IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4105

User and Developer–User Systems Development Using a Spreadsheet Program

Anders Avdic
Department of Informatics, Örebro University, SE-701 82, Örebro, Sweden,
Tel: +46 19 30 31 20, Fax: +46 19 33 25 46, anders.avdic@esi.oru.se

INTRODUCTION

In the early days of computers, expertise was needed in order to use computers. As IT tools have become more powerful and user friendly, more and more people have been able to use computers and programs as tools when carrying out working tasks. Nowadays it is even possible for people without special IT training to construct Information Systems (IS) that only IT specialists could have done some years ago.

In this paper the conditions and effects of User Systems Development (USD) using a Spreadsheet Program (SP) are discussed. USD is performed by a User-Developer (UD), a person who acts both as a user and a systems developer. A typical feature of a UD is that he has a good knowledge of the business and the work related to the Information System (IS) in question, which is called the User Developed Application (UDA).

In Figure 1 the difference between Traditional Systems Development (TSD) (1) and USD (2) is outlined in order to demonstrate the nature of USD in contrast to TSD since TSD is familiar to the IS community. To the IT-specialist, knowledge about IS development tools (e.g. methods, program languages) (1a) is in primary focus when developing TISs (1c). This is the core of his professional knowledge. Knowledge about business (1b) is of course essential but not primary. To the UD knowledge about business (2a) is in primary focus and knowledge about IS development tools (2b) is just a mean to accomplish business-oriented tasks, eventually by developing UDAs (2c). The IT-specialist has access to knowledge about IS development tools that is hard to access for non-professionals. Some business knowledge is hard to access to the IT-specialist, since this knowledge is not in the professional knowledge domain of the IT-specialist. The UD on the other hand is the expert on business knowledge. His professionalism depends on his knowledge about business. No one can replace him in this matter. In order to perform USD the UD needs some knowledge about IS development tools. It is not possible though to have access to as much knowledge about IS development tools as the IT-specialist has.

To both the IT-specialist and the UD both kinds of knowledge are to some degree necessary. In order to make an information system, the most important kind of knowledge is in general knowledge about business, since the information system is about the business. The thick arrow in Figure 1 demonstrates this circumstance.

In order to develop information systems, knowledge about business has to be transferred from business specialists to IT-specialists. This transfer is problematic since people have different frames of references. (Yourdon 1989, Alter 1996) The entire intention of the sender can therefore not be transferred to the IT-specialist. The IT-specialist can on the other hand not fulfill the requirements since he cannot completely understand the business specialist. Complex systems development tasks still have to be performed through TSD, but as more powerful systems development tools are at hand, the possibilities to perform USD are enhanced from year to year. Spreadsheet programs have properties that give the UD access to IS development features without being an IT-specialist. Other ways to overcome this gap is to perform systems development with a participative approach e.g. RAD. (Tudhope et al 2001). The systems discussed in this paper are often small and local and thereby often not suitable for traditional systems development projects.

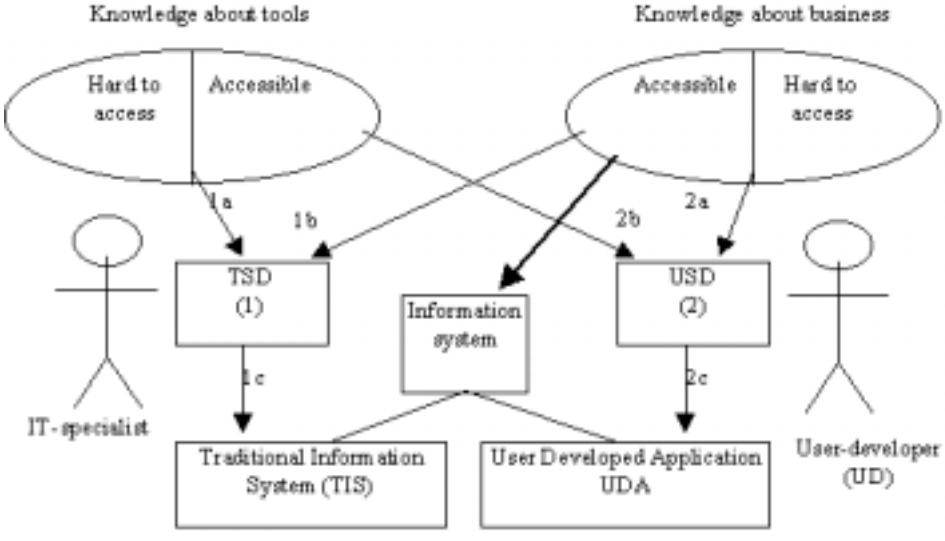
With the discussion above in mind the basic research questions in the paper are the following:

