

Research

Developing end-users' systems development competence

An exploratory study

Ojelanki K. Ngwenyama *

University of Michigan, Ann Arbor, MI, USA

This paper presents an approach that will increase end-user competence in systems development by a method based on *Collaborative Action Learning*. The primary objective of the approach is to provide a structure and process in which end-users, in real-world situations, can continuously develop explicit as well as tacit knowledge of systems development. The results described here are from an exploratory study in a major research project that has, as its primary objective, the provision of an end-user collaborative action learning systems development methodology. The lessons learned from two action research case studies are summarized.

Keywords: Action learning; CASE; Collaborative work; End-user computing; End-user systems development; End-user training; Information systems; Participative systems development; Training

Correspondence to: O.K. Ngwenyama, School of Business Administration, University of Michigan, Ann Arbor, MI 48109-1234, USA.

* This research is part of an ongoing project – The Program for Research on User Developed Information Systems – supported in part by The School Business Administration, and The AMERITECH Program For Information And Organizations, The Institute For Public Policy Studies, The University of Michigan, Ann Arbor. The author thanks the three anonymous reviewers, Ruby Beale, Michael Cohen, and Poppy McLeod, for their critical comments on earlier drafts of this paper, and Edgar Sibley for his many useful editorial comments.

1. Introduction

The proliferation of personal computers, increasing demand for information processing, huge development backlogs, and user dissatisfaction have contributed to the phenomenal growth of end-user systems development in the last decade. During this same period, a large number of “user friendly” fourth generation tools and application generators have been developed and marketed with promises of improving development productivity, and easy use by end-users [19,30,34,39,53, 55]. Many companies also invested significant amounts of resources in information centers, end-user training, and end-user support personnel without realizing adequate returns. A great deal of these resources have been invested in identifying appropriate tools for end-user systems development (ESD), [33,35,38] and strategies for end-user training [7,8,18,44,47,48]. However, im-



Ojelanki K. Ngwenyama is Assistant Professor of Computing and Information Systems with the School of Business Administration and the Institute For Public Policy Studies of the University of Michigan, Ann Arbor. His academic training includes a Ph.D. (1988) in Computer Science and Information Systems from the Thomas J. Watson School of Engineering, State University of New York – Binghamton, an MBA from Syracuse University, New York, an MS in Informa-

tion Systems from Roosevelt University, and baccalaureate degrees in computer science and engineering. Dr. Ngwenyama research focuses on understanding how people in everyday social activity appropriate and innovate on information technology to solve relevant problems. His research is reported in journals such as *Accounting, Management and Information Technology, Information Systems, ACM Transactions on Information Systems, and IFIP Transactions*. He is a member of IFIP WG 8.2, and TIMS College on Information Systems.

provements in end-user systems development productivity have not matched expectations [49], neither has systems standards and quality [26].

Several researchers have suggested that new approaches to end-user training need to be developed [12,16,21,23,27]. The problem most often identified with traditional approaches is their focus on construct based training, which emphasizes the mastery of software tools. Construct based training according to Karten [25] teaches the user how the software works, but not how to use it. Olfman and Bostrom [46] have found that action oriented hands-on training is most effective in developing ESD skills. Further, Cheney et al. [11] have found that organizational and work-group support influence end-user learning. Another key determinant to learning that this type of training lacks is motivation [9,28]. However, it appears that traditional approaches to end-user training have ignored research findings [41,61], which have demonstrated the importance of: (a) personal relevance, (b) learning environment, and (c) instruction style. Much work needs to be done in developing methods, tools, and strategies that focus on the special problems of end-user systems development.

This paper discusses a *Collaborative Action Learning (CAL)* approach to empowering end-users with the necessary skills to find solutions to their information systems needs in an ongoing fashion with minimal external help. The CAL approach offers a new way of conceptualizing end-user systems development: (1) as an Action Learning process; and (2) as a self organizing collaborative discourse. Its primary aim is to provide a structure, and set of development and process management techniques that allow end-users to improve their competence in real systems development situations.

2. The competence problem

The problem of end-user competence is well documented in the literature. Suchman [59] and Knolmayer and Disterer [29] have found that the lack of analytical and conceptual skills inhibits end-users from effectively transforming and representing their working knowledge of the organizational environment into a computer process-

able form. In two different studies of end-user spreadsheet application development, Brown and Gould [4] and Carlson [6] observed that end-users have difficulty in analyzing and representing formulas that are part of the information structure of their work environments. Canter et al. [5] also found that end-users have difficulties conceptualizing information needed to navigate and extract data from complex databases. Many researchers have pointed out that expert problem solvers depend heavily on a vast reservoir of tacit knowledge that they have developed through work experience [50,57]. Many of the conceptual and analytical difficulties that end-users encounter during systems development can be traced to the lack of experience-based knowledge. A key issue of any approach to developing end-user competence is the structuring of enabling conditions for developing knowledge about experience-based systems development.

