DIFFUSING SOFTWARE PRODUCT AND PROCESS INNOVATIONS

Edited by
Mark A. Ardis
Barbara L. Marcolin
DIFFUSING SOFTWARE PRODUCT AND PROCESS INNOVATIONS
IFIP - The International Federation for Information Processing

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP's aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP's events range from an international congress to local seminars, but the most important are:

- The IFIP World Computer Congress, held every second year;
- open conferences;
- working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is less rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.
Contents

Preface ix

MESSAGE FROM THE GENERAL CHAIR

Telling an Innovation Story 1
E. BURTON SWANSON

PART I Implementation and Coordination Issues

A Web Innovations on Software Process-Center for Diffusing Techniques 5
CARLA FREERICKS (BLANCK PURPER)

Diffusion and Adoption of IT Products and Processes in a Danish Bank 17
JAN PRIES-HEJE AND SUSANNE TRYDE

PART II New Interpretations of Diffusion Theory

The Phenomenon of Diffusion 35
TOR J. LARSEN

A Perspective of the Innovation-Diffusion Process from the Self-Organizing System 51
TOSHIO MITSUFUJI

The Illusion of Diffusion in Information Systems Research 67
TOM MCMASTER
### Contents

**PART III  Software Process**

**Understanding and Changing Software Organizations**

*Karlheinz Kautz, Henrik Westergaard Hansen, and Kim Thaysen*

87

**Diagnosing Diffusion Practices Within a Software Organization**

*Ingegerd Andersson and Kerstin Nilsson*

111

**PART IV  Contextual Factors**

**The Diffusion of Components**

*Richard Veryard*

131

**Across the Divide: Two Organisations Form a Virtual Team and Codevelop a Product**

*L. Levine, G. Syzdek*

147

**What's Wrong with the Diffusion of Innovation Theory?**

*Kalle Lyyninen, Jan Damsgaard*

173

**PART V  Communication of Information**

**Influences of Sources of Communication on Adoption of a Communication Technology**

*William D. Stuart, Tracy Callaway Russo, Howard E. Sypher, Thelma E. Simons, Lisa K. Hallberg*

191

**Knowledge Creation in Improving a Software Organisation**

*Pouya Pourkomeylian*

205

**PART VI  Experience Reports**

**How To Live With Software Problems**

*Klaus Jul Jeppesen*

225

**Introducing Concurrent Functional Programming in the Telecommunications Industry**

*Bjarne Dacker*

241

**Change and Adaptive Behavior in Organizations**

*Andrew P. Jansma*

257
Contents

In Search of an Efficient EDIcebreaker 277
HELLE ZINNER HENRIKSEN

Process Definition in Web-Time 293
ALAN S. KOCH
Preface

The International Federation for Information Processing (IFIP) Working Group 8.6 was established by IFIP in 1994 as a working group concerned with diffusion, transfer and implementation of information technology. Among other activities the working group conducts a series of conferences and workshops, including this one, held April 7-10, 2001 in Banff, Canada.

The goal of this conference was to bring together practitioners and academics who share a common interest in the diffusion of innovations related to software products and processes. Special attention was paid to those areas where practitioners and academics could forge new ties and share different points of view.

The editors thank the members of the organizing and programming committees for their help and support. We also thank Yana Lambert, from Kluwer, for her assistance in the preparation of these proceedings.

Mark Ardis and Barb Marcoli
In this conference, we come together to examine the diffusion of software innovations. We consider both software products and processes. For many years, I have had a particular interest in business application software and its design, implementation, use, and maintenance. Most recently, my research has focused on this software in an innovation context. Specifically, I have become interested in software associated with certain grand ideas for innovation, termed "organizing visions," defined as focal community ideas for the application of IT in organizations (Swanson and Ramiller, 1997). Examples of organizing visions would include data warehousing and mining, enterprise resource planning (ERP), and customer relationship management (CRM). I have become curious about the "innovation stories" associated with these visions, and, in particular, about how the innovation arises and takes a certain "career path" in which it achieves ascendancy for a time, but then eventually fades away, often displaced in the community’s attention by still another vision. I believe that if we are to understand these innovations, we must tell the stories of their often inter-twined careers.

While the career of an organizing vision, and the diffusion of the software associated with it, thus takes place at the level of the inter-organizational field, the innovation story can also be told for the individual organization. These individual stories can then be assembled within the broader picture. In the following brief remarks, I suggest how such an individual story might be told.

As I shall describe it, in the context of organizing visions, an organization’s own innovation story has four parts. It begins with the organization’s comprehension of the innovation. It continues in the next part with the innovation’s adoption. In the third part, it tells of its
implementation. It concludes by speaking to the innovation's assimilation. As the story unfolds, it focuses on several questions: What? Why? When? How? Who? I will explain briefly. Table 1 summarizes.

Table 1. Innovation Story Summary

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Comprehension</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adoption</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assimilation</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

An organization innovates when it adopts and implements an idea, practice, or object that is perceived as new to it (Rogers, 1983, p. 11). What is interesting here is that only a small fraction of organizations are the “true innovators” in the sense that they are among the first to undertake an innovation. The rest follow in their footsteps. Once an innovation thus comes to the early attention of a community, each subsequent prospective adopter faces first and foremost its comprehension. What is ERP all about, for instance, and why should firms undertake it? Here the organizing vision for the innovation is engaged as it is being constructed and propagated in the wider inter-organizational community (Swanson and Ramiller, 1997). Indeed, the original vision for ERP was put forth by the Gartner Group in 1990 (Wylie, 1990). It was subsequently stretched and refined and elaborated upon by many interested parties over a decade.

From its comprehension of the innovation, the organization may consider whether and when to adopt it. Once again, it asks why, but now the question is made specific to the firm’s own situation and opportunity for action. Why or not should our firm undertake ERP, for instance, and if it should, when is the right time to do it? Here the risks and possible rewards of undertaking the innovation are assessed. The business case for adoption is assembled. Organizational readiness is further determined. If the firm moves early, might it obtain a competitive advantage? If it moves later, with the majority, might it be more likely to be successful? If it holds out past this period, might the firm fall too far behind and face competitive failure? The stakes may be high. Where the firm thus commits to ERP, for instance, it may budget tens of millions of dollars toward acquiring the software and needed expertise.

Should the organization decide in favor of the innovation and commit monetary and human resources to it, its next task is implementation. Again, it asks the “when” question, but now typically focuses on a project schedule. It also addresses how implementation is to be accomplished. When shall we have our ERP up and running, for instance, and how shall we make it happen? For instance, shall we take a “big bang” approach and implement
the full ERP system at all locations, or shall we take some phased approach (Brown and Vessey, 2000)? Do we have the right “know-how” in our project team to make our ERP implementation successful (Swanson, 2000)? Again, the stakes may be very high. A significant number of ERP implementation projects have run aground, and a few firms may have failed altogether as a consequence (Davenport, 1998).

Finally, once the organization has implemented its innovation, in the sense of making it operational, it is by no means done with it. Now, the innovation must be taken up and absorbed over time by those people intended to work with and benefit from it. Its users must take true ownership of it. Again, the organization may ask how this is to be accomplished. It may further ask “who” the organization might then be in the sense of being transformed by the innovation. How can we improve the use of our ERP such that we benefit best from it, for instance, and who will we then be, among our industry’s global leaders? With ERP, we note that assimilation may not come easily at first, as organizational performance may dip upon installation, as users struggle with the new system (Ross, 1998). Even when users become acclimated to ERP, they may be slow to see its potential and use it to best advantage. Achieving expected ERP benefits can thus be problematic (Davenport, 1998; Willcocks and Sykes, 2000). Some benefits may be much more likely to be achieved than may be others (Swanson, 2000).

In summary, in telling an innovation story such as that for ERP, we must be cognizant of its multiple parts, both at the firm level and at the broader institutional level. Historically, diffusion-oriented researchers have focused primarily on the story’s adoption chapter. Information systems researchers have also over many years also examined aspects of the implementation chapter. Relatively little research in any depth has been done on the concluding assimilation chapter. Here, studies of user satisfaction with systems, for instance, may touch upon assimilation, but they do little to reveal its important elements. Zuboff’s (1984) classic study of the transformation of production work may provide a better example of the type of research needed here. Finally, almost no research has been conducted on the problem of comprehension, which forms the essential first chapter of the innovation story. Our own research on this subject barely scratches the surface (see, e.g., Ramiller and Swanson, 2000). Where organizing visions are concerned, much thus remains to be illuminated.
REFERENCES


A Web Innovations on Software Process-Center for Diffusing Techniques

Carla Freericks (Blanck Purper)
Universität Bremen, FB3 Mathematik/Informatik, Postfach 33 04 40, D-28334 Bremen, Germany
Cpurper@informatik.uni-bremen.de
http://www.informatik.uni-bremen.de/gdpa

Abstract: There are two facets on the diffusion of technological innovations on software process: companies who want to promote their products to the right target public and individuals or institutions who are searching for the right innovation to solve their problems. This paper proposes a bridge between them: a process web-center. It is a web-based system to disseminate integrated, classified and updated information on software process technologies independent of projects, particular research areas or commercial tools.

1. INTRODUCTION

In an effort to boost a strategic place in the highly competitive market, software development organizations are striving to build up, redesign and improve their software development process. Software process technologies are treated by most of the recent software engineering conferences, symposia, magazines and journals. Besides, it is widely accepted that software developed under a process schema will produce better software quality. Lifecycle standards are seen as a means to modernize software development, as instruments for continuous improvement, and as checklists for software assessment. However, information on process technologies and related standards is usually extensive and spread over the internet, dispersed among technical papers and magazines, conferences, scattered in different mailing-lists and forums, process standards, commercial tools, research
Carla Freericks (Blanck Purper)

projects, training courses etc. There is not much on-line material that addresses learning environments for diffusing software process technologies. The few sites available are restricted to specific areas in process technologies that are being promoted by the institutions owning the site. As a consequence, process modelers not only spend valuable time in understanding the new issues in process technologies, but also seeking suitable, up-to-date information about them.

Despite the fact that some organizations create their own niches to collect information on some of the above mentioned subjects, the information is usually focused on a particular process. It seems that there is no widely known national or international organized body or mechanism supplying such information without being bound to a specific project.

This paper exposes applied research in this direction. It describes the process web-centre, entitled GDPA\(^1\) [Purp99a, Purp99b], that is a web-based system to disseminate integrated, structured and up-to-date information on software process technologies independent of projects, particular research areas or commercial tools.

2. DIFFUSION BASIS

Today, after 4 years of continuous development, improvement and service, the process web-centre consists in an interwoven net of more than 8,000 web-pages holding more than 32,000 internal and external links with more than 2,000 accesses to the home-page per month.

One third of the accesses to the process web-center is from German IP-domains, the second third is from ".COM" domains; the remaining accesses are from a variety of countries. More than 70% of the accesses are from commercial companies and ca. 25% are from educational institutions. Around 30% of the accesses are from "frequent users", i.e. users that have periodically accessed the GDPA home-page in the last 4 months. The percentage of frequent users has gradually increased in the last months with a growth of ca. 29% per month.

The key factors of GDPA are:

1) The information is constantly updated

Although GDPA can be completely downloaded at client site, users periodically return to the GDPA home-page to check what is new. The latest news that are shown in the front page are classified in four categories:

\(^1\) GDPA is a tool of the UniForM Workbench [KPOB 99], a project developed by Universität Bremen, Universität Oldenburg, and Elpro LET GmbH, partially sponsored by the German ministry of education and research (BMBF - Bundesministerium für Bildung und Forschung, 1996-1998).
- remind deadlines for conferences, project proposals, etc,
- process community news about products, books, techniques, conferences, etc,
- GDPA news about the functionality of the software,
- V-Model news about the German software development standard.

All previous news can be accessed by subject or by date. The constant update of information is one of the most appreciate aspects in GDPA.

2) All internal and external links are checked weekly

One of the factors that most disincentive the use of a web-site is the obsolescence of the links to external web pages. Once a week, a tool automatically checks all external links of GDPA and produces a report with a list of errors and warnings. In case it is not possible to determine the new address of an invalid link in GDPA, this link is merely not activated until the new address is discovered.

3) The web is the database

In order to allow the installation of GDPA at the client site without requiring extra efforts for setting up and for running on almost all platforms it was necessary to resign from using databases. For this reason, the software was designed so that all pages could be displayed by the typical web browsers.

4) It can be completely downloaded to the client site

GDPA was specially designed to work independently without necessity to remote calls and other on-line requests to operate it. Thus, the user can operate it directly at his workstation or local intranet. This is beneficial for both the user and GDPA. When the user accesses GDPA off-line, he has very low display time per web page. On the other side, the number of accesses to the GDPA server is considerably reduced and hence, the resources are not overloaded.

5) It is free of charge for non-profit use

GDPA is a research project at the Bremen University in Germany for diffusion software process technologies and standards for the IT community. Although GDPA has absolutely no advertisement, the fact of being free of charge has considerably contributed for its dissemination. Another benefit, is the active participation of the users. Monthly, circa 50 e-mails are received to report corrections and provide news for diffusion.

6) It is completely web-based, without complex html frames

This feature is being explored by many companies which are integrating GDPA in their own web-site. Basically, they access each web-page directly without concerning about the split of information among different frames.
3. PROCESS WEB-CENTER

The architecture of GDPA underwent a long and gradual evolution from simple html pages to the actual web-center. The Process Web-Center is an information service on the internet that:

- provides integrated information on software process technologies,
- has a specific "Learning Technology System Architecture" (LTSA),
- whose links are processed at a meta-level,
- stores the standards of software processes into meta-process, and
- applies an ontology library.

3.1 Integrated Information on Software Process Technologies

The following tables present the main data provided in the process web-center GDPA:

a) Software Process Standards

It is widely believed that the use of a predictable and organized software process is strongly correlated with the production of high-quality software. Process standards provide a representation of ideal processes bestowing competitive advantage. This leads to a growing interest of the industry in applying renowned standards to gain certification, and hence acceptance, in the international market-place.

In the process web-center GDPA, emphasis is given to process standards [Purp99d]. However, GDPA is a public domain site and as a consequence, might only contain the text of the standards on software process, available free of charge. Meanwhile, GDPA contains all the 3 books of the standards GD250 also known as the "V-Model" [GD250], GD251 [GD251] and GD252 [GD252] in English and German (ca. 2000 printed pages) [Purp00a].

An outstanding feature of GDPA that differs from conventional approaches is the distribution of the 3 standards in a web-based format built upon the concept of open-source standards. Because of the open-source availability, a vast number of requests to incorporate other international standards have been received. This may suggest that there is an increasing demand for changing the static modality in which standards are currently being distributed to the software community.

Table 1. Software Process Standards

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GD250</td>
<td>German Standard - V-Model</td>
</tr>
<tr>
<td>GD251</td>
<td>Methods Allocation</td>
</tr>
<tr>
<td>GD252</td>
<td>Functional Tool Requirements</td>
</tr>
</tbody>
</table>
b) V-Model Learning Environment

Mähönen [Maho 00] argues that standards are hard to assimilate partly because of the difficulty in understanding the text rather than in implementing the technological aspects. However, one might argue that standards are regulations and are not intended to be auto-didactic instruments. But one is also tempted to say that this argumentation leads to an impasse in the diffusion of process standards. These positions reinforce the idea that a learning environment should be provided externally. Different means such as frequently asked questions (FAQ), mailing-lists and tutorials might be employed for this purpose. When the learning environment for the V-Model standard [Purp00a] was integrated in GDPA, it had an increase of 12% of its accesses.

Table 2. V-Model Learning Environment

<table>
<thead>
<tr>
<th>FAQ</th>
<th>V-Model Frequently Asked Questions (in German)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailing-List</td>
<td>V-Model Mailing-List (in German)</td>
</tr>
<tr>
<td>Introduction</td>
<td>Introduction to the V-Model (in English)</td>
</tr>
</tbody>
</table>

c) Software Process Terminology

More than 800 original definitions in English and German are included in the glossary of GDPA. One entry in the glossary might have more than one definition. For example the word "process" has 8 definitions from different sources. It seems that the glossary in GDPA is mostly appreciated by authors who need definitions for writing papers. Normal users could be satisfied with just one definition for each word. Nevertheless, a few users pointed out that the variety of definitions is useful to understand the flexibility in the terminology.

We use the ontology approach for a comparative evaluation of process terminology and taxonomy [Free00].

Table 3. Software Process Terminology

<table>
<thead>
<tr>
<th>Acronyms</th>
<th>Acronyms ordered alphabetically</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glossary</td>
<td>Glossary ordered alphabetically</td>
</tr>
</tbody>
</table>

d) Software Process Publications

The priority in GDPA is for publications presented in congresses, conferences, workshops and symposia such as the International Process Technology Workshop (IPTW), International Software Process Workshop (ISPW), International Conference on Software Process (ICSP), European Workshop on Software Process Technology (EWSPT), etc. Around 1600 publications are catalogued in GDPA, accessible by different indexes: ordered by author, book, congress or subject.
Despite the fact that the majority of the users of GDPA who are from industry do not have so much time to read research papers, they do find it extremely useful to know when the first research publications on the topics appeared that are currently being presented as a novelty to industry.

To conclude with the issue about publications, let us look at the construction of the index by subject. It took almost 3 months to define and refine a consensual and practical index. It was a complex endeavor and required an excellent knowledge of the subject.

**Table 4. Software Process Publications**

<table>
<thead>
<tr>
<th>Author</th>
<th>Publications ordered alphabetically by author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>Publications ordered chronologically by book</td>
</tr>
<tr>
<td>Congress</td>
<td>Publications ordered chronologically by congress, conference, workshop</td>
</tr>
<tr>
<td>Subject</td>
<td>Publications ordered by subject</td>
</tr>
</tbody>
</table>

**e) Software Process Directories**

The directories add a "personal touch" to GDPA. Although the 1500 entries in the directories do not help in understanding nor diffusing the innovations of software process technologies, they are highly demanded by the users. More than 70% of the e-mails received by GDPA are addressing suggestions and corrections w.r.t. the directories.

All innovations with respect to projects and products are mentioned in the home-page of GDPA, which has a link to an internal Projects/Products catalogue web-page. This page contains:

- some brief description of the innovation,
- a classification (to categorize the publications by subject),
- internal links inside GDPA, such as to the institutions directory, to e-mails sent to mailing-lists, internal studies etc.,
- an external link to an address where the user may find more detailed information, and
- in some cases an extended depiction of the industry best-practices.

**Table 5. Software Process Directories**

<table>
<thead>
<tr>
<th>Institutions</th>
<th>Institutions ordered alphabetically</th>
</tr>
</thead>
<tbody>
<tr>
<td>Educational</td>
<td>Educational institutions ordered by country</td>
</tr>
<tr>
<td>Persons</td>
<td>Persons ordered alphabetically</td>
</tr>
<tr>
<td>Projects/Products</td>
<td>Projects/Products ordered by acronym</td>
</tr>
<tr>
<td>Standards</td>
<td>Standardization institutions ordered by country</td>
</tr>
<tr>
<td>Who's Who</td>
<td>Who's who in process technologies</td>
</tr>
</tbody>
</table>

**f) Indexes**

The following indexes are indispensable mechanisms for a direct access to the web-pages and are quite simple to construct.
Table 6. Indexes

<table>
<thead>
<tr>
<th>Indexes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calendar</td>
<td>Calendar of events ordered by year</td>
</tr>
<tr>
<td>Figures</td>
<td>Index of figures ordered by subject</td>
</tr>
<tr>
<td>Graphs</td>
<td>daVinci [FW94] graphs ordered by activity (V-Model)</td>
</tr>
<tr>
<td>Magazines/Journals</td>
<td>Magazines/Journals in Computer Science - only by subscription</td>
</tr>
<tr>
<td>Magazines Kiosks</td>
<td>Magazines/Journals in Informatics - Kiosks</td>
</tr>
<tr>
<td>Memberships</td>
<td>Affiliated Institutions in Computer Science</td>
</tr>
<tr>
<td>Publishers</td>
<td>Publishers in Computer Science</td>
</tr>
<tr>
<td>Tables</td>
<td>Index of tables ordered by subject</td>
</tr>
<tr>
<td>Web-Site</td>
<td>Index of topics in GDPA</td>
</tr>
</tbody>
</table>

3.2 Learning Technology System Architecture (LTSA)

Computational learning environments are essential to facilitate effective comprehension and assimilation of software innovations. In recent years, learning environments are an issue fostered by many governmental agencies, academia and industries throughout Europe, America and Asia. The working group IEEE "1484 Learning Technology Standards Committee (LTSC)" is launching an architecture to become a standard for all learning technology systems [IEEE 1484].

Foreseeing an unproblematic interoperability with external learning environments, we adopted the architecture IEEE LTSA (Learning Technology System Architecture) although it is still a document proposal and not an official standard yet [Purp99c]. However, in order to employ the LTSA in the learning environment of GDPA, it was necessary to make 2 adjustments: we first added the process "Artifact" for the artifacts (hardware, software or documents) that are produced by carrying out a software process; secondly, we expanded the store "Knowledge Library" to support the "Experience Factory" proposed by Basili [Basi 93].

3.3 Meta-Level Links

In order to rapidly incorporate the frequent innovations on process technologies without major reorganizations in the GDPA structure, the links and objects are treated on an abstract level, called meta-level. The management of objects on the meta-level is not new [KP 81], [TS92] and has shown to be useful to support system evolution. Figure 1 is a simplified and illustrative example. For instance, instead of linking a specific tool (T1) with a specific development phase (P1), the process web-centre keeps the information of the link between the meta-model of T1 (MT1) with the meta-model of P1 (MP1). Thus, the link between T1 and P1 is automatically
deducted from the link between MT1 and MP1. This structure makes it possible that any other tool that “matches” MT1 is automatically linked to P1.

![Figure 1. Links on the meta-level](image)

However, not all links in the process web-centre are established on the meta-level. For example, the links between authors and their published papers, institutions and their commercial tools, among others, are assigned directly between the objects and not on the meta-level.

### 3.4 Meta-Processes for Software Process Standards

To cope with the user's demands for extending the process web-center to other process standards, a substantial restructuring of the database model was necessary. This model should support:
- the diversities among the process models of the standards,
- the formalization of the rules extracted from the standards,
- an experience library based on the experience factory [Basi93].

The GDP model was built upon the concept of *meta-processes* [ Purp00b ] and for its construction seven standards were analyzed: GD250 [GD250], ISO 12207 [ISO12207], ISO 9000-3 [ISO9001-3], IEEE 1220 [IEEE1220], IEEE 1074 [IEEE1074], ESA PSS-05 [PSS05] and AQAP-150 [AQAP150]. After many months of refinement, the meta-process resulted in a simple structure, which is organized in 5 layers. Each layer attends to one of the following questions:
- **What is to be done?** The binding set of activities and artifacts which are required and produced during the system development.
- **With what?** All the possible methods, techniques and tools which might be used to perform/execute an activity.
- **Is it done?** Appraisals based on rules, regulations, checklists and assessments lists which can be used either to determine compliance with a standard or to set out the actual state of a process.
- **How is it to be done?** All the practices, recommendations and improvements that can be undertaken based on the results of the previous question.
- **Why is it done?** The explanations and justifications for any of the former questions. These are also known as "rationales".
3.5 Ontology Library

An ontology is an explicit specification of a conceptualization [Grub93]. The ontology approach is an excellent method for comparing innovations on terminology. However, it is costly and is not easy to implement. So far, the ontology library implemented in GDPA is for internal use only.

In GDPA we developed a methodology for systematic comparison of concepts, definitions and taxonomies described in research papers, technical reports and standards. Four properties are defined [Free00]:

a) *synonymy* among different terms,
b) *overloading* meanings for the same term,
c) *inconsistency* in the description of a term, and
d) *self-reference tautology* when it is not possible to define the meaning of a term.

The first two properties elucidate the distinctions and similarities among terms. The last two properties detect potential errors in the definition of a term which should be meticulously reviewed.

Currently, this approach has been applied to the comparison of circa 400 terms described in 12 publications on process technologies. Unexpectedly, all the 12 publications analyzed in this study presented at least one semantic inconsistency or one tautology. This result is quite disturbing and it may be a sign that an ontology approach for comparative evaluation of process terminology and taxonomy of should be employed more intensively.

4. USERS PROFILES

Albeit we do not have any official survey about how the users apply the information provided in GDPA, it is possible to identify their profiles by the numerous e-mails with requests that were received. We can identify 4 major groups:

- **Users of the V-Model standards**: They are ca. 60% of the GDPA users. They are primarily concerned on: a) accessing the text of the V-Model standard in the web-based format, b) retrieving the e-mails sent to the V-Model mailing list from the text of the standard, c) use the learning environment for the V.-Model such as V-Model first steps, Frequent Asked Questions, etc.

- **Academic and industry researchers on process technologies**: Most of the e-mails received from this group can be resumed in two sort of requests: a) to update and to correct the information in GDPA about their research, b) to expand the entries in the glossary.
Producers of software process technologies: As GDPA is not bound to a software product or project, producers are quite cautious about demanding changes. However, they do send information about new product releases, new training courses, etc.

Users of software process technologies: Usually they inform about the problem they have for downloading the 13 MB GDPA ZIP file. In most cases this difficulty due to the internal net policies on the clients. Another frequent request is the update of the information about themselves in the GDPA directories.

5. CONCLUSIONS AND FUTURE WORK

This paper outlined the framework of the process web-center entitled GDPA for diffusing software process technologies. The relevant technical aspects and some of user's reactions during and after each technical aspect of its implementation were reported.

GDPA focuses on dissemination of the innovations in software process technologies in general without being bound to a software tool or project. In this sense it covers a wide range of innovations but it does not get into the details of putting the innovation into practice and ensuring that it is introduced in the best way. For this reason, GDPA is complementary to the individual initiatives, with have exhaustive information for the diffusion of a specific software tool, project or method. One could claim that there is a reciprocal dependence between those two modes of dissemination.

In particular, GDPA promotes the dissemination of the standard by enabling a expeditious on-line and off-line access to its target public, available for most of the World-Wide-Web (WWW) browsers. GDPA also supports process modelers to navigate throughout the standard according to their own strategy of learning by supplying a consistent, extensive interwoven hypertext documents, and facilitates the incorporation of the standard increments into the existing process by providing a flexible structure for process elements (elements such as activities, artifacts, agents, etc.)

During the last four years that we have been implementing, maintaining and improving the process web-center for process technologies we had more than 100 informal personal talks with persons who are currently and periodically using GDPA. In general they find it interesting and useful but they also expect many future enhancements.

Aside from the typical maintenance of GDPA and the incorporation of the innovations in process technologies, our next endeavor will be concentrated on the development of a virtual web-center where the owners
of the information themselves update the information. In the last months we have been working on the definition of the business processes for input, update, certification and delivery of the information that is going to be managed by the owners. This is not a trivial task and requires a lot of coordination and organization, particularly for the process of certification of the information.

Our second goal is to cover other process standards. This not only requires a huge programming effort but also involves a long course of meetings with the institutions launching the standards to arrive at a final agreement.

ACKNOWLEDGEMENTS

I would like to thank the numerous users of GDPA who sent their suggestions/claims. I am also much indebted to Prof. Bernd Krieg-Brückner from Universität Bremen for the fruitful and harmonious years of support.

REFERENCES

Carla Freericks (Blanck Purper)


Diffusion and Adoption of IT Products and Processes in a Danish Bank

Jan Pries-Heje and Susanne Tryde

1Department of Computer Information Systems, Georgia State University, 35 Broad Street, Atlanta, Georgia 30302, USA
Tel. +404 651-3822
Fax: +404 651-3842
Email: jpries@cis.gsu.edu

2Danske Bank, Holmens Kanal 2-12, 1092 København K., Denmark

Key words: Diffusion and adoption of IT, Implementation of IT, Organizational learning.

Abstract: This paper is about a successful diffusion and adoption effort launched in 1998 and continued for more than two years. The effort took place in the IT organization of a major Danish bank. As part of the effort a diffusion framework was developed to be used by IT projects. This framework was used in more than 30 projects to improve the changes of successful diffusion of the products being developed in the projects. And it worked. Interview data from May and June 2000, interviewing projects up to a year after they used the framework, shows a high level of satisfaction and successful product diffusion.

1. INTRODUCTION

Many resources are spent in IT companies developing IT products and processes that are never used as planned. There may be many reasons for this. Perhaps, management lost interest in the project. Perhaps, the project developing the new product failed to establish contact with future users and their real needs. Or perhaps, the project forgot to take advice from stakeholders in the project, who consequently blocked diffusion and adoption.
Problems connected with diffusion or implementation of software are not unknown. Rogers' (1995, pp. 5-6) book "Diffusion of Innovations" defines diffusion in the following way:

"Diffusion is the process by which an innovation is communicated through certain channels over time among the members of a social system ... diffusion is a special type of communication, in which the messages are about a new idea. The newness of the idea in the message content gives diffusion its special character. The newness means that some degree of uncertainty is involved in diffusion. Uncertainty is the degree to which a number of alternatives are perceived ... Diffusion is a kind of social change, defined as the process by which alteration occurs in the structure and function of a social system."

The central elements in diffusion are:
1. An innovation.
2. Communication through certain channels over time.
3. The diffusion and the change it leads to in the target group.
4. The uncertainty related to the many alternative ways in which the idea may be employed, the many ways in which communication may take place and the many ways in which changes may occur.

An innovation may be a product, a service or an idea that somebody perceives as new (Kotler, 1998, p. 439). Often, innovation is used as a term for a consumer's view on newly introduced products (see e.g. Mowen, 1995 or Rogers, 1979). According to Rogers, it is not important whether an innovation is new from an objective point of view as long as somebody perceives it as new.

The element of communication requires that developers and users arrive at a common understanding by creating and sharing information. Rogers (1995, p. 17) distinguishes between two kinds of communication channels; mass media and personal channels. By mass media, we understand TV, radio, newspapers, newsletters, etc., while personal communication may involve face-to-face messages or meetings. When it comes to convincing a target group, personal communication channels are generally most effective.

In connection with the communication and time dimensions, there is a range of phase models. Cooper & Zmud (1990) and Kwon & Zmud (1987) present the most prevalent phase model describing the diffusion of IT-products. It operates with six phases called: (1) Initiation. (2) Decision. (3) Adaptation. (4) Accept and adoption. (5) Routine procedures. (6) Infusion/penetration.

Related to the phase model a more normative model describing implementation strategies to be used in different situations has been developed by Ken Eason (1988). And following the same line of thought Targama (1978) and Bendix & Andersen (1995) have developed
Scandinavian models that can be used to choose an implementation strategy based on questions such as: Was the project idea initiated from the target group? To what extent is the successful implementation dependent on target group acceptance? And are the consequences for the target group primarily seen as positive or negative?

The target group involves a social system that in itself sets certain limits to how diffusion can take place. The social system consists of related persons, groups and departments, and both formal and informal structures have an effect. However, it is characteristic that people that is adopting an innovation needs to make the adoption decision themselves. Therefore, many researchers have focused on the individual’s resistance towards change (see e.g. Levine, 1997) or on the chaos perceived by the individual when a foreign element – innovation – enters his/her world (Weinberg, 1997).

All in all, there are a number of descriptive theories and models dealing with diffusion. However, to bring them into practical use is not in an easy task. For specific IT development tools Mathiassen and Sørensen (1997) have developed a framework. But for more general use very little is reported in the IT literature.

2. DANSKE BANK AND DANSKE DATA

Danske Bank is one of the largest banks in Scandinavia with more than 20,000 employees. Within Danske Bank there is a large IT department called Danske Data. From 1996 to 2000 the IT department worked as an independent software house. But for strategic reasons Danske Data was made an internal IT department again in 2000. In this paper we will use the term Danske Data to refer to the IT organization in the bank and Danske Bank to refer to the remainder of the organization.

Danske Data is physically situated at four different locations in Denmark, namely in Ejby west of Copenhagen, in Lyngby north of Copenhagen, in Brabrand close to Aarhus in Jutland, and in Nykøbing Falster 150 km south of Copenhagen. Approximately 900 employees are working in Danske Data.

The main products produced by Danske Data are applications for banking and insurance. Primarily applications are meant to run on a mainframe, but some applications are for Internet banking, client/server-environments, and PCs. Danske Data mainframe systems are up and running 24 hours a day, and every day 9 million transactions are carried out from 11,000 workstations. The developers within Danske Data typically have IT educations, at bachelor level, or come from a background in banking. Recently more and more employees with a master’s degree have been hired.
Danske Data was assessed in May 1997. From this assessment it became clear that many products and processes developed by Danske Data were not diffused and adopted as intended. In the assessment report it was said: "Why are so many procedures well described but not used?" (Delta, 1997). The assessment report suggested that the company should make a further analysis of diffusion and adoption, which was subsequently done. A task force of three people including the authors of this paper was set up.

3. RESEARCH METHOD

To cope with the problem identified - lack of diffusion and adoption - we have used action research. Bob Galliers (1992) describes action research as an approach that allows us to create new theoretical knowledge in addition to something that has practical value for the organization under research. The approach that we adopted in our action research is based on the five phases recommended by Susman & Evered (1995): (1) Specification of infrastructure in project. (2) Diagnosis of problem. (3) Planning of actions. (4) Implementing actions. (5) Evaluation of results. Repeat phase (2) to (5), if necessary.

The infrastructure of the project was set up in the fall of 1997. Insightful people from all parts of the organization being it project managers, line managers or people from the methodology department were asked and agreed to be part of a task force.

In two workshops in January and February 1998 the task force diagnosed the diffusion and adoption problem. In the first workshop the cause of the inadequate diffusion and adoption was narrowed down to a number of issues. In the second workshop, one month later, identified a number of solutions to the problems, and in April 1998 the company implemented organizational changes solving many of the structural problems. The remaining problem was to ensure diffusion and adoption of products and processes from IT projects. How does an individual project ensure that its new product will be used as expected when it has been completed? To answer this specific problem the infrastructure of the project was changed from a 12-people task force to a small group of three, each working half of their available time.

While studying the organization’s successes and failures in the first workshop we quickly realized that attempts to ensure diffusion by adding some additional activities at the end of the project often fail. It is necessary to start such attempts so early in the project that they will have an effect on the product itself. Therefore, we decided to make a framework to be used in a one-day workshop for projects focusing on diffusion and adoption right
after the requirements to a given product had been defined. At that specific point of time, the project group knows how the product is going to appear although no specific solutions have yet been prepared.

A major reason for choosing a one-day workshop was that one of the few successful diffusion efforts we identified in the organization was a so-called analysis workshop used to bring customers and developers together and agree on scope and requirements. Therefore projects within Danske Data were familiar with the workshop concept so when we planned our actions we decided to take advantage of this mechanism already in place.

Concurrently with studying cases from our own organization, we read the studies of others on diffusion and adoption of IT. The literature did not give us a holistic and action-oriented solution to diffusion and adoption, but different authors who had analyzed various elements of the issue (as mentioned in the introduction) did inspire us.

Based on the case study and details from the various articles, we developed our first version of our framework to be used in a workshop around New Year 1998-99. Our first test took place in three projects in February 1999. The test resulted in a number of adjustments. The same thing happened after the second iteration in two workshops. It was not until the third iteration that the framework and the workshop found a form that was acceptable for internal projects. A fourth and fifth iteration was carried out in the fall of 1999 and the spring of 2000 to adapt the framework to external projects – meaning projects involving customers from the bank or outside the Bank.

When using action research for the kind of study described here there is a number of things to be aware of. First of all you need to strive for rigorous and disciplined action research (Baskerville & Wood-Harper 1996). To ensure that all our data collection was done during and immediately after the workshops. We video-taped every workshop and used the tapes to make sure that we had captured every important piece of information. And we wrote a summary for each workshop that were send to the participants so they could acknowledge that we had captured all decisions and discussions correctly.

