

Association for Information Systems AIS Electronic Library (AISeL)

DIGIT 2004 Proceedings

Diffusion Interest Group In Information
Technology

2004

MODIFICATIONS AND INNOVATIONS TO TECHNOLOGY ARTIFACTS

KEVIN DESOUZA

University of Illinois at Chicago, kdesou1@uic.edu

YUKIKA AWAZU

Institute for Engaged Business Research, awazu@engagedenterprise.com

ARKALGUD RAMAPRASAD

University of Illinois at Chicago, prasad@uic.edu

Follow this and additional works at: <http://aisel.aisnet.org/digit2004>

Recommended Citation

DESOUZA, KEVIN; AWAZU, YUKIKA; and RAMAPRASAD, ARKALGUD, "MODIFICATIONS AND INNOVATIONS TO TECHNOLOGY ARTIFACTS" (2004). *DIGIT 2004 Proceedings*. 3.

<http://aisel.aisnet.org/digit2004/3>

This material is brought to you by the Diffusion Interest Group In Information Technology at AIS Electronic Library (AISeL). It has been accepted for inclusion in DIGIT 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

MODIFICATIONS AND INNOVATIONS TO TECHNOLOGY ARTIFACTS¹

KEVIN C. DESOUZA*

Department of Information & Decision Sciences
University of Illinois at Chicago
601 South Morgan Street, M/C 294
Chicago, Illinois 60607
USA
Phone: +1 312 829 8447
E-mail: kdesou1@uic.edu

YUKIKA AWAZU

Institute for Engaged Business Research
Engaged Enterprise
555 West Madison Street
Tower1, Suite # 3705
Chicago, Illinois 60661
USA
E-mail: awazu@engagedenterprise.com

ARKALGUD RAMAPRASAD

Department of Information & Decision Sciences
University of Illinois at Chicago
601 South Morgan Street, M/C 294
Chicago, Illinois 60607
USA
E-mail: prasad@uic.edu

*** CORRESPONDING AUTHOR**

NOVEMBER 17, 2004

**Paper Presentation at Diffusion Interest Group in
Information Technology - DIGIT 2004 (Washington, D.C)**

¹ We thank members of DELTA for their participation in the project. We also would like to acknowledge comments received by two anonymous reviewers that helped us improve the quality of the paper.

MODIFICATIONS AND INNOVATIONS TO TECHNOLOGY ARTIFACTS

ABSTRACT

What happens to a technology artifact after it is adopted? It has to evolve within its particular context to be effective; if it doesn't, it will become part of the detritus of change, like the many genes without a discernible function in a living organism. In this paper, we report on a study of post-adoption technology behavior that examined how users modified and innovated with technology artifacts. We uncovered three types of changes conducted to technology artifacts: personalization, customization, and inventions. Personalization attempts are modifications involving changes to technology parameters to meet the specificities of the user; customizing attempts occur to adapt the technology parameters to meet the specificities of the user's environment; and inventions are exaptations conducted to the technology artifact. The paper presents a grounded theoretic analysis of the post-adoption evolution based in-depth interviews with 20 software engineers in one multi-national organization. We identify a life-cycle model that connects the various types of modifications conducted to technology artifacts. The life-cycle model elaborates on how individual and organizational dynamics are linked to diffusion of innovations. While the research is still in progress and the post-adoption evolution model has to be refined, the research has significant value in understanding the full life-cycle of adoption of technological artifacts and how is maximum value derived from them.

1. INTRODUCTION

The current IS literature on innovations, is rich in studies addressing factors that contribute (or hamper) the decision to adopt technologies (Zmud, 1983; Swanson, 1994; Rai, 1995). In comparison, to this rich literature there are only a few studies that have examined post-adoption decision behavior (Limayem et al. 2003; Bhattacharjee 2001). As noted by Bhattacharjee (2001), “while the initial acceptance of [IT] is an important first step towards realizing [IT] success, long term viability...and its eventual success...depend[s] on its continued use rather than first time use”. Our understanding of how users interact with technologies after their decision to adopt is scant. Researchers like (Orlikowski, 1993; Majchrzak et al. 2000; Poole and DeScantis, 1990; Orlikowski and Yates, 1994; Yates and Orlikowski, 1992) have built on the work of Giddens (1984) to uncover the dynamics of technology structuration in collected (i.e. teams, groups, or organizations) settings. Their findings tell us that technology use does not occur in a deterministic fashion, rather it is emergent. Technology is frequently structured by the individuals to meet their contexts. While we know that technology gets structured we do not know the nature of these structurations. In this paper, we will describe three ways in which users modify (structure) technology artifacts.

The IS literature, has also, for the most part treated users as passive in takers of technology. To do this, is to ignore the fact that users are “knowledgeable” and are “creative” in how they use technology. As rightly pointed out by Nambisan et al. (1999, p. 365) “Technology users, by and large, have been treated as passive recipients of innovative artifacts. Indeed, a dominant view in the IS innovation literature continues to be a technology transfer perspective where the locus of creative activity is the IT organization”. With the trends in current technology development we cannot afford to ignore user-technology interaction dynamics. As noted out by von Hippel and Katz (2002), and Thomke and von Hippel (2002), customers (users of technology) are innovators. Many organizations have abandoned the act of trying to figure out customer requirements in the design process of product development, and have equipped users with toolkits. This is because much of

the information possessed by customers is “sticky” (von Hippel, 1991); hence the customer has a hard time articulating these needs to the product designers. User toolkits allow the customers to conduct innovations and build variations or products to meet their idiosyncratic and peculiar needs. The use of toolkits has been shown to increase customer satisfaction, save organizations the cost and effort involved in articulating user needs, and also reduce design and development times (Thomke and von Hippel, 2002). Now, if we were to examine the state-of-the-art in information technology and systems development we see a similar movement occurring. Traditionally, organizations spent great effort, time, and resources to elicit user requirements prior to systems development. The use of the waterfall development model was popular, however, as we quickly realized, users cannot clearly articulate their needs. More frustrating for designers was the fact that requirements changed on a temporal basis. This resulted in poor quality of systems development, runaway projects, decline of trust in the IS/IT function, and many more adverse effects (Keil and Rai, 2000). Today, we are moving to more agile development methods e.g. Extreme Programming. The goal here is to bring the user into the design and development phases. By bringing the user closer to the design, feedback will be forthcoming on a regular basis and a more acceptable system will be calibrated. The next logical step is to put the user in control of innovations and modifications to technology artifacts. In the mobile phone industry, users are being provided with toolkits that they can use to customize the interface, write their own code, develop their own procedures, write games, etc (Füller et al., 2004).

As researchers we must focus more energies to gain an understanding of how users *modify* (a form of innovation) technology artifacts. An understanding of this will help us better prepare for innovations and manage the innovation cycles. For instance, organizations can use the innovative power of users to decide how to enhance its existing product offerings and to understand future trends in the marketplace. As pointed out by von Hippel (1996), in his conceptualization of “lead users” - “Lead users face needs that will be general in a marketplace – but face them months or years before the bulk of the marketplace encounters them”. These users innovate with technology

at a rapid pace, and have foresight as to the future enhancements and updates needed to current technologies and applications. As such, an organization will be foolish not to tap into them for market and forecasting insights, moreover, their “modification” to the technologies could be used in future version or product updates.