2.1. Aspects of systems development competence

The expertise of competent systems developers can be classified into two categories: (1) knowledge of the organization and (2) knowledge of technology [13,22]. Each of these may be further divided into implicit background or ready-to-hand knowledge and explicit foreground or present-at-hand knowledge [14,36]. These categories can be defined as a coordinated schema, comprising a set of goal directed scripts that contain functional subparts for problem identification, problem framing, and problem solution in specific areas (organization, technology, etc.). The problem identification part might consist of knowledge of the salient features by which a class of systems development problems (e.g., information modeling) can be identified and categorized, while the problem framing and solution parts might comprise the appropriate strategies for making sense of the context of the problem-domain and of the solution procedures for analyzing and debugging specific problems: See *Figure 1*.

Implicit background knowledge consists of inarticulable routines, conceptual skills, and tacit and intuitive knowledge acquired from practical experience [45]. Although most implicit knowledge cannot be logically articulated or explicitly represented, it enables the competent systems developer to innovate when faced with novel

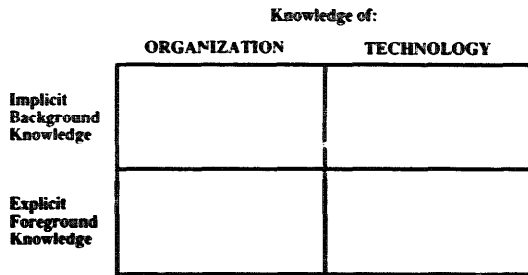


Fig. 1. Aspects of systems development competence.

problem situations [54]. In recognition of the importance of this aspect of competence in systems development, information systems faculty insist on *case work* for students studying programming, systems analysis, data base design, and other similar coursework. Several researchers have also pointed out the importance of this type of knowledge to successful system development [5,13,14,36]. Suchman and Wynn [59] and others have discussed the problems encountered by end-users who try to develop their own systems without adequate experiential knowledge.

2.2. Research approach

This work is of an exploratory nature; it has served as a pilot study in a major research project on end-user systems development. The fundamental conjecture of our research is that there are systems development skills that cannot be acquired except by experience. We wish to construct an ESD methodology that provides the framework and enabling conditions for a collaborative, continuous learning process to aid end-users during their systems development. The philosophical orientation of the research project can be summarized as: (1) Developing "action oriented knowledge" of ESD. (2) Grounding our knowledge claims in ESD practice. (3) Encouraging free and open participation by end-users. For these and other reasons, we have selected an action research approach for testing, validating, and improving the effectiveness of our methodology. This approach is not new. Several studies have adopted it, most notably Checkland [10] and Wood-Harper [64] who have argued that there is no alternative for studying methodologies. According to Land [32], "we have no choice but to go out into the world and try them (techniques and methodologies) out."

3. Conceptual foundations

The theoretical framework of the CAL approach is rooted in Holographic Organization theory [42], Action Learning theory [52,43], and Communicative Action theory [20]. Its design ideals are: (1) action learning as a valid form of end-user systems development practice; (2) communicative action as a basis for group interaction during the development process; (3) heterarchy, shared authority, and responsibility as a basis for meaningful collaboration and learning; and (4) continuous learning as a strategy for improving the competence of end-users and the quality of the information systems that they build.

3.1. Holographic organization theory

Holographic organization design rests on four basic principles: (1) requisite variety; (2) redundancy of functions; (3) minimum critical specification; and (4) learning to learn. *Requisite variety* implies a diversity of knowledge in the work group: it must match the variety and complexity of the organizational context in order for it to deal with the challenges posed by the context; i.e., critical organizational knowledge and skills must be understood by the work group. *Redundancy of functions* requires that each member of the work group be able to carry out several functions and that cross training will occur throughout the work group. Each team member should be a generalist, having multiple skills so that s/he can perform each other's function and substitute for one another as the need arises. The general idea is to develop a pattern of overlapping skills and a knowledge base that enhances the capacity and flexibility of the work group for continuous and effective self-organization.

The principle of *minimum critical specification* requires the infrastructure of the work group to provide "enabling conditions" so that the group can function as an "inquiry-driven action team." This implies a heterarchical structure in which there is shared responsibility and ownership of the work group products. Roles are not permanently differentiated, they are clarified from one problem-situation to another. Each problem situation allows different group members to make contributions according to their ability. This keeps the work group flexible, diversified, and emer-

gent. Coordination is a collaborative process and it is supported by a functional responsibility matrix. *Learning to learn*, the fourth principle, refers to the work group's capacity for single and double loop learning. Members of the group must both value their work and the products they produce, and remain open to learning new skills that would enable them to deal with new challenges in the organizational environment. Since there are few predetermined rules for guiding behavior, direction and coherence must come from the shared values and norms that drive the group.