Furthermore when we adjusted our framework – as described above – we always tried things twice and looked for at least two consistent observations of non-satisfactory workshop results before we adjusted the framework. The interpretation of results was done in a group of two or three with at least one researcher and one practitioner taking part. The decision to change or adjust the framework was agreed by the whole group every time. Thus by doing the adjustments in such a rigorous way we hoped to avoid the danger of making to many “in-flight changes” that complicated or compromised our framework instead of improving it.
For evaluation we used a questionnaire that all workshop participants filled out after the workshop. This gave us valuable information that we used to re-design the framework, which we have called the first to sixth iteration above. However, this kind of evaluation did not give us insight into the longitudinal consequences. Was the framework really working? Did the products really diffuse better? These were some of the questions that we were asking.

To answer the questions we interviewed 12 participants from six projects that had completed a workshop up to year earlier and that in most cases were over their implementation. For the interviews we used a structured interview guide, we transcribed the interview, and we coded and analyzed the interviews. The interviews took place in May and June 2000, and after careful analysis we closed the project concluding that we had successfully contributed to solving the problem of lack of adoption and diffusion.

4. **THE FRAMEWORK**

The framework consisting of a one-day workshop is focusing both on the product that is to be diffused, and on the diffusion process in itself. The workshop is divided into three phases called analysis, design, and planning, each covering two stages.
The workshop starts with an introduction so the participants know what to expect at the end of the day. At the same time the workshop facilitators learn about the project and the product.

In the analysis phase the purpose is to get a common understanding of the size and effort the implementation process will require.

The design phase ensures that the project group focus on how the product has to be "packaged" to be seen as a whole product by the target user group. Furthermore the best implementation strategy is selected in the design phase.

In the planning phase, all the threads are gathered from the previous operations. An outline of the implementation plan is prepared comprising all the activities identified through the first three phases.

### 4.1 Analysis Phase

The analysis consists of two stages. First, we focus on a range of questions in order to characterize the project. The purpose is to create a common understanding of the product that is going to be the outcome of the project. Second, we focus on who is playing which roles in the project.

The first stage of the analysis is titled "Characterize implementation project" and includes four questions. Inspiration to the first part of the
analysis comes from Mathiassen and Sørensen (1997). Table 1 shows the questions in the left column and typical replies in the right column.

Table 1. Questions used to characterize project

<table>
<thead>
<tr>
<th>Questions</th>
<th>Typical replies</th>
</tr>
</thead>
</table>
| 1. Why should the product be diffused?        | • There is a need for ...  
• Company strategy requires ...  
• We must have this technology ... |
| 2. What is the purpose of the product?        | • To support administration  
• To support project management  
• To support IT development  
• To support IT operations |
| 3. Who must change their way of working when the product has been implemented? | • Software developers (host developers, client/server developers)  
• Experts in specific fields (database, economics, personnel)  
• Managers (project managers, line managers, senior managers) |
| 4. Where and in which sequences will the product be used? | • By a few individuals  
• By selected projects  
• By selected organizational units  
• By the whole organization |

The second stage of the analysis is titled “Determining who should play which parts” In this connection, we apply a role model inspired by Checkland & Scholes (1990) and Bendix & Andersen (1995).

The main idea of the role model is that five roles must be filled if implementation is going to succeed. If one of the roles has not been filled, implementation will fail. The five roles are:

1. The target group are the persons who are going to use the product
2. The owner/sponsor is responsible for initiating the implementation and scoping the direction. Towards the end of the implementation the owner /sponsor also is responsible for demanding the results coming out of the implementation.
3. The manager of implementation is the person doing the actual implementation work. Often this role is named the project manager.
4. The champion/ambassador are the persons who actually makes the people from the target group take the innovation into use.
5. “Other secondary stakeholders” consists of all other interested parties not taking any of the four primary roles.

In figure 2 the roles are placed around a big arrow that symbolizes implementation from the first tentative idea about a product at the extreme left to the product hits the target group at the right where it is diffused and put into use. Based on a detailed description of the roles, the project group has to fill the roles in relation to their specific product.

![Figure 2. The role model indicating central roles.](image)

### 4.2 Design Phase

The design phase consists of two stages. First, we try to define “the whole product”. Second, we examine how to organize the implementation of the product. The purpose is partly to clarify what kind of product we have to make in order to improve the chances of success in implementation. And partly to create a framework for an implementation plan that can be used in the subsequent phase. The idea behind “the whole product” is that the user of a new product normally expects to get more than a mere technical solution. Therefore, we are working with three product levels (Moore 1991):

1. The core product
2. The whole product
3. The expanded product

The core product is the developers' idea of what they have to prepare in order to give the customers what they promised. I.e. a developer who is going to develop a HomeBanking system may regard the computer program as the core product.

The whole product is the customer's idea of what he/she will get - i.e. the core product including related supplementary products to ensure that the product is easy to use. I.e. a customer who is buying a HomeBanking system will expect to get the disks with the program in addition to a quick guide, hot-line telephone support, and a tutorial.

The expanded product is not relevant until the whole product has been put into operation and the customer or the developer suggests supplementing the existing product with an additional service. For example, the customer might wish that the HomeBanking system also included a tax calculation feature.

In the workshop, we start by having a brainstorm indicating the ideas that the project group has about the “whole” product. Then, the project group is guided through a range of questions characterizing the target group and the core product. This procedure leads to a range of supplementary products that can support implementation of the core product. The end result is a clear definition of the core product and which supplementary products must be developed in order to meet the target group's expectations. The questions posed fall in three groups:

1. First we address the complexity and size of the project. The larger the project is, the less possible it becomes that the people implementing the project also can fulfill the role as ambassador or champion towards the target group. I.e. a project group of 8 people can never reach a target group of 8000 working in branches of the bank within a reasonable time frame. Complexity describes whether the product being developed has many interfaces to other systems, and whether it solves many problems for the target group.

2. Second we examine familiarity of the product that is to be developed in the project. How many functions are new and how many were in a former system? Will the organization or workflows be changed? And do the target group have experience with the technology applied.

3. Third we ask whether the product is aimed at a homogenous or a mixed target group?

For each of the eight combinations of the three dimensions we have identified a number of measures and actions that may be included in the implementation plan. When we relate the specific project to the three dimensions, we get a number of specific measures and actions that the project group can discuss whether to use in the concrete implementation.
The second stage of the design phase deals with how to organize implementation. For this purpose we have used a theory on implementation strategies developed by Ken Eason (1988). The theory comprises five different strategies from the most revolutionary strategy – where the users abandon their old system one day and adopt the new system on the following day – to the most evolutionary approach where the users over a period of months or years, little by little, start using more and more of the new system. Eason’s five strategies are called:
1. Big Bang
2. Parallel application
3. Phased introduction
4. Experimental diffusion
5. User-based experiment

The ‘Big Bang’ strategy implies that all users shift to the new system all at once. ‘Parallel application’ means that the old system continues to function in a certain period after the new system has been launched. ‘Phased introduction’ implies that the system is divided into phases in such a way that the entire target group gradually start applying more and more of the system. Or it may imply that the whole system is put into operation by part of the target group and subsequently the rest of the group will follow step by step, part by part. In ‘Experimental diffusion’, the system is tested by part of the target group. After a period of time, the experiences are compared with the test results, and it is decided to which extent the experiment should be diffused to others or made permanent. In the ‘User-based experiment’, the users themselves test the system in order to determine what they can achieve by using the system.

The relationship between the five strategies seen in relation to user participation and revolution-evolution is shown in figure 3.
The five implementation strategies are not mutually exclusive, but can be applied in combination.

In the workshop we analyze each of the five implementation strategies focusing on advantages and disadvantages. We emphasize that the demand for user participation in the development process is increasing from low participation in the ‘Big Bang’ approach to high participation in the ‘User-based experiment’. In terms of implementing the approaches, there is a movement from a very controlled approach in ‘Big Bang’ to a very experimental approach in ‘User-based experiment’.

However, ‘Experimental diffusion’ and ‘User-based experiment’ are not real options as it has already been decided to implement the system/product. The purpose of the workshop is to choose the most suitable way of implementation.

An outline of the implementation plan is made with phases and benchmarks depending on which strategy is chosen. For example, the ‘Big Bang’ approach has only one phase (before implementation) and one benchmark (implementation completed) as everything must be prepared for the big moment, the ‘Big Bang’.

In the end, the design phase will give the project group a clear idea of how implementation should be carried out in general.
4.3 Planning Phase

Similar to the two previous phases, this phase also has two stages: a risk analysis and an implementation planning stage. In the first stage of the planning phase, we perform a risk analysis in order to ensure that we haven’t overlooked any potential problems. Risk management deals with identifying and reacting to potential problems in time. Risk management often leads to proactive activities. In our risk analysis we focus on the following six issues:

1. Identify risks. Here we use a standard list of the ten highest risks in the organization based on the organization’s own experience.
2. Assess the probability for every risk on a scale from one to five.
3. Assess the consequence of every risk on a scale from one to five.
4. Make a list of priorities by multiplying probability with consequence.
5. Find activities – proactive or supportive – that relate to the three most important risks.
6. List the chosen activities as items that should be included in the implementation plan.

The second stage is to outline an implementation plan. During the workshop all activities mentioned by the project group are being written down on small notes. These notes – which are appropriate implementation activities – are now placed on a blackboard or a table where the benchmarks and the intermediate states are outlined. It may look as indicated in figure 4.

The last stage of the workshop provides the project group with a good outline of the implementation plan. It may be applied directly in connection with estimating the implementation phase and as input to the project plan and the final implementation plan of the project.

The intervening condition

The desired end condition

Figure 4. The implementation plan made with yellow stickers on a whiteboard

5. LESSONS LEARNED

While developing the approach using action research we have learned a number of lessons as explained below.
5.1 **First Lesson: Use detailed description of the target area**

The first stage of the approach has changed a number of times. We started out with a few who-what-where questions, but soon realized that we needed more knowledge about the project in question to be able to effectively facilitate the use of the approach. Therefore we now use the questions shown earlier in table 1. Furthermore we also ask what motivates or motivated the owner or sponsor to initiate the project, and we discuss the desired result of the project at length. All in all it typically takes 60-80 minutes to describe the target area in enough detail so we – as facilitators coming from outside the project – are able to identify ourselves with the project group and its context.

5.2 **Second Lesson: Detailed stakeholder analysis is not necessary – the role model is better**

When we first developed the approach we used a traditional stakeholder analysis to identify all stakeholders. In one case we identified more than 30 stakeholders. Unfortunately we could not use all that information for anything meaningful. Therefore we now use the role model (figure 3) as the only stakeholder analysis. We believe the role model covers the major roles in any organizational implementation – at least the ones we have met. And we have often found that one or even two of the roles were not filled with “actors” from the organization in the concrete project. So the use of the role model often leads to the identification of a major potential problem; the lack of an important role being played in the implementation enacting.

5.3 **Third Lesson: You can easily focus too much on implementation strategy**

Literature on diffusion and adoption of IT often considers implementation strategy at length. At the beginning of our action research we had the same strong focus on strategy. Based on Targama (1978) and Bendix & Andersen (1995) we went through a tedious process on determining whether to use an expert implementation strategy, a line management implementation strategy, a project change implementation strategy, or a process change implementation strategy. The decision on the strategy then led to a number of general recommendations.

Example: IF the dependency of active acceptance from the target group is low AND the consequences for the people that are influenced by the change
introduced are mainly negative, THEN use an expert based strategy, and be aware of the huge resistance to change this strategy may create. However after having tried out this way of looking at implementation strategy a number of times we realized that we were wasting our time. The advice that came out was simply too general and not useful to the projects. Instead we found Eason’s model (figure 3) that is now included in the approach. We use Eason’s model to select not a strategy but an approach. The gain from using Eason is that the project gets an idea of possible phases in the organizational implementation. Eason’s model has worked well in projects that have used the approach very early, which means just after the scooping of the project. On the other hand we have also found that projects that are well into the analysis or design of the technical solution have less to gain from the use of the model. Our recommendation is therefore to use this stage of the approach only for projects early in their life cycle.

5.4 Fourth Lesson: Determination of the whole product is essential

In the first version of our approach we had three dimensions or question types that we used to determine the whole product. However, we found that two of dimensions – familiarity of application and familiarity with technology - were overlapping, at least in the minds of the participants. We also found that size was very difficult to determine at an early stage of a project. Therefore we ended up with the three questions we now have.

Furthermore we have found that it is necessary to explain in detail what the whole product means. Often we have found that the more technically oriented people in software projects have needed a lot of explanation before they understand that a technical solution is not enough. In the concrete we have used the five adopter categories (Rogers 1995), from the "innovators" to the "laggards", and the gap - or "chasm" as Moore (1995, 1998) calls it – to explain why different parts of the target group in many projects needs more than just a core product.

5.5 Fifth Lesson: Implementation risks should be discussed before the planning

The content of the implementation phase in our approach has not changed since the first draft. But we have switched the two stages, discussing risks, and outlining the implementation plan. First we started out with the planning. Then we tried to identify and analyze risks. But we found that the participants in the workshops were quite unwilling to discuss risks
after having outlined a plan. Instead we tried doing the risk analysis before the planning, and it worked well. Thus we have learned that the openness and reflection needed to look into potential problems - risks - are better achieved before planning than after.

6. DISCUSSION AND CONCLUSION

The approach for organizational implementation that we have developed is to our knowledge quite unique. The uniqueness, however, does not lie in the six stages one by one but in the combination and cohesion of the stages and phases as we have described in this paper.

One can of course use each of the stages of the approach for different purposes. But if you are to use the whole approach we have found that a one-day 7-hour workshop with 6-8 participants representing important stakeholders in a software process improvement project is very fruitful. The evaluations we have gotten from approximately 30 projects have been that the time spent in a workshop using the approach was well spent - the benefits gained were much greater than the investment in time needed. They also have told us quite unanimously that they never would have been able to come up with such a detailed and targeted improvement plan as the one they ended up with at the end of the day.

To validate our approach we have also interviewed 12 workshop participants from six workshops including users from the target group of one project. The interviews took place 6-12 months after the workshop at a time where the implementation planned at the workshop had been carried out. Each interview lasted between 30 minutes and 1 hour. In the interviews we used the summary from the workshop to go over each stage of the workshop trying to evaluate the details, and we asked about the workshop as a whole trying to evaluate the workshop in general. In figure 5 you find citations from the workshop participants we interviewed 6-12 months after the workshop.
"I am pretty sure the workshop made a difference ... we have done things in this project that we would never have done without."

"The organizational implementation has succeeded beyond our expectations. We have had very few significant problems."

"If I should criticize something I believe we used a lot of time on a plan we weren't committed to use."

"Especially the activities in the proposed plan have been helpful."

"We came to focus on a number of activities that we hadn't thought of before the workshop."

"We got the summary from the workshop and it was strongly recommended that we should identify an owner/sponsor. But we never did."

Figure 5. Citations from evaluation interviews.

Our preliminary interpretation of the interview results is that in 5 out 6 projects the participants actually had carried out the activities included in the implementation plan. In general they were quite satisfied with the results. And they still evaluated the workshop as a very valuable and influential experience. But it is difficult to say anything definite before all implementations have been completed in well over a year (from fall 2000). The fact that many projects of their own accord have asked to participate in an implementation workshop also indicates that the concept works. Finally in October 1999 another maturity assessment – like the one in May 1997 that gave birth to our study – was carried out by independent assessors from outside Danske Data (Delta, 1999). This assessment found that diffusion and adoption was not one of the top five problems in the organization any more and emphasized the successful diffusion and adoption of the IT processes from three projects with whom we have had workshops.

In conclusion, we find that the framework described here appears to be a good bid for a solution to the problem with diffusion and adoption that many companies are facing. Furthermore a project manager can use our approach, either coming from the development side or from the user side, or it can be used by another kind of group responsible for software process improvement. The use of the approach can either rigorously follow the six stages in the recommended order (phases), or any of the six stages can be used separately.
REFERENCES


The Phenomenon of Diffusion

Red Herrings and Future Promise

Tor J. Larsen
Department of Technology Management, Norwegian School of Management, Elias Smiths road 15, P.O. Box 580, 1302 Sandvika, Norway
Phone: +47 67 55 71 78
Fax: +47 67 55 76 78
Tor.J.Larsen@bi.no
http://infomgt.bi.no/Larsen

Abstract: Diffusion is at the core of WG 8.6.2. Employing Rogers’ diffusion theory while in principle addressing other sorts of phenomena is an historic research problem. The applicability of Rogers’ theory is discussed using the perspectives of mechanic and organic organizational settings, reaching the conclusion that Rogers’ diffusion theory has only limited validity. Diffusion is defined generically as the spread of IS/IT among almost any organizational unit and its constituencies. No theory of diffusion has been developed as yet. Hence, diffusion, at best, might is an umbrella for strategy, innovation, network theory, social structural theory, and a host of other approaches to understanding change in organizational settings. Researchers need to clearly define their research scope and theory base, if we as a group are to contribute to the cumulative research, the principal prerequisite for ensuring value for practice. No doubt, in the near future, more IS/IT products, frameworks, and methods will be seen. Organizations must embark on multiple change processes that require other business, managerial, and methods approaches than are in place today while at the same time maintaining the use of well established and understood practices. These are issues that WG8.6 should address.

2 Excellent reviews of research issues in diffusion theory and comprehensive lists of published work are found in Bayer and Melone (1989), Fichman (2000); Moore and Benbasat (1991), and Conger (1995), and Wolfe (1994). Since this article does not include a bibliography, the reader is encouraged to review these sources.
1. INTRODUCTION

The International Federation of Information Processing (IFIP) Working Group 8.6 uses diffusion as an umbrella in defining its goals. Contributors to WG8.6 working conferences have presented multiple views on software diffusion and Rogers’ diffusion theory has been frequently used and cited (Prescott and Conger 1995; Rogers 1995). Substantial criticism has been raised with regard to research using diffusion theory a platform. It is argued that in Rogers’ diffusion theory, the research scope is too narrow, the actual observed richness of human behavior is not taken into account and the theory’s specification of diffusion decisions, when compared to real organizational diffusion management, render us with more questions asked than answered (Bayer and Malone 1989; Damsgaard and Lyztinen 1997; Wolfe 1994).

Of particular concern here is the difference in meaning between Rogers’ diffusion theory and the diffusion as the word is defined. According to the Concise Oxford Dictionary, diffusion is defined as “sending forth or shedding abroad.” Hence, related to our field, the semantic meaning of diffusion would include the transition among units of analysis of anything that is wholly or partially an information technology (IT) or information systems (IS) related innovation (Swanson 1994) among nations, organizations, groups, or individuals. It would also include agencies or virtual communities supporting IS/IT diffusion – run by the United Nations, governments, chartered diffusion organizations, consultants, software developers, IT departments, or local change champions. In short, diffusion includes almost anything and leaves little out.

In contrast, Rogers’ diffusion theory at its core is relatively precise. Diffusion is defined as “the process by which an innovation is communicated through certain channels over time among the members of a social system” (Rogers 1995, p. 5). Somebody develops an innovation. The innovation has (user) features that can be fairly exactly described and it is clearly separated from other physical objects or abstract phenomena. The innovation is in essence without modifications, spread to people who individually decide whether or not to adopt the innovation. Information about the innovation is initially spread through channels such as professional associations and journals. Next, news about the innovation is communicated through a social network where the first adopters are key. From these prerequisites follows the division of adopters into the categories of early adopters, early majority, late majority, and laggards but also the S-shaped growth curve. Rogers’ diffusion theory is heavily pro-innovation, otherwise defining some users as laggards would not be a key part of it.
An historic research problem has been using Rogers’ diffusion theory while in principle addressing all other sorts of issues that may or may not be diffusion related. This paper discusses the appropriateness of employing Rogers’ theory in organizational settings. Next, diffusion is put in context. The discussion section debates the future of diffusion research. The last section presents conclusions.

2. ROGERS’ THEORY IN ORGANIZATIONAL SETTINGS

Contrary to popular wisdom, a thorough review of the literature documents that studies within the IS/IT field keeping true to Rogers’ diffusion theory are preciously few. The classic example of research keeping true to Rogers is Brancheau and Wetherbe’s (1990) study of the diffusion of spreadsheet software in the heyday of end-user computing. They found that, at the time of the study, IT departments had not yet awoken to the challenge of end-user computing. The decision to use spreadsheet software was left to the discretion of the individual user, in agreement with the prerequisite in Rogers’ theory that the decision to adopt is decided at the individual level. A few users were found to be early adopters and some were found to be laggards. The finer shades of early and late majority could not be verified. Additionally, with regard to information source, early adopters were oriented toward professional associations and journals while later adopters depended upon early adopters for information about the new innovation. The cumulative adoption curve was found to be linear as well as sigmoidal.

The second research contribution keeping true to the theory is Moore and Benbasat (1991). Using a workstation as the innovation, they developed a scale for measuring the key concepts of relative advantage, compatibility, complexity, etc. The scales were confirmed using split sample research design.

These two studies strongly indicate that (at least aspects of) Rogers’ diffusion theory has relevance. However, problems exist. For example, today employees, as the general rule, cannot decide what IT tools they would like to use. The compelling reason is that the present IS/IT portfolio has become so large and complex that individual adjustments cannot be tolerated. The organization, through its IT strategy, infrastructure design, and IT department, decide the end-user IS/IT portfolio and features. Individual users do not decide their workstation capabilities but must make choices in accordance with established specifications. With regard to individual attitudes and perceptions, it may be argued that Moore and Benbasat’s scale has nothing to do with Rogers’ diffusion theory. Faced with any IS/IT
innovation (and perhaps other types of innovation as well, provided the name of the innovation is adjusted), people may exhibit attitudes and perceptions similar to those measured by Moore and Benbasat's instrument. Hence, the instrument might have validity in many settings but may not measure aspects of a Rogersian diffusion process. A good example of how the instrument may be used is found in Karahanna, Straub, and Chervany (1999). They studied differences in beliefs among users and non-users of Windows. Although the focus of the study was before and after adoption (of Windows), the authors did not frame their study within Rogers' diffusion theory.

The problems with other research claiming to study Rogers' diffusion theory are more profound. A common approach is to divide a sample into users and non-users or early adopters and late adopters (for example, Drury and Farhoomand 1996; Premkumar and Potter 1995). The delineation is used to study dependent constructs such as information satisfaction, relative advantage, product championship, or cost. These phenomena might be of interest and might have importance but there is little connection to the core constructs of Rogers' diffusion theory. In fact, differentiating between the two categories of use and non-use is convenient but cannot be said to represent theoretical reflection. More likely than not, using a totally expected and common phenomenon such as use and non-use, which is relatively easy to measure as the independent driver, will almost by default document differences within most dependent constructs. The end result may be confusion and theoretical nonsense.

The impression that researchers pay lip service to Rogers theory and published results is strengthened through the finding that work published after 1991 addressing the issues of relative advantage, complexity, etc. frequently does not use, discuss, nor in fact cite, Moore and Benbasat's article. It may be the case that Moore and Benbasat's scale does not have validity. If so, we would expect that authors, who for all practical purposes should be aware of the scale, would discuss why it could not be employed. The lack of sincere theory building is a fundamental problem that may be general in nature within our field and not necessarily a criticism directed at Rogers' diffusion theory in particular.

In summary, four main reasons why Rogers' theory does not have high explanatory power in organizational settings are presented (Fichman, 2000). First, organizational IS/IT innovations are more complex than Rogers' diffusion theory specifies. Second, the IS/IT innovation processes unfolding in the "adopting" organization are richer and more diverse than sigmoidal. Third, the division of (future) users into the categories of early adopters, early majority, late majority, and laggards is at best unproven but more likely an introduction of social complexity, yet simplicity, that implies
semantic meaning contrary to innovation process needs. Fourth, decisions are overwhelmingly organizational rather than individual.

Keeping with the concept of adoption, an illustration of the complexity in IS/IT innovation is taking a large standard software application into use (ERP, document handling, etc.). Because of organization specific needs, applications are commonly not installed as is but changes are made beyond parameter settings or module selection. These changes may apply to the entire standard application package but may also be business area specific (marketing, production, procurement, etc.). Since off-the-shelf software applications may not totally fit expressed needs, smaller ISs are usually created as an integral part of the process "to fill the gaps." Once installed, it is quite common to find that departments, groups, and individuals do not use application features that the formal software owners and planners deem critical. Also, departments, groups and individuals more often than not create IS and non-IS shadow solutions to bridge the gap between installed software application features and the way business is actually done locally. The formal software planners and owners are oftentimes not aware of the amount and importance of non-use and local innovations. Last but not least, shadow solutions may spread from the "creators" to others who see their business value. In addition to the degree of non-communication between developers and users that the phenomenon of shadow solutions documents, severe problems may occur with regard to quality and support. These themes are illustrated in Figure 1.
The data/information and processes contained within a standard application package (SAP)

**Question:** Other than parameter setting and module selection, is the SAP installed as is or does the host organization partially change it to meet specific needs?

If so, are these changes developed and implemented initially only or is change to the SAP an ongoing process?

**Infusion questions**
- Are the data/information and processes:
  - used as is?
  - partially used?
  - not used at all?

**Smaller ISs** consciously developed as extensions to the SAP (these are, probably, still organizational ISs)

**Question:** Were these IS developed initially only or is the development of smaller ISs an ongoing process?

Other local ISs developed because of partial use, non use or non-existent SAP or smaller IS support

**Question:** How are these developed, diffused, maintained and used?

**Research questions? Interaction?**

**Individual user needs**

**Research questions? Interaction?**

**Organizational needs**

---

*Figure 1. Process Elements in the Adoption, Development, and Use of Standardized Software Applications*

It can be concluded that most processes included in Figure 1 are not Rogersian diffusion processes. For example, the main development process of adjusting a standard software application to specific organizational needs has very much in common with the well established field of systems development. Within this area, research reports on strengths and weaknesses with regard to

- phase content and sequence
- the ability of developers, managers, and users to specify present and foresee future requirements
- user participation, comparative knowledge levels, and training needs
- implementation challenges
- the role of maintenance when a system is taken into use

are abundant. Hence, stating that diffusion does not apply as the grand theory for understanding the adoption of standard software packages should not be a surprise.

In fact, traditional systems development theories, such as the waterfall model, have been supplemented and partially replaced with evolutionary development and prototyping. With regard to the adoption of standard
software applications, Cooper and Zmud (1990 - referring to an unpublished manuscript by Zmud and Apple 1989) suggested the phases of initiation, adoption, adaptation, acceptance, routinization, and infusion. A generic model for IS/IT innovation consisting of the idea phase, the creation phase, and the usage phase has been suggested (Larsen 1998). Models describing the interactions between individuals and the organization when innovation is key include complex issues, phases, and reciprocal interactions (Glynn 1996). Compared with Rogers’ diffusion specifications, models of the nature discussed above more richly describe the source of an organizational innovation, the complex process in its development, and the processes that occur after an innovation is taken into use. Complex innovations also may develop in the patterns of evolution, dialectic, life cycle or teleology, or their combinations (Van de Ven and Poole 1995; Robey and Boudreau, 2000).

Diffusion theory might be of partial value. For example, it may be a starting point for understanding how shadow solutions spread through an informal social network. Additionally, the concepts of trialability, complexity, etc. may assist us in understanding user beliefs, attitudes, and behaviors with regard to a new system. Knowledge about user reactions might be used in planning and implementing actions aimed at securing strengths and minimizing risks. Such actions may include the redesign of particularly negatively viewed features or additional communication with users to explain the compelling reasons why (aspects of) an IS must be accepted.

The underlying theme in the discussion so far is that the employment of any particular theory must be firmly based on a succinct description of the research setting and objective. Finding when and where Rogers’ diffusion theory (or the concept of diffusion) is relevant may be a challenge in the traditional standard software application domain. Looking forward into the near future, we seem to approach a development situation characterized by a shift in focus from acquiring finished solutions to a focus on strategies, IT platforms, and tools that are change resilient. That is, rather than seeking to buy a standard software application from a particular vendor, organizations will increasingly look for IT products that allow the maximum degree of freedom for making changes with as little effort as possible. We see this change in attitude in the area of E-business, but the prerequisite that software solutions are change robust will also increasingly apply to almost every aspect of hardware and software. The differences between traditional issues within systems development using components created outside the organization and traditional approaches and future innovation oriented practices are illustrated in Figure 2.
Note: Issues using or that may use Rogers’ diffusion theory (elements) are printed in bold.

Figure 2. Main Issues Within Traditional Software Adoption and Future Innovation Oriented Practices

It is evident that using Rogers’ diffusion theory in the new setting requires even more precise argumentation than is the case in traditional settings. Also, the concept of diffusion becomes exceedingly problematic. We may study the diffusion of software applications and tools. However, each component is quite likely only a smaller piece of the organizational IS/IT infrastructure and portfolio. Organizations will be concerned with creating genuine business-value-added use of IS/IT. Hence the overall objective will be combining off-the-shelf components in ways that are tailored to the organization, otherwise organizations cannot differentiate themselves in the marketplace but converge toward look-alikes. The emerging business benefits are created through innovations large and small. The focus will be on knowledge and information and not on data processing. Diffusion theory may be applicable but it will hardly have a dominant role on the center stage.

3. THE CONCEPT OF DIFFUSION IN CONTEXT

The previous section argued that the employment of diffusion theory requires careful thinking about research setting and concept applicability. Diffusion as an umbrella term may not be a rich and timely concept for understanding how business value is created when the need for innovation
The Phenomenon of Diffusion

Dominates. Continuous arguments about value in discussing diffusion may resemble the proverb of “the futility in flogging a dead horse to make it move.”

Building on the observation that many institutions and actors may have a role in diffusion (Damsgaard and Lyytinen 1997), Fichman (2000) argued that diffusion studies may be categorized along the two axes of primary (community level) vs. secondary (adopter level) and perceived vs. objective (characteristics). The principle forwarded here is “reactions to an innovation or clusters of innovations.” The author argues that understanding these reactions may influence the formal adoption decision process positively. Since one might infer that the IS/IT innovation is known, the principal mode of this type of research is reactionary. The contrasting view would be that structures (for example, attitudes and beliefs) are not located in organizations or in (IS/IT) technology, but are enacted by users (Orlikowski 2000). Saying that a person is in the driver’s seat indicates that reactions to existing innovations are only part of an innovation process. It is obviously true that IS/IT innovations exist in the market place but they cannot be incorporated into an organizational innovation process unless a person or group decide that a particular IS/IT should be adopted and taken into use (Larsen 1998).

A qualified answer with regard to the applicability of the concept of diffusion requires putting it into a wider context than already presented. Van de Ven and Astley (1981) forwarded the notion that organizational theories might be viewed in the two dimensions of within an organization versus within the wider environment and structural versus process approaches to theories and issues. A framework adapted for IS/IT innovation is presented in Figure 3.
It would seem appropriate to observe that the concept of diffusion carries relevance within the structural approaches. Governments create programs to encourage IS/IT industry growth and increased use in the society as a whole (European Commission 1996). Universities and industry create special technology institutions mandated to educate prospective users and push novel technologies into social as well as industrial organizations (Charlton et al. 1998; Swan, Newell, and Robertson 1998). Within the organization, managers and IS/IT experts unite forces to develop and implement, for example, E-business applications. Organizations use consultants extensively in the hope of learning how to harvest benefit from the latest IS/IT developments.

The troublesome aspect is the semantics of diffusion. It implies that somebody has something that others would benefit from, if only the recipient is educated and convinced about the positive aspects that would be gained. Hence, diffusion channels becomes a vital part of the diffusion “Weltanschauung” (Checkland and Scholes 1990). To a large degree, the diffusion owner decides the distribution channel mix and activity level. Ultimately, diffusion implies an elitist view with regard to innovation. Those who have transmit to those who have not. Those who know decide over
those who do not know. Those who have and know are in the driver’s seat while those who have not and those who do not know are passengers.

Abiding by “diffusion thinking” might be a very efficient implementation approach provided the knowledge “diffusion recipients” possess is not required, making the core innovation right. Seemingly, this is increasingly not the case (Nonaka 1995). Top managers and IT/IS experts do not have sufficient insight into strategic IS/IT needs and requirements. To make sure that the business-IS/IT needs are satisfactorily explored and that solutions serve opportunities, many interest groups must be included in the innovation processes. This is the core concept of the process approach. It implies appropriate collaboration and the acumen to include general as well as in-depth expertise from idea birth, throughout the development process, throughout a system’s usage phase, including system termination (Larsen 1998).

Obviously, the process approach does not imply a near global innovation happening and including everyone, or that we may be victims to benign human thinking; that is, true democracy means that everyone by default contributes equally. It is exceedingly difficult to believe that routine jobs will disappear, that everyone is genuinely interested in innovation and taking responsibility, that everyone from birth is equally equipped, or that the global society and its industries in the foreseeable future can afford “process” as its sovereign political foundation. Conversely, changes will occur, process is a fundamental need to ensure survival, customer value must be created, jobs as we know it will evolve, and firms must make profits (Rifkin 1995).

Hence, the process approach includes the employment of structural solutions where efficiency and certainty (with regard to, for example, production process, products, or customer segments) dominate. Wherever structural processes represent the best approach, the concept of diffusion may play a role. The prerequisite forwarded here is that, to the best of actors’ ability, structure is decided as fitting the needs.

The division between organizing for flexibility and efficiency simultaneously is nothing new. The division has been extensively described with regard to organizational principles applied to research and development units versus the rest of the organization (Burns and Stalker 1994). Lawrence and Lorsch (1967) argued that production departments needed structure while the integration between production and marketing needed flexible solutions. Today, professional materials convey the impression that it is taken for granted that flexibility is needed in many business situations – although it cannot be said that theories that discuss when and where process or structure applies are in abundance.
Van de Ven and Astley point to this challenge by stating that although each of the quadrants (in Figure 3) are of importance and interest, the most fruitful research would be in investigating the interactions and tensions among them. The proposition is more relevant than ever.

4. DISCUSSION: THE FUTURE OF DIFFUSION

It may well be that our preoccupation with diffusion stems from the fact that the IS/IT industry relentlessly puts new concepts and products into the market place, as illustrated in Figure 4.

![Diagram](image)

*Figure 4. The Introduction of IS/IT Concepts Over Time*

The present surge of integrating IS/IT across place and time may, in historic perspective, be viewed as the most active period of our field ever. The emerging technologies may soon be taken into common use. However, the solutions we see emerging today may not fully meet customer expectations. For example, the computing processing time is too long. The reason why is easy to understand. New web services require that not only data but also relatively extensive amounts of code (Java, Applets, Agents, etc.) are transmitted back and forth to make transactions work. Although
moving process bars are made part of the user interface to stall user impatience, we know for a fact that wide band transmission will be introduced to minimize waiting time but also to make true two way multimedia services happen. Another characteristic is that most web services offered today mirror yesterday’s industry structure. Airlines, banks, insurance companies, or car dealers offer solutions that mirror each firm’s business needs. For a customer, finding the cheapest and most convenient travel route, the best bank partner, the best fitting new car or car insurance using the web is a time consuming and frustrating process.

This is why, as Figure 4 suggests, a new main IS/IT development stage is under way: the “natural human behavior orientation.” This stage has information needs, as the customer wants it as its focus. It is anticipated that the customer will ask focus on issues such as:

- I want my banking, finance, and insurance taken care of as a whole. Who are the providers? What will it cost? What additional services are offered?
- For our summer vacation, my family, consisting of two adults and two children, would like to go to Maliorca. Are there hotels or apartments available? We would like to take a variety of excursions. What kinds of excursions are available? We would like a rental car that meets certain specifications of size and type. Can the requirements be met? Who are the providers? What is the cost? What additional services are offered?
- Here are the specifications for a new car; up to two years old would be acceptable. What can car dealers offer? What is the financial deal? What do customers having this car think? What are the “customer watcher” remarks and findings?
- I want my workstation to be organized as a dashboard.
- As a researcher, I want easy access to publications, methods overviews, analysis tools, and research instrument creation

Most probably, the customer would like to service-browse sitting on the sofa using the television. The first wave of services of this nature are already offered. The argument made here is that this will be the rule and not the exception. Last but not least, the interaction will be genuinely multi-media – video presentations of place, product, and product use will be included.

