, we all use each of the senses and each of us also has a Preferred Representational System (PRS), one that we use most when we speak, learn, or communicate in any way. For example, when learning something new, some of us may prefer to see it or imagine it performed, others need to hear how to do it, others need to get a feeling for it, and yet others have to make sense of it. In general, one system is not better than another, and sometimes, it depends on the situation or task that we are learning or doing as to which one or more representational systems might be more effective than another.

Supporters of NLP believe that word predicates let us know what is the person's state of consciousness. They believe that specific, sensory-based, word predicates are chosen when a person is using a specific representational system. The predicates indicate what portion—of internal representations - they bring into awareness [10]. Such predicates may be identified and used to improve communication among the analyzed subjects, for example.

One of the major problems in communication, be it informal or technical, is the difficulty to arouse interest on the receiving end, the person who is reading or listening to your message. Many times, the person who receives the message does not assimilate what is being transmitted, be it a simple message or a technical diagram. NLP can then be one approach to improve communication. The challenge lies in identifying the representational system that is being used by the subject and match the same system for empathy construction. Empathy is an emotional response to other person through sharing other's affective state, as well as it is a cognitive capability to think in other person's perspective [32, 33]. The matching consists of identifying the predicates that indicate a representational system and use them, or other predicates that belong to the same system, for communication [10].

In order to exemplify this matching process, consider the following question "have you seen the logic of the algorithms that I showed you?", and the following answer "not yet, I am going to examine them carefully, once I get a clear picture of the whole system." This is a coherent answer to the question from the sensory system matching perspective. The sensory-based words "seen" and "showed" in the first phrase indicate a visual processing, and the response used the same system through the visual sensory words "examine them" and "clear picture".

In this context, detecting the developers' representational preferences may enhance the empathy in the team communication, i.e., each member may be more stimulated in his/her Preferred Representational System, enhancing the effectiveness of communication, software comprehension, and the solution of activities of development and maintenance.

Allocating a person in a task, considering his/her technical abilities as well as his/her personality, is essential for the success of any software project. The productivity secret is to adjust the project needs with its members' personalities. Detecting, for instance, that a system analyst barely uses his/her visual representational system may help solve his/her difficulties with project diagrams or stimulate his/her reallocation to another activity. Many times, a member is lost because of wrong job allocation. A good programmer may become a not-so-good analyst. In other situations, a person's preferential cognitive system may not match his/her colleagues' profile, or the way the organization works.

Our research deals with the identification of sensory-based words used by developers in discussion lists. We then use these words to characterize the preferred representational systems of the developers and analyze these against their profile and role in the projects.

Neurolinguistic criticism

NLP experimental research basis is insufficient. The literature in academic journals is minimal, and Thompson et al.'s study [34] is a good example. There has been virtually no published investigation into how NLP is used in practice. The experimental research consists largely of laboratory-based studies from the 1980's and 1990's, which investigated two particular notions from within NLP, the 'eye movement' model, and the notion of PRS.

Heap [35], in particular, has argued that on the basis of the existent studies, these particular claims of NLP cannot be accepted. Heap conducted a meta-analysis of these and appears entirely justified in criticizing the unequivocal claims made in NLP literature. It is notable, however, that Heap's meta-analysis included many postgraduate dissertations. His bibliography refers only to sources of abstracts of those dissertation studies, not to the dissertations themselves. Thus, his meta-analysis appears based on the reported outcomes of these studies, not on critical appraisal of their methodology or validity.

Einspruch and Forman [12] and Bostic St.Clair and Grinder [36] have also argued that the types of study reviewed by Heap are characterized by problems affecting their reliability, including inaccurate understanding of NLP claims and invalid procedures due to (for example) the inadequate training of interviewers, who therefore may not have been competent at the NLP techniques being tested. Heap himself offers only an 'interim verdict' and acknowledges Einspruch and Forman's view that 'the effectiveness of NLP therapy undertaken in authentic clinical contexts of trained practitioners has not yet been properly investigated' [35].

Given these concerns, for example, Tosey and Mathison [13] suggest that the existing body of experimental research cannot support definitive conclusions about NLP. It seems clear that there is no substantive support for NLP in this body of experimental research, yet it also seems insufficient to dismiss NLP.

Our study does not test NLP techniques, but rather shows an association between NLP based-measures and developers' roles and profiles.

Text mining basis

Our work is based on text mining (TM), a technology for analysis of large collections of unstructured documents, aiming to extract patterns or interesting and non-trivial knowledge from text [37].

Preprocessing

Similar to conventional data mining, text mining consists of phases that are inherent to knowledge discovery process [38]. Classification of knowledge discovery phases may vary for different authors, but most comprise at least data selection, preprocessing, mining, and assimilation. Text mining pays special attention to preprocessing because its data is unstructured for computer analysis. In other words, after setting the base with texts to be mined, it is necessary to convert each document to a format suitable for a computational algorithm.

One may use three different ways—Boolean, probabilistic, or vector-based models—to structure the information of a text document for computational analysis. The vector model utilizes geometry in order to represent documents. Introduced by Salton, Wong, and Yang [39], this model was developed to be used in a retrieval system called SMART. According to the vector model approach, each document is represented as a term vector, and each term receives a weight that indicates its importance in the document [39].

In more formal terms, each document is then represented as a vector, which is composed of elements organized as a tuple of values: $d_j = \{w_{1j} ,.. , w_{ij}\}$, where d_j represents a document, and w_{ij} represents a weight associated to each indexed term of a set of t terms of the document. For each element of the term vector, a dimensional coordinate is considered. This way, the documents can be placed in a Euclidian space of n dimensions (where n is the number of terms), and the position of the document in each dimension is given by the term weight in this dimension.

In this model, the consultations are also represented by vectors. This way, the document vectors can be compared with the consultation vector, and the similarity between them can be easily computed. The most similar documents (those that show the closest vectors to the consultation vector) are relevant and returned as a response to the user. Besides, documents that show the nearest vectors can be considered similar to the target document.

A term vector is built by the following steps.

Term extraction

Researchers from the information retrieval field claim that the main difference between data and information retrieval is exactly the relevance of the information obtained [40].

In general, not all terms that compose a document are relevant when one intends to extract high-level information. So, in order to compose a term vector for a text, it is necessary to identify words with high semantic content, selecting only those that are meaningful for the objective at hand.

The task of term extraction from a document consists of various steps, all of them contributing for the final purpose of producing a vector with high semantic content [41]. They are described as follows:

(d) Lexical analysis: The original document is not always represented in a purely textual format. Therefore, it is necessary to convert it to a standardized format, eliminating any attributes of presentation formatting.
(e) Character conversion to uppercase or lowercase: Such procedure enables equal words written with a character in a different format in uppercase or lowercase—for example, neuro and Neuro may be interpreted as the same term.
(f) The use of a word list to be ignored: commonly called stopwords. This list consists of a relation of words that have no significative semantic content (e.g., prepositions, conjunctions, articles, numerals, etc.) and consequently are not relevant for text analysis.
(g) Morphological normalization: aiming to cluster terms with the same conceptual meaning, e.g., the words compute and computation. A conversion algorithm of terms to radicals may be applied in this case. In the example, the words "compute and computation" have the same radical "comput", so they can be reduced to this term.
(h) Selection of simple or compound words: in some cases, during the preprocessing of a document, several joint words (phrases) may be managed as a single term. This selection can be done using predefined word lists or statistical and syntactic techniques.
(i) Normalization of synonyms: Words with the same meaning can be reduced to a specific term, for example, the acronym SEL and the composition Software Engineering Lab, both have the same meaning.
(j) Structural analysis: This step consists of associating information to each term regarding its positioning in the document structure, in order to distinguish it from a homonym term situated in another position.

Assigning weights

The process of associating numeric values to each term previously extracted is known as assigning weights. In general, the settlement of the term weight in a document can be resolved with two paradigms [42]:

(k) The more a term appears in the document, the more relevant the term is to the document subject;
(l) The more a term occurs among all documents of a collection, the less important the term is to distinguish between documents.

This calculation can be done in two ways:

(m)Binary or Boolean—The values 0 and 1 are used to represent, respectively, the absence or presence of a term in the document.
(n) Numeric—It is based on statistical techniques regarding the term frequency in the document.

