Do Software Engineers Have Preferred Representational Systems?

Methanias Colaço Júnior
Competitive Intelligence Research and Practice Group – NUPIC and Department of Computing
Federal University of Sergipe (UFS), Itabaiana and São Cristóvão, Sergipe, Brazil
Software Engineering Laboratory, Federal University of Bahia (UFBA)
Salvador, Bahia, Brazil
Email: mjrse@hotmail.com

Maria de Fátima Menezes
Statistics Department, Rede/Redecard S.A., Barueri, São Paulo, Brazil
Competitive Intelligence Research and Practice Group, Federal University of Sergipe (UFS)
Itabaiana, Sergipe, Brazil
Email: airam_yes@hotmail.com

Daniela Corumba
Competitive Intelligence Research and Practice Group, Federal University of Sergipe (UFS)
Itabaiana, Sergipe, Brazil
Email: danielacorumba@gmail.com

Manoel Mendonça
Software Engineering Laboratory, Federal University of Bahia (UFBA)
Salvador, Bahia, Brazil
http://wiki.dcc.ufba.br/LES/ManoelMendonca
Email: manoel.mendonca@ufba.br

Breno Santana Santos
Department of Computing, Federal University of Sergipe (UFS)
São Cristóvão, Sergipe, Brazil
Email: breno1005@hotmail.com

Human beings have different cognitive channels by which they perceive and interpret information. One may use visual, auditory, and kinesthetic channels for that. Software engineering has been striving for years to improve the practice of software comprehension through the use of better cognitive channels. The area has slowly moved towards graphical descriptions of software systems since its origins, when software systems were essentially described by text. However, are there other channels that can be used? Moreover, do different software engineers have different preferred representational systems? Do those channels vary with the task at hand? The literature on cognitive research indicates that the answer for those questions should be yes.

Manuscript received: 4 May 2013
Communicating Editor: Colin Fidge

This paper is a first step towards the study of representational systems preferences in software engineering. It uses a survey combined with Item Response Theory (IRT) data analysis to show that software engineers indeed have Preferred Representational Systems, and points out that those preferences can be identified. These findings can stimulate researches on multimedia aids (video and audio) to help software construction and comprehension, and help scientists and managers to devise better ways to communicate with, and manage, software engineers.

ACM Classifications: *A.1, G.3, D.2, D.m, H.1.2*

Keywords: *Neuro-linguistic, software comprehension, mental imagery, experimental software engineering, cognitive styles, IRT*

1. Introduction

Software comprehension is one of the most important activities in software engineering. Software must be understood in order to be properly maintained, evolved and reused (Maletic and Kagdi, 2008; Hawes *et al*, 2015; Sripada and Reddy, 2015). To gain an understanding of the system's inner workings, developers use resources such as: (1) examples, analogies, and code execution; (2) visual descriptions, diagrams and graphic models of the system; and (3) source code and textual descriptions of the software. These resources are complementary and can be combined. However, is there a Context-Specific Preferred Representational System (PRS) for software engineers? And, is there a preferred order or combination of the representational systems for better supporting the comprehension process?

Historically, software engineering has heavily used structured and unstructured text to represent information. Visual resources like diagrams and non-conventional visualization metaphors are also being increasingly used in software engineering (Diehl, 2007; Moody, 2009). Studies show that the way software engineers process those resources impacts on the success of that processing (Hungerford *et al*, 2004), for both text (Maldonado *et al*, 2006) and diagrams (Travassos *et al*, 1999). However, we do not know of studies that specifically evaluate what types and combination of representational systems are preferred by software engineers, similar to that shown in Kumano *et al* (2011) and Tan *et al* (2014).

This is a broad question in the sense that different people may have different preferences in different contexts. Actually, the concept that there are different paths for cognition – using different representational systems – is well accepted in psychology (Dent, 1983; Matthews, 1991; Peters *et al*, 2008; Lord *et al*, 2015). According to it, internal mental processes such as problem solving, memory, and language consist of visual, aural and kinesthetic representations that are activated 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 person's performance. Moreover, this belief has raised new theories such as neuro-linguistics, which proposes the use of a PRS in specific contexts (Bandle and Grinder, 1979).

In this paper, we present a survey to identify the preferred representational systems of software engineers. It uses a questionnaire based on neuro-linguistic theory to do that. The questionnaire consists of 16 questions that have, each one, an option related to a specific representational system. The questions delve into common software engineering situations and the participants can choose more than one option when answering it, indicating multiple preferences of representational systems.

The questionnaire was made available on the internet (http://www.neurominer.com/survey/) and people from the software engineering community were invited randomly to answer it. A sample of 209 software engineers provided complete valid answers to the questionnaire.

Item Response Theory (IRT) data analysis was used to show that the questions can indeed differentiate the PRS of the participants. After the analysis, the results indicated that software engineers (SE) have different preferred channels, but the order of preference for the channels is very balanced among the participants. Moreover, results showed that software engineering methods and techniques are very much based on visual and kinesthetic resources.

The paper is organized as follows. Sections 2 and 3 introduce, respectively, the psychology concepts of Cognitive Styles and Neuro-linguistics. Section 4 discusses related works. Section 5 details Item Response Theory as an instrument to measure and statistically test psychological properties of individuals. Section 6 presents the IRT-based survey and analyzes and comments on its results. Finally, Section 7 closes the paper with a discussion of future research.

2. 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 (Cassidy, 2004). 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 (Cassidy, 2004) and cognitive styles (Riding and Cheema, 1991). 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 (1937) described cognitive style as a common habit or a notably personal way of solving, thinking, noticing and recalling problems. Garity (1985) 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 (1986), in his study about "personality type, preference of learning style and instructional strategies", claims that 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 (2008) defines "cognitive styles" as the ways that the subjects conduct their cognitive tasks and "learning styles" as the ways that the subjects conduct 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 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.

Based on neurolinguistics, our work considers that the words or expressions used by subjects are related to their preferred ways of representing information in memory. This paper proposes a questionnaire specifically conceived to capture the cognitive preferred representational system (PRS) of software engineers. The PRS is the way a person, in specific contexts, prefers to use to communicate and learn (Dent, 1983; Matthews, 1991; Peters *et al*, 2008).

3. Neuro-Linguistic Programming

Neuro-Linguistic Programming (NLP), created in the 70s, is now a quite popular approach for 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 a model for comprehension and learning (Tosey and Mathison, 2003), few academic works exist on the subject.

NLP was developed by Bandler and Grinder (1979). Bandler, experienced in mathematics and gestalt, studied at Santa Cruz University in the 70s, where he developed collaboration with John Grinder, a linguistics professor. According to Grinder and Bandler, the term NLP derived from studies of patterns or programming created by the interaction among the brain (neuro), the language (linguistics) and the body, which result in efficient and inefficient behaviours. In other words, a subject's brain is comprised of standardized connections among neurological processes, language and behavioural strategies of learning (programming) (Dilts, 1980).

One of the main features of NLP consists of guidance for learning as the key for personal change and development, premising that people are intrinsically creative and capable, acting according to how they understand and represent the world, instead of how the world is. The literature constantly cites Korzybski's statement "the map is not the territory" (Korzybski, 1994), a reference to individual understanding that everyone has mental model, according to his/her experience, beliefs, culture, knowledge and values.

Tosey and Mathison (2007) presented NLP as an epistemological perspective, with scientific principles which are not usually presented in its literature. 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 analyzing the use of a preferred representational system for cognition.

This representational system (or internal representation) is highly dependent on context (i.e. varies with the situation) (Einspruch and Forman, 1985). This way, some people, in specific contexts, may prefer to use one or more basic systems to communicate and learn (Dent, 1983; Matthews, 1991; Peters *et al*, 2008). 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-charts;
2. Aural, that involves sound 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.

Our research deals with the identification of preferred representational systems of software engineers by analyzing such preferences within software development tasks.

