Toward a Systemic Notion of Methodology:

Practical Consequences and Pragmatic Importance of Including a Trivium and the Respective Ethos, Pathos, and Logos

Nagib Callaos & Bekis Callaos
Universidad Simón Bolívar and
The International Institute of Informatics and Systemics (www.iiis.org)

Unedited version (July 11, 2014) posted on the web in order to collect constructive feedback and criticism that will be addressed in the final version of this draft. Consequently no text of this draft is allowed to be cited without explicit assent of the author. Enquiries and Comment might be sent to ncallaos at att.net

Abstract

In spite of the increasing use of the term "*methodology*", its concept is frequently confused with that of "*method*", and, sometimes, it is hardly differentiated from that of "*technique*" or "*tool*". This situation of conceptual confusion is generating effects of undesirable practical incidence. Thus, a defining process of "*methodology*" and its related concepts could be of theoretical and practical interest. The objective of this paper is to make a first step in said defining process. Accordingly, we will try: (1) to identify different definitions of method and methodology, (2) to differentiate them and to relate them, between themselves, and with the concepts of "tool" and "technique"; and (3) by means of the definitions worked out, to differentiate between closed-efficient-systematic and opened-effective-systemic methodologies.[1] We will differentiate between systemic and systematic methodologies and show the practical usefulness of this kind of distinction. We will also show that a systemic methodology necessarily contains cybernetic relationships between thinking/informing/knowing and doing, in the context of implicit or explicit meta-methodological processes of action-research, action-learning, and action-design. This kind of relationship requires 1) collaboration and effective human communication and, hence, 2) an updated suitable *Trivium*, which includes adequate *Ethos*, *Pathos*, and *Logos*.

Keywords: Methodology, Tool, Method, Technique, Definition, Systemic, Openness, Systemic, Systematic, Trivium, Ethos, Pathos, Logos.

Introduction

Systems research is being increasingly concerned with systemic methodologies, and information systems engineering with development methodologies. Many methodologies, which can legitimately claim to be based on system approaches, are being designed and implemented. In the area of information systems development, a bewildering profusion of "methodologies" is disconcerting managers of information/communication technologies,

[1] Some texts of the initial sections of this paper are 1) updated adaptations from other publications or 2) digests from sections or chapters of larger works made by the same authors (Callaos and Callaos, 1991; Callaos 1993; Callaos, 1995; Callaos, 2013, Callaos and Callaos, 2013)

and the frequent failures in the implementation of related systems are perplexing them. Furthermore, "methodologies for methodology choice" started appearing a long time ago (Keys, 1988), and multi-methodological approaches has been proposed, through which different possible methodologies are combined and mixed in practice according to the nature of the specific system being considered or developed as well as to the nature of its specific context (see, for example, in the area of management science methodologies, Mingers and Gill, 1997). More recent papers are still referring to the problem of selecting an adequate methodology in information systems development (see, for example, Center for Medicare and Medicaid Services, 2008, Office of Information Services)

Paradoxically, little attention is being paid to the meaning of the word "methodology" in systems sciences and in information systems engineering. Few authors refer to what might be, or ought to be, the concept of methodology. Baldwin (1960) asserted that “we can appeal to few works as showing what methodology ought to be.” This situation did not change since then~~,~~ according to the information we have. Most of the definitions of "methodology" that could be found in systems literature do not go too far from the definition given in general dictionaries. And, in the area of information systems, the concept of "methodology" is frequently confused with the concept of "method". Even worse, its meaning is intermingled with that of "technique", "instrument", and "tool".

Explicitly or implicitly, several authors equate, or confuse, the notion of “methodology” with “method~~s~~”. Cronholm and Ågerfalk (2004) assert that “the term ‘methodology’ is often used when what is actually referred to is ‘method’.” They also provide several examples of authors using both terms as synonyms. Among these authors are Jayaratna (1994), Stamper (1988) and Checkland (1981). Brinkkemper (1996) states, according Cronholm and Ågerfalk (2004), that “the misuse of the term methodology standing for method is a sign of the immaturity of our field, and should consequently be abandoned.”

Very few extended reflections on the concept of methodology could be found in the literature of Systems Science or Engineering, or in the area of information systems. Baldwin (1960) affirms that, "We can appeal to few works as showing what methodology ought to be." This lack of concern with regards to the notion of methodology and the conceptual confusion found in the literature where the word is being used, are having, in our opinion, *undesirable effects in practice*. In the information systems area, for example, Computer Assisted Software Engineering (CASE), as well as any meta-software, is, as we will see, a tool, not a methodology as it is frequently referred to by some of the respective products’ vendors. It is our experience that in those cases where a CASE was acquired, and it was thought that a methodology was being provided, the CASE's insertion in the information systems development was a failure. Another example was the situation where the CASE was consciously acquired as a *tool* that would support some steps of the methodology being used, or to be used, by the information systems developers. This kind of practical usefulness derived from conceptual clarity and distinctiveness is not limited to the information systems area. Adequate system practice requires clear system thinking, and this needs conceptual definitions or adequate reflections on the associated notions[2]. Hence, to define "methodology" and its related concepts is of practical usefulness, in addition to its theoretical importance. Indeed, the activity of defining is at the core of scientific research (Bridgman, 1927; Stevens, 1935; Bridgman, 1938; Ackoff, 1962) in spite of the fact that, as Ackoff points out**,** "too few scientist take it very seriously", and "the meaning of concepts are too often taken for granted" (Ackoff, 1962, p.174).

"Methodology" seems to be a concept which meaning is taken for granted. From the perspective of Husserl's phenomenology, "methodology" has been an *"operational"* concept not a "*thematic"* one. Operational concepts are those that are frequently used to clarify other concepts, but they, in turn, are not clarified. They are "shadow concepts" that stand behind the "being-clarified thematic concepts", supporting the clarification process while maintaining them un-clarified, or "in the shadow" (Fink, 1968). "Methodology" has been, up to the present, an "*operational concept*" used frequently to make possible the operations of thought while clarifying "*thematic concepts*. In this sense, the concept of methodology has been useful by maintaining itself "in the shadow" in order to clarify other concepts. It is our opinion, that the time is ripe for the concept of "methodology" to jump out the "shadows" and get clarified, or "thematized". To start the "thematizing" process means to initiate a defining process or description of the related notion. This is the initial objective of this paper. A following objective will be to show the practical implications of defining "methodology" and having a clearer notion of it and its related concepts.

Systemic Definition

Elsewhere (Callaos, 1991, Callaos, 1995 pp. 35-100), in a meta-defining process, we identified more than 30 different definitions of "definition", and concluded that a systemic definition should be done in a comprehensive way, i.e. in such a way as to include most alternative definitions, or several meanings of the term "definition". Consequently, it is advisable to be explicit about the meaning with which we are using the word "definition" in this paper. The systemic definition, we are attempting here, would have the following characteristic:

1. From the epistemological perspective, a systemic definition is oriented toward the *pragmatic-teleological* truth of Singer-Churchman (Churchman, 1971). This will be achieved by means of:

    1.1. Taking into account the "*telos*", or "*the purposes of the definer*", as Ackoff stressed it (Ackoff, 1962). We have already summarized these purposes above.

    1.2. Relating the definition to past and present usage of the word in order to serve *pragmatic communicational needs* (Ackoff, 1962).

[2] For a distinction between "concept" and "notion" see, for example, Callaos, 2013.

1.3. Making the definition *operational* (Ackoff, 1962; Bridgman, 1927; 1938, Stevens, 1935) in order to be useful in a pragmatic context. We will address the operational issue from a qualitative perspective, not by the definition of quantitative measures. Although quantitative measures can be formulated, it is not the objective of this paper to do so.

2. From the methodological perspective, the variety of past and present usages of the word and the different ways in which it was defined should be structured by means of a logical infra-structure, or by means of a bootstrapping process (Alvarez de Lorenzana, 1987). In this way the definition will be *comprehensive*, *open* and *adaptive*, both as a product and as a process, and we will have the bases that could support a progressive "spiraling" process according to an *Evolutionary* Paradigm (Alvarez de Lorenzana, 1987; Laszlo, 1987).

Etymological Meaning for a Tentative Definition

Ackoff stressed the fact by which "historical analysis of the use of a concept can often reveal a trend in the evolution of the concept or a consistent theme of meaning which persist through numerous variations" (1962; p.148). This is why he exhorts to initiate a scientific defining process by formulating a tentative definition based on the evolving core identified by historical analysis. It is our experience that Ackoff's approach on this issue is an intellectually valuable and praxiologically practical one. It is also our experience that taking this approach to its extreme, by going to the etymological meaning of the word being defined, is also intellectually helpful because it might suggest a pre-tentative definition. The *suggestive* effect of historical linguistic analysis had been stressed by several authors (Navarte, 1981; p.158). Being the root of following meanings, the etymological definition frequently suggest a general concept from which more specific ones were generated through the uses of the associated term through its history. This is why we think that the etymological source may help us into abstracting a general definition from the varieties of the specific ones that appeared through history. And this is why we think, and our experience has been, that the etymological meaning of a word is intellectually useful for capturing the systemic essence of the meaning of a word and/or support a tentative pre-conception of the concept being defined.

The Notion of "Method"

As we said above, authors of books related to methods in systems engineering, as well as in other engineering fields, present and describe methods (sometimes very complex ones) without paying attention to the nature and concept of "method". Even in Science and Philosophy, when authors' subject is related to the notion of method, they usually treat the subject of method(s) *in concreto*, specific method(s) or methodologies(s) in a given area. Their subject is usually "scientific methods", "social sciences methods", "philosophical methods", "experimental methods", etc.; without any need being felt to raise the basic question of the nature of "method" *quâ* method. Surprisingly we could not find an entry to "method" in the latest edition of the *Routledge Encyclopedia of Philosophy* (Craig, 1998), nor could we find an entry for “method” in the *The Stanford Encyclopedia of Philosophy* (The Metaphysics Research Lab, 2013). In the specific area of Systems Engineering or Systems Methodologies, we find even a more surprising and perplexing situation. In books in the areas of Systems Engineering, Systems Methodologies, Information Systems Developments Methodologies, Software Developments Methods, etc., it is not easy to find an entry of the word “method” or “method definition” in their subject index. The very few times when we find some definition of the term, it usually goes no further than a dictionary’s definition, not even an encyclopedic one. The meaning of method is frequently taken for granted and is used as if it were some kind of a primitive. In our opinion, this is one of the most important causes of the conceptual confusion found with regards to the term and one of the reasons of the rate of failures found in the application and deployment of methodologies in the areas of systems design and implementation, especially in Information Systems and Software Engineering.

The good news, however, is the emergence of some initial awareness, though a weak one yet, with regards to this issue. For example, as we indicated above, Cronholm and Ågerfalk (2004) are already alerting in this regards. They assert that “the meaning of the concept of method [in the area of Information Systems Development] is not always clear…there are different concepts (and terms) used for the same phenomenon and also the same concept (and term) for different phenomena.” This confusion of synonyms and homonyms is very dangerous, not just in the intellectual and epistemological domains, but also in the praxiological one, as we found out, over and over again through our direct and indirect experience of 30 years and the development of more than 150 information systems. Conceptual confusion with regards to the notion of method, and its related notions might represent, and actually do represent in most cases, the difference between success and failure in projects in the area of systems design and implementation, including information, informing, and knowledge systems. Applying a combination of *reflective practice* (Schön, 1983), *reflective methodology* (Alvesson and KajSköldberg, 2001) and *reflexive research* (Etherington, 2004) in the context of our experience we arrived to the *intellectual certainty* and *practical necessity* that more clarity was needed regarding the notion of “method,” which, in turn, might support the initiation of a process oriented toward a systemic definition of the concepts of “method” and “methodology,” and their related concepts.

We did that following the mentioned above Ackoff’s (1962) approach to scientific definitions, especially with regards to a historical review of the different ways in which the term has been defined and the different uses that come from it. Elsewhere, (Callaos, 1995; pp. 101-234) we made a comprehensive and detailed historical review of the uses of the term “method”, the several definitions made about it, and the diverse perspectives with which the concept had been approached. We made, then, several conclusions about the meaning of the term and the diversity found in it. We also attempted to find a coherent platform supporting the different senses in the meaning of the term, and the possible common threads among them. The good news is that one of the common threads we found then is strongly associated with the etymological meaning of the term. This etymological root of the term is the one with the most common sense we, explicitly or implicitly, found in the different uses of the term “Method”. It is one of the most important—if not the most—ingredient of its conceptual core. So, let us describe briefly this etymological meaning, and sketch the historical origins of the concept, in order to 1) make some important conclusions, 2) differentiate the concept from others with which it is usually confused, 3) increase the precision of its semantic frontier, and 4) identify some of its most important characteristics.