3.2. Action learning theory

The fundamental conjecture of *Action Learning* is that people learn most effectively by doing. It is concerned with developing modes of social practice that enable individuals and groups to learn continuously as they deal with everyday situations. The action learning process combines: programmed instruction (P); questioning, critical inquiry, and experimentation with unfamiliar problem situations (Q); and critical reflection (R) on the Q experience [53,58,61]. Action learning strategies connect direct participation of those affected by an organizational issue with effective ways of dealing with it, and seek to facilitate the on-going creation and invention of means that enable a high level of organizational competence and effectiveness which cannot be achieved by traditional training methods alone. The action learning process is a continuous cycle of searching for new and improved ways of dealing with problems via critical reflection on past experience, deciding on and planning new approaches

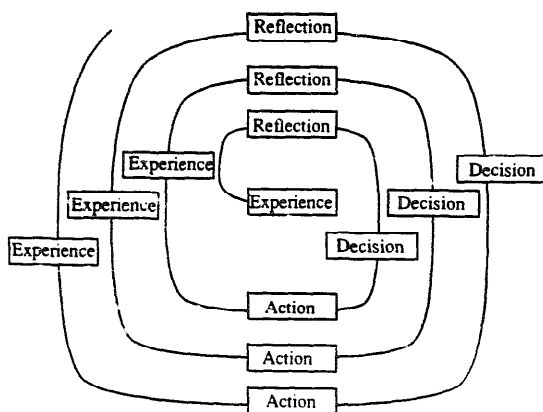


Fig. 2. Action learning.

and strategies, putting them into action, experiencing the results, critically reflecting on them, etc. (See *Figure 2*).

Three basic principles define the action learning approach: (1) The importance of co-determination in resolving problem-situations that involve multiple stakeholders; (2) Normative concerns are intertwined in the collaborative process [40,65]; (3) The action learning process is a collaborative relationship in which *experts* and *lay-people* hold learning in common trust as a basis for the resolution of the problem-situation.

3.3. Communicative action theory

Communicative Action is concerned with the achievement of effective collaborative action via free and open discourse (i.e., critical debate). It provides a framework for the group interaction processes in which participants can debate relevant issues and hammer out agreements for joint action. A set of ideals that serve as general rules for the process are:

1. All participants must have equal status in the group process.
2. All participants must have equal opportunity to raise issues, challenge, or defend the validity of all actions or statements.
3. All communications must be clear and understandable. No form of jargon may be used to mystify or erect barriers to communication.
4. All propositional statements must relate to the existing state-of-affairs.
5. All statements and actions must be appropriate to the situation under consideration.
6. All participants must say what they mean and take the action that is mutually agreed.
7. All proposed actions will be meaningful and effective for achieving the intended goals.

It should be clear that this approach assumes Theory Y behavior by the participants. For the debate to work, each participant must share the same seven values. Although there is no intention to monitor the process for adherence to the "rules," the communicative action process requires that each claim made by a participant must be considered in an open and honest debate, and that the claim may stand or be rejected based on four criteria: (1) *truth*, the state of affairs does exist; (2) *legitimacy*, the statement or action is

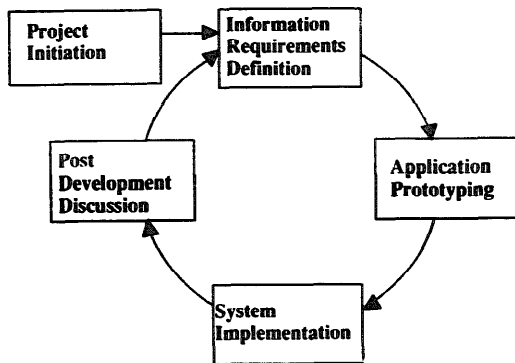


Fig. 3. The CAL development life cycle.

appropriate within the context; (3) *sincerity*, the actor really means what s/he says or intends to act as promised; (4) *comprehensibility*, statements and actions have clear and coherent meaning.

4. The CAL development process

The CAL approach to end-user systems development joins together the principles of self organizing, action learning, and communicative action into a form of practice in which the participants collectively engage in a continuous cycle of experiential learning (See *Figure 3*), as they search for and implement solutions to real-world information systems problems. Critical debate is the primary technique for challenging the taken-for-granted, breaking down organizational myths and mysteries, as well as exposing the limitations of the learning process [2]. Critical debate is a form of “public reflection” or “auto-critique;” the emphasis placed on it is based on the recognition that solutions depend on how we engage in learning about problem-situations.

Critical debate is a part of every phase of the CAL development process. It is through this process that information processing and management alternatives are surfaced, carefully analyzed, and change options prioritized and implemented. For example, let us suppose the analysis of the work process is conducted in a critical debate about current operating practices. The team would examine the current practices, rules and procedures for effectiveness, efficiency, legitimacy, and relevance and search for alternatives that could improve them. For decision making and priority setting, the *Strong Preference Model*

(SPM), [31] is used. In this approach, each team member is required to state whether s/he has a strong preference for or against a specific outcome of the decision making process, or whether s/he cares only that a decision is made. The idea is to shift the decision making process away from superficial issues and power interactions to a process of real collaboration for problem solving. The following conditions must be fostered: (a) agreement that learning from experience is an important norm of the CAL process; (b) patience for the development and transition through false consensus; (c) agreement that authority and responsibility is always shared; (d) the possibility that learning and change initiatives are evolutionary.