4. Related Works

In this section, we discuss related works from two perspectives: NLP scientific evidences and a related questionnaire.

4.1 NLP Related Works

The area of NLP provides only some scientific evidence of its assertions. Despite this, there are several publications about preferences for some specific representational systems in the cognitive and learning processes, even in computing (Sivilotti and Pike, 2007).

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 increase empathy in communication. Van Baaren *et al* (2003) did an experiment at a restaurant in the south of the Netherlands in which half of the studied waitresses used the "Chameleon Effect" to serve customers. Results showed that the average tip value almost doubled for the waitresses who matched the customer's language and behaviour. Bailenson and Yee (2005) analyzed subjects who interacted with artificial intelligence based software – an agent which simulates a subject giving an explanation. The agent that imitated the subject's movements was more convincing, receiving more positive evaluations. This virtual reality study shows that even a non-verbal automatic imitator can gain empathy from a human subject.

Turan and Stemberger (2000) tested the NLP hypothesis that matching processes enhances empathy in communication. The relationship between matching and empathy increase was significant. Politeness was also related to the empathy increase; however, when the politeness effect was controlled, the relation between matching and empathy remained significant.

Sivilotti and Pike (2007), presupposing some students' preferences for the kinesthetic processing in certain contexts, developed and tested a set of kinesthetic activities for a course on (software) distributed systems, with undergraduate and graduate students. The article presents detailed descriptions of the exercises and discusses the factors that contributed to their success and failure.

This evidence supports NLP techniques and establishes an empirical basis for our current and further studies.

4.2 Questionnaire and IRT Data Analysis

Fleming presented a questionnaire developed and used at Lincoln University to identify the preferences of students for particular modes of information representation (Fleming, 1995). According to the author, the questionnaire can help educators to develop strategies to communicate with students. It certainly helps to overcome the predisposition of many educators to treat all students in a similar way. Used by teachers, the questionnaire may also show the mismatches between their and the students preferential representation systems and learning styles. This may motivate them (the teachers) to move from their preferred mode(s) to others in order to reach more students.

Named the VARK model, the questionnaire is now the basis of a commercial service for educational planning (http://vark-learn.com/the-vark-questionnaire/?p=questionnaire). A 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, e.g., a subject may be Visual (V) to learn martial arts for a period of time and become Kinesthetic (K) after that.

Fleming's questionnaire has evolved over the years and it is now in its seventh version. It consists of 16 questions that have, each one, an item related to each learning style covered. According to him, experience suggests that if there are too many questions (25+) some people may experience questionnaire fatigue or boredom (VARK Learn, 2010).

Multiple options can be selected for each question. This is desirable because, in spite of preferences, humans use multiple cognitive channels. Unfortunately, this makes it more difficult to analyze the questionnaire score. Leite *et al* (2010) consider that the score used in VARK is arbitrary, because it is based on standard deviation, creating a dependence on the existence of a reference population to normalize the scores of an individual. Moreover, the approach considers that all questions are equivalent with respect to the measurement of the latent trait.

The questionnaire presented in this paper follows an approach similar to VARK, but it is strongly contextualized to Software Engineering. Contrary to VARK, our questionnaire enforces preference on multiple choice answers, i.e., it requires an ordering on multiple choice answers and does that to capture the individual's preferred strategy, even when a person is multimodal (uses more than one cognitive system).

Our approach considers also that questions are different with respect to the measurement of the latent trait. It uses Item Response Theory (IRT) to estimate item (question) difficulty. This method places item difficulty on the same scale as a person's ability. The scale is developed so that the higher the person's ability is compared to an item's difficulty, the more likely the person is to answer that item correctly. A person at the same point on the scale as an item has a 50 percent chance of answering it correctly. People below the item's value have less than a 50 percent chance and people above the item's value have greater than a 50 percent chance. In this way, any point on the scale describes both an item's difficulty and a person's score. The next section details the IRT concepts used in this paper.

Finally, this work is part of a family of experiments to detect and validate PRS in software engineering. Previous studies used a completely different approach to show that software developers have different PRS (Colaço *et al*, 2009; 2010; 2012). Those studies used text mining to analyze OSS mailing lists. The approaches classified the PRS of participants of two large OSS projects based on the words they used in their public messages. The classifications were then compared with the subject public profile on the project home pages.

5. Item Response Theory

This section explains the Item Response Theory (IRT), the basis of our data analysis. The concepts presented here can be useful to other empirical software engineering researchers that wish to develop questionnaires for widely heterogeneous populations.

Test Theory studies statistic modeling techniques to measure the level of subjective properties of individuals using tests or questionnaires. Classical Test Theory (CTT) focusing on individual scores (total points on tests) and on examined subjects' behaviours (Baker, 2001), makes tests a non-comparable isolated analysis, when applied to a population with different characteristics. For this reason, researchers in the areas of psychology and education sought a model to address this problem.

The Item Response Theory (IRT) arose in the area of psychometrics as an instrument to measure subjective variables and provide statistically the level of certain psychological properties in the individual. In IRT, the focus is the level of the variable of interest in the subject (Singh; 2004; Karim, 2010).

In IRT, the item statistics are independent of the examined group. The scores are independent of the difficulty of the test to describe the subjects' abilities. The model does not require strictly parallel tests to assess the reliability or trustworthiness of individuals. It firstly expresses the item (question) level instead of the test level. For these reasons, it has no variation when compared to different populations, whether in culture, gender, socioeconomic or ethnic groups (Reise *et al*, 2005). It also has advantages in relation to the sample sizes, allowing for valid estimates to the parameters even in relatively small samples.

This approach is widely used in exams that are applied multiple times to heterogeneous populations, such as the Graduate Record Examination (GRE) and the Test of English as a Foreign Language (TOEFL).

Statistical models have their applications and specific properties for each type of data and analysis which they propose. The Item Response Theory has its specificity in measuring variables that are not apparent in the individual, which cannot be known in themselves, but only by their effects. Such variables are measured by other secondary variables that are observable and, somehow, are indicative of the most fundamental, called latent variable, which can be a skill, ability, level of satisfaction, level of proficiency in reading, mental health, or stress level, among others (Reeve and Fayers, 2005). In order to measure a latent variable, usually a test is developed with a series of questions or items, which express some aspect of the subjective variable (Baker, 2001).

The first analysis unit of IRT is the item (question). The main goal consists of estimating the parameters of the items and grasping their metric properties instead of having a standardized test (Reise *et al*, 2005). These parameters measure how much the item (question) contributes de facto to the latent variable measure.

In the model, the scores are not determined by the simple counting of the given answers that refer to the measured latent variable, but by the comparison of different functions of "information content" of each item (question) to different groups. The model uses only the answered items to estimate the latent variable and works even in the presence of missing data. For this reason, it provides parameters for an evaluation of ability levels, making the comparison between subjects in different moments possible.

The estimation of the latent variable is represented by the greek letter theta (θ) for each one of the examined subjects. The scale of the measured variable produced by the item parameters is represented in standard deviations. Baker (2001) considers that, whatever the ability, it can be measured on a scale having a midpoint of zero, a unit of measurement of one, and a range from negative infinity to positive infinity. Since there is a unit of measurement and a zero point, such a scale is referred to as existing at an interval level of measurement. The underlying idea is that if one could physically ascertain the ability of a person, this ruler would be used to tell how much ability a given person has, and the ability of several persons could be compared.

While the theoretical range of ability is from negative infinity to positive infinity, practical considerations usually limit the range of values from -4 to +4 or -3 to +3. The second interval, for example, contains 99.97% of all subjects of a population. The discussions in this paper will deal only with ability values within -4 to +4 range.

The objective of IRT is to mathematically measure a latent characteristic, or "construct", considering the particularities of the item and the probability that the subject selected or preferred a specific item, given the item score (Wu and Adams, 2007). This relation is the basis of IRT and, after being graphically represented, it has an elongated S-shaped curve. This curve is called the Item Characteristic Curve (ICC).