- What new possibilities can computer users develop in order to perform tasks, when they can develop ISs without help from IT specialists?

The question can be subdivided into the following questions:

- How can UDs take not-easily-formulated knowledge into consideration when performing USD?
- What kind of IS development tool knowledge does a UD need in order to perform tasks?
- What other conditions affect the UD:s possibilities to perform USD?
- Which effects are the results of USD?

Figure 1: The relation between knowledge and development



METHODS

The studies described in the paper have a strategy that is qualitative, hermeneutic, and Grounded Theory (GT) influenced.

The *qualitative* aspect is manifested through the aim of understanding and interpreting the reality of the UDs. (Patton 1987) A typical feature of qualitative methods is triangulation. (Yin 1994) The methods used are in depth interviews, studies of SP-UDAs, participant observation and theoretical studies. The *hermeneutic* approach is manifested through the explicit use of the researchers pre-knowledge of USD and of the respondents. (Helenius 1990) Interviews and contacts

have been frequent. The researcher's knowledge of USD has resulted in long discussions of different ways to solve specific problems. This has resulted in an atmosphere where the researcher has gained access to the respondents' situation as UD in a way that would have been impossible if e.g. survey studies had been performed. This change of perspective is also typical of hermeneutic studies. In the empirical studies the focus has alternated between product and process (SP-UDA and USD). Another change of focus also closely related to USD is the change of focus between the actual work and UDA-development. While the UD is focused on the tasks at hand, the observer has been focused on the developmental aspect. The research methods can be labeled as *Grounded Theory (GT)* influenced since the empirical research has proceeded the theoretical studies. (Patton 1987) Another GT aspect is the aim of an unbiased data collection situation. The choice of both research questions and respondents has been a result of a specific intention but the data collection has been performed with an awareness of the importance of initially not knowing what really happens when UDs perform USD. The data analysis has therefore been performed according to GT methods (data collection, open coding and selective coding). The coding activities have aimed at finding a key variable. The variable found is "integration". The importance of this is discussed below.

Three empirical studies have been conducted between 1998 and 2000. Two studies have focused on UDs. Interviews have been conducted with three people in an industry and three people in a public authority. In the third study three IT specialists are interviewed about problems related to UDA. The UD-interviews have been open and focused on the systems made by the UDs. Questions have been asked about why the systems were developed, which alternatives and problems there were, how the systems had been developed, how they were used in the UDs work and how the UDs perceived the effects of USD. In depth interviews with each UD have been carried out between three and ten times depending on how many UDAs the UD had developed. The interviews have been taped and coded. The analysis has been performed according to GT (see above). The studies are shortly described below.

The industry is a board mill with 750 employees. It can be characterized as a multi-goal, dynamic business acting in a keen competitive market. The area of board production is technically and chemically advanced and the board machines are very complicated. The customer's quality demands are increasingly detailed and not easily achieved. The three UDs work as controller, production planner and production division manager. The controller is an experienced SP user and the other two are somewhat less experienced.

The 50 employees in the public authority work in building, traffic, environment, and maps construction units. The persons interviewed were clerks in different departments. Activities in the authority are characterized by their public nature. This means that business should neither be profitable nor involve a loss. Other important goals are that the municipality inhabitants´ best interests should always be taken into consideration and that activities should be carried out with openness. This demand for openness means that the grounds on which decisions are taken should be both available and comprehensible. One of the clerks is a more experienced SP user than the other two.

The three IT-specialists were an IT-manager, a systems developer and a consultant systems developer. Findings show that the IT-specialists perceived problems related to the USD like lack of documentation, unstructured applications and limited data processing capabilities in SP. The solutions proposed to these problems were SP training, SP-version upgrading and more structured UDA:s.