4.1. Project initiation

Team selection is carefully carried out in accordance with the principle of *requisite variety*. The objective is to match the internal diversity of team knowledge with the realities of the organization. The basic assumption is that each team member brings to the process specific aspects of organizational knowledge relevant to ESD. Three team roles require special mention: coordination, control, and documentation. These are functionally differentiated and rotate among the team members. A *team facilitator* keeps the process going, coordinates the various activities, and maintains the functional responsibility matrices. A *documenter* records the minutes of the meetings and details of agreements. A *technical consultant* provides support in the use of the iCASE tools and runs the application generation process. The last of these is initially filled by an IS specialist, but it is taken over by other members of the team when they become proficient with the software tools. The objective of programmed instruction is to provide basic training in organizing, process, systems analysis, and information technology. The programmed instruction focuses on three areas: (a) group interaction, (b) systems concepts, i.e., defining root definitions and conceptual modeling, and (c) prototyping in the iCASE environment. The group interaction and systems concepts training is an intensive three-day programmed learning and participatory workshop that focuses on the CAL process. The idea of collaborative end-user systems development is in-

troduced and discussed and group interaction skills are taught. The skills training encompasses communication, systems concepts, decision making, idea generation, and collaboration. The iCASE tools training is conducted as a two week "hands on" workshop, limited in focus, covering issues relevant to prototyping; that is, business systems description and analysis, business systems design, and application generation. Case studies are used to demonstrate the concepts.

4.2. Information requirements definition

The objective of this Phase is the development of a clear and concise statement of the system requirements. The sequence activities of this phase are the development of: (1) the Root Definition (RD) of the business system; (2) a Conceptual Model (CM) defining the business activities and supporting information per the RD; and (3) the Information System Definition (ISD). The process starts off with a discussion of the stated goals and objectives of the business system. The *facilitator* records each participant's understanding of these goals and objectives on a flip chart, then opens up the subject to debate and clarification. A consensus statement of the goals and objectives is hammered out in everyday language. This, then, becomes the basis upon which the RD is developed. Development of the CM is a two step process: (1) Definition of the business activities. (2) Definition of the information to support the activities.

The CM serves as a normative model from which the existing business system is critically analyzed to identify operating activities, procedures, and rules that may be irrelevant, ineffec-

tive, and inefficient. Examples of the four classes of questions considered are:

Relevance questions

- Are the operating procedures of the system outdated because of changes in the business environment?
- Is there a need for new policies and procedures?

Legitimacy questions

- Are restrictions within the business system based on policy or organizational norms?

Effectiveness questions

- What changes can be made to improve the effectiveness of the system?
- Are the appropriate resources available to ensure that procedures can be carried out as designed?
- Is there a need for new procedures?

Efficiency questions

- Are the operating policies and procedures optimal for the current business environment?
- Are there more efficient ways of conducting the business?
- Is the design of the business system cost efficient?

4.3. Application prototyping

The application prototyping phase is organized around the spiral model of software development and enhancement [3] and supported by iCASE tools. It has four activities: Application Design, Generation, Testing and Critique, and Modification (See *Figure 4*). The focus is on well bounded organizational activities that can be implemented effectively to support the work of the users. Such applications have been defined as *vertical prototypes* [17]. We are not interested in prototyping for requirements definition or developing throw-away applications.

Application Design: Using the requirements statement, the team decides what business activities they want the information system to support, what functions it must perform, and the order in which the functions will be developed. The design starts with the specification of the formats for input and output, and user interaction dialogues. The data are then specified, along with the operations.

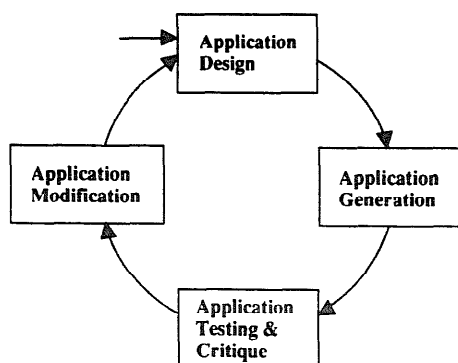


Fig. 4. The application prototyping cycle.

Application Generation The application is generated using iCASE tools. The *Testing and Critique* activity focuses on verifying the design and identifying inadequacies and difficulties in usage. Samples of data are selected and used for testing. A critical debate about system usability is conducted, and suggestions for improvements are documented. The *modification* activity is concerned with deciding what parts should be re-designed and how it should be extended to accommodate improvements. The cycle is then repeated until the team has developed, tested, and approved an application that is acceptable for everyday use.

4.4. Implementation

Implementation involves a smooth transition into the general work activities. As a part of this, the team develops training, operating, and problem documentation manuals; workshops are conducted. These cover the basics as well as issues of user responsibility.

4.5. Post development discussion

This involves "public critical reflection" upon the process and product of the exercise. It is the "R" of the action learning process and focuses on two issues: (1) What did we learn as a group? (2) What would we like to improve in the process and product? Using a flip chart, the team members itemize and discuss the problems they encountered. The development process is then discussed and the members are asked to suggest changes that might make it more effective. Suggestions include possible extensions and other activities of the business function that might benefit from future development. The *documenter* records the findings and produces a report that is circulated for approval and finalized.

5. Background of the case organization

ALPHA Inc., the host of our case studies, is a large multi-industry conglomerate that is almost 300 years old. It began as a small family business trading with native North Americans. The firm is currently organized into six companies involved in: mining, manufacturing, petroleum, financing,

real-estate development, and retail sales. The retail sales company operates an innovative consumer oriented department store chain, which is concerned with staying ahead of its competitors. In its merchandising operations, it seeks to foster a culture of dynamism and change, rewarding its employees for new ideas. The business is functionally organized into: Marketing, Purchasing, Finance, Operations, and Information Resources Management (IRM). Each function reports to the Chief Operating Officer via a Vice President. The IRM function is regionally distributed with a small centralized group comprising Strategic Planning, Information Systems Architecture, Head Office Systems, and Procurement. Head Office Systems is composed of: Systems Development and Maintenance, Operations and Technology, and End-User Support.

