The Emergence of a Software Support System

Making Sense of an ‘Ad Hoc’ Approach to IT-adaptation

Ola Henfridsson[a] & Greger Lundström[b]
ola.henfridsson@informatik.umu.se[a] and
Greger.X.Lundstrom@telia.se[b]
Department of Informatics, Umeå University, Sweden[a]
Telia ProSoft, Sweden[b]

Abstract

This paper examines the processes by which a software support system was adapted and re-adapted in a fast-growing software company. Instead of introducing the system according a master plan, the company adopted a more relaxed approach where problems were handled ‘ad hoc’. Up to the time of this study, two new versions of the system had been put into use and, furthermore, several changes in the organizational use of the system had been observed as a way to master and exploit up-coming contingencies in daily activities.

Keywords: sensemaking, IT-adaptation, action, software support system.

BRT Keywords: AE, DD, GA

1 Introduction

In a global economy characterized by on-going development of new routines and transactions, the processes by which information technology gets adapted and re-adapted in use seem to be increasingly important. Even though this holds primarily for firms working in businesses sensitive to technological development, one might say that the increased focus on how to make the most out of information technologies already in use is a quite general phenomenon. Prior research on adaptation of groupware technology (Ciborra, 1996a; Ciborra and Hanseth, 1998a) indicates, for instance, that organizations such as LaRoche and Unilever have integrated the new technology into their work practices using what might be called an ‘ad hoc’ approach to IT-adaptation. In doing that, these organizations have mastered and exploited up-coming contingencies as they occur in daily IT-use.

As a central issue of making the most out of IT-adaptation, this paper concerns the actual creation of technological conceptions. Indeed, we intend this paper as an exploration of how people develop their understanding of IT in use to fit their current projects and interests. This intention implies an interpretive research philosophy (see e.g., Orlikowski and Baroudi, 1991; Walsham, 1993) where we decided to use the interpretive case study as our methodology (Klein and Myers, 1999; Walsham, 1995a). In order to facilitate the understanding of this topic, the paper introduces yet another theoretical perspective to IT-adaptation – a sensemaking perspective (Weick, 1979, 1995). This is not to say that this perspective outperforms alternative perspectives such as actor-network theory (H or structuration theory in terms of understanding IT-adaptation in general, but rather that the sensemaking perspective is suitable for understanding the *creation* of IT-conceptions particularly.

The specific case explored concerns how a particular software support system developed by Pure Atria – DDTS – was adapted in a fast-growing software company – Finance Software Group (FSG, a pseudonym). The software support system was introduced as a way to support and facilitate customer relations. Instead of introducing DDTS according to some sort of a master plan, FSG adopted a more relaxed approach where up-coming contingencies were handled 'ad hoc'. Considering the knowledge-intensive business in which FSG operates, this was a natural choice to cope with the sometimes over-whelming changes that turn old recipes upside down. Since the introduction in early 1994, two new versions of the system have been put into use and, in addition, several changes in the organizational use of the system have been observed. The paper intends to make sense of the particular changes resulting from different key groups' efforts to make sense of DDTS in their daily work

A better understanding of IT-adaptation can be useful for leveraging IT in conditions of change and business transformation. In particular, this understanding can be a means for occasioning and managing periods of radical learning. Indeed, most process methodologies successfully support the identification of new business processes. They do also, however, carry with them certain implicit assumptions that largely ignore the opportunities residing in the small events in every-day organizational activity. Hopefully, this paper can contribute to a complementary understanding of business transformation and IT.

The remainder of the paper is organized as follows. Section two provides a review of what related literature can offer for understanding the emerging process of IT-adaptation, while section three outlines a sensemaking perspective for understanding this. Section four describes the investigated case and the research methods used for investigating it. Section five offers an interpretation and analysis of the investigated case, while section six concludes the paper.

2 Understanding IT-adaptation: Related Literature

There exist a variety of different theoretical frameworks with which to understand IT-adaptation. All these frameworks have their strengths and weaknesses depending on the particular context they are applied to. In what follows, we interpret what these frameworks have to offer for understanding our particular problem, namely, sensemaking in IT-adaptation. Before reviewing these frameworks, however, it is worth mentioning that one important difference between the classical literature on adaptation of production technology and the emerging literature on IT-adaptation is the level of analysis. Consider, for instance, Rogers' (1995) seminal work on diffusion of innovations that views such diffusion as a result of existing communication channels, time, and degree of shared interests between individuals. Others have been able to reveal how experience causes productivity improvements over time (see e.g., Alchian, 1963; Conway and Schultz, 1959). In recent information systems research, however, more emphasis has been put on behavioral aspects of technological adaptation, or, using the words of Richard Cyert and James March, on ‘the internal operation of the firm’ (Cyert and March, 1992). This emphasis has resulted in research on, for instance, the timing of technological adaptations

(Tyre and Orlikowski, 1994), improvisation in technological adaptation (Orlikowski, 1996; Orlikowski and Hofman, 1997) and appropriation of groupware technology (Ciborra, 1996b). These efforts assert that the core of IT-adaptation resides in the internal procedures, assumptions, routines, relationships and knowledge with which information technology is appropriated in the ordinary flow of organizational day-to-day activity. Having this said, the following paragraphs explore a handful of different theoretical frameworks for understanding the micro-level processes of IT-adaptation (See Table 1).