We may safely expect that more IS/IT products, methods, and knowledge will be diffused than ever. There will be experimentation, success, and failure. The phenomenon of IS/IT diffusion has a rosy future. Yet, the problem is that diffusion is generic. A word that may be used for everything is not a good starting point for deep understanding. Hence, focus must be articulated: diffusing new products among industries is most likely best viewed as industrial marketing, making individual customers use a service
has very much to do with consumer marketing, and the development of new IS/IT solutions implies strategy and systems development.

It is, therefore, necessary that a group using IS/IT diffusion as its umbrella term identifies special interest groups for concrete aspects of diffusion deemed key. It may well be that subject focus and neutral information would have helped us understand the recent phenomenon of e-commerce and dot-com more objectively. If so, maybe the present situation of dot-com failure being the rule rather than being the exception could have been predicted and made less frequent. It is interesting to observe that Gartner group has introduced “hype” as the explanation for the usage development pattern for new IS/IT phenomena. Almost without exception, new IS/IT gadgets are over-valued and over-used. Because of severe failures, a painful usage reduction period must be experienced before a much smaller, stable trajectory with regard to utility emerges.

As a result, the IS/IT field is over time constantly shifted from being in the palace with the King to being in the hard place. The advent of the E(lectronic-business) is giving way to the M(obile-business). Everybody wants a share in it, but the E and the M might very soon be viewed as part of everyday life and not generate much interest (Earl 2000). This is an area where a group such as WG8.6 could make a contribution, that is, being the conveyor of sober views and research with regard to the societal implications, business potential, stakeholder awareness, marketing, and solution development requirements of new IS/IT products.

REFERENCES


The Phenomenon of Diffusion


A Perspective of the Innovation-Diffusion Process from the Self-Organizing System

Toshio Mitsufuji
Department of Info-Media, Slebold University of Nagasaki, 822 Yoshimuta-Go, Nagayo-Cho, Nagasaki-Ken, Japan 851-5086
+81 (95) 813-5086
mitsufuji@sun.ac.jp

Abstract: The purpose of this paper is to examine the conventional diffusion theory and the critics against it, and to propose a conceptual model for the innovation-diffusion process into the social system. Applying the evolutionary perspective focusing on the self-organizing system, the conceptual model for the innovation-diffusion process is built. Based on the investigation, this paper argues the necessity of reconstructing the diffusion-innovation theory in terms of the self-organizing system. Especially, this paper proposes the reconsideration of the critical mass, innovator/imitator dichotomy, and the meanings of such parameters as internal and external influence factors as well as the number of the potential adopters.

1. INTRODUCTION

This study aims at rethinking the innovation-diffusion theory and trying to propose a conceptual model for interpreting the diffusion process in terms of the self-organizing system. The conventional innovation-diffusion theory was first integrated and established around 1960s. Since then, many diffusion studies have been performed in various fields based on it. However, partly due to the recent trends that the development of innovations is expanding at a great rate, there have appeared many phenomena that the conventional diffusion theory cannot explain appropriately. As a result, the conventional theory has been sometimes criticized from several academic fields.
This paper will examine the conventional diffusion theory and the critics against it, and will propose a conceptual model for the innovation-diffusion process into the social system, assuming that it is a dynamic and self-organizing system. Based on the analysis, this paper will argue the necessity of reconstructing the diffusion-innovation theory in terms of the self-organizing system.

2. THE CONVENTIONAL INNOVATION-DIFFUSION THEORY

Rogers first published "Diffusion of Innovations" in 1962, which was one of the most seminal works in the innovation-diffusion research area. According to him (1995), "diffusion is the process by which an innovation is communicated through certain channels over time among the members of a social system" (p.5). Bass (1969) analyzes statistically the innovation-diffusion process of durable goods based on the Rogers' diffusion theory in the field of marketing, dividing the diffusion process into external and internal influence factors. (See also Mahajan and Peterson, 1985; Mahajan, Muller and Bass, 1990). Mansfield (1961), an economist, introduces an imitation model to analyze the innovation-diffusion process of industrial machines among industries.

In these diffusion thoughts that first appeared several decades ago, the development process of an innovation is usually divided into such phases as needs/problems, research, development, commercialization, diffusion and adoption, and consequences, occurring one by one. In addition, it is also assumed that an innovation itself does not change significantly to influence its innovation-diffusion process while it diffuses into the social system. In other words, the innovation process is implicitly considered to be independent from the diffusion process.

When these thoughts are together called a conventional innovation-diffusion theory or model, it can be pointed out that it regards technological development as a linear process. Referring to the overview of innovation in terms of R & D activities, Kline and Rosenberg (1986) argue that "models that depict innovation as a smooth, well-behaved linear process badly misspecify the nature and direction of the causal factors at work." Furthermore, they propose the chain-linked model, in which research, invention, innovation, and production are linked mutually and various feedback loops are formed among these phases. Finally, they assert persuasively that "innovation is inherently uncertain, somewhat disorderly, made up of some of the most complex systems known, and subject to
changes of many sorts at many different places within the innovation organization."

Williams and Edge (1996) depict the recent development of the "social shaping of the technology." In their article, they contend that the linear model "describes technologies as 'applied science', emerging through a sequential flow from basic science, through applied R&D to commercial production and use/consumption: it conceives the cycle of invention-innovation-diffusion as separate 'stages' in an essentially linear process" (p. 847). Latour (1987), a social-shaping theorist, describes the process toward the establishment of the scientific theory and/or the technological artifacts as being quite complex and uncertain. He explains a series of innovation and/or invention processes from the beginning to the establishment in the social system, referring to the examples such as the development of work stations, the biochemical discovery and so on. Based on these studies, he argues the diffusion model as being "[s]pewed out by a few centres and laboratories, new things and beliefs are emerging, free floating through minds and hands, populating the world with replicas of themselves" (p.133). In addition, instead of the diffusion model, he proposes the translation model in which the interpretations are "given by the fact-builders of their interests and that of the people they enroll" (p.108).

Utterback (1994) argues the dynamics of innovation development, referring to the developmental phases that are classified into fluid, transitional, and specific ones. Uncertainty conditions are predominant in the fluid phase, while "market acceptance of a product innovation and the emergence of a dominant design are its hallmark" (p.96) in the transitional phase. Further, in the specific phase, the technology becomes mature and various incremental improvements take place. Tushman and Rosenkopf (1992) propose a model of technical change that is driven by sociocultural processes of variation, selection, and retention, referring to the evolutionary theory, based on empirical surveys such as cement manufacturers, the diagnostic imaging industry, and so forth. In addition, they analyze the technological change in terms of the evolution theory, and propose a cyclical model, which has four components, that is, technological discontinuities, eras of ferment, dominant designs, and eras of incremental change. Because this is a cyclical model, technological discontinuities begin after the eras of incremental change. Technological discontinuities, a dominant design, and an era of incremental change are introduced on the analogy of the variation, selection and retention within the process of the biological evolution theory, respectively. According to them, Technological substitution, design competition, and community driven technical change occur in the eras of ferment, which arise after the technological discontinuities, until a dominant design appears as the result of the selection process. (See also, for example,
In economics, evolutionary economics has attracted many scholars' attention (Dosi, 1991) and several theories are proposed such as the technological trajectory (Dosi, 1982), the path dependency (David, 1985), the lock-in mechanism (Arthur, 1988) and so forth. Metcalfe (1995) argues that the economics of technology can be outlined from "two quite different perspectives, the equilibrium view and the evolutionary view" (p. 410). The characteristics of the latter are "not equilibrium and state but process and change" (p. 414). Neo classical economic theory, which treats innovations and technologies as exogenous variables from the equilibrium perspective, has been criticized for a long time.

Concerning the relationships between evolution and self-organization, Kauffman (1996), a biological evolution scholar of Santa Fe Institute, describes the evolutionary process with confidence that "much of the order in organisms may not be the result of selection at all, but of the spontaneous order of self-organized system" (p. 25). In addition, he mentions that the evolution process is composed of both selection and self-organization. Bak and Chen (1991), another scholars of Santa Fe Institute, assert that many composite systems have the characteristics of the self-organized criticality and they evolve naturally toward the critical state, in which even a very tiny event will cause the chain reactions that can affect infinite elements in the system. They still argue that some specific laws exist in the self-organizing system. In addition, Bak (1996) describes that "self-organized critical systems evolve to the complex critical state without interference from any outside agent" (p. 31).

The concept of the self-organizing system seems to be not necessarily emphasized within the framework of the innovation-diffusion theory, at least not fully developed within it. On the other hand, several studies have been conducted to analyze the technological change and innovation process, based on the concept of the self-organizing system as mentioned above. In this paper, I will try to reconstruct the innovation-diffusion process and to make some propositions in order to interpret the innovation-diffusion theory in terms of the self-organizing system.

3. MODELING THE INNOVATION-DIFFUSION PROCESS

Conventional innovation theories assume that innovations do not vary significantly while they diffuse into a social system. In actuality, innovations alter their aspects considerably when they diffuse into the social system.
Various interactions that cannot be ignored exist between innovations and the social system. On the other hand, the recent development of the complexity systems theory shows that many dynamic social systems in which complex interactions exist among elements follow the theory of the self-organizing system. This paper assumes that considerable interactions exist when innovations diffuse into the social system. On this occasion, the self-organizing systems are defined as the "systems, that even when they start from an almost homogeneous or almost random state, spontaneously form large-scale patterns," following Krugman (1996, p. 3).

3.1 Some premises for actors and factors affecting an innovation-diffusion process

If the innovation-diffusion process shows the self-organizing pattern, there must be specific laws, as is talked by Bak and Chen (1991). In general, the social system must have different dynamic characteristics from those existing in the evolutionary process of creatures or physical/chemical phenomena. An innovation-diffusion process is a dynamic one, in which human beings take part, so that there are various states peculiar to human beings and social systems they construct. What is characteristic about the technological innovations is that many of them are embodied in the artifacts. This paper will examine the innovation-diffusion process of these artifacts put on the public consumers' market. Assuming that there are considerable interactions between an innovation and its diffusion process into a social system, this paper will make following premises.

1. Potential adopters of an innovation make their decisions whether they will implement it or not, considering both internal and external influence factors.

2. Innovation providers not only attempt to appeal to potential users in order to make the innovation diffuse into the social system, but also make an effort to enhance its efficiency and effect.

3. Various other actors and social groups take part in the innovation-diffusion process in order to promote or block the diffusion of an innovation, forming its external environment.

4. An artifact that embodies the innovation is the focal point among users, providers, and other relevant actors and social groups. Once an artifact is put on the market, it is by no means easy to change the design, so that it constitutes an idiosyncratic point within the social system and structurizes the social system.
3.2 An innovation-diffusion model taking into account the self-organizing system

Based on the premises mentioned above, I will propose an innovation-diffusion model in terms of an evolutionary perspective taking into account the self-organization as well as the selection process. Tushman and Rosenkopf (1992) show the cyclical model of technology on the analogy of evolution theory: variation, selection, and retention. Besides, Utterback (1996) argues the appearance of the dominant design after the fluid phase. They build their cyclical models of technological change based on various empirical studies, and I will follow their classification on the innovation-diffusion process. Incidentally, while they focus on the technological cycle, I will pay specific attention to the process of technological diffusion so that I will modify their classification some.

As the development of technologies speeds up, various innovations are continuously put on the market. Thus, for example, Basalla (1988) urges that technological development be not only continuous but also selective, forming an "evolutionary" process, the idea of which may strengthen the standpoint of a cyclical model of technology. However, it must be of important meaning to interpret the technological development as the diffusion process, because by doing so we would be able to more fully understand the relationship between the innovation and the social system in which it diffuses. In actuality, many innovations, which ordinary people have neither imagined nor believed in beforehand, have revealed themselves repeatedly and attracted a great deal of attention. Moreover, I will stress the importance of the self-organization as well as the selection process. Accordingly, referring to the cyclical model of technology, I will develop an innovation-diffusion model in terms of the evolutionary perspective taking into account the self-organizing system as follows.

1. Appearance of an innovation that overthrows the existing technological paradigm.
2. Trial and error by engineers and related organizations' staff as well as the participation of various social groups.
3. Appearance of the dominant design and passing through the irreversible phase.
4. Stabilization and fixation of the innovation in the social system.
5. Full-scale penetration of the innovation into the social system.

Next, I will explain each phase, sometimes referring to the innovation-diffusion process of Japanese word processors into the Japanese society around 1980s. I have chosen the innovation-diffusion process of Japanese word processors as the object of this analysis, because of several reasons. First of all, the implementation of Japanese word processors or, more
generally, their functions into the Japanese society has a tremendous impact on it. Before the appearance of Japanese word processors, ordinary people thought that the usage of such devices as writing a letter sitting in front of a desktop machine was a sheer fantasy. Second, this innovation-diffusion process occurred in the rather closed social system, say, Japanese society, so that if anything it was convenient to follow and examine the process. Third, because they are consumer goods as well as office goods, we can expect that there have been various interactions between the innovation and the social system, and they have affected each other enormously.

3.3 The appearance of an innovation that overthrows the existing technological paradigm

In the first stage, an innovation appears that overthrows the existing technological paradigm in a social system. The word "paradigm" is defined for the time being as an exemplar serving as a technological model. The environment surrounding the existing paradigm has changed without being noticed so that the social system has exceeded the self-organized criticality (Bak and Chen, 1991) to yield an innovation.

An innovation is brought up in the social system, denying the existing technologies and/or ideas that have been regarded as natural and common. An innovation arises from those human activities as foresight, insight, intuition and so on, induced by the environment change. For example, Japanese word processors were first put on the Japanese market in 1978. Until then, even the scholars of relevant areas such as information engineering never believed that Japanese ordinary people could make use of such equipment as to write Japanese characters. Because Japanese characters are composed of ideograms as well as phonograms so that they have many characters, some scholars had thought that it was almost impossible to type Japanese languages using such keyboard as QWERTY-type. The appearance of the Japanese word processors surprised Japanese people as well as professionals to a great extent.

3.4 Trial and error by engineers and related organizations' staff as well as the participation of various social groups

After the appearance of an innovation, engineers or relevant professionals witnessing the innovation conduct various kind of trial and error to compete with each other. In addition to these professionals, relevant social groups such as frequent users, consumers' organizations, professional users' groups and so forth will join the innovation-diffusion process so as to interpret it.
Pinch and Bijker (1987) call these behaviors interpretative flexibility, after examining the history of the bicycle. This may correspond with the fact that the explosion of the biological creativity took place in the Cambrian, the phenomenon of which Kauffman (1995, p.199) depicts using the analogy of the self-organization. Sometimes firms operate antennae shops to explore the consumers' trends. On the other hand, users and/or consumers in the social system communicate with each other, trying to obtain appropriate information in order to decide whether they should adopt or not. In this stage, as no one knows the optimum design, various artifacts (= products) are put on the market (Utterback, 1996). All sorts of efforts should be done so as to enroll members of the social system including engineers, relevant organizations, and various social groups and to receive much attention from members of the social system (Latour, 1987).

Conventional diffusion theories usually do not consider the modifications of the innovation while it diffuses, but take it for granted that the innovation is invariable. In actuality, the environment surrounding an artifact, in which an innovation is embodied, transforms all the time based on the technological evolutions, cultural change, alteration of the institutions, and the change of other various factors.

Bass (1969) proposes the diffusion model as follows:

$$\frac{dx}{dt} = ax(1-x/K) + bx(K-x),$$

where,
- $x$: cumulative number of adopters at time $t$,
- $K$: total number of potential adopters in the social system,
- $a$: internal influence factors, and,
- $b$: external influence factors.

According to Mahajan and Peterson (1985, pp. 15-17), internal influence factors represent horizontal channels of communication, decentralized channels of communication, unstructured, informal channels of communication, while external influence factors do vertical channels of communication, centralized channels of communication, structured channels of communication, and formal channels of communication. These parameters of $a$, $b$, and $K$ must change as the innovation diffuses into the social system and interact with it. From this perspective, it may be more convenient to express the Bass model as follows, adding the discontinuance factor,

$$x_{n+1} = x_n + ax_n(1-x_n/K) + bx_n(K-x_n) + cx_n,$$
where,  
\( x_n \): cumulative number of adopters at discrete time \( n \), and,  
c: discontinuance ratio of the innovation.

Of course, \( a \), \( b \), \( c \), and \( K \) are the functions of time \( n \). For example, the potential adopters of Japanese word processors were at first, say, journalists, engineers, and specialists in offices who make out contracts. However, as Japanese word processors diffuse into the Japanese society, the potential population of adopters grows, including professionals other than engineers, ordinary office workers and so forth. In addition, while Japanese word processors diffused into the Japanese society, their features changed significantly, without doubt, influencing the parameters \( a \), \( b \), and \( c \). At least in the case of the Japanese word processors, the more an innovation diffused into the social system, the more their features fitted the requirement of the potential users, the number of whom expanded as the time went by.

From a standpoint of the strategic management, an innovation provider can control these parameters while enrolling potential adopters, so as to succeed in the market. From a standpoint of the technology policy, policy makers can control these parameters in order to make an innovation diffuse into the social system. However, it is not necessarily desirable to implement an innovation immediately, but "in many cases delay is economically desirable" (Metcalfe, 1995, p.483). Exactly in this point, the strategic management and technology policy must be implemented appropriately in accordance with a lapse of the innovation-diffusion process.

3.5 Appearance of the dominant design and passing through the irreversible phase

Under passing through the turmoil period, the social system in which an innovation has appeared becomes changing its structure. If an innovation acquires the support of potential adopters while changing its features, successfully diffusing into the social system, the innovation will reach an irreversible phase with the symmetry breaking \(^{(1)}\). An innovation is embodied by the dominant design, which is, according to Utterback (1996), "the one that wins the allegiance of the marketplace, the one that competitors and innovators must adhere to if they hope to command significant market following." Sometimes an innovation may have changed its features completely. By the time a dominant design appears, various relevant technologies to support an innovation are improved, and reverse salients (Hughes, 1983) are dissolved. For example, video cassette recorders (VCRs), which were put on the Japanese market in 1975, did not diffuse much for a time being. At last, after the establishment of the distribution system for the
recorded tapes of VCRs as well as the appearance of the dominant design, they began diffusing.

An innovation has altered its features, as the social system in which the innovation was thrown has changed its structure. An innovation interacts with the social system, while various artifacts that embody the innovation are selected in the process of this interaction. For example, after Japanese word processors were put on the Japanese market first in 1978, many variants appeared in the social system one after another. Finally a dominant design appeared in the process of the selection around 1985, which introduced in most cases the QWERTY-type keyboard, even though there remained a few variants concerning the layout of the keyboard. In this period, relevant engineers improved display devices, memory devices, printers, and so forth vigorously, and reverse salients were gradually dissolved. Although the price of the word processors was 6,300,000 yen (about 50,000 dollars) per copy at first, it dropped dramatically until 1985, stabilizing at about 100,000 yen (about 700 dollars) per copy since then. The social groups using Japanese word processors expanded from various specialist groups to the more generic ones such as ordinary office workers to ordinary people in the society. It can be said that the innovation is shaped not only by engineers but also by various social groups. Finally, users could make use of Japanese word processors with a certain amount of satisfaction around 1985. It can safely be said that the social system altered its structure.

The appearance of the dominant design may roughly correspond with the critical mass point in the diffusion theory, because the innovation-diffusion process takes off on a large scale after that. However, it differs from the critical mass point by definition, because it does not matter how many people in the social system adopt the innovation but what is essential is that the irreversible or self-sustaining state with symmetry breaking takes place in the social system. Quasi S-shaped diffusion curve may begin from this point, even though it lacks left and lower part of the "S". 

3.6 Stabilization, fixation and full-scale penetration of the innovation into the social system

Once an innovation has surpassed the irreversible phase, obtaining the dominant design, it becomes embedded into the social system. Some innovations diffuse explosively when the dominant designs appear. An innovation achieves the stable position in the social system. The potential adopters do not care very much whether other members of the social system have adopted it or not, but pay more attentions to the efficiency and the cost-effectiveness of the innovation. The population of the potential adopters increases from rather the specific group to the more generic group in the
social system. Innovation providers must follow the dominant design in order to sell their products in the market place (Utterback, 1996). They concentrate their effort on the improvement of the cost-effectiveness of the product that embodies the innovation rather than exploring the variants, in order to prevail in the market.

After the stabilization and fixation of the innovation, full-scale diffusion process occurs within the social system. Now the innovation-diffusion process enters into the closure phase (Pinch and Bijker, 1987), in which a consensus emerges and the points at issue apparently disappear. The innovative artifact becomes an ordinary one and infiltrates into the social system. The social system enters into the state in which it accepts and allows the existence of the innovation. It alters its structure from previous one, and it cannot remain effectively without the innovation. As far as Japanese word processors are concerned, after they have diffused into the Japanese society to reach about 40 per cent of the Japanese household, they are being replaced by multimedia personal computers that have the Japanese word processing function. Now, even cellular phones have the Japanese word processing function in it. People in Japan cannot work or even live their ordinary lives without devices having the Japanese word processing functions.

4. SOME IMPLICATIONS AND CONCLUDING REMARKS

I have proposed the innovation-diffusion model based on the evolutionary perspective, focusing on the self-organizing system. I have depicted this model referring to the actual innovation-diffusion process of Japanese word processors in the Japanese society. Based on the result of the conceptual investigation as well as the observation of an actual innovation-diffusion process, I will make some propositions in order to reconsider the innovation-diffusion theory.

First of all, the concept of "critical mass" should be reconsidered. Critical mass is defined as "the point at which enough individuals have adopted an innovation so that the innovation's further rate of adoption becomes self-sustaining " (Rogers, 1995, p. 313). The critical mass is of the social system perspective, while the dominant design is of the technology perspective. Very likely, the critical mass would take place following the appearance of the dominant design, because without that the full-scale diffusion process of the innovation would not occur. Furthermore, the formation of the critical mass will result from the interaction of the social system with the artifact not only embodying the innovation but also changing its features as it diffuses into it. At any rate, we can hardly
Toshio Mitsufuji

determine the critical mass point due to the difficulty of the measurement of individuals' propensity toward the innovation, but we will have to be satisfied to determine the appearance of the dominant design. Some innovation-diffusion studies conducted in the 1950s and 1960s such as the hybrid corn for the US farmers (Griliches, 1957), the medical innovations in the US doctors' society (Coleman et al. 1966) focused on the innovations which some authoritative laboratories had already established. That is, the dominant design has been settled before the diffusion process begins. The model proposed here is in some sense a kind of an extended one of the conventional diffusion theory.

Secondly, we need to rethink the innovator-imitator dichotomy. Supposing that innovators would exist, they would act ardently in the second phase of this model as both engineers or members of relevant developing organizations who exert themselves to improve and enhance the innovation, and members of relevant social groups who exert themselves to interpret and shape the innovation. After the appearance of the dominant design in the social system, it is hard to think that there still exist innovators, at least it is almost meaningless to assume the existence of innovators in practice. In addition, we will need to reconsider the role of the opinion leaders, change agents, and so forth in compliance with this context.

Thirdly, on the premise that the internal influence factors (a), external influence factors (b), and the number of the potential adopters (K) affect the innovation-diffusion process, changes in these parameters cannot be ignored as an innovation diffuses into the social system. Especially these parameters before the appearance of the dominant design will be different from those after that period. Very likely, internal influence factors before the dominant design will be larger than after that, while the external influence factors before the dominant design will be smaller than after that period. Finally, the number of the potential adopters will increase as an innovation diffuses into the social system.

Therefore, fourthly, as these parameters would be controllable, so that strategic planners or policy makers could exhibit their abilities in order to diffuse the innovation. Even if it would be very hard to estimate precisely the rate of adoption, we could manage the innovation-diffusion process on the whole. Without doubt, the strategies to be taken before the appearance of the dominant design should be different from those after that period (5).

In this paper, I have focused on the innovation-diffusion process of a physical artifact into the consumer market, say, B to C market. However, it will be possible to expand and apply this investigation to the process innovation or IT software in the B to B market. For example, some dominant design for the business process in a firm should be established in its implementation process. Moreover, as a firm will also be able to control the
internal and external influence factors as well as the number of potential adopters in it, a planner should construct strategies to implement the innovation. In the same context as mentioned above, it would be clear that the fast diffusion of an innovation be not necessarily desirable.

To sum up, this paper has proposed a conceptual model to describe the innovation-diffusion process on the basis of the self-organizing system. Following the result of this investigation, it is necessary to survey empirically the applicability of the self-organizing system on the innovation-diffusion process. Consequently, the diffusion theory should be reconsidered in terms of the self-organizing system, for example, about the meanings of the critical mass, the S-shaped curve, and the dichotomy of the innovator and imitator model. In addition, even though it is hard to predict the innovation-diffusion process, we may be able to control the process so that we can and need to develop the strategic management and/or technology policy in order to implement the innovation successfully into the social system. Finally, applicability of the self-organizing system on the innovation-diffusion process of the IT products and processes should be surveyed in more detail.

**NOTES**

1. According to the theory of the dissipative structure (Prigogine & Nicolos, 1989), "[t]he emergence of the concept of a space in a system in which space could not previously be perceived in an intrinsic manner is called symmetry breaking" (p.12).

2. In spite of the discontinuities between before and after the appearance of the dominant design the diffusion curve seems to be smooth. Some scholars may say that why it is smooth. There may be several reasons to interpret the apparent smoothness, as Phillips and Kim (1996) describe; that is, (1) marketing data are usually aggregated, (2) marketing events are affected by environmental and economic forces, (3) data sources are dictated by the needs of business, and, (4) marketing data sets may be biased by virtue of excluding cases.

3. Assuming that there are two products, old and new one, it may sometimes take place that the new product suddenly diffuses and obtains most part of the market share. This phenomenon is called punctuated equilibrium (Loch and Huberman, 1999; Krugman, 1996), which forms one of the self-organizing system.

4. According to Mitsufuji (2000) describing the implementation process of electronic network systems in Japanese firms in 1990s, many of them did not care very much whether other firms had adopted the systems or not, but paid more attentions to their efficacy and applicability. Certainly because they had acknowledged the electronic network systems to some degree, they did not need to care much of other firms.

5. Geroski (2000) mentions especially about the public policy on the technology innovation, comparing the models of technology diffusion, that even if it has only "a
little window" for the innovation process, "small initial effects can have very large ultimate consequences."

REFERENCES


A Perspective of the Innovation-Diffusion Process


The Illusion of Diffusion in Information Systems Research

Tom McMaster

Information Systems Institute, University of Salford, M5 4WT, UK

Key words: Diffusion, diffusionism, colonialism, actor-network theory, SSM, Scandinavian tradition

Abstract: It is argued here that many of the ideas behind diffusion are remnants of European colonialism. As such they reflect a power imbalance in the relationships between users and traditional IS developers that we would be well-advised to rethink in favour of more participative and democratic design and development processes.

1. INTRODUCTION

Diffusionism does not consist of a single idea. It is a hugely dominant and pervasive set of concepts that have influenced, and continue to influence many of the disciplines and worldviews that are found in the cultural institutions of modern Western societies. Not only do these include the fields of anthropology, sociology, geography, education, and marketing among others (Rogers, 1995, pp.42-43), but it is the contention here that these ideas also influence how we see the world in much more subtle and ubiquitous ways. Diffusionist thinking may well have become part of the mental furniture of our everyday lives, and if so, it therefore profoundly affects the way that we see and how we interpret our surroundings. Why this might be so, relates to several hundreds of years of being inculcated into ideas that Western Europe is superior to the rest of the world, and that its citizens have enjoyed an inherent right, indeed, have had a moral obligation to colonize everywhere else. The arguments will be presented during the course of this paper, but for the moment a useful example to illustrate the point might be
provided by considering the traditional belief that a teacher standing in front of a class is somehow the source of (or at least a conduit for) knowledge—a common enough ‘diffusionist’ view—but one nevertheless that after a few moments of considered reflection will no doubt be recognized for the fanciful notion that it actually is!

The extreme ubiquitousness of these handmaidens of diffusionism—those assumptions and beliefs that have pretty well imbued it with all the qualities of being a self-evident truth, have meant that scholars almost never question it. Instead they conduct studies into matters that derive from, and/or are dependent upon the implicit acceptance of diffusion, such as, ‘rates of adoption’, and other such detail that form the subject of much of the diffusion literature. By doing so it is argued here, they perpetuate the illusion that diffusionism is somehow more than the pseudo-science that it is taken to be in this paper.

Here we are concerned with these matters insofar as they affect and impinge on issues that concern the information systems (IS) community at large, and in particular, IS researchers. At one level this paper is intended to provide a non-Rogerian introductory account of the tradition and history of diffusion—one that would almost certainly never be found in contemporary diffusionist work. At another it is intended to provoke a debate among IS diffusion scholars by presenting a view of diffusion that is altogether different and far less sanitized than that which is generally promoted. A few words then about the motivation and the content of the paper are obviously in order.

Briefly, concepts of diffusion are incompatible with those of actor-network theory (ANT—see for example Latour, 1987, 1993), an approach to thinking about the world to which this author subscribes (Vidgen & McMaster, 1996, McMaster et al., 1997a, McMaster et al., 1999). For example at a fundamental level, diffusionism takes ‘facts’ to be pre-existing (often hidden), waiting to be uncovered at some point by heroic discoverers and inventors—human characters that are deemed important and necessary to the diffusionist mindset. It is the mechanisms behind the ‘dispersal’ of these facts (ideas, products, artifacts etc.) through society that constitute the essence of the diffusionist model.

ANT on the other hand has no need for inventors and other heroes, since there are no facts lying around waiting to be invented discovered or otherwise found. Facts are instead dynamic networks formed by the alignment of allied interests (the ‘network’). As new human and non-human candidates (or actants as they are sometimes called) along with their respective interests join and converge with the general trajectory of the lengthening network, so its shapes, meanings and messages change. Over
time the network hardens into what is ultimately and essentially a ‘black box’ or what we come to think of as a ‘fact’.

In this sense the ideas of actor-network theory are more closely related to radical constructivist thinking than to the positivistic determinism underpinning classical diffusionist thought. Since the views represented in the two theories are mutually exclusive, it follows at least from an actor-network perspective, that there must be an explanation other than that promoted by diffusionists for the preponderance and dominance of their views. This paper thus is an attempt to tell a missing story. It does not purport to be ‘the truth’ (a diffusionist concept). It is instead rather more in the tradition of soft systems thinking (Checkland, 1981, Checkland & Scholes 1990, Checkland & Holwell, 1998), and other interpretive approaches to research that encourage multiple perspectives on the issues under investigation. It is the opinion and indeed the experience of this author that such approaches can, and generally do enrich our understanding of the objects of our study.

Here then, we consider something of the origins of diffusion theory, a dimension generally missing from modern diffusionist work. The paper is organized in the following way: the next section presents the background and a starting point for our story, noting that claims for the study of diffusionism may not be as mature as some sources would have us believe. Section 3 presents a historical account of diffusionist thinking that provides a context for seeing diffusion as part of a rarely uncontentious tradition extending back over several hundreds of years. Section 4 encapsulates the various arguments used to justify the superiority of European colonizers over those whose lands and property they coveted, and provides for us a possible framework for examining links between colonization and some contemporary IS/IT practices. In section 5 we construct a model of diffusion that shows its relationship with colonization, and in the final section we consider in conclusion, what this might mean for IS research and practice.

Let us begin by looking at the background to some of these issues.

2. BACKGROUND

Given Rogers’s claim for diffusion that “no other field...represents more effort by more scholars in more disciplines in more nations” (Rogers, 1995, p.xv), it is perhaps not surprising to note the formation of IS-oriented diffusion interest groups over recent years. These include DIGIT (Diffusion Interest Group in Information Technology) and the IFIP Working Group 8.6, which focuses on the ‘diffusion, adoption and implementation of information technologies’ (IT). In addition there are a considerable number of
researchers in the IS domain who work on matters related to the diffusion of IT outside of any formally organized group. What might be remarkable though, is that when we examine the outputs of (at least one of) these groups, we find little to support the kind of shared, coherent background knowledge that such a claim might reasonable lead us to expect. Consider the following table (table 1) which is a citations frequency list based on the first three IFIP WG8.6 meetings' (Levine, 1994, Kautz & Pries-Heje, 1996, McMaster et al., 1997b).

<table>
<thead>
<tr>
<th>Reference</th>
<th>Frequency</th>
<th>Reference</th>
<th>Frequency</th>
<th>Reference</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attewell 1992</td>
<td>5</td>
<td>Rogers 1983</td>
<td>5</td>
<td>Rogers 1983</td>
<td>11</td>
</tr>
<tr>
<td>Rogers 1983</td>
<td>4</td>
<td>Others * 8</td>
<td>2</td>
<td>Kautz &amp; McMaster 1994</td>
<td>3</td>
</tr>
<tr>
<td>Fichman 1992</td>
<td>3</td>
<td>Others * 315</td>
<td>1</td>
<td>Rogers 1995</td>
<td>3</td>
</tr>
<tr>
<td>Ives &amp; Olsen 1984</td>
<td>3</td>
<td>Others * 315</td>
<td>1</td>
<td>Zmud 1984</td>
<td>3</td>
</tr>
<tr>
<td>Markus 1983</td>
<td>3</td>
<td>Others * 315</td>
<td>1</td>
<td>Others * 18</td>
<td>2</td>
</tr>
<tr>
<td>Tomatzky &amp; Klein 1982</td>
<td>3</td>
<td>Others * 315</td>
<td>1</td>
<td>Others * 18</td>
<td>2</td>
</tr>
<tr>
<td>Others * 40</td>
<td>2</td>
<td>Others * 315</td>
<td>1</td>
<td>Others * 18</td>
<td>2</td>
</tr>
<tr>
<td>Others * 503</td>
<td>1</td>
<td>Others * 315</td>
<td>1</td>
<td>Others * 18</td>
<td>2</td>
</tr>
</tbody>
</table>

The Levine book contains some 30 papers offering between them 550 unique references, of which 503, that is 92%, are single ‘one-off’ citations. The other 47 references are variously cited more than once, thus; 1 reference is cited 5 times, 2 references 4 times, 4 references 3 times and 40 references twice. The Kautz and Pries-Heje book fares rather ‘less well’ with 325 single references in 12 published papers. 315, that is 97%, are one-off references, the other 10 are cited more than once; 1 reference appears 5 times, 1 reference 3 times, and 8 references twice. In the final offering, McMaster et al., there are 22 papers containing 458 references of which 435 (95%) are again ‘one-off’s’ with 23 replicated; 1 reference 11 times, 1 reference 4 times, 3 references 3 times, and 18 references twice.

Remarkable more perhaps for what might be missing than from what is actually there, such an apparent absence of common knowledge might seem astonishing, but a note of caution requires that we take care what we interpret from these figures. The sample is limited to 64 papers in IFIP

* We exclude the 4th meeting, the joint IFIP8.2/8.6 event held in Helsinki in 1999, because of the difficulty in separating the various submissions into their respective camps.
WG8.6 publication, so this clearly does not provide conclusive evidence of a lack of common knowledge in the field. No, but neither can we entirely dismiss what small evidence we might have, since this group was formed specifically to accommodate researchers in the diffusion of IS/IT. Also, a not unreasonable argument might suggest that 64 papers provides us with a perfectly fair and reasonable sample. What we can do, is to note that there seems on the face of it to be some discrepancy between what we might expect (given Rogers’s claim), and the reality on the ground. We should also note that the picture could be changed. If for example we were to remove ‘self-references’ from the data set, then the result would be that the paucity of ‘overlap’ (in background knowledge) would be even more marked than it now is.