The numeric weights can be represented by measures such as

- Term frequency (tf): simple method which consists of the number of times that a term w_i occurs in a document *d*. This method is based on the premise that the term frequency in the document provides useful information about the relevance of this term for the document.
- Document frequency (DF): it is the number of documents in which the term w_i occurs at least once.
- Inverse document frequency (idf): it defines the relevance of a term in a set of documents. The bigger this index is, the more important the term is to the document in which it occurs. The formula to calculate idf is as follows:

$$idfi = \ log \ (|D| \ /|\{d : t_i \ \varepsilon \ d\}|) \tag{1}$$

where $|D|$ represents the total of documents, and $|\{d\text{: } t_i \ \varepsilon \ d\}|$ represents the number of documents where the term t_i appears.

- tf-idf: It combines the term frequency with its inverse frequency in the document, in order to obtain a higher index of its representativeness. The formula to calculate tf-idf weight is as follows:

$$(tf\text{-}idf)_{ij} = tf_{ij} \ \chi \ idf_i \tag{2}$$

Grammatical classes and noun phrases

To further strengthen the semantic meaning of the structured data, our work uses word composition. Words that have similar semantic and syntactic behaviors can be clustered in the same class, creating syntactic or grammatical categories, more commonly named parts of speech (POS). The three main ones are noun, verb, and adjective. The nouns refer to people, animals, concepts, and things. The verb is used to express action in a sentence, whereas the adjectives express noun properties.

The POS detection is important, because in specific contexts, two or more words with different grammatical categories may have one unique meaning. The semantical composition of words is known as a Noun Phrase [43]. Noun phrases (NPs) cluster words in a context, and its detection can improve the search accuracy in texts. Usually, a noun is the central element (head part) which determines the syntactical character of a NP, and a verb or an adjective modifies this noun (mod part).

In order to implement NP detection, it is necessary that a dictionary specifies which words can appear together. In general, it is not necessary to store words in a compound way because this process demands time and does not enhance the system efficiency significantly. What can be done is to store information about the distance between words, and the consultation technique is responsible for evaluating whether words are adjacent or not.

NEUROMINER, the tool discussed in this article, uses the vector spatial model, transforming the developer's emails into vectors, classifying the words grammatically and identifying NPs, as well as assigning weights to the extracted terms.

Linguistic Inquiry and Word Count for Neuro-linguistic

Motivation

We identified research that tries to pinpoint people's preferred representational systems, but those researches are only in psychology, and in domains like sports and education [29]. We also found some software engineering papers that use text mining to identify developers' general emotional content. However, these papers do not try to relate the developer's personality, or other psychological aspect, to the software engineering activities themselves [15, 44]. This gap of knowledge stimulated us to use text mining to investigate the association between a psychological concept—PRS—and software development roles and activities.

Our tool, NEUROMINER, uses Linguistic Inquiry and Word Count (LIWC) to classify the Preferred Representational Systems (PRS) of developers in a given context. We could not find any tools that make automated neurolinguistic text analysis and, as discussed later, our LIWC approach can be adapted to other domains.

Finally, due to the scarcity of scientific research about NLP itself, this paper generates the opportunity to show empirical results of applying one of its principles to our, human-intensive, domain.

Neurominer

NEUROMINER combines statistic and text mining techniques with sensory predicates of NLP, aiming to classify programmers' PRS.

The basic characteristics of NEUROMINER are:

- Use of a neurolinguistic dictionary;
- Use of ANOVA for PRS classification. An ANOVA is an Analysis of the Variation present in an experiment. It is a test of the hypothesis that the variation in an experiment is no greater than that due to normal variation of individuals' characteristics and error in their measurement. In this way, ANOVA was used to classify the PRS of developers, statistically analyzing the differences between the means of each Representational System (RS) for each individual;
- Use of an ontology to identify Software Engineering and neurolinguistic terms combined in noun phrases;
- Use of synonym normalization resources with dictionaries for Brazilian Portuguese [45, 46] and for English [47, 48].

This paper will not focus on Neurominer internal architectural, but rather in its NLP and PRS classification approach.

Building and using a NLP dictionary

According to NLP, the words a person chooses to describe a situation—when they are specific to representational system (i.e., sensory-based)—let us know what his/her consciousness is. This predicate indicates what portion of internal representations the person brings into awareness [10].

The goal of our work is to identify the most used RS and the percentage of use of the others. For this, we have adopted a LIWC approach similar to the one presented by [15]. As shown in Table 1, it uses a NLP dictionary with four basic dimensions composed of sensory-based words or phrases [10, 14].

The Concept dimension was created to increase contextual classification power. A noun phrase (NP) such as 'brilliant algorithm' indicates a visual PRS cue used in the context of software engineering. The tag column of Table 1 indicates that the dimension is part of a modifier (PRS) or head (SE context) of the NP. In this very simple way, NPs formed with SE ontological concepts have a bonus multiplied to the score in our text mining approach.

The concepts were extracted from software document ontology discussed by Witte [49] and described by Wongthongtham [50], which is based on various programming domains, including programming languages, algorithms, data structures and design decisions such as design patterns and software architectures. Our goal is to verify the direct relation of sensory-based words with Software Engineering context. This way, we can find noun phrases formed with ontological concepts and sensory-based words or phrases, our first innovation.

Email mining with Neurominer

Figure 1 summarizes the text mining main steps. The approach is summarized only briefly, since details about preprocessing [49], and clean messages [15, 51] have already been published.

Step 1 includes steps such as stemming, part-of-speech tagging, and noun–phrase detection. For example, in the latter step cited, we use the MuNPEx approach (Multi-Lingual Noun Phrase Extractor) [52].

After downloading the email archives, the system parses each email for meta-data as discussed by Bird [51], and places its relevant information into a data mart [53]. This data mart was designed based on a software engineering data warehousing architecture proposed by us in previous papers, our first innovation [54, 55].

The process only uses the text actually written by the sender and its timestamp. It removes all diffs, attachments, quoted replies, signatures, code, and HTML that is not part of a diff.

We adopted a daily frequency-based cumulative approach. In step 2, the system finds and counts the senders' sensory-based words and phrases by month, considering the NLP dimensions in the dictionary.

Table 1 Neurolinguistic dimensions

Dimension	Example word	Tag
Visual	'brilliant'	Mod
Auditory	'dissonant'	Mod
Kinesthetic	'concrete'	Mod
Concepts	'algorithm'	Head

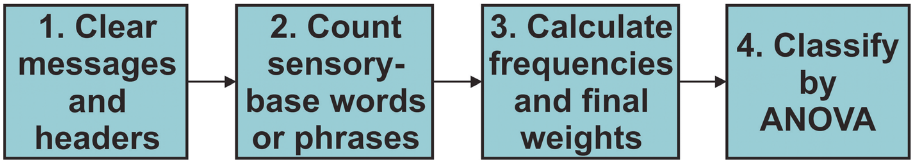
1. Clear messages and headers
2. Count sensory-base words or phrases
3. Calculate frequencies and final weights
4. Classify by ANOVA

Fig. 1 Text mining process chain

In step 3, the system uses a text mining approach for the NLP classification of individuals, instead of the traditional document classification, our second innovation. In it, the set of all emails written by a developer is treated as a 'big text' to be classified. A simple approach for that is to count all the words found in all emails of a developer and verify the percentage of each representational system. However, aiming more detailed analyses of evolution, the system considers the daily frequencies of the words.

Our alternative to the basic tf-idf formulation (see Text mining section) computes weights or scores for sensory-based words. The values are positive numbers so that it captures the presence or absence of the word in a month. Equation (3) indicates that neuro weight assigned to a word j is the term frequency tf(j) (i.e., the ratio between word count and the sum of number of occurrences of all words) modified by scale factor for the importance of the word. The scale factor, for our approach, is called daily frequency df(j), which is the ratio between the number of days containing word j and the number of loaded days. Thus, when a word appears in many days, it is considered more important and scale is increased.

$$neuro(j) = (tf(j) \ + \ df(j)) \ x \ b \tag{3}$$

In addition, a bonus *b* is also multiplied to the measure. The bonus can be 1 or 2, where *b* will be equal to 2 if term is a NP or phrase, and 1 if term is a simple word. This bonus is determined in the LIWC dictionary used to classify individuals PRS. It was also agreed weight 2 for NP and sensorial phrases in order to highlight their importance in relation to simple terms.

At the end of each month, the term weights are recalculated and a general total of weights (final weight) are stored for each representational system. Lastly, each representational system monthly mean is computed.

In the step 4, we use ANOVA (analysis of variance) to determine if the monthly means for each different RS are statistically different.

The family of experiments

The rest of this paper describes an experimental evaluation of our approach. The presented experimental processes follow the guidelines by Wohlin et al. [56]. For each experiment, first sections will focus on the experiment definition and planning. The following sections will present the obtained experimental results.

