Soft Systems Engineering Tools in Requirements Elicitation

MARCEL J. SIMONETTE, FABIO L. L. SANCHES, EDISON SPINA
KNOMA – Laboratório de Engenharia de Conhecimento
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, Travessa 3, nº 158 – Sala C2-37
São Paulo, BRASIL
{marceljs,fabio.sanches,spina}@usp.br
http://www.poli.usp.br

Abstract: - **Humans beings are the main actors in any system created in the Information and Communication Technology. From system conception to its discard, humans are present. Humans are the system designers, system developers, system users and humans are affected by the system, positively or negatively. Although human is the main actor, several systems engineering methods do not include humans' aspects during the system development. Given the cost involved in system changes after understand its requirements, one must include the human dimension in system requirements elicitation phase. This requirements engineering process is a system inside the system to be treated, it is a system where the components are humans activities. This paper deals with the use of soft system approach in the elicitation process and discusses a perspective of method selection, as a way to identify human requirements for reduce the discrepancy between the expected features of a system and the ones perceived.**

Key-Words: - **Human Factors, Requirements Elicitation, Requirements Engineering, System Engineering, System Thinking, Soft Systems,**

1 Introduction

Systems Engineering, differentially from others traditional engineering disciplines, do not follow a fundamental phenomenon's set based in physical properties and relations. Instead, it deals with the necessary knowledge to manage these phenomenon's, dealing with the system emergent properties, looking for a way to get control about the system entropy [1], [2].

Requirements Engineering is another engineering discipline alone, crucial in the development of any product or service. This Engineering has a life cycle that leads Systems Engineer in the process of requirements elicitation, negotiation, documentation and validation of the systems to be developed. Systems Engineer makes use of this process to execute a task that Kossiakoff and Sweet [3] calls Concept Definition phase, and INCOSE [4] calls Concept Stage. Both refer to the initial phase of various life cycle models placed by engineering statements to systems development.

The development process used by engineers to create Information and Communication Technology (ICT) Systems – whether Agile, eXtreme Programming, Prototyping, Rational Unified Process, or any other method – is irrelevant to the need for understanding the System Requirements [5]. The importance of correct requirements understanding has already been pointed out in terms of software development cost at the end of the 1980s, when Boehm and Papaccio [6] argued that to correct defects that were found after System delivery has a cost 50 to 200 times greater than if these defects had been identified in the early stages of the life cycle. The Software System industry still has problems when requirements are the subject, as pointed by Firesmith [7], the software industry data suggests that nearly 80% of the software rework may be assigned to requirements problems. Another importance about corrected requirements understanding is the knowledge resulted by this process, which is subsidy for various other phases of System life cycle [4].

Systems made by humans are technical and technological triumphs, bringing to humanity products and processes never seen before. Many of these systems have human and social interfaces that demand a series of conditions that are recognized by engineering, that use approaches to treat the humans factors involved in all System life cycle [8], [9], [10].

Engineering must avoid human error in systems built by man. This demand has led Requirement Engineering to identify human factors, and Systems Engineering to consider then in their projects. However, in doing so, the human appears in the Systems as another component representing the cognitive and ergonomics aspects of a System consisting of user, product and environment [2], [11], [12]. To improve the human-system interaction, much has been done in what concerns the Systems usability, but we must go beyond.

The progress that humans are getting in building tools, methods and artifacts are promoting an increasingly revolution in humanity, passing through Stone Age, Bronze Age, Iron Age and, as stated by Ackoff [13], the Machine Age and the Systems Age, our age today.

Go beyond human factors in ICT Systems design is a necessary action since we must not repeat, in Systems Age what Ackoff stated to be the great irony that occurred in the Machine Age, where the humans creations, to free man from work, have led to a dehumanization [13], [14].

This paper proposes the use of Consensual Methods as Soft Systems Engineering Tools to identify humans' requirements in the requirement elicitation process. Furthermore, a perspective to Consensual Methods selection is presented. This session introduces the need for going beyond human factors in ICT Systems development. Second session presents the motivation to develop this work; third session proposes an approach to go beyond human factors; session four focus the approach at Requirement Elicitation and session five presents the Consensual Methods used by Soft System Engineering School; session six illustrates a perspective to a Consensual Method selection on the basis of System life cycle demands, and it is followed by final considerations, that comments some works in progress, acknowledges and references.

2 Systems and Requirements

The word System has a subjective nature, and refers to organization forms recognized by humans. The constructivist view of reality determines that a system does not exist in real world regardless human mind [1].

Life on Earth can be considered as complexes interconnections between two Systems that humans can recognize: The Natural Systems and the Systems created by humans. Checkland [15] classifies the Systems created by humans in three distinct classes: Designed Physical Systems, Designed Abstract Systems and Human Activity Systems.