What then can, or are we to make of diffusion and its claims? Let us consider something of the history and tradition of diffusionist thought.

3. A DIFFUSIONIST TRADITION

Rogers (1995, pp. 39-40) attributes the beginnings of diffusionist thinking to Jean-Gabriel de Tarde, a French sociologist and magistrate who became professor of modern philosophy at the Collège de France in 1900. What is interesting and significant for us, is that Tarde held ‘invention’ to be the source of all progress (‘progress’ itself let us note, is a quintessentially Victorian creation – see Bowler, 1989). Extremely few people (Tarde thought about one person in a hundred) are inventive, others merely imitate. These views were reflected in his best known work “The Laws of Imitation” published in 1890. The concept of progress, and the ideas that invention is rare and that the majority of people have a propensity only for mimicry, are central to the notion of diffusion.

Sir Grafton Elliot Smith, founder of Manchester University’s internationally renowned Medical School, at the end of the 19th century was one of the so-called ‘Grand Diffusionists’. These were a group of early English and German anthropologists about whom Rogers provides an all too brief, somewhat less than cursory reference. Smith refers in his writings to a number of earlier diffusionists including the 16th century Spanish writer Bernal Diaz. Diaz it seems had attributed the wonderful buildings he saw in the Yucatan peninsula of that period, to middle-eastern Jews of biblical times (Smith, 1933, p. 38). While the provenance of uncorroborated second-hand claims such as this might rightly be questioned, a far more serious oversight by Rogers however must be that of the Society for the Diffusion of Useful Knowledge (SDUK). This British institution was convened in 1825 with an impressive array of distinguished scientists, philosophers,
Tom McMaster

politicians, military officers and clergymen – all renowned senior practitioners in their respective fields. Of the 66 founding members, about half were Fellows of the Royal Society.

A publication entitled “The Penny Magazine of the Society for the Diffusion of Useful Knowledge” appeared weekly between 1830 and 1845 and this was distributed across the length and breadth of Great Britain. It was presented in an innovative illustrated format that paved the way for later publications such as The Journal of the National Geographic Society in the US in 1888, nearly sixty years later. One of the Penny Magazine’s stated aims was the promotion of literacy, although given the nature and position of some of the committee members it undoubtedly served as a useful medium for political propaganda.

What is interesting and possibly significant about the association with the Royal Society, is that a few years prior to the founding of the SDUK – in 1810 to be precise, the Scottish botanist, Robert Brown was admitted as a Fellow of the Royal Society. Brown was the man upon whose observations the term “Brownian Motion” was coined – that is the diffusion of microscopic particles in fluids. We may speculate as to what connection might have existed between Brown’s studies of physical microscopic diffusion processes, and the use of the term ‘diffusion’ in the anthropological sense of describing the sorts of socio-cultural phenomena that interest us. But neat as it might otherwise have been, Brown unfortunately was never a member of the board of the SDUK.

Other commentators and writers on diffusion who preceded Rogers included Bryce (1914) who wrote about the British Empire and the diffusion of British law throughout the world (we shall return to the relationship between diffusion and imperialism). There were also the aforementioned ‘Grand Diffusionists’ – Fritz Graebner and Wilhelm Schmidt in Germany, and in England William J. Perry and Sir Grafton Elliot Smith. Their views are epitomized by Smith (1927, 1933) who claimed to have found carvings of elephants and scenes from Vedic mythology in Mayan America, as well as Roman helmets in Hawaii. In a nutshell they thought that all culture emanated from a single source; specifically Egypt. However their contemporaries in social anthropology regarded such extreme and uncompromising views as eccentric to say the least.

At that time the counter-argument to diffusion was the ‘psychic unity of mankind’ – basically a belief that all (isolated) cultures and societies inevitably evolved through fairly predictable stages of growth; from ‘primitive’ to ‘savage’ to ‘civilized’. Similarities in their respective cultural artifacts and traits were part of a natural and inevitable evolutionary development process (a response to the recent impact of Darwin’s work),
and were not necessarily the result of the sorts of influences necessary for diffusion to have occurred.

In the 1920's Malinowski, one of the most important anthropologists of the 20th century, and considered to be the founder of social anthropology, said that “… culture can only be contracted by contagion, and that man is an imitative animal” (Malinowski, 1927), thereby reinforcing the views of Tarde. There was Wade too (1938), writing on the clandestine organization and diffusion of philosophic ideas in early 18th century France, and Bell (1948) who wrote on the diffusion and decay of Hellenism in Egypt.

These few examples represent only a tiny number of pre-Rogierian diffusion researchers and commentators. There are very many more who worked in this tradition (and rarely uncontentiously), both before and since Rogers’s first contribution in the early sixties (Rogers, 1962). However it is the political geographer Blaut’s account of diffusion that primarily informs this work (Blaut, 1987, 1992, 1993). Linking diffusion firmly to European colonial expansionism between the 15th century, and its culmination in what Hobsbawm (1987) describes as the Age of Empire, Blaut thus attributes the beginning of diffusionist thinking precisely to 1492.

4. 1492 & THE EUROPEAN ‘MIRACLE’

In 1492 the opening up of sea routes to the New World in the West, and around the southern tip of Africa to the East marks the beginning of a period of European expansionism that reached its peak in the late 19th century. By that time the British Empire was at its inglorious height and the arguments used to support its activities – the very ideas that underpinned the theory of classical diffusionism, were fully evolved. Although war, invasion, subjugation, slavery and exploitation were all features of European expansionist activity, these were not seen at that time as they might be today. They were the unfortunate effect of having to combat resistance and therefore could be justified. The colonizing Europeans felt they had certain moral duties, and these included bringing civilization to uncivilized savages, the true religion to heathens, and science, technology and other benefits to superstitious natives who were comparable only to children or animals at best. Since animals and children could not ‘own’ land or property, they (the colonizers) thus had a moral obligation to ‘administer’ these lands on behalf of the indigenous native sub-species, at least until such time as they might, through proper colonial tutelage, be brought into a state of advanced civilization (this of course represents the ‘generous’ view!).

Classical diffusionism was the creed that evolved to justify these activities. The supporting ‘evidence’, mostly taking pseudo-scientific forms
of argument, fell into various categories that can broadly be identified in terms of biology, and environment, as well as the cultural forms of rationality, technology and society. These arguments are nothing more than fallacious theories that were (and often still are) used as justifications or explanations for the myth of the so-called ‘European Miracle’. That is a belief in the innate ‘superiority’ of Europe and Europeans over the rest of the world.

In a somewhat diluted form, many of these arguments are still used to support diffusionism today. However before looking at some instances of these, it should be noted that sometimes it is difficult if indeed at all possible to distinguish specific issues as belonging to one particular category rather than to another, since some issues are often deeply enmeshed in more than one. It should also be noted that the term ‘European’ as it is used here is no longer confined to the geographical entity ‘Europe’, but also now includes the United States, Australia, Japan and other locations to some extent. In other words we are talking about the prevalence of modern Western societies’ values and cultures, whose origins were once confined to only to a few Western European countries, but which now are found across the (so-called) developed world.

A brief description of these categories and arguments – a summary indeed of Blaut’s (1993) more than excellent account, follows.

4.1 Biology

To prove their superiority over others, colonizers (Europeans) have often used ‘biological evidence’. This has worked on the two fronts of race and demography. In terms of race, the general assertions have been that Europeans are biologically superior to non-Europeans; they are brighter, better, and more intelligent due to heredity. Some Europeans, most notably those of Nordic stock (the ‘master’ race) were also considered to be superior to other Europeans. Non-Europeans were regarded as an inferior subrace at best, and at worst were considered not to be of the same species at all. Non-Nordic Europeans were often considered to have defective genetic materials accounting for ‘feeblemindedness’ and criminal inclinations. Support for these views saw the introduction of ‘race science’ into schools biology curricula in pre-WW2 Germany and mass extermination of the Jews by the Nazi regime. It also saw the introduction of eugenicist sterilization programs in the United States and Europe (especially Scandinavia), and one assumes, the euphemistic ‘ethnic cleansing’ atrocities perpetrated in more recent times in Central Africa and the Balkans.

The demography argument or Malthusian population theory (so named after Thomas Robert Malthus, the 18th century economist and demographer)
is concerned with population growth, or to put it a slightly different way, the control of sexual desire. Greatly simplified, this says that Europeans exercise a certain ‘moral restraint’, that allows them to control their sexual drives. They therefore are not inclined to suffer from overpopulation problems. Non-Europeans on the other hand have no such control, and as a result are constantly plagued by shortages of food and other resources necessary to maintain their ever-increasing populations. They therefore also pose a threat to the others.

4.2 Environment

Environmental determinism is the theory that the natural environment influences humans without reference to, or the mediation of culture. It is still used to explain the rise of European power and falls into two sets of theories; those used to explain why tropical regions (Africa and South America), and arid regions (Middle East and the Orient) are inferior to cooler regions (Europe) on the one hand. On the other, explanations as to why ‘temperate’ Europe is far superior to other parts of the world.

Tropical climates inhibit the progress of civilization. In the 19th century this argument was used to show why Africans remained uncivilized and must therefore naturally accept European colonial control. This was one of the core theories of classical diffusionism. The ‘tropical-nastiness’ doctrine consists of three main theories. The first concerns itself with the supposed negative effect of a hot, humid climate on the human body and mind, the second with supposed inferiority of tropical climates for food production, and the third with the supposed prevalence of disease in tropical regions. Where occasionally tropical regions were conceded as lush and bountiful, the argument went that this then presented too little of a challenge to humanity, and that no progress therefore took place except under colonial guidance. Other circuitous theories disqualify ‘the arid Orient’ from progress and civilization on the basis that arid regions have been denied the opportunity for development, because aridity necessitates irrigation, and irrigation leads necessarily to the kind of civilization that is historically stagnant.

‘Temperate Europe’ arguments include claims that northwest Europe represents a unique marriage of farming, iron and rain-watered land with deeper, wetter, more fertile soils that do not require irrigation. This means that farmers don’t have to spend as much time at farm work in order to satisfy their needs and quota of surplus, whereas Asian farmers need to work harder to achieve the same product. European superiority due to environment also includes a ‘capes and bays’ argument – that is, that the peninsulas, bays
and navigable rivers of Europe provide a natural basis for communication and trade denied to other continents.

4.3 Rationality

By the beginning of the 20th century – the heyday of the doctrine of classical diffusionism, most European thinkers came to accept the doctrine of the ‘psychic unity of mankind’, at least to the extent of agreeing that all of humanity shares a common ability to progress towards modernity. This took the form of a theory that Blaut (1995, p.95) calls the dualistic-developmental conception of human rationality. The elementary dualism was a distinction between the mentality of child and that of adult. The human mind has developed from a prehistoric condition which was mental childhood. Non-Europeans within this theory are seen as undeveloped, and childlike, but given the psychic unity of mankind thesis, they can be brought to adulthood, modernity and rationality through the colonial tutelage of rational Europeans. Ancient people are governed much more by emotion and passion than by intellect, just as is the case with modern children, and with some modification to the theory, the same applies to modern European women. Women too however would be able to experience mental development, and would eventually be rational enough to vote, and to hold public office.

European history is either to be explained as the fruit of mental development or has been intimately accompanied by such development in a process that was fundamentally the same as the psychological development from childhood to adulthood. Europeans became more rational as history progressed just as children acquire rationality in the course of ontogenetic development. The contrast is often also extended to psychotics, who were sometimes seen as having an arrested mental development. At the center of this model is the Rational Modern Adult European Man, contrasted with ancient European man, modern European children, modern European women, as well as modern non-Europeans.

Much of the ideas contained here may be encapsulated in the term ‘Weberian’, because Max Weber made important use of the idea that capitalism for example, was the culmination of a process of social evolution that reflected intellectual progression, the ascent of rationality from ancient to modern (European) society. Outside of Europe, societies were in varying degrees traditional and irrational.

4.4 Technology

When examined more closely, most so-called technological claims for the superiority of Europeans are generally to do with inventiveness rather
than technological determinism per se, and therefore belong to the ‘rationality’ argument described above. Nevertheless there are technological claims, though these are just as fallacious as the arguments in the other categories described. Examples of such claims include the invention of the iron stirrup, the horse collar, and three-field crop rotation among others.

The iron stirrup in medieval times permitted a new form of mounted warfare, and created the phenomena of the medieval knight. Since knights became manorial lords, this led to feudalism; a necessary stage in the Weberian evolution of capitalism, and thus the modern Western world.

The horse collar (the discovery of horsepower) transformed agriculture and grain transport in northern Europe by permitting horses to replace oxen for pulling plows and wagons. This led to the intensification and expansion of commerce since horsepower was faster and cheaper. Villages became larger because of an expanded radius of travel from home to field thus allowing villages – or towns as they were becoming, to have a church, a school and a tavern. Boys could thus learn their letters so that there could now be news from distant parts, and so urbanization and preparation for the characteristics of modern city life took place.

Three-field crop rotation was important because it reduced the proportion of fallow land from half to one-third. Oats could be grown more widely, hence a greater use of horsepower, and legumes could be grown resulting in a vastly improved European diet. This in turn explains the expansion of population, the growth and multiplication of cities, the rise of industrial production and the outreach of commerce during medieval times.

4.5 Society

At a societal level, various arguments have been proposed to account for European superiority. These fall into a number of sub-domains such as ‘state’, ‘church’, ‘class’ and ‘family’. For the purposes of this article, only state and class will be summarized.

The state argument is not too dissimilar to the environmental argument; that the fortuitous location and unique cultures of northern Europe bred a race of freedom-loving, individualistic and antidespotic people from which the modern democratic state, always immanent, naturally and inevitably sprang. In non-Europe a natural irrationality combines with environmental disadvantages to produce for example the kind of ‘oriental despotism’ found in China, India and the Islamic Middle East. The political infantilism of non-Europe is explained in terms of psychological deficiency, and irrationality in matters of intellectual vitality and innovativeness. Furthermore there is a moral failing in attitudes relating to the desire for progress, resistance to
domination, and the will to forgo animal pleasures – all of these we will remember, coupled to an inferior natural environment.

Classless regions and peoples are irrelevant to the explanation of Europe’s superiority and the world’s historical progress because classless societies are necessarily both unprogressive and primitive. Thus in many atlases of world history, sub-Saharan Africa does not exist – or at any rate no maps of Africa are shown from Upper Paleolithic times to the early 16th century. Such arguments typically ascribe causality (towards social ‘progress’ and modernization) to kings and other elite groups (the history of England is defined by kings and queens), and that the uniquely European phenomena of the medieval aristocracy was the central causal force behind the European miracle. The aristocracy was a band of comrades joined by bonds of feudal loyalty – a democracy in its own right, who also through the acquisition of private property also effectively invented capitalism. Elsewhere the aristocracy was ground under by despotic polity or became corrupted into a caste system such as in India, where the possibility for modernization therefore ceased to exist.

Now that we have established something of the pre-Rogerian tradition of diffusionist thought and at least some of the links between diffusion and colonization, let us turn our attention to the task of constructing a model of diffusion.

5. BUILDING A MODEL OF DIFFUSION

Notwithstanding Rogers’s definition of diffusionism, Jett and Kraus (1973) note that there is much confusion and misunderstanding surrounding the notion and the term – even they say, by those claiming to be diffusionists! These authors take the trouble to try and clarify the issue (ibid. p.144), thus;

“To unravel the facts, we have to distinguish between (a) the locally limited emergence of specific cultural traits, attributable to independent development (free from outside intervention of diffusion) and (b) the geographically separate occurrence of similar (or even identical) culture traits, which can be attributed either to independent parallel development, or to outside influence (diffusion)”.

Figure 1 illustrates the possibility of diffusion, but also the difficulties associated with its recognition. Diffusionists acknowledge the independent origin of solitary isolated cultural traits, but dispute the possibility of their independent, geographically separate parallel occurrence. Cultural isolationists generally attribute almost all cultural origins to independent development, but presenting this so-called ‘cultural problem’ as independent development versus diffusion, is to misrepresent it. The quarrel between
isolationists and diffusionists say these authors, centers on the mode of origin of geographically separate cultural similarities.

In figure 1, we see that the influence of a trait that has arisen in one (social) system may stimulate the development of the same or a similar trait in another system. None of this is by any means certain — making causal links is rarely easy and never simple. However let us assume the case. Then some features of diffusion are worth noting. The main and obvious one is that influence is overwhelmingly one-way. We may consider the first social system to be where the ‘creativity’ takes place — the center of innovation, and the other, the recipient community. As we have alluded to previously, diffusion is entirely dependent on the idea that there is a (single) source of innovation, and that others are capable only of imitation. There is however a small feedback trickle from the recipient community — counter-diffusion, because this is invariably negative; we shall see why in a moment, but to all intents and purposes the main process is unidirectional.

Let us return again to Rogers (1995, pp.262-280), because the characteristics of his ‘classes’ of adopter (table 2) will help us to refine our model further.
Table 2. Adopter classes and characteristics

<table>
<thead>
<tr>
<th>Innovators (2.5%)</th>
<th>Early Adopters (13.5%)</th>
<th>Early Majority (34%)</th>
<th>Late Majority (34%)</th>
<th>Laggards (16%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Venturesome</td>
<td>Respect</td>
<td>Deliberate</td>
<td>Skeptical</td>
<td>Traditional</td>
</tr>
<tr>
<td>Cosmopolitan</td>
<td>Integrated into social system</td>
<td>Followers</td>
<td>Cautious</td>
<td>No opinion leadership</td>
</tr>
<tr>
<td>(cosmopolite)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daring / risk-takers</td>
<td>Localite</td>
<td></td>
<td>Unable to deal with uncertainties</td>
<td>Ultra-localite</td>
</tr>
<tr>
<td>Resourceful</td>
<td>Often ‘opinion leader’ / role model</td>
<td>Scant resources</td>
<td></td>
<td>Social isolates</td>
</tr>
<tr>
<td>Gatekeepers (bring innovations from outside to inside the system boundary)</td>
<td></td>
<td></td>
<td></td>
<td>Point of reference is the past</td>
</tr>
</tbody>
</table>

Rogers (1995, p. 262-280) describes the various characteristics and qualities of adopters – from on the one hand ‘innovators’ – a small minority of resourceful risk-takers, progressive and cosmopolitan, and characterized by the term ‘venturesome’, to on the other hand ‘laggards’. These by comparison are backward looking country bumpkins afraid of any kind of change, though somewhat more politely described as ‘traditional’. As a matter of interest this author has asked many people over the years to classify themselves into one or another of these categories – never once has anyone ever described themselves as a
The Illusion of Diffusion in Information Systems Research

6. DISCUSSION

Some effort has been made in this paper to try and show the relationship between colonization and diffusionism, and to construct and present a model of diffusion that highlights those similarities. The colonial activities of...
conquering, settling and exploiting are diffusion in practice, but is there any relevance in this for IS community?

We may feel that we are not affected by these ideas, but consider the following table (table 3), which is part of a correspondence that was circulated to the membership of ISWorld towards the end of 1999 by a person well-known to the international IS research community. It was sent ostensibly as 'humor'. However knowing the sender it must be said that it is not even remotely possible that sender could ever have meant anything unkind by it.

Table 3. Rural American computer terms translated

<table>
<thead>
<tr>
<th>Term</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log On</td>
<td>Make the fire hotter</td>
</tr>
<tr>
<td>Log Off</td>
<td>Don’t add any more wood</td>
</tr>
<tr>
<td>Monitor</td>
<td>Keep an eye on that fire</td>
</tr>
<tr>
<td>Download</td>
<td>Getting’ the firewood off the buggy</td>
</tr>
<tr>
<td>Floppy Disk</td>
<td>What you git from carryin’ too much firewood</td>
</tr>
<tr>
<td>RAM</td>
<td>The thing what splits the firewood</td>
</tr>
<tr>
<td>Hard Drive</td>
<td>Getting’ home in a heavy rainstorm</td>
</tr>
<tr>
<td>Prompt</td>
<td>What the postal service used to be</td>
</tr>
<tr>
<td>Window</td>
<td>What to shut when it’s cold outside</td>
</tr>
<tr>
<td>Screen</td>
<td>What to shut in mosquito season</td>
</tr>
<tr>
<td>Byte</td>
<td>What the mosquito’s do</td>
</tr>
<tr>
<td>Bit</td>
<td>What the mosquito’s did</td>
</tr>
<tr>
<td>Mega Byte</td>
<td>What the Arkansas mosquito’s do</td>
</tr>
<tr>
<td>Micro Chip</td>
<td>What’s left in the bag after you eat the crisps</td>
</tr>
<tr>
<td>Modem</td>
<td>What you did to the hay fields</td>
</tr>
<tr>
<td>Dot Matrix</td>
<td>Ol’ Dan Matrix’s wife</td>
</tr>
</tbody>
</table>

Funny or not, it has the effect of unfairly belittling naïve users who are not given a voice to speak for themselves. And who has not heard stories of naïve users trying to use the mouse as a foot-pedal, of the CD drawer as a coffee-cup holder? The implications are not so different from earlier views of the (noble or otherwise) savage that enabled the colonizers to take possession of their lands and properties. I recall one senior systems professional saying “I don’t give them (users) what they want...I give them what they need”. This is more subtle perhaps than the users who said that the latest IT department’s initiative was just another “empire-building exercise”, but the colonizing attitude still comes through. How surprised should we be then that perhaps up to 80% of new systems implementation initiatives ‘fail’?

What we can learn from colonization, is that sooner or later the ‘colony’ is going to become independent by one means or another – either by previous arrangement such as in the handing back to China of Hong Kong on 1st July 1997 after originally being ceded to Britain in 1842, or by a
The Illusion of Diffusion in Information Systems Research

unilateral declaration of independence such as that made by Ian Smith on behalf of the former Rhodesia, on the 11th November 1965. It is perhaps too early to make judgements about the example of Hong Kong, but in the latter case today (35 years on), problems continue to make international news headlines – poor black Zimbabweans illegally occupy white-owned farms, and are encouraged to do so by anti-colonial rhetoric of that country’s leader, Robert Mugabe.

In the IT/IS domain, we see more and more of the IT Department’s traditional customer base declare a kind of independence as they ask why they still need a central IT department. As technology becomes increasingly easier to acquire and use, and the departments grow their own in-house expertise, they are inevitably beginning to question the relationship they have had with traditional IT professionals. What is needed is a more equitable relationship based not as it so often has been in the past on exploitation, but instead on trust and mutual respect. Over the last three decades various attempts have been made to address these problems, including for example the Scandinavian ‘Collective Resource Approach’ (Ehn and Kyng, 1987), in the UK Enid Mumford’s ETHICS approach (Mumford, 1983), Checkland’s Soft Systems Methodology (Checkland, 1981), as well as hybrids combining various aspects of each (Avison and Wood-Harper, 1990). As we attempt to design and implement new systems to support evolving business forms for the new millennium, will it prove that we have indeed learned something? Or will that insidious human propensity to colonize, either under the guise of ‘diffusion’ or perhaps some other euphemism continue to get the better of us?

ACKNOWLEDGEMENTS

My warmest thanks to Richard Vidgen and to Karlheinz Kautz for their comments on an earlier draft of this paper. Thanks also to my fellow IRIS23 team members (Kari Thorensen, Helle Zinner Henriksen, Petri Hallikainen & Kari Nurminäkki, Katarina Riabacke, Jan Kristensen, Ingegerd Andersson & Kerstin Nilsson), for their supportive and insightful comments on this work at the IS world’s oldest annual event (and its very best-kept secret!). Thanks also to the 8.6 reviewers for their critical comments and their helpful suggestions for improving the paper.
REFERENCES


Bell, H. I., (1948), Egypt: from Alexander the Great to the Arab conquest: a study in the diffusion and decay of Hellenism, Oxford University Press, Oxford.


Bryce, J., (1914), The ancient Roman Empire and the British Empire in India / The diffusion of Roman and English law throughout the world: two historical studies, Oxford University Press, London.


Latour, B., (1993), We have never been modern, Harvester Wheatsheaf, Hemell Hempstead.
The Illusion of Diffusion in Information Systems Research


Wade, I. O., (1938), The clandestine organization and diffusion of philosophic ideas in France from 1700 to 1750, Princeton University Press, Princeton.

Understanding and Changing Software Organizations
An Exploration of Four Perspectives on Software Process Improvement

Karlheinz Kautz¹, Henrik Westergaard Hansen², and Kim Thaysen²

¹Copenhagen Business School, Department of Informatics, Howitzvej 60, DK-2000 Frederiksberg, Denmark
Karl.Kautz@cbs.dk
²MINDPASS AS, Vardevej 1, DK-9220 Aalborg, Denmark
hwh@mindpass.com, kth@mindpass.com

Abstract: In this article we discuss four different perspectives on software process improvement, which are all based on quite different assumptions. The objective is to expand the views on software process improvement and contribute to a wider understanding of software process improvement. This might facilitate the application of software process improvement and assist in further spreading the approach. The different perspectives are expressed through four different metaphors for the work of process agents. These describe process agents as (1) technical experts, (2) facilitating participants, (3) political agents, and (4) individual therapists. We argue that the four perspectives do not preclude each other and that they can be applied to more or less effect to understand different process improvement situations. The advantages and disadvantages of each perspective for improvement work will be discussed and illustrated by examples from an ongoing software process improvement project.

1. INTRODUCTION

Software process improvement deals with understanding and changing development practices in software producing organisations. A person who has the explicit role to understand and change a software organisation’s software development processes is called a ‘software process change agent’, a ‘software process improvement consultant’ or, in short, a ‘process
consultant’ or ‘process agent’. Behind software process improvement theory and practice there lie a number of assumptions, both explicit and implicit, about the world in general, about software development in particular and about the knowledge about it that can be produced. These assumptions or perspectives indirectly guide the way process agents perform their work.

In the traditional software process improvement literature one perspective is predominant. However, in this article we will discuss four alternative perspectives on software process improvement that are all based on somewhat different assumptions. The objective of the article is to present these alternative perspectives in order to expand the views on software process improvement. By doing this we want to contribute to a wider understanding of software process improvement and to illustrate the complementary ways in which process agents could analyse and assess a software organisation when attempting to introduce change and improvement in the work processes of these organisations.

As it is our belief that theorising about software process improvement can benefit from research performed in the broad domain of organisational science and in particular from research on organisational change. We utilise a conceptual framework based on Burrell and Morgan’s (1979) work on different perspectives in organisational analysis. This framework has been applied previously in the field of systems development by Hirschheim and Klein (1989). Their work concentrated on different ways of understanding the problem areas for which information systems and software are developed, focusing in particular on the organisation and the people using IT. They examine the different ways in which systems developers view organisations and apply different methods. However, Hirschheim and Klein do not address the ways in which process agents and software developers view and understand software organisations.

Here we apply Burrell & Morgan’s framework (1979) to understanding software process improvement focusing on the roles of process agents. We argue that process agents act according to different perspectives or logics. These can be expressed through four different metaphors: process agents as (1) technical experts, (2) facilitating participants, (3) political agents, or (4) individual therapists. It is argued that the four paradigms do not preclude

---

3 In this field also other authors, f. ex Nurminen (1988), Avison & Wood-Harper (1990), and Walsham (1993), distinguish different perspectives. Beyond that Dahlbom & Mathiassen (1993) provide philosophical considerations about diverse frameworks for systems development. Borum (1995) introduces an alternative framework for understanding organisational change in general and Kienholz (1999) differentiates viewpoints on inquiries as vital elements of learning organisations. None of these, although may be inspiring, will be discussed here.
each other and that they can be usefully applied in different situations of process improvement endeavours. The advantages and disadvantages of each perspective for improvement work will be discussed and illustrated by examples from an ongoing software process improvement project. This might contribute to a more successful application of software process improvement and a further spread of the approach.

The article is structured as follows. The next section presents the background and methods for the research. Following this the relation between Burrell & Morgan's (1979) framework and software process improvement is clarified. This provides the basis for a detailed presentation of the four perspectives. These are then compared and, based on this comparison, conclusions are drawn concerning the work of process agents and the performance implications of software process improvement projects.

2. BACKGROUND AND RESEARCH METHODS

The results presented here are based on a software process improvement project in a small Danish software company. This company had 60 employees and was developing one main product, namely an intelligent WEB portal. The authors were actively involved in assisting the organisation through this process improvement project over a period of over 2 years. During this time as process agents they performed a variety of different roles and activities. These included:

1. observation and assessment of the organisation's current status
2. analysis and interpretation of the problems and actual assessment results
3. elaboration of procedures and standards
4. introduction of appropriate procedures, techniques and tools
5. education of personal
6. participation in working groups
7. supervision, tutoring, mentoring, and coaching.

Given that the authors participated actively in the entire process and intervened and changed the organisation’s practices, the work can be characterised as a longitudinal study based on the principles of Action Research (cf. Argyris & Schön, 1991).

From the outset the improvement work was oriented to a large extent towards the Capability Maturity Model (CMM - Humphrey, 1989). The idea of this model is that all software organisations can be categorised according to one of five levels of maturity. To improve, they should pass sequentially through all stages of maturity using the 'IDEAL' model (McFeeley, 1996). The IDEAL model describes the phases, necessary activities and resources which are needed to implement and manage software process improvement
in an organisation. The model (I for initialisation, D for diagnosis, E for establishing, A for acting and L for leveraging) is cyclic to allow for continuous improvement. This model was successfully utilised in the project that forms the empirical basis of this research (Kautz et al, 2000).

At the end of the first improvement cycle, a process evaluation was produced. This concluded that there was considerable deviation in this organisation from the models of process improvement prescribed in much of the literature. The models and methods prescribed had, then, to be adjusted in a number of ways in order to be made applicable to this organisation. Among other issues, as process agents we were confronted with the actual problems experienced by staff - namely the lack of appropriate meeting guidelines and structures - and digressions from the IDEAL and the CMM models. That said, we also found elements of these models useful to frame our work.

This experience provided the stimulus for this paper. In particular it indicated a need to understand the differences between the theoretical models and methods as proposed in the conventional literature and our own experience and approach. Thus the remainder of this paper attempts to explain different perspectives on process improvement and to provide a framework which can be used when considering the applicability of different approaches in future process improvement endeavours. The main emphasis is on showing how various perspectives can be used constructively within process improvement projects, rather than on a critique of the dominant model per se. The purpose is to show how the process agent, by adopting different perspectives, can identify the different areas of improvement necessary for successful software process improvement. Each perspective thus provides a different focus on software process improvement and the perspectives complement each other. The four perspectives are introduced next, based on a conceptual analysis of the literature, and are illustrated by examples from the software process improvement project in the small Danish software enterprise in which the authors participated.

3. SCIENTIFIC PARADIGMS AND SOFTWARE PROCESS IMPROVEMENT

Burrell & Morgan (1979) use the concept of ‘paradigm’ in their work. This suggests that human beings see the world in particular ways, or through particular lenses, and are necessarily aware of, or conscious about, their own predetermined world views. Process agents are no exception. Kuhn (1962), in his analysis of scientific work, defines such unquestioned, scientific assumptions within a discipline as ‘paradigms’. As long as one paradigm
dominates, scientists work within the domains of ‘normal science’ - they try to make facts fit the theory. However, if too many anomalies are found, a paradigm ‘shift’, or change, takes place. Kuhn (1962) puts forward the notion that in science paradigms take over from and replace each other. However, in the humanities different paradigms can co-exist. Burrell & Morgan (1979) argue from a social science perspective and suggest that there are four competing paradigms to understand organisations that exist in parallel.

The four paradigms are shown in Figure 1. These differ according to two underlying dimensions. The first dimension is defined in terms of different philosophical approaches to sciences, roughly speaking by distinguishing between objective approaches (with characteristics like realism, positivism, determinism, and the belief in quantitative methods and universal laws) and subjective approaches (characterised by no belief in the existence of a social world external to the individual, anti-positivism, voluntarism, ideographic methods and personal understanding and enlightenment). The second dimension is determined in terms of different views on society, roughly speaking by distinguishing between explanations of society as based on social order, consensus, social integration, satisfaction of personal needs (this being called a sociology of regulation) and explanations of society as concerning structural conflicts, dominance and power, contradictions and deprivation (a sociology of radical change).

Taken together the two dimensions generate four paradigms: functionalism, interpretivism, radical structuralism, and radical humanism. To be within one paradigm means that one sees the world in a particular way, which is fundamentally different from each of the other paradigms. In other words, these are fundamentally different ways of analysing, understanding and handling social phenomena. One cannot work within more than one paradigm at a given moment in time. As Burrell & Morgan (1979) express it “they are alternatives in the sense that one can operate in different paradigms sequentially over time, but mutually exclusive, in the sense that one cannot operate in more than one paradigm at any given point in time, since in accepting the assumptions of one, we defy the assumptions of all the other”. However, the distinction among the paradigms is useful as it provides “a tool for establishing where you are, where you have been and where it is possible to go in the future”. This is the foundation for the following arguments in this paper since it is these paradigms which allow process agents to look in different ways at organisations. They provide them with different ways for understanding and changing software organisations’ development processes.
Applying these four paradigms to software process improvement leads to four stereotype descriptions of process agents and their work:

As a technical expert, the process agent operates in a functionalist paradigm. He believes that he can fully understand the problem area with the help of a formal assessment based on a predefined best-practice based model. Empirical data is objective and the truth is shared. Every qualified researcher can find it provided, of course, they use the correct scientific method. A functionalist shows statistical relations between phenomena. His assumption is that with rational and structured action, he can get the improvements implemented in a fast and efficient way. He also believes that the organisation can be completely controlled by introducing procedures and standards to perform work processes. As a technical expert he has ‘unique’ knowledge about how the process should be best carried out and this knowledge has to be transferred to the organisation.

As a facilitating participant adopting an interpretative perspective, the process agent bases his work on the assumption that the world is socially constructed. He tries to understand the processes, even if he believes that several different perceptions of reality exist and that complete understanding is impossible. He observes social processes to learn more about the participants’ subjective opinions and the ways in which these are constructed. Facts are not static, but based on changing social definitions - the parts of the phenomenon can only be understood in relation to the

4 The usage of the male form is no expression of gender discrimination, but merely serves readability.

5 These are pseudo explanations, they demonstrate statistical correlation between observable facts, but the statistics themselves can not give explanations.
context as a whole and vice versa. The ideal is to understand people in situations, but not to explain and predict. As intersubjectivity creates reality, it is impossible to relate to the future. The process agent accepts that there is no one complete solution for all organisations. Problems, solutions, and approaches can not, therefore, be determined by the process agent alone. The focus here, then, is more on the process agent as performing a consulting and facilitating role where the members of the organisation discover for themselves the improvements and solutions which are relevant for them in their particular situation.

The process agent as political agent in the radical structuralist paradigm will try to recognise and resolve structural conflicts among different stakeholder groups in the organisation, but he actively sides with one group. The process agent strives for change through influencing the tensions and contradictions among organisational members. Understanding is related to objective, not personal, value carried in facts concerning structural relations and relations of dominance. A political agent supporting one group uses dialectics - the definition, analysis, and debate of thesis and antithesis - to elucidate the situation under investigation and brings them into play to persuade or convince the opposite party. The process agent believes in radical change. At the extreme he sees conflict and chaos as ‘healthy’ - i.e. as something that contributes to continuous improvement. Dialectical arguments provide possibilities for breaking down deep-seated structural conflicts and states of dominance.