The Item Characteristic Curve is described by a logistic regression function of the probabilities that the subjects endorse an answer related to the latent variable measured, according to the level of this variable in the subject (Baker, 2001). This interpretation is considered valid when the items have dichotomous responses (correct or incorrect, yes or no). To polytomous responses, the ICC represents the amount of ability on the latent variable that the subject needs to have in order to endorse such answers.

The Item Response Theory is divided, mainly, by dimensionality, item shapes and number of parameters of the items to be estimated. Regarding dimensionality, most IRT models claim that there is only a dominant fitness to execute a specific activity, although one knows that the human actions are a result of a series of motivations. These are the unidimensional models, which consider that the measured latent variable is the only one responsible for the performance in the set of items of the test (Baker, 2001). Another assumption closely linked to unidimensionality is the local independence, which states that all items of the test are independent of each other.

In relation to the type of responses, IRT is divided into:

- **Dycomotous:** when responses are correct vs. incorrect, has vs. has not, i.e., the responses have binary values; and
- **Polytomous:** when responses are n-ary. There are two main groups in this case. Models of responses that assume a given order, known as graded responses, and models of nominal responses that do not impose an order for the responses.

In relation to the number of parameters to be estimated, one has (Reeve and Fayers, 2005):

- **One parameter logistic model or Rasch model:** a model that follows the assumption that the probability of success of an item is influenced by its difficulty, i.e., this model has only the difficulty parameter;
- **Two-parameter logistic model:** in this model, the probability of success of an item is influenced by the discrimination, besides the difficulty;
- **Three-parameter logistic model:** it assumes that the probability of success of an item is influenced by the difficulty, discrimination and random success.

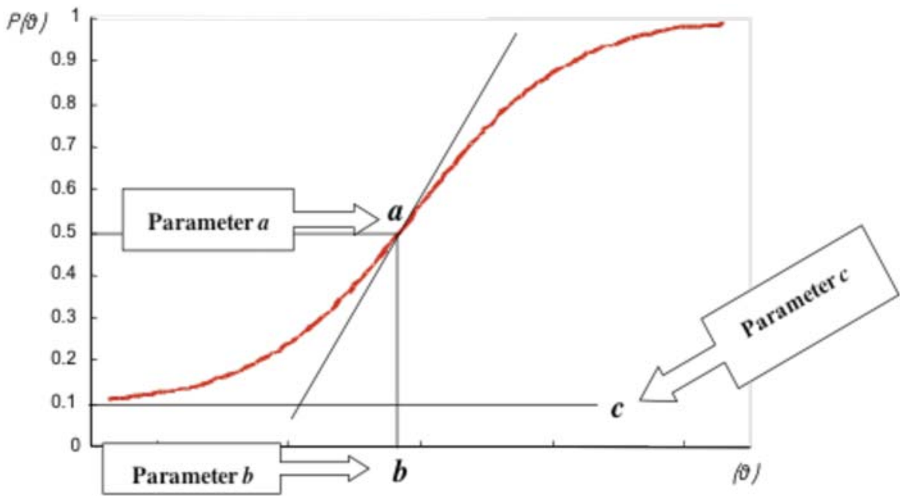


Figure 1: ICC of IRT Model with three parameters

In relation to the interpretation of the parameters generated, we can summarize that the point in the scale θ theta where the probability of success is 50% is called parameter "b" – difficulty parameter. The bigger "b" is, the bigger will be the fitness level required for the examined subject to have the chance of 50% of hitting the item.

Parameter "a" – the discrimination parameter – indicates the slope of the curve at the inflection point (at the same 50% probability). The steeper this parameter is, the more discriminative the item will be. By answering this item a person identifies more strongly to the latent variable that it represents. In items with dycotomous responses, the discrimination refers to the level in which the item properly differentiates the examined subjects in the behaviour the test intends to measure. In these cases, the discrimination expresses how much the expected scores of the item change depending on the level of latent trait.

The parameter "c" refers to choosing at random. This defines the point in which the subject with low proficiency may answer properly to the stimulus. It is defined by the distance from the origin to the point where the curve cuts the ordinate. In the Figure 1 example, the curve indicates that a subject has a 10% chance of guessing the right answer at random.

Our work employs a two-parameter logistic model, as explained in the Results section. The LTM (Latent Trait Model) package of the R Software project (http://www.R-project.org) was used to calculate the parameters of the items and subject scores (Rizopoulos, 2006).

6. Experiment

This paper presents our work as an experimental process. It follows the guidelines set by Wohlin *et al* (2012). This section will focus on the experiment definition and planning. The following section will present the data analysis.

6.1 Goal Definition

Our main goal is to identify what types of representational systems are the preferred by software engineers. This goal is formalized using the GQM Goal template proposed by Basili and Weiss (1984) as presented by Van Solingen and Berghout (1999):

- **Analyze** used representational systems
- **with the purpose of** characterizing them
- **with respect to** NLP context-specific PRS
- **from the point of view of** the software engineers
- **in the context of** software development and maintenance tasks

6.2 Planning

Context selection: Some of the most consolidated methodologies of representation of software (called productive and safe) are based on an exhaustive use of diagrams (Moody, 2009) (e.g., UML diagrams), which may benefit learners with a visual profile (V), due to the use of visual symbols inherent to this way of representation. Since it is not the only way of representing and comprehending software, we ask "Do software engineers have other Preferred Representational Systems?" To answer this question, we have created a questionnaire based on neuro-linguistic theory, raising specific questions to verify what these preferred representational systems are. This way, the experiment focuses on analyzing software engineering professionals. The participants answered the questionnaire and submitted his/her responses online. The gathered data were then analyzed using IRT to determine the PRS of each participant.

Hypothesis formulation: Software engineers exercise multimodality, i.e., they frequently stimulate and use the visual, aural and kinesthetic systems; however, in specific contexts, they may have preferences in relation to them, indicating the different strategies of comprehension and prioritization of artifacts.

Hence, the hypothesis we are trying to confirm is:

Null hypothesis H_0: Software Engineers have the three systems balanced, showing similar levels in the scores generated by the IRT model, i.e., there are no significant differences among the first, second and third classifications of the system.

Alternative hypothesis H_1: Software Engineers preferably use a system which indicates the best strategy of comprehension and prioritization of artifacts, i.e., there are significant differences among the first, second and third classifications of the system.

H_0: $\theta_1 = \theta_2 = \theta_3$

H_1: $\theta_1 > \theta_2 > \theta_3$, where

θ_1, θ_2 and θ_3 are the scores related to the first, second and third classifications, respectively.

Participant selection: The first participants were invited randomly, using the authors' contacts in industry and academia, among experienced professionals, graduate students and researchers of software engineering. Besides those, several professionals from the OSS community were invited and filled in the questionnaire. The questionnaire is available on the internet so that more professionals could voluntarily contribute to it.

Instrumentation: The hypothesis was tested through a questionnaire available on the internet. It was built based on the adaptation from VARK, featuring everyday situations of software engineers in which they can use the three representational systems: Visual, Aural and Kinesthetic.

The questionnaire can be accessed in <www.neurominer.com/survey> and consists of two parts:

1. The first part comprises 16 questions related to the usual tasks of Software Engineers (see Table 1). Each question consists of three alternatives, each referring to a representational system that can be selected by the subject to tackle the task at hand. The questions are answered assigning numbers to the alternatives according to the participant's preference. The first choice is numerated as one (highest preference), the second is two and the third is three. The respondent does not need to select all choices. He/she can choose either one, two or three alternatives for a valid answer. The participant needs to answer at least 12 questions (75%) to submit a valid questionnaire. In order to avoid undesirable effects of memory and tendencies to choose the same alternative to different questions, we used a table of random numbers to select the position of the questions in the questionnaire (Hill and Hill, 1998). Following this approach, in each new entry to the website, the questionnaire creates a new random order, for both the alternatives and the questions.
2. The second part consists of characterizing the participant: position (manager, analyst or programmer); gender (male, female); experience in maintenance (1–3 years, 3–5 years, 5–10 years, more than 10 years); experience expressed as the number of maintained systems (1–5, 6–10, 11–20, more than 20); and previous utilization of a software visualization tool (yes or no). The participant may also provide information such as country and state, besides indicating his/her email address in case he/she desires to receive the results of the questionnaire.