First and second experiments

This section will focus on the first and second experiments realized in two large-scale OSS projects: Apache Server and Postgresql.

Goal definition

The main goal of our study is to evaluate if OSS top committers have a PRS. This goal is formalized using the GQM goal template proposed by Basili and Weiss [57] and presented by Solingen and Berghout [58]: *Analyze* Project top committers *with the purpose of* evaluation *with respect to* NLP context-specific Preferred Representational Systems *from the point of view of* software engineering researchers *in the context of* development mailing lists of OSS projects.

Planning

For the context selection, the experiment will target OSS projects.

Hypothesis formulation The issues we are trying to explore are as follows:

(o) We are interested in verifying if OSS top committers have a PRS.

(p) Besides that, we believe top committers are more kinesthetic than auditory and visual. Our belief is that experienced programmers of the OSS community rely heavily on their experiences, and are less dependent on visual and auditory artifacts than the general population of OSS software engineers.

Considering the arduous manual work of searching for valid emails used by top committers and, as a consequence, the small sample size due to the low number of top committers, a formal statistical test will not be performed for the second issue.

However, considering the large number of emails that will be mined, the test of the existence of a PRS top committer for each selected will have large power. We will also do a detailed qualitative analysis of the top committers' profiles in order to sanity check NEUROMINER measures.

NEUROMINER will be used to calculate the final weights for each representational system, as well as representational systems monthly means (see Email mining section).

Formally, the hypothesis we are trying to confirm is:

Null hypothesis H_0: OSS top committers have the same representational system monthly mean.

H_0^{PRS}: μ(Visual final weight) = μ(Auditory final weight) = μ(Kinesthetic final weight)

Alternative hypothesis H_1: At least one of the representational systems' monthly means is different from the others.

Participant and artifact selection To answer our research questions, we extracted email messages from the Apache [59] and Postgresql Projects [60] mailing lists. For the Apache, we analyzed the body of all email messages between 1996 and 2005 (35,483 messages) and selected the four developers who had the greatest number of commits. Those are the same developers studied by Rigby and Hassan [15]. For Postgresql, we analyzed the body of all email messages between 1997 and 2006 (57,159 messages) and also selected the four developers who had the greatest number of commits. In both projects, two top committers still contribute to the project, and others have already left.

We also created clusters of all other developers for both projects. During data reporting, we will refer to this general population measures as the cluster.

The analysis is completely non-intrusive to developers as the data was drawn directly from the project mailing lists. For each developer and cluster, once a month, we calculated the PRS using the method described in the Email mining section. At the end, we had one data point of mined emails per month for each subject. Clusters were mined for 3 years (36 months). Top committers were mined for the last 10 years, but data points were produced only for those months in which they posted at least one email at the project discussion list. NEUROMINER then tested the population distribution and calculated the analysis of variance of the monthly PRS scores for each participant (all calculation was double checked using SPSS [61]). The population distribution for each sample is normal.

Instrumentation—Neurominer OLAP module The OLAP module was developed to provide graphics generation and analytical navigation in the data that describe the developers' profile. Next, the presentation of some OLAP features will be done with results of the Apache project.

Looking at the accumulated scores in Fig. 2a (See the chart on the left), it is observed that the predominant profile for the top committer B, one of Apache top committers, is the visual. In the chart on the right (See Fig. 2b), the evolution of the profile of this developer can be seen in the period from January to December 1999.

Another important aspect is the analysis of the mentioned terms that scored higher. For this, a drill-down can be done together with a ranking, which give access to 10 terms or the 10 phrases that have achieved the highest scores. The result is shown in Fig. 3, in which it is interesting to note the presence of concepts of Software Engineering, such as Server, Compiler, and Module, combined with sensory words. Going further, it is also interesting to note the term pain as one of the predominant. As the top committer B has already left the project, this may be an evidence of dissatisfaction in his latest posts, since the term pain indicate grief or discomfort.

Experiment operation

This section describes the data validations for the performed experiments.

Data validation In addition to analysis of metadata and selection described in the Neuromine section, the experiments performed the following validations:

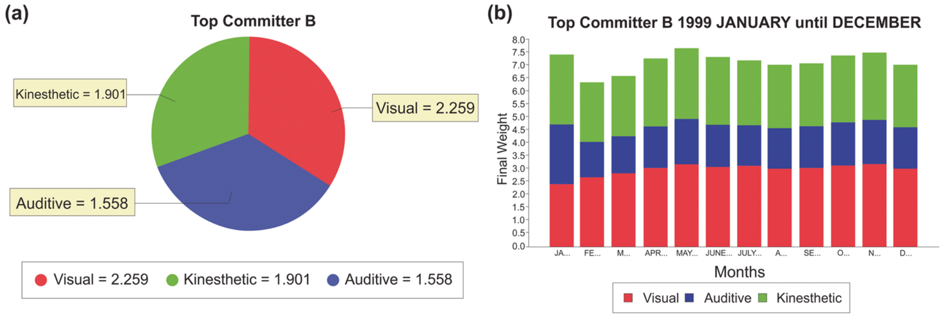

Fig. 2 **a** Current profile of the top committer. **b** Profile evolution of the top committer B

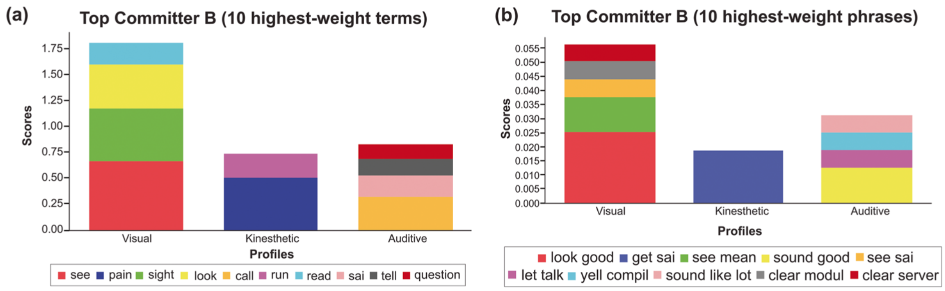

Fig. 3 a Terms most scored by top committer B. **b** Phrases most scored by top committ

(q) With OLAP module, for each developer, the terms and phrases that most said were checked in emails of source (until 10 emails), checking if really contextualize something said by the individual. The concern was to avoid that the term or phrase was part of an unspoken text by the sender, such as quotes and phrases of others, common in fixed subscriptions.

(r) All calculations for ANOVA were revalidated using SPSS [61].

Results

Tables 7 and 8 (see Appendix 1) summarize our results. The column Totals represents the number of months (data points for each participant), days, and emails. For each representational system, the final weight is shown for the set of all sensory-based words found and the monthly average of this weight. The column ANOVA *p* value reports *p* values for the null hypothesis.

Aiming to facilitate the visualization of the results, we create Tables 2 and 3 based on Tables 7 and 8 in Appendix 1, containing only primary information (participants, project output signal, monthly mean for each PRS, and ANOVA *p* value) to analyze the results.

Analysis and interpretation

For the statistical testing, we established an apriority significance level (α) of 0.05. Tables 2 and 3 show that our first hypothesis is accepted as we obtained the *p* value of 0.000 for all means but one, developer G. The results for the clusters and developers A–F and H are significantly lower than 0.05, strongly rejecting the null hypotheses.

Table 2 Overview of the results for the Apache top committers

Participant	Left the project?	Monthly mean			ANOVA *p* value
		Visual	Auditive	Kinesthetic	
A	Yes	2.7274	2.4645	**2.9774**	0.000
B	Yes	**2.6680**	1.6667	2.2514	0.000
C	No	2.3577	2.0152	**2.6210**	0.000
D	No	0.6013	**0.6581**	0.4684	0.000
Cluster	-	7.7567	7.3756	**9.5515**	0.000

Table 3 Overview of the results for the Postgresql top committers

Participant	Left the project?	Monthly mean			ANOVA *p* value
		Visual	Auditive	Kinesthetic	
E	Yes	**0.929891595**	0.530733142	0.638343065	0.000
F	Yes	**0.958569136**	0.421918608	0.631261987	0.000
G	No	0.718432684	0.629946198	0.725690981	**0.085**
H	No	**0.855757104**	0.644466038	0.600263054	0.000
Cluster	-	0.717685413	0.617197033	**0.752834067**	0.000

We observed that Developers B, D, E, F, and H did not have a higher value for the Kinesthetic RS. This contradicts our initial hypothesis that top committers are more Kinesthetic than Visual and Auditory. Moreover, this is also the PRS of the general population, i.e., the PRS of developers clusters for both OSS projects is Kinesthetic (see Cluster row in Tables 2 and 3).