The term "methodology" is a composite from "method" and "logos"; thus, it means "science of method" or "theory of method". In turn, the word "method" is derived from the greek " μέθοδος" (*methodos*) which is a composite of "μετά" (*meta*) and "οδός" (*odos*) (Weekley, E., 1967). "οδός" (*odos*) has three different meanings: (1) way, road, street, path, route, track, highway; (2) trip, journey, voyage; (3) procedure, means, way of doing something. "μετά" (*meta*) means: between, medium, interim, intermediate (Sebastian, 1972). On the other hand, "Way" means the course taken or to be taken, in ***getting from one place to another*** (Watson, 1972); i.e. a way mediated between where I am and where I want to be, between a present situation and a desired future one, between establishing an objective and achieving it. Thus, we can easily infer that, from its etymological origin, “method” is:

1. An established ***way*** *of achieving an* ***end***, an established ***means that mediate between a given initial state and final one****.*

2. An ***action*** that achieves an end, an action that moves a person from an initial state to a final one.

Consequently, from its etymological, we can infer two basic senses in the meaning of “method”: as ***a-temporal*** (non temporal) and a ***temporal*** order, as a ***logical*** and a ***chronological*** sequence, as an established way to achieve an end and the action taken to achieve this end by means of this way, also as a *product* and a *process*. Socrates used the word “method” to denote a way of *doing* or *teaching* a technique. Plato and Aristotle used “method” to indicate a way of *knowing*. Socrates associated “method” with the achievement of some *useful* end, while Plato and Aristotle related “method” to the way of reaching the *truth* (Callaos, 1995; pp. 101-2). Aristotle used the word "method" in an analogical mode; he used it to mean an "intellectual way", a "thinking way". Since then, "method" has almost exclusively been used to mean "a thinking way to get true knowledge", and, as such its meaning was reduced to that related to the logic of science or philosophy. It was even conceived as forming an integral part of Logic. The "*Logique de Port-Royal*” (Part IV, Ch.2), for example, defined method as "the art of ordering adequately a series of diverse thoughts for discovering a truth or for proving a known truth" (Ferrater-Mora, 1980). However, with the increased importance of the System Approach which is pragmatically and teleologicaly oriented, and with the emergence of initial studies on praxeology (Gasparsky and Pszczolowski, 1983; Kotarbinski, T., 1965; Skirbekk (ed.), 1983), the word "method" is increasingly being used to mean a way of “*doing*". Thus, it could be concluded that method is presently being used in four fundamental senses: (1) a way of thinking, (2) a way of doing, (3) thinking according to a way, and (4) doing according to a way; by way, it is meant: a "*from-to*" medium. Briefly, ***method is a pre-established and/or post-established way for or a process of acting and/or thinking***. As it could be noticed, these definitions implicitly include eight (2x2x2, figure 1) different, though related, senses of the word “method.” Explicitly or (mostly) implicitly the word “method” is used in one if these eight senses or used as a confusing amalgam of different senses. The context in which the word is used to provide the specific sense in which is used, or misused in a confusing way with different senses, i.e. as a homonym referring to different things or phenomena. We have frequently found out this kind of confusion in both practice and theoretical or methodological articles or books. It is not the same to have a means or to be able to use it for the actual process of using it. It is not the same an established method, for people to be trained in its use than a way or a means which is generated by an intuitive process of “trial and error” or by serendipity. Methods and meta-methods should be distinguished, as well as methodology and meta-methodology. Below we will alert with regards to other distinctions which confusions have a negative impact in both the intellectual and the practical domains.

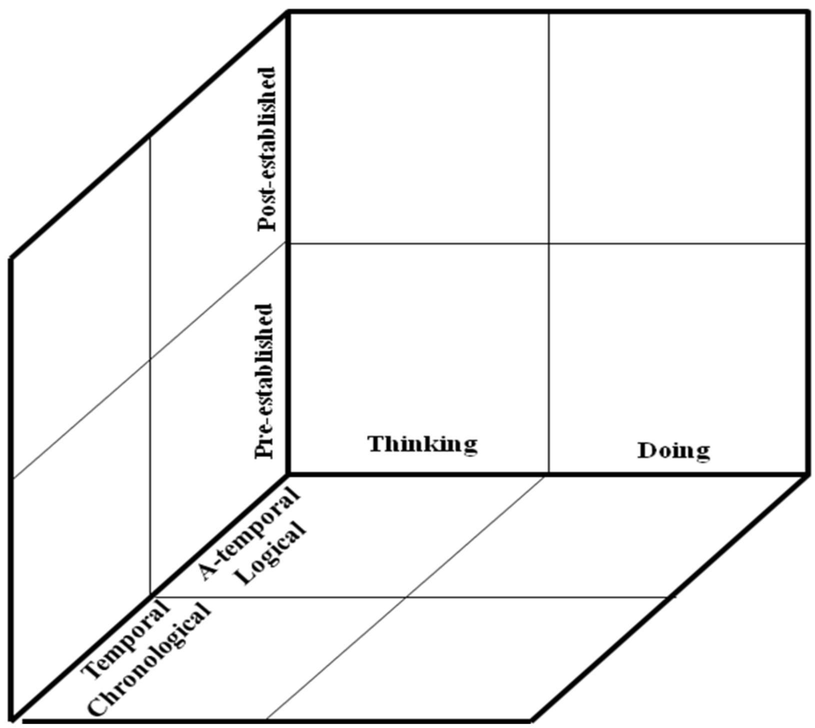


Figure 1

For about 2500 years the notion of method has been mostly analyzed (with some scattered exceptions to be found, especially, in the Renaissance) in an epistemological context, not a praxiological one. Meanwhile, action and production methods have continued to be created and used, but they were seldom a subject of intellectual or theoretical inquiries, and studies; essays and treatises on the notion of method had mostly been generated in an epistemological context. However, with the System Movement and by means of the Systems Approach, the notion of method is again being considered, defined and examined in the technical, praxiological, and educational domains. Beside the classical inductive/deductive, abductive and analytic/synthetic methods, used and examined traditionally in Logic, Science and Philosophy, methods in Production, Systems

Engineering Design, Organizational Re-Engineering, Information Systems Development, Decision Making Support, Software Engineering, Operation Research, Organizational Cybernetics, etc. have been emerging, as well as meta-methodologies (methodologies for methodical design), multi-methodologies, and methodological theories. This add to the importance of clarifying, with some details, the *meaning* of "method", the *analysis* of the notions of Method and Methodology and the *synthesis* of specific methods and methodologies.

From the etymological meaning of the term, we concluded that a method is a *way*, a "*road*", for thinking and/or doing, as well as the *process* of this thinking and/or acting. It is both a *pre-established written* (and/or visually represented) procedure and the *action* executed according the procedure. In any sense of the word, a method should be characterized by its ***initial*** and ***final*** points, by necessary initial conditions and sought one(s). So, a method always is a "***method-from***" and a "***method-to***". A *"from-to" indication should be included in any essential description of a specific method*. This is not always done *explicitly* when authors describe methods. Practitioners are certainly not always conscious of this fact when applying a method, but it is *implicit* in their mind and both (the "from" and the "to") might change according to the information and the learning produced by the respective action required to execute the method.

Pragmatic/Praxiological Importance of Explicitly Clarifying the Notion of Method

Not considering the pragmatic and praxiological importance of being clear and explicit regarding the "from-to" essential nature of any method might have detrimental consequences in practice, with highly undesirable potential consequences. In information systems development, for example, *it is not unusual to find practitioners working with a "top-down" method when they have no access to the "top" or there is no such "top" because no user has a global vision of the system. In these cases the practitioner is at the "bottom" and working with a "top-down" method, which is a contradiction and a non-sense.* We have seen this kind of contradiction in professional practice with an alarming frequency, especially when the top-down methods were the methodical fad of the moment. This kind of lack of clarity and conceptual confusions had produced, according to our consulting experience, huge amount of waste in person-powers and financial and managerial resources. In many occasions the opportunity costs of not meeting the deadlines were huge ones. Both efficiency and effectiveness are significantly decreased in projects applying a top-down method or methodology when the initial situation requires a bottom-up method or methodology. Some information systems projects even had zero efficiency and effectiveness because all that had been done in the respective project had to be thrown away to the trash.

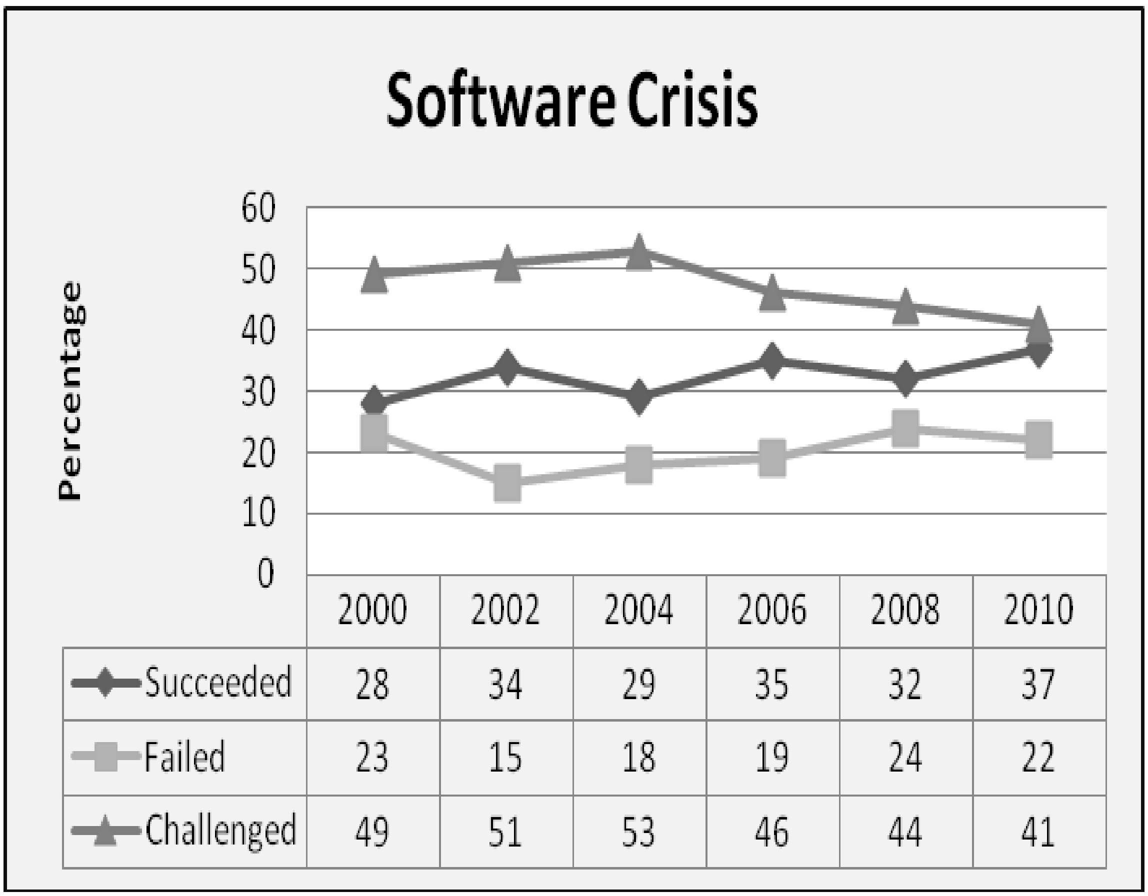
USA and international statistics coincide with our experience. There was no significant improvement in software development since the term "software crisis" was coined in the first NATO Software Engineering Conference in 1968. The observed lack of efficiency and effectiveness in software development has been shown by the Standish Group in the last 18 years. The Standish Group's studies had included more than 70.000 projects in information technologies. Figure 2 resumes the most general results during the 2000-2008 time period. The source of these data is the Standish Group's Chaos Summary for 2010.

In average:

- 32.5 % of the projects were considered successful (i.e. delivered on time, on budget, and with the required features and functionality) or which effectiveness and efficiency was as expected.

- 20.16667% of the projects failed (i.e. they were cancelled prior to completion or delivered and never used, i.e. having no effectiveness and no economic efficiency.

- 47.33333 of the projects were considered challenged (i.e. that were late, over budget, and/or with less than the required features and functionality), i.e. having low effectiveness and/or low efficiency.