	Questions	*Visual*	*Auditory*	*Kinesthetic*
1	When you are introduced to a new system, do you prefer …	seeing a UML diagram?	talking to yourself and to other people?	using it first to feel how it could evolve?
2	When you have a maintenance task to execute, do you prefer …	imagining different perspectives?	listening and talking about options to solve the task?	exposing your ideas no matter what?
3	When you implement a class, which is tested successfully, do you prefer …	designing a clear diagram for everybody to see?	telling everyone in the team the news?	greeting and patting everyone on the back?
4	When the possibility of a tool or methodology change is being discussed, do you prefer …	imagining the possibilities?	discussing the options?	having a flexible attitude?
5	In the training courses, do you prefer …	summarizing the meaning?	listening to the message, word by word?	taking the main point of the message?
6	When you need to understand a new class, do you initially prefer …	seeing an image which presents the respective class?	listening to someone talking about it?	looking through it and executing a test?
7	In the requirements analysis, do you prefer …	seeing a general schema of the user's views?	listening carefully to users comments?	feeling the pressure of the needs?
8	In the definition of software architecture, do you prefer …	having a general view of the situation?	listening and discussing ideas?	giving lots of suggestions?
9	If you need to program in a non-familiar place, will you prefer …	thinking and visualizing the unusual day you'll have?	talking about your programming (on that day)?	just living the experience of how you'll spend the day?
10	When you need to use a new library, do you prefer …	searching for a specialist's vision?	talking to someone who is able to discuss such use?	using another programmer's experience?
11	To acquire a new bibliography on Software Engineering you rely on:	an attractive and useful appearance (inside and outside the book).	a professional who has talked about it and recommended it.	the presence of real case studies and examples.
12	When you interview a less experienced programmer to work with you, do you prefer …	examining all aspects of his/her potential?	asking questions about specific comments in his/her resume?	mastering the tools and knowledge experienced by the interviewed?
13	To learn a new programming language, paradigm or design pattern, do you prefer:	observing the documentation diagrams.	talking to professionals who master the technique.	starting using the technique to learn by practicing.
14	When you use a website of a Software Engineering tool, what do you like to have:	screenshots showing what each module implements.	the opportunity to ask questions and talk about the tool and its features.	lots of examples of good and bad uses of the tool.

Continued on following page

15	You were suddenly invited to present your latest implementation. How would you prepare your presentation? Think of what you would like to see first if you were in public. Remember also that there is little time to prepare it.	you would make diagrams and graphics to show.	you would write down the key points of the code and practice the speech.	you would join real and practical examples to explain.
16	Now try to remember a moment which you could totally comprehend a new code. How did you better comprehend it?	by using diagrams and graphics.	by listening to the programmer's explanations and asking questions.	by executing and writing the code for some type of maintenance.

Table 1: Questions and Answers

Threats to validity: Different issues may occur during the participation of the subjects in the survey.

- **Instrumentation properly prepared (internal validity):** The participants answered the questionnaire with no supervision; hence there is a possibility that they did not understand specific questions. In order to mitigate this problem, a pilot version of the questionnaire was applied to nine software engineers (among them, two were native English speakers), in order to make it clearer and more objective.
- **Representative population (external validity):** The range of participants that answered the questionnaire was significant; however, it will be interesting to increase the number of participants, raising the representativeness of the software engineer population as well. An analysis of the participants' characteristics is presented in the next section of this work.
- **Distribution of the set of participants (conclusion validity):** The experience of the software engineers or their functions may affect the results; however, both experience and participants' functions are fairly distributed (see next section).

6.3 Survey Results

The results are conducted in three parts, as follows: raw results; statistical analysis of the results using IRT and three statistical tests: χ2 test, ANOVA and Tukey HSD; and general interpretation.

6.3.1 Raw Results

Two hundred and nine (209) software engineers answered the questionnaire in a period of 8 months (from March to October 2010). All submissions were made through the internet.

Country of Origin: Most of participants were Brazilians (89%), but there were respondents from several other countries as well. Brazil = 186 (89%), United States = 6 (2.9%), Canada = 4 (1.9%), India = 3 (1.4%), China = 2 (1%), Afghanistan = 1 (0.5%), Andorra = 1 (0.5%), Argentina = 1 (0.5%), Finland = 1 (0.5%), Germany = 1 (0.5%), Ireland = 1 (0.5%), Romania = 1 (0.5%) and Minor Outlying Islands = 1 (0.5%).

Position: managers = 47 (22.5%), analysts = 88 (42.1%) and programmers = 74 (35.4%). Contacting several of the respondents, we found that many of those who classified themselves as an analyst also do a fair amount of hands on programming.

Experience: 1–3 years = 88 (42.1%), 3–5 years = 48 (23%), 5–10 years = 35 (16.7%), more than 10 years = 38 (18.2%) (see Figure 2).

Experience (number of systems maintained): 1–5 = 108 (51.7%), 6–10 = 56 (26.8%), 11–20 = 20 (9.6%), more than 20 = 25 (12%) (see Figure 3).

Gender: female = 39 (18.7%), male = 170 (81.3%).

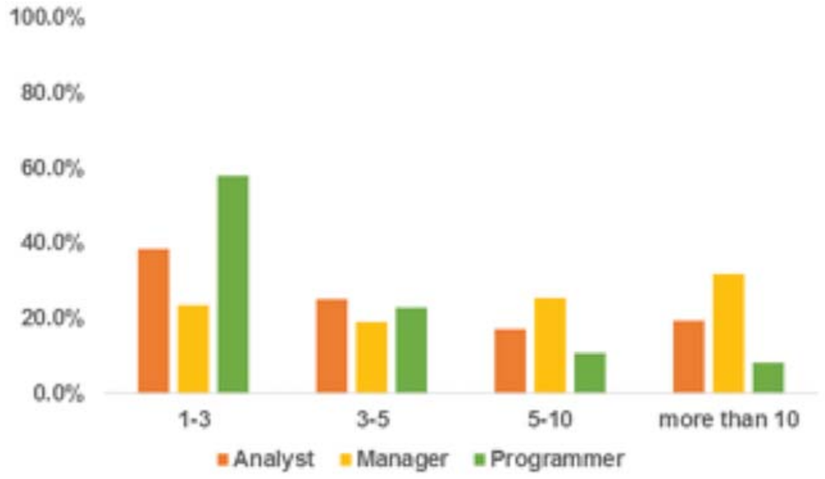


Figure 2: Raw results according to the subjects' experience

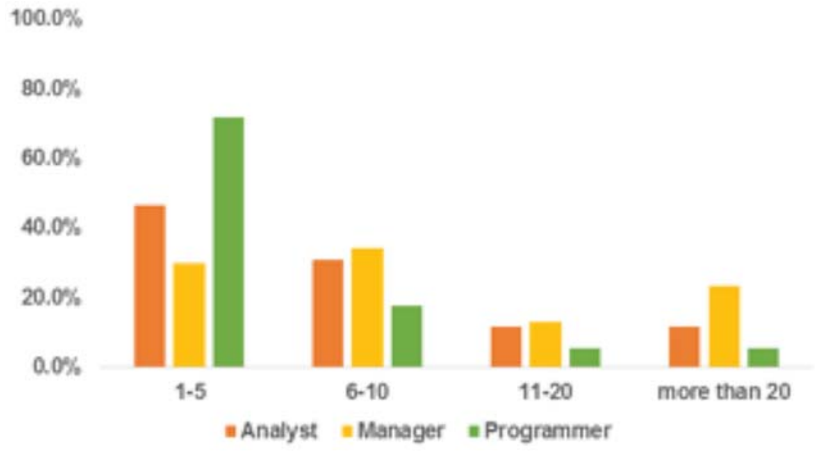


Figure 3: Raw results according to the number of maintained systems by the subjects

6.3.2 Statistical Results

The questionnaire was used to measure the levels of the three latent variables in the same subjects and to observe how the variables were distributed in the population studied. The possibility to indicate more than one preference in each question aimed to verify the multimodality.