2.1 Theoretical Perspectives for Understanding IT-adaptation

Orlikowski (1996a) and Orlikowski and Hofman (1997) outline two action-oriented models – 'a situated change perspective' and 'an improvisational model for change management' – sharing basically the same situational view on IT-adaptation. Orlikowski and Hofman (1997), for instance, make an interesting distinction between what they call 'anticipatory', 'emergent' and 'opportunity-based' changes. This distinction is indeed suitable for seeing how IT evolves as a result of social actors possessing anticipations and expectations, adopting and enacting emergent patterns of use, and, not the least, exploiting up-coming opportunities. Fruitful as this classification is, however, it does not grasp the actual dynamics of these changes. The presented analogy of the jazz band works well for imaging how users' interaction with technology develops over time, but the analogy does not reveal how users act on technology and thereby contributing substantially to its constitution. Another interesting model of pragmatic action in IT-adaptation is represented by Ciborra and Lanzara's (1994) attempt to provide a new vocabulary for dealing with the ambiguous and interactive settings that systems' designers and organizational actors find themselves in. Drawing on the work of Chris Argyris and Donald Schön (see e.g., Argyris, Putnam and McLain Smith, 1985; Argyris and Schön, 1996), they outline an analysis where they argue for seeing IT-artifacts as constantly interacting with "…both the structural and institutional arrangements associated to a given division of labour and the assumptions, frames, and mental images that people hold while routinely enacting and practicing that specific division of labour" (Ciborra and Lanzara, 1994, p.63). Using the concept 'formative context' for referring to the institutional arrangements that form the routinely behavior that organizational actors enact, Ciborra and Lanzara argue that a major obstacle for successful IT-adaptation is limited capability to inquire into that formative context. Proposing a view of systems design as context-making, they conclude by sketching a new agenda for design where on-line practical experiments are seen as an important step of creating systems and routines for self-questioning. Realizing this, they mean, provide an organization with a capability to avoid vicious circles and overcome destructive inertia that otherwise would impede successful IT-adaptation.

Hanseth and Monteiro (1997) outlines a view on how behavior can be inscribed in technology. Their analysis, which draws on actor network theory (Akrich and Latour, 1992; Callon, 1991; Latour, 1987), concerns specifically the case of information infrastructure standards in the health care sector. They describe a "close-to-technology" view of how EDIFACT was becoming an actor in itself, containing programs of action that instructed the use and development of the investigated information infrastructure. As a result of EDIFACT, Hanseth and Monteiro suggest that the standard inscribes barriers on end-user involvement. Because of the apparent complexity, there is no way that users can play any significant role in shaping the development of these infrastructures. On a more general level, Hanseth and Monteiro introduce a number of useful concepts from actor-network theory describing the process by which IT becomes what it is (see also, Monteiro and Hanseth, 1995). Partly contrary to the other views presented in this literature review, this perspective offers a number of ways to open-up the technology itself, and thereby handle Kling's (1991) relevant criticism of a tendency in our field to make ‘convenient fiction’ by abstracting the technology to the extent that it is of no relevance when analyzing social aspects of IT-use and development.

Theoretical perspective	Authors	View on IT-adaptation
Action	Orlikowski (1996a) Orlikowski and Hofman (1997)	• IT evolves as a result of social actors' anticipations and expectations, enactment of emergent patterns of use, and exploitation of up-coming opportunities
	Ciborra and Lanzara (1994)	• IT-artifacts can be seen as formative contexts that form the routinely behavior that organizational actors enact.
Actor-Network	Hanseth and Monteiro (1997)	• New patterns of IT-use emerge as a result of a series of translations ‘inscribed’ into the technology.
Interpretive	Orlikowski and Gash (1994)	• The degree of congruency of key groups' underlying assumptions, expectations, and knowledge of technology largely determine the outcome of IT-adaptation.
Phenomenology	Ciborra (1996b, 1997) Ciborra and Hanseth (1998b)	• The way that things are perceived, or cared for, largely determines the outcome of IT-adaptation.

Table 1. Overview of existing literature.

Orlikowski and Gash (1994) provide a coherent and useful account of how 'technological frames' can be used to identify different key groups' underlying assumptions, expectations, and knowledge of technology. They suggest that in cases where there exist significantly divergent technological frames between, for instance, managers and users, there is quite a chance that conflicts and other types of difficulties arise, and distort the development, use and change of technology. Incongruency of technological frames, therefore, can be seen as an obstacle to successful IT-adaptation in organizations. While addressing that subset of cognitive structures that concerns technology in an interesting way, however, Orlikowski and Gash (1994) has little to say on how these technological frames come about. Considering our case, where emphasis is put on the actions resulting in particular technological frames, it would be very intriguing to consider the consequences of Orlikowski and Gash's thinking for the birth, development and death of certain frames.

Somewhat related to Ciborra and Lanzara's (1994) action perspective, Ciborra (1996b, 1997) and Ciborra and Hanseth (1998b) launch yet another model of IT-adaptation. This model draws to a greater extent on phenomenology (Heidegger, 1962, 1977; Schutz, 1967), pointing at the importance of certain ways to perceive things for successfully integrating IT into organizational day-to-day activity. Ciborra (1996b), for instance, outlines how certain forms of care – perception, circumspection, and understanding – describe and reveal an organization's capability to establish relevant learning processes in IT-adaptation. Circumspection is described as the form of care during which invention and discovery of opportunities occur. To successfully benefit from new IT, Ciborra suggests that the technology should not be taken as something which is fully controlled, but rather as something that inherits potentials that might be exposed as surprises and opportunities in the domain of pragmatic action. A certain attitude conveying care and hospitality must therefore be maintained to establish the relevant preparedness for up-coming 'windows of opportunity' (c.f., Tyre and Orlikowski, 1994).