	2000	2002	2004	2006	2008	2010
Succeeded	28	34	29	35	32	37
Failed	23	15	18	19	24	22
Challenged	49	51	53	46	44	41

Source: The Standish Group: Chaos Summary for 2010, p.3 (years 2000 – 2008)
and The Standish Group as reported by Ritu and Gill, 2012, p. 601 (year 2010)

Figure 2

A research made by IBM (2012) identified five areas influencing project success or failure

- Project management (54%): Activities defining and controlling the IT project
- Business (21%): Aspects of the project dealing with project funding, internal rate of return and business data
- People (14%): The team that carries out the IT project
- Method (8%): The dimension involving approach, procedures and tools (*notice the con-fusion of methods with other notions, which are related but not the same as the notion of method*)
- Technical (3%): Aspects of the project regarding hardware and software, testing and interfaces between components

Consequently, someone who is aware of this study might conclude that methods have no significant influence on determining the failure or the success of a software project. This might be a misleading conclusion because:

1. The study associates (or con-fuses) the notion of "method" with "Approach", "tool" and "procedure". Procedures are methods, but methods cannot be reduced to procedures. "Tool," as we will see below is not "method;" it is what moves us, what displaces us through a given method, what carries us ***from*** the present situation ***to*** the desired one. But to correctly identify where we are, where we want to go, and what the most adequate carrier are methodical decisions that have nothing to do with the notion of "tool". And approach is such a general term that might include many things (as intellectual or theoretical perspectives) that might have nothing to do with the notion of "method."

2. As we just mentioned above and as we will see below an adequate use of a method include decisions, which have to be taken by software engineers, systems engineers, project managers, and probably by stakeholders and other business people in charge of different functions or responsibilities in the organization. Consequently, the incidence of the real notion of "method" regarding the failure or success of a software development cannot be reduced to what IBM includes in this notion. For example the methods used depend on the methodology selected and on the way it is being managed including the kind of project management being followed. Project management alone has a 54% of incidence in the success or failure of a software development project. Then, according to the same IBM study, method and methodologies might have a 62% incidence of the success or failure of a software development project. According to our experience this 62% of incidence is a minimum, especially if we are developing information systems tailored to the changing needs and requirements of an organization or sub-organization.

Let us now contrast the above statistics with the empirical evidence we had in our experience for about 30 years, 23 of which included managing or directing about 150 projects of tailored information systems development, installation, deployment and maintenance in the largest Venezuelan corporation, where we applied a methodology initially designed along our academic activities and later modified and adapted to different situations. The design of this methodology was based on a clear and explicit definition of the terms used, and was described along with its applicability in a 50.000-words book. In this paper we are trying to provide a summary related to clarifying the meaning of "method" and "methodology", which was an important part of what we consider a proven success for about 150 projects of software engineering in the context of tailored information systems development.

We think that one of the best empirical evidence in tailored information systems development should be related to legally fulfilled contracts that a developing organization or a consulting firm can show. At Callaos and Associates, based in Venezuela, we have achieved the following results thanks to a methodology based on explicit clarity on the notions and concepts involved, the most important of which are "method" and "methodology". We can resume the results achieved during 23 years of software-based tailored information systems development and maintenance as follows.

1. Clients were charged, according to the respective written contracts, a cost equivalent to the lowest one in the World, which is the software development in India ($ 125 per function point), according to Capers Jones' "*Software Productivity and Quality Today: The Worldwide Perspective.*" This means that we have a methodology that proved to allow us to develop, support and maintain software at ***the highest economical efficiency*** in the world. According to Capers Jones, these costs, in Venezuela, are $ 190 per function point, which means that Venezuelan economical efficiency is, in average, 52% lower than our legally proven costs.

2. We delivered, and supported/maintained software with a ***quality equivalent to the USA average***, according to Capers Jones Statistics, and the MIT study in this area.

3. Our ***temporal efficiency*** has been 30% larger than the one corresponding to the USA productivity average, i.e. our delivery time was, and is, 30% lower than USA's productivity average. This is according to the very well known Boehms' equations.

4. Our ***productivity has been 40% more that the USA average***, according to Boehms' equations and about 50% more according to Capers Jones statistics.

5. We charge our clients just for the lines of codes that get in use. This way, our clients have a ***100% effectiveness*** *guarantee*, if we measure effectiveness as the relation among lines of codes that get into use and the lines of codes delivered. Since about the 15% of the delivered function-points, or lines of code, were not used, and consequently, not paid according to the contract, then we can conclude that real effectiveness, the internal one, was ***85%***. The 15% of ineffectiveness was transformed in lowering the *economic efficiency* in our organization in 15%.

Consequently, according to Standish Group 37% of the projects succeeded on 2010, while we have been having better numbers in the last 23 years, we think that ***combining reflective practice and theoretical analysis and clarification of the main concepts and notions*** might very probably be one of the ways of improving the effectiveness in designing and implementing methodologies. *Contributing to this improvement is a main purpose of this paper.*

As we have been showing, and continue to show below, an explicit clarification of the concept or notion of method *is not just intellectually important, but also pragmatically very useful* because it might very probably have a huge impact in the efficiency and effectiveness of professional practice, in general, especially in Systems Engineering, information systems development, informing systems design, and software development.

Top-down and Bottom-up Methods

What we will address in this section can be applied to thinking (or knowledge production) and doing (professional, pragmatics, and/or praxiological) methods. But we will focus on the latter because of the above manifested purpose of this paper.

Returning to what we showed before the last section, it is evident that *a method is adequate or effective* ***if, and only if****, our initial conditions are those required for the initiation of the method,* ***and*** *our objective is the same method's final point*. An obviously boring example, even a tautology, is to say that a "top-down" method goes from the "top" to the "bottom", and a "bottom-up" method goes from the "bottom" to the "top". But, while this is so obvious, why do most computer installations or software houses use the methodical standard, which reigned for at least 20 years (the 70s and 80s), and was a "top-down" one? As we asked above, how can we apply a "top-down" method if we have access just to the bottom of a situation? Shouldn't we apply a "bottom-up" method to get to the "top" and then, and only then, it would be feasible to apply a "top-down" method? *Aren't both methods* ***polar opposites*** *and not contradictory opposites*? If so, why have most computer installations and software houses excluded the use of "bottom-up" methods when they standardized through the use of "top-down" methods? To use "top-down" methods we should be at the "top" in our initial conditions, and this is not always the case in an information systems development project. "Top-Down" methods should be used when it is adequate and feasible. If its use is not feasible an alternate method or a complementary one that would change our initial conditions should be used, as it is the case of the "bottom-up" methods.

On the other hand, using "top-down" methods does not necessarily mean *always* using them. And, if for some reason (to reduce maintenance costs, for example), we need to always use them, this doesn't mean that we should ***only*** use them, ***excluding*** any other or any complementary opposite method. A similar argument might be done for the latest methodological tendencies characterized in standardizing with "bottom-up" methods. It seems that the fad pendulum is going back to the bottom-up methods. ***If the concept of "method" were handled adequately, there would be no "top-down" and "bottom-up" standards***, with a methodological and methodical pendulum swinging between both of them every 10 or 20 years. If the concept of "method" were adequately understood and appropriately handled, methodologies would contain, or at least encompass, the possibility of both "top-down" and "bottom-up" methods, so any of them, or an adequate combination of both, could be applied according to the initial conditions at hand, the purpose or the end sought, and the nature of the project.

Combining "top-down" with "bottom-up" methods is actually a systemic process, where both methods complement each other and frequently make each other possible and feasible. This combination might also generate a cybernetic process if both kinds of methods are part of a loop of control or—better—part of an integral control, where both methods *feedback* and/or *feedforward* each other in an adaptive loop. It is clear that "bottom-up" and "top-down" methods are opposites, but this opposition is not a contradictory, but a polar one. Both do not contradict each other, as it is explicitly or implicitly suggested, when the methodological standards are reduced to one of them. They oppose each other as polar entities and, thus, complement each other and frequently require each other.

For similar reasons we might make analogous arguments for thinking methods, i.e. induction (a kind of bottom-up that goes from the specific to the general) and deduction (a kind of top-down that goes from the general to the specific) might be combined via cybernetic co-regulative loops (by means of negative feedback and feeforward) and potentially producing synergies and emergent properties (by means of positive feedback). Thinking and doing are usually interrelated, influencing each other, but our emphasis will be on methodologies oriented to doing/producing/designing systems or supporting decision making, not just to think about, reflect on, or analyze them.

Definitions of "Tool" and "Technique"

Several authors use the term "method" as a synonym of the terms "technique" or "tool," which might, in our opinion, con-fuse the respective concepts, with potential negative impact in both theory and practice. Other authors point out explicitly that there is a difference between the meanings of "method" and "technique", but, in our opinion, they are not very successful in clearly identifying this difference. Gortari (1984), for example, insists that there is a difference between "method" and "technique" and that techniques are included in methods and they should not be confused with each other. Furthermore, he says that a technique could be included in several methods but no method is included in a technique (Gortari, 1984; pp. 16-17). In our opinion, his arguments are about the *existence* of a difference, but not about what the difference really is. In order to show the difference, we will use, as an illustrative metaphor related the etymological origin of method as a "physical way", or a "physical road". Elsewhere, (Callaos, 1995; pp. 243-281) we provided a more analytical and detailed approach to show the conceptual differences between these two notions. Here we will use the physical metaphor as a vehicle of illustrating the conclusions we achieved in the mentioned work.

Associated to a "physical way" (e.g. road, highway, street, driveway, bike track, etc.) there are means of transportation, vehicles adequate to traversing it. An automobile, for example, is associated and adequate to a highway but it is not adequate for a sidewalk. Analogously, associated to a method there are tools that are used, or could be used, for traversing it, or passing through it. Although it is not possible to traverse a way without a transportation means, and a transportation mechanism needs an adequate way to accomplish its transporting function, and although "way" and "transportation means" are strongly associated and coexist in practice, they are very different concepts. Similarly, a method is a way, a "from-to" medium, and a tool/technique is a "means" for traversing the medium from its initial to its final state. A method gives its user ***direction***, and its associated tools/techniques give him/her means for ***propulsion*** in such direction. ***Method is effectiveness-oriented and its tools/techniques are efficiency-oriented***. Consequently, methods should not be confused with their tools and techniques. They are very different concepts, although very associated ones. They are as different as a highways and the cars traversing them, and as associated as highways and cars. This conceptual distinction has a very significant usefulness in practice. Now let us use the same metaphor to express the difference between the concepts of "tool" and "technique".

The meaning of the word "tool" has several senses. In the sense we are using the word here, "tool" means "something (as an instrument or an apparatus) *used* in performing an operation or necessary in practice of a vocation or a profession"[3] (Merriam-Webster, 1999); "anything *used* in the performance of non manual work"[4] (Watson, 1972). Anything to be used needs a *user* with the corresponding skill; it needs a *technique*. The word "technique" derives etymologically from the Greek "τεκνη" ("*tékhnē*") which means art, craft (Weekley, 1967) ability, capacity, skill, talent (Corominas, 1976). Thus, as we concluded elsewhere (Callaos, 1991; 1995, pp. 243-281) after a comprehensive analytical approach, a technique always exists in a ***subject***; it has a subjective existence, while a tool always has an ***objective*** existence. "Tool" and "technique" are two sides of the same coin: the "tool" is its *objective* side and the corresponding technique for using it is its *subjective* side. "Tool" and 'technique' are very different, though related, concepts. They are as different as a car and the driving capacity of its driver. Both concepts should not be confused as they usually are. ***Their confusion has not just theoretical impact, but also a practical one***. A tool, for example, might be bought, but it is impossible to buy a technique. It should be acquired, apprehended, learned, gained by effort, by experience. A car might be bought, but the driving capability has to be gained by experience. This is so evident, that it is surprising how frequently we find managers, especially information systems development managers, and even academics managers confusing both concepts. Some managers try to buy techniques! Some academics try to teach techniques, confusing knowledge with experience, and instruction with training! A professor can transfer knowledge and information related to a tool but cannot transfer the technique required to effectively use the tool. *Just by means of using the tool the technique can gradually be acquired.* Technologies include tools and techniques. Although sometimes the term "technology" is associated just with its objective aspect, it also include its subjective side, i.e. techniques to use the physical tools.

---

[3] Italics added

[4] Italics added

In summary, "method", "tool" and "technique" are three completely different concepts. They are strongly associated but very diverse. A method is characterized by initial/final states and a *direction*. A tool/technique combination is the *driving force* to traverse, to apply a method; it is the "*propulsion*" needed to get moved in the direction dictated by the method. *Methods are* ***effectiveness-oriented*** *means, and tools/techniques are* ***efficiency-oriented*** *means*. In turn, "tools" are ***objects*** used by ***subjects*** trained with the corresponding "techniques." **A** ***method is something to be known, a tool is something to be bought, or constructed, and a technique is something to be trained in and experienced***. ***Methods require understanding and comprehension, tools require money (to buy them) or persons-months (if they are to be construed), and techniques require aptitude and attitude*****.** Now we are in a position to attempt a definition of methodology.