With respect to the second issue defined in the Hypothesis formulation subsection, we found out that there are four visual, two kinesthetic, and one auditory top committer. Looking at their profiles, we realized that most of them are quite concerned with following procedures and documenting information, contradicting our initial stereotype of a hardcore OSS developer.

Moreover, the other developers being kinesthetic on average, leads us to believe that most people that post in the list are indeed involved with practical activities in the project. In this case, this contradicts with our initial belief that many posts were by newbies or people that were simply curious—wanted to hear—about the project.

Even where there is dominance of the Kinesthetic RS, the results show that OSS developers also have significant visual and auditory RS. This may indicate an opportunity to introduce better visualization tools and better support for cooperative work, increasing direct developer interaction, in OSS development.

According to the top committers' profiles, included in the websites of the projects Apache [62] and Postgresql [63], we found out that Developer B had a strong involvement with the project architecture and the work to hybridize Apache. This seems to support his/her Visual PRS (see Table 2 and Fig. 4).

Developer D—the most singular subject among the top committers—has an Auditory PRS and also a strong Visual RS. His/her profile indicates that he/she contributes heavily with the project documentation and his/her predominant working language is XML. This possibly matches the mined profile, as one would expect strong listening and reading capabilities from people involved in OSS documentation.

These insights are quite aligned with the results presented by Rigby and Hassan [15]. This paper reports that the measures collected for Developer D were the least associated with the other subjects in the study. Our study, however, went further and indicated a classification that directly matched the subject profile and project role.

Regarding the Postgresql top committers, the first thing that catches the eyes (see Fig. 5) is that three of them are highly visual. Moreover, the visual PRS is high

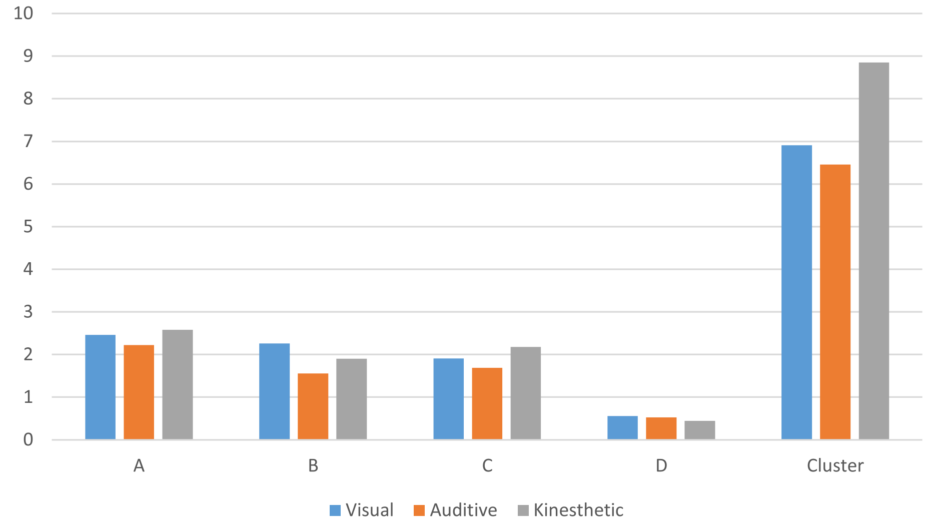

Fig. 4 Apache results

even for Developer G and the project cluster itself. Top committers E, F, and G are highly involved with both documentation and implementation. Top committer G, the only one who is not classified in any category, *p* value 0.085, also works on performance testing and tuning, which may be related to his/her relatively high kinesthetic score. He/She also works with user groups and on providing general direction for the project advocacy, which may be related to his/her relatively high auditory score. Top committer H, by far the most active top committer of them all, is visual but also has a high auditory score, even higher than his/her kinesthetic score. His/her scores may be explained by the fact that he/she is not only highly involved with development, but also does training and maintains the project FAQ and TODO list.

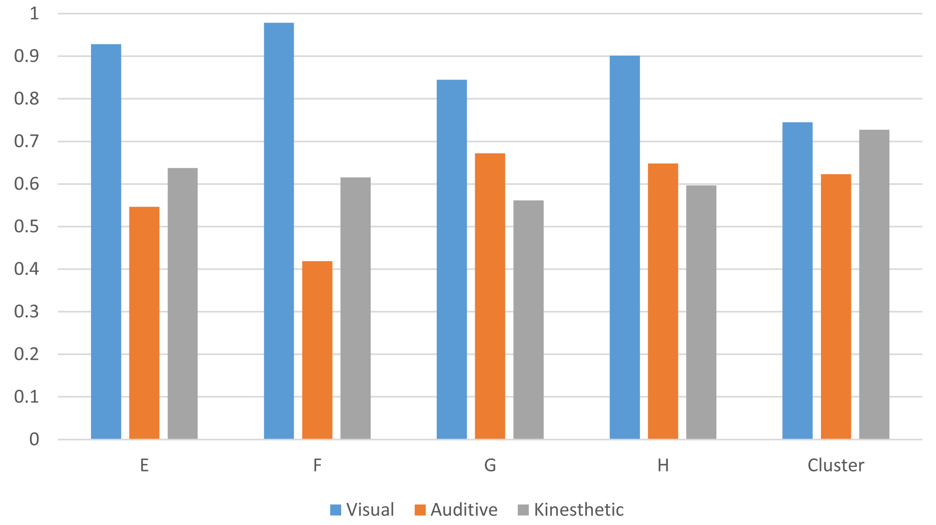

Fig. 5 Postgresql results

Threats to validity

In spite of the fact that Apache and Postgresql are mature, real world, and large projects, and our results seem to be quite consistent with the obtained top committer profiles, the PRS measures still need further investigation to assure external validity.

The next section will focus on the industrial setting. The completely different setup and higher control over the study environment will help to increase the generalization power of the results.

We obtained the top committer profiles through the project sites. Better analysis would be possible with more extensive information. Gathering more profiling data would help us improve our analysis. Aiming at this, we developed a questionnaire to characterize and assess the PRS of software engineers [64]. Moreover, it was performed; a survey with 209 software engineers, validating the effectiveness of the questionnaire and revealing a great diversity of PRSs in the population studied. This population served to calculate the IRT (Item Response Theory) scores of industry programmers, who answered the questionnaire in the third experiment (see next section). This questionnaire and the survey application results that refined it were published in [64].

We contacted the top committers by email and asked them to fill it out. Unfortunately, they could not find the time to fill it out.

Third experiment

Aiming to increase the generalization power of the results, and due to the participants of previous experiments not responding to our questionnaire (see Threats to Validity in section 6.1), we conducted a third experiment in the industry (closed-source projects), which was realized with the same circumstances of the previous.

Goal definition

The main goal of this new study is to evaluate if industry programmers have a PRS. This goal is formalized using the GQM goal template proposed by Basili and Weiss [57] and presented by Solingen and Berghout [58]: *Analyze* programmers of closed-source projects *with the purpose of* evaluation *with respect to* NLP context-specific Preferred Representational Systems *from the point of view of* software engineering researchers *in the context of* development mailing lists of projects.

Planning

The experiment targets developers of closed-source projects. As part of the experiment, it is necessary to replicate, for industry programmers, the experiment with OSS programmers.

Hypothesis formulation The issue we are trying to explore is: We are interested in verifying whether industry developers will have a PRS.

Formally, we will confirm the same hypothesis of the experiment performed with developers OSS:

Null hypothesis H_0: Industry programmers have the same monthly average of the scores for the three representational systems.

H_0^{PRS}: μ(Visual final weight) = μ(Auditory final weight) = μ(Kinesthetic final weight)

Alternative hypothesis H_1: At least one of the means is different from the others.

Participant and artifact selection The choice of developers was for convenience. The authors of this article managed to release five programmers from a company of which they are consultants. The company produces and distributes beverages and soft drinks for two provinces in Brazil.

For legal reasons, we will not use the names of the participants in this study. Letters are used to identify each developer. Table 4 lists these programmers along with two measures of experience in software maintenance.

The execution was non-intrusive because the data were taken directly from the mailing lists of programmers; not even the developer knew that the emails would be posted and analyzed one day.

For each programmer, NEUROMINER was performed to examine the body of the email posted by them on the mailing list of a software project developed by the company. This sample involves 4604 messages posted by five programmers between 2008 and 2010. Finally, each developer was also individually interviewed about his/her job profile in the company.