Given the above gaps in the literature, the goals of this paper are – [1] to elaborate on ways users modify technology artifacts and [2] to highlight a generic process model that connects the various types of modifications in a process maturity fashion. The rest of the paper is organized as follows. Next, we briefly elaborate on our methodology. In section 3, we will highlight the various kinds of modifications. Following this, in section 4, we will propose the generic life-cycle model and also discuss a few variants that might exist. Section 5, concludes the paper with a look at our ongoing research, and implications for practitioners and scholars.

2. METHODOLOGY

The focus of this paper is on theory building rather than theory testing. Due to the lack of existing frameworks to guide our investigation and due to the novel nature of the phenomenon being examined we chose to conduct a qualitative research study (Trauth, 2001; Benbasat et al., 1987). Among the rich array of qualitative methodologies available, we chose to approach the research questions using a grounded theory approach (Glaser and Strauss, 1967). The grounded theory approach has several salient peculiarities that make it apt for the current research. The aim of grounded theory is to allow the *theory to emerge*, rather than impose an existing theoretical frame of reference (Glaser and Strauss, 1967). Grounded theory allows for interactive data collection. The researcher can begin coding once data is collected, where ambiguity and equivocality exists, the researcher is allowed to go back to the research site and seek clarification. New information is then synthesized with the existing conceptualizations and reinforcements or modifications are conducted. Following, Tyre and von Hippel (1997), Van de Ven and Poole (1990), and Van de Ven

and Polley, (1992), we focused on specific events as the unit of analysis. The event of interest was the modification of the technology artifact by the user.

Table 1: Demographics on Interview Respondents

Mean Tenure in the Organization	6.23 Years (Min 1.2; Max 8.2)
Mean Tenure in Software Engineering Positions	1.82 Years (Min .5; Max 2)
Previous Employment Positions	
Accounting and Financial Analyst	1
Business Analyst	5
Marketing and Client Services	7
Management Consultants	7
Education and Professional Training	
Number that Possessed Formal IS Education	0
Most Common Education Degree	Bachelors Degree in Marketing and Management
Highest Education Level	MBA (<i>n</i> =4)
Lowest Education Level	Some College (<i>n</i> =1)
Information Systems and Programming Experience	
Expertise in Programming	Novices (10); Novices-Average (6); Average-Expert (4); Experts (0)
Attended Programming Computer Classes / Certificate Programs	Yes (12); No (7); In-Process (1)
Number of Programming Computer Classes / Certificate Programs	2 (Min: 0; Max: 3)

Data for the study was gathered from one organization. The organization, DELTA, a pseudonym, is in the software development business. The organization has offices in 6 North American locations, 2 Europe locations, and 1 location each in Australia and South America. We gathered data from the US Midwest location. Data was gathered through multiple mediums. First, the organization decided to allow its software engineers to participate in a survey. The survey elicited ways in which the engineers modified their Integrated Development Engine and basic demographic information about the software engineers, such as tenure with the organization, experience with programming, etc (see Table 1). Table 1 contains demographic information on our interviewees. We elicited questions in four main areas - employment histories; education and professional training, information systems and programming experiences, and basic information on the type of modifications conducted to the IDE. Table 1 reports on the first three types of information elicited, and we will describe the type of modifications in the rest of the paper. This

survey helped us gauge the kinds of modifications we would encounter and gave us a background to begin interviews.

DELTA's software engineering force was peculiar in one respect. Most of the software engineers did not come from traditional programming or computer science backgrounds; rather, they were originally in the business and management domains of the organization. These included being in areas such as marketing, consulting services, operations management, and even accounting and financial functions. This salient point makes our findings more interesting, as our sample of software engineers truly represented "customers" of technology. We gathered data from software engineers who were relatively new to the organization or who had been programming for no more than 2 years. Most of our interviewees had transitioned into their "programming" roles due to downsizing efforts at the organization. The organization decided that it was in their best interest to have individuals who possessed business knowledge conduct the design functions as well so that there would be less ambiguity and risks in understanding client needs. We chose to focus our attention on 'new' software engineers as this allowed us to gain a sense of how novices to given a technology would engage in customization, modifications, and innovations to the technology artifact. Prior research has shown that novices and experts do not solve problems or approach problem formulation in similar manners (Simon, 1947). Focusing on novices allows us to construct a process model of how changes to the technology occur as the individual improves her knowledge of the application and also the individual's surrounding (her team or group) gets acquainted with the application. Focusing on 'new' engineers affords us an opportunity to understand the life-cycle of modifications and innovations to technology artifacts that will not be possible if our sample consisted of 'expert' engineers or those who possessed significant experience-bases.

We interviewed 20 software engineers on their usage of an Integrated Development Engine (IDE), specifically the Microsoft Visual Studio. Software engineers were asked to elaborate on the nature of modifications conducted to the IDE and the antecedents and consequences of these changes. We asked them to try to recollect their experiences from the most distant event they

remembered in terms of modifying the IDE. Using this as a starting point, we moved to the present time. Each interview lasted for about one hour. Interviews were recorded and later transcribed for analysis. In addition to interviews we examined the technology artifacts. Interviewees were asked to bring their laptops to the interviews; this enabled them to visually demonstrate the nature of the modifications conducted to the default IDE interface. Conducting observations to the technology artifact enabled us to verify the accuracy of the software engineer's recollection, and also enabled us to check for over or under-estimation of the nature of the modification.

Before, moving on the rest of the paper, cautionary comments are in order about the methodology. This study is on going. We are yet to analyze all of the data as such our findings are preliminary, we may even have to go back to DELTA to seek more information. Second, even though we strived to reduce errors due to recallability of past events by observing the modifications to the technology artifacts, these errors might still be present. We asked engineers about historic events and did not actually see the factors that led up to the modifications. Hence, as part of our on going effort, we plan to investigate the research question in a different organization to seek further validity of our findings. Even with these cautionary comments, we believe the contributions of this paper are significant and warrant discussion in the IS community.

3. MODIFICATIONS TO TECHNOLOGY ARTIFACTS

Individuals use technology to accomplish needs. This need can be one of administrative assistance (e.g. use of a calculator), strategic planning (e.g. decision support tools), and even for entertainment and leisure (e.g. playing computer games). Regardless of the type of need, users of technology are rational, in that they will use technology when there is an economic justification to do so. It is this rationality, which makes for the underpinnings of conducting modifications to technology artifacts. Consider a simple example, if you are accustomed to having page margins set at 1 inch all around, and the default on your word processor is 1.25 inches, what will you do? One option, the non-economical one, is to manually alter the page setup for every document you create.

The more economical approach is to customize (modify) your word processor to meet your needs. This modification needs to be done once to suit your needs. The costs of conducting the modifications are low compared to the future benefits.