Definition of "Methodology"

As we said, "Methodology" (with a capital letter) is the science of methods or the theory of methods. We also said that method has been largely identified by the Logic of Science and/or Philosophy. Thus, in this sense, by "Methodology" is meant: "science of science", and, as such, it will be just one Methodology, epistemology-oriented with no praxiological reference. But, when we use the term in plural, i.e. "methodologies", we have to use the word in another sense. In such a sense, "methodology" would mean "a theory of methods in a given area of knowing and/or doing". This definition would include both: epistemological and praxiological orientations. On the other hand, "theory"[5] means "organized body of ideas" (Watson, 1972), a coherent corpus of knowledge (cognitions) related to a domain of objects" (Ferrater-Mora, 1980).

Since organizing and achieving coherence requires ***relating***, "theorizing" also requires some kind of "relating". Consequently, we might suggest that a "methodology", as a theory of methods in a given domain of thinking and acting/practicing, is a *"set of related or 'relation-able'* [6] *methods with their respective tools and techniques"*. ***A methodology is a "network of methods with their respective tools and techniques"****; it is "a graph of ways of thinking and/or acting/doing"*.

So, the concept of methodology should not be confused with the concepts of method, tool or technique. It contains all of them but should not and could not be reduced to any single one of them. A graph is conceptually different from its arcs and its nodes. A network of highways, avenues and streets, a city, for example, is not to be confused neither with some of its highways, avenues, streets, or a sequence of them (methods) nor with the cars (i.e. tools) traveling in the network, nor with the driving training (techniques) of the cars' drivers.

Returning to the example mentioned above, a meta-software like CASE (Computer Supported Software Engineering) is not a methodology, as it has been presented by many

[5] We made a comprehensive analysis regarding the notion of "theory" in (Callaos,1995, pp. 243-281)
[6] We will clarify in the next section the difference between "related" and *"relation-able" or "relatable"*

vendors since its origins. It is a tool, or a set or system of tools. In the top-down approach of Systems Analysis (e.g. Structured Analysis, originated in the 70s) for example, is a method, the Data Flow Diagrams are some of the tools used in this method, and the aptitude/attitude to elaborate Data Flow Diagram is a technique. A methodology for developing information systems could include *different* top-down alternate methods for systems analysis. It needs not to be necessarily reduced to just one top-down method , as it has been frequently the case for authors that *confuse the concept of methodology with the concept of method*. The top-down Structured Analysis *standardization* "decreed," by information systems mangers, in many organizations, especially in the 70s and the 80s, has been the result of conceptual confusions. A methodology for developing information systems may include different top-down methods. It may even include bottom-up methods accompanying top-down methods and, consequently, complementing each other. This inclusion of top-down and bottom-up methods in the same methodology is seen as contradictory and non-sense by some authors, consultants, and practitioners that confuse the concept of "methodology" with "method". But, *with the definition and conceptual distinction we are making in this paper, a methodology could contain opposite methods in order to integrate them in a whole, if a particular situation calls for it. Otherwise, if "methodology" is confused with "method", the possibility of complementary polar opposition vanishes and may emerge an intellectually and/or praxiologically delusional mirage of contradictions between methods, which in reality are polar opposites.* Let us explain that with more details and illustrate it with the metaphor we have been using.

Systematic/Closed and Systemic/Opened Methodologies

We have defined methodology as a "set of related or 'relation-able' methods, including their respective tools and techniques". If a given methodology consists of methods with fixed relations among them, and if its relations are not contextual-dependent and they do not depend on the particular situation where it is being applied, and if the methodological relations are "pre-given" and the particular context of each application is to be adapted to a pre-structured sequence of "ways" or methods, then we will be talking about a *closed* methodology or a closed methodological system.

On the other hand, however, if the relations among the methods are not "pre-given", not pre-fixed, but decided according to strategies, rules and heuristics that take into account the particular situation where the methodology is to be applied, then we would be talking about an *open* methodology or an open methodological system. The potentiality of this openness would be intensified if a given methodological application could include new relations as well as new methods, tools and techniques.

Open and closed methodologies have their respective advantages and disadvantages, their own pros and cons, and strengths and weaknesses. In the case of a closed methodological system, a high level of methodical standardization could be achieved and, as a consequence, the methodological *efficiency* would increase, but its *adaptability* will decrease, and, hence, its *effectiveness* might be hampered in a dynamic methodological context. On the other hand, an open methodology will be more adaptive and, hence, more effective in dynamic contexts, while its efficiency would be decreased because of the diminished possibilities of standardization. In the first case we would have efficiency-oriented and ***systematic*** methodologies, and in the second case we would be working with effectiveness-oriented ***systemic*** methodologies. Systematic methodologies require less decision-making efforts, as well as less psychological and managerial energies each time they are going to be applied, but are more rigid, less flexible and of limited adaptability. Systemic methodologies, on the other hand, are more flexible and adaptive but less efficient because they require more decision-making efforts (up front and along the process), more psychological and managerial energies, and probably more person-power.

In dynamic contexts systemic methodologies should be preferred, and in stable contexts, systematic methodologies would be more recommendable. In the information systems development domain, for example, the level of "*systemicity*" or "*systematicity*" of the developing methodology should depend on the computer installation and its organizational context, as well as on the kind of information system to be developed and its business context. Developing an Executive Decision Support System (EDSS), for example, requires a methodology with more openness and *systemicity* than what would be needed in the case of a relatively stable electronic data processing (EDP) system, as is the case of a general ledger or a payroll, by instance. Stable organizational context allows us to apply systematic and, hence, more efficient methodologies. However, dynamic or uncertain environments~~,~~ require~~s~~ systemic methodologies that have the required adaptability level. In this case, a decreased efficiency is the obligatory cost that should be paid in order to have the required flexibility and adaptability that are imperative for the achievement of the sought effectiveness.

Methodologies are Informing and Knowledge Systems

In any kind of methodologies (systemic, systematic, or hybrid) there should be supporting *information* systems and *informing* processes, which might be human-based, computer-based, or hybrid ones. The latter are the most usual ones. Computers process data[7] and human beings process human ***information***. On the other hand, methods need to be known by at least one person, in order to ***know-what*** to do, and tools need to be handled by someone (with adequate aptitude and attitude), i.e. tools needs ***know-how*** and the *will* to apply this know-how. Systemic methodologies also require ***know-where*** and ***know-when***. Gerald Weinberg (1982) stressed the importance of the "know-where" and "know-when" a long time ago, even in the context of the Structured Methods (e.g. Structured Analysis, Structured Design, Structured Programming). Paraphrasing, with other terms and concepts, what he proposed in his book "*Rethinking Systems Analysis and Design,"* it is necessary to make some kind of insertion of the Structured Methods, which are highly systematic, into a more systemic context. In a systemic methodology we should also include a "***know why***."

[7] Depending on how we define "information", computers can process or cannot process information. In any case, computers can always process "data." For a detailed analysis regarding the differences between the notions of "data" and "Information" see Callaos and Callaos, 2002; for a more comprehensive and extended analysis see Callaos and Callaos 2013.

Banathy Jr. (1992) stressed the importance of this kind of knowing as ingredient of the information system being designed. Russell Ackoff also stressed the importance of the "know why" in consulting activities, especially those oriented to real life problem solving. He suggested asking "why" to clients or users on two levels: 1) "why do you want that"? and, after you get the answer, ask 2) why do you need or want the "why" of what you just informed me about? His reasoning was based on the frequent experience he had that when clients or users ask for *what* they thinks they want, they are actually asking, in an implicit and non-conscientious way, for *how* to achieve the real *what*. When they say *why* they want/need something, they probably mean *what* they want/need.

For similar reasons, the "know why" should necessarily be included in a systemic methodology, because the methodological effectiveness, and even the respective efficiency, might strongly depend on it, i.e. on the real why.

Two kinds of knowledge (know-what and know-how[8]) are required for many methodologies, including most systematic methodologies, and five kinds of knowledge (know-what, know-how, know-where, know-when, and know-why) are required for a systemic methodology. Since these kinds of knowledge should be related or relatable, then we can conclude that any kind of methodology require a ***knowledge system***, which very frequently has (explicitly or implicitly) to be a distributed, and a systemic one. To develop any information system or software we obviously need a knowledge system as a necessary condition. More precisely, we need a Human Knowledge System, which, in turn, might be supported by information systems, knowledge-bases, project control software, or meta-software (e.g. CASEs). We would like to stress the fact that no matter how sophisticated the automated support systems are (even if they are named "knowledge systems" or "expert systems"), they will always *support the human knowledge systems*, especially if we are talking about methodological systems. Some vendors con-fuse these supporting tools with methodologies and they are wrong from both the conceptual and the practical perspectives. Some vendors and managers (even Chief Information Officers) con-fuse the notion of methodology with a set of related written rules, procedures and ways of making diagrams for describing systems. *This is wrong. This would be like con-fusing a territory with its map. A map represents and describes a territory, but it is not a territory*.

A methodology necessarily requires, as a human subsystem, an adequate Human Knowledge System with two kinds of knowledge (know-that and know-how) in the case of systematic methodologies and five kinds of knowledge (know-that, know-how, know-where, know-when, and know-why) in the case of systemic methodology.

Power, Ethics and Systemic Methodologies

The Human Knowledge System we referred to above is mostly (though not uniquely) related to tools which effective use requires adequate handling by means of technologies. A

[8] For more details regarding the differences and the complementarities between the "know-what" and the "know-where" see Callaos, 2011.

very good car (tool) is not useful if it is not operated by adequate drivers (technique). In turn, as we have shown above, a technique requires both adequate aptitude (know-how) and appropriate attitude (willingness) in order to be effective. Consequently, it seems that if we have the knowledge and the required willingness to apply it (on behalf of both the developers and the users) it would be sufficient for the success of the system to be developed, installed, operated, used, and maintained. Regretfully, this is not always true. Frequently, it is also necessary to have the ***power*** to do it, especially when the installation of the systems, their beta tests[9], user training and initial use requires a significant person-power from the user organization or sub-organization. This kind of person-power is not usually controlled by the manager of the developing project, let alone by the analysts or programmers. Consequently, a technically and organizationally good information system could fail to be successfully installed with adequate user training and beta testing because its project manager lacks the power he needs regarding the users to get them to use the system in an adequate way and/or not to give the users other kind of work that gets evidently in conflict with the user training and the respective beta test.

The users might love the new developed system, but if their boss, the person who has the real/formal power over them, continuously asks them to do other things, disturbing and delaying their training, their using of the system and, consequently, its beta testing, debugging, and respective certification process, then the system could fail, money would be lost, users, developers and managers would be frustrated, professional prestige will unjustly be hurt, personal fights could spring, and the organization as a whole, would pay the economical and organizational loses at the end of the project.

This situation takes us to the ethical domain. Decisions taken by the users' manager could have significant professional and technical effects on the development and implementation processes. Whoever has the power to enforce his/her decisions might not be the same person who is going to be accountable for the effects of such decisions and their respective enforcement. This kind of situation has always been organizationally unacceptable and ***ethically wrong*** *and questionable.* What is not good for the organization as a whole is not organizationally ethical. And, *what is, or seems to be, good for a person and bad for another, or for the organization, generates ethical problems that should examined.*

In systemic methodologies, an inverse problematical situation might also occur with ethical effects. If the relations among methods are not pre-established and fixed (as it is the case in systematic methodologies), but are decided, and re-decided, as the developing process goes

[9] We are using the term "beta test" to refer to the activities of bug fixes which are mainly identified by the users while using the system in real life operations. This is why the beta test of customized information systems requires a high volume of work from both the user and the technical staff. It also requires significant *emotional intelligence* as well as *psychological and managerial resources*. In the phase of beta test bugs in analysis and design are also identified and new emerging requirements emerge as consequence of the business and the organizational dynamics. This is why it is frequent that the corrective and the perfective maintenance of the system starts before the beta testing had finished which exponentially increase the complexity of this combined process. This phase is the most dangerous one for the failure of a technically and organizationally good and adequate tailored information system.

on and adapts to its dynamic environment, decisions taken by the project manager and by the analysts/designers/programmers might have negative organizational effects, beyond the probably justified technical reasons that supported these decisions. In this case *a **"technical good" is achieved by impairing the "organizational good."*** It is the opposite situation to what we described above, i.e. a situation *when inadequate use of sub-organizational power and/or "local good is achieved, impairing the 'technical good' and the "'organizational good'."* When the good of one part is achieved by lessening the good of other part and/or the total or the common good, then an ethical problematic situation is at hand, and it should be treated as such. *The larger the system being developed and implemented, the more probable ethical problems would emerge*.

