

8-15-1997

# Examining the Differences between Methodical and Amethodical ISD

Jungwoo Lee  
*Georgia State University, jlee@gsu.edu*

Duane Truex  
*Georgia State University*

Follow this and additional works at: <http://aisel.aisnet.org/amcis1997>

---

## Recommended Citation

Lee, Jungwoo and Truex, Duane, "Examining the Differences between Methodical and Amethodical ISD" (1997). *AMCIS 1997 Proceedings*. 113.  
<http://aisel.aisnet.org/amcis1997/113>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1997 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Examining the Differences between Methodical and Amethodical ISD

[Jungwoo Lee](#) & Duane Truex

Department of Computer Information Systems, Georgia State University

P.O. Box 4015, Atlanta, Georgia 30302-4015

[jlee@gsu.edu](mailto:jlee@gsu.edu)

## Abstract

This paper reports on a research program designed to investigate the differences in systems developer's mental models as they develop information systems applying formal system development methods as compared to *amethodical* development.

## Introduction

A central issue in research and practice of information systems development has been that of how to guide the information systems development process such that process outcome will be a high-quality information system. Hundreds of information systems development methods exist at the present time (Hirschheim, Klein & Lyytinen, 1995), each addressing different perspectives in developing information systems (Leifer, Lee & Durgee, 1994).

Despite 30 years of ISD methods evolution and research, the field has not reached a consensus as to what constitutes a good and proper development method or that any single information systems development method is sufficiently general enough to cover all development contexts and contingencies (Wand & Weber, 1995). Nor is there a clear understanding as to how development methods are actually being applied. A small number of anecdotal cases can be found in the literature suggesting that formal methods may reduce creativity and serve only as a social defense (Wastell, 1996), or that even those claiming to embrace formal ISD methods are just "faking" a rational process (Glass, 1994; Parnas & Clements, 1986). In defining the term "software crisis," Veldwijk, Boogaard and Spoor (1994) blame the inherent inflexibility of development methods as the main cause for this software crisis. Growing body of evidence indicates that practitioners, while stating that they follow specific formal development methods, actually do something different due to pragmatic and opportunistic reasons (Bansler & Bodker, 1993; Baskerville, Travis & Truex, 1992).

This 'something' has been researched in many different ways. The generic term of 'Method Engineering' (Kumar & Welke, 1992) has been given to the efforts of fitting methodological components extracted from generally recognized methods to individual development projects recognizing situational contingencies (Brinkkemper, Lyytinen & Welke, 1996). This approach recognizes the ideographic nature of systems development and treats each development project as if it is being custom engineered while depending heavily on the reuse of methodical components. This approach still assumes that ISD is a rational, managed and controlled process. Many proponents of method engineering go so far as to suggest a contingency approach wherein finite set of organizational situations may be mapped to a finite set of methodological solutions (van Slooten & Brinkkemper, 1993).

A more radical form of ideographic system development approach has been described as amethodical development (Truex, Baskerville & Travis, 1997). The term was coined to describe situations where systems development requires more opportunistic, nonlinear, creative and pragmatic approach, which has been largely neglected in methodical approach. This amethodical perspective will be more elaborately contrasted in the next section of this paper. While there are limited anecdotal reports of the other-than-methodical process in the workplace, there are no formal studies that delineate clear differences in the development approaches from the perspective of the ways developers frame and understand ideographic nature of development situations and choose to behave opportunistically. This paper introduces an aspect of

ongoing research, which is intended to explore coexisting methodical and amethodical systems development activities.

## **Literature Review**

### ***Distinction between Methodical and Amethodical ISD***

While ISD research has essentially reified methods, it offers little fundamental understanding of what it means to be methodical and how methods are actually applied in the field. What does it mean for ISD to be methodical? Both Oxford and Webster's dictionaries primarily define the term "method" as meaning "the procedure for obtaining an object." The secondary definitions fasten on such ideas as "orderly," "systematic," "regularity," and "regimen." Method is clearly a concept of process rather than representation. The term "methodical" is the adjectival form. In the mainstream discourse of our field, method is the term used for an orderly, predictable and universal approach to ISD.

"Methods are procedures which came to full flower in the validation of medical cures and in the development of pedagogical procedures and curriculum in the middle ages. Methods began as explanations but became procedures." (Ong, in Coyne 1995 p. 210)

What does it mean for ISD not to be methodical? We use the term *amethodical* to refer to this concept. The term is a construct that connotes an open set of attributes that are essentially not methodical. Thus it does not imply anarchy nor chaos. Amethodical ISD rather implies management and orchestration of systems development without a predefined sequence, control, rationality, or claims to universality. An amethodical development activity is so unique and unpredictable for each information system that even the criteria of contingent ISD methods are irrelevant (Truex, Baskerville & Travis, 1997).

### ***Systems Development as a Cognitive Simulation Process***

Systems development is a series of cognitive activities designed to transform the requirements into the implemented form of an information system. In cognitive science, Johnson-Laird (1983) expanding on Craik (1943), suggests that human problem solvers formulate a mental model of the situation as environmental cues concerning the problem are being received and accumulated. Actual problem solving behavior is a set of simulations that evaluate alternative solutions against this mental model. If this theory of the mental model is applied to systems development, the cognitive activity of systems developers could be described by their mental models of the object system and of any given IS development method.