Results

Table 9 (see Appendix 1) summarizes the results obtained by NEUROMINER on the assumption formalized. The Total column shows the number of months (data points for each participant), of days, and of emails exploited. For each representational system, the final weight is displayed based on all predicates sensory found, as well as monthly averages of these weights. The column ANOVA depicts the *p* values calculated for the test of the null hypothesis.

Aiming to facilitate the visualization of the results, we create Table 5 based on Table 9 in Appendix 1, containing only primary information (participants, monthly mean for each PRS, and ANOVA *p* value) to analyze the results.

For statistical tests, we established a significance level (α) of 0.05. Table 5 shows that the null hypothesis is rejected for 4 developers, who obtained a *p* value of 0.000. The only developer that was not rated was I, with *p* value of 0.105, higher than 0.05.

Table 4 Experience of programmers available

Programmer	Year of experience in maintenance	Number of systems already maintained
I	3	6
J	2	3
L	3	5
M	2	2
N	3	6

Table 5 Overview of the results for the industry developers

Participant	Monthly mean			ANOVA *p* value
	Visual	Auditive	Kinesthetic	
I	0.790360072	0.783063282	0.836974247	**0.105**
J	0.476327848	0.539162195	**0.926387548**	0.000
L	**0.759316714**	0.444731946	0.696729164	0.000
M	**0.976824693**	0.622630894	0.561374571	0.000
N	0.642536173	0.769016124	**0.944168916**	0.000

In summary, the results for developers J, L, M, and N are significantly smaller than 0.05, allowing strong rejection of the null hypothesis and confirmation that they have a PRS (highlighted in bold).

Analysis and interpretation

Regarding profiles, as shown in the experiment performed in the OSS, there was an encouraging result.

In this case, we had a bonus, the questionnaire answered by participants (it discussed in the Threats to validity subsection under the First and second experiments section). This questionnaire follows an approach similar to the VARK model, a questionnaire developed and used at Lincoln University to identify the preferences of students for particular modes of information representation [29]. However, our questionnaire is strongly contextualized to Software Engineering.

Once the analysis of emails mining with NEUROMINER has been done, the next step was to analyze the developers according to their responses to the Neurolinguistic questionnaire.

Table 6 shows a comparative of the generated classifications by NEUROMINER vs. the generated classifications by questionnaire. Whereas the classifications include the order of preference, as well as are made by completely different approaches, the results show good consistency between both approaches.

Addition to the use of NEUROMINER and questionnaire, we interviewed all programmers.

All 5 programmers work directly and deeply with the code; however, in his interview, the programmer L, classified as visual, was the only one who said he routinely used Visual diagrams and Entity relationship [65], creating, reviewing, and optimizing system models. Programmer M reported no constant activity documentation or use of diagrams. Thus, according to his VAK classification, we

Table 6 Neurominer's ratings vs. questionnaire's ratings

Participant	Questionnaire	Neurominer
I	KAV	Not classified
J	KAV	KAV
L	KVA	VKA
M	VAK	VAK
N	KAV	KAV

recommended the partner company to test his potential in tasks that require the use of visual artifacts, as well as surveys and interviews with validation requirements. His second preference, auditive, may indicate affinity in listening and better capturing user needs.

Threats to validity

The difficulty in getting cooperation from industry in the release of its programmers resulted in a small sample size. It is necessary to replicate this experiment with the largest possible number of programmers.

Related works

Regarding NLP, there are some scientific articles showing evidences of its assertions. In addition, there are several publications about preferences for some specific representational systems in the cognitive and learning processes, even in computing [66].

The basis for models and techniques presented by NLP can be found in psychological studies that involve the so-called "chameleon effect", which concerns nonmatching and matching stimuli to the empathy increase in communication. Van Baaren et al. [67] did an experiment at a restaurant in the south of Netherlands in which half of the studied waitresses used the "chameleon effect" to serve customers. Results showed that the average value of the tips almost doubled for the waitresses who used matching language and behavior. Bailenson and Yee [68] analyzed subjects who interacted with artificial intelligence-based software—an agent which simulates a subject giving an explanation. The agent that imitated subject's movements was more convincing, receiving more positive evaluations. It was the first virtual reality study that showed the effects of a nonverbal automatic imitator in order to gain empathy.

Turan and Stemberger [14] tested the NLP hypothesis about matching processes which enhance empathy in communication. The relation between matching and empathy increase were significant. Education was also related to the empathy increase; however, even when it was controlled, the relation between matching and empathy remained significant.

Paolo et al. [66], presupposing some students' preferences for the kinesthetic processing in certain contexts, developed and tested a set of kinesthetic activities for a distributed systems course, with graduation and post-graduation students. The article presents detailed descriptions of the exercises and discusses the factors that contributed to their success and failure.

Fleming presented a questionnaire developed and used at Lincoln University to identify the preferences of students for particular modes of information representation [29]. Named the VARK model, the questionnaire is now the basis of a commercial service for educational planning (http://www.vark-learn.com/english/page.asp?p=questionnaire). The acronym originates from questionnaire classification of the learning styles: "V" is for visual learners, "A" is for auditory learners, "R" is for reader/writer learners (people that best learn through seeing printed words), and "K" is for tactile/kinesthetic learners.

The VARK classification differs from the NLP classic classification because it includes the readers-writers category on top of the usual visual, kinesthetic, and aural categories. According to Fleming, results show that students with preferences for R and V information use their eyes to “take in the world” but they have preferences within that sensory mode; some like text, and others like diagrammatic or iconic material—information that is symbolically displayed [29].

Another point raised by the VARK data is that the same subject may have different profiles in different areas (martial arts, music, languages, etc.) for different time periods, i.e., a subject may be Visual (V) to learn martial arts for a period of time and become Kinesthetic (K) after that.

These evidences support some NLP techniques and establish an empirical basis for further studies.

Considering text mining in Software Engineering, independent from the database, linguistic analyses have been used to comprehend the development of OSS softwares. Witte at al [49]. considered the semantic importance of the documents written in natural language in the process of maintenance and reengineering. The result of the research consisted of creating a text mining system capable of filling software ontology with information extracted from these documents.

Other works have already considered email-specific analysis to study OSS development process [51, 69]. Pattison et al. [70] studied the relation between the several software entities mentioned in emails and the number of times these entities are included in the changes made.

Three works are closest to the research presented here. In the first, Scialdone et al. [44] used emails to evaluate the social presence in maintenance groups of OSS projects. Social presence theory classifies different communication media along a one-dimensional continuum of social presence, where the degree of social presence is equated to the degree of awareness of the other person in a communication interaction. According to the social presence theory, communication is effective if the communication medium has the appropriate social presence required for the level of interpersonal involvement required for a task. On a continuum of social presence, the face-to-face medium is considered to have the most social presence, whereas written, text-based communication, the least. It is assumed in the social presence theory that in any interaction involving two parties, both parties are concerned both with acting out certain roles and with developing or maintaining some sort of personal relationship [71, 72].

Core and peripheral members were compared, and the results showed that respect behavior to another one’s autonomy may contribute to the survival of the group and continuity of the project. The work does not raise alternatives to social presence or solutions to increase empathy. It is based solely on psychological and social measures. It establishes no relation between these aspects and software engineering roles and profiles.

The second work is Rigby and Hassan’s study [15], which analyzed the content of Apache discussion list to find the developer’s personality and general emotional content. Like ours, this work uses a LIWC tool (Linguistic Inquiry and Word Count) [73] to help ratings. However, the work uses a general-purpose psychological analysis tool. It was neither developed to explore emails nor to preprocess text mining and score terms.

Lastly, in the context of Collaborative Systems, Santos et al. [74, 75] developed a collaborative messenger library (NeuroMessenger) that uses neurolinguistics, psychometry, and text mining to promote empathy among interlocutors, from the PRS identification and suggestion of textual matching. After the experimental evaluation, the higher performance with the use of NeuroMessenger, in favor of empathy, was noticeable. In addition, the use of the same pattern of text between interlocutors, in Collaborative Systems, increased the empathy between them.

Previously, we presented initial results for the use of neurolinguistic ratings by mining development discussion lists [76–79]. This works motivated and guided the need for extended studies and details about innovations and technologies involved, which are now presented in this article.

Conclusion and future work

We presented a text Neurolinguistic mining tool that is capable of extracting sensory-based words from software mailing lists. The system is novel in four important aspects: (1) it automates parts of NLP practices; (2) it combines a SE taxonomy with sensory-based words; (3) it adapts traditional text mining process to NLP practices; and (4) it uses specific Text Mining Data Mart in a software engineering data warehouse. The approach itself is novel in its use of NLP concepts in the software engineering area.