Table 2: Definitions of Modifications²

Type	Definition	Example
Personalization	Changes to the technology artifact by modifying pre-defined user options to meet the needs of the individual user.	Personalizing the appearance of Toolbars
Customization	Changes to the technology artifact by modifying pre-defined user options to meet the needs of a collected setting.	Customizing the directory structures for program output to meet organizational standards
Invention	Changes to the technology artifact by creating add-ins or using existing functions for novel purposes	Inventing debugging add-ins to facilitate effective and efficient testing of software modules

Modifications to technology artifacts can be simple and complex (see Table 2). The motivation for modification can be to meet the needs of the user or to meet the needs of a collective entity such as a team or organization. In our research, we uncovered three types of modifications. Modifications to technology can be examined based on two criteria – scope and role (Orlikowski, 1992).³ What comprises the technology can be considered as scope and how the technology is used in the organization is the role. In this paper, we will focus on the changes to the scope of the technology. However, we must admit that some of the changes to the scope will impact the role of the technology. An IDE has a definite purpose, which can be broadly stated as enable for the effective and efficient generation and management of software applications. While modifications to the scope of the technology will not result in a change to the overall goal of the IDE, it might help users realize new facets and functionalities that were latent. For the purposes of this paper we view modifications and innovations to technology artifacts as type of technology structuration. Specifically, we are concerned with technology structuration involving changes to the components

² We thank an anonymous review for helping us clarify our thinking on the definitions and labels for each type of modification.

³ We thank an anonymous reviewer for drawing our attention on this point.

of the technology artifact, i.e. changes to items belonging within the scope of the technology artifact.

Personalization: Modifications for Flexibility

Personalization is defined as changes to the technology artifact by modifying pre-defined user options to meet the needs of the individual user. A user's need for flexibility when working with a technology artifact drives the need for personalization. Consider the following comments by software engineers in Table 3.

Table 3: Interview Quotes on Personalization

“If you were to visit most of our start-up screens [of the IDE]...you would find that they are all different...some have 10 icons for a File Menu...some may have 20 [icons]...some others have toolbars combined...mine reflects the most common options I use...”
“Being in QA [Quality Assurance]...my world rotates mostly on two views of the IDE...the testing and debugging options...are what I need to use the most often...having them buried down two levels is a pain...I move them to the foreground...”
“I like my screen layout the following way...this is how I have it on my home PC...it gives me a comfort zone...I do not have to scramble around every time...customizing the layout was one of the first things I did...there was enough to learn about the tool without getting confused about locations of familiar items...”

The motivation for conducting acts of personalization is to make the technology *flexible* to the needs of the user. Drawing on the usage of the term flexibility in industrial engineering (Barad and Nof, 1996), we can define being flexible as the ability to work within a given range. In the context of technology, flexibility calls for changing the established parameters of the technology. For example, changing the appearance of a toolbar by moving one or more icons is an act of modification for flexibility. Changing the background of the display screen is also a modification for flexibility. The user is not creating anything new here; rather they are personalizing an existing option of the technology within the bounds set by the technology creator. Personalizing the technology artifact can be seen as a way to make the technology flexible with the users style of work, work practices, and other preferences. In our discussions with software engineers, we found a wide assortment of modifications for flexibility. These included – changing the default directory pointers, customizing the appearance of the screens, customizing drop-down menus, etc.

Personalizing is the most basic and simple in nature, and is driven by the needs of the *user*. Each user will decide, based on their mental models and task peculiarities, the nature and scope of the customization. The more tech-savvy a user is and the more often a user interacts with the technology the greater is the propensity to conduct acts of modifications for flexibility. Economics dictates that a user is better off personalizing the technology artifact once, rather than attempting to modify it in a repeated per-use basis.

Customization: Modifications for Adaptability

Customization calls for making changes to the technology artifact by modifying pre-defined user options to meet the needs of a collected setting. Here the user is *adapting* her technology artifact to make it suitable for effective and efficient conduct of work practices in a collected setting.

Customizations for adaptability are modifications to the technology artifact motivated by the need for effective and efficient conduct of group work. Modifications for adaptability differ from the modifications for flexibility on one salient point – here, the user is adapting to the external environment. Modifications for adaptability are not driven by the individual needs of the user, but are a result of the user’s involvement in an environment. The environment can be the user’s work team, group, or even the organization. Consider the following comments by software engineers in Table 4.

Table 4: Interview Quotes on Customization

<p>“During the initial days, we were ‘experimenters’...none of us knew the most ideal way of doing something...we had twenty different ways of doing something simple...today...we have agreements in place...they determine how we must use the Damn Thing (IDE)...I had to re-organize my directory structures to meet these requirements...”</p>
<p>“Some complain we have standards...but these have come about from screw-ups...mainly the right hand washing away what the left does...so we all customized our tools to some universally agreed dimensions...”</p>
<p>“My initial changes (personalization) are still in place...these do not affect anyone...but how I name files or where I post them or managing version issues, these are bigger than me...we agreed on global standards that all developers would adhere to...these called for changes to our individual IDE environments...small stuff but important...”</p>

In our discussions with software engineers, we deduced that the need to adapt is governed by one's workgroup and project. As discussed at length, in the work on adaptive structuration, the use of technology is socially constructed and is hence influenced by the social context. For example, Orlikowski (1993) demonstrated the existence of the structuration processes in her study of computer-aided software engineering tool adoption. Majchrzak et al. (2000) described the existence of structuration in studying the adaptation of collaborative technology in virtual team settings. We will not elaborate on the work on the adaptive structuration here, however we wanted to acknowledge the contribution of this work to the understanding on how adaptation occurs in group settings.

We do however want to discuss a finding not addressed thus far in the adaptive structuration work – the role of standards. The topic of IS standards has recently been an area of heightened research interest (King and Lyytinen, 2003). In the context of software engineering, customizations occur to meet standards. Standards can be categorized based on the dimensions of purpose (reference point or compatibility) and enforcement (voluntary or mandatory) (Hemenway, 1975; Antonelli, 1994). Software engineers must customize their technology for all combinations of the 2X2 matrix of standards. In the context of working in a group, software engineers have to customize their directory parameters to point to the common repositories in order to jointly work on the code; these represent mandatory standards that seek to enhance compatibility. As most software engineering is now conducted on a global basis, IDEs must be synchronized in terms of language, date, time, etc. These are mandatory standards that seek to enforce clear reference points. Two or more individuals working in close quarters might create their own standards to facilitate ease of sensemaking (Weick, 1979). In our discussions, we were advised of a case where three software engineers voluntarily decided to customize their IDE desktops for uniformity so that each of them could use the other's PC in case one was away from the office and work called for the use of the PC. This represents a voluntary standard meant to increase compatibility and serve as a reference point between the engineers. To summarize, modifications for adaptability are conducted

to customize one's personalized technology artifact to meet the standards set by his/her work group.

Inventions: Modifications for Exaptability

Inventions are changes to the technology artifact by creating add-ins or using existing functions for novel purposes. While adaptation is accumulation of small changes over time to improve an existing function, exaptation is accumulation of small changes that results in development of a new function. Exaptability is defined as the ability to develop new functions or the utilization of a structure or feature for a function other than that for which it was developed through natural selection (Gould, 1991). Inventions include additions to the existing technology artifact and/or discovering new functions for existing components of the artifact. Inventions are commonly defined as either a new combination of components or a new relationship between previously combined components (Henderson and Clark, 1990; Schumpeter, 1939). As pointed out by Schumpeter (1939, p. 88), "innovation combines components in a new way, or...carrying our new combinations". Frustrations with the existing technology artifact coupled with unfilled necessities are critical determinants of one's motivation to invent. Consider the following comments in Table 5.