As an individual therapist in the radical humanist paradigm, the process agent assumes that reality is socially constructed. It is a product of the individual subject who can be influenced by psychological and social processes and focuses on how human beings can be encouraged to leave their ‘psychological prison’. Understanding is produced by investigating how individuals create their psychic worlds and how this delimits their world. The process agent works with the different individual subjects’ attitudes and opinions, because he recognises that the world(s) are created by the individual(s). It is not essential in this paradigm that the developers have a shared understanding of the process, but the strength lies in the different thinking among members of staff who have different views on the process. Process improvement happens through ‘treatment’ of the personal limitations that hinder the ability of the human to unfold and think in different ways and thus also limit the organisation’s success.
4. THE FOUR PARADIGMS

The basic idea of all software process improvement is that there is a relation between quality of the product and the organisation's capability to perform the software process - the quality of the process. In the four paradigms different approaches are taken to improve this process. These will now be presented in more detail using examples from the improvement project in which we participated.

4.1 The functionalist perspective: the process agent as technical expert

In the functionalist paradigm, process improvement is based on prescriptive reference models, such as the CMM, representing a fictional optimal state and defining the so-called key process areas that constitute this state. The overall objective of the improvement process is to ground an organisation's work processes upon a rational approach. The assumption is that, through standardisation based upon a reference model, a common foundation from which to estimate, plan, control and perform development can be achieved. The objective of working with process improvement is profit maximisation through better quality and productivity.

According to this paradigm management and process agents are the main actors. They define the goals and objectives for the improvement endeavour. A process group has the leading role in the implementation of the improvement. The process agent is, in this situation, the professional, technical and impartial expert who identifies an organisation's strengths, weaknesses, maturity level and profile through an objective evaluation of the current situation in relation to the chosen reference model. Through this evaluation, the process agent develops and implements an improvement plan. Professional staff have to be at the assessors' disposal to provide the data that is required.

To understand the problem area the organisation's current practices are assessed in comparison to the reference model's prescriptions. As the reference model predefines which processes should be performed, the actual problems as experienced by staff are only of secondary interest, if considered at all. Questionnaires and individual interviews are the preferred means of investigation. To achieve objectivity, the answers to the questions and the observations made have to be based (for example in the CMM) on at least two different independent sources or have to come from at least two different data collection sessions.

To change the problem area process agents work with the predefined key process areas, look systematically at the organisation's procedures, standards
and policies and bring them into agreement with the reference model. Through standardisation a rational work process determining all development processes is described. In this way there is no doubt about how the process should be conducted. By following the descriptions of all key practices as presented in the reference model, procedures are defined for the execution of key processes. The questions from the questionnaire can be used as checkpoints for the elements of the reference model. As the processes defined represent ‘best practice’, following them will lead to the development of high quality software and satisfactory working conditions.

One example in our case of the functionalistic paradigm was management’s request for one character to describe the organisation’ capability with respect to the CMM. The process agents delivered this character by using the methodology’s approach for the determination of characters - mainly by counting the number of ‘yes’ and ‘no’ answers given by the project leaders to questions on a survey instrument concerning CMM level 2 (Kautz et al., 2000). A second example was the introduction of configuration management routines. The assessment had shown that no configuration management routines were in place in the organisation, nor were the employees familiar with the concept. Therefore the process group worked with this key process area on its own and without further consultation of the staff. Based on a literature review and available routines used in similar organisations (Kautz, 1998), the process agents, as technical experts, developed rules and support tools in an authoritarian manner and implemented these in the organisation when first versions of the organisation’s product had to be distinguished. Although not involved in the development process, the routines were accepted by the staff and have been utilised by them since. Finally, in the same way as technical experts the process agents developed a set of templates for requirements specifications.

The functionalistic approach has a number of disadvantages. First, it does not really take into account what staff consider to be problematic and actual problems. Second, the classification of maturity level, although a useful indication, provides only a limited insight in the situation. Finally, even proponents of the functionalistic approach (e.g. Zahran, 1998) acknowledge that assessments based on the functionalistic approach also have a large subjective element. This brings us to the next perspective.
4.2 The interpretivist perspective: the process agent as facilitating participant

Within this paradigm process improvement is based on the belief that software organisations are subjectively understood, based on human interpretation. Staff members from different organisational levels have different perceptions of what the problems are and how to solve them and, as every organisation is unique, there is no single identifiable best practice.

The main objective here is not to benchmark but, rather, to identify and develop a shared understanding of problem areas and improvements. Different objectives are recognised and acknowledged as legitimate. The process agent’s task is to combine these and to try to satisfy all stakeholder groups. The process agent’s objective is to achieve a form of agreement about what the problems are and how they can be solved. This is achieved through involvement and participation. Thus, according to this approach, all members of staff are main actors. Process agents consider all the different opinions with the aim of reaching consensus in the organisation through discussion and negotiation. This might take the form of compromise or persuasion where one group is able to convince another that it is right.

This approach builds from the belief that organisations can not be understood and appreciated solely on the basis of structured questionnaires and interviews which aim, for example, to classify the organisation according a maturity model and from there derive improvement proposals. The process agent, in this case, is convinced that not all strengths and weakness can be identified based on a pre-fixed, predefined questionnaire or interview schedule. It might be necessary to define questions about non-technical aspects such as organisational and cultural issues. For example, the Bootstrap methodology (Kuvaja et al., 1994), although also based on a predefined questionnaire, is an attempt in this direction. In this case the assessment methodology is used to start a dialogue with and among members of the organisation. The purpose is to comprehend and to look at problems from different angles. Therefore a significant part of the assessment is always a group interview or an in-depth discussion in which the process agents act as facilitators and participants. They promote debates and inform understandings with their observations. They support the organisational members who themselves identify the problem areas as they perceive them and not as they are determined by a reference model. Improvement proposals are developed by the staff through active participation in the discussions with the process agents. To achieve change the process agents support the establishment of working groups and act as
participants and facilitators but not as technical experts while solutions, procedures and standards are defined by the working groups themselves.

There are several examples for this paradigm in our case. At least two different objectives were identified and accepted, namely top management’s request for a maturity level character and profile and the project leaders’ need for better project and resource planning routines. Both demands were jointly satisfied. An example of shared identification of problems and solutions was the recognition of lack of discipline in meetings, which was mentioned in all assessment discussions. For example, many meetings were held, but the resulting information was not communicated to all the relevant people. There was a lack of structure and documentation rules. During the interviews the employees made significant proposals for improvement. A working group consisting of interested staff members was established and the process agents scheduled a date for their first group meeting and appointed one person as responsible for the preparation of that meeting. They also participated in that meeting. Then, the group needed two more sessions to develop a solution. They then informed the other staff members who accepted the proposals they had prepared. No further action for the uptake of the routines had to be taken as all employees had been involved in the definition process. Finally, after the templates for requirements specification had been in use for some time, different needs for the description of requirements emerged. A new working group, in which the process agents again participated only at the outset, was established. This group developed a second set of templates, which were subsequently utilised by all other staff members successfully.

A problem with the interpretivist view is that when assessments are only based on open discussions and subjective perceptions, problem areas as described in the improvement models might not be recognised at all. This brings us to the next perspective.

4.3 The radical structuralist perspective: the process agent as political agent

In this paradigm, process improvement is based on an understanding that the world objectively exists external to individual cognition and independent of human consciousness and interpretation. Reality in software organisations consists of tangible and observable tensions, contradictions, disagreements, and paradoxes between people concerning existing development practices and improvement proposals. These tensions exist between many stakeholder groups: between top management and project management, between top management and development staff, project management and development
staff, between management and process agents, and process agents and developers, and one group might exercise power upon the other.

As political agents process agents look for, identify and resolve conflicts between different stakeholder groups. However, they do so by choosing a side rather than necessarily aiming at a compromise. The objective is to resolve contradictions.

Understanding and change are interrelated. The process agents use dialectical analysis, identifying or developing a thesis and an antithesis, and building a synthesis to clarify a problem and propose a solution. They are not fundamentally interested in the different perspectives different stakeholders have on the world. For them, these are expressions of conflict and dialectic contradictions between different interest groups. Their aim is to try and find regularities and rules to apply to the dialectical contradictions.

In the belief that people are shaped by external factors, the process agents believe that by influencing the contradictory factors people’s actions can be changed. However, they are aware that sometimes it is not enough to simply change people’s perceptions of a situation. Sometimes for example, real change in the distribution of resources is needed to improve the situation.

Through the process of shaping dialectical tensions the process agents trigger change. As a starting point for change they primarily use debates. In discussions for example, they often attempt to negate the prevailing position and by so doing in a dialectical manner they try to elucidate truth, - that is the truth of the party they have chosen to support. Thus, they might engage in confrontation with those who have power, although this is not necessarily inevitably. Members of staff are thus both objects when subject to influence and subjects when involved actively in the improvement process.

In contrast to a functionalist, who is sure what to do and which processes to change, a political process agent acknowledges that dialectical tensions are continuously changing and that it is therefore impossible to precisely predict organisational development. Therefore they do not attempt to precisely design the work processes for the developers, but instead use this uncertainty as an opportunity to experiment with alternative possible solutions.

We can illustrate the political perspective in our case using two examples: After two separate discussions and assessment sessions with management and project leaders we identified two different perspectives on project planning. Management saw a project plan as a definitive contract between themselves and the developers to be drawn up at the beginning of a project - the developers committing themselves to optimal performance within a given time frame. Management naturally wanted to minimise this time frame. The project leaders however, saw project plans as a device to be used during the whole development process. It was to be used to optimally
structure activities and to plan, re-plan and distribute resources in order to avoid bottlenecks throughout the project. For project leaders therefore, it was not essential to produce an entirely ‘perfect’ plan at the start of the project. What needed to be ensured was that it was updated appropriately during the process. As process agents we had to clarify the project plans’ significance for the course of a project. We convinced management that the overall scope and associated tasks within a project could be defined without necessarily determining and subsequently sticking to detail planning from the outset. Although this was understandably difficult for them, management recognised that such detailed planning was not possible for innovative projects. In this case then, we supported the project leaders in their perception of project plans as tools to be used throughout the project rather than as a binding contract which up front specified the course of the project in its entirety.

The introduction of a requirements specification also had a political dimension that was understood with the help of dialectics. After having previously ignored requirement specifications, management had subsequently emphasised the importance of them. We had to moderate their expectations and request time, as staff did not see the necessity for managing requirements and developing requirements specifications nor did they know how to develop them. We therefore had to convince staff that because of permanent time pressure they actually did not have sufficient time to not manage their requirements. We thus became the negation of their perception of what good development practice was. As a synthesis in a timely process we developed and demonstrated a way to handle requirements through the use of simple templates. The templates themselves were functionalistic (see sec. 4.1) and were subsequently re-developed co-operatively (see sec. 4.2).

It is a significant challenge for process agents to manage all the contradictions at all levels within an organisation. This brings us to the last perspective on process agents, where the focus is upon individual staff members.

4.4 The radical humanist perspective: the process agent as individual therapist

This perspective assumes that process improvement is grounded in an understanding that individual staff members are the starting point for any improvement in an organisation. The humanist paradigm deals therefore with learning about individual’s strengths and weaknesses, their background, their knowledge, and their limitations and breaking down the barriers that hinder them as a fundamental prerequisite that will improve their capabilities and thus increase their effort.
As therapists, process agents move beyond being aware of different interest groups with different views - acknowledging that there is no world external to the individual, but that there are different individual views of the world, which are based on individuals' different mental models of the world. Process agents therefore see conflict as subjectively created and not objectively existing. Conflicts delimit the developers' unfolding worldviews. When they are resolved and the developers are rid of these limitations, a reflection process can start which might result in improvements.

Improvement aims to develop emancipated, engaged, motivated, and innovative staff. Improvements can therefore be achieved through promoting personal development rather than through the use of standards and procedures. From this perspective, it is not essential that staff have a shared understanding of the process. In fact the strengths for the organisation lie in staff having naturally different perspectives on software development.

To understand the problem and to alter practice the process agents try to come close to the individual subject and to involve themselves in individuals' daily life. In so doing they try to understand how staff members create, modify and interpret the world they are a part of. During the formal assessments, and beyond, in informal conversations, process agents engage in a close dialogue with the individual in order to find out which barriers and conflicts hold them back from improving their own and others development processes. The process agents help the staff members not only to judge their existing situation, but also influence them to engage in a reflection and change process.

In our case, several examples - especially the introduction of requirements specifications can illustrate the perspective of process agents working with individuals. Initially, a highly respected project leader was chosen as a champion for the whole improvement endeavour to eliminate a possible block by the development staff. Requirements specifications were not originally considered a necessary and valuable development task. Numerous individual sessions were needed to work on managers' individual subjective attitudes and to open the developers up to the idea of developing ways to document requirements using templates. However, even when doubts were assuaged, several staff members refused to sponsor or promote the introduction of these more formal routines. They wanted to avoid a confrontation - to be perceived as campaigners for change in a comparably egalitarian organisation with many informal work practices. The confirmation that the majority of staff actually wanted greater levels of formalisation eventually resolved this situation. In addition, the refinement and amendment of the specification templates was initiated based on knowledge about individual staff members and their influence on removing further obstacles. For this purpose a working group consisting of newly
employed, greatly esteemed staff members was formed to work on the refinements. This approach led to the ready acceptance of the refined templates.

The requirements specification example also demonstrates that the radical humanist paradigm would be too ambitious and unrealistic if process agents attempted to deal with all individual staff members' subjective perceptions and attitudes. In addition, process agents have to behave in a similar fashion to a psychiatrist and this might be somewhat overwhelming when confronted with some limitations that do not stem directly from the organisations' work practices, but from the staff member's personal background.

The requirements specification example also begins to highlight the way in which different paradigms are intertwined. This will be illustrated in more detail in the next section.

4.5 Shifting Perspectives and Paradigms

In our project the different paradigms have been used in different situations and contexts. To take what from our perspective was the most appropriate action we initially unconsciously, but later, following the first evaluation, more consciously shifted from one paradigm to another under certain circumstances. In the course of our project, therefore all paradigms were utilised, the improvement of the requirements management and specification process as described in the preceding sections serves to illustrate this point.

In the following subsection we provide two more coherent examples to demonstrate when and why we shifted paradigms. The first deals with the introduction of another individual improvement action, namely the implementation of the key practice estimation as part of the CMM's level 2 key process area project planning. The second covers the first full improvement cycle of the project following the IDEAL model.

4.5.1 The Key Practice Estimation

The starting point for the introduction of estimations was the fact that during the initial assessment staff constantly mentioned that they were permanently under tremendous time pressure and that the only estimate for performing a task was a fixed deadline set by top management.

As a first step the process agents scheduled a meeting where they facilitated a discussion to bring the different points of view and opinions out into the open. In that meeting management argued that the estimates fitted well, while staff disagreed. However, management made public how they
reached their estimates – fundamentally these were purely based on market pressures. For example, estimates would be driven by the need to present the firm’s innovations at a trade fair before competitors did. Although staff still thought that they had to work too hard to finish a deliverable within deadlines, they now understood the rationale behind the estimate. Staff therefore accepted it for the time being, agreeing as a compromise with management to start working on a more advanced estimation method.

Earlier we described how the process agents as political agents supported the project leaders’ campaign for project planning (see sec. 4.3). In the case of estimation a dialogue had been initiated with all individuals from the different stakeholder groups to trigger a different way of thinking and a more positive attitude with regard to estimates. The developers were used to working with deadlines that were not based on realistic estimates and overruns were normal. Thus, they did not doubt the benefits of an estimation method. However they, and to an even greater extent management, had some reservations concerning the usefulness of estimates. They were fundamentally perceived as lacking certain preciseness and the finality that deadlines had. To resolve this contradiction, the process agents initiated a discussion about estimates as flexible devices for the distribution of resources and the management of work loads – less overworked staff would obviously be advantageous to both parties, and argued for the necessity of a trial. This being accepted by all involved, the process agents developed a very simple estimation method distinguishing between best, medium and worst case scenarios in terms of calendar and person days. Recognising that this method was purely functionalistic, it directed attention at the process of estimation and with increasing experience and feedback, it was subsequently changed and replaced by a more sophisticated approach based on collected data.

4.5.2 Following the IDEAL Model

Following the IDEAL model in the initialisation phase we acted entirely as technical experts to convince the organisation how we could help them to improve their development processes. We presented typical problems from other organisations, stated our knowledge about process improvement, and emphasised the benefits of a planned, structured course of process improvement organised as a project. Among other things we presented CMM’s level 2 processes in detail.

In the diagnostic phase a tailored CMM-inspired approach was chosen to perform a specific appraisal and a more general organisational analysis. The project leaders completed a questionnaire especially designed for CMM level 2 assessments and development staff were interviewed before and after
the questionnaire sessions. In all more than 50% of the employees were directly involved in these activities. In addition, documents were reviewed and observations were made. The questionnaires were completed while the process agents were present for necessary clarifications. The results of the interviews and questionnaire data were the basic material for the requested, quantified profiles. The answers from the questionnaires were then supplemented and substantiated by the interview results. For these the process agents had developed an interview guide that was based on the survey instrument, but which used more open questions. During the interviews the process agents asked the employees what they experienced as problems and not what a model like the CMM defined as a potential problem area. Thus, problems that had nothing or little to do with the CMM, e.g. the lack of structure to meetings were identified. The interviews were not merely used as a means to collect data, but also to generate a discussion and dialogue with and among the developers that were involved. Subjective opinions were expressed and the developers pinpointed not only problems, but also made significant proposals for solutions. Thus, the process agents did not simply act as technical experts, they also clearly acted as facilitators and to a certain extent as therapists in the interview sessions. Finally, however, as the process agents had to satisfy different stakeholders, an entirely functionalistic maturity profile as demanded by management as a part of an assessment report was produced and presented to the organisation together with other results and recommendations.

In the establishing phase the process group worked with three main tasks, namely a further refinement of the improvement proposals, a prioritisation of the proposals and the development and documentation of the final plan for action. The governing parameters for the prioritisation were to delimit extra economical resources and to delimit the additional workload for the employees. Through placement in a life cycle model for the product development it had become clear which improvement proposals fitted best to which development activities. We proposed radical change as some of the processes we suggested did not exist in the organisation. Although we recommended some measures that were not covered by the CMM, we undoubtedly used our technical expertise to make and support the propositions. The work in this phase was also influenced by the fact that all participants in the meeting where the diagnosis results were presented judged two acute problems as so important that they immediately founded two technical working groups to resolve these problems with the approval of management.

The first activity in the acting phase, which can also be considered as an establishing activity was the founding of the two working groups. Here clearly a participatory approach was taken. All employees were in line with
their own preferences assigned to one of the two temporary groups. The process agents scheduled dates for first group meetings and appointed one person as responsible for the preparation of that meeting. They also participated in the first meeting of each group. Afterwards the groups worked on their own to develop solutions that were acceptable to all staff.

In the leveraging phase at the end of the first cycle we collected the experiences of all involved and produced a process evaluation. One can argue that we did so as technical experts, but by exposing the intermediate results and the full report to working group meetings, by putting forward clear standpoints favouring certain stakeholder groups, and by using it in individual dialogues, this position could be challenged. As a result we applied the four paradigms much more consciously as shown in the case of introducing the estimation routines described above. This brings us to a more general discussion about the characteristics of the four paradigms and the overall usefulness of the framework, which will be subject of the final section of this article.

5. DISCUSSION

We will now discuss and compare the four paradigms as archetypes by emphasising the main differences in their methodological approaches concerning the process agents’ roles, their primary focus and interest and the applied data collection methods and investigation techniques as their basis for improvement work.

5.1 The Process Agents’ Roles

Technical experts are distant observers, they attempt to be neutral and objectively analyse an organisation and determine a maturity level. Participating facilitators are actors, they want to support the understanding of actual development problems. Political agents are primarily observers, who detect conflicts, but are actors when they become involved in problem solving. Therapists are actors, but when they collect data they attempt to be neutral.

Both roles have advantages and disadvantages. An actor is not limited in the way in which possible problems are identified. However, it can be difficult to generalise from such data and it can also be difficult to distinguish what is a result of the agent’s influence and what is an original insight from involved staff members. For neutral observers these problems do not exist, but their data is naturally imperfect as there are limitations of what they can see.
5.2 The Process Agents' Primary Focus

Technical experts have a focus on the chosen reference model and thus a mechanistic approach to software processes because they use the same process model to understand and design processes in many organisations. Technical experts are interested in deficiencies with regard to models and standards and aim at long term effects based on ‘hard’ empirical data. There is emphasis on ‘physical’ changes of standards, procedures, guidelines and change will often be implemented using an authoritative approach.

Facilitating participants have a distinctive focus on the actual processes being used and not on a predefined process model and thus have a more practice-based approach. The starting point is the organisation’s current situation and its existing processes, products, characteristics and objectives. Participating facilitators are interested in satisfying the interests of different stakeholder groups. They initiate and take part in working groups where staff are involved in the development of specific organisational solutions.

Political agents are interested in the structural and power-related conflicts and contradictions, which exist in organisations. They use dialectics to analyse the situation and try to influence the relationships between different, conflicting stakeholder groups. By creating disruption in the organisation they provide a starting point for improvement proposals concerning changes in the organisation’s structures, power relations, resources, and technical systems. In doing so they take a personal stand and support one side of the disputing parties. This allows both for model-based and individual organisational improvements.

Therapists have a psychological focus on individual staff members. They try to understand personal limitations and try to change and work with the individuals’ capabilities and to support their personal development as a basis for improvement. This approach tends to concentrate on influential individuals like decision makers and opinion leaders.

The advantage of the mechanistic approach is that a reference model provides a good overview of the whole problem area and allows comparisons to be made and facilitates rapid initiation of improvements. The disadvantages are that the model might not precisely fit the organisations’ needs and that a standardised solution might not actually improve the organisation’s processes. Uncritically adopting a model as a basis for improvements, thus results in a situation where the developed improvement proposals will not solve the actual problems and where staff might reject the suggestions as they might feel that the model and the accompanying actions have been forced upon them.

The advantage of the more practice-based approach is that the improvements will accommodate the needs of the organisation and can be
implemented early in the course of an improvement project as they focus on the problems the staff perceive in their daily work. In contrast to the mechanistic approach where the improvement strategy is almost provided up front before the problems have actually been articulated, this approach relies upon all stakeholders consensually agreeing upon what the improvement project should cover. Involvement reduces the risk of resistance, many people can influence the decision making process and rapid acceptance is possible. This requires significant competence of all those involved, otherwise the improvements will be spontaneous, uncoordinated and might only have a short-term effect. One might also work with the 'wrong' improvement because the developers' understanding about developing improvements might be insufficient. Finally, when the implementation of improvement actions is grounded in the agreement of all competing interests, very little might actually be improved because no agreement can be reached. Thus, this approach is a resource intensive process, especially if long term impacts are aimed for.

The advantage of the dialectics-based approach is that social and organisational barriers are identified. Solving these problems and changing structures often might be a prerequisite for more technical process changes. There is however a risk that producing too much turbulence might jeopardise any improvement action. As conflicts and contradictions are evolving during change it might be hard to develop long term improvement plans and to predict the effect of improvement proposals. There is also a risk that political agents take one side only - especially management's side. As they take sides and deal with confrontations, they might not always be very popular and major resistance against change might come from the side, which they have chosen not to support. Finally, applying dialectical analysis might lead to a limited view: one might see conflicts and power relationships in everything and only focus on contradictions and not on processual problems.

The advantage with this approach is that the process agents get close to the individual staff members' working life, which might make these individuals feel appreciated. They might subsequently aid and contribute to the process agents' acceptance in the organisation. Process agents will know staff better and individual improvement might be visible faster. These improvements might further increase the acceptance of the process agents and create a basis for further process improvements. The therapeutic approach however, demands considerable psychological competence and is a resource and time intensive process. It is therefore unrealistic to investigate the whole organisation and all the employees. There is a danger that an organisational overview is lost both generally and in the detail. It might also lead to a situation where many individual improvements are achieved, but only a few or none become commonly accepted. The therapeutic approach
is limited to individuals and personal development is expensive. However, improvements that accommodate the single individual are identified and these may in some cases profit both the individual’s development and the organisation.

5.3 Data Collection Method and Investigation Techniques

Technical experts build data collection primarily on quantitative methods, where model-based process improvement is based upon a rigid evaluation of an organisation in relation to the chosen reference model. Staff as informants and providers of data are treated as objects. Technical experts use questionnaires and surveys as investigation techniques to speedily and efficiently acquire a ‘limited’ amount of data from a large population. This data can then be benchmarked against the model using statistical methods to find compliance and deviation.

Participating facilitators use qualitative methods as they wish to gain a thorough understanding of a socially constructed work place. As the emphasis is upon sharing perceptions and achieving a consensus about improvements, all involved are seen as subjects. Participating facilitators will primarily utilise group interviews and discussions as they are interested in the exchange of opinions and in this way, different perspectives and arguments can be provoked and elucidated.

For political agents it is an explicit aim to objectify what has been brought to light subjectively. This can be done using a qualitative method to develop hypotheses and a quantitative method to subsequently verify them. Thus a combination of debates, interviews and questionnaires can be appropriately applied. Staff are informants in the pursuit to find one truth with the help of dialectical analysis.

Therapists use qualitative methods as they wish to develop insights from value-based attitudes. They search for individual and unique problems and barriers that restrict personal professional development. Data collection is not that important, but how the informants are treated is. Thus as therapists want to explore situations and issues in depth, they use individual interviews and unstructured conversations as means of data collection.

Both data collection methods and the investigation techniques have advantages and disadvantages. Quantitative methods deal with explanations, qualitative ones with understanding. Qualitative methods are close to the data source. They are based on subjective statements and they aim to capture the specific and unique, whereas quantitative methods focus on the objective, observable, and verifiable. Questionnaires have the advantages of making the investigation repeatable. However there is a danger of
misinterpretation and little or no possibility to go into depth. The major drawback of group interviews and discussions as a data collection method is that when no agreement can be reached or certain individuals dominate they can be ineffective. Finally individual sessions can be rather resource and time demanding.

6. CONCLUSIONS

In the research presented here, based on our practical experience, we reflect upon how process agents perform improvement projects, e.g. understand and change software processes. The reflection takes its starting point in the traditional, rational perspective, but shows how three other perspectives might contribute to process improvements. Examples from each of the different paradigms both individually and in combination have been used to explain the way process improvements can be stimulated.

The reflections on our project using the four paradigms and discussing their advantages and disadvantages have provided us with a better understanding of what we were doing. It helped us to recognise why the project was not a straight forward, rational process, despite the fact that it took place in the scope of the IDEAL model and the CMM. Utilising Morgan & Burrell's framework led us to deal with considerations, especially radical structuralistic and radical humanistic ones, which typically we would not have taken into account. After all, process agents are not supposed to participate in nor educated to deal with structural conflicts and to get close to people. This explains why we had to deal with what appeared to be anomalies with regard to the rational model that our work was originally based upon. However handling these not as problems and deviations, but as natural parts of software process improvement resulted in a successful project. This might be an argument for providing process agents with an enhanced education covering more than just technical knowledge. One that equips them with the necessary resources that allow for more than limited assessments and adjustments of the software processes they encounter during their work.

In summary, our work thus shows how process agents can use the paradigms more consciously in their improvement work by choosing the paradigm and its accompanying methods and techniques that accommodate and are appropriate for a given situation. The broader perspectives that have been presented here might therefore contribute to the wider diffusion and more successful application of software process improvement approaches.
7. REFERENCES


Diagnosing Diffusion Practices Within a Software Organization

Ingegerd Andersson¹ and Kerstin Nilsson²
¹Volvo Information Technology, Dept. 9714, HCIS, S-40508 Göteborg, SWEDEN
²The Viktoria Institute, Göteborg University, P.O. Box 620, S-40530 Göteborg, SWEDEN

Key words: Diffusion and adoption, software technology, diagnostic approach.

Abstract: New software development methods, tools, and techniques are being developed at an increasing pace. To take advantage of this new technology, organizations struggle to implement at a similar pace. There is an increasing awareness that the expected benefits are not being achieved. To remedy this situation, software organizations need to better understand the weaknesses of current diffusion practices.

This paper presents lessons from a large software organization in which the diffusion practices were diagnosed. The diagnosis relied on state-of-the-art theories and combined mapping techniques, workshops, and in-depth interviews to gain insights into current practices. The paper offers practical advice on how to diagnose diffusion practices within software organizations and contributes to the understanding of key issues in these practices.

1. INTRODUCTION

Software organizations spend a lot of effort on developing and introducing new software technologies. However, the expected results—in terms of improved software quality or increased productivity—are often not achieved (Huff 1992; Lyytinen and Hirschheim 1987; Pressman 1997; Weinberg 1997). On a more general level, numerous research efforts have explored issues related to diffusion and adoption of information technology (Cooper and Zmud 1990; Eason 1988; Kautz 1995; León 1995; Markus
1983; Mathiassen and Sørensen 1997; Mitroff and Linstone 1993; The Standish Group 1995). Thus, it is by now well-known in the research community that diffusion and adoption of IT is a complex task that requires attention and planning. However, this knowledge does not seem to reach practitioners within software organizations.

In order to improve any activity, one must establish knowledge about current practices. This knowledge is needed for developing useful improvement strategies that tackle existing problems and barriers and thereby lead to improved practices (Checkland and Scholes 1990; Humphrey 1990; Lanzara and Mathiassen 1985; Pressman 1997). In this paper, we offer practical advice on how to diagnose diffusion practices within software organizations and contribute to the understanding of the key issues in diffusion efforts.

Our findings are based on a diagnostic project conducted at Volvo IT AB, a large Swedish software organization with approximately 2,500 employees worldwide. One thousand of these employees are involved in systems development and maintenance. A department called “Common Skills and Methods” (CSM) has the responsibility to “direct, develop and maintain methodologies and tools used in the systems development process.” Each year a number of implementation projects are conducted within this department. The purposes of these projects, ranging from one man-month to six man-years, are to implement new methods, tools, and techniques. The diffusion practices applied in this department were the targets for the diagnosis performed.

We chose to use a case study approach for our research. This approach provided us with a means to gain a rich insight into the organizational phenomenon of diffusion practices (Yin 1994). Our case, the diagnostic project at Volvo IT AB, relied on state-of-the-art theories (Checkland and Scholes 1990; Cooper and Zmud 1990; Eason 1988; Huff 1992; Kautz 1995; León 1995; Markus 1983; Mathiassen and Sørensen 1997; The Standish Group 1995; Weinberg 1997) and combined mapping techniques (Checkland and Scholes 1990; Lanzara and Mathiassen 1985), workshops (Fredriksen et al. 1998), and in-depth interviews (Patton 1990) to gain insights into current practices. In order to categorize data and find essential relationships, a grounded theory approach was used in the workshop setting (Strauss and Corbin 1990). In conducting the project we drew on experiences from similar attempts in a Danish software organization (Fredriksen et al. 1998).

Our argument is structured as follows: Section 2 presents insights from the IT diffusion literature together with the diagnostic approaches that we applied in the software organization. We present the diagnostic process and

---

6 Official presentation is on the corporate intranet (http://violin.it.volvo.se/dept/it9710).
results in section 3. The findings regarding both diffusion issues and the diagnostic process are discussed in section 4. The paper ends with a conclusion in section 5.

2. DIAGNOSTIC FOUNDATION

Diagnosing a diffusion practice has two sides to it: the diagnostic result, which is a measure of some important characteristics of a diffusion effort, and the diagnostic techniques applied to unveil these characteristics. We consider each of these in turn.

2.1 Diagnostic result

In choosing diffusion characteristics to diagnose, we have utilized knowledge from research related to problems with diffusion and adoption of IT in general. A survey of this research shows that there are three major directions that can be identified: factor research, process research, and political research (Cooper and Zmud 1990). These three research areas have helped us identify key characteristics of the diffusion practices at Volvo IT. We will describe the diffusion characteristics of each research area and the measures we have used to diagnose these characteristics.

Factor research focuses on individual, organizational, and technological forces that enable diffusion effectiveness. These enablers are discussed in a number of papers (Cooper and Zmud 1990; Huff 1992; Kautz 1995; León 1995; Mathiassen and Sørensen 1997; McMaster and Vidgen 1995; The Standish Group 1995). As a means for diagnosing enablers we have chosen to evaluate six of the most frequently mentioned enablers from the papers above:

- Top management support for the diffusion effort ensures credibility and attention in the organization.
- A clear statement of underlying reasons and the intended goals sets the scope and focus for the diffusion effort and prevents inadequate expectations.
- Goals and implementation rationale must be communicated throughout the organization, and appropriate communication with stakeholders must be initiated and maintained.
- Skills development is a very important part of a diffusion effort.
- Monitoring and evaluation of goals makes it possible to control and direct the diffusion effort.
- Good IT design that fits organization structure and work patterns ensures a smooth diffusion and adoption process.
Process research focuses on social change activities and management of the diffusion process (Cooper and Zmud 1990; Eason 1988; Huff 1992; Weinberg 1997). The diagnosis of the diffusion process is based on classifications of the change model and the implementation strategy involved. To diagnose the change activities, we used Weinberg’s classification of change models (Weinberg 1997). His four models provide a means to measure the ability to control and direct the change process:

- The diffusion model is based on the theory that change more or less happens without interference. Change is unpredictable, both in time and space, and can degenerate on the way.
- The hole-in-the-floor model is based on the diffusion model but adds control. The idea is that if change is planned correctly, it will diffuse in a controlled manner. However, this model leaves out the human factor. It assumes that the receiver should understand why and how to use what is being diffused.
- The Newtonian model is based on Newton laws: To change in a certain direction, you must push in that direction, and the bigger the system and the faster you want the change, the harder you must push. This strategy works for a period of time, but then people probably get tired and less productive and make more mistakes. Thus, one problem is that the force can come back as a boomerang or it can go in another direction.
- The learning curve model is based on the assumption that people cannot respond instantly to change but need a certain time to learn and respond. The curve is affected by a number of psychological factors, such as relevant skill, motivation, and attitude. It suggests that projects manage the change with personal selection and training.

Management of the diffusion process is diagnosed using Eason’s classification of implementation strategies (Eason 1988). These strategies provide a means to measure the level of planning applied as well as the management of the actual implementation process.

- The Big Bang strategy means instant changeover. This is necessary sometimes, but it requires careful planning because everything from technology to organization has to work simultaneously. After the change, extra resources are needed, and organizational performance usually declines initially.
- The parallel running strategy means running the old system in parallel with the new. This strategy provides an insurance policy against failure, but there is inevitably a cost involved in handling two types of work simultaneously. A problem with this strategy is how and when to make
organizational changes, because it is impossible to maintain two parallel organizations at the same time.

- The phase introduction strategy means introducing the change slowly over a period of time. This can be achieved by introducing the change in phases or by addressing the organization one part at a time. The phase introduction strategy lets users get used to the change at a suitable pace. However, each group has the same capability to learn; thus, the last group requires the same attention as the first.

- The trials and dissemination strategy means that a small-scale implementation with one group is used to test the change before it is introduced on a wider basis. This strategy is based on the assumption that most problems will be solved after the trial period, and therefore, the implementation in the rest of the organization will go smoothly. Careful planning and participation of each new group in the implementation process is important, because lessons from the implementation in one group may not apply to the rest of the organization.

- The infrastructure and incremental application strategy is a mixture of strategies whose common themes are evolution and local, user-led design. Three infrastructure elements are necessary: technical support, implementation support, and organizational policies. The key is to balance between user freedom and organizational control.

*Political research,* finally, acknowledges the need to manage the different interests of diffusion stakeholders (Checkland and Scholes 1990; León 1995); (Markus 1983; Mitroff and Linstone 1993). Political interaction theory suggests that implementing new information systems often changes power positions in an organization. The ones who lose power are likely to resist the change. Identifying and predicting resistance will help prevent it or deal with it if it arises (Markus 1983). As a means of identifying diffusion stakeholders, their interests, and relations, we have used rich pictures, described in (Checkland and Scholes 1990).

### 2.2 Diagnostic techniques

The diagnosis was performed both at a general and a project level. The reason for this was to complement findings regarding common problems and patterns with findings from specific projects.