The estimation of item parameters in this work was done through the R Software System (http://www.R-project.org), using the LTM (Latent Trait Model) package for dichotomous data, that implements the two-parameter logistic model (Rizopoulos, 2006). The two-parameter dichotomous model was used in this research because we have three different latent variables, which may not be in the same range of aptitude, i.e., they are not three alternatives of answers that measure the same latent trace. Each question has three alternatives, which can be chosen in a preferred order and each one represents one latent variable (Visual, Kinesthetic and Aural). In this way, we statistically consider three equal questionnaires for each variable, so that the answers are 0 or 1 for each question according to the selected representational system.

In order to accomplish the statistical analysis, it was necessary to make the answers dichotomous. The data input for R Software was done in three steps: (1) assignment of value 1 to the answers referring to visual system and 0 to the others; (2) assignment of value 1 to the answers referring to aural system and 0 to the others; and (3) assignment of value 1 to the answers referring to kinesthetic system and 0 to the others. These three files were individually used in the R Software, with which we obtained results.

The item characteristic curves were computed for each item in the interval ±4 in the continuous trace. Such a range is usually used for maximum visualization of information.

We observed the behaviour of the discrimination parameters in the 16 questions. A range of values between 0 and +2 is expected for this parameter. The discrimination parameters had considerable variation in all three systems (Visual: 0.17 – 1.49; Aural: -0.08 – 1.27; and Kinesthetic: 0.21 – 2.16). Values over one are considered good. Figures 4, 5 and 6 show the ICCs curves for the three most and least discriminatory questions for each of the three systems. Questions 6, 13 and 16 showed the highest discrimination for the visual system. Questions 6, 8 and 13 showed the highest discrimination for the kinesthetic system. Questions 2, 6 and 16 showed the highest discrimination for the aural systems.

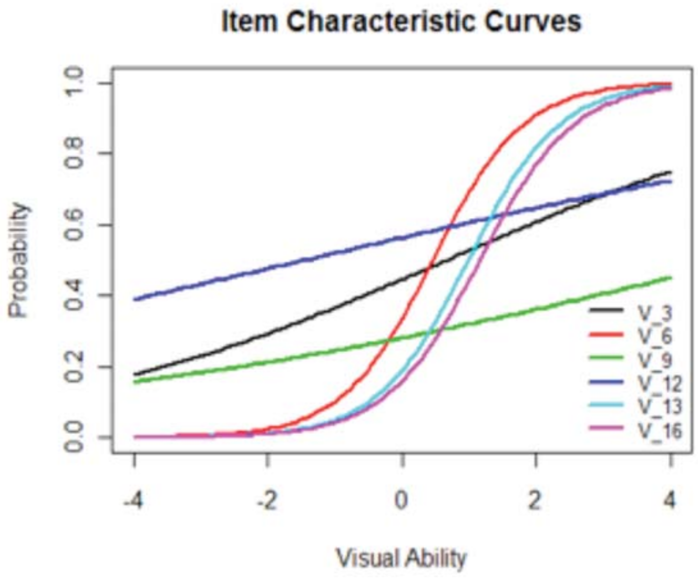


Figure 4: ICC of the three more discriminant items and three less discriminant ones of visual system

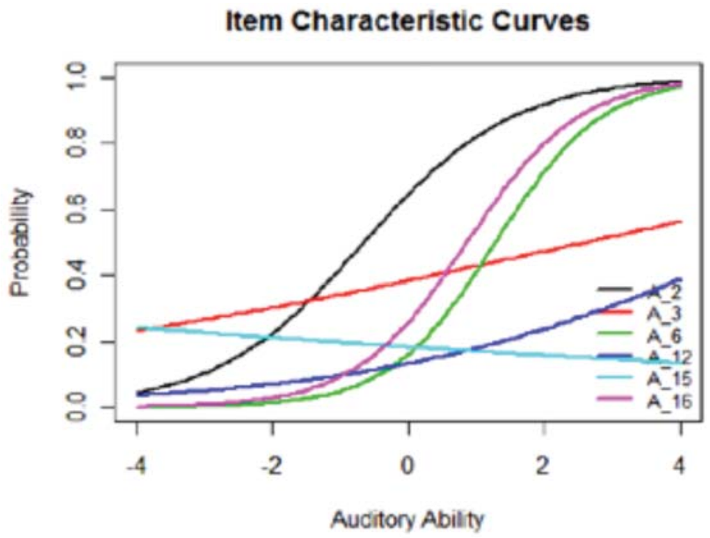


Figure 5: ICC of the three more discriminant items and three less discriminant ones of auditive system

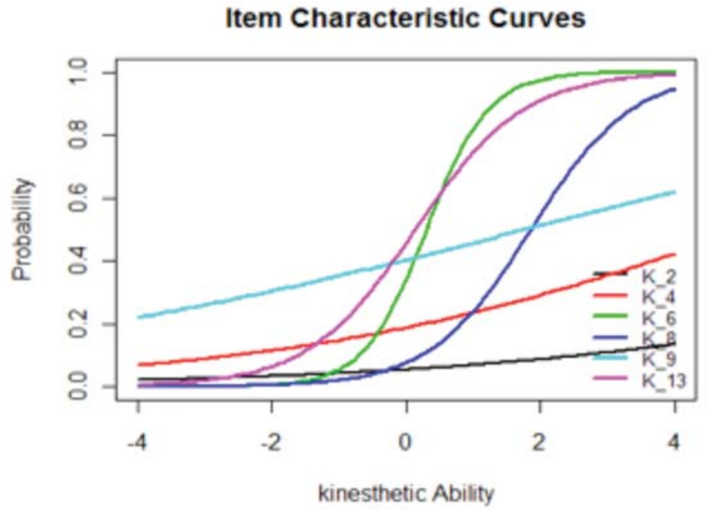


Figure 6: ICC of the three more discriminant items and three less discriminant ones of kinesthetic system

In relation to the difficulty parameter (bi), which indicates the region, in the proposed scale, where the item has higher information, we can highlight that there were difficult and easy items for the three systems (see Table 2). This parameter does not represent the quality of the item, because there are very difficult and very easy items that do not present good discrimination. The three items with higher difficulty were, in descending order, items 9, 5 and 11 (visual); 12, 5 and 3 (aural) and 2, 4 and 7 (kinesthetic). Regarding the items with lower level of difficulty, in ascending order, we have: 12, 8 and 6 (visual); 15, 2 and 7 (aural) and 5, 14 and 13 (kinesthetic).