2.2 Assessing the Creation of Technological Conceptions in IT-adaptation

In sum, all theoretical frameworks presented above can be said to be representatives for what Walsham (1995b) refers to as the emergence of interpretivism. Hence, they deal with the subjective and inter-subjective meanings that people create and associate with phenomena in the world. Consequently, they also study technology use and development through this meaning-creation. One important benefit of doing this is that they all deconstruct the artificial distinction between design and use, which otherwise risks to work as an obstacle to understand interaction between technology and organizations (c.f., Orlikowski, 1992; Orlikowski and Robey, 1991). There are also differences between the frameworks. The phenomenological and interpretive perspectives, for instance, are focused on people's conceptions of technology, rather than on the actual creation of these conceptions. As suggested above, this is both relevant and valuable, but, it also misses certain dimensions of the dynamics of IT-adaptation.

An important difference between 'sensemaking' and other interpretive perspectives such as hermeneutics and phenomenology is that the sensemaking perspective focuses on the actual making of frameworks with which we interpret, while interpretive perspectives, on the other hand, study how things are put into context by using already existing frameworks. Indeed, the sensemaking perspective builds on our belief that it is through people's active production and assignment of meanings to IT that this technology becomes useful in organizational contexts. Our case study of the adaptation of a software support system in FSG is therefore oriented towards how people – technicians, product developers, help-desk workers and account managers – invented new usage of information technology over time by attributing and making new sense of its integration in their daily work.

So, how can we understand the creation of technological conceptions in IT-adaptation, then? Next section explores the potential of Weick's (1979, 1995) sensemaking perspective for understanding IT-adaptation.

3 A Sensemaking Perspective on IT-adaptation

While early organizational theory treated technology as a determining force with anticipated outcomes in terms of efficiency and workflow, the merging of information and technology into what Zuboff (1988) calls 'the smart machine' created technologies that were equivoque. As Weick (1990, p.2) points out, such technologies admit "…

several possible or plausible interpretations and can therefore be esoteric, subject to misunderstandings, uncertain, complex and recondite". The questions concerning what a particular technology is and what it should be used for are to some extent in the hands of organizational actors. Weick (1979, 1995) outlines four concepts useful for understanding the creating part of IT-adaptation: enactment as bracketing, enactment as self fulfilling prophecies, identity construction and retrospection. At a general level, 'enactment' refers to the process by which social beings build their own environment through acting on that environment in compliance with their beliefs about it (Weick, 1979, pp. 147-169; see also e.g., Porac, et al, 1987). Indeed, our environment is as much a product of human action as it is an objective force that restricts such action. This resembles Giddens' (1984) 'duality of structure', a notion that unites the earlier opposition between the subjective and objective dimensions of social reality. Human actions are enabled and constrained by structures, Giddens asserts, yet these structures are the result of previous actions.

Concepts	Illustrations
Bracketing	• Contrary to initial conceptions, social workers bracketed the communications technology ‘First Class’ as a tool for getting in touch with people easily (Henfridsson, 1999).
Self-fulfilling prophecies	• User expectations building on single-user applications created the groupware Lotus Notes to work as a single-user system in a management consultancy firm (Orlikowski, 1996b).
Identity construction	• Introducing medical image technology at a community hospital influenced and was influenced by the identity-construction of radiologists and radiological technologists (Barley, 1986).
Retrospection	• People at Statoil re-considered their conception of the early use of Lotus Notes in retrospect (Monteiro and Hespø, 1999).

Table 2. Conceptual illustrations.

Firstly, enactment as *bracketing* (Weick, 1979) refers specifically to the punctuating of ongoing flow of experience. In IT-adaptation, it refers to the process by which organizational actors, consciously and unconsciously, select certain aspects of their interaction with technology as relevant. As a result of this process, their conceptions of the very same interaction are transformed. One way to understand this bracketing is to consider how social workers in a Swedish social services department bracketed the communication technology 'First Class' as a tool for getting in touch with people, rather than building their conception on more fancy functionality such as real-time discussions (Henfridsson, 1999). Initial visions about First Class' ability to enable vertical integration, active information consumers and the learning organization were bracketed into a more homogenous conception of First Class as a way to get in touch with people easily. Trough this bracketing process social workers created and defined what First Class meant to them in this particular context. At a general level, IT gets bracketed to fit better with the practical day-to-day activity, partly irrespective of whatever plans and opportunities that might be related to its initial introduction.

Secondly, enactment as *self-fulfilling prophecies* typically involves the creation and discovery of the expected (Weick, 1995, p. 35). As an illustration, consider Orlikowski's (1996b) study of how Lotus Notes was integrated into the work practice of a large services firm. Despite classroom training and self-study, the premises underlying groupware were not recognized. Instead, the experiences of single-user applications were transferred to the introduced groupware. Not surprisingly, the expectations building on single-user applications created Notes to be a single-user system.