Systems methodologies, and especially systemic ones, are ways of thinking, deciding and acting. Ethical considerations could be avoided in the thinking processes, but there is no way to avoid ethics when deciding and acting. The action cannot be isolated from its potential ethical consequences. We can restrict logic and science to the "what-is" realm and avoid "what-ought-to-be" considerations. But this a-ethical stand cannot be taken when action decisions are taken, including methodological ones. The lack of attention to potential ethical consequences in methodological decisions is in itself non-ethical in our opinion. Consequently, ***the implementation of a methodology in a specific project of information systems development includes ethical issues and requires an appropriate blend of Science, Engineering and Ethics**, especially if the methodology to be applied is a systemic one.* In a systemic methodological context ethics considerations should not be avoided because decisions and actions are needed beside thinking processes. Moral considerations should be parts of a methodological implementation; ethics should be a component of information system development, especially if the methodology to be used is a systemic one.

Ethical naturalism denies the prima facie distinction between establishing facts about the world (science) and making value judgment on them (ethics). Harrison (1967) points out that "according to ethical naturalism, moral judgments just state a special subclass of facts about natural world." We are in a need of what we can call "***ethical methodologism***," according to which, paraphrasing Harrison, ethical knowledge would be just a special kind of the diversity of different kinds of knowledge needed for a good development and implementation of a system, i.e. ***technically good, organizationally good and ethically good***. Thus, the human knowledge system supporting a system development and implementation process should contain six kinds of knowledge: *"Know-What", "know-who", "know-why", "know-where, know-when" and "know-what-ought-to"*.

The ethical component in the system development and implementation requires not just the related knowledge, but the means to enforce the consequences of this knowledge, and to protect and be protected from unethical decisions and actions. Thus, ethical rules should be generated, agreed upon and enforced by legal means. Consequently, *a systemic methodology should contain a legal component addressing ethical issues*. Research on legal aspects of information systems and, in general, Information Technologies, started several decades ago (see, for example, Young, 1992). Analogous research in the methodological area will help into adding an adequate legal component to a systemic methodology. Actual contracts on information systems development projects center their emphasis on the systems as a *product*, yet more emphasis is needed on the system as a *process* i.e. on the methodological system and not just on the information system to be developed.

Let us resume our conclusions before the next section: A systemic methodology contains:

(1) a set of methods, tools and techniques;

(2) a set of alternative relations by which methods are related one way or another according to the nature of the information or knowledge system to-be developed and implemented, to its environment, and to the dynamic and changes in its development/implementation process;

(3) a set of "know-where", "know-when" and "know-why" (beside the know-what and the know-how required by the techniques mentioned in point 1), as well as heuristic rules based on experience and ethical considerations, which will support the selection of the most adequate relations among methods and people;

(4) the set of the methods-mentioned in point 1. These contain not just informatical and intellectual methods, but also managerial, ethical and legal ones. Consequently, the set of tools and techniques also include managerial, ethical and legal ones.

The implementation of a systemic methodology requires adequate decisions and actions in order to assemble an adequately efficient, effective and ethical knowledge/action ad hoc system whose objective will be the development and implementation of an adequately efficient, effective and ethical sought information system according to its users' needs and requirements. Thus, an implementation of systemic methodology seems to be at a second level of *meta-efficiency, meta-effectiveness, meta-knowledge, and meta-ethics*, i.e. to develop efficient systems in an efficient way, effective systems in an effective way, information/knowledge systems via methodological information/knowledge systems, and the ethics of generating ethical decisions and actions from other human beings. Consequently, *isn't some degree of wisdom desirable for the implementation of systemic methodologies?*

Importance of an Updated *Trivium* in Information Systems Development[10]

A computer-based information system has necessarily two main sub-systems, one for *Electronic Data Processing* and another for *Human Information Processing* (Figure 3). The development of a computer-based information system is an organizational process which also has, in essence, two similar subsystems.

[10] A more general work on this topic, which would complement this section, can be found in Callaos, 1995, pp. 527-530

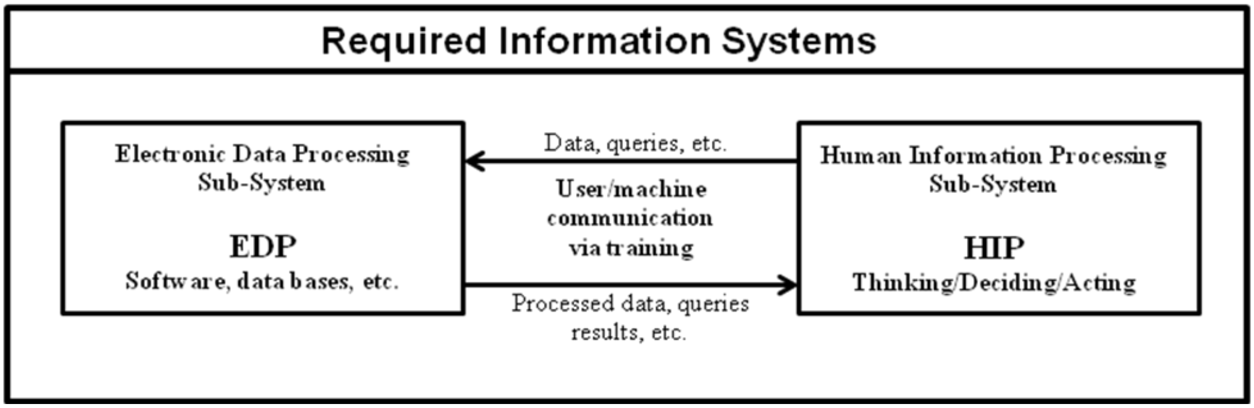


Figure 3

Both sub-systems and the interface between both of them should be addressed by analysts/synthesists, programmers, testers, etc. in the context of the methodology being used in the development of the respective information system. Otherwise there would be no information system developed, but just an EDP system (software, data bases, etc.) The data should be transformed to human information which, in turn, would support human cognitive processes in order to have an information system. The methodological system (related methods, tools, techniques, analysts, designers, programmers, tester, supervisors, managers, etc.) should address both the EDP sub-system to be created (via artificial languages) and the HIP sub-system which is the source of the system's requirements and the receptor of training needed to operate the EDP system. Consequently, the methodological systems and processes interact with the EDP sub-system via ***artificial languages***, and with the users via ***natural language***. This means that the methodological system should be able to have both effective and efficient communication with the computer via artificial languages and with the users via natural languages (Figure 4).

Our experience through 30 years showed us that persons participating in the methodological system, who are in charge of developing the information systems, are more able to communicate with the computer via artificial languages than with the users via natural language. *Adequate translation between* ***artificial languages*** *and* ***natural language*** *are necessary conditions of the success of the developing process.*
This translation is the responsibility of the technical staff (analysts, designers, programmers, etc); it is not responsibility of the users. ***Hybrid languages*** have been created since, at least, the 70s to support an adequate translation between artificial and natural languages. It is the responsibility of the computing staff, technicians, and professionals to handle these hybrid languages. It is not the user task to handle these hybrid languages and to certify that the requirements were well collected.

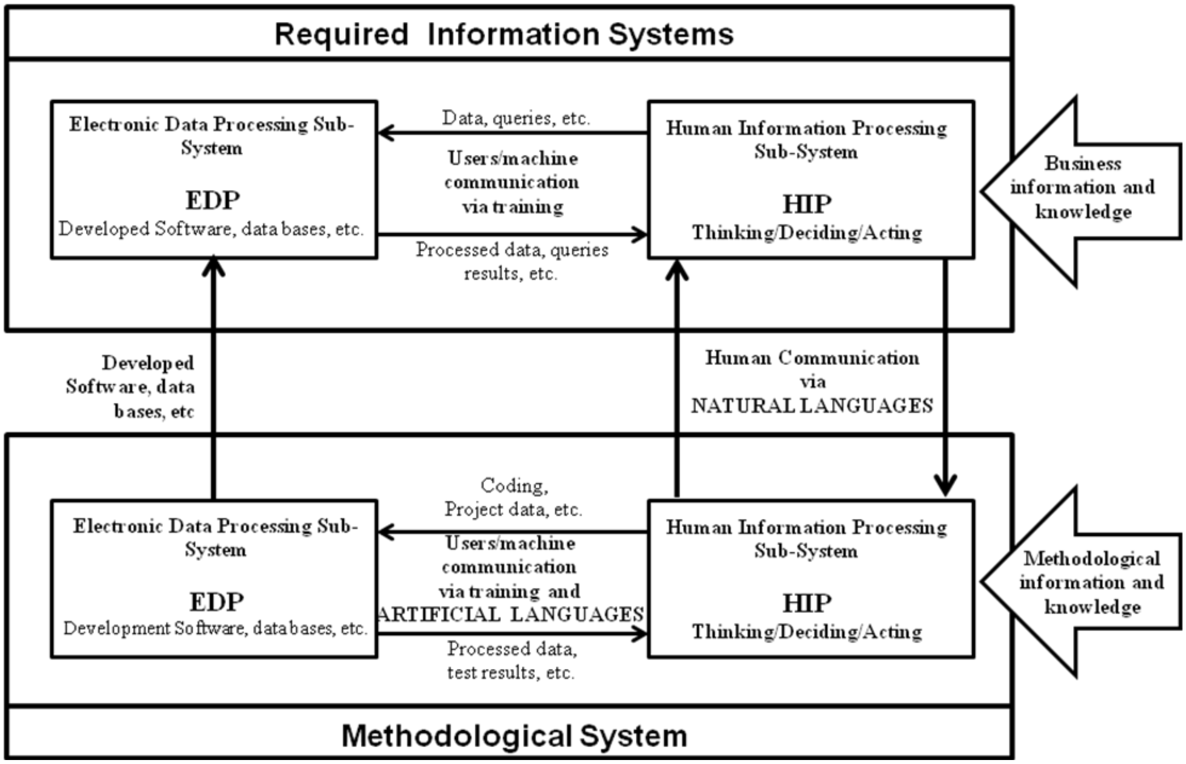


Figure 4

Many methodological experts and reputable consultants insisted that it is the responsibility of the user to make the certifications related to the translation between the natural and hybrid languages. For example, at least in the 70s and the 80s, according to some very used methodologies (as the structured ones for example), authors, consultants, and professionals emphasized that the user should be able to understand Data Flow Diagrams (produced by analyst), Tree Diagrams (produced by program designers), Structured Dictionaries (produced by programmers and data base designers), etc. in order to be able to certify the requirements indentified regarding the system to be, or being, developed. ***This is wrong!*** It is also unfeasible in many situations. These hybrid languages were sufficiently (but not completely) effective in communicating analysts, designers, programmers, and testers, but they were ineffective in communicating these technicians and professionals with the users. It was (and sometimes it is still) expected that the users would handle effectively the natural language and the hybrid language related to the specific business of the organization or sub-organization. Furthermore, frequently the users were more able at managing their *organizational hybrid language* than their natural language. Consequently, in any case, in the worst and best scenario, computer engineers (at least the analysts and the project supervisors and managers) are who should handle the natural language in an adequate way. This means that communication with the users in natural language should be handled

correctly and ***effectively***. To use the language effectively for communicating with the users, the analysts (or requirement engineers), supervisors and project managers (at least) should 1) carry on effective ***dialogs*** and 2) be able to ***express*** themselves effectively. Consequently, ***correct speaking, dialog ability, and expressing effectively*** seems to be necessary condition to elicit, verify, and validate the respective requirements of the information system to be, or being, developed. These three aptitudes are also required along the developing process especially in the beta test phase where emotions and pain are at their highest in the project life cycle.

Even the computer engineers or technicians (analysts, designers, programmers, testers, etc.) require human communication skills to apply their technical methods. They need natural language as well artificial and hybrid languages to communicate correctly in their professional, managerial and technical activities. Consequently, to communicate correctly, to carry on effective dialogue and to express effectively is necessary even to communicate the group of people who are applying technical methods in the context of software engineering or any other kind of engineering activities where some kind of methodology is necessary, or desirable, in order to achieve the engineering objective (product or service). Hence, correct, adequate, and effective natural language handling is required for both

1) interfacing with users for adequate a) requirements elicitation and b) training them in and efficient and effective use of the system developed for them; and

2) communicating adequately in the context of the developing team while applying their respective technical methods.