It is possible to investigate, describe and learn from Natural Systems, create and use Physical and Abstracts Systems and use Engineering methods to deal with Human Activity Systems. In all this classes, there is a search for controlling their emerging properties. Systems Engineering works to deal with this control, synthesizing Systems that have the desired properties and eliminating or reducing the unwanted ones, leading the Engineering in control of complex Systems, where the elements are diverse and have intricate inter-relationships [2], [3]. Hitchins [2] defines System Engineering as the art and science of creating whole solutions to complex problems, and this is the definition adopted by the authors of this paper.

2.1 System Theory

Ideas as holism as an interdisciplinary science and the growing recognition of existence and utility of isomorphism between disciplines has created a awareness that certain concepts, ideas, principles and methods, were applicable to Systems in general. Klir [17] argued that this led to the concept of General Systems, General Systems Theory and Systems Theory; he also stated that the term General Systems Theory is due to Ludwig Von Bertalanffy, who has used it in speeches in the '30s, although the presence of the term in his book took place just after Second World War.

Skyttner [1] states that System Theory deals, in an abstract way, with Systems general properties, regardless physical forms or application domains. It provides a way to abstract reality, simplifying and at the same time capturing system multidimensionality. As an epistemology, it structures not only thinking about reality, but also thinking about the own thinking. As an applied science, it is a metadiscipline, with content capable of being transferred from discipline to discipline.

Systems Theory is knowledge about knowledge and attempts to add and integrate those aspects that seen to be not adequately addressed by the classic science, the science of the Machine Age.

2.2 System Thinking

Systems ideas provide a way of thinking about any kind of problem. System Thinking is how System Theory is put into motion to thinking problems. System Theory has its laws and principles that are a kind of language framework of Systems ideas, a holistic language. A language of Systems, interaction and design, that enables to understand and frame problems [15], [18].

Checkland [15] states that Systems Thinking is not itself a discipline, except to the extent that there will be few people whose professional concern is with Systems concepts as such.

The words holism and systemic, so frequently used in the Systems movement, are founded on understanding the concept of wholeness, focused on

System view, surrounding environment and the contextual frameworks within which Systems exist.

Descartes's dictum that every problem should be broken down into as many separate simple parts as possible – reductive analysis – is the most successful technique that has ever been used in science. System Thinking is an approach to problems where reductionist method of science cannot cope, and Hitchins [2] states that it came to the attention of the Engineering, which had experiencing difficulties in applying their engineering practices (reductionism and determinism) to Systems that included humans.

Kralj [19] states that forgetting about the context is very easy. Usually engineers are specialists in parts of reality, and the contact with others parts of this reality make them strangers. However, parts of reality do matter, as there are interdependence between these parts of reality, otherwise it would not be a reality, meaning that context matters even more than parts alone. This way of thinking about problems, not only separates the parts of the problem, but also considers the parts as a major problem, and their relationship.

Ackoff [20] suggests three ways in which problems can be addressed: They can be resolved, solved or dissolved. To resolve a problem is to find an answer that is "good enough"; one witch satisfies it. To solve a problem is to find the correct answer, as in solving an equation. To dissolve a problem is to change the situation in some way such that the problem disappears.

Hitchins [2] states that there are two approaches, two System Engineer Schools, to treat a problem:

1. *Hard Systems School*: Its concern to create systems that can be introduced in a problematic situation to solve the problem.
2. *Soft Systems School*: Its concern to look at the problem symptoms and try to repair, decrease or work around it, in order to suppress the symptoms to resolve the problem. The result is not a new System, but one that has been "mended", "repaired", "enhanced", "improved", etc.

The first school is characterized by the concept of Hard System Solution, where the solution has a clear purpose and will be developed, delivery, put to work, supported and eventually replaced at the end of its life cycle. While recognizing the importance of interaction and process, this school emphasized functional, structural and architectural aspects of the solutions. The Soft System Solution characterizes the second school, which investigates the problem to be treated, seeking to understand the problem nature, looking for practical experiences and interactions with the problem, trying to understand the situation and propose solutions to improve the situation [2].

Checkland [21] points out that in literature there are statements that Hard approach is appropriate for well-defined technical problems and that Soft approach is suitable for situation of unclear definition, situations involving human and cultural aspects. He argues that these definitions do not characterize correctly the difference between Hard and Soft approach, since the right idea is regarded as how the word System is used, which is related to a perception that people have of the System. An engineer can use the Hard approach with problems that he can observe and treat with traditional Engineering methods. It is related to Hard Systems (Natural Systems, Abstract Systems and Physical systems), but this approach may not have successful when applied to problems where complexity and confusion are observed, where there is not a consensus between the people involved in the problematic situation about the problem definition that is causing the situation being experienced.

The goal of Soft approach is to determine the problem, since different people involved with the problematic situation have different priorities issues. Such issues are not resolved by a single decision-maker, but by group decision-making [19].

2.3 Requirements

System Engineering has several life cycle models, and, in all of them, the initial phase is a Requirement Process. In this process, the Elicitation phase gives the elements to System Engineer understand the problem to be treated. To do it, the engineer needs draw upon the knowledge and experience of the organization directors, managers, employees, etc., that have a problematic situation, i.e., that are demanding a System.