Thirdly, IT-adaptation depends largely on how the technology used relates to the *identity construction* of its users and designers. Barley (1986) shows, for instance, how changes in the role relations between radiologists and radiological technologists were triggered by the introduction of computed tomography (CT) scanners in two community hospitals in Massachusetts. In the hospital referred to as the Suburban, introducing the CT scanner resulted in role reversals, where the traditional ground for the radiologists' identity was severely shaken. The accustomed precedence over radiological technologists in interpreting whether a scan evidenced pathology could not be sustained in situations where inexperienced radiologists were pressed to provide immediate readings. Partly a result of the new technology and partly a result of a relative unbalance in experience between the job categories in this particular hospital, the medical image technology at the Suburban hospital influenced and was influenced by the identity-construction surrounding its introduction and adaptation. Basically, identity construction can be described by saying that depending on who we are, we make sense of different things, while who we are depend much on the situation we face. As early as in 1934, Mead (1967) conceptualized this idea in his discussion of the 'I', 'Me' and 'Self', where one important distinction is that of the 'I' from the 'Me'. While the 'Me' is social (out there), the 'I' is psychic (in here). Only separable as analytical elements, they both stand in a reciprocal relationship to each other. The 'Me' is social in that "…the 'me' is the organized set of attitudes of others which one himself assumes" (Mead, 1967, p. 175) and "…it is the presence of those organized sets of attitudes that constitutes that 'me' to which he as an 'I' is responding" (Mead, 1967, p. 175). It is the conversation between the 'I' and the 'Me' that constitutes the self, the identity. So depending on the organizational setting we face, we will apprehend different sets of organized attitudes constituting the 'me' to which the 'I' is responding (Mead, 1967). This means that *what a situation means is defined by who I become while dealing with it or what and who I represent*. In this way, the co-evolution of the organizational setting I face, the people I interact with and the role I take partly determines what IT in use becomes.

Finally, people cannot understand their interaction with IT until the moment of interaction has passed. In other words, individuals make sense of their interactions in *retrospect*. Users and designers of IT can never experience the moment of interaction as it is, because time has always gone by before they know what they have done. Consider, for instance, how people at Statoil re-considered the role of Lotus Notes over time (Monteiro and Hespø, 1999). As it was realized that Notes was something more than a tool for rationalization, the conception of the early use of Notes also changed. The fact that people often perceive their actions in the world as direct and not as activities in a world that has gone by can be understood by Schutz' (1967) definition of time as existing in two different forms: 'pure duration' and 'discrete segments'. 'Pure duration' is a "coming-to-be and passing-away that has not contours, no boundaries, and no differentiation" (Schutz, 1967, p. 47), whereas 'discrete segments', which refers to how people usually know experience, have to do with the fact that people step outside this continual flow and direct attention to an experience. The main difference here is that experiencing is a continual flow, Schutz asserts, while the act of directed attention brackets the continual flow of experience and produces that which turns out to us as experiences. The act of attention presupposes an elapsed, passed away experience, one that is already in the past.

In sum, we have identified four concepts for assessing the emerging process of IT-adaptation. Altogether, these concepts provide a vocabulary for understanding the creation of technological conceptions in IT-adaptation. To further develop these concepts in the context of IT-adaptation, next section outlines a case study of the introduction of a software support system – DDTS – in a fast-growing software company.

4 The FSG-Case

After a brief presentation of the organizational context, this section outlines a description of how DDTS as a system was adapted and re-adapted. The intention is to identify certain episodes of relevance for understanding what happened at FSG.

4.1 FSG – Historical Background and Description

FSG is a software corporation that specializes in developing applications for financial markets. FSG's main product – called Finance Tool (a pseudonym) – integrates decision support, risk management and performance management with effective transaction management. Among its customers one can find organizations like ABB, Ericsson, the European Central bank, Pharmacia/UpJohn and Unilever. In addition to the actual software, FSG also offers support services to its customers.

FSG was established in 1992/1993, and the firm has since then experienced a rapid growth in terms of turnover and number of employees. While there were around 4 employees in 1994, FSG tripled its number of employees the first two years and doubled its number of employees the following years, why its total number of employees in the beginning of 1999 was 193. Among the staff, which all but a few held a University degree, one could find 30 nationalities in 1998. At the time of our study, FSG was established in the Baltic region, Finland, France, Germany, Sweden, South Africa, Switzerland and the firm was also planning to establish an office in the US.

In the end of 1998, FSG was organized as follows. The Sales Department was responsible for the relations with potential as well as existing customers. When an agreement was settled – i.e., a customer decided to acquire FSG's Finance Tool – the Client Relations department took over the customer interaction. This department's primary responsibilities were to implement and upgrade Finance Tool. Furthermore, the department was responsible for after-sales activities such as offering upgraded versions of Finance Tool, support licenses and user training. In cases of implementing or upgrading Finance Tool, the Client relations department set up projects with assigned project managers, financial and technical consultants.

When Finance Tool was implemented, the customer was transferred to Knowledge Center for support and Finance Lab for education, even though the overall customer responsibility was still with the Client Relations department. During this process, the Product Management department analyzed customer requests for new functionality. If Product Management decided that the requested functionality is reasonable, they specified what to be developed and assigned the task to the Development department, which developed new functions. Development's responsibilities also included developing code to solve bugs detected during implementation, upgrading and support.

