

User Spreadsheet Systems Development

Anders Avdic

Örebro University, Sweden

INTRODUCTION

In the early days of computers, expertise was needed not only to develop systems, but also to use them. 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 develop information systems that only IT specialists could have done some years ago.

BACKGROUND

In this article the conditions and effects of user systems development (USD) using a spreadsheet program (SP) are discussed. USD is characterized as a sub-area of end-user computing. USD is performed by a user-developer, a person who acts both as a user and a systems developer. A typical feature of a user-developer 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).

To a large extent USD is a question about learning. User-developed applications are often developed in order to learn and understand. 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 the user-developer's professional knowledge. Knowledge about business (1b) is of course essential, but not primary. To the user-developer knowledge about business (2a) is in primary focus and knowledge about IS development tools (2b) is just a means 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 for the IT specialist, since this knowledge is not in the professional knowledge domain of the IT specialist. The user-developer 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 user-developer 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.

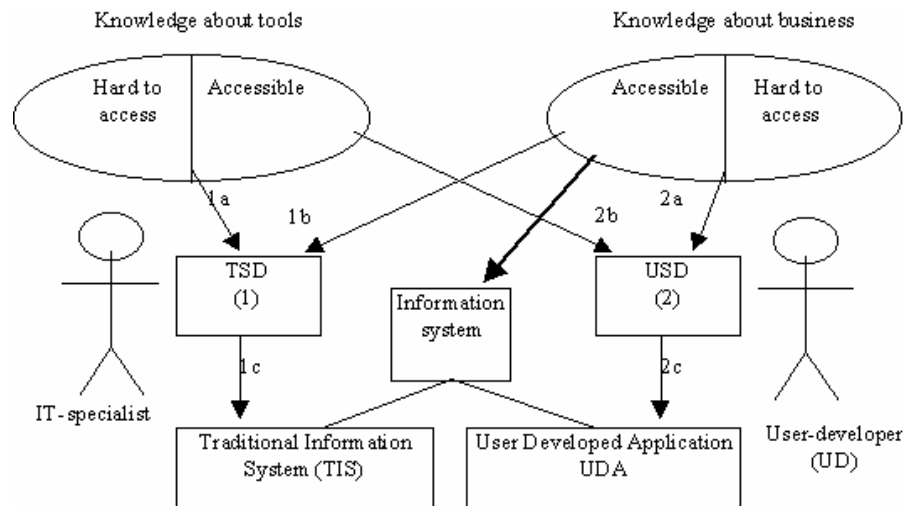
To both the IT specialist and the user-developer, both kinds of knowledge are to some degree necessary. In order to make an information system, the most important kind of knowledge is in general knowledge about business, since the information system is about the business. The thick arrow in Figure 1 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 whole intention of the sender can therefore not be transferred to the IT specialist. The IT specialist cannot, on the other hand, 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 user-developer access to IS development features without being an IT specialist. Of course there are other ways to overcome this gap, for example, by performing systems development with a participative approach like RAD (Tudhope, Beynon-Davies, Mackay & Slack, 2001). The systems discussed in this article are often small and local, and thereby often are not suitable for traditional systems development projects.

CONDITIONS AND EFFECTS OF USER SYSTEMS DEVELOPMENT

As a framework model, a modified version of the model of generic practice (the ToP model) (Goldkuhl & Röstlinger, 1999) is used to systemize empirical findings and related theory. The model can be used to specify the conditions and result of a specific practice, such as 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* (the user-developer) and their *actions* (user systems development). The last cat-

Figure 1. Relation between knowledge and development (Avdic, 1999)



egory is the *result* of the practice (the application). When a user-developer 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, such as 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 developer's practice can improve the main practice without consulting an IT specialist. The use of the model makes it possible to show how different practices exchange conditions and effects. The result of user developer practice might, for example, be a condition of the controller practice. The model is described in Figure 2.

The ToP model is slightly related to the Work Systems model (Alter, 2002), in that the model focuses on practice without specific references to IT artifacts. The ToP model emphasizes knowledge aspects more explicitly, which makes it especially suitable to analyze the practice of user systems development. The nature of ToP model categories is described below.

Information Systems (Result)

A UDA is an information system, and an information system 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 user-developers with a good knowledge of the business, while TISs are built by IT specialists.

User Systems Development (Actions)

Traditional systems development can be characterized by the notion of the 'Life Cycle', where tasks are specialized and activities are separated and systemized. USD and TSD are profoundly different in many ways. USD actions are seldom organized nor planned (Avdic, 1999). Specific work-related tasks or problems make the user-developer aware of some information need. USD is looked upon as work rather than systems development by the user-developer. From the user-developer's point of view, any tool is useful that might help him solve work-related problems. Compared to TSD, USD is characterized by integration rather than specialization. Where TSD professionals get specialized in programming, analysis, or database design, the user-developer integrates skills and performs the entire life cycle by himself.

Success factors of USD have been discussed in the scientific community for more than two decades. 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 (Kruck, Maher & Barkhi, 2003; 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 user-developers 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, Raymond & Rivard, 1998).

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/user-spreadsheet-systems-development/14727

Related Content

Selecting Indispensable Edge Patterns With Adaptive Sampling and Double Local Analysis for Data Description

Huina Liand Yuan Ping (2024). *Journal of Cases on Information Technology* (pp. 1-26).

www.irma-international.org/article/selecting-indispensable-edge-patterns-with-adaptive-sampling-and-double-local-analysis-for-data-description/335945

The Influence of National and Organisational Culture on Knowledge Sharing in Distributed Teams

Kerstin Siakas, Elli Georgiadouand Dimitrios Siakas (2020). *Information Diffusion Management and Knowledge Sharing: Breakthroughs in Research and Practice* (pp. 533-555).

www.irma-international.org/chapter/the-influence-of-national-and-organisational-culture-on-knowledge-sharing-in-distributed-teams/242148

Inter-Sector Practices Reform for e-Government Integration Efficacy

Teta Stamatiand Athanasios Karantjias (2011). *Journal of Cases on Information Technology* (pp. 62-83).

www.irma-international.org/article/inter-sector-practices-reform-government/56309

B

(2007). *Dictionary of Information Science and Technology* (pp. 46-68).

www.irma-international.org/chapter//119563

Organizational Culture and Employees' Computer Self-Efficacy: An Empirical Study

YiHua P. Sheng, Michael Pearsonand Leon Crosby (2003). *Information Resources Management Journal* (pp. 42-58).

www.irma-international.org/article/organizational-culture-employees-computer-self/1247