The System Engineer needs to talk with people that are demanding the new Systems and to the people that will be affected, positively or not, by the system. Usually all these people are organized in groups, formals or not, with different purposes; such that the whole has no clear purpose and the groups pull in different and often conflicting directions.

The elicitation phase is essentially a Human Activity System; the use of Soft Systems approach can bring some degree of order to the situation of multiple demands, purposes, issues and problems. This approach can give order to the Requirements Gathering Process, and achieve a point where designs and solution can be manifested, and the System Engineer can identify the three requirements types that Kano [22], [23], [24] states that must be present on any products or services. These requirements types are:

1) *Normal requirements*: These are the requirements that are explicitly required. Requirements identified when the engineer ask to people involved with the System what they wish.
2) *Expected requirements*: These requirements are so basic that sometimes people may fail to mention them, because they think that it is unnecessary request them explicitly. A System without these requirements is very dissatisfying, but meeting these requirements often goes unnoticed.
3) *Exciting requirements*: These requirements are the ones that, if not present in the System, their absence will not be perceived, will not dissatisfy the people with interests in the System. As this requirements are not formalized by requirement process participants, i.e., they are not apt to voice them, it is the engineer responsibility to explore the problem and opportunities to uncover such unspoken desires. For example, as the engineer increase his knowledge about users needs and the problematic situation, he can use his experience to propose features that were not requested but that can improve the system efficiency and effectiveness.

These requirements allows the engineer to understand how meeting or exceeding the stakeholders expectation affects satisfaction in their relationship with the System. The presence of these three types of requirements, and the identification and consideration of human dimensions, are essential for people involved with System development feel welcomed by the System. The engineer may draw upon a variety of methods to extract requirements, but to respect the human and get the information from stakeholders to development the System he needs to develop a consensus from the representative group of individuals. In addition, the engineer must consider what Kumlander [25] states about requirements: "the requirements of a system are not perfect". Usually the requirements have an uncertainty, due to errors and loss of information and due to changes during the development of a System, because of new regulations or decisions that have to be adopted.

Considering the human is beyond the considerations that exist on human factors literature [8], [9], [10]. Human factor specialists still add value to Systems development, but it is necessary to go beyond human factor and have humans' dimensions, as values and intentions, identified in the beginning of System design, in the Requirement Elicitation phase. The costs to change System Requirements after the beginning of its implementation, and the Robertson and Robertson [5] claim that a product or project fail unless there are correct understandings of the Systems requirements, by engineer and by the people that are demanding the System, justifies the capturing of human dimensions in the beginning of the Requirement Process.

3 An approach to go beyond human factors

During the 1940s began the Systems Age. An Age concerned with systems that allow choice of both meanings and purposes, and has humanization as one of the central problems [13], [14], [20].

Skyttner [1] states that to understand humans and environment as part of a System of interactions, it is necessary to study this interaction in multiple perspectives and holistically.

The application of System Thinking Methods and System Theory Principles and Laws can provide valuable tools to a System Engineer. Tools through which he can see the System, the environment and the social and technical context in which the System will be used.

3.1 Sociotechnical System

Eric Trist and Fred Emery, who worked together as consultants at the Tavistock Institute in UK, have created the term Sociotechnical System to describe the interactions between two Systems: the technological and the social [26].

Ackoff and Trist [27] stated that Systems researchers treat humans as statistics-generating machines, or as entities that respond to stimuli in mechanical ways, and that, sometimes, human beings are simply excluded from the models. Sociotechnical Systems includes humans and technologic knowledge necessary to understand how a System should reach its major goal: include humans as part of the System, a System that generally is controlled by rules and policies from organizations. The human performance must be seen as embedded in a work environment shaped in subtle ways by technology and human behavior [28].

3.1.1 ICT Social and Technologic Infrastructure

ICTs Systems must be made considering human, social institutions and technology. It is a Sociotechnical System, where there is a social infrastructure (human and social institutions), and a technology infrastructure. The consideration of these two infrastructures is crucial in order to identify the correct factors for the quality of services and to identify the stakeholders' expectations, to give them the experience that they expects, surprising them whenever it is possible [2], [29], [30].

Electronic infrastructures (e-Infrastructures) are the basic resources used by ICTs. These resources are computers organized into networks, which together constitute a large computing and storage power, allowing that resources, facilities and services be provided to educational and researcher communities to conduct projects together, promoting, changing end preserving knowledge [31], [32].

The social impact of ICTs technology is present not only in academic context; it is changing the world society as well, as the e-infrastructure allows ICTs to create Systems in which communication and business operations are almost immediate. As a example, e-infrastructure connect global cities as Tokyo, London and New York - cities that are headquarters of world's greatest companies and of the most important stock exchanges, the technology closes a nearly uninterrupted daily round of global market shares.

3.1.2 ICT Complexity

Considering ICT e-infrastructure alone, a computer grid, for example, it is only a technological artifact. It has a purpose, meaning, only when one or more people use it to accomplish some task, as information search or data process to solve problems.