This work is part of a family of experiments to detect and validate PRS. Previous studies showed that developers have different PRS using this method and other approaches [76, 78, 79]. In this paper, for example, we combine two approaches: survey and text mining. These works are the first steps on a promising road toward understanding latent traits of software engineers through the use of psychometrics techniques.

The results are encouraging. For OSS setting, in spite of being contrary to our expectation, the PRS scores clearly differentiate the top committers from the general population (i.e., clusters) of the projects, according to the monthly means in bold (PRS score), listed in Tables 2 and 3. In the industry environment, the developers also have a PRS. Moreover, the scores are aligned with the developers' profiles, indicating that they indeed can be used to profile people to software engineering tasks and, possibly, better communication. It is worth noting that the classifications presented in this work are not fixed, i.e., they initially represent only the greater use of one or other system within the context analyzed.

Thus, in specific contexts, a particular sensory system may take dominance (for example, (a) being primarily aware of external kinesthetic representations—bodily movements and sensations—while training and (b) concentrating preferentially on auditory comparisons while analyzing client requirements); representational system preferences thus tend to be a contextual artifact in that when an individual considers specific contexts, his/her language can reflect how he/she processes the information relating to the process of considering that context. In certain cases, a person may find himself/herself with certain rigid representations and strategies which preclude behavioral choice. In such a case, one representational system may predominate and be important for enhancing empathy.

Our future work will address three key issues: (1) examine the empathy of exchanged messages to assess communication success over PRS alignment; (2) better profile PRS scores with usage of software engineering artifacts and the roles that a person plays in a project; and (3) devise new ways to measure PRS. Thus, these next steps will make the results more accurate, conclusive, or ever.

Appendix

Summary of results

Table 7 Apache top committers results

Participant	Left the project?	Totals (1996–2005)			Visual		Auditive		Kinesthetic		ANOVA *p* value
		Months	Days	Emails	Final weight	Monthly mean	Final weight	Monthly mean	Final weight	Monthly mean	
A	Yes	53	773	4357	2.458847752	2.7274	2.222439202	2.4645	2.579237886	**2.9774**	0.000
B	Yes	33	320	1082	2.258647848	**2.6680**	1.557743226	1.6667	1.900853069	2.2514	0.000
C	No	72	1213	4279	1.904784203	2.3577	1.684352265	2.0152	2.17853237	**2.6210**	0.000
D	No	42	366	644	0.557085631	0.6013	0.526795847	**0.6581**	0.441884768	0.4684	0.000
Cluster	-	36	1091	25121	6.906216384	7.7567	6.456228874	7.3756	8.849529521	**9.5515**	0.000

Table 8 Postgresql top committers results

Participant	Left the project?	Totals (1997–2006)			Visual		Auditive		Kinesthetic		ANOVA *p* value
		Months	Days	Emails	Final weight	Monthly mean	Final weight	Monthly mean	Final weight	Monthly mean	
E	Yes	62	899	2854	0.928310296	**0.929891595**	0.546308965	0.530733142	0.637687699	0.638343065	0.000
F	Yes	53	478	1284	0.97883767	**0.958569136**	0.419031696	0.421918608	0.615530516	0.631261987	0.000
G	No	55	536	1176	0.845112965	0.718432684	0.672360338	0.629946198	0.561439549	0.725690981	**0.085**
H	No	121	2728	17712	0.901184217	**0.855757104**	0.648274323	0.644466038	0.596461266	0.600263054	0.000
Cluster	-	36	731	34133	0.745095331	0.717685413	0.623314426	0.617197033	0.727219871	**0.752834067**	0.000

Table 9 Results for the industry developers

Participant	Totals (2008 – 2010)			Visual		Auditive		Kinesthetic		ANOVA *p* value
	Months	Days	Emails	Final weight	Monthly mean	Final weight	Monthly mean	Final weight	Monthly mean	
I	30	357	674	0.750382512	0.790360072	0.737687699	0.783063282	0.898610331	0.836974247	**0.105**
J	20	387	442	0.498330721	0.476327848	0.546430665	0.539162195	0.986144411	**0.926387548**	0.000
L	29	523	1342	0.751891345	**0.759316714**	0.451976343	0.444731946	0.697414428	0.696729164	0.000
M	18	374	1192	0.926541967	**0.976824693**	0.634382161	0.622630894	0.576707234	0.561374571	0.000
N	32	621	954	0.644618737	0.642536173	0.774446612	0.769016124	0.962536713	**0.944168916**	0.000

Abbreviations

ANOVA: Analysis of variance; DF: Daily frequency; GQM: Goal-question-metric; IRT: Item response theory; LIWC: Linguistic inquiry and word count; MuNPEx: Multi-lingual noun phrase extractor; NLP: Neuro-linguistic programming; NP: Noun phrase; OLAP: Online analytical processing; OSS: Open-source software; POS: Parts of speech; PRS: Preferred representational system; RS: Representational system; TF: Term frequency; TM: Text mining

Authors' contributions

All the results presented here were evidenced and analyzed by the authors, as well as by the participant company, with regard to the industrial experiment. In this paper, our main contribution was to show, in deep, the family of the performed experiments for the identification of industrial and OSS developers' context-specific preferred representational systems. The authors read and approved the final manuscript.

Authors' information

Methanias Colaço Júnior holds a PhD in software engineering business intelligence from the Federal University of Bahia. He also holds a MSc in computer science from the Federal University of Campina Grande (UFCG), and a Bachelor in computer science in Tiradentes University (UNIT-1994). He started his professional activities in 1993 and joined the Tiradentes University (UNIT) and Federal University of Sergipe as a professor in 1996. At the UNIT, he headed Business Intelligence research and projects from 1998 to 2008. He also served as a leader of the Decision Support Systems Group at the Sergipe Bank (BANESE) from 1998 to 2009. He is a consultant in several technology fields and to several enterprises in Brazil, with special emphasis in Data Analytics, Data Science, Data Mining, Business Intelligence, Experimentation, Business Process Management (BPM), Strategic Management, Data Warehouse (DW), and Software Engineering. Colaço is also a professor in the Computer Science and Business Administration Department at Federal University of Sergipe, Brazil, where he is part of two academic research programs: (i) Postgraduate Program in Computer Science—PROCC; (ii) Competitive Intelligence Research and Practice Group—NUPIC. He is a lecturer in seminars and enterprise courses about Experimental Software Engineering, Data Science, Business Intelligence, Data Mining, NLP, Neurolinguistics, Customer Relationship Management, Balanced ScoreCard, Big Data Analytics, Strategic Management, BPM, Item Response Theory (IRT), and Decision Support Systems.

Breno Santana Santos holds a Master's degree in Computer Science from Federal University of Sergipe (UFS-2018) and a Bachelor's degree in Information Systems from Federal University of Sergipe (UFS-2015). He is part of the academic research program at Federal University of Sergipe, the Competitive Intelligence Research and Practice Group (NUPIC). His practical experience includes the following areas: Data Mining, Text Mining, NLP, Analysis of Psychological Characteristics of Individuals, and Approaches to gain Empathy through Text Mining Techniques. Currently, he is a PhD student in a Postgraduate Program in Electrical and Computer Engineering at Federal University of Rio Grande do Norte and is working on research related to Data Science, NLP, and Complex Network Analysis on education data.

Manoel Mendonça is a professor of computer science at the Federal University of Bahia (UFBA) and Special Projects Manager at SENAI-BA. He holds a PhD in computer science from the University of Maryland at College Park (UMCP), a MSc in computer engineering from UNICAMP (Brazil), and a Bachelor's degree in electrical engineering from UFBA. From 1994 to 1997, he was a visiting scientist and was awarded a doctoral fellowship from IBM Toronto Laboratory's Center for Advanced Studies. From 1997 to 2000, he worked as a Faculty Research Associate at UMCP and as a scientist at the Fraunhofer Center Maryland. He joined Salvador University (Brazil) as a professor in 2000. There he headed the university's Computing Research Center and helped to create the first computer science master and doctoral programs in his home state of Bahia. From 2008 to 2009, he was the president of the Special Commission for Software Engineering of the Brazilian Computer Society (SBC). He joined UFBA in 2009 and has headed the university´s software engineering lab since then. In 2012, he founded the Fraunhofer Center for Software and Systems Engineering at UFBA. He has worked as its director since then. During 2015 and 2016, he took a leave of absence from UFBA to work as the Secretary of Science, Technology, and Innovation of the State of Bahia. Dr Mendonça has published over 130 technical papers. His main research interests are in software engineering and information visualization. He is a member of SBC and a senior member of IEEE and ACM.