Table 5: Interview Quotes on Inventions

"The creators of the IDE were forced to be broad and all inclusive in options, tools, functionalities, etc...however we still need more...we have been forced to create our own solutions and add-ons as they are needed for our work..."
"Why did I write this script...to be frank, because I needed it and it saved me time"
"Everyone was using the directory function for storing files...this is nice...but I came up...and it helps in addressing version control issues...we have local and global directories that are synchronized via a routine...naming conventions are addressed in the background and conflicts in version managed...this was not part of the original IDE...but it has sure made things easier here..."

Modifications for exaptability include creating add-ins, scripts, modules, etc to enhance the productivity of the technology. These modifications are "in-addition" to the existing technology and to be used in conjunction with the original technology. In the case of IDEs these add-ins are used to increase the efficiency and effectiveness of programming assignments. For instance, one

developer, working on a financial trading module, was frustrated with the way he could run testing using the default setup of the IDE. His frustration led him to compose a macro that read his test data, ran his program, and outputted results. Results were then fed through a statistical package for analysis and the final output was visually displayed using a graphics editor. Exaptations such as these can be considered as inventions.

In addition to building new components, exaptation is also the use of existing functions in novel ways. These are most commonly referred to as “work-a-rounds”. Due to the limitations with the technology artifact, users find new ways to use existing functions in order to meet their needs. The simplest example is found in the use of statistical packages. Most statistical packages are highly restrictive in terms of the number of variables, types of variables, parameter requirements, etc. To counter these restrictions users create schemes such as “dummy coding of variables” to work around them. In the context of IDEs, we found software engineers also create work-a-rounds for increasing the effectiveness of tasks. Most work-a-rounds created had to do with the testing and debugging phases of writing code. For instance, the uses of the work-a-rounds were common to tweak the input and output of test data. In one case, a software engineer was frustrated with the lack of effective integration between output files of Microsoft’s Excel and Project software. He took it upon himself to build a work-around using Visual Basic that would integrate the two output files so that a project manager could easily move data between a costing tool (that used a spreadsheet interface) and his administration tool (that used a project management/Gantt chart layout).

Exaptations can occur to meet the needs of the individual or a group. Individuals, as the case above described, may get frustrated with the existing functionality of the technology and develop their own inventions. Similarly, a team working on a project may exert effort to innovate a new feature because of the benefits it poses to their project and work. Individually might collectively pool resources in order to build a new technology feature or add-in. As can be witnessed from the proliferations of altruistic software communities, users have a tendency to

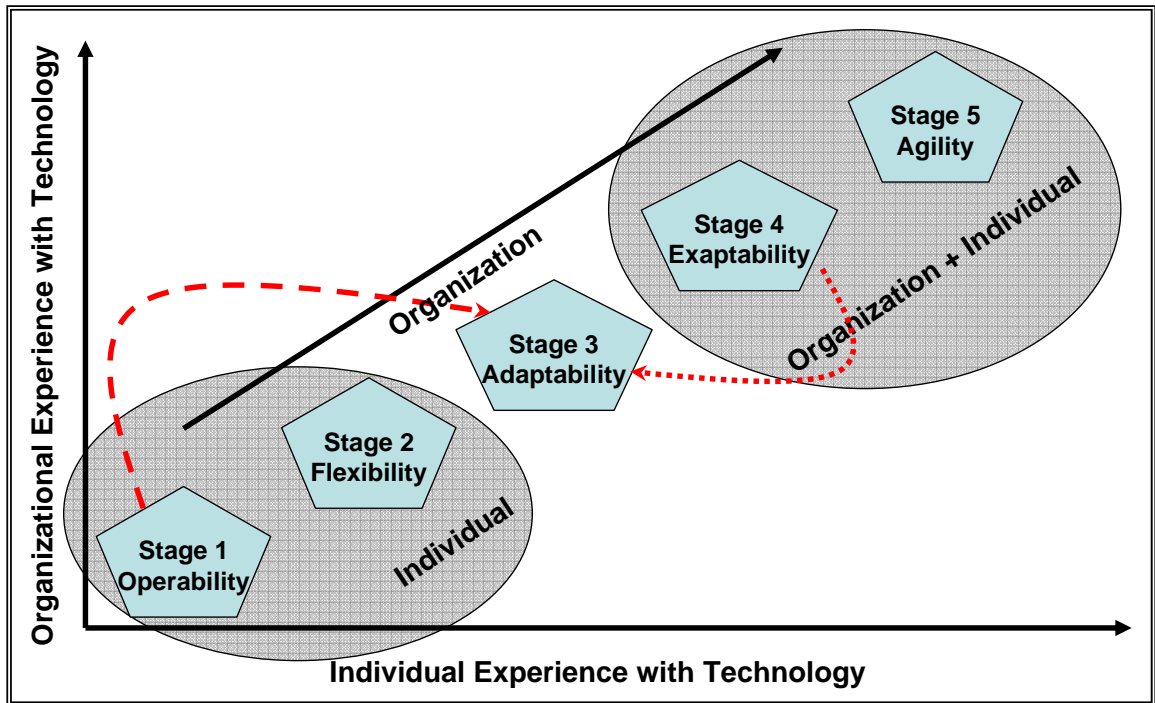
contribute resources when there is hope for a better and more robust solution than what is currently available. Exaptations are the most complex form of modifications conducted to technology artifacts.

4. PROCESS MODEL FOR USER MODIFICATIONS

How do these modifications occur in the organization? We think there is a sequence to them, and will now propose a process model to explicate how modifications occur in the context of the organization. This life-cycle model was deduced from our discussions with software engineers. We must note that this only “one” possible view and was the most dominant view based on our data analysis, towards the end of this section we will discuss possible alternations/variants that are plausible.

As noted by Van de Ven (1992), the first step towards studying a process activity, is to clearly define the meaning of process. In this paper, we use the term process to mean a sequence of events that describe how things change over time. This definition of process, “takes an historical development perspective, and focuses on the sequences of incidents, activities, and stages that unfold over during the duration of a central subject’s existence” (Van de Ven 1992, p. 170). Examples of prior process models in the literature include the work of Mintzberg et al. (1976) on unstructured decision making and Cohen et al. (1972) on garbage can decision models. Upon defining the meaning of process, we must now be specific on the type of process model we are building. In this paper, we are constructing a life-cycle model (Van de Ven and Poole, 1988; Piaget, 1954). Life cycle theories assume that change is evident and that change is recognizable as the artifact is transformed from its present state to a future state (Van de Ven and Poole, 1988; Van de Ven, 1992). As an example of a change model, consider the work of Piaget (1954). Piaget (1954) proposed various stages a child will go through as they learn and acquire knowledge about themselves and their environment, passing through the various stages contributes to the maturity in child development.