Difficulty parameter			
Questions	Visual	Auditory	Kinesthetic
1	0.646337	1.1747505	0.9298117
2	0.9029319	-0.6545072	11.3411584
3	0.6664545	2.6205637	2.4384595
4	1.1587243	0.1239423	5.0827783
5	3.4281543	3.6750966	-1.2966636
6	0.4569115	1.2858888	0.3037288
7	1.263104	0.101664	3.0356221
8	0.2380938	0.4554262	1.8603075
9	5.0674268	1.5746194	1.7733581
10	1.3409829	0.9117935	2.0398499
11	2.9625598	0.200122	0.3844435
12	-1.4506931	5.2858088	1.7353657
13	0.9839088	1.5159333	0.1381952
14	0.9848366	2.5229249	0.136623
15	0.8747037	-16.7234852	0.3147457
16	1.1590743	0.863275	0.1625408

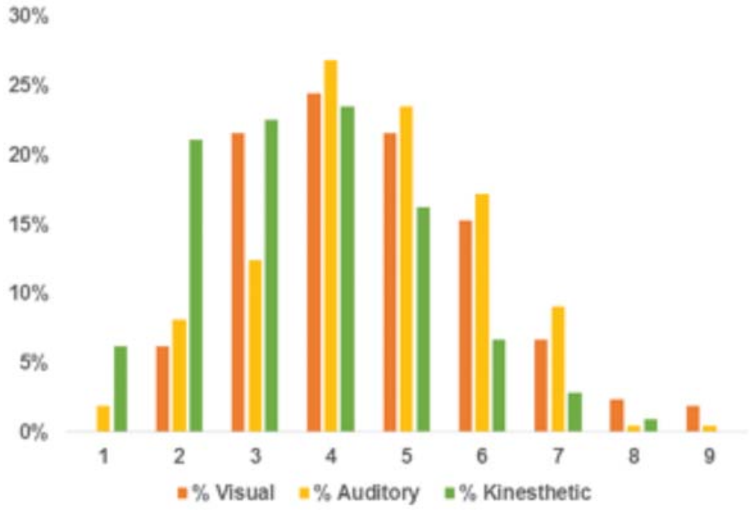
Table 2: Difficulty parameter

The difficulty parameter means that in order to give an answer referring to the visual system in item 9, we must have a bigger latent trace, i.e., the visual ability must be higher than the item difficulty. Item 12 on the other hand can be answered by people who have little visual ability. In other words, this visual option is chosen by most of the software engineers. If the subject did not endorse his/her answer, his/her ability was lower than the item difficulty.

After obtaining the item (questions) parameters, we proceeded with the subjects' scores in each system. The system score for a subject is given through the algebraic calculus of the two-parameter model.

As it would be impractical to make a graphic with the score frequencies, since they are continuous, intervals of classes were created so that we can see the behaviour of the population in the three systems. Scores were divided in nine sets of intervals in order to analyze their distribution, varying from -1.541 to 2.636. The estimated scores have little variance between each other, and the score distribution, as IRT claims, presented a normal distribution, implying that some subjects of the population have low level in the systems, others have high level, and most of them have medium level, as it can be seen in Figure 7. For instance, the subjects who are in range 1 (Visual – 0%, Aural – 2% and Kinesthetic – 6%) have lower scores (less or equal to -1.5) and subjects in the range 9 (Visual – 2%, Aural – 0% and Kinesthetic – 0%) have the highest scores (more or equal to 2.0).

The next step is to do the classification of the participants. The higher the score obtained for a system is, the higher the subject's prioritization for the system is. The software engineers' preferred orders of representational systems were then obtained. A sequence of the letters AKV indicates the preferred order among the systems, for instance, KAV indicates Kinesthetic as the first, Aural as the second and Visual as the third preference. The results were: 18.7% of the participants were AVK, 18.2% KAV, 18.2% VKA, 15.8% AKV, 14.4% VAK, 12.9% KVA, 0.5% VK, 0.5% VA, 0.5% KV and 0.5% V. Figure 8 shows a graphic representation of the classifications.



Scores were divided in nine sets of intervals, varying from -1.541 to 2.636

Figure 7: Scores Distribution

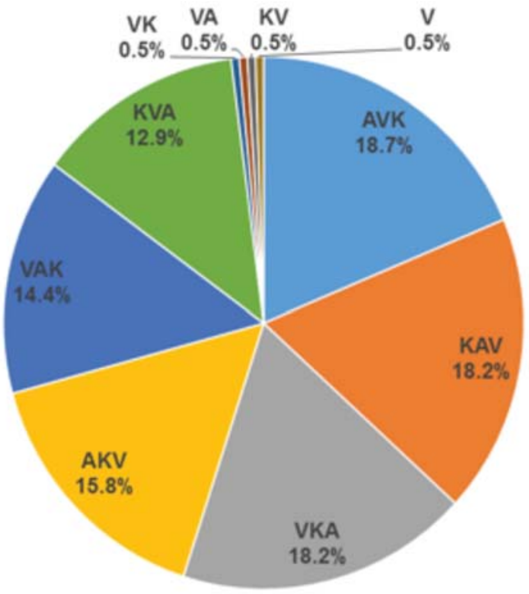


Figure 8: Classification of the Software Engineers' preferences

IRT claims invariant classifications. We wanted to verify it anyway, checking the classifications for different positions and experience. Through a $\chi2$ test, we checked the expected and observed frequencies to evaluate if there was a significant difference between them. For analysts, managers and programmers there were no differences in the classifications, in the significance levels of 5% and 10%. Considering experience, there were no differences in the classifications either, in the significance levels of 5% and 10%. Hence, independent of position or experience, the most used cognitive strategy among the interviewed is AVK (18.7% of the total of subjects).

Finally, we tested the hypothesis previously formulated. The goal was to prove that the classification presents a significant difference between the subjects first, the second and the third preferred representational system.

In order to accomplish this analysis, we used the ANOVA test and Tukey HSD. ANOVA indicates the probability for the null hypothesis to be true, i.e., probability of no differences existing between any classifications. If the null hypothesis (H_0) is rejected, we can infer that there is indeed a difference among the classifications tested, i.e., the subjects indeed have a PRS. For this, we use the individual score of the subjects for their first, second and third choice. It is very important to note that it does not matter what the PRS of a given subject is. We want to show that there is a significant difference between the scores of the first, the second and the third option of the subject, no matter what they are.

ANOVA					
Score	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	254.761	2	127.380	821.897	.000
Within Groups	96.710	624	.155		
Total	351.471	626			

Table 3: ANOVA with scores associated with the preference order

All we have to do is to observe the theta scores for each subject's first, second and third options, and then ANOVA those scores for the 209 data points. Table 3 shows the results. The null hypothesis is rejected with a very high significance (p value < 0.05; see the Sig. column with p < 0.0001). There is at least one difference between the groups analyzed. We then used Tukey's test to identify which groups are different from each other. Table 4 shows that groups 1, 2 and 3 are all different from each other.

(I) Option	(J) Option	Mean Difference (I – J)	Std. Error	Sig.	Interval	
					Lower Bound	Upper Bound
1	2	.91902	.03851	.000	.8285	1.0095
	3	1.55266	.03851	.000	1.4622	1.6431
2	1	-.91902	.03851	.000	-1.0095	-.8285
	3	.63364	.03851	.000	.5432	.7241
3	1	-1.55266	.03851	.000	-1.6431	-1.4622
	2	-.63364	.03851	.000	-.7241	-.5432

Table 4: Tukey test with scores associated with the preference order

6.4 Analysis and Interpretation

The results indicate that software engineers (SE) have different preferred channels. The ANOVA and Tukey results showed that there is a significant difference that enables us to reject the null hypothesis. However, there are many other interesting pieces of information that came out of the data.

The order of preference for the channels is very balanced among the participants. The larger was the AVK group (18.7%) and the smaller was the KVA group (12.9%). This is a small difference considering we have six possible orders of preferences (AVK, AKV, VAK, VKA, KAV, and KVA). This indicates that not only is there a significant difference in the order of preferences, but also that the population itself is very diverse in terms of what channels they prefer.

