A Neurolinguistic Method for Identifying OSS Developers' Context-Specific Preferred Representational Systems

Methanias Colaço Júnior [a,b], Manoel Mendonça, Mario André de F. Farias
[a] Software Engineering Laboratory - LES
UFBA – Federal University of Bahia
Salvador/BA – Brasil
mjrse@hotmail.com, manoel.g.mendonca@gmail.com, mario-fa@infonet.com.br

Paulo Henrique, Daniela Corumba
[b] Competitive Intelligence Research and Practice Group - NUPIC
Information Systems Department - DSI
UFS - Federal University of Sergipe
Itababaiana/SE – Brasil
diretor.ph@gmail.com, danielacorumba@gmail.com

***Abstract* — Open Source Software (OSS) 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 OSS 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 is carried out in an experiment to assess the Preferred Representational System of top developers at Apache server and Postgresql mailing lists. The results showed that the PRS scores of the top-committers clearly differ from the general population of the projects. Qualitative analysis also indicated that the PRS scores obtained are aligned with the top committer's profiles.**

***Keywords:* open source; text mining; neurolinguistic; mental imagery; experimental software engineering.**

I. 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 [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 psychology area [6, 7, 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 a 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 evidences 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). NEUROMINER combines text mining and statistic analysis techniques with NLP sensory-based words in order to classify programmers.

NEUROMINER was used in an experiment which analyzed top committers and subjects of two large-scale OSS projects: Apache Server and Postgresql. The results showed that the measured PRS scores can indeed differentiate top committers from the general population. Qualitative analysis also indicated that the PRS scores obtained are aligned with the top committers profiles.

The rest of this paper is organized as follows. The next section introduces NLP. Section 3 reports text mining definitions used throughout the article. Section 4 describes our approach to LIWC and to mining software development mailing lists. In the Section 5, we detail an experimental validation of our approach. Section 6 discusses related works. Finally, Section 7 closes the paper with a discussion of future research.

II. NEURO-LINGUISTIC PROGRAMMING

A. 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 [16], 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 [17] "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.

In [13], an article written by NLP scientific research group, 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 Fritz Perl's models, 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, 7, 8]. Most authors in the area recognize the following basic systems:

(1) Visual, that involves internal images creation and the use of seen or observed things, including pictures, diagrams, demonstrations, displays, handouts, films, and flip-chart;

(2) Auditory, that involves sounds reminders and information transferred through listening; and

(3) 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 consciousness state of a person. 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. 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 effectivity 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. 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 estimulate 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 OSS 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.

B. Neurolinguistic Criticism

NLP is dismissed as theoretically impossible or implausible, especially in websites where one cannot fully trust.

The literature in academic journals is minimal, and the reference [47] 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 [48], 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 criticising 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 [49] have also argued that the types of study reviewed by Heap are characterised 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’ [48].

Given these concerns, in [13], for example, Tosey and Mathison 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 it shows an association between NLP based-measures and developers’ roles and profiles.

III. 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 [18].

A. Preprocessing

Similar to conventional data mining, text mining consists of phases that are inherent to knowledge discovery process [19]. Classification of knowledge discovery phases may vary for different authors, but most comprises 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 [20], 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 [20].

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.

B. 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 [21].

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 [22]. They are described as follows:

1. 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.

2. Characters conversion to uppercase or lowercase: such procedure enables equal words written with a character in a different format in uppercase or downcase – for example, neuro and Neuro may be interpreted as the same term.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

C. 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 [23]:

1. the more a term appears in the document, the more relevant the term is to the document subject;
2. 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:

1. Binary or Boolean – The values 0 and 1 are used to represent, respectively, the absence or presence of a term in the document.
2. 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, more important the term is to the document in which it occurs. The formula to calculate idf is:

$$idf_i = \log \left(|D| \ / \ |\{d: t_i \ \varepsilon \ d\}| \right)$$

Where $|D|$ represents the total of documents and $|\{d: t_i \ \varepsilon \ d\}|$ 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:

$$(tf\text{-}idf)_{i,j} = tf_{i,j} \ x \ \ idf_i$$

D. 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 [24]. 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.

IV. LIWC FOR NEUROLINGUISTIC

A. Motivation

We identified works that try to pinpoint people’s preferred representational systems, but those researches are only in psychology, and in domains like sports and education [25]. 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 [26, 15]. 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 researches about NLP itself, this paper generates the opportunity to show empirical results of applying one of its principles to our, human-intensive, domain.