Figure 1: Process Model for User Modifications to Technology Artifacts



The model we propose in this paper is a maturity/life-cycle model (see figure 1). It explicates the types of modifications conducted as the user becomes more sophisticated (mature) in their use of the technology. As the users knowledge on how use the artifact increases, we can expect the sophistication of modifications to increase in complexity. In order to increase one's knowledge of the technology, the user must interact with the technology in a frequent manner. Much of the learning associated with technology, is learning-by-doing (Tyre and von Hippel, 1997), rather than learning before doing. Because of the need to repeatedly interact with the technology artifact, users will be rational and find ways to make it economically efficient.

We will now describe the various stages of the process model. The model is composed of five stages. Each stage signifies a maturity level of how the technology is used by the individual. Operability and agility are the beginning and closing stages respectively, and the remaining three signify the types of modifications that occur. The levels are influenced by individual characteristics, the local group the user belongs too, and the organization at-large. We will first focus on the linear trajectory between the various stages, represented by the black line. Following

this, we will discuss some variations that are plausible, these are represented by the dotted red lines.

Stage 1: When a user is first introduced to a given technology artifact, he/she must learn the “bear essentials” needed to get the technology in a state in operation – the “operability” stage. The operability stage is influenced by whether the user has had prior exposure to the technology (e.g. a past version of the software) and has prior exposure with similar technology artifacts (e.g. prior use of Notepad or WordPad will help a user gain operational knowledge of how to work with Microsoft Word) (Huber, 1990; Tyre and von Hippel, 1997). The operability stage is also present when users take it upon themselves to experiment with new technologies without organizational mandate. For instance, the diffusion of SMS messaging systems in organizations has been shown to occur from a bottom-up approach. A select group of users may begin to use it to enable easy communication, and then the use may spread to other members of the organization. When the users first begin to explore with new technologies they are left on their own to figure things out, hence they must rely on their personal knowledge or access to personal knowledge resources such as friends who may know about the technology.

Stage 2: Over time and through continued exposure and interaction with the technology artifact, the user will begin to conduct modifications – the “flexibility stage”. These will take the form of personalization. As discussed earlier, personalization enables the user to customize the artifact to meet individual needs and preferences. The users begin to increase their comfort zone with the technology artifact, and in doing so are more capable and amenable to taking risks in personalizing the technology artifact. As one software engineer remarked, “during the initial usage of the IDE, I was scared to mess around...I did not know what would happen if I changed an option...would it be that I would screw things up...this may call for me to re-load everything....”. Once a user attains a comfort zone they are willing to personalize the technology artifact so that they do not want to keep re-doing mundane tasks, such as changing the directory name from the ‘default’ one to one that is needed by their task.

Stage 3: As the technology diffuses through the organization and its usage increases by organizational members, standards will emerge in order for organized work to take place in an efficient and effective manner. At this time users will be forced to *customize* the technology to meet these requirements – the “adaptability stage”. Standards emerge or are enforced for the simple reason of conducting group work in an effective and efficient format.

Stage 4: Users continue to innovate with the technology after adaptation to organizational standards, these innovations lead to the development of novel functionalities – “exaptability” stage. At the exaptability stage users are looking at ways they can push the technology artifact further. This will require users to realize the weaknesses, limitations, and shortcomings of the artifact and building suitable solutions. Not all users will have the capacity to innovate nor the resources required to do so. As noted by one engineer, “Jason [another Software Engineer] is ahead of most on the learning curve...sometimes he comes up with new ways of doing something that the rest of us marvel at...I am not so wise...Plus, I do not have the same time constraints as Jason...he is on projects that have more slack....mine [projects] are factory-minded, in and out, and with the least time and effort...”.

Stage 5: As a user continues to innovate with the technology, he/she will ascend to the status of an expert or super-user. The user becomes knowledgeable about the intricacies of the technology artifact and can make changes to it under pressures of time and resource constraints. The user will be able to work with the technology artifact in an agile manner. The agile stage is characterized by high proficiency in the use of the artifact. At the agile stage, a user is not just using the technology but is exploiting it to the maximum, and figuring out how to enhance it by adding or changing the artifact. In von Hippel’s conceptualization, we can consider users of technology at this level as “lead users” (von Hippel, 1996).

The above model is supported by the literature. Most novice users when first introduced to a technology are overwhelmed by the complexity of the artifact. Due to this overwhelming nature, users opt to satisfice (Simon, 1947). The primary concern of the user is to get the technology

artifact in an operable condition – operability stage. As such, their first response to accumulate basic knowledge on how to operate the artifact without getting bogged down by details. In order to do this, they are likely to engage in acts of exploitation (Cyert and March, 1963). Exploitation of past knowledge they possess is conducted. This past knowledge could be experiences with a similar technology (such as another programming interface), a technology used in the context of a similar task (such as the use of MATLAB for the calibration of financial and statistical operations), or even their own past knowledge about technology in general (such as the use of a drop down list). These recollections are based on previous events; as such they are acts of learning-before-doing. An individual must also engage in learning-by-doing. Only through the process of experimenting with the technology i.e. putting past knowledge to work in the context of the new technology, will a user be able to comprehend whether the past knowledge is of any use or not. Experimentation is a fundamental activity in the calibration of innovations and also is a vital aspect of any learning process (Adler and Clark, 1991; Thomke, 1998, Smith and Eppinger, 1997). As noted by Von Hippel and Tyre (1994, p. 25), “The need for learning-by-doing indicates that the innovation process will often be iterative and that developers typically can’t “get it right the first time””. It is through the continued exposure and experimentation with the technology that the user will increase his/her stock of knowledge regarding the artifact (Huber, 1991; Senge, 1990; Von Hippel and Tyre, 1994).

Prior research on innovation is supportive of our conceptualization of the flexibility stage. Research has shown that users have a greater propensity to conduct innovations in the development of new products and services to meet their local needs rather than engage in acts of innovation which appeal to a broader audience due to the costs involved in securing their innovations (Harhoff et al., 2002). Moreover, users are more likely to use their existing knowledge to develop the innovation rather than search for outside knowledge due to the cost involved in conducting the search (Luthje et al., 2002). Local information is that which an innovator already has on-site prior to innovating. Local information, in our context, is the peculiarities and idiosyncrasies of the user

and how they want to engage the technology. As pointed out by Harhoff et al. (2002), one of the benefits of users first focusing on their local needs is the “low-cost innovation zone”, users need not concern themselves with the needs of the population at-large. Focusing on the needs of the population at-large is risky (as one may not be able to develop innovations beneficial to the rest) and also is costly in terms of effort. As noted by Davenport et al. (2003), in their study of intellectual asset re-use, on average it takes three times more effort to develop a knowledge nugget for use by the organization at-large than to create one for personal use.

We must note that not all users will engage in acts of personalization, to the same degree or frequency. Novice users of technology are on average risk averse, compared to experts (Kahneman and Tversky, 1979). Conducting acts of personalization call for conducting changes to the original parameters of the technology, as such they contain an element of risk. Experts, who possess domain knowledge, will be better able to judge the degree of risk and either conduct or refrain from personalizing the software. Novices may not be able to make this judgment and hence may avoid personalizing the artifact, at least during the initial periods of technology use.