The results also showed that all channels were used by an overwhelming number of software engineers (98% of the participants). There are sizeable groups that prefer the aural, kinesthetic or visual channels. So none can be ignored or down played. In fact, the aural channel is the number one preference of 34.5% of the participants. This is interesting considering that most tools of our trade still have limited support to audio and video communication and filming/recording integrated into them. Moreover, except for requirements engineering, few software engineering methods emphasize aural or video communication. Would not it be viable to associate audio explanations to software architecture, design of code artifacts, source code, for example? Why don't we record or film the interviews with clients and provide them to the programmer so they can better understand their requirements? The results presented here stimulate us to consider some of those options.

The Item Response Theory is a robust and disseminated technique for measuring subjective traits of human beings. It is potentially a very useful tool for the empirical software engineering community. They can be employed to measure software engineer abilities and preferences over the area of heterogeneous populations, domains and project contexts.

In the context, a latent variable cannot be measured directly, so it should be validated to the questionnaire's respondents themselves (Wu and Adams, 2007). The subject is not capable of specifying, by his/her perception, the level in which some variables are distributed, since multi–modality exists among subjects (Leite *et al*, 2010). In order to measure the level in which such variables are distributed, it is necessary to use a model like IRT that does provide assertiveness in its parameters.

Finally, we should point out two concerns that a lay person may raise about some of the questions:

1. The answer for some questions may seem overly limiting, because we only give three options for each question. However, from a psychological point of view, we just want to find out the preference among those options. Moreover, the questionnaire values the total set questions (Peters *et al*, 2008; VARK Learn, 2010).
2. Some answers, such as "having a flexible attitude", may seem overly generic and some software engineers might always select them, even if not as their first choice. To reiterate, each answer, even if generic, has a psychological background and a goal (Dent, 1983; Matthews, 1991; VARK Learn, 2010). Moreover, we calculate the IRT discrimination and difficulty parameters for each question (see Section 5). The lower the discrimination parameter is, the less discriminative the item will be. By answering this item, a person identifies less strongly to the latent variable that it represents. The excessive selection of an item will also lower its score.

7. Conclusion and Future Works

This article presented a questionnaire to identify the preferred representational system (PRS) of software engineers through a simple yet powerful experimental strategy. As speculated by our hypothesis, the results of the study indicate that software engineers have different PRSs. As important as that, the results also show that:

1. Software engineers use all three representation systems, and;
2. All three representation systems are very important for some part of the population.

The score of the aural system is more an important indication that all channels were chosen by an overwhelming number of software engineers (98% of the participants). Despite this, except for requirements engineering, only a few software engineering methods emphasize aural or video communication.

The order of preference for the channels is very balanced among the participants. The larger was the AVK group (18.7%) and the smaller was the KVA group (12.9%). This is a small difference considering we have six possible orders of preferences (AVK, AKV, VAK, VKA, KAV, and KVA). There are sizeable groups that prefer the aural, kinesthetic or visual channels. So none can be ignored or down played, in other words, the area could invest more on researches about multimedia aids to help software comprehension. This is interesting considering that most tools of our trade still have limited support to audio and video communication and filming/recording integrated into them. Why don't we record or film the interviews with clients and provide them to the programmer so they can better understand their requirements?

In fact, the aural channel is the number one preference of 34.5% of the participants. In this case, the types of responses in the "auditory" column in Table 1 can be interpreted as meaning that programmers are inclined to get help from their peers when facing a technical problem: however, seven questions (3, 5, 7, 9, 12, 14 and 15) describe other situations and have a psychological background and a goal (Dent, 1983; Matthews, 1991; VARK Learn, 2010). Moreover, in the future, a programmer will be able to click on a link in the source code and to listen to explanations recorded by a co-worker. In Colaço *et al* (2014), we combine two approaches (questionnaire and text mining) in a controlled industrial environment and we also experiment with this situation (clicking and listening to recorded explanations about part of a source code). In this study, we interview the subjects to do a complete matching between the PRS identified by the different approaches and we identify initial positive responses to deal with recorded explanations (help from their peers, but on the fly).

We consider this an important finding of this work, because the three systems can be stimulated in the order of preference. In other words, it is noteworthy that the PRS is only the way a person (strategy), in specific contexts, prefers to use to communicate and learn, but all channels can be used in a strategy.

The use of IRT also showed that the technique has much to offer to our area. IRT can help software engineering researchers to measure latent traits over heterogeneous populations, facilitating meta-analyses over different organizations and cultures. We also believe that we can successfully use the measures of power of discrimination of items (questions) provided by the technique to improve questionnaires like ours.

The developed questionnaire is part of a family of experiments to detect and validate PRS. Previous studies showed that developers have different PRS using a completely different approach that used text mining to analyze OSS and industry mailing lists (Colaço *et al*, 2010; 2012; 2014). These works are the first steps on a promising road toward understanding latent traits of software engineers through the use of psychometrics techniques.

As future works, we will start to look at multimedia aids to help software construction and comprehension, video interviews to help "use cases" comprehension or to substitute it in some cases, and video records to help Requirements Engineering and Business Process Management (BPM).

References

ALLPORT, G.W. (1937): Personality. New York: Holt New York.

BADENOCH, S.N. (1986): Personality type, learning style preference, and strategies for delivering training to a select group of store managers. Unpublished doctoral dissertation, University of Minnesota.

BAILENSON, J.N. and YEE, N. (2005): Digital chameleons: automatic assimilation of nonverbal gestures in immersive virtual environments. *Psychological Science*, 16(10): 814–819.

BAKER, F.B. (2001): The Basics of Item Response Theory. ERIC Clearinghouse on Assessment and Evaluation.

BANDLER, R. and GRINDER, J. (1979): *Frogs into Princes: Neuro-linguistic Programming*, Vol. 15. Moab, Utah: Real People Press.

BASILI, V.R. and WEISS, D.M. (1984): A Methodology for Collecting Valid Software Engineering Data. IEEE Transactions on Software Engineering, 10(6): 728–738, November.

CASSIDY, S. (2004): Learning styles: an overview of theories, models and measures. *Educational Psychology*, 24(4): 419–444.

COLAÇO, JR, M., MENDONÇA, M.G. and RODRIGUES, F. (2009): Data Warehousing in an Industrial Software Development Environment. In: *33rd Annual IEEE/NASA Software Engineering Workshop*, 131–135.

COLAÇO, JR, M., MENDONÇA, M.G., FARIAS, M.A. and 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*, Cape Town, SA, 126–129.

COLAÇO, JR, M., MENDONÇA, M.G., FARIAS, M.A., HENRIQUE, P. and CORUMBA, D. (2012): A Neurolinguistic Method for Identifying OSS Developers Context-Specific Preferred Representational Systems. In: *International Conference on Software Engineering Advances*, Lisbon, 112–121.

COLAÇO, JR, M., FARIAS, M.A., MACIEL, I., HENRIQUE, P. and MENDONÇA, M.G. (2014): Triangulating Experiments in an Industrial Setting to Evaluate Preferred Representational Systems of Software Developers. In: *28th Brazilian Symposium on Software Engineering (SBES)*, Maceió, 71–80.

DENT, K.A. (1983): Cognitive Styles: Essence and Origins: Herman A. Witkin and Donald R. Goodenough. *Journal of the American Academy of Psychoanalysis*, 11: 635–636.

DIEHL, S. (2007): *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*. New York: Springer Science & Business Media.

DILTS, R.B. (1980): *Neuro-linguistic Programming:* Volume I (The study of the structure of subjective experience). California: Meta Publications.

EINSPRUCH, E.L. and FORMAN, B.D. (1985): Observations concerning Research Literature on Neuro-Linguistic Programming. *Journal of Counseling Psychology*, 32(4): 589–596.

FLEMING, N.D. (1995): I'm different; not dumb. Modes of presentation (VARK) in the tertiary classroom. In: *Research and Development in Higher Education, Proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA)*, HERDSA, 18: 308–313.

GARITY, J. (1985): Learning styles: Basis for creative teaching and learning. *Nurse Educator*, 10(2): 12–16.