In 1989, having completed an enterprise-wide information engineering project to define an Information Systems Architecture and long term information management strategy, ALPHA INC. decided upon end-user computing as a strategy for developing small to medium size information systems. The Catalogue Sales (CS) department of the Marketing function located at head office was targeted as a key area for EUC. From 1984 to 1989, catalogue sales had grown dramatically. This coupled with a high projected growth over the next five years, provided the impetus for top management to make the development a strategic issue. Two main areas were deemed critical: Customer Service and Distribution Management. In 1989, the CS department consisted of a manager reporting to the VP of Marketing, three supervisors, and 32 other personnel. During the period of high sales, little attention was given to the CS department; consequently, it was given the dubious distinction of the least desirable place to work with the highest personnel turnover rate. In September 1989, only one CS employee had more than 5 years experience in the area; the others ranged from five months to two years.

5.1. Alpha's information systems infrastructure

By September 1989, the information systems development infrastructure had become very sophisticated. The Information Systems Architecture Group had completed the second phase of a study that defined the long range plan, an infor-

mation systems architecture, and a ten year development and implementation program. In support of this major strategy initiative, the IRM organization was expanded to include Data Administration, End-user Support, and CASE Support Groups within Head Office Systems. A second important aspect was the integrated Computer Aided Software Engineering (iCASE) environment; it was configured around personal computer work stations linked to a mainframe repository with systems descriptions managed by a Central Encyclopedia.

5.2. *The case studies*

Case No. 1: Order processing

The first candidate of the CAL approach was to provide information technology support for the effective and efficient telephone order processing, and contribute to improving customer and employee satisfaction. The order processing function had long suffered the consequences of a lack of timely information on product availability, pricing, and customer in-house credit status. The order entry information system (OEIS) was a stand alone system supporting on-line inquiry of customer and inventory data and overnight batch processing of orders. The customer and inventory data were almost always out of date and invalid due to irregular file maintenance. As one employee said, "if we really want to know what products are in stock we call the warehouse . . . and as for price, the customers get that information from advertising before we do." Customer service employees, therefore, kept "buffer stocks" of inventory information they received from "contacts" at the warehouse, and copies of the catalogue to which they refer while talking with customers on the phone. Further, the system offered no facility for checking the credit of customers; this information was managed only by customer accounts.

Case No. 2: Distribution management

The proposed distribution information system (DIS) was to support the management and operation of the CS function. Distribution personnel are responsible for shipping and tracking customer orders and returns. Through the years, shipping and tracking orders had become a logistical nightmare, mainly because of the lack of

valid inventory status information and shipping coordination difficulties. The distribution policy, at the early stages, stipulated that orders would be filled from two major warehouses located just outside the city where the Head Office was located. Using the information generated by order processing, distribution personnel conferred with people in the warehouses to arrange shipping. The mode of shipping is selected based on the size of the order, the customer's address and preference. There are several options and classes of shipping service in addition to the firm's delivery service. In cases where the customer has no stated preference, distribution personnel must select the most cost effective and reliable service. In order to do this, they must have information on: (a) the warehouse with items closest to the customer location; (b) delivery schedules for the in-house service; and (c) pick-up and delivery schedules of outside contractors.

5.3. *The CAL development projects*

The project teams each consisted of eight volunteers from the CS group, the researcher, and a technical expert on loan from the Information Systems Architecture Group. Two members of the OEIS development team had previous experience of one year and nine months as user representatives assigned to systems development teams. The experiences of the other volunteers ranged from four years to five months within the CS department. During the development of the OEIS six temporary part-time employees were hired to fill gap created by the volunteers. At the end the project two were retained, they continued working beyond the completion of the DIS project. Acquiring these additional resources was not problematic, because it was usual practice for the CS department to hire temporary help every few months, when they were overloaded.

The development time for the OEIS was six weeks, exclusive of the *Programmed Instruction* period. The project team traversed the application generation cycle three times in order to accommodate the modifications and improvements. The system encompasses a hierarchy of input and display screens, and application modules which accesses three databases and creates an orders database. It now supports five processing activities: (1) creating and updating customer

records; (2) checking customer credit; (3) checking inventory status; (4) checking catalogue data; and (5) entering orders and checking order status.

The development of the DIS was less of a problem: success of the OEIS significantly quelled opposition, gained respect for the end-users, and gave legitimacy to the CAL process. In a more cooperative environment, the DIS project team was able to develop a working system within three weeks with two iterations of the CAL cycle. However, this system has since gone through several more iterations, completed by the users, to expand its functionality. It was designed around a hierarchy of four input/output screens, three hard copy reports, access to the catalogue items, inventory and orders databases. In addition, a shippers' and distribution tracking database was created.