B. Neurominer

NEUROMINER combines statistic and text mining techniques with sensory predicates of NLP, aiming to classify programmers's 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;
- 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 [50] [27], and for English [51] [28].

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 I, it uses a NLP dictionary with four basic dimensions composed of sensory-based words or phrases [10, 14].

TABLE I. NEUROLINGUISTIC DIMENSIONS

DIMENSION	EXAMPLE WORD	TAG
Visual	'brilliant'	Mod
Auditory	'dissonant'	Mod
Kinaesthtic	'concrete'	Mod
Concepts	'algorithm'	Head

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 I 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 in [29] and described in [30], 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 [29], and clean messages [15, 31] 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 in [31], and places its relevant information into a data mart [32]. This data mart was designed based on a software engineering data warehousing architecture proposed by us in previous papers [33, 34].

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.

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.

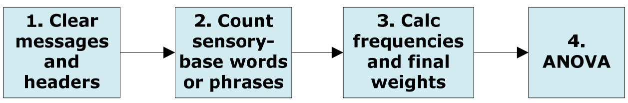

Figure 1. Text mining process chain

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 (1) indicates that *neuro* weight assigned to a word j is the term frequency (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.

$$neuro(j) = (tf(j) + df(j)) \times b \tag{1}$$

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.

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 means are statistically different.

V. Experiment

The rest of this paper describes an experimental validation of our approach. The presented experimental process follows the guidelines by [35]. This section will focus on the experiment definition and planning. The following section will present the obtained experimental results.

A. 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 [36] and presented in [37]:

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

B. Planning

Context selection: The experiment will target OSS projects.

Hypothesis formulation:

The issues we are trying to explore are as follows.

1. We are interested in verifying if OSS top committers have a PRS.

2. 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. This hypothesis is:

Null hypothesis H_0: OSS top committers have the same frequency for the three profiles (Visual, Auditory and Kinesthetic).

Alternative hypothesis H_1: The frequency of OSS Kinesthetic top committers is higher than Visual and Auditory.

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 [53] and Postgresql Projects [54] 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 [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 Section 4.2.2 (email mining). At the end, we had one data point of mined e-mails 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 e-mail 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). The population distribution for each sample is normal.

C. Results

Tables II and III 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.

D. Analysis and Interpretation

For the statistical testing, we established an apriority significance level (α) of 0.05. Tables II and III 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.

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 (see Cluster Row in Tables II and III).

With respect to the first point, we found out that there are four visual, two kinesthetic and one auditory top-committers. 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.

The second point, 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, and counters our initial belief that many posters were by newbies or people that were simply curious – wanted to hear – about the project.

TABLE II. 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	.000
B	Yes	33	320	1082	2.258647848	2.6680	1.557743226	1.6667	1.900853069	2.2514	.000
C	No	72	1213	4279	1.904784203	2.3577	1.684352265	2.0152	2.17853237	2.6210	.000
D	No	42	366	644	0.557085631	0.6013	0.526795847	0.6581	0.441884768	0.4684	.000
Cluster	-	36	1091	25121	6.906216384	7.7567	6.456228874	7.3756	8.849529521	9.5515	.000

TABLE III. 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	.000
F	Yes	53	478	1284	0.97883767	0.958569136	0.419031696	0.421918608	0.615530516	0.631261987	.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	.000
Cluster	-	36	731	34133	0.745095331	0.717685413	0.623314426	0.617197033	0.727219871	0.752834067	.000

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.

Digging a bit deeper into the top committers' profiles [55] and [56], 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 II and Figure 2).

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 in [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 Figure 3), is that three of them are highly visual. Moreover, the visual PRS is high 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 highly involved with development, but also does training and maintains the project FAQ and TODO list.

E. Threats to Validity

In spite of the fact that Apache and Postgresql are a mature, real world, 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.

A new study is being run in an 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. This questionnaire is publicly available at [57].