The technological, human and social components of an e-infrastructure system cannot be seen only as the sum of its components. There is a complex interaction among them and emergent properties; that emerges from the System as a whole, not specifically from any of its components.

Another issue that contributes to ICT System complexity is that many of the Systems used today were not develop in an integrated way. They were put together in a gradual way, resulting in a kind of patchwork, with new and old technologies, people and social institutions. New designs must respect this scenario, considering new and old technologies, and the several actors (as user, consumers and social institutions). These actors want to optimize their decisions, thinking in their own subsystems, proposes and interests [33].

3.2 Beyond human factors

Sociotechnical Systems have an intrinsic complexity, and the traditional engineering approach have difficulties in handling with it, both in human mapping (its values, intentions, etc.) as in social institutions purposes mapping, which often are seen only as part of the context, without belong directly to the System. The reductionism, a characteristic of the traditional approach ends up treating human and social dimensions as constants, or some times, ignore them.

The Soft Systems approach addresses the requirements to understand the problem domain of the ICT Systems and helps to identify the human and social dimensions. The former is because the activity to understand the problem domain is essentially an activity where the components are human activities, and the second because there is an intrinsic complexity of accurately identify human and social dimension during all the System life.

The approach to go beyond human factors is the use of Soft System Methods with an evolutionary approach strategy, as proposed by Soares [34]. The evolutionary approach (Fig. 1) deals with the interaction between reality and thought, and the interaction between problem and solution. Soares sets solution as overcoming restrictions or improvements in an existing reality through an action, considering solution as an indicative of an improvement, i.e., a response that satisfies, but does not solve the problem. From the two interactions exposed above, there are four actions that generate a cycle to resolve (not always solve) a problem. These actions are as follows:

- *Understanding*: When the engineer constructs an understanding, an abstract representation of a real problem;
- *Design*: When the engineer creates a response to the problem that satisfies the problem in thought dimension;
- *Implementation:* The construction of the response to the problem in terms of reality;
- *Use:* Set up of a response to the problem, in the environment of the problem.

The set up of a response to the problem may cause changes in reality, emerging scenarios not previously determined, giving rise to new demands, as there is a problem re-definition. The treatment sequence of the problems leads to an evolutionary spiral (Fig.1), which keeps track of the steps taken in identifying the human and social dimensions. This control is a fundamental tool in the requirement process [5], [34], [35].

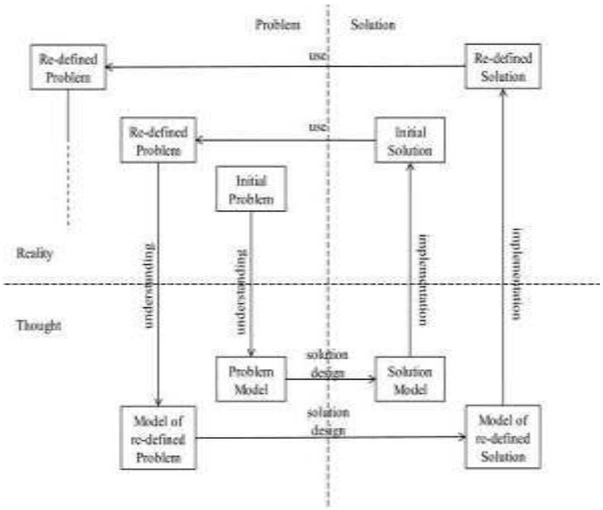

Fig.1 - Evolutionary spiral representation [34].

Although the identification of human and social dimension during all the System life is important to the System success during all the life cycle, the first step of the process is crucial.

4 The Beginning: Requirements Gathering

The human and social institutions mapping must be done at the beginning of the system life cycle, when the System Engineer is identifying the needs that the System should meet, the problem domain, i.e., the System Requirements.

Understanding the System Requirements is the first step to determine the System construction possibilities and the engineer must be very careful in this activity. If mistakes occurs in this early phase of the requirement process, the System will have to be adjust later, or will become obsolete before its time, or will be reject by users or, even, the System will fails in order to bring the benefits expected from it. Such mistakes also raise development and deployment costs and, several times, causes non-compliance with agreed deadlines [5], [36], [37].

The generic practices addressed by live cycles models may, or may not, be applied to an organization. Recommendations to adapt the activities described by these models to the situation where they will be applied are common, and depend on people decisions and judgments, that take many organizations to have their own approach; as this scenario can lead to a situation where each organization has its model, the Engineering has its standards.

The standards for Systems Engineering proclaim that the steps in a life cycle had to correspond to the progressive transitions in the principal System Engineering activities and be capable of being mapped into the principals' life cycle models in use by the System Engineering community. A life cycle model that serves as a framework to this work is the one proposed by the international standard ISO/IEC 15288: Systems Engineering – System Life Cycle Processes [38].

INCOSE Systems Engineering Handbook [4] states an analysis of the System life cycle process per ISO/IEC 15288, showing the process of the life cycle that are inputs to others. In this analysis, the processes: Requirements Analysis, Architectural Design, Implementation, Verification, Validation, Operation, Maintenance, Disposal, Decision-making, Risk Management, Configuration Management, Information Management and Quality Management are dependents from the Requirements Definition Process, that is the first process in life cycle process per ISO/IEC 15288.