Adaptation will occur as a means to synchronize individual efforts for the achievement of organizational goals. Without adaptation, users will engage in conflicts when the technology is used as there will lack of conformity.

The recent work on user toolkits and customer innovations supports the fact that users have the capability to radically modify products and designs to meet their needs – exaptability stage. The rich literature on decision making (Simon, 1947), has attested to the fact that experts have the ability to deduce patterns with ease, use their rich source of experiences to solve novel problems, and even re-design artifacts. Users that possess deep knowledge about technology artifacts are lead users (von Hippel, 1996). They have foresight and use the technology in an optimal and complete manner; as such they are apt to discover the limitations of the artifact. This forces them to innovate to meet their needs. By example, a novice using Microsoft Excel may not know the limitation of the powerful spreadsheet tool; however, most expert users write their own macros and routines to

improve the functionality of the tool. The agility stage is the end stage of the life-cycle model. Here is where one is deemed an expert in the use of the technology.

Alternative Connections between Stages

While we have discussed a straightforward and linear progression between the various stages, there can be variations. Due to space limitations we cannot cover these in any depth, but we will like to allude to them, and point out that future research is needed to investigate them in more detail.

There can be instances where Stage 1 is followed by Stage 3, with Stage 2 being skipped. This dynamic is possible under several scenarios. First, when the individual user is getting started with a fairly mature technology that has a rich history in the organization. The individual user may need the time and space to get operational with the technology. However, soon after this they will be introduced to existing organizational standards and will be asked to customize their artifact to meet these requirements so that they can begin to conduct work in the collected setting. Only after this, will the user be able to increase his/her exposure to the artifact and begin to reach a comfort zone to personalize the technology. Second, if there is an organization-wide initiative to introduce a software application, chances are high that there would be a dominant group overseeing the effort. This group may calibrate standards and rules to govern the usage of the application so that coordination and compatibility issues are addressed. Individual users, under this scenario, will also be required to customize their individual artifacts to meet these standards and then personalize the rest of the artifact to their peculiarities.

Another common feedback loop is where Stage 4 (Exaptability) is followed by Stage 3 (Adaptability). We posit that this could be common in cases where we have a highly innovative group of technology users. The innovative class of users will constantly see ways to push the boundaries of the technology. In doing so, they will spread such knowledge to other users, both through formal and informal mechanisms. Formal mechanisms include the introduction of procedures and practices in the work projects and assignments. Informal mechanisms include

discussions with peers and through personal interactions. Once these innovations gain traction and become widely acceptable, they will call for re-definition of standards and customizations to the revised standards will follow.

Linking the Individual and Organization Dynamics

The process model displayed in figure 1 is interesting in the fact that it appreciates the role played the individual technology users, his/her group, and the organization at-large. As noted in the diagram, the sophistication of user modifications to technology increases as the individual and organizational experiences with the technology deepens. This by itself is not likely be a novel finding, however, the manner in which increase in experiences of the three entities (individual, group, and organization) affect the kinds of modifications is interesting.

As the technology is first introduced to the users of an organization or a user decides to experiment with new technology, many a times, users are left on their own to figure out how to get it operational to meet immediate needs. Once operational, we see the emergence of flexibility acts to make the technology more suited to the user. The stages of operability and flexibility are largely dominated by individual user decisions and preferences. The role played by the users' local group or the organization at-large is minimal. The reason for this is the fact is, just like the user, the rest of the organization is still grappling with how to use the technology. As such there is not much knowledge in-house to help individual users.

Over time and with experience, individual users become sophisticated and comfortable with the technology, and the use of the technology increases in the organization. Soon, conflicts will arise. This is because there will be lack of compatibility and synchronization in how the technology is used. Economics dictates that it is in best interest of everyone to develop standards. The development of standards can be top-down or emergent. In highly distributed organizations, we postulate, standards are likely to emerge from the bottom up, this is because of the lack of a dominating authority and the differences in technology usage across the various centers. As pointed

out by Desouza and Evaristo (2003) organizations can choose knowledge management standards to emerge from the local offices to the regional centers, the headquarter plays the role of the integrator and manages the various standards. In organizations that are centralized in nature, it is reasonable to expect that standards will be pushed down from the top. A good example here is standard development at Defense installations like the Army or the Navy. Standards are calibrated by a dominant group in the organization, for example a division in-charge of communication may develop communication protocols. These standards are then enforced through out the organization. Regardless of whether standards are developed top-down or bottom-up, there must be enough of critical mass in terms of active users to justify the investment in standards. The development of standards, involves the individual user, his/her local group, the organization at-large. Adaptability will need to occur to meet the standards.

Rationally speaking, standards are not updated in real-time or on a regular basis. Standards are slow to change as it is costly and a resource intensive effort. As such, users seldom stop at the adaptability stage. Users will continue with use of the technology, and continued use will lead them to discover shortcomings with the artifact. They will then engage in acts of exaptation to meet their needs. The exaptation level is where the difference between experts and regular users starts to become clear. Not all users will engage in acts of exaptation. At the exaptability stage, it is critical that an organization have mechanism to connect regular users, their groups, and the organization at-large with the experts who modify the technology. Unless this occurs the organization's experience with the technology may not grow effectively, the experts will increase their personal stock of experience and may use the technology in more effective manners, while the rest of the organization will be struggling with shortcomings and will attempt to work on problems for which a solution already exist (with the experts). If the organization is able to tap into the exaptations conducted by the experts, these can be evaluated by the communities such as the local group the expert belongs to or from the members of the extant organization. If the modifications are found to

be suitable, they can be diffused through out the organization and standards in place can be updated, this being the best scenario for both the individual user and the organization.

Given enough time, usage, and exposure to the technology the individual and/or organization are bound to reach a stage of agility. Organizations that are successful in knowledge sharing and innovation diffusions will become agile due to innovations by individual users and their associated adoption, assimilation, and diffusion in the organization. Less successful organizations may find differences in the knowledge possessed by their users about the technology artifact. There will be “experts” who can work with the technology in an agile manner and the “rest” who have limited knowledge about the technology and its capabilities. This situation will not be ideal for the organization as conflicts in the use of the technology are bound to occur between the two sets of users. DELTA, the research site, designated one day as “Show-Me-Day”. Show-Me-Day was a half-day event that took place once every six weeks and consisted of presentations made by software engineers to their peers. Engineers who had customized, modified, or invented add-ons to the IDE were asked to make brief presentations to showcase their work. These presentations worked as a means to infuse new knowledge into the software engineering community and help stimulate further discussions, critiques, and collaborations on modifying the IDE for the effective conduct of work. This is an ideal way we see both the organization and individuals interacting for innovations with technology artifacts.

To summarize, users do not modify technology artifacts in isolation from the rest of the organization. As discussed above there is a rich inter-play of dynamics between individuals, groups, and the organization in how technology modification is conducted.