 ISBN: 978-1-61208-230-1

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

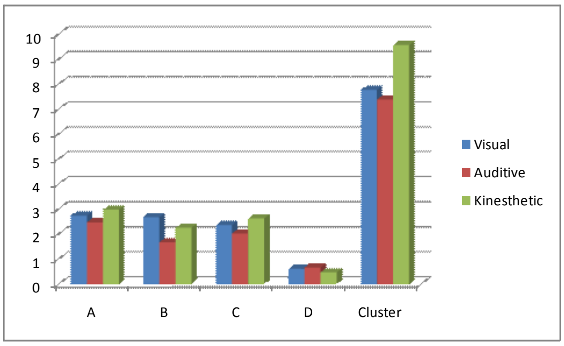

Figure 2 Apache Results

VI. RELATED WORK

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 [38].

The basis for models and techniques presented by NLP can be found in psychological studies that involve the so-called "Chameleon Effect", which concerns non-matching and matching stimuli to the empathy increase in communication. Reference [39] 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. The reference [40] analysed 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 non verbal automatic imitator in order to gain empathy.

Reference [14] tested 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.

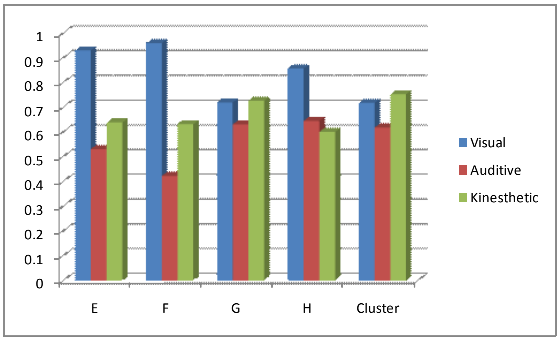

Figure 3 Postgres Results

Paolo et al. [38], 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 for 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 [25]. 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 the 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 [25].

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. [29] 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 [41, 31]. Pattison et al. [42] studied the relation between the several software entities mentioned in emails and the number of times these entities are included in the changes made.

Two works are closest to the research presented here. In the first, Scialdone et al. [26] 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 social presence theory, communication is effective if the communication medium has the appropriate social presence required for the level of interpersonal

 ISBN: 978-1-61208-230-1

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 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 [43, 44].

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 [15], which analysed the content of Apache discussion list to find developer's personality and general emotional content. Like ours, this work uses a LIWC tool (Linguistic Inquiry and Word Count) [45] 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.

In [46], we presented an initial report for the use of neurolinguistic ratings by mining development discussion lists. This work motivated and guided the need for extended studies and details about innovations and technologies involved, which are now presented in this article.

VII. 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.

The results are encouraging. In spite of being contrary to our expectation, the PRS scores clearly differentiate the top-committers from the general population of the projects. Moreover, the scores are aligned with the participant 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, ie, 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. (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 behavioural choice. In such a case, one representational system may predominate and important for enhancing emphaty.

Our future work will address three key issues: (1) examine the empathy of exchanged messages to assess communication success over PRS alignment; and (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.

Acknowledgements

This work is partially (for Methanias and Manoel) supported by INES - National Institute for Software Engineering (www.ines.org.br) funded by CNPq (grant 573964/2008-4).

References

[1] Klemola, T. and Rilling, J. Modeling Comprehension Processes in Software Development. Proceedings: First IEEE International Conference on Cognitive Informatics, pp. 329-336, 2002.

[2] Diehl, S. Software Visualization -Visualizing the Structure, Behavior and Evolution of Software. New York: Springer Verlag, 2007.

[3] Hungerford, B. C., Hevner, A. R and Collins, R. W. Reviewing Software Diagrams: A Cognitive Study. IEEE Transactions on Software Engineering, vol. 30, no. 2, pp. 84-95, 2004.

[4] Maldonado, J. C., , Carver, J., Shull, F., Fabbri, S. C. P. F., Doria, E. S., Martimiano, L., Mendonca, M., and Basili, V. Perspective-based reading: A replicated experiment focused on individual reviewer effectiveness. Empirical Software Engineering An International Journal, 11(1): pp. 119–142, 2006.

[5] Travassos, G., Shull, F., Fredericks, M., Basili, V.R.: 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), pp. 47–56. ACM, New York, 1999.

[6] Dent, K. A. Cognitive Styles: Essence and Origins: Herman A. Witkin and Donald R. Goodenough. Journal of the American Academy of Psychoanalysis, 11: pp. 635-636, 1983.