4.2 Research Methodology

This study can be classified as an interpretive case study (Walsham, 1995). As such, there are certain quality standards and principles that can be used to develop as well as to evaluate the conducted research. As Klein and Myers (1999) claim, one of the most fundamental principles of interpretive research is that of the hermeneutic circle. This principle suggests that a phenomenon cannot be understood without assessing it as a part in a bigger whole. The whole cannot, however, be understood without assessing the meaning of the parts and their relationships. Good interpretive research, therefore, strives for understanding its object of study by re-considering the preconceptions of it through a dialogic process between whole and parts. In our study, we have tried to understand the adaptation of DDTS in FSG by considering certain details in its usage in the context of the surrounding behavioral and structural changes. This attempt is illustrated in our case presentation in section 4.3, where we outline micro-level behavioral changes in the light of structural changes and technological changes.

The study was conducted at site between October 1998 and February 1999. During this period, a total of 19 semi-structured interviews were conducted with staff in different positions. Some participant observation was conducted to complement our understanding of the work setting. The period between 1994 and 1998 has been historically reconstructed by means of both written documentation and interviews. The written documentation consisted of manuals of DDTS and a thorough investigation of the articles published in the corporate journal.

4.3 DDTS – The History of a Software Support System

Software support systems go under many labels – incident tracking systems, helpdesk support systems, defect tracking systems. In common, these systems share an intention to support the relation between software developers and their customers. In addition, these systems are quite often part of the internal support to in-house consultants and client-service representatives in the field. Prior research indicates that the use of software support systems largely influences the organization of work procedures in customer service departments (see e.g., Orlikowski and Hofman, 1997; Pentland, 1992). They define certain ways of working depending on the assumptions and knowledge about support work built-into them. Emerging new ways to conduct this kind of knowledge work are therefore likely to occur as a result of IT-adaptation.

In what follows, three types of changes that evolved around the introduction of DDTS in FSG are outlined: *technological changes, behavioral changes, and structural changes*.

Technological changes:
In 1994, FSG introduced DDTS (Distributed default tracking system) – a UNIX-based software support system developed by Pure Atria. The system was intended to redirect some of the time-and-energy-consuming helpdesk calls to an electronic medium. It was hoped that as a result of this redirection, technical specialists and developers could attribute time to such services in-between other tasks, and not, as it was at this point in time, constantly be interrupted by support calls. With DDTS, however, users called helpdesk that manually entered the problem as an item into the system.

Partly as a result of what was considered as an impractical user interface, a second version of the system was introduced in 1996 in form of an in-house developed World Wide Web-interface to facilitate the interaction with the system. This version of the system was announced in the corporate journal as a system intended to enhance communications between FSG's helpdesk and Finance Tool's users. On the one hand, DDTS would efficiently support users encountering problems with Finance Tool and, on the other hand, the system would help FSG to get direct feedback from users of their products. A very important difference with the new version was that users were requested to register their requests directly into DDTS. This meant that users had to report the problem encountered quite extensively. To be useful for helpdesk, the information had to contain issues such as step-by-step instructions to reproduce the problem, transaction numbers, and system information such as user ID, Finance Tool version and platform used.

A third version of DDTS was launched in 1998. One important feature was the improved search function with which anyone using Finance Tool could enter questions into the system.

Behavioral Changes:

By 1998, there were four different departments – Client Relations, Development, Product Management and Knowledge Center –using DDTS in their daily work. At the Client Relations department, there were two categories – Project Managers and Technicians – using DDTS on a regular basis. For the project manager, DDTS worked as a resource for controlling the project. Using DDTS, the project manager was able to follow how the project developed and foresee the potential problems the project was facing. As what concerns the technicians working at Client relations, DDTS facilitated the discussion of specific problems through its labeling system. At the Product management department, DDTS was used partly as a source of information about customer required functionality and partly to keep in control different bugs and problems reported by customers. It also worked as a communication link between product management and development where product management could specify to development what should be developed and then to follow up if it got developed. At helpdesk DDTS functioned as a communication link between FSG's customers and helpdesk where customers could channel their problems.

We have identified four behavioral changes in the gathered data material. These changes are summarized in Table 3.

I. DDTS required customers to be more precise: During the early use of DDTS, there was only one person assigned to what was then called helpdesk. At that time, customers did not report their problems directly into DDTS; instead, customers called or sent mail to helpdesk, where a helpdesk worker entered the perceived problem into the system. As the helpdesk worker at that time expressed when looking back at what happened, this procedure involved some obstacles:

> ”Despite that customer calls often were time consuming, they were often confusing and got misinterpreted.” (COO assistant, former helpdesk worker)

In order to deal with this problem, FSG launched a second version of DDTS in 1996, where customers were requested to enter their problems directly into the system. While this was easier said than done, FSG’s customers slowly turned to use DDTS more frequently. An unexpected, but perceived advantage of having customers entering their problems directly into the database was that customers had to think through their perceived problems:

> "When you are writing things down, you can be more precise about the real problem and it becomes easier understanding and analyzing the problem…" (COO assistant, former helpdesk worker)

As a result of this change, helpdesk could distribute more time to actual problem-solving.

Behavioral Changes	Outcomes
I. DDTS required customers to be more precise	• Customers had to frame the problem before sending it to FSG. • Helpdesk could distribute more time to actual problem-solving.
II. DDTS-items as obstacles for 'real' work	• Technicians did not care for data quality. • Technicians were interrupted even more because of their view of the system.
III. DDTS as provider of functionality requests	• As an unexpected consequence of using DDTS, Product Management acquired functionality requests directly from customers.
IV. Grasping the parts, but losing the whole	• Product Management became "bug-solvers", rather than product developers. • Strategic development of Finance Tool was partly undermined. • A certain 'top-down' approach to problem-solving was imposed on Product Management.