We used a workshop setting to diagnose at the general level. This is the same approach as was used in the Danish project (Fredriksen et al. 1998). The workshop was conducted as a structured brainstorm with a number of activities. First the participants were asked to identify problems related to diffusion and adoption. The problems were then categorized and condensed
to a manageable number. Finally, the identified problems were related to each other using a relation diagram (Fredriksen et al. 1998). A relation diagram shows how actions—or, as in our case, diffusion and adoption problems—relate to each other and which problems are the major cause of others.

At the project level, we used a combination of diagnostic techniques, including in-depth interviews (Patton 1990; Yin 1994), diagnostic maps, ecological maps (Lanzara and Mathiassen 1985), and rich pictures (Checkland and Scholes 1990).

Interviews provide a means to gain important insights into specific situations. In our case, we needed in-depth understanding about the whole process of specific projects, as perceived by the involved actors. The interview technique used was standardized open-ended. We chose this technique because the practitioners that we interviewed were very busy, and the open-ended interview technique is considered suitable when there is limited time available for interviews (Patton 1990).

Diagnostic maps were used to focus the interviews. These maps are a useful tool for locating and describing existing problems and dysfunctions in projects, as perceived by the involved actors (Lanzara and Mathiassen 1985). For each problem discovered, the following questions were asked: What happened? Why is it a problem? What are the consequences? What can be done? During the questioning, new problems can be discovered and broken down.

Ecological maps (Lanzara and Mathiassen 1985) describe how problems in a situation can be related to the surrounding environment. These maps summarize problems identified by diagnostic maps and relate them to either internal or external conditions.

Rich pictures, from Soft Systems Methodology, describe actors, relations, and conflicts, as perceived by the involved actors (Checkland and Scholes 1990).

3. Diagnostic Practice

In this section we will describe the diagnosis performed. First the diagnostic process and its context are described and then the diagnostic result is presented.
3.1 Diagnostic process

Volvo IT implements and maintains all IT solutions at Volvo. Volvo IT’s objectives are to create results for the business, with emphasis on speed of delivery, flexibility, responsiveness and accountability for results.

Department Common Skills and Methods (CSM) is a knowledge center within Volvo IT; there are approximately 120 employees in the department. CSM has the responsibility to support system development at Volvo IT with tools and knowledge. A Strategy and Coordination group within CSM directs several projects (15 man-years in total) with the main focus on diffusion and adoption of new methodologies and tools.

Preceding this study there was no documented knowledge of CSM’s ability to diffuse and implement new technologies. To find out how CSM managed diffusion and adoption, we arranged a workshop and conducted in-depth interviews with actors in specific diffusion projects.

The intention of the workshop was to get a general view of common problems in relation to diffusion and adoption of new technologies. Participants chosen had at least 2 years (some more than 20 years) of experience with diffusion and adoption projects in the organization. They represented parts of the organization involved in diffusion efforts conducted by CSM. Participants were chosen with the intention of getting views on current diffusion and adoption practices from different perspectives. Their occupations were: a SDSM manager, a SDSM strategist, two project leaders, a methodologist, a manager for a department introducing new techniques in the Web area, and a manager for a group which handles CSM’s R&D resources.
The workshop was organized as a discussion, where the participants were asked to consider the following scenario: “You are asked to diffuse and adopt a new technology or method from CSM into your organization. What are your concerns and objections?” Figure 1 describes the process where the open workshop resulted in a relation diagram. The problems identified during the discussion were categorized, condensed, and used as a basis for the relation diagram presented in figure 2. Each workshop participant then verified the relation diagram.
Additionally, we interviewed four implementation projects, where two were deployed in the organization and two were on their way to being deployed. When choosing the projects, we considered: how big they were, whether their product was implemented, and if their target group was Volvo IT globally or only sub-units of Volvo IT. We wanted the projects to be different in these aspects. All of the projects had project leaders from CSM. The selected projects were:
- Project 1: 65 person-months, delivered a model to Volvo IT Sweden.
- Project 2: 50 person-months, delivered a method to Volvo IT globally.
- Project 3: 15 person-months, delivered a method to Volvo IT Sweden.
- Project 4: 0.5 person-months, delivered a tool to Volvo IT globally.

For each project, we interviewed two persons: one provider and one receiver of the product. They were interviewed separately, and each interview lasted approximately one hour. Because of the limited time and budget available, all persons interviewed were from Sweden. One person conducted the interview and the other documented it. The interviews started with general questions in which the interviewee was asked to describe the product: it’s target group, when it was implemented, and how it was implemented. Then, the diagnostic mapping technique was used to identify five of the biggest problems experienced by the interviewee during the implementation phase (see appendix for an example).

Figure 2. Relation diagram, showing common diffusion issues
After the interviews the diagnostic maps were synthesized in an ecological map. The problem causes identified in these ecological maps were later categorized into general types of conditions to easier identify patterns (see appendix for an example). We also checked against the available project documentation to see if the causes and consequences were mentioned there.

To identify stakeholders and their conflicting interests, we used rich pictures. Both the diagnostic maps and the ecological map were used as inputs to the rich pictures. The diagnostic process is illustrated in figure 3.
3.2 Diagnostic result

The documentation from the diagnosis at the project level (diagnostic maps, ecological maps, and rich pictures) was structured using the three directions of diffusion and adoption research: factor, process, and political. Table 1 summarizes the results found for each area.

*Table 1. Factors and diffusion processes identified in the four projects*

<table>
<thead>
<tr>
<th>Factors</th>
<th>Project 1</th>
<th>Project 2</th>
<th>Project 3</th>
<th>Project 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top management support</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>A clear statement of underlying reasons and intended goals</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Goals and implementation rationale communicated</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Skills development</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Monitoring and evaluation of goals</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Good IT design</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Change model</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Diffusion model</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Hole-in-the-floor model</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Newtonian model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning curve model</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation strategy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Big bang strategy</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel running strategy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phase introduction strategy</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Trials and dissemination strategy</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Infrastructure and incremental strategy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

+/- Denotes whether the factor was present/planned at the time we diagnosed.
3.2.1 Project 1

Project 1 had top management support. The project’s goals were clearly defined, but there was no direct communication with stakeholders. Monitoring and evaluation of the project’s goals was not done. Skill development was not taken care of in the project; it was considered a post-diffusion issue. A two-day course was offered after the release, but it was not required and not many attended it. The implementation strategy for Project 1 was clearly a Big Bang with instant changeover, as the product was introduced everywhere in the organization on the same day. Unfortunately the product (a model) was not ready and the planned support organization was not fully established at the release date, causing a negative attitude in the organization. The change model was hole-in-the-floor; the project had no intention of controlling the diffusion after the release date; instead, that responsibility was placed on middle management. Conflicting interests and ambitions between top and middle management were found. Middle management did not see the benefits of the model, which was just considered as extra work for their employees, to whom they filtered the information. Usage of the model was mandatory, but as middle management did not prioritize the implementation, it was difficult to motivate any education.

3.2.2 Project 2

Top management support was present. The project’s goals and rationale were clear and communicated throughout the organization. Skills development was addressed, but there was a shortage of educated mentors in the beginning. Even so, the project received a lot of goodwill and positive feedback through its education program. Project 2 used both trial and dissemination strategy and phase introduction strategy, as they started with a few pilots and then chose to implement one project at a time in a controlled way. Each pilot had planned extra time for participants to learn and work with the new method, so the change model could be characterized as a learning curve model. These choices were based more on personal experiences with past diffusion and adoption efforts than on a conscious choice of change model. Conflicting interests crippled the implementation somewhat because some powerful customers refused the use of the model.
3.2.3 Project 3

Project 3 had top management support in word but no real support in terms of resources, so skill development suffered from not getting enough time and money. Goals and intentions were directly communicated to target groups, but there was no interaction on how to prioritize. They had made rough plans of how to follow up on the project and perform some kind of monitoring. The project representatives worked together with each target groups in the organization and tailored their individual implementations. This implies the use of a phase introduction strategy combined with a learning curve change model. The new routines were given low priority by affected groups' management, though, as they prioritized their daily operation. Conflicts existed, as the project's steering committee had different (lower) ambitions than the project manager. Additionally, the ISO certifying authority required a different time schedule than did Volvo IT management.

3.2.4 Project 4

Project 4's product was never implemented. They had no expressed management support and no goals were communicated or even defined. The product was recommended in company guidelines, but no resources or plan for implementation existed. The product was just expected to diffuse by itself; i.e., the diffusion change model was implicitly used. The product was recommended, but no resources were supplied to support the usage of it. When systems development projects found the product and looked at it, they often considered it too expensive and too complex for most of their needs and chose another unsupported product.

4. FINDINGS

The aim of our research was to offer practical advice on how to diagnose diffusion practices within software organizations and to contribute to the understanding of key issues in these practices. First, we discuss our findings regarding key diffusion issues in the diagnosed projects at Volvo IT. Thereafter, we discuss our experiences with the diagnostic techniques used.

In the diagnostic project we identified key diffusion issues at both a general and a project-specific level. The reason for this was to get a nuanced picture of current diffusion practices.
When investigating the different cases presented above and comparing them with the result from the workshop (figure 2), it is possible to find some common patterns regarding diffusion and adoption.

Management support is important at all involved levels. In the projects studied, management support varied substantially. This of course affected the attention given to the different efforts. However, the major problem related to management support that we found was not lack of support, but lack of clear prioritization between various efforts and conflicting goals between diffusion efforts and operational activities.

None of the projects investigated had tried to identify stakeholders or possible conflicting interests between stakeholders. Thus, there were no preparations for handling conflicts, and some conflicts were not even discovered.

There is a need to improve communication with all actors involved in and affected by a diffusion effort. The level of communication also varies between the cases, but there are examples of almost no communication and of communication through a third party. None of these communication approaches were perceived as successful. In order to control the content and direction of the communication, it is important to have direct interaction with involved actors and stakeholders. The lack of proper stakeholder analyses made it difficult to reach all involved actors.

The level of planning involved in each diffusion effort also varied. There are examples of quite sophisticated implementation planning, but there are also examples of the opposite. Personal experiences with past diffusion efforts seems to be of decisive importance for the level of planning. Thus, the choice of implementation strategy is seldom based on a conscious selection of the most suitable implementation approach in a given situation. We also found that lack of proper planning greatly affected the quality of any estimations regarding time and cost.

We can conclude that diffusion is not recognized as a practice of it’s own within department CSM. Consequently, there are no efforts to monitor or evaluate current diffusion efforts. Unless consistent evaluations are undertaken, it will be difficult to structure experiences and extract useful guidelines for future efforts.

In the diagnostic project we used a number of diagnostic techniques. Our experiences with these techniques are summarized on three levels: using the techniques; creating understanding; changing attitudes.

When diagnosing the different diffusion projects, we needed a means to quickly identify problems related to diffusion in department CSM. Diagnostic maps helped us structure the interviews and findings. They also made it easy to focus the interviews on diffusion problems. We did not find the ecological maps as useful as the diagnostic. One reason for this might be
that ecological maps are aimed at finding solutions to problems and our focus was to identify problems.

Conflicts played a central role in several of the diagnosed projects. Rich pictures provided us with a simple but powerful ability to identify involved actors and relations between them. The interviewees found rich pictures helpful in understanding the situation at hand.

We discovered that a workshop requires a lot of planned facilitation to be successful. Our workshop was organized as a structured brainstorm, and the discussion easily diverged. We as workshop facilitators should have been more active in keeping the discussion on track. Important preparations for a workshop such as ours involve: definition of the question(s) to discuss, a clear definition of the subject, an understanding of the expected results, and knowledge of how to structure these results.

Factor, process, and political research areas provided the theoretical framework needed to prepare the diagnosis, digest the material, and finally, present our findings. It was important to use language and measures that quickly created an understanding for the elements we wanted to emphasize. The theories also gave us knowledge of the huge complexity of diffusion and adoption and the importance of planning an implementation.

Our diagnosis was focused on problems experienced during diffusion efforts. This was a conscious choice partly based on the limited time available and partly on our aim to find out which areas of the current diffusion practice could be improved.

The interviewees and the workshop participants all had a positive attitude toward the focus on diagnosing diffusion and adoption. Among the persons that participated in the diagnoses, there was a common feeling that diffusion and adoption was neglected in CSM's projects. Our effort to diagnose diffusion efforts within CSM has brought up the issue of implementation as an important challenge. As a result, a new project will start in the autumn of 2000, with the intention to improve diffusion practices within CSM. Furthermore, it has been recognized that diffusion and adoption is a very neglected part of the system development process. This will also be addressed in the new project.

Returning to the theory of diffusion and adoption, our study has confirmed previous findings regarding key diffusion and adoption issues. To improve the theoretical framework, we might broaden our studies with knowledge management and organizational learning.
5. CONCLUSION

The diagnosis described in this paper covers three major areas of diffusion issues: factors, processes, and politics. Although all diffusion problems might not be revealed, our research shows that the diagnosis provides insight into important problems with current diffusion practices. Apart from this, the diagnosis also creates an awareness of diffusion issues within the diagnosed organization and provides a baseline for improvement efforts.

The major diffusion and adoption issues that we discovered during our diagnosis of Volvo IT are related to management support, lack of stakeholder analysis, and lack of knowledge about implementation and change strategies. All of these issues can be addressed by improving knowledge about diffusion and adoption and by adding diffusion and adoption as an important issue on the project agenda.

ACKNOWLEDGMENTS

We thank Susanne Tryde, Ann-Dorte Fladkjaer Nielsen, and Jan Pries-Heje for sharing their experiences from Dansk Data. Also, thanks to Kerstin Forsberg, Lars Mathiassen, and Dick Stenmark for constructive comments on this paper.

REFERENCES


Diagnosing Diffusion Practices Within a Software Organization 127

IFIP WG 8.6, working conference on the diffusion and adoption of information technology. Oslo, Norway: Chapman & Hall.


### Table 2. Diagnostic map

<table>
<thead>
<tr>
<th>Problem?</th>
<th>Why?</th>
<th>Consequence?</th>
<th>What can be done?</th>
</tr>
</thead>
<tbody>
<tr>
<td>The model was not ready for delivery when it was released. The review process, for example, was not clearly specified, and the various reviews were not synchronized. The information taught in courses was obsolete after one month.</td>
<td>A lot of work had to be redone in order to adjust to the new versions. People had to be educated twice because the model changed so much.</td>
<td>Required a lot of extra time, which delayed projects. The model got a bad reputation.</td>
<td>Pilots should be used to verify a model before it is released. A model should not be released before it is completed and everything is in place.</td>
</tr>
<tr>
<td>Not enough marketing of model mentors.</td>
<td>With the help of model mentors, the projects would have reduced the time and effort required.</td>
<td>Increased the time required to learn and use the model. Some projects didn’t find it worth the effort to use the model.</td>
<td>If there is limited support available, it is a bad idea to make a model mandatory for all projects. An iterative implementation with a lot of support is much better.</td>
</tr>
<tr>
<td>Unclear, sometimes conflicting, messages.</td>
<td>People were uncertain about what was true and what was not.</td>
<td>Required a lot of extra time. The model got a bad reputation.</td>
<td>See above.</td>
</tr>
<tr>
<td>Learning the model required a lot of initial effort because it was complicated. There was no overview of the model.</td>
<td>Time is money and projects have a limited budget.</td>
<td>The model was not used.</td>
<td>There should be some form of introduction and readers digest that gives a good overview of the model.</td>
</tr>
<tr>
<td>The model is designed to handle the most complex projects, and there is no light version available for small and simple projects.</td>
<td>It is impossible for one model to efficiently control both large, complex projects and small, simple ones.</td>
<td>The model is not suitable for small projects and thus is not used for these.</td>
<td>See above.</td>
</tr>
</tbody>
</table>
**Special requirement:** Delivery date was predetermined by management. Big Bang introduction was also predetermined.

**Management:**
- Not enough support from middle management.
- Descriptions of consequences were not communicated through the organization.
- Messages were filtered.
- Not enough resources were in place to support Big Bang.

**Project control:**
- Projects using the model had not calculated with the extra cost that comes with using a new model.

**Practice:**
- Insufficient planning and coordination in Project 1.
- The introduction was primarily focused on middle management, even if more stakeholders were identified.
- Insufficient coordination between tracks within Project 1.

**Product:**
- Inconsistent messages from Project 1 to its receivers.
- The product was not complete when delivered.
- Support was missing at introduction.
- Steering committee members lacked the education required to support a project manager.
- Project 1 did not manage to give an overall picture of the model.

**Motivation:**
- The model is a complex product that requires a lot of time and resources to get in to.
- Because of the extra cost, usage of the model can be hard to motivate, especially in small projects.

*Figure 4. Ecological map*
The Diffusion of Components

Richard Veryard

veryard projects

Abstract: This paper takes an ecological perspective on diffusion factors within the software component market. It analyses the characteristics of software components that are favourable to diffusion, and poses a radical critique of traditional notions of software requirements and software quality. It also suggests a strategic view of software components and other technological artefacts as evolutionary envelopes rather than fixed collections of properties.

1. INTRODUCTION

1.1 Diffusions

This paper is about (the study of) technology diffusing across a landscape. The landscape is a complex ecosystem, in which the technology we have chosen to study is competing for attention and resources with a range of other similar and diverse technologies. (Perhaps we should study not Diffusion but Diffusions – to emphasize the complexity and diversity of the diffusion phenomenon itself.) The ecosystem may involve a community of intelligent agents – which may be human or artificial, individual or collective – with a structure of relationships including delegation as well as mutual obligations and responsibilities.

We typically study the diffusions of a specific technological device or artefact – or perhaps a class of similar devices. In this paper I’m going to talk about software components. My reason for choosing to restrict myself to this class of artefact is that they present themselves as having certain properties that will help to simplify our discussion.
The fundamental notion of software componentry is the separation of a description of the services offered by a component from the description of the internal mechanisms (which software engineers sometimes call implementation) by which these services are delivered. (Note that the word “implementation” literally means that the job is finished. A software engineer may indeed think that the job is finished when the program code is written – and perhaps tested – but there are other stakeholders who regard this as only the beginning of a much larger and more difficult job.)

1.2 Components and Commodity

I shall start with an account of technology derived from Borgmann, which I characterize thus: All technology aspires to the status of commodity. As I see it, this means at least two related things: firstly that the technology is usable – and used – to the greatest possible extent; and secondly the technology is packaged (encapsulated) in some set of products and services. (The notion of commodity also has some specific implications in economics, which we won’t go into here.) In Borgmann’s account, the drive to greater commodity can be seen in terms of ever-increasing availability of some technologically mediated benefits: easy and safe, wherever and whenever you want it.

Within software engineering, the logic of commodity is found most strongly in the notion of component-based development. Software components are expected to be reusable – and this reuse is supposed to be a good way of achieving economies of scale and increased productivity in the development of software systems. The expectation of software reuse is what I call a design mandate – it represents a complex amalgam of aspiration, prediction, justification, imperative and quality judgement, as shown in Table 1.
Table 1. Elements of the Design Mandate

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aspiration</td>
<td>Software designers want their components to be widely used. This desire may be reinforced by extrinsic motivation, such as financial reward.</td>
</tr>
<tr>
<td>Prediction</td>
<td>Given certain conditions, certain levels of software reuse and productivity are expected.</td>
</tr>
<tr>
<td>Justification</td>
<td>Component-based software engineering, together with any infrastructure needed to enable the mandated level of software reuse, is cost-justified against the predicted productivity benefits.</td>
</tr>
<tr>
<td>Imperative</td>
<td>Designers are required to produce software components that have the requisite characteristics for reuse.</td>
</tr>
<tr>
<td>Quality judgement</td>
<td>A “good” component is one with the potential for wide reuse.</td>
</tr>
</tbody>
</table>

Design methodologies are typically based upon several such design mandates. The epistemological and sociological status of these mandates is far from straightforward – as I have argued elsewhere, the software engineering industry is awash with claims that are difficult if not impossible to prove convincingly [Veryard 2001].

Surely then, within this engineering discourse, there should be a keen desire to understand which characteristics of software components and other artefacts are likely to be associated with wide dissemination and use – particularly as there are many knowledge-based artefacts (such as methodologies) that claim to be predicated upon reuse. And yet there seems to be little general understanding of diffusion within the software engineering community. (There is some understanding, of course, but it is itself poorly diffused – for reasons that even the technology diffusion community has somehow failed to master.)

1.3 Software Success

Software engineering is a complicated game, with many players playing different roles. There are many stakeholders interested in the “success” of particular software artefacts within this game. In particular, many players wish to attach themselves to successful components, in one capacity or another. This entails a desire to predict and control software characteristics that might be related to success.

What is a successful component? What does success mean? One fairly obvious notion of success is in terms of diffusion and use – a successful component is one that is widely disseminated and used. This often brings financial and other rewards for the originators of the component – but of course success for the component doesn’t guarantee success for the
producer. A component may be given away in the hope of achieving some other advantage, or may be stolen by software pirates – the software may be on everyone’s computer but the producer is penniless. Exactly the same principle applies to viruses and worms – a successful virus is one that infects millions, regardless of any consequences to the producer.

1.4 Unit of Adoption

<table>
<thead>
<tr>
<th>Agent</th>
<th>Granularity of the adopting agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adopts</td>
<td>Granularity of the decision to adopt</td>
</tr>
<tr>
<td>Device</td>
<td>Granularity of the adopted device</td>
</tr>
</tbody>
</table>

Perhaps the event of greatest interest to the student of technology adoption is the adoption decision. This occurs when an actor (which may be a person or community, or an agent or artefact to which some responsibility has been delegated) decides to adopts a device. A manager with sufficient authority may make this decision on behalf of a large community of users.

In many situations, the adoption decision is a joint one. A central planner makes a recommendation, which individual users are encouraged to accept, perhaps by an alteration in the cost-benefit equation for the individual user. Perhaps the central planner negotiates a price reduction, or absorbs some elements of the total cost of ownership.

This adoption decision may also be tentative or contingent. A person may decide to adopt a device on trial, perhaps for use in a pilot project. Further adoption decisions may be held off until more information is available.

This decision is made on the basis of some view of the potential utility of the device for the user(s), as compared with the expected total cost of ownership. The decision may also be dependent on the degree of confidence of these estimates, together with an assessment of any relevant risks.

This supposedly rational decision is highly sensitive to the way the overall system is conceptualized, as well as the costs, benefits and risks that are deemed relevant to the decision. An observer that takes a different view of these things may be critical of the decision that is made, and may regard it as a manifestation of a defective rationality. In particular, when a potential adopter declines to adopt something, which someone else (such as the vendor) thought he ought to have adopted, this is often characterized as resistance.

Note also that an adoption may be involuntary – and this is related to the granularity issue. You accept an email or email service, or download some
software from the Internet – and find that you’ve also unwittingly accepted a virus. Or perhaps a colleague gets infected first, and then passes it around the company. There are thousands of software components on my computer – from cookies to DDL files – and many of them have been loaded automatically as a consequence of some other installation or interaction. I hope they’re all benign – but I can never be sure.

1.5 Artefacts and Agency

In this piece, I’m talking about technological artefacts, including systems and components, as if they possessed intention and value. This is a fairly common trope, and many readers will pass it without comment. However, some readers might find this manner of speaking anthropomorphic or animistic: surely things can’t have intentions – shouldn’t we be careful to attach intentions and values only to specific stakeholders – that is, people or communities of people?

Let me say straightaway that I’m not unsympathetic to this objection. In my consultancy practice, I frequently encounter floating statements of intention and value (or belief or risk or cost/benefit), and I often find that it helps to anchor them by attaching them to specific stakeholders. So it might appear that I’m contradicting this practice here by allowing artefacts to have intentions and values. Surely the intentions and values really belong to the system owners – whoever they are.

But there’s a problem here. Firstly, it isn’t always obvious who the system owners actually are – and even when it is, we shouldn’t be dependent on this identification for our analysis. After all, an artefact may continue to manifest certain patterns of intentionality, regardless of a formal transfer of ownership. Often there are many stakeholders, with conflicting and overlapping interests, and the behaviour of an artefact represents a complex balancing of these interests.

Furthermore, if we attribute intentionality to artefacts, we can then compare the intentionality of the artefact with the intentionality of its stakeholders. From a practitioner perspective, this is extremely useful. For example, we may be able to predict errant behaviour in unusual circumstances, without having to engineer or await test conditions.

Most importantly, we can talk about the artefacts in purely ecological terms, as if their owners were completely out of the picture. This is a very useful simplification – provided we don’t forget that it’s only a simplification and not the whole story. (The map is not the territory.)
2. ECOLOGICAL MODEL

In this section, I propose an ecological model for understanding the characteristics of successful software components. More details of this model can be found in my book [Veryard 2000].

For a software component to be viable, it needs to be simultaneously viable in two separate but connected “ecosystems”. It needs to be viable as a black box delivering some set of services to a community of users. And it needs to be viable as a white box, with some configuration of devices executing some set of mechanisms on some platform.

In the service ecosystem, it is services that are competing for survival. Service viability depends on three key principles.

Table 3. Principles of Service Viability

<table>
<thead>
<tr>
<th>Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pleasure Principle</td>
<td>Value comes from achieving an appropriate level/balance of excitement and attention. Foreground components gain value if they are interesting and attention-absorbing; background components gain value if they are routine and require little or no attention.</td>
</tr>
<tr>
<td>Connectivity Principle</td>
<td>Value comes from the number of other users of the same service or component, within some domain.</td>
</tr>
<tr>
<td>Availability Principle</td>
<td>Between two otherwise equivalent services, the more available service will usually win over the less available. Some aspects of availability are as follows (depending on the nature of the service): Global 24-hour access. Instant response. Any hardware and software platform. Available in Arabic, Chinese, English, Hindi, Russian and Spanish. Easy to use. Low entry cost. Good support. Minimum learning curve. High reliability. Safe and secure. Low risk.</td>
</tr>
</tbody>
</table>

In the device ecosystem, it is devices and mechanisms that are competing for survival. Device viability depends on four additional principles.
Table 4. Principles of Device Viability

<table>
<thead>
<tr>
<th>Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy Conservation Principle</td>
<td>Competitive survival depends on delivering the greatest quantity of service with the smallest amount of work. This is often called reuse; software reuse should be focused on achieving economies of scale in software, based on effective asset management and knowledge management. Energy conservation also entails an interest in the operational efficiency or performance of a component.</td>
</tr>
<tr>
<td>Consistency Principle</td>
<td>Getting the expected services (and their associated benefits) from a given configuration of devices. This in turn relies on an ability to predict and control the behaviour of components-in-use, including the emergent properties of large distributed systems. Covers reliability.</td>
</tr>
<tr>
<td>Flexibility Principle</td>
<td>The ability to easily substitute devices and reconfigure systems. Covers maintainability and portability.</td>
</tr>
<tr>
<td>Biodiversity Principle</td>
<td>The robustness, flexibility and evolution of the ecosystem depends on a reasonable heterogeneity of software and services.</td>
</tr>
</tbody>
</table>

Note that this set of seven principles covers and extends the quality characteristics of software products identified in ISO 9126: functionality, maintainability, efficiency, usability, reliability and portability – with the possible exception of functionality. (We’ll come back to that.)

If there are multiple ecosystems, where are the components? As a working hypothesis, I’m going to suggest the following answer. A successful component should have a place in each ecosystem. Just as a frog must be viable both in the pond and on the shore, so a successful component probably needs to be viable and meaningful in both ecosystems, and this in turn means that the component probably respects all seven ecological principles. Such a component is viable and meaningful, and is likely to survive and develop. I’m going to call this the component viability hypothesis. It leads to the following definition of component quality: A viable component is one that respects all seven ecological principles.

That doesn’t mean that every component – or even every successful component – must have these characteristics to the ultimate degree. But a component that fails to respect one of the principles is vulnerable to attack, and can be swept away by another component that is stronger in this characteristic.

The trend towards components is too recent to provide conclusive evidence for this hypothesis, but there are some early signs of its plausibility, as well as arguments from analogy.
Note that while some components may seem viable in isolation, other components may only be viable as a member of a kit or family or tribe of components. As in biology, the unit of viability may not be fixed. So sometimes we’ll apply the ecological principles to individual components, and sometimes we’ll apply them to groups of components. This may sound like an unnecessary complication, but it reflects what happens in real life – components are sometimes designed or evaluated in isolation, sometimes in groups.

Here’s a well-known example. One of the factors that made Apple Macintosh successful was the common look and feel of Macintosh applications, although these applications were developed by independent teams in lots of different software companies. This consistency is a property of the tribe, not just of a single application. Thus the viability of a single Macintosh application is bound up with the viability of the whole tribe. That’s ecology for you.

Here’s another example. In the early years of laptop computers, travellers with laptops faced enormous difficulties if they wanted to connect these computers into hotel telephone systems. Although many hotels have now greatly improved the services available to the business traveller, and the telephone systems themselves are much more reliable, an increasing number of businessmen now use their own mobile telephones rather than the hotel telephone for such purposes. A few hotels may invest huge amounts in making provision for business travellers, but this investment is wasted if the businessmen don’t bother using these services. And now that they have found a satisfactory alternative, they might not start using these services again until the majority of hotels offer them, and perhaps not even then.

3. IMPLICATIONS

3.1 Requirements

The biological approach to requirements engineering is radically different to the traditional approach, and is based on biological and ecological metaphors.
Table 5. Ecological approach to software requirements

- First we identify an **ecosystem**, which may contain both human users and existing artefacts.
- Then we identify **services** that would be meaningful and viable in this ecosystem.
- Then we procure **devices** that enable the release and delivery of these services into the ecosystem.

This may be contrasted with the traditional approach to software requirements, which I call solution-driven, which may be either specific to a single situation (Table 6) or generic across some domain (Table 7).

Table 6. Specific solution-driven approach to software requirements

- First you identify a group of users who need a software solution for an identified business problem.
- Then you define the requirements on the software system. (For example, this may be specified as a set of use-cases.) These requirements may be based on a model of the business process, and are negotiated with the users.
- Then you design the software system as a set of interacting components.

Of course, if you are trying to build generic components for multiple use, there may not be a specific business process to analyse, or even a specific software system to design. Furthermore, there may not be any specific users to negotiate requirements with. Undaunted by this, software engineers typically adopt the same approach but at a different level of abstraction. A **domain** is defined, which is a generic business process or generic area of automation. Many software artefacts are designed as generic solutions, including frameworks and platforms.
Table 7. Generic solution-driven approach to software requirements

- First you identify a group of domain experts, who are supposed to stand proxy for a class of potential users.
- Then you define the requirements for the domain, in collaboration with the domain experts.
- Then you design a generic kit of interacting components, which will be usable for any system or business process that satisfies the generic domain description.
- Then you assemble systems from these components that satisfy the specific needs of particular users within the target area.
- Real business components need to be provided with staff, resources and infrastructure.

The solution-driven approach seems to imply a division of labour: in the software industry, some engineers shall specialize in the creation of small lumps of functionality (called software components); while other engineers shall specialize in assembling these components to produce large lumps of functionality (known as software applications or systems).

The solution-driven approach assumes that it is meaningful to think about requirements in terms of a fixed lump of functionality or capability, delivered to a fixed community of users. In business, this is known as the business; in software, this is known as the software system or application. It also assumes that one person or team has design control over this lump. In business this is supposed to be the CEO and her direct reports; as for software, there are several possible job titles, including system architect.

The limitations of this approach emerge when we are faced with large open distributed dynamic networks of business and software. It is both a business imperative and a technological imperative for business organizations to connect their business processes into these networks. These networks lack central design authority or architectural control, and evolve organically. Overall functionality and structure may change unpredictably from one day to the next. Connecting to these networks raises a number of difficult management dilemmas, including control, security and stability.

Taking our cue from Kevin Kelly, these networks are “Out of Control”. Traditional engineering approaches are inadequate for operating effectively in this environment. As Kelly has shown, biological and ecological metaphors seem to have more relevance than engineering metaphors.
3.2 Quality

3.2.1 Viability versus Quality

The ecological model focused on viability rather than quality, although there is undoubtedly some relationship between these concepts. There is an enormous literature on software quality – both in terms of the desirable characteristics of software artefacts and in terms of the development processes that are supposed to guarantee (or at least promote) these characteristics. From a traditional quality management perspective, it might seem appropriate to define a “good” component as one possessing some set of measurable intrinsic characteristics, which are somehow correlated to some set of needs. After all, this correlation between characteristics and needs is crucial to the official ISO definition of quality. (Although this definition is an holistic one – ISO 8402 refers to the totality of characteristics of an entity – quality management practice often attempts to treat these characteristics partially and separately.)

But when we try to discuss the diffusions of components, such notions of quality or “goodness” get us into difficulties.

For a start, in an open distributed world, there is no single value system or global intentions against which the goodness of a component – or anything else for that matter – can be evaluated. There are many stakeholders, and many perspectives. Any fixed position on component quality is going to be arbitrary.

3.2.2 Ecology takes no moral position

When we talk about diffusion of software components in terms of quality and purpose, we find ourselves apparently forced to choose between competing notions of goodness, based in turn on competing (if implicit) notions of ethics or rationality. If we talk about viability instead, this allows us to side-step this choice – and we can then reason directly about the characteristics of software components that are correlated with diffusion, regardless of who thinks these components are “good” or “bad”, from whatever perspective.

3.3 Utility and Hedonics

The Roman architect Vitruvius, who lived at the time of Jesus Christ, defined quality as commodity, firmness and delight. Bill Gates has quoted this definition, and explained it as shown in Table 8.
In their study of the adoption of home computers, Viswanath Venkatesh and Sue Brown make a distinction between utility and hedonics, in order to account for a degree of subjectivity in the adoption decision. Hedonics seems to correspond roughly to the Vitruvius concept of Delight, while utility corresponds to the seemingly more objective notions of Firmness and Commodity.

In domestic purchases of computers, a person may be willing to admit that they have purchased a more expensive model simply for aesthetic reasons. Some cheap home computers are bulky and unattractive. We might therefore expect to find particular emphasis on utility factors among purchasers of cheaper models, while hedonic factors would be stronger among purchasers of the more expensive models.

When the same person is buying computers for his/her company, however, there may be a reluctance to admit the influence of hedonic factors. Instead, purchasers will explain the selection of more expensive models by claiming higher utility – even though sometimes these claims seem fairly thin or optimistic.

From the standpoint of a software producer (such as Mr Gates), it is important to understand and quantify the impact of all factors on the adoption and diffusion of software components, including the hedonic ones.

This brings us to a different notion of hedonics, which includes all aspects of quality, rather than being contrasted with utility. Economists have developed hedonic pricing models, to account for the costs and benefits of various aspects and characteristics of quality in the prices of goods and services. Hedonic pricing methods are also used by environmentalists for attaching value to public goods and ecological assets. [Freeman, 1993].

Hedonic pricing is a method for assessing the price-contribution of each quality characteristic, by analysing a class of similar products with differing quality characteristics. It is used, among other things, to adjust productivity and other macroeconomic data – since in markets where there is a constantly rising level of product quality (both input and output) it would otherwise be impossible to compare productivity figures over time. (This is of course particularly relevant in economic measurement of the IT industry.)

Diffusion theory demands something very similar to this. If we want to compare the diffusions of various components across some landscape, over
time, then it seems desirable to factor in the improvements in "quality" or other relevant characteristics that take place during our study. If I wait for six months before installing a new version of something, is this because (a) I’m waiting for the early bugs to be fixed, (b) I’m waiting until lots of people start sending me documents I cannot read without upgrading, or (c) I’m waiting until the price drops.

3.4 Resistance

One starting point for discussing resistance is that of a change agent, who has a set of change goals, and regards anything or anybody who gets in his/her way as a nuisance. Resistance is stupid and has to be overcome, using force, guile, patience or whatever other strengths and resources the change agent can access. This is all defined in terms of the change agent's goals.