The three aptitudes in natural language handling, mentioned above, were what the ***Medieval Trivium*** was about: ***Grammar***, for correct expression, ***Dialectic***, for adequate dialogs, and ***Rhetoric*** for expressing effectively. Now, let us be intellectually honest: how many students of Computer Science or Engineering graduate have adequate skills in these three *necessary conditions for successful information system developments*? How many business organizations identify this lack of necessary skills and foster/support their improvement in its IT personnel? How many computer scientists or engineers are aware of the importance of these three skills for their effectiveness as professionals? How many university professors in computer Science/Engineering or in information systems, in business schools, are aware of this problem? How many universities are really addressing it?

Paradoxically, because of Medieval Trivium was so known and the associated skills were so explicitly developed that the word “trivial” emerged; which is derived from “Latin *trivialis*, i.e. found everywhere, commonplace, which in turn is derived from *trivium.” (Merriam Webster).* Are these three ways, tri-vium, still common in our “advanced” Society? Is it common where it is more needed as it is the case of information systems development? Is it even common in Higher Education? Is it common in academic informing and activities? Is it common in Academic Writing? Is it common in scientific research communication? Are we aware of the importance of the Trivium for Inter-disciplinary communication?

Because these three vital skills are not as common (trivial) as they were in the Middle ages, we are convinced that ***an updated and adapted Trivium is important in academic and professional activities.*** Since this paper is on the topic of systemic methodologies we cannot overemphasize the importance of an updated and adapted Trivium as part of the ways, methods, which should be included in such methodologies especially if we are referring to systemic ones. (Figure 5) These three ways, or methods, for ***effective human communication*** should be included along with technical methods required to ***communicate with the computer***. Otherwise, we will not have the adequate means to address both sub-systems shown above in figure 3 and 4.

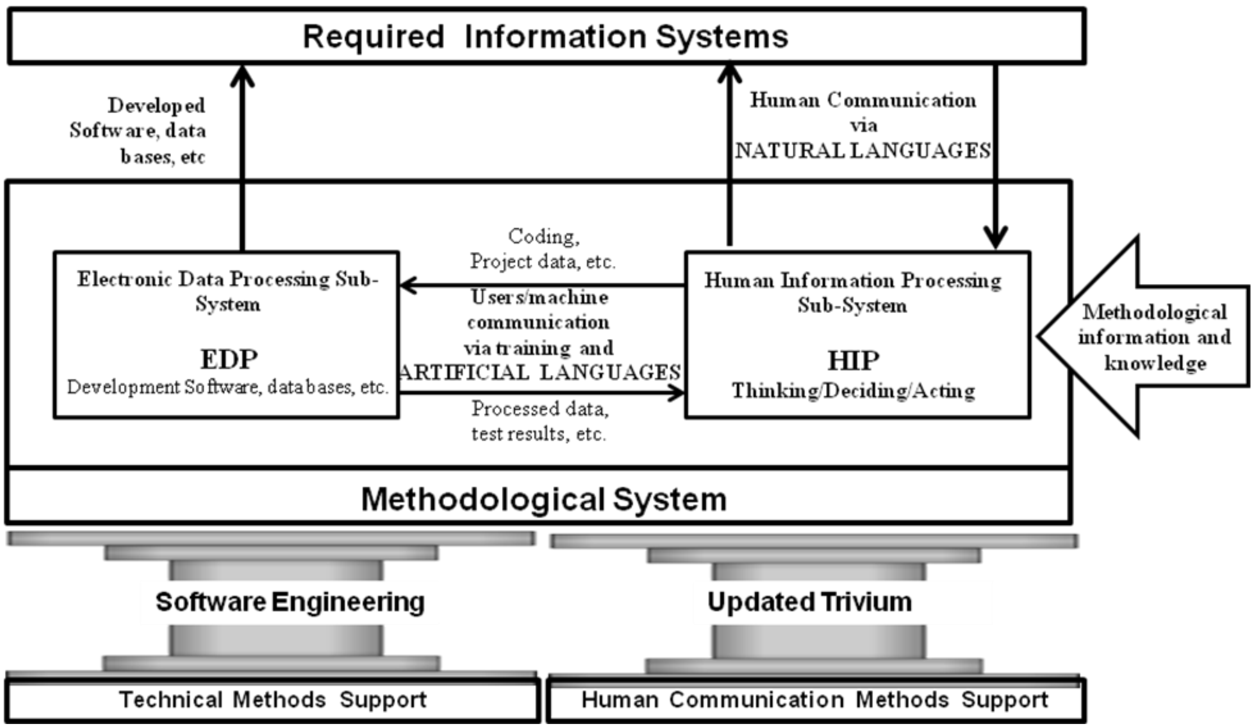


Figure 5

Ethos, Pathos, and Logos in Systemic Methodologies

Correct expression, in the context of Systems Development Methodologies, has been achieved, in part, with technical writers who make adequate language editing or actually write the manuals for the users of a given system. The area that is still not adequately handled is the one related the Requirements Engineering phase. In this part natural language is mostly *spoken*, among users and system professionals, not *necessarily written*. Consequently, *technical writers can provide limited support, at the end of the process but not during it.* Since requirements engineering activities are usually handled by people (named system analysts) who have at least 7 years of experience and seldom by programmers who just graduated in Computer Science or Engineering, then the experience of the analyst might provide him/her with the support required for expressing him/herself in a sufficiently ***correct*** way. This experience might also help in managing a productive dialog in an ***effective*** way, or in identifying an adequate facilitator for the meetings where analysts and users maintain dialogical processes as part of the requirements elicitation activities. Consequently, two of the three ways or methods (an updated *tri-vium*) for an adequate human communication might be handled by means of the experience acquired by the analysts. But, rhetoric, in its original and technical sense (effective expression, i.e. expression that achieves the objectives of the expresser) is not easily attained just by experience. Hence, it might be advisable to shortly describe the importance of Rhetoric, and its three essential components (*Ethos*, *Pathos*, and *Logos*) in increasing the probability of success in the application of a methodology, especially if it is a systemic one.

We suggest that the traditional triad of Ethos (credibility), Pathos (emotion), and Logos (logic) are applicable and/or being (implicitly or explicitly) applied, and/or should be applied in any methodology that require collaboration of different individuals using natural language and, in general, human communication. Consequently, *methodologies that support information systems development require the triad of Ethos-Pathos-Logos for improving their effectiveness*, especially if in the case of systemic methodologies which are the most adequate ones in uncertain and dynamic contexts, where methodological *adaptability* is almost a must.

Ethos, Pathos, and Logos have been always considered as pillars of effective writing, which is *one-way communication*, i.e. from the writer to the reader. In the case of implementing a methodology for information system development, for example, communicational processes are, at least, *two ways*, because they mostly are held in dialogical or conversational contexts. Consequently, Ethos, Pathos, and Logos should support both sides of the dialogical-conversational process. Analysts, designers, project managers, and business/organizational managers should be aware about this issue in order to have effective two-directional (or multi-directional) communication processes. If this is not the case, and most frequently it is not, then those who are applying and managing the application of the methodology should communicate this fact to the other side. This means that a *meta-communication* process is necessary in order to get a minimum of awareness regarding this issue from the other side of the required communicational process. This is one of the reasons why a ***second order*** level of Ethos, Pathos, and Logos (*meta-ethos, meta-pathos, and meta-logos*) would improve the methodological effectiveness.

In the context of information systems development, communications in *natural* languages are needed 1) to illicit users' requirements, 2) to manage the project, 3) frequently to communicate analysts, designers and programmers, and 4) to adequately relate with the persons that hold organizational power in the area which will be supported by the system. This is especially important in the last development phase in which the system would be beta-tested and debugged through its initial operation in real life situations and procedures (in the session above entitled "Power, Ethics and Systemic Methodologies" we provided some details regarding the importance of *power and ethics* regarding this phase). The four different kinds of communications, we just referred to, require to be made via natural languages, and convincing/persuading are necessary conditions for methodological effectiveness. These four kinds of communication require:

1. *Convincing* by means of the ***character or source credibility*** (***Ethos***) of any of the speakers involved in the communication. If the interlocutor in a dialog or conversation is unethical and/or is perceived as unethical, he/she would not be effective in any dialog, conversation, or discussion.

2. *Persuading* with the support of adequate ***emotions*** (***pathos***) in both communicational sides. There is mounting evidence regarding cybernetic relationships among cognitive processes and emotions. Consequently, the cognitive processes involved in communication influence and/or are influenced by negative and positive emotions. The speaker's tacit emotions can influence the reception of what is being communicated in negative or positive ways. Simultaneously, the emotion that the speaker might generate in the receiver of the message might influence his/her cognitive processes and, hence, his/her understanding, decision and action. More details regarding the cybernetic loops between cognition and emotions (or more precisely among cognition, conation, and affect) can be found at www.iiis.org/Nagib-Callaos/Cognition-and-Knowledge/, and at the references on which this article is based.

3. Persuading interlocutors by the use of ***reasoning*** and ***logical arguments*** (***logos***). In this context, *technical logic* is a necessary condition, but not a sufficient one. *Business* and *organizational logic* should also support any reasoning made when applying a methodology, especially if it is a systemic one. Sometimes, personal logics and hidden agendas (especially regarding manager holding power over the users of the system) should also be addressed in order to have an adequate methodological effectiveness.

Meta-Ethos is required because professionals, users, and managers involved in the information systems development would certainly have not the time or the resources to generate ethical behaviors from all of those involved in the development project and those to be involved in the use and maintenance of the system. This would need an organizational change which, in most circumstances, is not feasible with the time and the resources available for the respective developing project. Even though, an ethical behavior from the professionals, designers, programmers, and managers of the project might be contagious and might spread similar kind of behavior, or inhibit non-ethical behavior. But, the uncertainty regarding this possibility should move the ethical professionals, managers, designers, and programmers to increase the probability of inhibiting non-ethical behavior via explicit clauses in the contract and/or explicit organizational rules and procedures, adequately enforced and monitored by effective supervisor. This would be a process of decision making and acting in a ***meta-ethical context, i.e. the ethics of generating ethical behavior and/or finding the ways to inhibit non-ethical activities.*** This should be feasible in the context of the project. Otherwise the risk of failure would increase and managers and developers should be aware about it. According to our experience Ethics or Meta-Ethics are not always considered important for lowering the risk of failure in information system developments. Resuming we can conclude that *information systems developers should have 1) professional ethical behavior, and 2) meta-ethical concern regarding the possibility of generating ethical behavior and/or implementing the means to inhibit non-ethical ones.* In this context ethics has a ***pragmatic value***, not just a moral one

With regards to *pathos* a writer or an artist objective is mainly to generate or to transmit emotions to the reader or to the spectator. The means used by the writer are basically verbal ones, while the artists would also use audio visual means. Emotional transmission is most effective when it is done via contagious processes. In many situations the writer, speaker, or artist should *necessarily* feel the same emotion he/she need his reader, listener or spectator to have. This is why we are intentionally using the word "*transmit*" in this paper, i.e. "to pass or cause to go from one place or person to another"[11] As it is in the case of knowledge transmission where the transmitter should have the knowledge to be transmitted as a necessary (though not sufficient) condition to transmit it; analogously, to transmit an emotion the transmitter should have the emotion to be transmitted in order to be effective in the transmission process. In this sense effective writers or artists usually have, or first create in themselves, the emotion they want to transmit via contagiousness.

It is not possible to conceive or to imagine how ***apathy*** (without feeling, no-emotion) might generate ***sympathy*** (with feeling, shared emotion) or ***empathy*** ('passion, state of emotion,' from en 'in' + pathos 'feeling'[12]). It is even a "*Contradictio in terminis."* The best way to convey sympathy is to feel it. Otherwise, it would very probably be received as an effort to convey what is not felt. If that happen then problems related to *ethos* might raise, because the lack of credibility that this kind of situation generate. *Pathos is related to ethos*. This relation might be a *reciprocally reinforcing* one or *degrading* one. To be a reinforcing one it should be based on *sincerity*, which includes not trying to convey emotions that are not really felt. This requires, at least, an adequate level of ***emotional intelligence*** and ***appropriate ethos****,* beside a convenient intellectual and/or technical knowledge/experience/intelligence. This is especially required in the application of methodologies that are based on the contribution of several people from different backgrounds, experiences, and probably organizational cultures, as it is the case of systemic methodologies when applied in information systems development and maintenance.