Table 3. Overview of behavioral changes.

II. DDTS-items as obstacles for 'real' work: While staff at helpdesk experienced the use of DDTS as a relief to the extent that it saved quite a lot of time for them, this was not the overall conception of the system at FSG. The technicians, for instance, whose main responsibility was to implement Finance Tool at customer sites did not appreciate DDTS at all, at least not from the beginning. DDTS was perceived as a disturbing element that interrupted technicians from their daily routines. As one technician that used to work at the Client Relations department expressed it:

> "In certain ways, DDTS-items were annoying in that you got more to do, and in that you always seemed to get into more trouble when trying to handle them… DDTS-items interrupted your every-day work, and these items were things that added to your ordinary workload." (Technician, Knowledge Center).

It was obvious enough that while integrating Finance Tool with other systems or conducting customer specific developments were legitimate forms of work, the technicians considered that supporting customers was not. As a result, this produced some negative feelings about DTTS among the technicians. The usual approach towards items reported via DDTS was carelessness, which in turn created even more work for the technicians who got phone calls from customers having trouble with Finance Tool. These calls usually took even more time to handle than items reported via DDTS. Over time, however, technicians saw some benefits in using DDTS. The fact that every problem reported via DDTS got its own identity made it easier to distribute problems that were hard to solve. Another benefit reported was that everything got documented.

III. DDTS as provider of functionality requests: While technicians experienced DDTS as a disturbing part of their daily work, both Product Management and Client Relations saw the system as a practical resource in conducting their work. For Product Management, DDTS was even considered as a necessity. Product Management's main responsibility was to develop the product on a strategic level. The department processed internal and external functionality requests in order to further developing Finance Tool. To accomplish this, the Product Management department had to stay in touch with Sales, customers and the Development department. As customers started to use DDTS more frequently the system turned out to be a valuable resource for Product Management. It was detected that customers did not only report programming errors into the system, but also asked for new functionality. This change in customer behavior also occasioned a behavioral change of Product Management. Instead of just analyzing so-called GAP-lists delivered by Sales, Product Management started to analyze and handle functionality requests delivered through DDTS.

IV. Grasping the parts, but losing the whole: Despite the fact that DDTS initially was built for administering and solving programming errors, it was also used to handle complex requests concerning new functionality. However, this unintended use of the system had some interesting implications. Consider how the director of Product Management expressed it:

> "Today we are using DDTS a lot, but as a consequence of the fact that DDTS is built for solving programming errors, it controls the work to some extent. Instead of concentrating on the big picture and new areas, we turn out as problem solvers" (Product Management Director)

The fact that DTTS was built for administering small problems had impact on the way staff at Product Management conducted their work. Even big problems like new functionality were divided into smaller items so it could be reported through DDTS.

> "Everything is forced into DDTS. It is a fact that we take big issues, divide them into smaller pieces, DDTS items. This is quite a unnatural procedure." (Product Management Director)

While Product management got useful information about requested functionality through DDTS it also limited their creativity when it came to planning future releases. The underlying premises of DDTS build on small well-defined problems, while some of the customer reported requests were issues of more complicated nature. While DDTS was useful for planning the strategic development of Finance Tool, it also restricted their ability to grasp the whole picture.

Structural changes:

The early use of DDTS was conducted on an typically informal basis. The system was bought from an external vendor, Pure Atria, and there were special organizational arrangements that supported the use and development of DDTS. A first step towards formalizing the use of DDTS was taken in 1996, when the helpdesk function was formally acknowledged. It was first until 1998, however, that DDTS became a strategic system for FSG.

5 Interpreting DDTS in FSG

As indicated in the investigated case, DDTS was not a system ready-made for supporting FSG's intention to provide customers with support of highest quality. After nearly five years of use, one might suggest that the use of DDTS in FSG still develops through small adaptations in an emerging manner. Despite this suggestion, however, the use of the software support system matured over time in the sense that there emerged some institutionalized behaviors that were hardly managed in a straight-forward manner.

Indeed, the early use of DDTS triggered quite a lot of ambiguity, but, perhaps even more interesting, ambiguity about DDTS' role at FSG also occurred at several different occasions during the whole period between 1994 and early 1999. Each such occasion involved a lot of sensemaking where different key groups in the organization re-assessed their view of DDTS. During these re-assessments, or so-called 'windows of opportunity' (Tyre and Orlikowski, 1994), new ways to frame and bracket the system were explored. There were certain occasions of 'bracketing' that had major influence on the emergence of DDTS.

	Technological changes	The role and meaning of DDTS
1994	DDTS – a UNIX-based software support system developed by Pure Atria – was introduced.	Customer-relations technology
1996	An in-house developed WWW-interface was added.	Communications enhancer
1998	Improved search functions.	Knowledge management technology

Table 4. DDTS as an emerging technology.

Consider the announcement of DDTS as, first, a 'communications enhancer' in 1996, and, second, an 'organizational know-how provider' in 1998. These occasions of bracketing were reinforced by the establishment of organizational arrangements such as a formally acknowledged helpdesk in 1996 and its further development into a 'knowledge center' in 1998. There is little doubt that these announcements were triggers of quite major revisions of the role of DDTS in FSG. When helpdesk was transformed into a department called 'knowledge center', an implicit message to FSG's staff was that helpdesk is not solely a support service, but an important unit for sharing knowledge within the organization as well as in their customer relationships. Suddenly, what used to be an ordinary software support system was developed into a knowledge management technology. To be sure, the DDTS database contained a lot of information useful for developing Finance Tool as well as existing customer relationships. By launching a new department built around the DDTS system, FSG invented or constructed a meaningful context for using the system.