At our Ambleside conference, Linda Levine offered a more balanced view of resistance – where it is rational to resist if something is either intrinsically flawed or not suitable for the intended purpose. Resistance can thus be interpreted as due critique. I see this interpretation as leading to a stratification of resistance: resistance at the first level might count as cooperation at the second level. Conversely, cooperation at the first level can sometimes be interpreted as resistance at the second level: mechanical compliance that avoids real learning.

3.5 Evolution

How do components improve over time? Many software producers seem to believe that software gets better by adding functionality. This often results in a component’s becoming baroque and more difficult to use. Sometimes there are hundreds of functions that nobody wants – adding greatly to the size and complexity of the component and reducing its reliability, efficiency and flexibility. Such a software component would be vulnerable to another rival component that would offer the essential functions more simply, cheaply and reliably. One extremely popular suite of office software has a very high market share mainly because of the convenience of connectivity – it has become a de facto standard for document exchange – but this advantage might quickly disappear if rival products were able to use the same document formats.

Often the adoption of a software component entails a judgement about the future of the component. A short-term tactical decision merely evaluates the component with the characteristics it currently has. But if I’m going to build a component into my systems, or into my life, I need to know more than its present state – I also need to have some sense of how it is likely to
evolve. Are lots of other people going to use it, what kind of people, and how will this influence the producers when issuing new versions – or perhaps abandoning the product altogether? Will it get more reliable as it matures, or will it get more complicated? Will it be made available on new platforms? These are strategic procurement issues, and depend crucially on the expected patterns of diffusion for the component. The component can be understood as an evolutionary envelope – a set of forking paths, leading to a range of possible future properties and configurations – and this is the true object of a strategic adoption decision.

4. CONCLUDING REMARKS

4.1 Towards a unified diffusion theory

4.1.1 Good and Bad Components

At present, there are two largely separate fields of diffusion theory. Security specialists study the diffusion of components that represent security threats – software viruses and worms, among other things. Meanwhile, diffusion theorists mostly study the diffusion of “respectable” and “well-behaved” technologies.

An ecological theory of diffusion doesn’t take a moral stance towards components. We should expect the same forces and patterns to manifest themselves, regardless of how we sort the components into “good” and “bad” ones. This represents a unification of two previously separate fields of diffusion theory.

Resistance takes on an entirely different aspect in these two fields. Within a security context, as in the human body, resistance implies an effective set of defences against penetration or attack. Within a change management context, resistance is often seen as a problem to be overcome. A unified theory of diffusion could lead to a unified theory of resistance. Practitioners could use this theory to intervene in a more balanced way in the patterns of resistance.

4.1.2 Objective and Subjective Value

The study of diffusion can also usefully draw from economics and environmental science. Hedonic pricing seems to offer a way of putting all factors relevant to adoption and diffusion – whether objective or subjective – into a unified framework of preference and value.
Hedonic pricing is relevant to an empirical test of the ecological model. It would be extremely interesting to measure the hedonic weight of the seven ecological principles – in other words, their relative importance for the rate and extent of diffusion. It would also be interesting to measure the hedonic weight of alternative principles – for example, to prove my hunch that functionality has comparatively little direct effect on diffusion. This would provide empirical validation of the ecological model as a whole.

Hedonic methods would also be relevant to a measured study of resistance, since we would then be able to differentiate more precisely between different aspects of resistance.

4.2 Further work

Practical research in this area to date has been limited in scope. I hope that future research will be able to provide some empirical verification of the model proposed in this paper, both to feed into theoretical work on diffusion, and also to provide much-needed support to practitioners of all kinds.

FURTHER READING

For loads of material about software componentry, go to the CBDi Forum website – www.cbdiforum.com.


Across the Divide: Two Organisations Form a Virtual Team and Codevelop a Product

L. Levine¹, G. Syzdek²
¹Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. Tel. (412) 268-3893, Fax (412) 268-5758, ll@sei.cmu.edu
²Red Sky Interactive, Houston, TX. Tel (713) 354-1541, Fax (713) 354-1600, gsyzdek@redsky.com (formerly with Computer Associates Inc.)

Key words: collaboration, collaborative technology, virtual teams

Abstract: This paper takes a socio-technical perspective and presents preliminary findings from a study of a partnership between two organisations—where a virtual team, made up of members from both organisations, came together to codevelop a product. The authors assess what is gained and what is lost in substituting technology for the traditional (same place, same time) working environment and share lessons learned about the use of collaborative technology and processes.

1. INTRODUCTION

More and more organisations are requiring team members to work together in cyberspace—whether with employees who are geographically dispersed or in partnership with other organisations—in order to accomplish a mutual goal. These collaborative working relationships are about people working together, about sharing information, and about leveraging assets. Organisations are setting collaborative technology in place to help teams work together while reducing travel, speeding review cycles, and lowering costs in order to leverage knowledge.

From a research perspective, teamwork and collaboration technologies have been separately investigated. Relatively few studies have been
conducted with emphasis on the teamworking aspects of a distributed working environment. We know a great deal about teams: about how members interact and team characteristics, including information on small group communication, position, roles, subgroups, influence, etc. (Henry & Hartzler, 1998; Lipnack & Stamps, 1997; Sibbet & Drexler, 1994; Wellins, Schaaf & Harper Shomo, 1994; Isgar, 1993). We also know quite a lot about the technical aspects of collaboration technology (i.e., tools and their functionality) that enable distributed working environments (Kock, 1999; Ackerman, 1996; Grudin et al, 1996; Bannon & Schmidt, 1989). Too little research has tackled the interplay between these topics, namely the social dimensions of distributed work and collaboration technology in use—what we call the socio-technical system.¹ Too often, studies fail to address the practical implications of collaborative activity. Meantime, many organisations are at a loss for proven practice in this new area. We intend to address both of these gaps—to take a socio-technical perspective on distributed work and to offer practical information for others, based on our collaboration experience.

In this paper, we present preliminary findings from a study of a partnership between two organisations—where a virtual team, made up of members from both organisations, came together to codevelop a product. This work product developed by the team is a defined process for technology analysis, adoption, and installation, and is briefly discussed in the next section. The discussion that follows is based upon our experience using collaboration technology during a two-year project. By taking a socio-technical perspective, we emphasise how team members from distributed working environments joined together to solve problems—including to what extent freely and easily available collaboration technology could efficiently support the work. Based on our experience, we assess what is gained and what is lost in substituting technology for the traditional (same place, same time) working environment. We share lessons learned and provide some of our proven practices for using collaboration technology. A more comprehensive report may be published at a later date. The current analysis is considered preliminary for several reasons:

- data analysis is not complete: (a) “collaboration forms” developed and used by project team members to capture data, through time, are not included (b) observers’ findings are not presented in detail
- review of the literature is incomplete
- full search for explanatory frameworks (on collaborative activity) for mapping against data is incomplete
- research method, whether experience report, action research (Baskerville & Wood-Harper, 1996), or qualitative reflexive/narrative study is not treated in depth.
Across the Divide: Two Organisations Form a Virtual Team

The sections that follow include the purpose and rationale for the study, project description (high level), discussion of models and frameworks for collaboration activity, preliminary findings, and conclusions.

2. PURPOSE AND RATIONALE

By collecting data on the collaboration experience, as we were engaged in the process itself, we gained insight into barriers and enablers associated with the team’s real world use of collaborative processes and technologies. The approach allowed us to concentrate on developing the product while not losing sight of the lessons associated with collaborative work. We had two objectives in mind:
1. to codify our findings about benefits and problems in using collaboration technology
2. to then distil “collaboration technique” information (procedural knowledge) from those findings, and to encapsulate that technique in the product we were building.

We achieved the first objective—to capture and document our learning about collaboration processes and technology—but the development schedule did not allow for the creation of a collaboration technique as part of the final product. However, much of what we learned about collaboration and teaming did find its way, directly and indirectly, into the product we were building. In direct fashion, these discoveries influenced our perspective and deliberate solution development. More indirectly, all products have cultural assumptions built in, and INTRo was no exception as an artefact “inscribed” with the social and political circumstances of its creation.

A brief explanation about the nature of our partnership is in order. The primary goal of our organisations’ working relationship was to leverage our respective assets to produce the IDEAL-Based New Technology Rollout process, INTRo: a defined process for technology analysis, adoption, and installation. The partnership between the organisations was established when two of its members recognised the benefit that could be achieved in working together. Both organisations had performed work in the software process improvement arena. The Software Engineering Institute (SEI) is a research and development centre, sponsored by the Department of Defense, with a mission to improve the state of the practice of software engineering. The SEI has developed an improvement model that offers an effective high-level approach to adopting processes, methods, and tools. Platinum technology inc. (Platinum) was a software vendor that, among 165 commercially sold products, developed a tool for project and process management, including a best practices process library. Platinum also had a process model that its
implementation managers used to deploy the project/process management tool at customer sites. (*Platinum technology, Inc.* was acquired by Computer Associates, Inc. in 1999. Henceforth, Computer Associates is referred to as CA.) Together, the organisations sought to integrate the strengths of both models and to test their applicability in new technical areas. The work drew upon SEI know-how on software process improvement and change management and Platinum’s technical expertise in architecting processes and practical experience rolling out new technologies in customer and partner organisations.

Both organisations also supported the secondary goal to codify the lessons learned about the use of collaborative technology. From the start, it was clear that INTRo had to be developed in a distributed environment, by team members in Pittsburgh and Houston, and with heavy reliance on collaboration technologies. Using more advanced forms of collaborative technologies was a relatively new experience for both organisations. The SEI had done research on supporting collaborative processes, but the focus had been on process formalisation and automation (Heineman et al., 1994; Christie et al., 1996). Platinum’s teams collaborated internally and with business partners. However, they usually did so asynchronously using the standard set of communication mechanisms of email, fax and telephone. Email was also used to support threaded discussion, review, and feedback.

We recognised that the joint effort would stretch the limits of ordinary collaboration. The literature on collaboration discusses information sharing and communication. However, the rich codevelopment we were envisioning, involving synchronous and asynchronous work, aimed at producing a high quality, fully integrated product would push the boundaries of conventional collaboration.³

Our work together would include planning, design, development, decision making, problem solving, project management, technical reviews, negotiations, and documentation production. For our newly created team, the use of collaborative technology to codevelop a product in real time was a pioneering endeavour for our organisations. Our codevelopment, done through distant boundaries, was one of shared creation: a creation resulting from group understanding that no one previously held or could have created through an individual effort.

### 2.1 Approach

We have noted that there is a lot of material available from the fields of Communication and Organisational Development on teaming and group dynamics. Increasingly, attention is also being paid to the culture of organisations (Schein, 1999; Schein, 1996; Constantine, 1993). In Europe,
especially in Scandinavia, more significant work has been done on socio-technical issues in computer supported cooperative work (CSCW). Too often, the research community has pushed out a host of studies, largely academic and more limited in nature, often incorporating "toy tasks" (Kraut, Miller & Siegel, 1996; Redmiles, 1993; Carstensen, 1997; Stebbins, 1993; Albrechtsen & Jacob, 1998). The challenge is two part: (1) to bring the socio-technical perspective to research and practice, using multiple methods of inquiry to understand technology in use (2) to conduct and disseminate studies that are real world, where organisations and institutions can straightforwardly understand the implications and lessons and clearly act upon the same. We characterise this as a whole systems approach in the spirit of soft systems and living, complex adaptive systems (Checkland, 1999a; Checkland, 1999b; Gharajedaghi, 1999; McMaster, 1996; Wheatley, 1999).

Why do so few research studies, in the areas above and in information technology, fail to see the importance of such integration? It is clear that the work people do is influenced by the systems they use or choose not to use; it’s also clear that systems design, for example participatory design, is influenced and benefits from information from system users. Technology maturation, including development and deployment, involves the mutual adaptation of technology and organisation (Leonard Barton, 1988a; Leonard Barton, 1988b). Nonetheless, integrative approaches are not common (McMaster, Vidgen & Wastell, 1998) nor do we often see an applied research effort that traces a real world project, developing a product.

The present inquiry strives to understand the dynamics at work in a complex collaboration and to offer practical information about some of our proven practices. Our project was not about pure experimental use, about just playing with "cool tools." The effort, from the start, had a strategic pragmatism about it—the stakes were high, the goals and expectations were aggressive, and the use of new technology was consistently in service of the work. We hope this paper contributes alongside much of today’s more theoretical work by providing practical examples of collaboration technology in use.

3. PROJECT DESCRIPTION

This paper operates at several logical levels concurrently, to consider (1) a socio-technical perspective on collaborative practice (2) our particular organisational and technical drivers and constraints (3) the task at hand, namely the codevelopment of INTRo (4) our findings with respect to the above levels.
We ask the reader’s patience with the presentation order for the next three sections—on project description, models and frameworks for collaboration, and on preliminary findings. Each of these sections might be presented first and used to frame the subsequent discussion. In addition, the subjects necessarily overlap. In essence, we are trying to succinctly describe what we have found on this project in terms of our collaborative work—which is a topic unto itself and one that touches all three sections.

In the project description and collaboration frameworks sections, we provide just enough contextual information to make the preliminary findings meaningful to the reader. The project description includes the timeline for the project and information on the composition of the virtual team. We also briefly comment on how we captured data on the collaboration process as it was occurring.

3.1 Project Start-Up

A process design workshop, January 1998, was held to kick-start the development process. The workshop allowed the opportunity for five core team members to meet each other and to discuss:
- principles upon which the INTRo process would be based
- scope of the process, roles and responsibilities
- major products
- overall structure of the process
- business case, risks
- schedule, process development process for INTRo

The workshop included an overview of the process development approach and tools (used at Platinum) which would provide the authoring environment. The workshop also allowed time to discuss the collaborative working arrangements: how the work would be done, collaboration processes and technology, infrastructure, and data capture on collaboration. We knew that in order for the team to successfully develop INTRo, we had to rely heavily on collaborative technologies. Team needs were noted, including the following:
- Develop the process in real time
- Edit data in shared applications in real time
- Share the flow of information among team members
- Build trust and camaraderie among virtual team members
- Have occasional face-to-face meetings with team members
3.2 Team Composition

Initially, in 1998, the core team consisted of two technical staff from SEI and two from Platinum. The extended team included contributors, reviewers, pilot participants, making up a total of five members. In 1999, the core team was reduced to one technical staff from SEI and one from Platinum. Extended team members continued to play a critical role in review and a greater role in pilot activity. A total of three members participated in 1999-2000.

3.3 Timeline

A high-level timeline of INTRo development, recording key activities appears in Figure 1. This timeline reflects major milestones in actual development, not from plans. In addition, twelve face-to-face meetings took place between Sept 1997 and October 1999. These meetings, which had a strong reinforcing effect, included technical interchange, alpha reviews, process overviews at the pilot site, walkthrough reviews, conference presentations, and requirements sessions (focused on change requests and enhancements).

It is worth noting that initial estimates for INTRo, related to scope, size, and schedule, projected a one-year effort. During the course of development, it became evident that INTRo was a larger and more complex product than originally envisioned. In addition, since we were developing process guidance which users would be implementing, we felt it would be important to pilot INTRo. We adjusted our plans. At the end of year one, (1998), we would produce a beta version, for trial or pilot use. The following year, we would deliver the final version 1.0, which would incorporate as many changes as possible resulting from pilot feedback, not simply comments from walkthroughs and reviews.
3.4 Data Capture

During project start-up, discussion of collaboration was key: how would the team work together to codevelop INTRo and to codify lessons on collaboration and virtual teams? Team members agreed to

- capture data through the use of a “collaboration form,” recognising that these forms would evolve according to need
- invite observers to virtual meetings, for several sessions. This would allow multiple perspectives on the collaboration activities. Observers would be asked to record and share their observations. (One observer was an expert in Instructional Design; a second observer specialised in group dynamics, teaming, and change management.)
- document and publish findings in white papers, technical reports, and articles
- based on experience and lessons, attempt to build a “collaboration technique” to be encapsulated in INTRo

Preliminary decisions were made about the different technologies needed to support asynchronous and synchronous work. As a result, examples of the asynchronous technologies that project members selected and used included email with/without attachments, voice mail, and fax. Typical synchronous technology chosen and used included telephone, videoconference, and a desktop conference tool. Real-time meetings involved multi-party interactive sessions (easily available at both organisations’ desktop workstations) where team members communicated...
Across the Divide: Two Organisations Form a Virtual Team

with each other directly, using Microsoft’s NetMeeting. This conferencing tool allowed participation in cooperative interactive visualisation and display of data. The team was able to get significant mileage out of NetMeeting. This single application enabled sharing and collaboration with a host of applications, in particular, the process authoring tool being used to define INTRo, file transfer, and whiteboard. Further elaboration is found in the sections on collaboration models and preliminary findings.

4. MODELS AND FRAMEWORKS FOR COLLABORATION ACTIVITY

The framework below identifies different types of collaboration, in terms and categories and distinctions commonly used in the computer supported collaborative work (CSCW) literature. Technologies that support collaboration are indicated—according to whether the work is performed by individuals at the same time (synchronously) or at different times (asynchronously). In addition, distinctions are drawn between activities that facilitate communication and information sharing. Along with this labelling comes some inaccuracy; it is easy to see places of overlap. For example, communication is an essential ingredient in information sharing, even though we may want to set information sharing apart since it appears more interactive in nature, suggesting dialogue or back and forth exchange. Types of collaboration and mechanisms used by the team are indicated with a check mark (✓) in Table 1.

We believe that communication and information sharing fall short as labels to account for the nature of our collaboration experience, which involved significant joint activity. Many of the technologies used for information sharing were also used to support real-time codevelopment. This raises the question of whether existing frameworks are sufficient to delineate the full range of activities that people engage in when they work together. To better capture our experience, we would add “codevelopment” as a category. Even then, we would require additional models or descriptions to characterise the richness of our cooperative work—qualitative issues, such as effectiveness and performance; and in terms of what worked well, how we stitched things together, and why we did what we did. Such concerns are critical for deeper understanding of collaborative practice involving the use of technology.

A collaborative working environment is about people working together, about sharing information, and making decisions based on knowledge transfer and dialogue. In the description of our collaborative arrangement, we show how team members from distributed working environments joined
together to solve problems and to what extent the collaboration could be supported through computer technology.

Table 1. Types of Collaborations and Mechanisms
Adapted from Grudin et al, 1996.

<table>
<thead>
<tr>
<th>COLLABORATION TYPE</th>
<th>COMMUNICATION TYPE</th>
<th>INFORMATION SHARING</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASYNCHRONOUS</td>
<td>Email + attachments ✓</td>
<td>Discussion groups</td>
</tr>
<tr>
<td></td>
<td>Voicemail ✓</td>
<td>Group Editing ✓</td>
</tr>
<tr>
<td></td>
<td>Videomail</td>
<td>(shared software revision control)</td>
</tr>
<tr>
<td>SYNCHRONOUS (REAL-TIME)</td>
<td>Chat, ✓</td>
<td>Application Sharing</td>
</tr>
<tr>
<td></td>
<td>Telephone, ✓</td>
<td>And</td>
</tr>
<tr>
<td></td>
<td>Video conference ✓</td>
<td>Applications that Share Co-development</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(NetMeeting: Share Applications &amp; Collaborate, File Transfer, and occasional use of whiteboard; applications shared include Word, PE, PowerPoint)</td>
</tr>
</tbody>
</table>

Table 2 further illustrates the time and place distinctions for synchronous and asynchronous work.

Table 2. Time/Place Distinctions and Example Mechanisms

<table>
<thead>
<tr>
<th>Time</th>
<th>Place</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same</td>
<td>Same</td>
<td>Face to Face Meetings</td>
</tr>
<tr>
<td>Different</td>
<td>Different</td>
<td>Asynchronous: email, fax, voice mail</td>
</tr>
<tr>
<td>Same</td>
<td>Different</td>
<td>Virtual Meeting Environments, Synchronous, Telephone, video conferencing, chat</td>
</tr>
<tr>
<td>Different</td>
<td>Same</td>
<td>Serial Working</td>
</tr>
</tbody>
</table>

Sometimes, the terms *loosely coupled* and *tightly coupled* are also used to distinguish between synchronous or asynchronous work.

- **Loosely Coupled**
  - Asynchronous
  - Email, voice mail

- **Tightly Coupled**
  - Synchronous (real-time)
  - Telephone, audio/video conferencing, chat

Collaborative applications, involving loose coupling, are characterised by work on shared objects at different times. Conversely, team members that
use tightly coupled collaboration work at the same time as others on a common shared workspace (Mitchell & Graham, 1995). Responsiveness and notification are critical in tightly coupled collaborative systems. "Responsiveness refers to the immediacy of local system reaction to local user actions, whereas notification refers to the immediacy of remote propagation of local user actions... When dealing with tightly coupled systems, users should be notified of other users' actions with as little delay as possible" (Mitchell & Graham, 1995). Tightly coupled applications are essential with real-time codevelopment activity. When NetMeeting, a tightly coupled collaborative application, begins to break down by delaying responsiveness and notification, this hinders the execution of tasks, whether brainstorming, simple free flow of ideas, design, drafting, or composition.

5. PRELIMINARY FINDINGS

For purposes of structure, we have categorised the preliminary findings according to whether they relate to people, process, or technology issues.

5.1 People

This category includes information on the attributes of individuals and teams, as well as on organisational culture and professional subcultures. Some researchers treat people and organisational concerns separately—a distinction that may prove important in a subsequent full report. For the present, we use the single category of People.
5.1.1 Team Characteristics

The characteristics of the core team are noteworthy, especially the high levels of commitment, energy, focus, and communication. These characteristics were there almost from the beginning, which may account for how like-minded individuals identified a need and worked as self-starters, or to coin a phrase, "team-starters" to lobby their respective organisations, band together and create a joint effort. Some would describe this behaviour as that of a high performing team (Sibbet & Drexler, 1994). A reinforcing effect also operated. High energy fed team interaction and resulted in a "can do" attitude and strong productivity; in turn, the positive attitude and productivity generated increased energy and commitment.

Since this codevelopment experience is being told from the team's perspective, we recognise that readers may or may not choose to accept our characterisation of the team at face value. In self-reporting, directness and reliability are trade-off concerns.
One observer offered the following description of the team:
- like minded
- found middle ground
- jelled together
- small group
- same vision
- personalities of team members plays a role
- high performance team qualities
- self managing

Over time, there were changes in the composition of the team. For example, in the Spring of 1999, a core team member from Platinum left the project and the organisation. As a result, another CA member picked up additional responsibility. Initially, this adjustment worked well; however, six months later, the same team member was struggling to meet her various responsibilities and commitments. As a result, she had to cut back time spent on INTRo development.

Two SEI contributor/reviewers were involved in the first year of development (1998), and different reviewers came on board for the second year (1999). These changes were relatively smooth. In February 2000, two months before the final product freeze (targeted for April 2000), the CA lead member left the project and the organisation, handing off to a colleague from CA. This disruption was more significant, resulting in several months project delay. The new team member had deep knowledge of the authoring environment and tooling but limited understanding of the INTRo product. (Original target release for INTRo was September 2000; release is now projected for January 2001.) Overall, the team responded, accommodated and moved forward as best it was able.

It is worth noting that the duration of the codevelopment effort was longer than most collaborative projects. This allowed the team to cycle through the processes of forming, storming, norming, and performing, with the ability to continue to perform through time, leveraging team growth and learning. Conversely, as noted, the project endured two acquisitions—as LBMS was acquired by Platinum technology Inc. in 1998 and then, roughly one year later in 1999, as Platinum was acquired by CA. These acquisitions brought different degrees of instability to project work and were, at minimum distracting and at maximum more disturbing. For approximately 3-4 months in 1999, the uncertainty and cultural and political changes surrounding the second acquisition affected morale and significantly reduced productivity.
5.1.2 Mental Models

Team members constructed a shared mental model of the project. They had a common vision, values, preferences, and work styles. Some differences emerged—between the views and practices of the industry organisation (Platinum/CA) and the R&D institute (SEI). However, even while experiencing dissonance, team members placed value and store in their divergent perspectives, realising that the variation would balance and improve the work products and provide a richer team mix. Different perspectives were most evident about time to market and project management. The industry perspective favoured the earliest possible release of INTRo whereas the Institute perspective wanted to ensure that INTRo was accurate, fully tested, and complete. A compromise was struck. Thus, Platinum released the initial version of INTRo in its product suite (January 1999). SEI chose to consider this a beta version, and made it available for limited access and distribution on a password-protected site.

In terms of project management, Platinum’s approach was more detailed and rigorous than SEI’s. We speculated that the different approaches to project management reflected differences between academia and industry practice, and perhaps also where these two organisations fell along a technology development life cycle. The SEI conducts applied research on the maturation of new technology; Platinum does advanced development, building commercial products and offering consulting services.

In addition to team members explicitly and deliberately valuing difference, the parties shared an overall pragmatism. This pragmatic bent helped to navigate difference and still allowed the team to optimise according to the strengths and expertise of the parties. Through strategic thinking and an emphasis on alignment, reuse and dual use, the team sought to maximise the impact and benefit of its development efforts. Planning and designing for data capture on the team’s collaboration experience, during the course of codevelopment, is an apt example.

5.1.3 Sponsorship

At SEI, sponsorship of the project was sustained throughout, although the effort had some of the attributes of a small skunk works or outlaw project, directly supported (sponsored and championed) by a senior manager. On the industry side, for several months during the acquisition (of Platinum by CA in 1999) support waned, when the project lost a key team member who was also a champion. As a result, industry-side sponsorship was fragile and undetermined for a time and needed to be rebuilt. Gradually, support was re-established when the new lab manager came on board at CA. This manager
proved a strong advocate and supporter. This was evident even through the last set of staff changes at CA.

5.2 Processes

5.2.1 Establishing Trust

One issue that cuts across people and process issues is the matter of building trust. This is especially important for collaborative distributed work. Moreover, trust involves levels—being able to trust another to be a good and sincere individual is not the same as being able to trust another’s project management skills. The trust that is built on a virtual team is based on a growing sense of the others’ experience in the work/subject area, credibility, as well as the ability to make sound judgements. Working at a distance, without much history or direct personal contact, one wonders about collaborators in these ways: Is the message logically sound? Does the information flow? Did the experience she was describing seem sensible or correct, like the right thing to be doing in the situation? Can I relate? If the answers to such questions are reassuring, gradually, through time, trust can be built.

On our virtual team, trust was built early on and for the most part remained high. Only during the tensest times, was trust in question. This happened twice—first, when Platinum was acquired by CA and the project lost a member/champion on the Platinum side in Spring 1999, and second, when the CA lead member left the project and the organisation, in February 2000, two months before project freeze. On these occasions, team members struggled with balancing the needs of the project with their needs as individuals.

Acquisitions are notorious for raising political and cultural instability and our project was no exception. The effort proceeded through two acquisitions—perhaps in itself a badge of success. At acquisition and merger, matters of individual and organisational trust blur. During the first acquisition (of LBMS by Platinum in 1998), there was a relatively short period of uncertainty. Platinum managers arrived on the LBMS scene early and worked quickly to characterise the company’s future direction, integration, and transition. During the second acquisition (of Platinum by CA in 1999), there was more uncertainty for a longer period of time. Integration took longer, most likely because Platinum was only one of the many companies that CA acquired over a short period of time. The entire team was unsure about what to make of CA and about future levels of interest and support for our collaborative project, although this was clearly
more pressing for Platinum/CA staff. At this time, individuals were wary about trusting their employers and what lay ahead. We have already seen that sponsorship was gradually re-established when the new lab manager came on board at CA. This went a far distance in rebuilding trust at the organisational level.

5.2.2 Maintaining Contact

Researchers maintain that working relationships require some amount of direct personal contact. We felt this to be true and sought face to face meetings at intervals for this reinforcing effect. As noted, twelve face-to-face meetings took place between Sept 1997 and December 1999, with both parties travelling roughly the same amount of the time. For a few meetings, for example a visit to the pilot site or to a conference, not all team members were present. However, face to face meetings were designed to combine a working session with another engagement, whether it was a technical interchange, review (alpha, walkthrough), requirements session, visit to pilot site, or conference presentation. Only the initial planning visit in November 1997 and the Process Design Workshop, which launched the project, served a single purpose.

5.2.3 Decision Making

Decision making processes should not be seen in isolation from the characteristics previously identified: the team’s high-performing nature, its common purpose and shared mental models, strong commitment and high levels of trust. No single decision-maker or leader dominated. Rather, because the team as a whole valued the different orientations, perspectives and experiences of its parties, members typically deferred to whichever party had greater expertise with respect to an issue. With project planning and tracking, and process architecture, Platinum/CA provided more steering, whereas the SEI exerted greater influence on research direction, scope, and pilot methods. Open communication, discussion, and negotiation preceded all decision making.

The team never experienced differences of opinion that derived from powerplays, moves for territory or from the management of a group where mistrust leads to fearfulness about grabs for power. In the use of NetMeeting, where power and control issues can surface, as the person sharing the application tends to dominate the virtual meeting, individuals led discussion without any attempts to control the interaction.
5.2.4 Project Management

When managed as a project, codevelopment has higher complexity in a distributed work environment and higher overhead. A fair question we might ask is: how is project management different between conventional and virtual teams? Certainly a key issue concerns trust and the need for new thinking about the role of oversight (Handy, 1995). Virtual teams are much more likely to be successful if they are able to function as autonomous, adaptive, and self-organising entities (Davenport & Pearlson, 1998).

If we were discussing conventional project management, we would likely consider meeting management under the Process category. However, since our meeting management, with a virtual team, was so tightly connected to the use of technology, we cover this topic in the final section on Technology.

5.3 Technology

5.3.1 NetMeeting as NetWorking

In many ways, NetMeeting is a misnomer for more intense NetWorking. NetMeetings, in our experience, were far more demanding than conventional meetings. These meetings were extremely intense, without much downtime, with participants closely focused on the computer screen. The kind of tuning out that one might do in a conventional meeting is difficult. Rather, participants listen hard to what is said and for additional cues and tone. This was especially evident in the first 6-9 months of the collaboration and eased, somewhat, after that time.

5.3.2 “Sensory Distortion”

One of our observers pointed out that sensory distortion was occurring. (This may be so even when video is in use since the quality is often poor.) In face to face meetings, when the conversation quiets, participants have visual cues to explain what is going on. Someone may be reading or writing on the chalkboard. Facial expressions, tone of voice, and body languages, all of which add depth to the perception in communication, are reduced or lost with NetMeeting, telephone, or email. During our NetMeeting experiences, we were unable to see such cues and we became quite adept at sensing and at managing silence. We offered information about silences, keeping each other posted about what was happening, and we shared our reactions freely and directly. Trust is a critical ingredient in such exchange, especially as the opportunities for misunderstanding may be greater in virtual meetings.
Somers, Rudman & Stevens (1997) corroborate our experience. They observe: 
“[m]eeting protocols emerge with collaborative systems. For example knowing who is present in the meeting, knowing who is controlling what application, knowing what everyone (especially the leader) is doing at any point in time, and making sure that all participants are looking at the same thing. Users rely heavily on the audio channel for this co-ordination and thus spend a significant amount of meeting time performing “co-ordination overhead” rather than the actual business of the meeting.”

5.3.3 Use of NetMeeting

The team’s use of NetMeeting extended to include:
- Project status meetings
- Demonstrations
- Reviews
- Codevelopment/Production

Project Meetings. These meetings allowed team members to report status related to the development schedule, progress on the work breakdown structure for INTRo, and other project issues. These sessions were extremely focused and demanding. NetMeetings generally lasted about 1.5 hours and were well structured, following agendas that had been created as part of the collaboration forms. These forms were developed to capture the meeting agenda, minutes, and to track problems/barriers identified with the use of specific technologies (most frequently NetMeeting in conjunction with application sharing).

Demonstrations. NetMeeting was used to give numerous demonstrations of the authoring tool environment and/or the current work on the INTRo to upper management and other interested parties of the organisations. Demonstrations were also given to market/communicate about the INTRo process to colleagues who might be interested in piloting INTRo after completion.

Reviews. The project team was attentive to process review checkpoints. Again, NetMeeting played an important role. Informal and formal reviews were held between team members and with upper management and external reviewers. During reviews, INTRo’s work breakdown and task descriptions were extensively re-examined, and changes were made online, in real-time. Often, the collaborate function was enabled, allowing edits to be made by remote team members.

Codevelopment/Production. Project deliverables such as project plans and schedules, and the design requirements document were reviewed and edited via NetMeeting and transferred to team members during the session through file transfer capability. Since marketing activities were an essential
Across the Divide: Two Organisations Form a Virtual Team

part of the development project, presentation slides and other marketing aides were also reviewed and easily edited online.

Example constraints that the team experienced at times, related to tooling and network performance, include:

- **Network performance.** There were times when we would lose complete connection during our session. This seemed to occur more frequently while sharing MS Word applications. There were times when the network was very slow at returning data back to the other party’s screen (screen refresh). Also, when the collaboration feature was turned on, the screen hesitated while the party was trying to add data. When network performance was extremely slow, frustration levels were high and system tolerance levels were low causing us to abandon the collaborative session. Network performance was generally better during earlier morning NetMeeting sessions (8:00 – 11:00 cst).

- **The tool used (NetMeeting) was equipped with audioconferencing capabilities to act as “Internet telephones”; however, the telephone on our desk did a better job. NetMeeting was not sufficient to support high quality audio conference. We used the telephone to bridge the audio connection. (This may be because the modem connection (28.8) is not sufficient.)

- **Technical difficulties such as software/network disconnects were disruptive and disturbed concentration. Real-time co-development in our collaborative environment was spent partially in tool learning.**

- **The model used in our working environment was one in which one user placed a call out to the others or answered a call from other meeting members. We did experience constraints such as secured intranets (firewalls) that did not allow entrance into the public directories, connections.**

**5.3.4 Meeting Management**

We have already alluded to the team’s need for project and meeting management practices. In part, this derived from the nature of the distributed work to be done by a virtual team and from the team’s interest in capturing its lessons on the use of collaboration processes and technology through observation, use of forms, etc. However, as a team, we also became increasingly aware of the critical need for advance planning and organisation in order to work productively for our periods of contact.

Mitchell & Graham (1995) observe that the “ways in which groups work together on a task can range from highly structured, such as in a board meeting, to very unstructured, such as in a brainstorming session between designers.” Our team worked together in a highly structured manner in
status/review meetings, and in more unstructured fashion for brainstorming sessions between developers. In structured sessions, roles were assigned: leader, observer, scribe, etc. In such situations, the passing of control and access to the shared workspace was planned in advance.

Preliminary work before a NetMeeting involved preparation and structuring the virtual collaborative session. The typical sequence of activities looked like this.

- Team members offer input, by email or discussion, for proposed agenda topics
- A team member (identified) reviews the previous collaboration form and action items, and deferred topics looking for additional agenda topics
- A team member (identified) develops a new collaboration form to include received inputs from other participants and deferred agenda topics from previous collaboration meeting
- Agenda is distributed in advance.

Building agendas in this manner created an increased level of participant involvement and assurance that a member’s issues would be a part of the agenda. Occasionally, during a meeting, the team needed additional time on a topic. We adjusted the agenda, deferring other topics to the next meeting date. Typically, the team worked hot issues and those of lesser importance were added as last items on the agenda.

We used the term “scripting” to describe how a discussion leader planned and prepared files to be shared before a NetMeeting began. After we learned how the technology worked, and that screen painting could become slow and cumbersome, we realised that jumping from application to application, and from screen to screen had a negative impact on what is seen, and how quickly, by meeting participants. After a few disjointed and confusing sessions, we settled on a scripting heuristic, encouraging discussion leaders to open files in advance to what it was that they wanted to share and walkthrough. This allowed for more seamless movement through the topics at hand.
5.3.5 Video conferencing

Picture Tel was briefly used during the course of our working relationship. Although tried only once, this tool was used to test/experience the video conferencing technology available at each site. The meeting was called to review presentation slides and finalise a conference preparation that was to be given by one of the team members from each site. Technical difficulties experienced during the video conference meeting included:
- voice transmission delay
- initial use instruction (since this was the first time using the technology, assistance was needed in connecting to each others' site)

It is important to emphasise that the use of the collaboration technology on this project was always in the service of the work. The team was always interested in hearing about different technologies that were available and that might help us downstream, but we were careful about learning curves and making sure we would get a strong return on whatever we were going to pick up. For example, the team discussed the use of Basic Support for Cooperative Work (BSCW), but it was never clear that we needed it or couldn’t get the value another way. We were aware that for a larger team, we might have really needed BSCW or some other repository based technology.