System analysts and information systems managers should create an adequate emotional atmosphere around the users in order to move them to make the required thinking and the necessary intellectual and psychological effort a) in generating the correct requirements and b) in using the system in its beta test phase (which is psychologically and organizationally a painful one). Consequently, they require an *adequate management of the emotional context.* In this sense, analysts and managers need to ***move*** the users to think, to act, and to have reasons to accept painful situations, especially in the beta-test phase. To ***motivate*** the users to be ***moved*** along the phases required by the methodology being applied require the

[11] http://www.thefreedictionary.com/transmit
[12] Online Etymology Dictionary

generation of a minimum of adequate ***emotions*** in the users. This is better achieved if the analysts have, not just ***emotional intelligence,*** but also the capacity of creating adequate emotions in themselves first in order to transmit them via contagiousness. *Analysts and managers empty of any adequate emotions can hardly generate adequate emotions in the users. Analysts and managers with negative emotions can hardly transmit positive ones.* Having positive (or adequate) emotions is a necessary (though not sufficient) condition for transmitting positive (or adequate) emotions in the users. This requires to adequately addressing a ***meta-emotional*** level, i.e. to have to required emotions in order to create adequate emotions in other people collaborating in the project, and to appropriately manage these emotions. This is necessary for any team, but it is more difficult to achieve when we have to adequately relate the emotions of different teams with different knowledge, experience, and cultural[13] backgrounds.

Empathy and ***sympathy*** are among the most important emotional characteristics that should be present on the side of the developers. Since it is not easy to find an adequate combination of technical competence with emotional and meta-emotional capacities, analysts and/or project managers should have this kind of adequate combination or to be aware about the lack of this technical/emotional combination in the team in order to include in it an appropriate facilitator whose function would be to bridge the different parties involved in the project and to pay special attention to the emotional and meta-emotional context.

Since analysts/managers have two-ways communications with the users (not just one way as it is mostly the case of writers and artists), then they should handle their emotions as well as the meta-emotions required to manage the emotions of the users. Analysts must feel the pain that users usually feel in the beta-test of the system in order to increase the probability of their success. Consequently, high level of ***sympathy and empathic capacity*** in system analysts and project managers are desirable for the success of any methodology, and it is probably a must for the effectiveness of systemic methodologies. From our experience we can affirm that there are no effective methodologies when ***apathetic*** analysts, designers, programmers, and project managers are applying it, or parts of it. This is specially truth if the methodology is a systemic one, because its intrinsic adaptability to errors, mistakes, organizational changes, environment uncertainty and entropy, require intellectual, psychological, and managerial efforts from the users which require adequate motivation; which, in turn, depends on ***sympathetic and empathic capacities of the methodological agents.***

We briefly showed above that ethos and pathos are reciprocally related. Both are also reciprocally related to logos in a methodological environment, especially if this environment is a systemic one, i.e. comprehensive, relational, adaptive, and teleologically oriented. The ‘logos’ of any methodological implementation is, by definition and nature, a

[13] We are referring here to different business, organizational, or sub-organizational cultures. We are also referring to the differences that usually exist between technical and non-technical cultures, as well as differences usually existing among different technical or scientific disciplines.

telic one, i.e. oriented by aims, goals, objectives, and purposes. End-means logic is the context of any other logics that might be necessary to be applied in any methodological implementation. Other logics (inductive, deductive, etc.) are, or should be, means in the context of end-means logic when instantiating a methodology; they are not ends in themselves. Any technical logic or tool is absolutely a means, not an end. This is conceptually evident but in our experience it has not been easy to get technical people's thinking/doing habits effectively oriented to this evident methodological truth. Many *know* very well this evident methodological truth but frequently *fail to apply it* while developing information systems. To fail in applying this evident methodological requirement increases the negative impact in the ethos and pathos contexts. It is frequent that programmers and other computing professional forget that *a computer program is a means and not an end. The ends are users' explicit and implicit, disclosed and undisclosed-yet requirements.* Computing processes are means to achieve these ends, and the programmers are, in turn, means to implement the required computing means. Accordingly, we might even conceive that computing professionals and technicians are ***meta-means***, i.e. means to implements means oriented to achieve the users' requirements which are the *real ends* in any methodological orientation.

In this sense, we might say that, conceptually, computer professionals and technicians are engineers, even if they do not have the title of engineers. The most basic concept of "engineer" might be identified in the origin of the term, which derives "from Old French *engigneor*, from Late Latin *ingeniare*; general sense of 'inventor, designer' is recorded from early 15c."[14] The term "engine" derives from "c.1300, 'mechanical device,' also 'skill, craft,' from Old French *engin* 'skill, cleverness,' also 'trick, deceit, stratagem; war machine' (12c.), from Latin *ingenium* 'inborn qualities, talent'."[15] Computer professionals and technicians, as well as their managers, should apply their *ingenium*, ingenuity, ingenuousness, skill, cleverness, in order to design, install, and maintain the necessary computing engine needed for the required information system. The more systemic the methodology being applied the more ingenuity is required, especially to adapt the methodology to the unexpected events, emergent new requirements, unexpected bugs, etc. all of which require emotional intelligence, empathy, sympathy, adequate pathos and meta-pathos, as well as respectable ethos and meta-ethos. *Both ends-means logos and engineering logos require adequate pathos and meta-pathos*. Consequently, logos and pathos are reciprocally related with each other in an information system development methodology especially if this methodology is a systemic one. The more systematic is a methodology the less ingenuity, ingenuousness, skill, cleverness requires. The more systemic is a methodology, the more it requires engineering abilities, in the original/conceptual sense of engineering. Our experience in designing, implementing, and maintaining non-computer-based informing human systems shows us that the same conclusion can be made for *informing/information systems* in general.

[14] Online Etymology Dictionary (italics added)
[15] Ibid.

Relationships of Ethos, Pathos, and Logos in Systemic Methodologies

We have shown above that, in systemic methodologies, Ethos and Pathos are reciprocally related as well as Pathos and Logos. Hence, Ethos, Pathos, and Logos are, directly or indirectly related (in this kind of methodologies) through cybernetic loops, i.e. co-regulative negative feedback (and feedforward) and co-amplifying (and potentially synergic) positive feedback. Figure 5 visualizes the reciprocal relationships and their potential cybernetic loops.

The cybernetic system shown in figure 6 hold, in turn, cybernetic relationships with 1) the organizational context with which the system being developed will interact and 2) the methodological system used while developing the target system. This methodological system includes a set of processes in which ***informing/knowing*** processes are related to ***doing***, through explicit and/or implicit cybernetic loops, as well. *Action-Research, Action-Learning, and Action-Design* are the explicit or implicit thinking/acting meta-methodological contexts usually supporting the application of a methodology. Systemic methodologies necessarily require this kind of thinking/acting contexts and support because, by their nature, they should be flexible, open, and adaptive. A systematic methodology is usually less flexible, open, and adaptive, if not rigid, closed, and non-adaptive. On the other hand, to apply a methodology for the design and implementation of a system to be used by other people require always, explicitly or implicitly, collaborative processes. The more complex is the system the more collaboration it requires in the respective thinking, informing, knowing, doing processes, as well as in the meta-methodologies of Action-Research, Action-Learning, or Action-Design supporting it. On the other hand, the more collaborative is a process, or a set of processes, the more adequate human communication it requires, i.e. the more it requires the support of an updated Trivium, which includes updated Ethos, Pathos, and Logos adapted to the task at hand (Figure 7).

More details regarding the cybernetic relationships that implicitly exist, or explicitly might be implemented among Ethos, Pathos, and Logos, in general and in the special case of Systemic Methodologies, are not among the main objectives of this paper, and require more research and/or practice-based reflections. Consequently, we will discuss these relationships, with more details, in another article.

Conclusions

As main conclusions we might indicate the following:

1. We have shown via verbal reasoning and statistics, related to the *productivity* and *quality* of developing software-based information systems tailored to specific users' requirements, that *Systemic Methodologies are more effective than systematic ones*, though they might be less efficient, i.e. requiring more person-power, managerial time, and psychological energy from both developers and users.

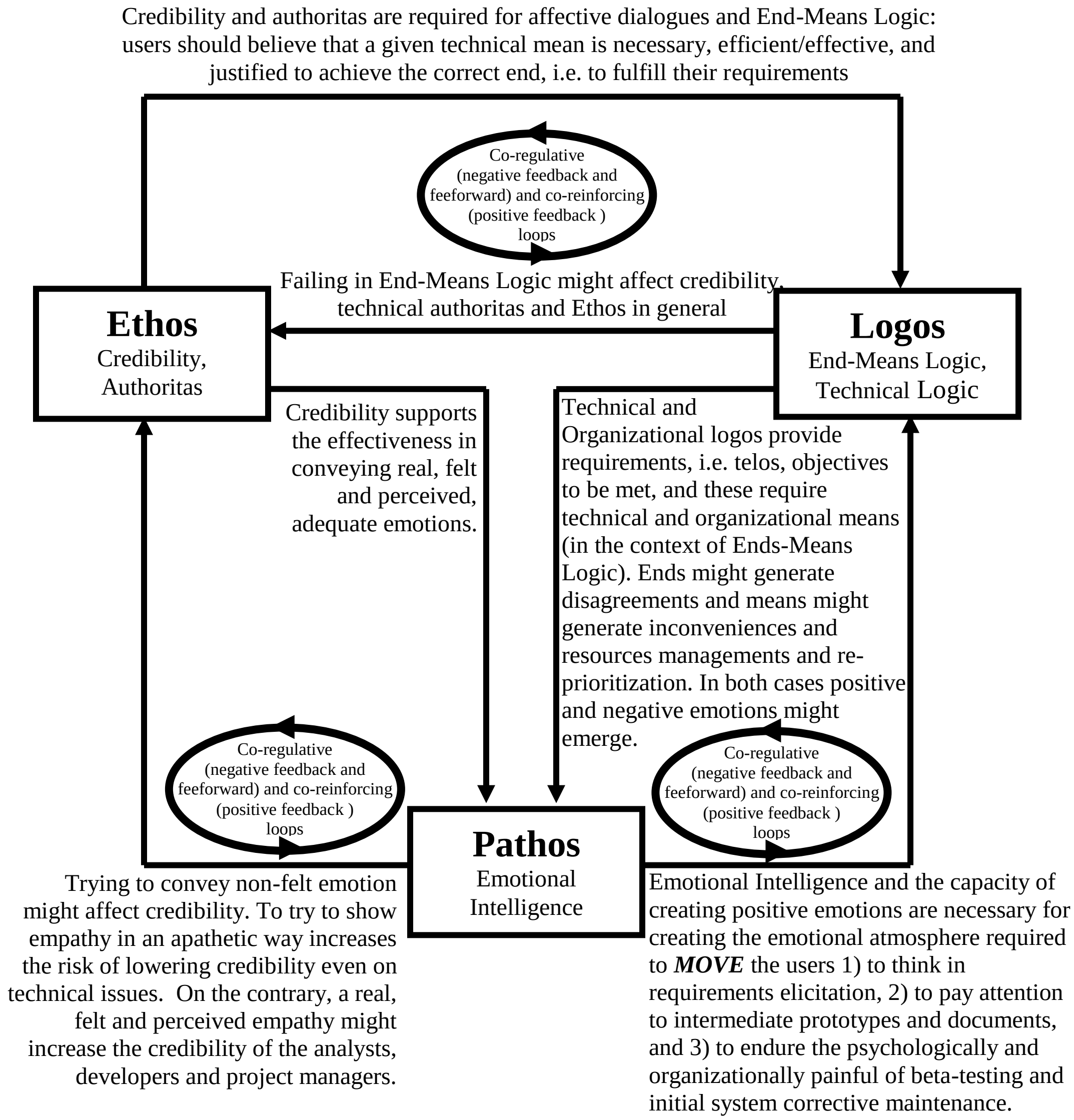


Figure 6

2. We also have discussed and showed, via experience-based verbal reasoning, the huge importance of providing the developers with an updated *Trivium*, in order to improve their skills in handling natural language which is a necessary condition for the effective application of their skills in Artificial Languages, as software and data-base designers and programmers.

3. As part of this updated *Trivium,* people involved in applying a systemic methodology for the development of an information systems and informing processes should adequately handle human communication and, consequently, the associated *Ethos, Pathos, and Logos*.