5. ON GOING WORK, PRACTITIONER AND RESERCH IMPLICATIONS

Before discussing the on going work and conclusions, we must acknowledge the limitations of the work. First, the above model is one possible explanation to the stages, which users follow in modifying artifacts. It is by no means the only one, variations can and will exist. As we have

briefly alluded to, for some organizations, adaptation may be the first phase. Users will be asked to adapt to the existing standards from the beginning. This is plausible in organizations that have a mature history of information systems development and rigid standards in place, users who join the organization will be asked to conform to these standards. Depending on the rigidity of the technology, acts of modification for exaptability may not be possible. For instance, if the technology is rigid and organizational constraints and regulations prevent user manipulation to its architecture, acts of exaptability will not occur. As we continue our research, we hope to uncover the situations under which the process model presented here works, and also the situations where we may have variants.

Second, we have not discussed the concept of repeated *feedback* loops here. Modification to technology is not a straightforward linear process. Rather, it is one of cycling between the stages and through repeated feedback. Due to space limitations we have not discussed these findings here. However, we would like to acknowledge they exist and are important. For example, once an expert comes up an exaptated way to use the technology and this insight is diffused through out the organization. The user community, with the exception of the expert, may have to start working with the modification at the operability stage. Working at the operability stage, and then moving through acts of flexibility will help the users get a better appreciation for the modified technology artifact, and they may be able to even improve the modification further. Third, we have been limited in presenting our findings for examination of practices in one organization. We understand and acknowledge the issues associated with generalizing our findings. We are also limited in the generalization of our findings about user modifications with technology artifacts outside the software engineering domain.

Third, our findings here must be viewed in light of our research sample – novice software engineers. While we do feel that novice engineers have a lot of characteristics of the general end-user population, and hence our findings may apply to general end-users of technology artifacts, we must test our findings in new samples to gain more support. A possible strategy for future research

might be to see how information systems students interact with programming environments such as and IDE during an introductory programming course. Findings from such an investigation may refute or support our tentative claims.

We view the work presented here as on-going and not completed. To the best of our knowledge, this is one of the only study in the IS literature, with the exception of the work on adaptative structuration, that focuses on understanding the process by why technology is modified by users. It is our goal to complete the validation of our findings by the time of the conference. We plan to elicit data from two more software organizations, one based in Europe and one in Asia. Gathering data from these sites, will allow us to see if there are distinctions in how technology is modified between geographic locations. Cusumano (2004), in his study of the *Business of Software*, noted that software is managed differently in the United States, Europe, and Japan; we will investigate if differences exist in terms of propensity to modify technology artifacts.

The present study has implications for practitioners. In other fields of product development, users are taking more responsibility for product design efforts. We believe that paying attention to how users modify technology artifacts is a viable first step to begin pushing design issues outside the IS organization and to the customer. Customers after all have a better understanding of their requirements than an outsider such as the IS function. We must hence resist the temptation to guess what the user wants, and give them the opportunity to construct their own innovations and products. In order to do this, we must change the way we design software and information systems. Designers must not focus on building an all encompassing system with all the bells and whistles; rather they must exert effort in building a stable environment and workspace with tools. Users can then use the tools to modify the technology as needed to meet their needs. Practitioners, especially those in the software development business, must take a more active role in involving lead users into the design and development cycles. These users represent a viable source of foresight and know-how that is waiting to be tapped into. A handful of organizations

have begun to host “user conferences”, these are forums that bring together lead users of their technologies, in the hopes of stirring discussions and generating innovative ideas.

The work here has significant research implications too. First, the study can be used as a building block towards understanding how users innovate with technology. Currently, not much attention has been focused on this issue. Technology is become ubiquitous and pervasive (Lyytinen and Yoo, 2002). While we may not completely understand why do users adopt technologies, the current IS literature has more than adequately researched this question, we must now focus on the more important question – “what do individuals do with the technology after adoption”. This study has provided tentative answers to this question. We encourage researchers to examine the work in product development, design studies, and engineering management as a guidance as to how might we better understand user innovation with technology. This paper also alluded to the concept of “experimentation”. The concept of “experimentation” has a rich history in the fields of problem solving and learning, using this work as a foundation can allow us to appreciate the iterative nature by which users interact with technology. Users go through the process of trial-error until a successful solution is found. In the case of user innovation with technology, the concept of “experimentation efficiency” as defined by Thomke (1998), “the economic value of information learned during an experimental cycle, divided by the cost of conducting the cycle”, is salient. When an experiment is costly and the incremental value of information learned is small, the experimental efficiency is low. Experimentation efficiency is not a static value; it is dynamic and will change during the process of experimentation. As users become more sophisticated in their usage of technology we can expect their experimentation efficiency to increase, this has implications on how they might move through process model discussed in this paper. Lastly, the study of user modifications with technology has bearings on research on IS standards. Today, we have a proliferation of collective systems for design and development of software such as the many instances of Open Source Development. Here standards are set as to what is acceptable behavior by the user group. These standards *emerge* from the bottom-up rather than being imposed top-down by

a governing body. One argument on why these standards emerge from the bottom-up is that these communities are composed of skilled knowledge workers. These knowledge workers like to be in control of their knowledge and more importantly, like to control how it is used to better technology. Unless we understand how is technology personalized, customized, and exapted by users, we will not be able to truly appreciate the emergence of standards.

In summary, we have described the various types of modifications conducted by users on technology. In addition, we have proposed a process model to link these modifications. We have also provided a rich array of practitioner and research implications. It is our hope, that the paper has opened up an avenue for interesting discussions within the IS community.

6. REFERECNES

- Adler, P.S. and K.B. Clark. (1991). "Behind the Learning Curve: A Sketch of the Learning Process," *Management Science*, 37 (3), 267-281.
- Antonelli, C. (1994). "Localized Technological Change and The Evolution of Standards as Economic Institutions," *Information Economics and Policy*, 6, 195-216.
- Barad, M. and S.Y. Nof. (1996). "Flexibility Measurement: A Framework for Analysis and Decision Making," *International Journal of Computer-Integrated Manufacturing*, 10, 296-308.
- Benbasat, I., Goldstein, D.K. and M. Mead. (1987). "The Case Research Strategy in Studies of Information Systems," *MIS Quarterly*, 11 (3), 369-386.
- Bhattacharjee, A. (2001). "Understanding Information Systems Continuance: An Expectation-Confirmation Model," *MIS Quarterly*, 25 (3), 351-370.
- Cohen, M.D., March, J.G., and J.P. Olsen. (1972). "A Garbage Can Model of Organizational Choice," *Administrative Science Quarterly*, 17 (1), 1-25.
- Cusumano, M.A. (2004). *The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad*. New York: Free Press.
- Cyert, R.M. and J.G. March. (1992). *A Behavioral Theory of the Firm*. Cambridge, MA: Blackwell Business.
- Davenport, T.H., Thomas, R.J., and K.C. Desouza. (2003). "Reusing Intellectual Assets," *Industrial Management*, 45 (3), 12-17.
- Desouza, K.C., and J.R. Evaristo. (2003). "Global Knowledge Management Strategies," *European Management Journal*, 21 (1), 2003, 62-67
- Füller, Johann, Bartl, Michael, Ernst, Holger, and Mühlbacher, Hans. (2004). "Community Based Innovation: A Method to Utilize the Innovative Potential of Online Communities," Proceedings of the 37th Hawaii International Conference on Systems Sciences, IEEE Press.
- Giddens, A. (1994). *The Constitution of Society: Outline of the Theory of Structure*. Berkeley CA: University of California Press.
- Glaser, B.G. and A.L. Strauss. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago, IL: Aldine.