FRAMEWORK AND RESULT

Since the paper claims to be GT influenced, studies of related theory have been greatly influenced by the empirical studies. As a framework model, a modified version of the model of generic practice (the ToP model) (Goldkuhl & Röstlinger 1999) is used in order to systemize empirical findings and related theory. The model can be used to specify the conditions and result of a specific practice, e.g. a controller practice or an IT specialist practice. The modified model consists of a set of conditional categories, *knowledge*, *norms*, *and tools*. The categories that express the specific practice are named *producers* and their *actions*. The last category is the *result* of the practice. When a UD develops UDAs he acts in at least two types of practices, the primary (e.g. controller) practice and the secondary (developer's) practice. Each practice is related to a profession, e.g. a controller and an IT specialist profession. The model makes it possible to separate the conditions of the different practices. It also makes it possible to discuss which parts of the developers practice that can improve the main practice without consulting an IT specialist. The nature of the categories are described below together with presentation of findings from the studies.

Information Systems (Result)

A UDA is an IS and an IS is a result of systems development. The difference between a traditional information system (TIS) and a UDA is mainly a question of how it is built. UDAs are built by UDs with a good knowledge of the business, while TIS:s are built by IT specialists. (Avdic 1999)

SP-UDAs can be divided into four categories according to the how long learning time the UD need in order to develop the SP-UDA. The first category is called "Simple SP-system". This rather common UDA consists more or less of structured text. It could be defined as a "pre-stage" to more complex UDAs. The next category is "Small SP-system". Typically it has simple formulas and eventually SUM-functions and simple diagrams. "Large SP-system" has more complex formulas and functions and can be distributed on several spreadsheets. The most complex UDA-system is "Application" It can be very complex and it can include programming code. (Avdic 1999) UDAs in the studies were of Small and Large types.

User Systems Development (Actions)

Traditional Systems Development (TSD) can be characterized by the notion of the "Life Cycle," where tasks are specialized and activities are separated and systemized. User Systems Development (USD) and TSD are profoundly different in many ways. USD actions in the studies were e.g. neither organized nor planned. Specific work related tasks or problems made the UD aware of some information need. USD was looked upon as work rather than systems development by the UD. Compared to TSD, USD is characterized by integration rather than specialization. Still it is systems development.

Success factors of USD have been discussed in the scientific community for more than a decade. The reasons why USD is successfully adapted in an organization have been claimed to depend on the presence of informal channels of communication and how common training on USD tools is. (Brancheau & Brown, 1993) Basic conditions (suitable tasks, equipment, knowledge, and certain independence) must be fulfilled to make USD *possible*. (Carlsson, 1993) If business and information needs are dynamic, USD can be *justified*. USD is *appropriate* when UDs also have access to well-organized data and get support from management and the IT-department. (Auer, 1998) Perceived importance is also claimed to be vital. (Blili et al 1998).

When discussing of how to manage and control USD, advocates of *high control* recommend (strict) organization of USD activities. (e.g. Andersen 1994) Advocates of *low control* consider USD as time saving and appropriate because of the lack of detailed monitoring. (e.g. Speier & Brown, 1997)

The discussion of what factors are determining successful USD is implicitly aiming at organizing USD activities with a certain degree of control. In our study this discussion is not really relevant since the UDs are professional in their respective profession. They have used IT tools if they have found it relevant in relation to their work tasks. Since they did not separate USD from running work, they had the same quality demands on the USD result as on the rest of their

work. In opposite to some research (e.g. Teo & Tan 1999) our study shows that the risk of poor quality in UDA information output should be related to the UDs professionalism rather than to design methods or tool properties.

User Developers (Producers)

A UD is a person with a good knowledge of the business who develops UDAs that supports the UD in his work. The UD is primarily a professional (e.g. a controller) who integrates to some extent the role of one or more IT specialists, when performing USD. The UD could have good knowledge about IS development tools. This does not disqualify him as a UD, it rather makes him even more efficient.