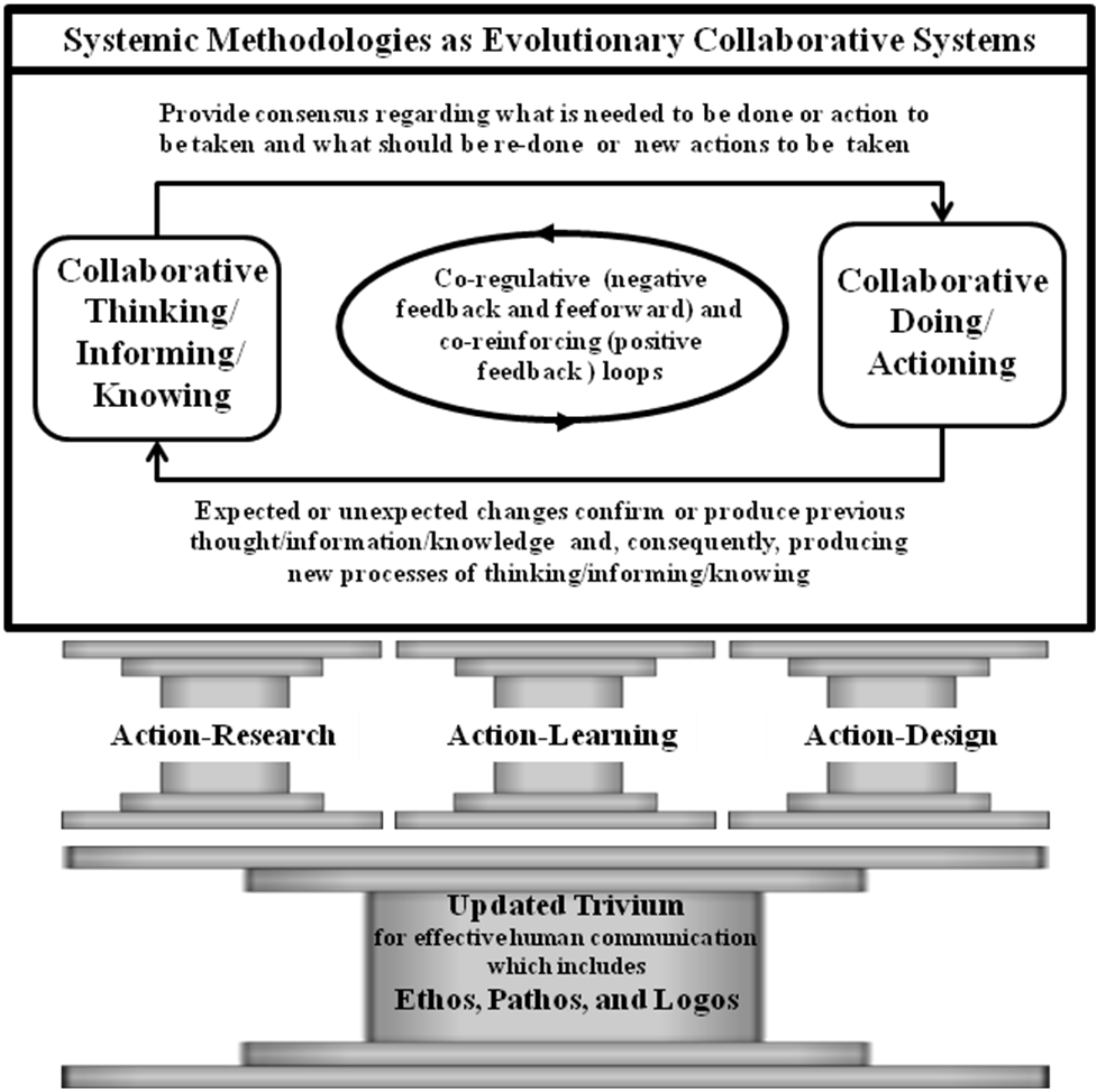


Figure 7

4. Because information systems development requires two main ways communication (actually it requires multiple ways), then there are situations in which developers and/or managers of the development project should also adequately handle the *meta-ethos, meta-pathos, and meta-logos* second level.

We recommend as next research activities or practice-based reflections, the following ones:

a) To analyze with more details via practice-based reflections, action-research, or action-learning the cybernetic relationships that implicitly exist, or explicitly should exist, among Ethos, Pathos, and Logos.

b) To generate research or practice-based reflections with regards the importance, even the ***pragmatic*** necessity, of applying Ethos, Pathos, and Logos in the context of other kind information systems or informing processes such as those in non-software-based contexts, as it is the case of Organizational Development or Change, Public Relations, Inter-National Relations, Inter-Cultural Communication, ***Academic Activities***, etc.

We started working on a project related to Academic Ethos, Pathos, and Logos, mainly based on the findings we have had during 50 years of direct academic activities, or indirectly via managing and organizing them. Consequently, we hope that the our next published article will be a first step in this direction.

References

Ackoff, R.L., 1962, *Scientific Method: Optimizing applied research decisions*. New York: John Wiley and Sons.

Alvarez de Lorenzana, J. M., 1987, "On Evolutionary Systems". *Behavioral Science*, Vol. 32: 19-33.

Alvesson, M. and Sköldberg, K., 2001, Reflexive Methodology: New vistas got Qualitative Research; London: Sage Publications.

Baldwin, J.M. (ed.), 1960, *Dictionary of Philosophy and Psychology*; Gloucester, Massachusetts: Petersmith.

Banathy, B. Jr., 1992, "Comments on Technology Driven Information Systems Design," in L. Peeno (Ed.), *General Systems Appraches to alternative Economics and Values*, Proceedings of the 36th Annual Meeting of the International Society for Systems Science; Denver, Colorado, July 12-17, , pp. 298-305.

Boehm, B., 1981, *Software Engineering Economics* Prentice-Hall, Englewood Cliffs, NJ

Bridgman, P.W., 1927, *The Logic of Modern Physics*; New York: The Macmillan Co.

Bridgman, P.W., 1938, "Operational Analysis", *Philosophy of Science*, Vol.5: 114-131.

Brinkkemper, S., 1996, "Method Engineering: Engineering of Information Systems Development Methods and Tools." In Wrycza-Zupancic (Ed.), *Proceedings of the Fifth International Conference on Information Systems Development* (ISD'96), Gdansk, Poland.

Callaos, N., 1991, *A Christian Approach to Technology*. It is being translated from spanish: *"Una Aproximación Cristiana a la Tecnología"*. Caracas: Universidad Simón Bolívar, Mimeo.

Callaos, N., 1993, "Ethical Considerations for Information and Knowledge Systems Development", *37th Annual Meeting of the International Society for the Systems Sciences*; July 5-9, 1993. Hawkesbury, Australia.

Callaos, N., 1995, *Metodología Sistémica de Sistemas*, Caracas: Universidad Simón Bolivar; 685 pages.

Callaos, N., 2011, *The Notion of Engineering*, accessed on August 24, 2013 at www.iiis.org/Nagib-Callaos/Engineering-and-Meta-Engineering/

Callaos, N. 2013, *The Notion of Notion*, accessed on August 21, 2013 at www.iiis.org/Nagib-Callaos/The-Notion-of-Notion

Callaso, N., 2013, *Knowledge and Cognition*, Accessed on September 15 at www.iiis.org/Nagib-Callaos/Cognition-and-Knowledge/

Callaos N. and Callaos, B., 1991, "A Systemic Definition of Methodology". In *Systems Science in the 21st Century; Integrating the New Sciences of Complexity in Service of Humans and their Environment*, (eds. S.C. Holmberg and K. Samuelson) Proceedings of the 35th Annual Meeting of the International Society for Systems Sciences; Ostersund, Sweden, July 14-20.

Callaos, N and Callaos, B., 2002, "Toward a Systemic Definition of Methodology: Practical Consequences" *Informing Science: The International Journal of an Emerging Transdiscipline*, Volume 5, pp. 1 – 11; accessed on August 24, 2013, at http://www.inform.nu/Articles/Vol5/v5n1p001-011.pdf.

Callaos, N and Callaos, B., 2013, *Toward a Systemic Definition of Methodology: Practical Consequences* (Extended version of Callaos and Callaos 2002), 99 pages; Accessed on August 24, 2014 at http://www.iiis.org/Nagib-Callaos/The-Notion-of-Information/, and at https://www.academia.edu/4415647/The_Notion_of_Notion

Capers, J., 1996, *Applied Software Measurement: Assuring Productivity and Quality*, McGraw-Hill, 2nd edition

Center for Medicare and Medicaid Services, 2008, Office of Information Services, accessed on August 21, 2013 at http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf

Checkland, P., *Systems Thinking, Systems Practice*; Chichester, UK, Wiley.

Churchman, C.W., 1971, *The Design of Inquiring Systems*. New York: Basic Books Inc., Pub.

Corominas, J., 1976, *Breve Diccionario Etimológico de la Lengua Castellana*; Madrid: Editorial Gredos.

Cronholm, S. and Ågerfalk, P. J., 2004, On The Concept of Method in information Systems Development; Örebro university, Department of Computer and information Science, Örebro, Sweden, accessed on November 23, 2013 at http://www.vits.org/publikationer/dokument/52.pdf

Craig, E., (Ed.), 1998, *Routledge Encyclopedia of Philosophy*; London and New York: Routledge.

Etherington, K, 2004, Becoming a reflexive researcher: Using Ourselves in Research, London: Jessica Kingsley Publishers.

Ferrater-Mora, J., 1980, *Diccionario de Filosofía*, Vol. 3. Madrid: Alianza Editorial.

Fink, E., 1968, "Los Conceptos Operatorios en la Fenomenología de Busserl". In Husserl, Cahiers de Royaumont. Argentina: Paidos. Originally published with the same title at Paris: Les Editions de Minuit.

Gasparsky W. and Pszczolowski, T. 1983, *Praxiological Studies*; Dordrecht, Holland: D. Reidel Pub. Co.

Gortari, E., 1984, *Metodología General y Métodos Especiales*; Barcelona, España: Ediciones Océano, S.A.

Harrison, J., 1967, "Ethical Naturalism," in P. Edward (Ed.), *The Encyclopedia of Philosophy*, New York: Nacmillan Pub. Co. Inc. and The Free Press, vol. 3, pp. 69-71.

IBM Systems Magazine, 2012, "Five Areas Influencing Project Success or Failure" included as link in Joseph Gulla, 2012, "Seven Reasons IT Projects Fail Avoiding these pitfalls will help ensure success," *IBM Systems Magazine*, accessed on august 23, 3013 at http://www.ibmsystemsmag.com/power/Systems-Management/Workload-Management/project_pitfalls/

Jayaratna, N., 1994, *Understanding and Evaluating Methodologies*; London: McGraw-Hill Book Company.

Keys, P., 1988, "A Methodology for Methodology Choice". *Systems Research*, Vol. 5, N° 1: 65-76.

Koertge, N., 1987, "Methodology Bootstrapping". In Centripetal Forces in the Sciences, (G. Radnitzky; ed.) New York: Paragon House Pub., pp. 129-137.

Kotarbinski, T., 1965, *Praxeology*. Oxford: Pergamon Press.

Laszlo, E., 1987, "Evolution: The New Paradigm", *World Futures*, Vol. 23: 151-160.

Navarte, C., 1981, *Problemas de Método y Teoría*. Santiago de Chile: Universidad de Chile.

Mingers, J. and Gill, A., 1997, Multimethodology: The Theory and Practice of Combining Management Science Methodologies," New York: John Wiley and sons.

Ritu and Gill N. S., 2012, “A Comparison among Various Techniques to Prioritize the Requirements,” IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 03, Sept 2012; pp.601-607.

Schön, D. A., 1983, The Reflective Practitioner: How Professionals Think in Action, Basic Books, Inc.

Sebastian, F.I., 1972, *Diccionario Griego-Español*. Barcelona, España: Editorial Ramón Sopena.

Skirbekk, G. (ed.), 1983, *Praxiology: An Anthology*; Norway: Universitetsforlaget.

Stamper, R., 1988, “Analyzing the Cultural Impact of a System.” *International Journal of Information Management*; Vol. 8, No. 3

Standish Group, 2010, *Chaos Summary for 2010*, accessed on August 22, 2013 at http://insyght.com.au/special/2010CHAOSSummary.pdf

Stevens, S. S., 1935, "*The Operational Basis of Psychology*", American Journal of Psychology, Vol. 47: 323-330.

The Metaphysics Research Lab, 2003, *The Stanford Encyclopedia of Philosophy*, Stanford University, http://plato.stanford.edu/contents.html#m.

Watson, O.C., (ed.), 1972, *Larousse Illustrated International, Encyclopedia and Dictionary*, McGraw-Hill International Book Co.

Weekley, E., 1967, *An Etymological Dictionary of Modern English;* New York: Dover Pub., Co.

Weinberg, G. M., 1982, *Rethinking Systems Analysis and Design*, Boston: Little, Brown, and Co.

Young, L. I., 1992, “A Jurisprudential View of Information Technology,” in L. Peeno (Ed.), *General Systems Approaches to alternative Economics and Values*, Proceedings of the 36th Annual Meeting of the International Society for Systems Science; Denver, Colorado, July 12-17, , pp. 262-282.