- Gould, S.J. (1991). "Exaptation: A Crucial Tool for Evolutionary Psychology," *Journal of Social Issues*, 47, 43-65.
- Harhoff, D., Henkel, J. and E. von Hippel. (2002). "Profiting from Voluntary Information Spillovers: How Users Benefit from Freely Revealing Their Innovations," MIT Sloan School of Management Working Paper (May), Cambridge, MA.
- Hemenway, D. (1975). *Industrywide Voluntary Product Standards*. Cambridge, MA: Ballinger.
- Henderson, R.M. and K.B. Clark. (1990). "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative Science Quarterly*, 35 (1), 9-30.
- Huber, G.P. (1991). "Organizational Learning: The Contributing Processes and the Literature," *Organization Science*, 2 (1), 88-115
- Kahneman, D. and A. Tversky (1979) "Prospect Theory: An Analysis of Decision under Risk," *Econometrica*, 47:263-91.
- Keil, M, and A. Rai. (2000). "Why Software Projects Escalate: An Empirical Analysis and Test of Four Theoretical Models," *MIS Quarterly*, 24 (4), 631-664.
- King, J.L. and Lyytinen, K. "Standard Making: A Critical Research Frontier for Information Systems", MISQ Special Issue Workshop Seattle, December 13-14, 2003. Available Online at <http://www.si.umich.edu/misq-stds/Workshop/index.html>.
- Limayem, M., Cheung, C.M.K., and Chan, G.W.W. (2003) "Explaining Information Systems Adoption and Post-Adoption: Toward an Integrative Model," Proceedings of the International Conference on Information Systems, Seattle, Washington
- Luthje, C., Herstatt, C. and E. von Hippel. (2002). "The Impact of Use Experience and Knowledge on Innovations: An Empirical Study of Innovative Activities among End-Users," MIT Sloan School of Management Working Paper.
- Lyytinen, K. and Y. Yoo. (2002). "The Next Wave of Nomadic Computing," *Information Systems Research*, 13 (4), 377-388
- Majchrzak, A., Rice, R.E., Malhotra, A., and N. King. (2000). "Technology Adaptation: The Case of a Computer-Supported Inter- Organizational Virtual Team," *MIS Quarterly*, 24 (4), 569-600.
- Mintzberg, H., Raisinghani, D., and A. Theoret. (1976). "The Structure of "Unstructured" Decision Processes," *Administrative Science Quarterly*, 21, 246-275.
- Nambisan, S., Agarwal, R., and M. Tanniru. (1999). "Organizational Mechanisms for Enhancing User Innovation in Information Technology," *MIS Quarterly*, 23 (3), 365-395.
- Orlikowski, W.J. (1992). "The Duality of Technology: Rethinking the Concept of Technology in Organizations," *Organization Science*, 3 (3), 398-427.
- Orlikowski, W.J. (1993). "CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development," *MIS Quarterly*, 17 (3), 309-340.
- Orlikowski, W.J. and J. Yates. (1994). "Genre Repertoire: The Structuring of Communicative Practices in Organizations," *Administrative Science Quarterly*, 39 (4), 541-574.
- Piaget, J. (1954). *The Construction of Reality in the Child*. New York: Basic Books.
- Poole, M. S., and G. DeSanctis, G. (1990). "Understanding the Use of Group Decision Support Systems: The Theory of Adaptive Structuration," in *Organizations and Communication Technology*, J. Fulk and C. Steinfield (eds.), Newbury Park, CA, Sage, pp. 173-193.
- Rai, A. (1995). "External Information Sources and Channel Effectiveness and the Diffusion of CASE Innovations: An Empirical Study," *European Journal of Information Systems*, 4, 93-102.
- Schumpeter, J.A. (1939). *Business Cycles: A Theoretical, Historical, and Statistical Analysis of the Capitalist Process*. New York and London: McGraw-Hill.
- Senge, P.M. (1990). *The Fifth Discipline: The Art and Practice of the Learning Organization*. London: Century Business.
- Simon, H.A. (1947). *Administrative Behavior*. 2nd ed/ New York: The Free Press.

- Smith, R.P. and S.D. Eppinger. (1997). "Identifying Controlling Features of Engineering Design Iteration," *Management Science*, 43, 276 – 293.
- Swanson, E.B. (1994). "Information Systems Innovation among Organizations," *Management Science*, 40 (9), 1069-1091.
- Thomke, S. (1998). "Managing Experimentation in the Design of New Products," *Management Science*, 44 (6), 743-762.
- Thomke, S. and E. von Hippel. (2002). "Customers as Innovators: A New Way to Create Value," *Harvard Business Review*, 80 (4), 74-82.
- Trauth, E. M., (ed.) (2001). *Qualitative Research in Information Systems: Issues and Trends*. Hershey, PA: Idea Group Publishing.
- Tyre, M.J. and E. von Hippel. (1997). "The Situated Nature of Adaptive Learning in Organizations," *Organization Science*, 8 (1), 71-83.
- Van de Ven, A.H. (1992). "Suggestions for Studying Strategy Process: A Research Note," *Strategic Management Journal*, 13, 169-188.
- Van de Ven, A.H. and D. Polley. (1992). "Learning While Innovating," *Organization Science*, 3 (1), 92-116.
- Van de Ven, A.H. and M.S. Poole. (1990). "Methods for Studying Innovation Development in the Minnesota Innovation Research Program," *Organization Science*, 1 (3), 312-335.
- Van de Ven, A.H. and M.S. Poole. (1988). "Paradoxical Requirements for A Theory of Organizational Change," In *Paradox and Transformation: Toward a Theory of Change in Organization and Management*, R.E. Quinn and K.S. Cameron, (eds.), Cambridge, MA: Ballinger, pp. 19-63.
- von Hippel, E. (1991). "'Sticky Information' and the Locus of Problem Solving: Implications for Innovation," *Management Science*, 40 (4), 429-439.
- von Hippel, E. (1996). "Lead Users: A Source of Novel Product Concepts," *Management Science*, 32, 791-805.
- von Hippel, E. and R. Katz. (2002). "Shifting Innovation to Users via Toolkits," *Management Science*, 48 (7), 821-834.
- von Hippel, E. and J. Tyre. (1994). "How Learning by Doing is Done: Problem Identification in Novel Process Equipment," *Research Policy*, 24, 1-12.
- Weick, K.E. (1969). *The Social Psychology of Organizing*. Reading, MA: Addison Wesley.
- Yates, J., and W.J. Orlikowski. (1992). "Genres of Organizational Communication: A Structural Approach to Studying Communication and Media," *Academy of Management Review*, 17 (2), pp. 299-326.
- Zmud, R.W. (1983). "The Effectiveness of External Information Channels in Facilitating Innovation within Software Development Groups," *MIS Quarterly*, 7 (1), 43-58.