Knowledge

When performing USD, knowledge is divided between the UD and the tool (SP). Certain kinds of (not too complex) knowledge are formalized into the SP and can be used in the SP-UDA. Other kinds can be formalized by the UD into the SP-UDA. Some kinds of knowledge (e.g. of critical evaluation of the relevance of formulas) cannot be formalized at all. Still this kind of not-easily-formalized (sometimes tacit) knowledge can be taken into consideration when using the UDA, since the UD (with business knowledge) is the user of the system. The findings also show that goals, not easily formalized, can be taken into consideration when performing USD.

Knowledge about tools can be used to deepen knowledge about business. UDs in the studies could make tacit knowledge explicit when developing USD, which in turn made it possible for others to evaluate and criticize the UDA and its output. The UDs were very conscious about that an ongoing change in the companys/authoritys environment made it important to develop not yet known knowledge about conditions and circumstances of their work. Our findings show that one important aim of the UD is to articulate knowledge about business and that UDA is one important mean to do this.

Norms

Norms and knowledge are closely related and sometimes hard to keep apart. One set of norms that are central in the paper is professional ethics. Professional ethics are crucial to the UD since the professionals' activities are monitored not by procedures but by professional and business ethics. Professional ethics as well as professional tacit knowledge (se above) cannot easily be transferred to IT-specialists in systems development projects. Therefore when USD is performed by UDs professional ethics and tacit knowledge can be taken into consideration in a way not possible in TSD. Findings of the industry study also show that investigations made by the UD when performing USD can change organizational norms. Ongoing questioning of business using UDAs can implicitly or explicitly challenge existing models as well as their norms. In the study the methods of measuring production was questioned which in turn resulted in changes in existing models and calculations. In the authority study changes in the political situation resulted in e.g. demands of new models to assess the value of real estate. This does not mean that revolutionary effects take place every time a UDA is developed.

Tools

USD tools are closely related to norms and knowledge, since norms and knowledge are implemented in tools. The main tool when performing SP-USD is of course the SP. The SP integrates functions for input, output, storage, processing, and presentation. This integration results in interactive development and use. The open nature of the SP can cause different kinds of errors. (E.g. Panko & Sprague 1998) Knowledge of business, tools, and design can prevent some of these errors. Another circumstance that makes SP suitable for UDA is the fact that they are very common. In Sweden almost all employees can have access to a SP.

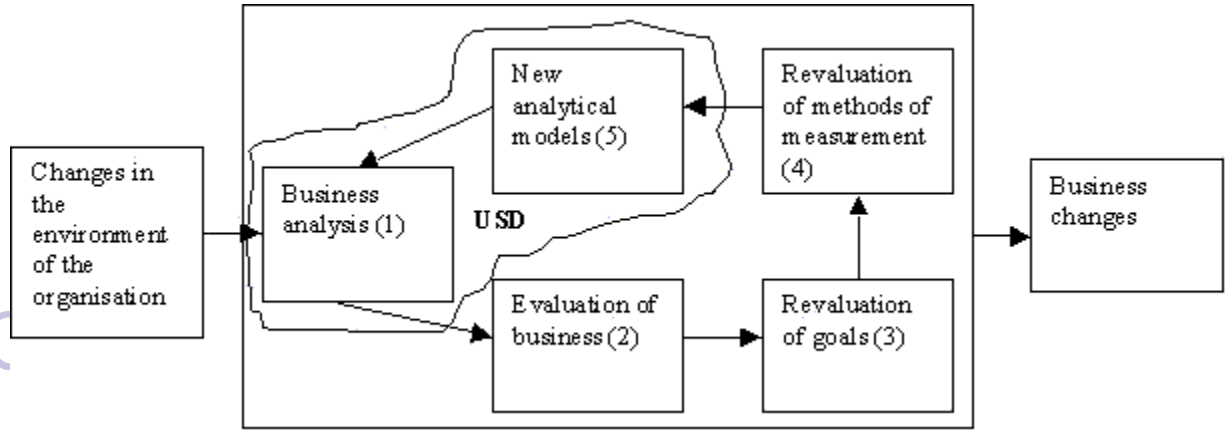
Because of the integrated nature of USD, learning, using and systems development take place at the same time. Learning applies to both the business and the tool. One conclusion of this is that training in the use of a tool can improve the quality of USD, which in turn can improve business. One way for the management to support USD is to initiate and encourage UD-tailored training in the use of tools.