One interesting observation is how the technicians' disinterest in customer support and DDTS resulted in even augmented the negative implications for their own involvement in support activities. Consider, for instance, how the technicians enacted DDTS-items as obstacles for 'real' work. When considering DDTS as irrelevant for 'real' work, the technicians' presuppositions were confirmed by the resulting effects of their initial view creating a self-fulfilling prophecy where no learning occurred. It seemed that they enacted some deeply embedded 'programs of action' that disabled them to see DDTS as anything else then an obstacle in their work. At this point, it seemed that a changed attitude towards DDTS required that the technicians altered their view of very fundamentals of FSG's activities. The establishment of a 'Knowledge Center' mentioned above might a move from management to influence this. This establishment certainly has the potential to trigger the sensemaking processes that make such alteration possible.

Another interesting observation concerns how the use of DDTS influenced the Product Management department to concentrate on small adjustments of Finance Tool, at the expense of more strategic developments. The details required submitting an item into the DDTS database did not encourage what might be called higher-level problems. If not impossible to submit, such problems had to be divided into smaller problems constructing a barrier for end users to participate in more fundamental issues. Instead, customer requirements concerning more fundamental adjustments of Finance Tool had to be mediated through sales in the beginning of the product acquiring process. In sum, the more customer requirements were mediated through DDTS, the more attention the Product Management department had to direct to small adjustments. DDTS can therefore be said to construct and maintain a certain way of further developing the Finance Tool, which, on the other hand, was found meaningful for people working at product management as they had, if using the system, direct access to customers.

Finally, one might consider that DDTS had some unintended consequences of considerable relevance for understanding how this technology was adapted. While DDTS was intended to facilitate relations between FSG and customers using their financial software, it also meant that the customers could access the present state of their submitted requests. This transparency – which was realized by DDTS – was not an unintended consequence that was appreciated by the technicians. Understanding this, can shed new light on the context in which technicians was one of the key negative groups about DDTS's potential. DDTS worked as an organizing principle for how technicians were evaluated by customers and, indirectly, also by management.

6 Concluding remarks

There is little doubt that society at large can be considered as more unpredictable and dynamic than it used to be. The emerging global markets coinciding with, for instance, a more wide-spread deployment of electronic commerce, seem to make whole businesses fragile and sensitive towards both emerging new technologies and ways of competing.

As a response to this development, many organizations try to be as rational as ever. The increasing complexity is responded to by launching models such as the CRM model for identifying key processes in customer relationships. The more predictions and anticipations made, however, the more uncertainty tends to arise when predictions fail. Our suggestion is not at all to let go, to adopt some sort of a ‘laissez-faire’ strategy to meet the challenges ahead. Rather, we subscribe to Ciborra’s (1996b) view that we need to cope better with ambiguity as a normal ingredient in every-day business activity. Instead of analyzing the integration of IT and organization by planning carefully, we emphasize a greater space for experimenting and improvisation at the level of practical day-to-day activity. This space cannot, of course, replace the need for formal methods of modeling IT and business transformation, but it is a good candidate for inventing meaningful use of IT.

This paper consolidates the growing body of knowledge on IT-adaptation. The paper develops an assessment of how technological conceptions are developed in IT-adaptation. By introducing concepts such as bracketing, self-fulfilling prophecies, identity-construction and retrospection, the complex sensemaking processes shaping the emergence of IT in use can be better understood. In the investigated case, DDTS was associated with new meanings over time as a result of sensemaking processes. Launched as a customer-relations technology, it was by 1998 considered meaningful as a knowledge management technology. This change process can be considered as a result of the surrounding behavioral, structural as well as technical changes involving attempts to making sense of a messy organizational reality.

7 References

Akrich, M., & Latour, B. (1992). A summary of a convient vocabulary for the semiotics of human and nonhuman assemblies. In W. E. Bijker & J. Law (Eds.), *Shaping technology/building society* . Cambridge, MA: MIT Press.

Alchian, A. (1963). Reliability of Progress Curves in Airframe Production. *Econometrica, 31*, 679-693.

Argyris, C., Putnam, R., & McLain-Smith, D. (1985). *Action science*. San Francisco: Jossey-Bass.

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

Barley, S. R. (1986). Technology as an Occasion for Structuring: Evidence from Observation of CT Scanners and the Social Order of Radiology Departments. *Administrative Science Quartely, 31*(1), 78-108.

Callon, C. (1991). Techno-economic networks and irreversibility. In J. Law (Ed.), *A sociology of monsters. Essays on power, technology and domination :* Routledge.

Ciborra, C. U. (Ed.). (1996a). *Groupware and Teamwork - Invisible Aid or Technical Hindrance?* Chichester: John Wiley & Sons.

Ciborra, C. (1996b). Introduction: What Does Groupware Mean for the Organizations Hosting It? In C. Ciborra (Ed.), *Groupware and Teamwork* (pp. 1-19). New York: John Wiley & Sons.