Daniela Corumba holds a Bachelor of Computer Science degree from the Federal University of Sergipe (UFS) and, for the past 6.5 years, has been working at different companies delivering solutions that combine her technical background with Supply Chain Management. She has experienced working in different countries such as India and South Africa. Her work experience started as a software engineer at Tata Consultancy Services in India, where she worked developing an activity management tool created in-house and used across multiple clients around the world. In 2011, she had her first experience in Supply Chain Management, when she worked at Vale in South Africa in the role of Strategic Procurement Analysis and developed a project to make Vale Logistics Network in Africa more efficient. After that, she worked as a Supply Chain Consultant at Manhattan partner in Brazil. During this experience, she developed a project for Puma Warehouse and won the XIII Abralog Prize in the category of Information Technology and Automation. In 2015, she worked as an IT Logistics Supervisor at Columbia (3PL). Currently, she works as a Supply Chain Designer at LLamasoft and delivers strategic solutions to Supply Chain businesses. Among her main career interests are Logistics Systems, Ecommerce, Operations, Logistics Operations, and Processes.

Mario André de F. Farias is a professor at the Federal Institute of Sergipe, Brazil. He has been teaching software engineering at a graduate level, is a chair of the Computation Regional School of Bahia-Alagoas-Sergipe (ERBASE), and is a member of the evaluation committee of scientific papers of CONNEPI. He received his master's degree in the Computational Modeling of Knowledge in 2011, at the Federal University of Alagoas, and a Specialization Analysis of Software from UCAM-Rio de Janeiro-BR. He is a Doctor of Computer Science from the Federal University of Bahia, Brazil. He is working within a context of Technical Debt (TD) and Empirical Software Engineering. His research interests include technical debt, mining software repositories, and empirical software engineering. He has worked in the software industry for 15 years in a variety of roles, such as Software Developer, System Analyst, and Project Manager.

Funding

Not applicable

Availability of data and materials

Not applicable

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Competing interests

The authors declare no competing interests.

Author details

[1]Software Engineering Laboratory (LES), Federal University of Bahia (UFBA), Salvador, Bahia, Brazil. [2]Postgraduate Program in Computer Science (PROCC), Federal University of Sergipe (UFS), São Cristóvão, Sergipe, Brazil. [3]Competitive Intelligence Research and Practice Group (NUPIC), Information Systems Department (DSI), Federal University of Sergipe (UFS), Itabaiana, Sergipe, Brazil.

Received: 8 September 2016 Accepted: 12 February 2021
Published online: 01 March 2021

References

1. Klemola T, Rilling J (2002) Modeling comprehension processes in software development. In: First IEEE International Conference on Cognitive Informatics. IEEE, Calgary, pp 19–20 August 2002
2. Diehl S (2007) Software visualization: visualizing the structure, behavior and evolution of software. Springer Verlag, New York
3. Hungerford BC, Hevner AR, Collins RW (2004) Reviewing software diagrams: a cognitive study. IEEE Trans Software Eng 30(2):84–95
4. Maldonado JC, Carver J, Shull F, Fabbri SCPF, Doria ES, Martimiano L, Mendonca M, Basili V (2006) Perspective-based reading: a replicated experiment focused on individual reviewer effectiveness. Empir Software Eng Int J 11(1):119–142
5. Travassos G, Shull F, Fredericks M, Basili VR (1999) Detecting defects in object-oriented designs: using reading techniques to increase software quality. In: Proceedings of the 14th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'99). ACM, Denver, pp 1–5 November 1999
6. Dent KA (1983) Cognitive styles: essence and origins: Herman A. Witkin and Donald R. Goodenough. J Am Acad Psychoanal 11:635–636
7. Matthews DB (1991) Learning styles research: implications for increasing students in teacher education programs. J Instr Psychol 18:228–236
8. Peters D, Gareth J, Peters J (2008) Preferred 'learning styles' in students studying sports-related programmes in higher education in the United Kingdom. Stud High Educ 33(2):155–166
9. Bandler R, Grinder J (1979) Frogs into princes: neuro-linguistic programming. Real People Press, Utah
10. Dilts R, Grinder J, Bandler R, DeLozier J (1980) Neuro-linguistic Programming. Meta Publications, California
11. Elich M, Thompson RW, Miller L (1985) Mental imagery as revealed by eye movements and spoken predicates: a test of neurolinguistic programming. Journal of Counseling Psychology 32(4):622–625
12. Einspruch EL, Forman BD (1985) Observations concerning research literature on neuro-linguistic programming. Journal of Counseling Psychology 32(4):589–596
13. Tosey P, Mathison J (2007) Fabulous creatures of HRD: a critical natural history of neuro-linguistic programming. In: 8th International Conference on Human Resource Development Research and Practice across Europe, Oxford Brookes Business School, 26-28 June 2007.
14. Turan B, Stemberger RM (2000) The effectiveness of matching language to enhance perceived empathy. Commun Cogn 33(3-4):287–300
15. Rigby PC, Hassan AE (2007) What can OSS ,mailing lists tell us? A preliminary psychometric text analysis of the Apache developer mailing list. In: Proceedings of the Fourth International Workshop on Mining Software Repositories, IEEE Computer Society, Minneapolis, 20-26 May 2007.
16. Cassidy S (2004) Learning styles: an overview of theories, models and measures. Educ Psychol 24(4):419–444
17. Riding RJ, Cheema I (1991) Cognitive styles: an overview and integration. Educ Psychol 11(3-4):193–215
18. Allport GW (1937) Personality: a psychological interpretation. Holt and Co, NewYork
19. Garity J (1985) Learning styles: basis for creative teaching and learning. Nurse Educ 10(2):12–16
20. Badenoch SN (1986) Personality type, learning style preference, and strategies for delivering training to a select group of store managers. University of Minnesota, Dissertation
21. Hartley J (1998) Learning and studying: a research perspective. Routledge, London
22. Hunt EB, Frost N, Lunneborg C (1973) Individual differences in cognition: a new approach to intelligence. Psychol Learn Motiv 7:87–122
23. Estes WK (1974) Learning theory and intelligence. Am Psychol 29:740–749
24. Underwood BJ (1975) Individual differences as a crucible in theory construction. Am Psychol 30:128–140
25. Lohman DF, Bosma A (2002) Using cognitive measurement models in the assessment of cognitive styles. In: Braun H, Wiley D, Jackson D (eds) The role of constructs in psychological and educational measurement. Erlbaum, New Jersey, pp 127–146
26. Lohman DF, Ippel MJ (1993) Cognitive diagnosis: from statistically based assessment toward theory-based assessment. In: Frederiksen N, Mislevy RJ, Bejar II (orgs) Test theory for a new generation of tests. Lawrence Erlbaum, New Jersey, pp 41–71
27. Riding RJ (1994) Personal styles awareness. Learning and Training Technology, Birmingham