DISCUSSION AND CONCLUSION

SP-USD is characterized by *integration, interactivity and capacity of questioning*. The notion of *integration* can be looked upon in several dimensions: 1) aspects of ISs (integration of collecting, storing, processing, and distribution of information), 2) roles (integration of developer, user and manager roles), 3) roles of actors in systems development (integration of analyst, programmer, database and designer roles), and 4) integration of processing functions of the IS. The integrated nature of USD results in *interactivity*. Interactivity means that the UD can change quickly between developing and using the SP-UDA. During the USD-process the UD knowledge of the business and USD increases. This is actually the goal of the UD. Since the UD knowledge of the business increases when performing USD, the UD can analyze and also *question* aspects of business (e.g. production measuring methods). The questioning aspect makes it possible to improve business.

SP-USD's can be used as a mean of controlling continuous changes in the environment of the organization by changing business with the help of USD. A business analysis (1) can result in a revaluation of the business (2), which can result in a revaluation of its goals (3) (and norms), which can result in a revaluation of methods of measurement (4), which can result in new analytical models (5) (UDA), which can lead to a new business analysis (1) and so on. USD is discussed as one way to meet change as a permanent business condition, which differs from traditional methods for systems development.

Figure 2: Continuous change and user systems development



This way of revaluating organizational goals can be related to double-loop learning as it is presented by Argyris & Schön (1996). This includes not only changes in behavior or strategies. It means that norms of the organization can be changed. The ongoing questioning of business practice that is performed through USD can imply this form of norm changing.

REFERENCES

Alter S, (1996) *Information Systems - A Management Perspective*, Benjamin/Cummings, Menlo Park, CA.

Andersen E S, (1994) *Systemutveckling –principer, metoder och tekniker*, Studentlitteratur, Lund. [In Swedish]

Argyris C & D A Schön, (1996) *Organizational Learning II - Theory, Method, and Practice*, Addison Wesley, Reading, Mass.

Auer T, (1998) "Factors Affecting End-User Computing: Skills", *TUCS Technical Report No 159, Department of Computer Science, University of Turku.*.

Avdic A (1999) *Användare och utvecklare – om anveckling med kalkylprogram*, Dissertation, Linköping university. [In Swedish]

Blili S, Raymond L & S Rivard, (1998) "Impact of task uncertainty, end-user involvment, and competence on the success of end-user computing", *Information & Management*, Vol 33, 137-153.

Brancheau J C & C V Brown, (1993) "The Management of End User Computing: Status and Directions", *ACM Computing Surveys*, Vol 25, No 4, Dec 1993, 437-482.

Carlsson S A, (1993) *A Longitudinal Study of User Developed Decision Support Systems*, Dissertation, Department of Informatics, Lunds University.

Goldkuhl G & A Röstlinger, (1999) *Expanding the scope – From language action to generic practice*, CMTO Research Papers No. 1999:06, Linköping University.

Helenius R, (1990) *Förstå och bättre veta*, Carlssons, Stockholm. [In Swedish]

Panko R R & Sprague jr R H, (1998) "Hitting the wall: errors in developing and code inspecting a 'simple' spreadsheet model", Decision Support Systems, Vol 22, 337-353.

Patton M Q, (1987) *How to Use Qualitative Methods in Evaluation*, Sage, Beverly Hills, CA.

Speier C & C Brown, (1997) "Differences in end-user computing support and control across user departments", *Information & Management*, Vol 32, 85-99.

Teo T S H & Tan M, (1999) "Spreadsheet development and 'what-if' analysis: quantitative versus qualitative errors", Accounting, Management and Information Technologies, Vol 9, 141-160.

Tudhope D, Beynon-Davies P, Mackay H & R Slack, (2001) "Time and representational devices in Rapid Application Development", *Interacting with Computers*, Vol 13, 447-466.

Yin R K, (1994) *Case Study Research – Design and Methods*, Sage, Beverly Hills CA.

Yourdon E, (1989) *Modern Structured Analysis*, Prentice-Hall, Englewood Cliffs, NJ.