5.4. Database administration issues

Several database administration issues emerged; two issues had significant impact on the process: (a) obtaining on-line access to the inventory database, and (b) designing and implementing catalogue items and catalogue sales customer databases. During the CAL development process, it was discovered that the enterprise-wide information engineering study identified neither a catalogue sales customer nor a catalogue sales items database. This discovery proved to be very embarrassing for the ISA group, and set in motion a great deal of political activity which revealed factions, differing interests, and conflict. One faction, those responsible for modeling the business, launched an offensive challenging EUC initiative, arguing that it was ill-timed and that it should be postponed until they could better support it. The DA group, on the other hand, readily agreed to allow read-only access to the inventory database and tried but failed to gain control of the EUC project. After much discussion, they accepted responsibility for and implemented the catalogue sales customer database. Top management negotiation at the VP level was required to get a catalogue items database developed and implemented. The negotiation process almost fell apart when a schema for the catalogue items database was generated by the iCASE system, because it could not be found in the Central Encyclopedia.

6. Discussion of key issues

The following discussion is based on data collected on end-users' perceptions about the effectiveness of the CAL process in supporting the development of their competence for developing information systems. It comes from: (1) daily diaries that participants were required to keep, and (2) audio records and *Documenter's* notes of post development discussions.

6.1. End-users' perceptions

The diary logs served as biographies of the learning experience, material for reflection, and a source of issues during the *Post Development Discussions*. Three common concerns emerged:

- The length of programmed instruction.
- Technical support.
- The structure and management of the project team.

In their early entries, participants noted that the programmed instruction was too short. When this issue was raised in the *Post Development Discussions*, the group members agreed that they felt unprepared to tackle the development of a "real system" up until they completed the first iteration.

Also, all participants felt that communication with the technical support consultant was a major problem. The general feeling was that he did not understand their questions. They felt that, because of the communication problem, they experimented too much with the process and iCASE tools.

Entries made early in the development process raised issues about leadership, power, and accountability. The team members seemed to want a clear distinction of "lines of authority." With regard to power, they wanted a mechanism for enforcing commitments. However, later diary entries (second and third weeks of development) shifted focus, putting more emphasis on how to make the structure and process work more effectively. During the *Post Development Discussion*, the participants expressed very different views. The reason for this shift in perspective was that they had been unprepared to make the transition from "being told what to do, ... to having the full

responsibility of deciding and doing what was to be done." They argued that it was their joint responsibility for the final product that gave them the impetus to keep trying when there were difficulties.

Ironically, the very features of the CAL process that were criticized were the ones that were praised. The programmed instruction was praised for its orientation and clarity. They felt that with more practice, they would have increased their confidence both with the CAL process and in the use of iCASE tools; this would enable them to communicate more effectively with the technical specialist.

They felt that there was a qualitative difference between working on the case studies and the experience of developing an actual system that they would later use in their daily work. It was clear that the responsibility for developing a "real system," along with the need to convince the "systems people" that they could do it, contributed to both the learning process and group cohesion.

The group process, although criticized, was cited as "enabling." They argued that the open and public nature of the discourse helped to reinforce learning and information sharing. The general perception was that the group process, and especially the rotation of role responsibilities was catalytic to mastering the systems development and process skills and learning about the organization. Three key issues were: (a) overcoming the fear of "sounding stupid," (b) controlling the length of discussion meetings, and (c) enforcing commitments.

6.2. Implications for management

Three important sets of management issues which emerged are: (a) systems development policy, more specifically the relationship between "professional" systems development and end-user systems development; (b) infrastructure, the appropriate IRM architecture, software tools, and technical support; and (c) the institutionalization of the CAL process and maintenance of an adequate level of end-user competence. A fourth issue concerned career paths for end-users. Clearly, the major systems development policy issues concern the division of responsibility between end-users and IRM. Responsibility for data

architecture, data base design, systems integration, quality control, and maintenance of end-user developed systems needed to be defined with clear management policy and procedures.

The infrastructure, Information Systems Architecture, and iCASE tools, were crucial factors to success. The iCASE environment with its user friendly interfaces, powerful modeling tools, and code generation capabilities enabled end-users with limited training to develop useful, high quality systems within very short time frames. The fact that the CAL approach was structured around specific organizational requirements (policies and resources) also significantly increased the chances of success. Several questions need to be considered when selecting or constructing methodologies: (1) What is the nature of the problem to be addressed? (2) Who is the user of the methodology? (3) What is the nature of development environment? (4) What kind of software tools will be used?

7. Concluding comments

Several practical lessons have been learned and/or re-enforced. An organization needs: (1) end-user systems development policies that define the responsibilities of users and the IRM function; (2) clear and definitive management commitment to the process; (3) systems development standards and an unobtrusive way of enforcing them; (4) a reward/incentive system which encourages end-user systems development; (5) a methodology and work environment that is supportive to learning by doing; and (6) appropriate software tools [1,15,24,55].

This research makes some theoretical contributions: (1) It demonstrates the viability of action learning as a basis for end-user systems development. (2) It offers a viable strategy for dealing with the problem of developing end-user computing competence. (3) Insights have been gained on the application of alternative team structures for systems development. However, although this research has helped to push beyond traditional thinking on ISD and has opened up new areas of inquiry, much work is yet to be done. Several issues which emerged during the development process warrant further investigation: (1) end-user difficulties with the team structure and group

process; and (2) the length of programmed instruction.