A proposal to reduce users' dissatisfaction, respecting the human aspects and getting the necessary information to System development is the use of Consensual Methods to get consensus about the Systems Requirements from all the people that have interests in the system. The consensual processes deal with the human activities involved in identifying the requirements and the human and social dimensions, seeking to reduce the discrepancy between the expected Systems features and the ones that will be perceived by the users. The Requirement Elicitation needs to go beyond the human factors that the Engineering usually indentify in this phase and then users can feel welcomed by the system.

5 Consensual Methods

Following are related the Consensual Methods used by the authors in their work. Hitchins [2] stated that these methods are specifically to the front end of the Systems methodology, they are: Brainstorming, Nominal Group Technique, Idea Writing, Warfield's Interpretive Structural Modeling, Checkland's Soft Systems Methodology, Hitchins' Rigorous Soft Method.

5.1 Brainstorming

This is an approach in which a selected group of people is encouraged by a moderator to come up with ideas in response to a topic or a trigger question. [2], [39].

5.2 Nominal Group Technique (NGT)

This technique is similar to Brainstorming. A moderator introduces a problematic situation to a group of people and asks to participants to write down their ideas about the problem on a sheet of paper. After a suitable delay for the people generate their ideas, all participants read their ideas and the moderator, or an assistant, write them in a flip chart. With all the ideas written, the moderator conducts a discussion about these ideas, and then the participants are invited to rank all ideas. An idea-ordered list is generated and this constitutes the ideas that have been produced by the group as whole [2], [39].

5.3 Idea Writing

This method takes TGN a little farther. The moderator introduces the theme, and the participants are asked to write their ideas, suggestions, etc., in a paper. After two or three minutes, the moderator asks to which participant to pass his sheet for another person, pass the sheet to two people on the left, for example. Who receives the sheet can see the ideas already written, which may lead him (her) to a new set of ideas. After a short time, the moderator asked for the sheet recirculation, this time, a different number of people. The process is repeated for about 30 minutes, or until the moderator notes that most people do not have more ideas. There are two purposes in this strategy: encourage the ideas emergence within the working group and hide the origin of a particular idea. The lists of ideas are worked later through Brainstorming or TGN to generate an action plan [2]

5.4 Warfield's Interpretive Structural Modeling (ISM)

It is as a computer-assisted learning process that enables individuals or groups to map complex relationships between many elements, providing a fundamental understanding and the development of action courses to treat a problem. Hitchins [2] argues that ISM is essentially context free, and that computer support is not essential, because it can be executed using only a pen and a paper. An ISM session starts with a set of elements (entities) to which a relationship must be establish. These entities are identifying using other methods; Hitchins [2] suggests the use of the TGN. The result of ISM is a kind of graph, where the entities are nodes and the relations are edges. All process can consume a lot of time, especially when there are many divergences between the group members. Therefore, this time is important. It is essential for participants to understand and recognize the arguments of each other, reaching the consensus [40], [41].

5.5 Checkland's Soft Systems Methodology (SSM)

This method promotes the understanding of a problematic situation through the interaction between the people involved in the problematic situation. It promotes the agreement of the multiple problem views and multiple interests, and can be represented by a model of seven stages. The stages one and two explores the problematic situation (unstructured) and express it in a rich picture. Stage three are the root definition of the relevant systems describing six aspect of the problem, which are called CATWOE, they are: Customers, Actors, Transformation process, Worldview, Owner and Environment constrains. In stage four, the conceptual models of the relevant systems are developed and in stage 5 the conceptual model are compared with the perceptions of the real situation. In stage six, an action plan is developed to the changes that are feasible and desirable; and in stage seven, the action plan is implemented. As a method developed from the Soft Systems Thinking, SSM does not produce a final answer to the problematic situation, it seeks to understand the problem situation and find the best possible response [2], [16].

5.6 Hitchins' Rigorous Soft Method (RSM):

As SSM, this method is based around the General-Purpose Problem-solving Paradigm and is context free. The people who are experiencing a problem, and have knowledge about it, provide information about it in meetings with the Systems Engineer. This investigation, which search for dysfunction sources related to the problem, can create a lot of information and data. In a way different from the SSM, RSM employs tools and methods for treating, organize and process information, where the action of "process" implies in a gradual reduction of the problematic situation by ordering the data, transforming them into information for the treatment of the problem. RSM has seven steps [2]:

1) *Nominate Issue & Issue domain*, where the problem issue are identified and a description of the situation are made;
2) *Identify Issue Symptoms & Factors*, that identify the symptoms of the problem, and the factors that make them significant to be explored;
3) *Generate implicit systems*, each symptom implies the existence of at least one implicit system in the problem situation;
4) *Group into Containing System*, in this step the implicit systems are aggregated to form clusters, one cluster for each symptom, named containing system, that can generate a hierarchy of systems, highlighting issues related to the problem;
5) *Understanding Containing Systems, interactions, imbalances*, in this step the interactions between the containing systems are evaluated;
6) *Propose Containing Systems Imbalance resolution*, this step use the differences between an ideal world, where the symptoms does not exist, and the real world, to propose sociotechnical solutions to the imbalances identified in the previous step;
7) *Verify proposal against original symptoms*, in this step the system model are tested to see if they would, if implemented, eliminate the symptoms' identified in step two and the imbalance found in step six. This model could also be tested for cultural acceptability by the people that are experiencing the problem.