28. Riding RJ (2000) Cognitive style: a review. In: Riding RJ, Rayner SG (eds) International perspectives on individual differences: cognitive styles, vol 1. Ablex, Connecticut, pp 315–346
29. Fleming ND (1995) I'm different; not dumb. Modes of presentation (VARK) in the tertiary classroom. In: Zelmer A (ed) Research and Development in Higher Education, Proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA), vol 18, pp 308-313.
30. Tosey P, Mathison J (2003) Neuro-linguistic programming and learning theory: a response. Curriculum J 14(3):361–378
31. Korzybski A (1994) Science and sanity: an introduction to non-Aristotelian systems and general semantics, 5th edn. Institute of General Semantics, New York
32. Santos BS, Júnior MC, Nunes MAS (2018) Approaches for generating empathy: a systematic mapping. In: Latifi S. (eds) Information Technology - New Generations. Advances in Intelligent Systems and Computing, vol 558. Springer, Cham. doi: https://doi.org/https://doi.org/10.1007/978-3-319-54978-1_89.
33. de Jesus Santos F, de Almeida AL, Santos BS, de Souza CCA, Santos MN (2018) Empathic computer science: a systematic mapping. In: Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems (IHC 2018), ACM, New York, NY, USA, Article 46, 5 pages. doi: https://doi.org/https://doi.org/10.1145/3274192.3274238.
34. Thompson JE, Courtney L, Dickson D (2002) The effect of neurolinguistic programming on organizational and individual performance: a case study. J Eur Ind Train 26(6):292–298
35. Heap M (1988) Neurolinguistic programming - an interim verdict. In: Heap M (ed) Hypnosis: current clinical, experimental and forensic practices. Croom Helm, London, pp 268–280
36. Bostic St.Clair C, Grinder J (2001) Whispering in the wind. J & C Enterprises, California
37. Visa A (2001) Technology of Text Mining. International Workshop on Machine Learning and Data Mining in Pattern Recognition (MLDM 2001). Springer, Leipzig, pp 1–11
38. Fayyad U, Piatetski-Shapiro G, Smyth P (1996) The KDD process for extracting useful knowledge from volumes of data. Commun ACM 39(11):27–34
39. Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620
40. Rijsbergen CJV (1979) Information retrieval, 2nd edn. Butterworths, London
41. Hiemstra D, Jong F (2001) Statistical language models and information retrieval: natural language processing really meets retrieval. Glot Int 5(8):288–294
42. Weiss SM, Indurkhya N, Zhang T, Damerau F (2005) Text mining: predictive methods for analyzing unstructured information. Springer Science & Business Media, New York
43. Cunningham H, Maynard D, Bontcheva K, Tablan V (2002) GATE: an architecture for development of robust HLT applications. In: Proceedings of the 40th annual meeting on association for computational linguistics. University of Pennsylvania, Philadelphia, pp 7–12 July 2002
44. Scialdone MJ, Li N, Heckman R (June 2009) Crowston K (2009) Group maintenance behaviors of core and peripheral members of free/libre open source software teams. In: Boldyreff C, Crowston K, Lundell B, Wasserman AI (eds) IFIP International Conference on Open Source Systems. Springer, Berlin Heidelberg, Skövde, pp 3–6
45. Maziero EG, Pardo TAS, Di Felippo A, Dias-da-Silva BC (2008) A base de dados lexical e a interface web do TeP 2.0: Thesaurus Eletrônico para o Português do Brasil. In: VI Workshop on Information Technology and Human Language. Federal University of Espírito Santo, Vila Velha, pp 26–28 October 2008
46. Maziero EG, Pardo TAS, Di Felippo A, Dias-da-Silva BC (2008) TeP 2.0. http://www.nilc.icmc.usp.br/tep2/index.htm. Accessed 10 Oct 2011.
47. Miller GA (1995) WordNet: a lexical database for the English. Commun ACM 38(11):39–41
48. WordNet Database (2011) Princeton University, Princeton. http://wordnet.princeton.edu/wordnet/download/. Accessed 12 Oct 2011.
49. Witte R, Li Q, Zhang Y, Rilling J (2008) Text mining and software engineering: an integrated source code and document analysis approach. IET Software J 2(1):3–16
50. Wongthongtham P, Chang E, Dillon T, Sommerville I (2009) Development of a software engineering ontology for multisite software development. IEEE Trans Knowledge Data Eng 21(8):1205–1217
51. Bird C, Gourley A, Devanbu P, Gertz M, Swaminathan A (2006) Mining email social networks. In: Proceedings of the 2006 International Workshop on Mining Software Repositories. ACM, Shanghai, pp 22–23 May 2006
52. Semantic Software Lab (2009) Multi-lingual Noun Phrase Extractor (MuNPEx). http://www.semanticsoftware.info/munpex. Accessed 20 Jul 2009.
53. Colaço M Jr (2004) Implementing decision support systems and data warehouses. Axel Books, Rio de Janeiro
54. Colaço M Jr, Mendonça MG, Rodrigues F (2009) Data warehousing in an industrial software development environment. In: 33rd Annual IEEE/NASA Software Engineering Workshop (SEW). IEEE Computer Society, Skövde, pp 13–14 October 2009
55. Colaço M Jr, Mendonça MG, Rodrigues F (2009) Mining software change history in an industrial environment. In: XXIII Brazilian Symposium on Software Engineering. Federal University of Ceará, Fortaleza 2009
56. Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2000) Experimentation in software engineering: an introduction. Kluwer Academic Publishers, New York
57. Basili VR, Weiss DM (1984) A methodology for collecting valid software engineering data. IEEE Trans Software Eng 10(3):728–738
58. van Solingen R, Berghout E (1999) The Goal/Question/Metric method: a practical guide for quality improvement of software development. McGraw-Hill Publishing Company, London
59. The Apache Software Foundation (2011) The Apache software foundation. http://www.apache.org. Accessed 20 Jan 2011.
60. PostgreSQL (2011) PostgreSQL. http://www.postgresql.org. Accessed 15 Jul 2011.
61. IBM Analytics (2011) IBM SPSS. http://goo.gl/eXfcT3. Accessed 5 Dec 2011.
62. Apache HTTP Server Project (2011) Apache contributors. http://httpd.apache.org/contributors/. Accessed 25 Jan 2011.
63. PostgreSQL (2011) PostgreSQL: contributor profile. http://www.postgresql.org/community/contributors/. Accessed 19 Jul 2011.
64. Júnior MC, de Fátima MM, Corumba D, Mendonça M, Santos BS (2015) Do software engineers have preferred representational systems? J Res Pract Inf Technol 47(1):23–46

65. Martin J, Carma M (1985) Diagramming techniques for analysts and programmers. Prentice Hall, Englewood Cliffs
66. Sivilotti PAG, Pike SM (2007) A collection of kinesthetic learning activities for a course on distributed computing: ACM SIGACT news distributed computing column 26. ACM SIGACT News 38(2):56–74
67. Van Baaren RB, Holland RW, Steenaert B, van Knippenberg A (2003) Mimicry for money: behavioral consequences of imitation. J Exp Soc Psychol 39(4):393–398
68. Bailenson JN, Yee N (2005) Digital chameleons: automatic assimilation of nonverbal gestures in immersive virtual environments. Psychol Sci 16(10):814–819
69. Mockus A, Fielding RT, Herbsleb JD (2002) Two case studies of open source software development: Apache and Mozilla. ACM Trans Software Eng Methodol 11(3):309–346
70. Pattison DS, Bird CA, Devanbu PT (2008) Talk and work: a preliminary report. In: Proceedings of the 2008 international working conference on Mining software repositories. Association for Computing Machinery, Leipzig, pp 10–18 May 2008
71. Sallnäs EL, Rassmus-Gröhn K, Sjöström C (2000) Supporting presence in collaborative environments by haptic force feedback. ACM Trans Comput Hum Interact 7(4):461–476
72. Stein DS, Wanstreet CE, Glazer HR, Engle CL, Harris RA, Johnston SM, Simons MR, Trinko LA (2007) Creating shared understanding through chats in a community of inquity. Internet nd High Educ 10(2):103–115
73. Pennebaker JW, Francis ME, Booth RJ (2001) Linguistic inquiry and word count: LIWC. Lawrence Erlbaum Associates, Mahwah.
74. Santos BS, Colaço Júnior M, de Souza JG (2018) A initial experimental evaluation of the NeuroMessenger: a collaborative tool to improve the empathy of text interactions. In: Latifi S. (eds) Information Technology - New Generations. Advances in Intelligent Systems and Computing, vol 738. Springer, Cham. doi: https://doi.org/https://doi.org/10.1007/978-3-319-77028-4_54.
75. Santos BS, Junior MC, Souza JGd (2018) An experimental evaluation of the NeuroMessenger: a collaborative tool to improve the empathy of text interactions. In: 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, 2018. doi: https://doi.org/https://doi.org/10.1109/ISCC.2018.8538442.
76. Colaço M Jr, Mendonça Neto MG, Farias MA, Henrique P (2010) OSS developers context-specific preferred representational systems: an initial neurolinguistic text analysis of the Apache mailing list. In: 7th IEEE Working Conference on Mining Software Repositories. IEEE, Cape Town, pp 2–3 May 2010
77. Colaço M Jr (2010) Identification and validation of the developers context-specific preferred representational systems. Doctoral Dissertation, Federal University of Bahia, Salvador
78. Colaço M Jr, Mendonça Neto MG, Farias MA, Henrique P, Corumba D (2012) A neurolinguistic method for identifying OSS developers' context-specific preferred representational systems. The Seventh International Conference on Software Engineering Advances (ICSEA 2012), Lisbon, pp 18–23 November 2012
79. Colaço M Jr, Farias MA, Maciel I, Henrique P, Mendonça Neto MG (2014) Triangulating experiments in an industrial setting to evaluate preferred representational systems of software developers. In: 2014 Brazilian Symposium on Software Engineering. Federal University of Alagoas, Maceio

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.