HARTLEY, J. (2008): Learning and studying: A research perspective. London: Routledge.

HAWES, N., MARSHALL, S. and ANSLOW, C. (2015): CodeSurveyor: Mapping large-scale software to aid in code comprehension. In: *Software Visualization (VISSOFT)*, 2015 IEEE 3rd Working Conference on, IEEE, 96–105.

HILL, M.M. and HILL, A. (1998): A construção de um questionário. Lisboa: Dinâmia.

HUNGERFORD, B.C., HEVNER, A.R. and COLLINS, R.W. (2004): Reviewing Software Diagrams: A Cognitive Study. *IEEE Transactions on Software Engineering*, 30(2): 84–96.

KARIM, J. (2010): An item response theory analysis of Wong and Law emotional intelligence scale. *Procedia-Social and Behavioural Sciences*, 2(2): 4038–4047.

KORZYBSKI ,A. (1994): *Science and Sanity: An introduction to non-Aristotelian systems and general semantics*, 5th edition. Institute of General Semantics.

KUMANO, S., OTSUKA, K., MIKAMI, D. and YAMATO, J. (2011): Analyzing empathetic interactions based on the probabilistic modeling of the co-occurrence patterns of facial expressions in group meetings. In: *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, IEEE*, 43-50.

LEITE, W.L., SVINICKI, M. and SHI, Y. (2010): Attempted Validation of the Scores of the VARK: Learning Styles Inventory With Multitrait-Multimethod Confirmatory Factor Analysis Models. Educational and Psychological Measurement, 70(2): 323–339.

LORD, S.P., SHENG, E., IMEL, Z.E., BAER, J. and ATKINS, D.C. (2015): More than reflections: empathy in motivational interviewing includes language style synchrony between therapist and client. *Behaviour therapy*, 46(3): 296–303.

MALDONADO, J.C., CARVER, J., SHULL, F., FABBRI, S., DÓRIA, E., MARTIMIANO, L., MENDONÇA, M. and BASILI, V. (2006): Perspective-Based Reading: A Replicated Experiment Focused on Individual Reviewer Effectiveness. *Empirical Software Engineering*, 11(1): 119–142.

MALETIC, J.I. and KAGDI, H. (2008): Expressiveness and Effectiveness of Program Comprehension: Thoughts on Future Research Directions. In: *Proceedings of the Frontiers of Software Maintenance*, CHI, IEEE, 31–40.

MATTHEWS, D.B. (1991): Learning Styles Research: Implications for Increasing Students in Teacher Education Programs. *Journal of Instructional Psychology*, 18(4): 228–236.

MOODY, D. (2009): The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35(6): 756–779.

PETERS, D., JONES, G. and PETERS, J. (2008): Preferred 'learning styles' in students studying sports-related programmes in higher education in the United Kingdom. *Studies in Higher Education*, 33(2): 155–166.

REEVE, B.B. and FAYERS, P. (2005): Applying item response theory modelling for evaluating questionnaire item and scale properties. *Assessing quality of life in clinical trials: methods of practice*, 2: 55–73.

REISE, S.P., AINSWORTH, A.T. and HAVILAND, M.G. (2005): Item response theory: Fundamentals, applications, and promise in psychological research. *Current Directions in Psychological Science*, 14(2): 95–101.

RIDING, R. and CHEEMA, I. (1991): Cognitive styles: An overview and integration. *Educational Psychology*, 11(3-4): 193–215.

RIZOPOULOS, D. (2006): Ltm: An R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, 17(5): 1–25.

SINGH, J. (2004): Tackling Measurement Problems with Item Response Theory: Principles, Characteristics, and Assessment, with an Illustrative Example. *Journal of Business Research*, 57(2): 184–208.

SIVILOTTI, P.A. and PIKE, S.M. (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.

SRIPADA, S.K. and REDDY, Y.R. (2015): Code comprehension activities in Undergraduate Software Engineering course – A case study. In: *Software Engineering Conference (ASWEC), 2015 24th Australasian, IEEE*, 68–77.

TAN, C.S.S., LUYTEN, K., VAN DEN BERGH, J., SCHÖNING, J. and CONINX, K. (2014): The role of physiological cues during remote collaboration. *Presence: Teleoperators and Virtual Environments*, 23(1): 90–107.

TOSEY, P. and MATHISON, J. (2003): Neuro-linguistic Programming and Learning Theory: a response. *The Curriculum Journal*, 14(3): 371–388.

TOSEY, P. and MATHISON, J. (2007): Fabulous Creatures of HRD: A Critical Natural History of Neuro-Linguistic Programming. In: *8th International Conference on Human Resource Development Research & Practice across Europe*, Oxford Brookes Business School, 1–17.

TRAVASSOS, G., SHULL, F., FREDERICKS, M. and BASILI, V.R. (1999): Detecting Defects in Object Oriented Designs: Using Reading Techniques to Increase Software Quality. In: *ACM Sigplan Notices*, ACM, 34: 47–56.

TURAN, B. and STEMBERGER, R.M. (2000): The effectiveness of matching language to enhance perceived empathy. *Communication and Cognition*, 33(3-4): 287–300.

VAN BAAREN, R.B., HOLLAND, R.W., STEENAERT, B. and VAN KNIPPENBERG, A. (2003): Mimicry for money: Behavioural consequences of imitation. *Journal of Experimental Social Psychology*, 39(4): 393–398.

VAN SOLINGEN, R. and BERGHOUT, E. (1999): *The Goal/Question/Metric Method: A practical guide for quality improvement of software development*. McGraw-Hill Publishing Company.

VARK LEARN (2010): A Brief Biography of Neil D. Fleming. Available in: http://vark-learn.com/introduction-to-vark/biography/. Acessed on: 25 jul. 2010.

WOHLIN, C., RUNESON, P., HÖST, M., OHLSSON, M.C., REGNELL, B. and WESSLÉN, A. (2012): *Experimentation in software engineering*. Springer Science & Business Media.

WU, M. and ADAMS, R. (2007): Applying the Rasch model to psycho-social measurement: A practical approach. Melbourne: Educational Measurement Solutions.

Biographical Notes

Methanias Colaço Júnior

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 1997. 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.*

Maria de Fatima Ribeiro Menezes

Maria de Fatima Menezes *graduated in 2009 in Statistics from the Federal University of Sergipe, Brazil, with a thesis in logistic regressions applied to a credit scoring model. She is part of two academic research programs at Federal University of Sergipe: (i) Statistical Models in Discriminatory Processes with focus on Discriminant Analysis, Cluster Analysis, Logistic Regression and Item Response Theory (IRT) and (ii) Competitive Intelligence Research and Practice Group. She has been a lecturer in seminars about Item Response Theory (IRT) at UFS Statistics and Technology Department. Her practical experience includes the following areas: credit risk, statistics models for Behaviour and Credit Score, Customer segmentation and life cycle, geographic information system (GIS), international bonds and investments, Bloomberg database. She developed this background in credit card and other financial service companies located in Brazil, Chile and United Kingdom. Currently, she works at Pairstech Capital Management LLP in London.*

Daniela Corumba

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 inhouse and used across multple 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 Analysi 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 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.

Manoel Mendonça

***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 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 Centre 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.*

Breno Santana Santos

***Breno Santana Santos** graduated in 2015 in Information Systems at Campus Professor Alberto Carvalho, Federal University of Sergipe, Brazil, with a thesis in Comparative Analysis of Text Mining Algorithms Applied to Historical Public Accounts. He is part of the academic research program at Federal University of Sergipe, the Competitive Intelligence Research and Practice Group. His practical experience includes the following areas: Data Mining, Text Mining and Analysis of Psychological Characteristics of Individuals through Text Mining Techniques. Currently, he is a Master's student in a Postgraduate Program in Computer Science at Federal University of Sergipe and is working on research related to Approaches to gain Empathy through Text Mining Techniques.*