6 Perspectives of Method Selection

The diversity of people involved in an ICT system development is a reality that Engineering must deal. Zhang [42] states that it is impractical to limit the diversity of people involved in a requirement process, and that, however, the methods to develop requirements are under the Engineer's control.

6.1 Perspectives

Kossiakoff and Sweet [3] stated that the function of system engineering is to guide the engineering of complex Systems, and that System Engineering is an inherent part of project management - the part that is concerned with guiding the Engineering effort itself. ICT Systems must take System Engineer approach as to deal with the System complexity increase.

Kossiakoff and Sweet [3] had structured a life cycle model that corresponds to significant transitions in Systems Engineering activities. They did it comparing three standards life cycle models: Department of defense Model (DoD 5000), International model ISO/IEC 15288 and National Society of Professional Engineers model (NSPE). This life cycle model, adopted as a life cycle framework to this work, has three brad stages, with eight distinct phases:

- *Concept Development Stage*: With the phases: Needs Analysis, Concept Exploration and Concept Definition;
- *Engineering Development Stage*: With the phases: Advanced Development, Engineering Design and Integration & Evaluation;
- *Post development*: With the phase: Production and Operation & Support

The five phases of the Concept Development stage (the Requirement Process) are shown in Table 1, with its main activities, purpose and inputs.

	Main Activity	Primary Purpose	Inputs
Advanced Development	Risk Abatement	Identification and reduction of development risks.	System functional specification and defined system concept
Engineering Design	Component Engineering	Ensuring that individual's components faithfully implements the functional and compatibility requirements.	System design specification and validated development model
Integration & Evaluation	System Integration	Ensure that all interfaces are fit and components interactions are compatible with functional requirements.	Test & Evaluation Plan and Engineered Prototype
Production	Production Process	Diagnose the source of problems and find effective solution.	Production specification and production systems
Operation & Support	Logistic Support System	Continuous training programs for operators and maintenance personnel.	Operation & Maintenance documents and installed operational system

Table 1: List of life cycle phases after the Concept Development stage.

6.2 Method Selection

The use of Consensus Methods to get consensus from the people about the Systems requirements are the proposal to reduce people dissatisfaction about Requirement Process, respecting the human aspects and getting the necessary information to development a System. However, to be adherent to the System life cycle process, these Consensual Methods also need to have outputs to the process that are dependents of the requirement definition process.

The comparison of Consensual Methods outputs considering the life cycle phases stated by Kossiakoff and Sweet [3] that follow the Concept Development stage is the technique proposed to choose the Consensual Method, or Methods, which better provides information to subsequent life cycle phases.

6.3 Table of Method Selection

To understand the appropriate use of the different Consensual Methods, the authors have constructed a table (Table 2) that summarize the adherence level of a Consensual Method to life cycle model phases demands proposed by Kossiakoff and Sweet [3]. The authors experience in dealing with these methods form the situational context, which is listed in the first cell of the table left column. The Consensual Methods are listed in the top row, and represents different ways to gathering the requirements.

+++: Method recognize the phase issues and provide means to deal with it; ++: Method support the phase issues but not as strongly as the previous one; +: Method address the phase necessity but weak or indirectly; -: Method does not address the phase issues.	Brainstorming	Nominal Group Technique (NGT)	Idea Writing	Interpretive Structural Modeling (ISM)	Soft Systems Methodology (SSM)	Rigorous Soft Method (RSM)
Advanced Development	+++	+++	+++	+++	+++	+++
Engineering Design	++	++	++	+++	++	+++
Integration & Evaluation	-	-	-	+	++	+++
Production	++	++	++	-	++	+++
Operation & Support	-	-	-	+	+	+++

Table 2: Table of Method Selection.

|The Table of Method Selection is illustrative, rather the comprehensive. It is based in empirical findings from authors' experience. It provides a practical starting point for organizing an approach to the requirements elicitation, or, as posted by Kossiakoff and Sweet [3], the Needs Analysis and Concept Exploration phases.

7 Final Considerations

From the perspective of comparing the Consensual Methods taking in consideration the life cycle phases, presented in table 2, the method that provides more information for the life cycle phases that that follows the requirements elicitation is Hitchins' Rigorous Soft Method (RSM). As a Consensual Method, it promotes consensus among people about the System requirements, in such a way that people feel welcomed by the process

Humans being have personality, hopes, fears, dreams, values, and intentions. Do not consider these human dimensions to build systems ultimately dehumanize human-system interaction, and is costly!