References

- [1] Alvai, M., R. Nelson, and I. Weiss (1988), "Managing End-User Computing as a Value-Added Resource." *Journal of Information Systems Management*, 5 (3), 26–35.
- [2] Argyris, C. (1982). *Reasoning, Learning and Action: Individual and Organizational*, Jossey-Bass, San Francisco.
- [3] Boem, B.WE. (1988), "A Spiral Model of Software Development and Enhancement." *Computer*, May, 61–72.
- [4] Brown, P.S. and J.D. Gould (1987), "An Experimental Study of People Creating Spreadsheets." *ACM Transactions on Office Systems*, 5 (3) 258–272.
- [5] Canter, D., J. Powell, J. Wishart and C. Roderick (1986), "User Navigation in Complex Database System " *Behavior and Information Technology*, 5 (3) 249–257.
- [6] Carlson, S.A. (1989), "Why Johnny Can't or Won't Spreadsheet." *Scandinavian Journal of Information Systems*, 1 (1), 119–144.
- [7] Carroll, J.M., Mack, R.L., Lewis, C.H., Grischlowsky, N.L. and Robertson, S.R. (1985), exploring Exploring a Word Processor." *Human-Computer Interaction*, 1, 283–307.
- [8] Carroll, J.M., Smith-Kerker, P.L., Ford, J.R., and Mazur, S.A. (1986), *The Minimal Manual*, IBM Research Report RC 11637.
- [9] Chaiken, S. and Stangor, L. (1987), "Attitudes and Attitude Change." *Annual Review of Psychology*, 38, 575–630.
- [10] Checkland, P. (1981). *Systems Thinking, Systems Practice*, Wiley.
- [11] Cheney, P.H., Mann, R.I. and D.L., Amoxoso, D.L. (1986), "Organizational Factors Affecting the Success of End-User Computing." *Journal of Management Information Systems*, Vol. 3, 65–80.
- [12] Chrisman, C. and B. Beccue, (1990), "Training for Users as a Management Issue." *Journal of Information Systems Management*, Vol. 7, No. 3, pp. 56–62.
- [13] Christensen, E. (1989), "The Impact of IS Design on Knowledge in Work." In H.K. Klein and K. Kumar (eds.) *Systems Development for Human Progress*, North Holland.
- [14] Ciborra, C.U. and G.F. Lanzara (1989), "Change and Formative Contexts in Information Systems Development." In H.K. Klein and K. Kumar (eds.), *Systems Development for Human Progress*, North Holland.
- [15] Cotterman, W.W. and K. Kumar (1989), "The User Cube: A Taxonomy of End Users." *Communications of ACM*, 32 (11), 1313–1329.
- [16] Erikson, I.V., R. Hellman, and M.I. Nurminen (1989), "A Method for Supporting Users" *Comprehensive Learning.* *Education and Computing*, 4.
- [17] Floyd, C. (1984), "A Systematic Look at Prototyping." In R. Budde, et al., (eds.), *Approaches to Prototyping*, Springer-Verlag, Heidelberg.
- [18] Galletta, D.F. and E.M. Hufnagel (1992), "A Model of End-User Computing Policy." *Information and Management*, 22, 1–18, January.
- [19] Green, J. (1984), "Productivity in the Fourth Generation." *Journal of Management Information Systems*, 1 (3), 48–63.
- [20] Habermas, J. (1984). *The Theory of Communicative Action: Reason and the Rationalization of Society*, Beacon Press.
- [21] Hayen, R., Cook, W., and G. Jecker, (1990), "End User Training in Office Automation: Matching Expectations." *Journal of Systems Management*, Vol. 41, No. 3 pp. 7–12.
- [22] Hellman, R. (1989). *Approaches to User-Centered Information Systems*. PhD dissertation University of Turkey.
- [23] Hellman, R. (1992), "Comprehension User Education for Successful End-user Computing," *Information Technology & People*, Vol. 6, No. 1 pp. 65–84.
- [24] Huff, S.L., M.C. Munro and B.H. Martin (1982), "Growth Stages of End-user Computing." *Communications of the ACM*, 31 (5), 543–550.
- [25] Karten, N. (1986), "End-user Demand Requires a New Approach to Training," *Data Management*, Vol. 5, pp. 11–19.
- [26] Karten, N. (1991), "Standards for User-Driver Applications Development." *Journal of Information Systems Management*, 8, 60–62.
- [27] Karten, N. (1991), "Integrating IS Disciplines into End User Training." *Journal of Information Systems Management*, Vol. 8, No. 1, pp. 75–78.
- [28] Keller, K.M. (1983), "Motivational Design of Instructions." in Reigelutts, (eds.) *Instructional Design Theories and Models*, LEA Publishers, N.J.
- [29] Knolmayer, G. and G. Disterer (1988), "User Developed Applications: State of Art in German Companies." *Proceedings of the First European Conference on Information Technology For Organizational Systems*, Athens.
- [30] Konsynski, B.R. (1984), "Advances in Information System Design." *Journal of Management Information Systems*, 1 (3), 4–31.
- [31] Kraus, W.A. (1984). *Collaboration in Organizations: Alternatives to Hierarchy*, Human Sciences Press.
- [32] Land, F. (1983), "Formal Methods in an Informal World." in *Proceedings of IFIP Conference*, Geneva, Switzerland.
- [33] Lehman, J. and J. Wetherbe (1989), "A Survey of 4GL Users and Applications." *Journal of Information Systems Management*, 6, 44–52, Summer.
- [34] Lin, C-Y, and C-H, Chung, (1991), "End User Computing in a CASE Environment." *Journal of Information Systems Management*, Vol. 8, No. 2, pp. 17–21.
- [35] Lin, C-Y (1990), "Systems Development with Application Questions: An End User Perspective." *Journal of Systems Management*, 41, 32–36, April.
- [36] Madsen, K.H. (1989) "Breakthrough by Breakdown: Metaptions and Structured DOMAINS," in H.K. Klein and K. Kumar (eds.) *Systems Development for Human Progress*.
- [37] Martin, J. (1989–1990), "RAD: Rapid Application Development." *PC Week*.
- [38] Matos, V.M., Misra, S.K., and P.J. Jalties (1989), "End-user Criteria for Selecting Fourth Generation Report Generator." *Journal of Systems Management*, 40, 28–32.
- [39] McGann, J.A. (1990), "Meeting the challenge of End User Computing." *Journal Of Systems Management*, Vol. 41, No. 4, pp. 13–16.