[7] Matthews, D. B. Learning Styles Research: Implications for Increasing Students in Teacher Education Programs. Journal of Instructional Psychology, 18 pp. 228-236, 1991.

[8] Peters, Derek, Gareth, Jones, and Peters, John. Preferred 'learning styles' in students studying sports-related programme in higher education in the United Kingdom. Studies in Higher Education. Vol 33, pp. 155 – 166, 2008.

[9] Bandler, R. and Grinder, J. Frogs into Princes: Neuro-linguistic Programming, Moab, Utah: Real People Press, 1979.

[10] Dilts, R., Grinder, J., Bandler, R. and DeLozier, J. Neuro-linguistic Programming: Volume 1 California: Meta Publications, 1980.

[11] Elich, M., Thompson, R. W., and Miller, L. Mental imagery as revealed by eye movements and spoken predicates: A test of neurolinguistic programming. Journal of Counseling Psychology, 32(4), pp. 622-625, 1985.

[12] Einspruch, Eric L. and Forman, Bruce D. Observations concerning Research Literature on Neuro-Linguistic Programming. Journal of Counseling Psychology, vol. 32, no. 4, pp. 589-596, 1985.

[13] Tosey, Paul and Mathison, Jane. Fabulous Creatures of HRD: A Critical Natural History Of Neuro-Linguistic Programming. 8th International Conference on Human Resource Development Research and Practice across Europe, Oxford Brookes Business School. Available in http://www.nlpresearch.org/, last access in Dez/2009, 2007.

[14] Turan, Bulent and Stemberger, Ruth M. The effectiveness of matching language to enhance perceived empathy. Communication and Cognition, Vol 33(3-4), pp. 287-300, 2000.

[15] P. Rigby and A. Hassan. What Can OSS Mailing Lists Tell Us? A Preliminary Psychometric Text Analysis of the Apache Developer Mailing List. Proceedings of the Fourth International Workshop on Mining Software Repositories, 2007.

[16] Tosey, P. and Mathison, J. Neuro-linguistic Programming and Learning Theory: a response. The Curriculum Journal Vol. 14 no.3 pp. 361 – 378, 2003.

[17] Korzybski, A. Science and Sanity. The International Non-Aristotelian Library Publishing Company. Available online at http://www.esgs.org/uk/art/sands.htm, accessed 14.01.2009, 1941.

[18] Visa. A. Technology of Text Mining. Proceedings of Machine Learning and Data Mining in Pattern Recognition, Second International Workshop, MLDM 2001, Leipzig, 2001.

[19] Fayyad, Usama; Piatetski-Shapiro, Gregory; Smyth, Padhraic. The KDD Process for Extracting Useful Knowledge from Volumes of Data. In: Communications of the ACM, pp.27-34, Nov.1996

[20] Salton, G., Wong, A., andYang C.S. A Vector Space Model for Automatic Indexing. Communications of the ACM, New York, v.18, 1975.

[21] C. J. V. Rijsbergen. Information Retrieval, 2nd edition. Butterworths, London, 1979.

[22] Hiemstra, D. and Jong, F. Statistical Language Models and Information Retrieval: Natural language processing really meets retrieval, Glot International 5(8), Blackwell Publishers, 2001.

[23] Sholom, M. Weiss et al. Text Mining: Predictive Methods for Analyzing Unstructured Information. Springer, 2005.

[24] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. GATE. In Proceedings of the 40th Meeting ACL, 2002.

[25] Fleming, N. D. 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), HERDSA, Vol 18, pp. 308 – 313, 1995.

[26] Scialdone, Michael J. Group Maintenance Behaviors of Core and Peripherial Members of Free/Libre Open Source Software Teams. In Proceedings of the OSS 2009, pp. 298-309, 2009.

[27] Maziero, E.G. Electronic Thesaurus for the Brazilian Portuguese. VI Workshop on Information Technology and Human Language, pp. 390-392, 2008.

[28] Cognitive Science Laboratory. WordNet - a lexical database for the English language, Princeton University, 2006.

[29] Witte, R., Li, Q., Zhang, Y., and Rilling, J. Text mining and software engineering. IET Softw – Special Section on Natural Language in Software Engineering, 2, (1), pp. 3–16, 2008.