Authors work seeks to validate the approach proposed, using RSM to considering humans and social institution during all system life cycle and, at this moment, the work is been applied to the followings authors' projects:

- *Soft Approach and Engineering Standards*. The near absence of reflection on humanism in system life cycle, leads to a system development with focus much more on functionality and usability than in humanities and social interfaces. Nevertheless, this reflection on human dimensions must not exist by itself; it must be supported in engineering standards like IEEE and ISO/IEC ones. The authors are working s in correlating the RSM with the life cycle standards ISO/IEC 15288 and IEEE 1220;
- *e-Infrastructure as sociotechnical systems*. KNOMA (the authors' laboratory at Escola Politécnica-USP) is a partner of the BELIEF-II Project, from Seventh Framework Program (FP7), and work in a key issue of the e-Infrastructure: The regard with humans, social institutions and technology. The sociotechnical view is crucial in order to identify the correct quality factors and the expectations of actors of the social infrastructure, to give the experience that these actors expect, surprising them whenever it is possible. A special topic in this project is the use of RSM to address e-Infrastructure projects at Amazon, respecting

the regional characteristics, and human and cultural dimensions;

- *ALCUE UNIT Model*. KNOMA is a partner of VertebrALCUE project, from ALFA III Program. One of the main project activities to its partners is to build an ALCUE UNIT; this is a key activity to build a cooperation infrastructure between high education entities. KNOMA ALCUE UNIT has a thematic focus in modeling e-Infrastructure as sociotechnical system, and building a network that will allow information exchange on mobility of teacher, students, and researchers interested on e-Infrastructure as sociotechnical systems subject. To build this information network, RSM is used and e-Infrastructure concept is applied, interconnecting computing resources that will permit information dissemination.

Acknowledgments
The research and scholarships are partially found by:

- BELIEF-II (**Bringing Europe's eLectronic Infrastructures to Expanding Frontiers - Phase II** http://www.beliefproject.org) is an EU FP7 project with the aim of supporting the goals of e-Infrastructure projects to maximize synergies in specific application areas between research**,** scientific and industrial communities.
- VertebrALCUE: The **VertebrALCUE** Project (http://www.vertebralcue.org) is an ALFA III Program project that aims to contribute to the development process of the regional integration among Latin American Higher Education Systems (HES´s) and the implementing process of the Common Area of Higher Education between Latin America, the Caribbean and the European Union (ALCUE in Spanish) by exploring and strengthening different levels of articulation of Latin America-Latin America and EU-Latin America academic cooperation through the design and implementation of a cooperation infrastructure at institutional, national and regional level.

References:

[1] Skyttner, L.: *General systems theory: Problems, Perspectives, Practice*, World Scientific Publishing Company, 2nd edition, New Jersey, 2005.

[2] Hitchins, D. K., *Systems Engineering: A 21st Century Systems Methodology*. John Wiley & Sons, Chichester, 2008.

[3] Kossiakoff, A. and Sweet, W. N., *Systems Engineering Principles and Practice*. John Wiley & Sons Inc., New Jersey, 2002.

[4] INCOSE: Systems Engineering Handbook Development Team of the International Council on Systems Engineering (INCOSE). *Systems Engineering Handbook*, V. 3. INCOSE-TP-2003-002-03, 2006.

[5] Robertson, S. and Robertson, J., *Mastering the Requirements Process*, Pearson Education, Inc., Boston, 2006.

[6] Boehm, B. W. and Papaccio, P. N.: Understanding and Controlling Software Costs, *IEEE Transactions on Software Engineering*, Vol. 14, No. 10, Oct. 1988, pp. 1462-1477, 1988.

[7] Firesmith, G. D.: Common Requirements Problems, Their Negative Consequences, and Industry Best Practices to Help Solve Them. In: *Journal of Object Technology*, vol. 6, no. 1, Jan-Feb. 2007, pp. 17-33, 2007.

[8] Chapanis, A., *Human Factors in Systems Engineering*. John Wiley & Sons, New York, 1996.

[9] Nemeth, C. P., *Human Factors Methods for Design*: Making Systems Human-centered, CRC Press, Boca Raton, 2004.

[10] Sandom, C., *Human Factors for Engineers*. Institution of Electrical Engineers, London, 2004.

[11] Jordan, P., *Design Pleasure Products: An Introduction to the New Human Factors*, CRC, Boca Raton, 2002.

[12] Ottens, M. and Stubkjaer, E., *A socio-technical analysis of the cadastral system*. In: ZEVENBERGEN, J., FRANK, A., STUBKJAER, E.: Real Property Transactions. Procedures, Transaction Costs and Models. IOP Press, Amsterdam, 2008.

[13] Ackoff, R. L., *Redesigning the Future: a Systems Approach to Societal Problems*. John Wiley and Sons, New York, 1974.

[14] Ackoff, R. L., Systems, Messes and Interactive Planning, In: Trist, E., Emery, F.; Murray, H.: *The Social Engagement of Social Science: A Tavistock Anthology: The Socio-Ecological Perspective*. University of Pennsylvania Press, Vol. 3, 1997, pp 417-438.