In clinical psychology, Kelly (1955) theorizing that an individual's perception of the world is represented in the form of personal constructs, and developed the clinical repertory grid technique to extract the construed cognitive structure of psychological patients. Building on this personal construct theory, Garg-Janardan and Salvendy (1987) suggested that the repertory grid technique may enable one to map and measure a mental model.

This research extends these theories into the systems development arena by examining how IS developers construct their conceptual problem space and map the chosen IS development approach to that problem space, especially concerning the effectiveness of methodical versus amethodical approaches to systems development.

In this regard, the aim of this research is to determine how training in formal development methods (1) affects the developer's framing of the conceptual problem space, and (2) by extension his ability to develop high-quality information systems.

## **Research Design**

To investigate the research question above, a controlled laboratory experiment is being conducted. The experimental treatment is the training (or lack thereof) in formal ISD methods. The treatment group (or the methodical group) consists of a group of students who have completed a system analysis course and are taking a systems design course at an urban university in the southeastern United States. This group follows a prescribed set of systems development methods that they are learning in the classroom. The control group (or the *amethodical* group) consists of the students who are taking an end user application programming course at the same university. Because of the sequential precedence of prerequisites, the students in this class have no exposure to formal information systems development methods at all, except to the degree it is implied in the programming language that they are learning.

All subjects are required to perform the same task: develop a prototype information system. The methodical group is required to use of various structured analysis and design techniques and produce prescribed deliverables during the project. The amethodical group will receive no methodical intervention at all. To ensure that possible differences in performance are due merely to presence of the treatment, both groups will be required to submit comparable intermediate deliverables. The experimental treatment is designed to be a 'between-subject' design with measurements being taken repeatedly from each subject.

The developer's mental model is established by measuring (1) perceptions that each individual has about the project cases and (2) the priorities that the person holds about the various components of the systems to be designed. This is done using repertory grid technique. The repertory grid technique uses a grid format two-dimensional table consisting of domain elements on each column and personal constructs on each row. The similarities and differences among the identified elements are evaluated against the personal constructs identified in each row. To ensure the reliability and validity in applying this clinical psychology technique to systems development, a two phased approach will be used.

At the initial phase, systems development experts will participate in a brainstorming session to identify a set of desired system components. Following Easterby-Smith (1980), seven to twelve elements will be selected given a case for systems development. Two general criteria in selecting elements are homogeneity and coverage. Elements should be drawn from the same category or domain and should cover overall range of problem domain. Inter-rater comparison statistics (Cronbach Alpha) will be used to ensure the agreement of selections among experts. In subsequent sessions, each expert and the researchers will specify constructs against which each element will be evaluated and assigned to a 10-points Likert scale. By compiling these experts' models, a standard repertory grid with standard value set common to experts will be established and used as a baseline measure for second phase. To confirm the results of repertory grid technique, this panel of experts will provide one additional measure: an importance rank ordering of system components.

In the second phase both treatment and control groups will complete the standard repertory grid as previously described both before and after the experimental treatment. These results will be analyzed for (1) differences and similarities between the two groups' cognitive structure, (2) the changes in each group, (3) their proximity to the expert ratings established in the first phase, and; (4) in turn the influence of these identified dimensions on the quality of the prototype information system.

Figure 1 summarizes the research model, which illustrates how the developer's cognitive structure is expected to mediate the relationship between the formal training (the methodical approach) and the quality of the produced information system.

## **Implications and Significance.**

Experiments on aspects of information systems, such as user interfaces, programming efficiency, and the like are common place. But laboratory experiments concerning the use of development methods are rare. We know of no laboratory experiments testing the effectiveness of the methodical versus amethodical approaches to systems development. This research makes a contribution by providing empirical evidence of the principal characteristics of methodical versus amethodical aspects of information systems development.

This research is expected to offer insights on the advantages and disadvantages of methodical approaches or conversely, those of amethodical approaches.

Secondly, it is the first research on the cognitive structure of systems developers. Though research on the mental model has proliferated in user interface design in relation to the representation form (Johnson-Laird, 1985), this is the first try at examining the influence of training on development methods on the mental models of systems developers and consequently on the quality of the produced system.

Third, for the practice of systems development, the identification of "amethodical" components in information systems development has the potential to change the whole view of the field. System developers may have to reexamine the traditional practice of information systems development in view of amethodical components that have been previously neglected. This reexamination, in turn, will help to improve the new practice of amethodical development, especially in rapidly changing and emergent contemporary organizations.

Finally, a validated instrument of developer's mental models will enable managers to better match developers to development settings and project teams. By knowing something of the developer's mental models, project managers may decide whether to co-assign developers with similar or different mental models depending on whether he or she wishes to construct teams oriented towards conflict or towards consensus.

A final outcome of this research will be a clearer understanding of how novice developers perceive the essential components of the target system. As many authors are predicting a substantial shift to end-user development environments where end-users simply select and combine preexisting system components, this knowledge of how novice developers cognitively construct model of target systems may prove useful to componentware builders as well as end-users themselves.

## **References**

**Available upon request (contact [jlee@gsu.edu](mailto:jlee@gsu.edu)).**