[30] Wongthongtham, P., E. Chang, T. S. Dillon, and I. Sommerville. Development of a software engineering ontology for multi-site software development. IEEE Transactions on Knowledge and Data Engineering, volume 21, Issue 8, pp. 1205-1217, 2009.

[31] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan. Mining email social networks. Proceedings MSR 2006, pp. 137–143, 2006.

[32] Colaço Jr., Methanias. Implementing Decision Support Systems and Data Warehouses. Rio de Janeiro, Brazil: Axel Books, 2004.

[33] Colaço Jr. , Methanias, Mendonça, M. G. and Rodrigues, F. Data Warehousing in an Industrial Software Development Environment. In: 33rd Annual IEEE/NASA Software Engineering Workshop, 2009A.

[34] Colaço Jr., Methanias, Mendonça, M. G. and Rodrigues, F. Mining Software Change History in an Industrial Environment. In: XXIII Brazilian Symposium on Software Engineering, Fortaleza, Brazil, 2009B.

[35] Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell, and A. Wesslén. Experimentation in software engineering: an introduction. Kluwer Academic Publishers, 2000.

[36] V. Basili and D. Weiss, “A Methodology for Collecting Valid Software Engineering Data,” IEEE Transactions on Software Engineering, vol.10(3): pp. 728-738, November 1984.

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

[38] Paolo A. G. Sivilotti and Scott M. Pike. A collection of kinesthetic learning activities for a course on distributed computing. ACM SIGACT news distributed computing column 26. SIGACT News 38(2): pp. 56-74, 2007.

[39] Van Baaren, R. B., Holland, R. W., Steenaerts, B., and van Knippenberg, A. Mimicry for money: Behavioral consequences of imitation. Journal of Experimental Social Psychology, 39, pp. 393-398, 2003.

[40] Bailenson and Yee. Digital chameleons: automatic assimilation of nonverbal gestures in immersive virtual environments. Psychological Science. v16. Pp. 814-819, 2005.

[41] A.Mockus, R. T. Fielding, and J. Herbsleb. Two case studies of open source software development: Apache and Mozilla. ACM Transactions on Software Engineering and Methodology, 11(3):pp. 1–38, 2002.

[42] David S. Pattison , Christian A. Bird and Premkumar T. Devanbu. Talk and work: a preliminary report. Proceedings of the 2008 international working conference on Mining software repositories, May, pp. 10-11, 2008.

[43] Sallnas, E.L., Rassmus-Grohn, K., and Sjostrom, C. Supporting presence in collaborative environments by haptic force feedback. ACM Transactions on Computer-Human Interaction, 7(4), pp. 461–476, 2000.

[44] Stein, D.. Creating shared understanding through chats in a community of inquity. The Internet and Higher Education 10(2), pp. 103-115, 2007.

[45] J. Pennebaker, M. Francis, and R. Booth. Linguistic inquiry and word count: Liwc 2001. Mahwah, NJ: Erlbaum, 2001.

[46] Colaço Jr, Methanias, Mendonça Neto, M. G., Farias, M. A. and Henrique, Paulo. 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, Cape Town, 2010.

[47] Thompson, J. E., Courtney, L., and Dickson, D. The effect of neurolinguistic programming on organizational and individual performance: a case study, Journal of European Industrial Training, vol. 26, no. 6, pp. 292-298, 2002.

[48] Heap, M. "Neurolinguistic programming - an interim verdict," in Hypnosis: current clinical, experimental and forensic practices, M. Heap, ed., Croom Helm, London, pp. 268-280, 1988.

[49] Bostic St.Clair, C. and Grinder, J. Whispering in the Wind J & C Enterprises, Scotts Valley, CA, 2001.

WEB REFERENCES

[50] http://www.nilc.icmc.usp.br/tep2/index.htm accessed in October, 2011.

[51] http://wordnet.princeton.edu/wordnet/download/ accessed in October, 2011.

[52] http://www.ipd.uka.de/~durm/tm/munpex/ accessed in July, 2009.

[53] www.apache.org accessed in January, 2011.

[54] www.postgresql.org accessed in July, 2011.

[55] http://httpd.apache.org/contributors/ accessed in January, 2011.

[56] http://www.postgresql.org/community/contributors/ accessed in July, 2011.

[57] www.neurominer.com/survey accessed in October, 2011.