[15] Checkland, P., *Systems Thinking, Systems Practice*, John Wiley & Sons, London, 1981.

[16] Checkland, P., *Soft Systems Methodology: A Thirty Year Retrospective*. Systems Research and Behavioral Science Syst. Res. 17, S11–S58, 2000.

[17] Klir, G. J., *Facts of systems science*. New York: Springer, 2nd edition, 2001.

[18] Mac, K., Adams G. and Mun J. H., The Application of Systems Thinking and Systems Theory to Systems Engineering, In: *Proceedings of the 26th National ASEM Conference: Organizational Transformation:*

Opportunities and Challenges. American Society for Engineering Management, Rolla, MO, USA, 2005, pp. 493-500.

[19] Kralj, D., Systems Thinking and Modern Green Trends, *WSEAS Transactions on Environment and Development*, Vol. 5, No. 6, 2009, pp.415-424.

[20] Ackoff, R. L., *Creating the corporate future: plan or be planned for*, John Wiley and Sons, Nova York, 1981.

[21] Checkland, P., *Soft Systems Methodology: a 30-year retrospective*. In: Checkland, P., Scholes, J.: Soft Systems Methodology in Action. John Wiley & Sons, Chichester, 1990.

[22] Mazur, G. H. and Bolt, A., Jurassic QFD. In: *Transactions of the 11th Symposium on Quality Function Deployment*, June 12-18, QFD Institute, Michigan, 1999.

[23] Watson, G. H., Conti, T. and Kondo, Y., *Quality into the 21st Century: Perspectives on Quality and Competitiveness for Sustained Performance*. ASQ Quality Press, Milwaukee, 2003.

[24] The Kano Model, Available at: (*http://kanomodel.com*)

[25] Kumlander, D., On Software Design and Development Supporting Requirements Formulation, *Proceedings of the 10th WSEAS International Conference on COMPUTERS*, Vouliagmeni, Athens, Greece, July 13-15, 2006 pp818-825.

[26] Trist, E., *The social engagement of social science: The socio-technical systems perspective*, University of Pennsylvania Press, 1993.

[27] Ackoff, R. L., Emery, F.: *On Purposeful Systems: An Interdisciplinary Analysis of Individual and Social Behavior as a System of Purposeful Events*, Aldine Transactions, 2006.

[28] Gonzalez, J.J., Sawicka, A.: A Framework for Human Factors in Information Security, *WSEAS International Conference on Information Security*, 2002.

[29] Bryl, V., Giorgini, P. and Mylopoulos, J., Designing socio-technical systems: from stakeholder goals to social networks, *Requirements Engineering*, Springer, London, Vol. 14, 2009, No.1, pp. 47-70.

[30] Ottens, M., Franssen, M., Kroes, P. and Van De Poel, I., Modeling infrastructures as socio-technical systems. *International Journal of Critical Infrastructures*, Vol. 2, 2006, No. 2/3, pp.133- 145.

[31] Campolargo, M.: e-Infrastructure: Changing How Research is Done, *Communication Magazine, IEEE*, Vol.42, no. 11, pp. 31-32, nov, 2004.

[32] European Commission. EGEE-II, Project; BELIEF, Project; OMII-Europe, Project; DEISA, Project; Geant2, Project. *A guide to European flagship e-Infrastructure projects***.** e-Infrastructure Unit, 2007.

[33] Houwing, M.; Heijnen P.W. and Bouwmans, I. Socio-Technical Complexity in Energy Infrastructures - Conceptual Framework to Study the Impact of Domestic Level Energy Generation, Storage and Exchange, In: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. Taipei, Taiwan, October 8-11, 2006.

[34] Soares, J. O. P.: *Especificação de Métodos de Desenvolvimento de Sistemas - Aplicação a Sistemas de Tempo Real*. Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 1986.

[35] Kotonya, G. and Sommerville, I., *Requirements Engineering: Processes and Techniques*, J. Wiley, London, 1998.

[36] Sydenham, P. H., *System Approach to Engineering Design*, Artech House Publishers, Norwood, 2003.

[37] Wasson, C., *System analysis, design, and development: concepts, principles, and practices*, John Wiley & Sons, New Jersey, 2006.

[38] ISO/IEC 15288, *System engineering – system life cycle processes*, International Organization/International Electrotechnical Commission, Geneva, 2002.

[39] Tague, N., *The Quality Toolbox*, ASQ Quality Press, 2005.

[40] Warfield, J. N.: *Societal systems: planning, policy, and complexity*. New York: John Wiley & Sons, 1976.

[41] Wright, J. T. C. Análise e estruturação de modelos baseada em inferência lógica: objetivos para o Porto de Santos. *Revista de Administração da Universidade de São Paulo*, São Paulo, vol. 30, no. 1, jan./mar. 1995.

[42] Zhang, Z.: *Effective Requirements Development. A Comparison of Requirements Elicitation Techniques*, INSPIRE 2007, Tampere, Finland (2007).