Ciborra, C. U. (1997). De profundis? Deconstructing the concept of strategic alignment. *Scandinavian Journal of Information Systems, 9*(1), 67-82.

Ciborra, C. U., & Hanseth, O. (1998a). Toward a Contingency View of Infrastructure and Knowledge: An Exploratory Study. *Proceedings of the 19th ICIS*, 263-272.

Ciborra, C. U., & Hanseth, O. (1998b). From tool to Gestell: Agendas for managing the information infrastructure. *Information Technology & People, 11*(4), 305-327.

Ciborra, U. C., & Lanzara, G. F. (1994). Formative Contexts and Information Technology: Understanding the Dynamics of Innovation in Organizations. *Accounting, Management & Information Technology, 4*(2), 61-86.

Conway, R. W., & Schultz, A. (1959). The Manufacturing Progress Function. *Journal of Industrial Engineering, 10,* 39-54.

Cyert, R. M., & March, J. G. (1992). *A Behavioral Theory of the Firm. (2 ed.).* Cambridge, Mass.: Blackwell.

Giddens, A. (1984). *The Constitution of Society - Outline of the Theory of Structuration*. Cambridge: Polity Press.

Hanseth, O., & Monteiro, E. (1997). Inscribing Behaviour in Information Infrastructure Standards. *Accting., Mgmt. & Info. Tech., 7*(4), 183-211.

Heidegger, M. (1962). *Being and time.* Oxford: Blackwell.

Heidegger, M. (1977). *The Question Concerning Technology*. New York: Harper and Row.

Henfridsson, O. (1999). In Touch with Technology: On 'Ad-Hoc Learning' in Technological Adaptation. To appear in *the Proceedings of ECIS'99*.

Klein, H. K., & Myers, M. D. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quartely*.

Kling, R. (1991). Computerization and Social Transformation. *Science, Technology and Human Values, 16*(3), 342-367.

Latour, B. (1987). *Science in Action*. Cambridge, MA: Harvard University Press.

Mead, G. H. (1967*). Mind, Self, and Society - From the Standpoint of a Social Behaviorist*.

Chicago: The University of Chicago Press.

Monteiro, E., & Hanseth, O. (1995). Social shaping of information infrastructure: on being specific about the technology. In W. Orlikowski, G. Walsham, M. R. Jones, & J. I. DeGross (Eds.), *Information technology and changes in organisational work*. (pp. 325 -- 343): Chapman & Hall.

Monteiro, E., & Hespø, V. (1999). Diffusion of Infrastructure: Mobilization and Improvisation. In T. J. Larsen, L. Levine, & J. I. De Gross (Eds.), *Information Systems: Current Issues and Future Changes (pp. 255-274)*. Laxenburg, Austria: IFIP.

Orlikowski, W. J. (1992). The duality of technology: Rethinking the concept of technology in organizations. *Organization Science, 3*(3), 398-427.

Orlikowski, W. J. (1996a). Improvising Organizational Transformation Over Time: A Situated Change Perspective. Information Systems Research, 7(1), 63-92.

Orlikowski, W. J. (1996b). Learning from Notes: Organizational Issues in Groupware Implementation. In R. Kling (Ed.), *Computerization and Controversy (pp. 173-189)*. San Diego: Academic Press.

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research, 2*(1), 1-28.

Orlikowski, W. J., & Gash, D. C. (1994). Technological Frames: Making Sense of Information Technology in Organizations. *ACM Transactions on Information Systems, 12*(2), 174-207.

Orlikowski, W. J., & Hofman, J. D. (1997). An Improvisational Model for Change Management: The Case of Groupware Technologies. *Sloan Management Review*(Winter), 11-21.

Orlikowski, W. J., & Robey, D. (1991). Information technology and structuring of organisations. *Information Systems Research, 2*(1), 1-28.

Pentland, B. T. (1992). Organizing Moves in Software Support Hot Lines. *Administrative Science Quartely, 37*, 527-548.

Porac, J. F., Thomas, H., & Baden-Fuller, C. (1989). Competitive Groups as Cognitive Communities: The Case of Scottish Knitwear Manufacturers. *Journal of Management Studies, 26*(4), 397-416.

Rogers, E. M. (1995). *Diffusion of Innovations*. (4 ed.): Free Press.

Schutz, A. (1967). *The Phenomenology of the Social World*. Evanston, IL: Northwestern University Press.

Tyre, M. J., & Orlikowski, W. J. (1994). Windows of Opportunity: Temporal Patterns of Technological Adaptation in Organizations. *Organization Science, 5*(1), 98-118.

Walsham, G. (1993). *Interpreting Information Systems in Organizations.* Chichester: John Wiley & Sons.

Walsham, G. (1995a). Interpretive case studies in IS research: nature and method. *Eur. J. Inf. Systs.*, 4, 74-81.

Walsham, G. (1995b). The emergence of interpretivism in IS research. *Information systems research, 6*(4), 376-394.

Weick, K. E. (1979). *The Social Psychology of Organizing. (2 ed.).* New York: McGraw-Hill.

Weick, K. E. (1990). Technology as equivoque: Sensemaking in new technologies. In P. S. Goodman & L. Sproull (Eds.), *Technology and organizations* . San Fransisco: Jossey-Bass.

Weick, K. E. (1995). *Sensemaking in Organizations*. Thousand Oaks: Sage.

Zuboff, S. (1988). *In the age of the smart machine: The future of work and power*. New York: Basic Books.