- [40] McWhinney, W. (1980). *The Resolution of Complex Issues*. Working Paper Series, Action Learning Group, York University, Toronto, Canada.
- [41] Moran, T.P. (1981), "An Applied Psychology of the User." *ACM Computing Surveys*, Vol. 13, pp. 1-11.
- [42] Morgan, G. (1986). *Images of Organization*. Sage.
- [43] Morgan, G. and R. Ramirez (1985). *Action Learning: A Holographic Metaphor for Guiding Social Change*, Working Papers Series, Action Learning Group, York University, Toronto, Canada.
- [44] Nelson, R.R., and Cheney, P. (1986), "Training End Users: An Exploratory Study." University of Georgia College of Business Administration, Department of Management, Working Paper #11.
- [45] Ngwenyama, O. (1987), Fundamental Issues of Knowledge Acquisition: Toward a Human Action Perspective on Knowledge Systems. PhD Thesis SUNY-Binghamton.
- [46] Olfman, L. and R.P. Bostrom (1988), "Recommendations for Designing End-User Software Training." *Office Knowledge Engineering*, 2, 26-39.
- [47] Olfman, L. and R.P. Bostrom (1988), "The Influence of Training on The Use of End-user Software." *SIGOIS Bulletin*, 9 (3), 110-117.
- [48] Olfman, L. and R.P. Bostrom (1991), "End-User Software Training: An Experimental Comparison of Methods to Enhance Motivation," *Journal of Information Systems*, Vol. 1, pp. 249-266.
- [49] Palvia, P. (1991), "On End-User Computing Productivity." *Information and Management*, 21, 217-224, November.
- [50] Polanyi, M. (1966), *The Tacit Dimension*, Doubleday Books.
- [51] Ramirez, R. (1983), "Action Learning: A Strategic Approach for Organizations Facing Turbulent Conditions." *Human Relations*, 36 (8) 725-742.
- [52] Raven, R. (1986), "Action Learning in a Developing Country." *Management Decision*, 24 (6), 3-7.
- [53] Reisman, S. (1992), "Power to the People: Endusers as Developers." *IEEE Software* 9: 112-118, March.
- [54] Rice, R.E. and E.M. Rogers (1980) "Reinvention in the Innovation Process." *Knowledge*, 1, (4), 488-514.
- [55] Rochester, J.B., (1989) "Applications for the 1990's." *IS Analyzer*, 27, (9), 1-12.
- [56] Rockart, J.F., and L.S. Flannery (1983), "The Management of End-user Computing." *Communications of The ACM*, 26 (10) 776-784.
- [57] Schon, D., (1983), *The Reflective Practitioner*, Basic Books, New York.
- [58] Smith, P. (1988), "Second Thoughts on Action Learning." *Journal European Industrial Training*, 12 (6), 28-32.
- [59] Suchman, L.A. and E. Wynn, (1984) "Procedures and problems in the office." *Office Technology And People*, Vol. 2, No. 2, pp. 133-154.
- [60] Suchman, L.A. (1987). *Plans and Situated Actions*. Cambridge University Press.
- [61] Suttan, D. (1989), "Further Thoughts on Action Learning." *Journal European Industrial Training*, 13 (3), 32-35.
- [62] Wlodkowski, R. (1982), *Motivation: What Research Says to the Teacher*, NEA Publications.
- [63] Wood-Harper, A.T. and D. Flynn (1983), "Action Learning for Teaching Information Systems Analysis and Design." *The Computer Journal*, Vol. 26, No. 1.
- [64] Wood-Harper, A.T. (1985), "Research Methods in Information Systems: Using Action Research." *Proceedings of the Sixth ICIS*, Indianapolis, Indiana.
- [65] Vicker, G. (1967). *Value Systems and Social Processes*. Basic Books.