5.3.6 Security

In interorganisational collaboration efforts, it is important to adopt a more open approach to information sharing having concern for access first and then security where protection is needed at some levels of access. Security firewall prevents interorganisational collaboration. Most collaboration tools are designed to allow access between two parties in common web gateway with an explicit connection through directories. However, in our experience when a firewall security was installed at one site, access to the directories was not possible. The work around for us with using NetMeeting was to connect via tcp/ip address. Security issues surfaced at various points during the life of this project, especially since each organisational acquisition had implications for the technical architecture and infrastructure. More extended discussion of this topic is beyond the scope of this paper.
6. CONCLUSION

Clearly, the work people do is influenced by the systems they use or choose not to use, and systems design is influenced by, and benefits from, information and feedback from system users. Yet, as we have observed too few research studies, especially in the American tradition, in Organisational Development and Communication (on group dynamics, teams) and in Computer Supported Cooperative Work and Information Systems, take an integrative approach to socio-technical issues. We have posed a two-part challenge: (1) to work in the margins of these disciplines using multiple methods of inquiry to understand technology in use (2) to conduct and disseminate studies that are real world, so that organisations and institutions can straightforwardly understand the implications and lessons and purposefully act upon the same.

This was our concern as we set about to explore the dynamics at work in our complex collaboration and to offer practical information on some of our lessons and proven practices. We hope this paper contributes alongside much of today’s more theoretical work by providing real practical examples of supporting collaboration technology in use.

Our experience, adapting to the use of collaboration technology and processes, resembles some of the already documented experiences of companies adopting alternative work arrangements for distributed work. “Successful virtual offices require radical new approaches to evaluating, educating, organising, and informing team members.” Companies must develop the management approaches that make virtual offices effective; management must rethink “the design of their business processes, and they must examine their control, measurement, and evaluation techniques for these new processes.” Distant team members must be well connected with the rest of the business (Davenport & Pearlson, 1998, pp. 51, 60).

And there are old pitfalls. Too often, especially with tools, we see a premature inclination to jump to a technological solution without paying attention to basics. Development teams are often over eager to automate processes, which are not yet well defined or used in manual operations (Christie et al, 1996). “Doing” computer supported cooperative work or using collaboration technology is not guarantee that contributors are collaborating, in the best sense of the word, or working productively as a team. These tendencies reveal our wishful thinking that adding technological support will magically allow participants to leap frog over a host of requirements. Here, technology can be seen, naively, as a silver bullet, allowing one to side-step consideration of the primary and fundamental ingredients associated with effective work practice.
Working in a new mode or new environment may turn out to be substantially different than a current baseline operation. But the failure to account for what constitutes good practice as a starting point only guarantees compromised success at the next technological level of complexity. A fool with a tool is still a fool. No technology can compensate for bad practice or substitute for an understanding of fundamentals; however, integrating, experimenting with, and piloting new technologies in practice can help us co-evolve fundamentals and technologies. For these reasons, we underscore the importance of a socio-technical perspective and related knowledge in multiple disciplines and in local practice. Initially, one may focus on collaboration technology and thinking about systems and processes. But in the end, effective learning organizations (Argyris & Schon, 1996; Brown & Duguid, 1996; Brown & Gray, 1995; Lundberg, 1991; Nonaka & Takeuchi, 1995; Schein, 1997) must come to grips with good practice in teaming, education, sharing information and archiving lessons, and corporate memory—recording and analyzing decision making and related history—for recurring and problematic themes, and all in a manner that is coherent, yet streamlined and accessible.

ACKNOWLEDGEMENT

The authors thank John Goodenough, Dan Green, and the anonymous reviewers for insightful comments used to improve this paper, and Suzanne Couturiaux for editorial assistance.

NOTES

1 "A socio-technical system is a system composed of technical and social subsystems. An example for this is a factory or also a hospital where people are organized, e.g. in social systems like teams or departments, to do work for which they use technical systems like computers or x-ray machines." PRINCIPIA CYBERNETICA WEB, Web Dictionary of Cybernetics and Systems, Bernd Hornung's Glossary definitions http://pespmc1.vub.ac.be/ASC/Socio- syste.html

2 The initial discussion about joint work began between the SEI and LBMS Inc. in October 1997. LBMS was acquired by Platinum technology Inc. in 1998, and Platinum technology was acquired by Computer Associates in 1999. General references in this paper are made to Computer Associates, unless a specific asset or activity was associated with Platinum’s tools, approach, consulting model, or customer set.

3 Typically, when people talk about collaboration, the processes they have in mind involve some combination of division of labor and/or review and feedback.

REFERENCES


What's Wrong with the Diffusion of Innovation Theory?

The case of a complex and networked technology

Kalle Lyytinen¹ & Jan Damsgaard²

¹Department of Computer Science and Information Systems, University of Jyväskylä
²Department of Computer Science, Aalborg University

Key words: diffusion of innovation theory, complex and networked technologies, electronic data interchange, EDI.

Abstract: This paper examines the usefulness of the diffusion of innovation research in developing theoretical accounts of the adoption of complex and networked IT solutions. We contrast six conjectures underlying DOI research with field data obtained from the study of the diffusion of EDI. Our analysis shows that DOI based analyses miss some important facets in the diffusion of complex technologies. We suggest that complex IT solutions should be understood as socially constructed and learning intensive artifacts, which can be adopted for varying reasons within volatile diffusion arenas. Therefore DOI researchers should carefully recognize the complex, networked, and learning intensive features of technology; understand the role of institutional regimes, focus on process features (including histories) and key players in the diffusion arena, develop multi-layered theories that factor out mappings between different layers and locales, use multiple perspectives including political models, institutional models and theories of team behavior, and apply varying time scales while crafting accounts of what happened and why. In general the paper calls for a need to develop DOI theories at the site by using multiple levels of analysis.
1. INTRODUCTION

The slow, and often unexpectedly painful adoption of information technology (IT) innovations (Attewell 1992; Lyytinen 1991) has lead scholars and practioners to seek to understand, manage and predict its diffusion.

One popular account to explain and predict rates of IT innovation adoptions is diffusion of innovation theory (DOI) as propagated by Rogers (Rogers 1995). The DOI tradition draws upon rational theories of organizational life adopted from economics, sociology and communication theory. It develops predictive accounts of the diffusion phenomenon that supposedly helps technology implementors advance the diffusion of selected technologies. DOI theory has gained wide popularity in the IT field, for example Prescott and Conger (Prescott and Conger 1995) found over 70 IT articles published in IT outlets between 1984-1994 that relied on DOI theory.

Overall, the DOI tradition has sought to explain individual adoption decisions or intentions to adopt. These decisions concern well-defined innovations (like TV sets or the use of a pesticide among farmers) and the adoption population is relatively homogeneous and has well defined boundaries. A host of factors including the availability of information concerning technology (like relative advantage, compatibility etc), adopters’ properties (like past experiences), characteristics of the social system (like management support, social norms, availability of change agents), and the communication process (through which media, how often) explains the adoption decisions. Scholars of IT diffusion have been quick to apply the widespread DOI theory to IT but few have carefully analyzed whether it is justifiable to extend the DOI vehicle to explain the diffusion of IT innovations too.

This paper questions the usefulness and applicability of DOI to explain the diffusion of a complex, standard-based and networked information technology. For this purpose we extract six conjectures from DOI and contrast them with data of the diffusion of Electronic Data Interchange (EDI) in three social contexts – Hong Kong, Finland and Denmark. By contrasting

---

7 Similar critical voices have been raised recently against a too simplistic and fixed view of IT. For example, (Ciborra 1996) discusses “drifting technologies”, and warns about a too static view of technology. (Grudin 1988) shows how social factors are inherently crucial in understanding the success or failure of the use of groupware technologies - not alone their static features. And (Hanseth 1996) explains IT diffusion as simultaneousness shaped as an infrastructure.
theory and field study data we can analyze the usefulness of the DOI theory to explain actual observed diffusion behaviors. The conjectures also invite further research that can complement the shortcomings of DOI theory.

This paper proceeds as follows. First we discuss DOI models and their locus. In section three we present and describe EDI technology from a diffusion point of view. In section four we distill six conjectures from DOI theory and test them using a Popperian approach of refuting the conjectures by providing one or more counter examples for each. Finally we discuss the implications of our analysis and sketch a path for a way forward to establish better theoretical accounts of IT diffusion.

2. DIFFUSION OF COMPLEX AND NETWORKED TECHNOLOGIES: THE CASE OF EDI

Complex and networked technologies include electrical supply systems, chemical industries and transportation systems. These systems contain messy, complex problem solving elements. They are both socially constructed and society shaping (Hughes 1987). They include physical artifacts, and the organizations that use and manufacture them, but they also relevant legislative and regulative bodies and scientific communities. Alignment of multiple interests is required for social construction of the innovation’s significance, the negotiation of standards, and the legitimation of the acceptable uses of the innovation. These systems are difficult to control and manage due to their messy institutional character, broad scope and longevity.

IS research dealing with the diffusion of networked technologies covers e.g. personal computing (Heikkilä 1995), airline reservation systems (McKenny 1995), collaborative computing (Star and Ruhleder 1996), NII (King and Kraemer 1995) and EDI. EDI is the focus technology of this paper and the following points characterize EDI as complex standard-based and networked technology.
1. EDI is inter-organizational in nature;
2. EDI links electronically organizations thus requiring considerable alignment of organizational procedures and policies
3. EDI is a complex, innovative and abstract innovation that requires considerable skills and know-how to implement and operate (Webster 1995)
4. EDI relies on an advanced telecommunication infrastructure which creates a large set of dependencies with other components of the technological system
5. EDI implementations are often built on third party operated Value Added Networks or Internet Service Providers which complicate the promotion of EDI and create additional dependencies in the technological system

6. EDI is based on standards (Damsgaard and Truex 2000). Therefore EDI uses create a high degree of organizational interdependence (Hørlück 1994), and necessitates institutional regulation

7. EDI requires a considerable user mass to be efficiently deployed

From a diffusion of innovation viewpoint EDI has several features that characterize complex and networked technologies. First, points (1), (2), (3) and (4) imply that its adoption creates path dependencies with earlier innovations. Second, points (1) and (7) suggest that the decision to invest in EDI is not solely dependent on singular adopters, but on "herd" effects of having sufficiently many simultaneous adoption decisions. Third, points (1), (4) and (6) suggest that the success of EDI adoption does not solely depend on individual adopters' goals and desires, but as well on the effectiveness of broader institutional and regulatory regimes. These regimes can employ measures to reduce innovation ambiguity and uncertainty. Fourth, points (3) and (6) imply that due to EDI's complexity it has high learning barriers (Attewell 1992). Fifth, points (1), (2), (4) and (5) blur the distinction between technical and administrative innovations so much heralded in the DOI research (Damanpour and Evan 1984). Sixth, points (1) and (2) raise the issue of how the unit of analysis should be defined in the study of the diffusion process. For example, new forms of customer-supplier relationships can be implemented along all parts of the value system, which easily expands the analysis to industries, and even whole economies, or communities of traders (Wrigley, Wagenaar, and Clarke 1994).

3. SIX CONJECTURES OF DIFFUSION THEORY RECONSIDERED

Rogers (1995) defines DOI as the process "by which an innovation is communicated through certain channels over time among the members of the social system". A typical model consists of sequential adoption and implementation stages. These stages help predict innovation of diffusion over time and space (Lyytinen 1991). DOI explains diffusion rates by the characteristics of the innovation, and the surrounding social system (Wolfe 1994). Factors that have been found to influence diffusion rates include: adopter characteristics, the social network they belong to, the communication process, the characteristics of the promoters, and the
innovation attributes including triability, relative advantage, compatibility, observability, and complexity. Variations in research constructs are usually restricted to the choice of adopting units, and to the number of variables included in the model. The models are not very specific about the items of diffusion, and seldom question whether the studied technology makes a difference (Monteiro and Hanseth 1995; Prescott and Conger 1995; Wolfe 1994).

The key question we ask in this paper is the following: are DOI theory’s concerns in explaining an individual adopter’s behavior with respect to a static technological artifact (Mahajan, Muller, and Bass 1990; Rogers 1995) in a homogeneous population sufficient to understand EDI adoption? Will DOI theory based analysis of EDI diffusion miss some important facets? We suspect that this is the case based on our observations from three diffusion studies (Damsgaard 1996; Damsgaard 1997; Damsgaard and Lyytinen 1997; Damsgaard and Lyytinen 1998). To demonstrate we shall examine six conjectures\(^8\) that underlie the DOI theory and compare these conjectures with our field data findings. We thus follow a Popperian advice and seek to refute DOI theory predictions by using a counterexample (Popper 1968) thus questioning DOI theory’s power to explain the diffusion of networked and complex technologies.

The six conjectures of the DOI model can be summarized as follows (Mahajan, Muller, and Bass 1990; Premkumar, Ramamurthy, and Nilakanta 1994; Prescott and Conger 1995; Rogers 1995; Tornatzky and Klein 1982):

1. An innovation (technology) has separate, distinguishable and objective features, which are easily recognizable by interested parties.
2. The technology moves in a discrete package from an independent innovator to the adopter through a constant social "ether" called here a diffusion arena.
3. The adopter’s choice to adopt forms an atomic, isolated decision, which is shaped by the push and pull factors.
4. The decision to adopt follows a rational calculus that is based on observed technological characteristics, and other relevant information made available to the adopter through communication channels.
5. The diffusion process evolves through distinct stages, which are determined by the push and pull forces and are distinguishable by changes in the adoption rate.
6. Finally, the diffusion process has neither feedback, nor any “effective” history.

The conjectures are consolidated in table 1.

---

\(^8\) We call them conjectures as they are mostly informed guesses used to derive conclusions.
Table 1. DOI theory conjectures and supporting literature

<table>
<thead>
<tr>
<th>Conjecture</th>
<th>References in support</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Technologies are discrete packages developed by independent and neutral innovators; (Hai 1998; Premkumar, Ramamurthy, and Nilakanta 1994; Rogers 1995; Tornatzky and Klein 1982)</td>
</tr>
<tr>
<td>2</td>
<td>Technologies diffuse in a homogenous fixed social ether called diffusion arena, which is separate from the innovation locale; (Mahajan, Muller, and Bass 1990)</td>
</tr>
<tr>
<td>3</td>
<td>Diffusion rate is a function of push and pull forces (Thirtle and Ruttan 1987)</td>
</tr>
<tr>
<td>3.1</td>
<td>Push factors include features of technology, and channels of communication, (Mahajan, Muller, and Bass 1990; Rogers 1995)</td>
</tr>
<tr>
<td>3.2</td>
<td>Pull is determined by adopter's rational choices; (Rogers 1995)</td>
</tr>
<tr>
<td>4</td>
<td>Adoption decisions are dependent on available information, preference functions and adopter's properties; (Rogers 1995)</td>
</tr>
<tr>
<td>5</td>
<td>Diffusion traverses through distinct stages, which exhibit little or no feedback; and; (Nolan 1973; Nolan 1979; Rogers 1995)</td>
</tr>
<tr>
<td>6</td>
<td>Time scales are relatively short and the diffusion history is not important. (Rogers 1995)</td>
</tr>
</tbody>
</table>

3.1 Technologies are not discrete packages

DOI research associates an innovation with distinct and measurable features (Hai 1998; Premkumar, Ramamurthy, and Nilakanta 1994; Rogers 1995; Tornatzky and Klein 1982). With this sort of definition, several difficulties emerge. First, it is not clear whether the list is complete and covers all features that affect adopter's behavior. For example, why technical elegance or style does not appear in the lists though studies in the history of technology demonstrate the contrary (Hughes 1987). Second, why all technological innovations should be characterized with the same set of attributes? For example can EDI be characterized with the same set of attributes like a Television? Third, what roles play these different

---

9 (Tornatzky and Klein 1982) list the following ten attributes: 1) compatibility; 2) relative advantage; 3) complexity; 4) cost; 5) communicability; 6) divisibility; 7) profitability; 8) social approval; 9) triability; and 10) observability. Whereas (Premkumar, Ramamurthy, and Nilakanta 1994) used the following subset for studying EDI diffusion: 1) compatibility, 2) relative advantage, 3) costs, 4) communicability, while (Hai 1998) used another set of six attributes: 1) relative advantage, 2) compatibility, 3) complexity, 4) triability, 5) observability, and 6) risk.
characteristics at different stages of diffusion? For example, compatibility may mean different things for the late and early adopters. Fourth, the assumption ignores the socially constructed nature of large technological systems. All studies demonstrate that such innovations are socially constructed, learning intensive, complex and networked (Pinch and Bijker 1987).

Complex technological systems have "interpretive flexibility" i.e. their significance varies from one context to another and from one time point to another (Karsten 1995; Orlikowski and Gash 1993; Pinch and Bijker 1987). Consequently, groups, organizations, and industries construct the meaning of the technology differently. Local culture, economic structure and the supporting infrastructure (education system, government policies) shape these constructs. This observation was confirmed in our studies. The ideas about what EDI was and meant and what connotations it carried varied radically in different sites and affected the adoption decisions (Damsgaard 1996).

IT technologies are learning intensive in that resources have to be continuously poured into their maintenance and modification (Heikkilä 1995). This changes the innovation over time. For example, how an organization integrates its internal systems with EDI is not a simple task and demands continuous learning to align organizational processes and structures and technologies. This feature was also demonstrated by the long time spans required to make EDI fully operational. Furthermore, integrated technologies co-evolved and had to be transformed. In one EDI adoption process a shipping line initially pushed a container terminal to adopt EDI. This application, however, required intensive learning from both parties, and obligated the container terminal to wholly rethink its organizational processes. What started as an straight forward data link that carried simple announcements of shipping information was gradually transformed into a highly complex and integrated IT application (Damsgaard and Lyytinen 1997).

In all cases we studied the adoption was not a simple decision of how to exploit EDI as a stand-alone technical solution. Instead it formed a part of a complex interplay of multiple technological systems (IT applications, telecommunication services, standards), partners' communication tactics, backward compatibility with other technological systems\(^{10}\), demonstrated benefits, and power play. In EDI adoptions local power play and institutional facilitation were the most common features that were considered during all

\(^{10}\) And not compatibility with adopters' understanding and needs as in DOI research, (Rogers 1995)
adoption decisions. Thus, the herd effect rather than any specific technological characteristic (tangible or in-tangible) led to the adoption decision (Bouchard 1993; O'Callaghan and Turner 1995).

3.2 Technologies do not diffuse in a homogenous and fixed social ether

In the DOI theory interactions between technology suppliers and adopters are expected to happen in a relatively homogeneous space. For example, the Bass model expects to estimate three diffusion parameters\textsuperscript{11} similar to the diffusion of entropy in an ideal gas (Mahajan, Muller, and Bass 1990). The conjecture is that the technology diffuses in this ether through the influences of these three “forces”. With complex technologies like EDI, however, the diffusion arenas are neither fixed nor homogeneous. Instead, institutional arrangements, the business context and technological and economic constraints reshape these arenas. Therefore, in analyzing EDI diffusion we found it necessary to employ institutional concepts to dynamically draw the borders of the diffusion space to understand what the studied processes were like. The institutional perspective helps focus on institutional measures and regimes that are involved in defining the scope and mandate for the diffusion process. Potent institutional changes can radically affect the speed and course of any diffusion process by redrawing its boundaries, redefining involved entities and changing incentives. Consider for example the amazing diversity of diffusion behaviors we observed in retail sectors in Hong Kong, Denmark and Finland, though the technologies and the adoption rationales were similar (Damsgaard 1997; Damsgaard and Lyytinen 1997; Damsgaard and Lyytinen 1998). In Hong Kong EDI triggered institutional intervention, in Finland it caused collaboration and establishment of an institutional arrangement to support diffusion, while in Denmark EDI was launched as a weapon in the ongoing struggle between two large retail chains. To a large extent these differences were due to variations in the institutional scopes and mandates.

\textsuperscript{11} These are coefficient of external influence, the coefficient of internal influence and the market potential.
3.3 The diffusion rate is not solely a function of push and pull forces

The DOI theory integrates two supplementary modes of explanation: the supply-push and the demand-pull theories (King and others 1994; Zmud 1984). Supply-push theories reckon that specific features of the innovation cause the EDI diffusion like its functionality, or the standards that enable its use. EDI is thus portrayed as a technological fix for organizations' supply-chain problems (Thirtle and Ruttan 1987). The demand-pull theories explain EDI diffusion by a growing demand for organizational coordination. Organizations need to improve their internal operations, and change their market positions by applying technical knowledge (Bensaou 1996; Porter 1985). Several IS studies have considered both forces simultaneously (Bouchard 1993; Delhaye and Lobet-Maris 1995; Premkumar, Ramamurthy, and Nilakanta 1994). Unfortunately the predictive power of the theory has been low and the results confounding (Hai 1998; Prescott and Conger 1995). For example the variance explained using the DOI theory constructs has constantly remained below 40 per cent. Our studies confirm that these "forces" did not form necessary and sufficient conditions for an adoption. Instead many adoptions could be explained whether the adopting organizations followed power dominant or consensus-seeking strategies, and what type of history they had with their EDI partners (Damsgaard and Lyytinen 1997).

3.3.1 Push factors include features of technology, and channels of communication

The push forces frame the adoption decision as a rational choice problem between an old and a new technology. The main source of decision information is mass media and word of mouth i.e. different communication channels (Rogers 1995). Our data shows a different reality. The push for EDI did not happen through the mass media or peer networks. Instead EDI was pushed by powerful actors (gatekeepers) - e.g. hubs, industry associations, or the government (Damsgaard and Lyytinen 1997). These entities used symbolic or real measures to push the technology involving demonstrations of vested power and/or biased communications. In contrast, many organizations that were well informed of EDI did not adopt, or were not able to adopt EDI due to scarce resources, power structures or the lack of skills and competence.
3.3.2 Pull is determined by adopter's rationale choices

In DOI theory adopter follows the ethos of “homo economicus”: the choice is based on a careful analysis of the technology features. Yet, in real life EDI adoptions were not celebrations of rationality - organizations seldom followed what their rational analysis suggested. Few organizations performed any cost/benefit analyses of the adoption (Bjørn-Andersen and Krcmar 1995). A typical decision would be the following: a container terminal was pushed by its customer to implement EDI, though the terminal knew that it would not benefit from adopting. EDI increased work and complicated the technological base. Yet, the “decision” was made as there were no alternatives i.e. the adoption was obligatory within a certain time period, if the container terminal wanted to remain in business. Therefore organizations chose between the lesser evil of adopting EDI thus following the well-known slogan: “EDI” or “DIE” (Delhaye and Lobet-Maris 1995; Webster 1995).

3.4 Choices are not functions of available information, preference functions and adopter's properties

In the DOI theory, adoption decisions are functions of available information, preference functions, risk and the adopter's properties. In EDI adoptions the choice parameters, however, fluctuated over time and over diffusion arenas in ways, which could not be derived from DOI theory. Consider the case of attempting to establish a strategic EDI network in Hong Kong (Damsgaard and Lyytinen 1997). The network sought to change the information exchange patterns in a large portion of the sea cargo transportation sector and adjoining sectors in Hong Kong. Many actors did not support the creation of the network but once it was assembled several fence sitters were afraid of loosing important business opportunities if they did not join i.e. their strategy was not to maximize their benefits, but to avoid losses. The choice was not related to available information about the technology but to business strategy. Choice parameters were also quite different in other situations. The garment industry in Hong Kong openly announced that it was not doing anything in regard to EDI before the

---

12 These issues could only be observed by examining the broader institutional context and how it redrew the boundaries.
13 (Rogers 1995) observes at the same time that early adopters differ from late adopters along these properties (i.e. knowledge, skill and risk taking behavior).
territory-wide EDI initiative Tradelink was operational. Instead in Hong Kong’s retail sector the intervention of the article numbering association radically reshaped the diffusion arena. This has significantly lowered the entry barriers in the whole retail sector. All these illustrations assert the same fact: choice factors fluctuate over time and social spaces.

Sometimes adoption factors can be locally unique. Consider the following: the high rentals for retail premises formed the prime motivating factor for the association of retailers and the article numbering association to initiate EDI in Hong Kong (McKendrick 1993):

“The biggest single threat to the retail industry and hence the supply chain in Hong Kong is escalating rentals imposed/demanded by Hong Kong’s landlords. For some, if not most retail sectors, the answer is not that simple as passing these costs on to the customers in higher retail prices. The industry has to become more efficient: Retailers, Manufacturers, and Suppliers have to work together to take costs out of the supply chain”

3.5 Diffusion does not necessarily traverse through distinct stages, which exhibit no feedback

In the DOI theory, the diffusion curve is divided into stages (Nolan 1973; Nolan 1979; Rogers 1995). Our observations suggest that complex technologies will not diffuse in sequential stages. Many times it was not clear what these stages would mean in relation to the observed behavior. In some situations adoptions took place in dyadic relationships where it was difficult to see what the notion of an early adoption would mean. Sometimes adoptions were effected by moves in one industry or across industries, and all adopters were early innovators by Rogers’ terms though they did not share their characteristics. In some situations the adoptions sought to cover the whole trading community (what would early and late mean in this case?). There were also reversed processes where the innovation was dropped or its’ use retarded, so strong outside competitors could not take advantage of its presence. We had also contradictory behaviors: we had “laggards” which were more visionary in their uses of EDI than those who Rogers calls “innovators” 14.

We also observed also that stages could be layered: the initiation stage would last for 15 years for some diffusion contexts. At the same time

14 This behavior was explained many times as a strategic choice: resources that were poured into building the first EDI implementation were magnitudes larger and much more risky. Many companies were simply waiting to find out what technology and standard would “win” before making their decision. In this way the companies sought to lower risks, save resources, and enjoy network externalities.
the overall diffusion in other types of solutions had well gone beyond the early adoption. The stages were also embedded, i.e. one diffusion arena could turn into another one. Organizations could for example move from a dyadic adoption to an industry level adoption, and vice versa (i.e. bilateral optimizations of industry level adoptions). This could result in stepwise adoption curves (rather than sigmoid). This shows that the penetration level and the diffusion rate between countries, industry sectors and organizations are highly interdependent and not independent as assumed in the DOI theory.

We observed also feedback loops. The local history, information available from the earlier trials and the dynamics of the diffusion arena all affected the shape of the diffusion curve. The case in point here is Tradelink, (Damsgaard 1996), which carried throughout its history the stigma from its earlier failures. This in turn leads to continued failures and inability to move beyond the initiation stage.

...TradeLink, I think, is the second or third attempt in Hong Kong. Hong Kong has had a number of false starts with the same people involved. So every time they carry a sense of mistrust and disbelief from previous generations. So there are quite a few psychological barriers... (Industry representative, 1995, reported in (Damsgaard 1996))

3.6 Time scales are not necessarily short and the history of decisions is not unimportant

In DOI theory used time scales are normally relatively short and the mechanisms that drives the diffusion do not change over time. Time scales range from few months to some years, and once the important technological and organizational characteristics have been determined they remain stable over time, so that the diffusion process is relatively deterministic. Moreover the past decision history is not regarded important. EDI, however, exhibits path dependencies, because it forms an add-on to the existing technology base. Accordingly many diffusion behaviors had to be traced relatively long back into the history of the social context. Consider the Finnish experiences with EDI adoption (Damsgaard and Lyttinen 1998). Finns moved into EDI very early due to the long and well-established history of industry wide cooperation in uses of IT in several sectors (especially banking and retail). These sectors developed and adopted pre-UN/EDIFACT solutions for inter-organizational data interchange. This created a need to collaborate to establish and maintain national Finnish standards for those sectors. Now
What's Wrong with the Diffusion of Innovation Theory?

Finns have to decide what to do. Either to dissolve the national proprietary standard and migrate to EDIFACT thereby throwing away major investments and jeopardizing existing local networks of collaboration. Or to stay with the proprietary Finnish standard that does the job nicely, but for national communication only. Hence, in order to understand EDI diffusion in Finland, it is necessary to trace back historical events until mid 1970’s that led to the prevailing consensus seeking strategies but also to the dilemma of today.

Another obvious example of the power of past decisions is that of driving on the left side or the right side of the road (Kindleberger 1983). Most people would agree that choosing one side only is far superior (either side) than leaving it to mutual adjustment. However traditions, sunk costs as well as the investments necessary to change the existing infrastructure and habits, and the no doubt heated discussions involved in making a choice makes it a discussion no one will engage in. The same principle applies for complex and networked technologies.

The diffusion trajectory is also contingent on feedback mechanisms, which form a universal property of any diffusion process. Therefore diachronic analyses should form an integrated part of a diffusion study. Feedback systems can operate on different time scales with different innovations. This is nicely exemplified by the choice that Finnish companies face when adopting EDI. Should they choose the Finnish proprietary standard, which is extensively applied and simple to use? Or should they choose the UN/EDIFACT that is more complicated and constantly changing, but allows them to communicate internationally? (Damsgaard and Truex 2000) Their choices will again have wide impact on what the trade associations will recommend and what standard the industry as such will choose in the future (Arthur 1989).

4. DISCUSSION AND CONCLUSIONS

The DOI research has had a considerable positive impact on IS research. Our analysis points out, however, that it falls short of some theoretical constructs that help address how complex networked technologies can and will diffuse. Several basic premises of the DOI theory therefore need a careful reconsideration in the context of the networked and complex technologies. In particular, DOI theory does not offer adequate constructs to deal with

15 We agree with Prescott’s and Conger’s conclusions when they note “DOI theory appears to be more applicable to IT applications, which have intraorganizational locus of impact” (Prescott and Conger 1995).
collective adoption behaviors (including the critical role of standards, critical mass, network externalities, sunk costs, path dependence etc.). The DOI researchers should be careful in analyzing the impact of the nature and meaning of the technology, the role of institutional policies and regimes, the impact of the industrial policies and strategies, and the importance of the installed base and learning inertia. Due to the inattention to these features DOI models could not explain EDI adoptions. Instead, we observed that the diffusion "factors" had to be changed radically due to the complex and networked nature of the technology, i.e. by expanding the scope and time scale of the diffusion study.

The analysis leaves us with a "theoretical" gap between the current main stream and our field study findings. Generally, DOI researchers have traded simplicity and generalizability against accuracy by using simple metaphors of "forces" and "diffusion rates". Consequently, DOI models resemble physical models of thermodynamics. Instead, our studies taught us to trade both simplicity and generalizability against accuracy. Consequently the models were process based, contextual and non-deterministic, and could identify necessary conditions for an adoption (Downs and Mohr 1976; Markus and Robey 1988).

As a step forward it is necessary to consider the following issues while studying complex networked technologies:
- Seek to understand the local complex, networked, and learning intensive features of technology.
- Seek to understand the critical role of market making and institutional structures in shaping the diffusion arena.
- Focus on critical process features and all key players in the diffusion arena.
- Develop multi-layered theories of diffusion that factor out mappings between different layers and locales.
- Use alternative theoretical perspectives that help extend analysis beyond questions of efficient choice. Good candidates include political models, institutional models and theories of team behavior in conflict-cooperative games (Wolfe 1994).
- Recognize the need for varying time scales when seeking to account for what happened and why.
- Develop theories at the site and with multiple levels of analysis.

We believe that armed with such theoretical guidelines DOI researchers will have a higher likelihood of providing faithful accounts of the diffusion of complex and networked innovations.
ACKNOWLEDGEMENTS

The first author presented an earlier version of this paper at the IFIP 8.6 Working Conference in Oslo, October 1995 with the title "Diffusion theory: Complex technologies, institutional arenas and time: Notes of diffusion research". The paper has also been presented in Case Western Reserve University (February 1997) and City University of Hong Kong (September 1997). We are in particular thankful for comments obtained from Richard Boland, Fred Callopy, Michael Ginzberg, Robert Davison, Carsten Sørensen, Ivan Aaen and John L. King. Thanks go also to Georgia State University in which the first author worked as a G. E. Smith Visiting Professor while writing the article. This research was in part supported by the Danish Research Agency, grant number 9900102.

REFERENCES


What's Wrong with the Diffusion of Innovation Theory?


Influences of Sources of Communication on Adoption of a Communication Technology

William D. Stuart¹, Tracy Callaway Russo², Howard E. Sypher², Thelma E. Simons², Lisa K. Hallberg²
¹Longwood College
²University of Kansas

Abstract: This study examined the influence of sources of information on end users’ decision to adopt an innovation. The study used an on-line survey to collect data regarding respondents’ perceptions of structured implementation activities and other sources of influence on their reported adoption of Microsoft Outlook at a large, Midwestern university. The research questions were based on Rogers’ model of the diffusion of innovations, and the work of Fulk, Lewis and Seibold, and Weenig on the influences of information sources on adoption of innovations. Results showed that respondents who were exposed to information from informal channels and structured implementation activities (e.g., informational meetings conducted at the unit level) were significantly different from those who received no information through these channels. Perceptions of quantity or quality of information received through informal and official channels were not significantly correlated with adoption. The results indicate that the implementation of Outlook was not viewed as a major event in the life of the organization, and suggest that diffusion of technological innovations may be different from diffusion on non-technological innovations.

1. RATIONALE

The purpose of this study was to investigate the diffusion of an innovation within an organization. Specifically, the research focuses on the communication campaign developed to persuade administrators, faculty, staff, and students at a large Midwestern university to adopt a new communication technology. The technology investigated in this study is the groupware product, Microsoft Outlook. As groupware products offer their
users a coherent variety of features, adoption in the current study was operationalized as the sum of subjects' reported frequency of use scores for all the features available. The investigation examined the influence of sources of information, source credibility, and valence of information on end-users' decision to adopt the new technology. These predictor variables were based on the literature regarding diffusion of innovations generally, and the diffusion of technology, specifically.

Diffusion theory is a popular means of investigating the proliferation of new ideas, policies, or products. Rogers (1995) defines diffusion as "the process by which an innovation is communicated through certain channels over time among the members of a social system" (p. 5). Rogers reports that 5000 studies using diffusion theory had been conducted by 1994 covering topics from the diffusion of hybrid corn in Iowa to videotape recorders, to water purification in Africa. Diffusion of innovations in organizations presents a specific context for the application and testing of diffusion theory, a context with unique challenges including the effect of interacting levels of responsibility and decision making, organizational roles, and organizational culture (Bunz, 1998; Speicher, 1997). The current study investigates the diffusion of a new communication technology within an intraorganizational context.

Researchers have demonstrated the importance of examining technology in organizations, because technology has the capacity to change organizations in a variety of ways. This can occur as technology changes the nature of jobs (Iacono & Kling, 1986), or changes the form of organizations themselves (Allen & Hauptman, 1990; Dawson, Drinkwater, Gunson, & Atkins, 2000; Fulk & DeSanctis, 1995). In addition to changes in the nature of organizations and jobs within organizations, technology may affect the social networks (Contractor & Eisenberg, 1990; Feldman, 1994) and social processes (Poole & DeSanctis, 1992) in the organization. It is important to understand new communication technologies, and how they diffuse, because they have and will continue to have profound affects on organizational life.

The current study pays particular attention to communication in the present diffusion initiative because communication is the cornerstone of diffusion (Marjahan & Peterson, 1985). The individual level decision can be influenced by both formal communication and informal communication (Lewis & Seibold, 1993, 1996; Weenig, 1999). Attention is paid to the valence of information gathered through social networks in the organization (Weenig, 1999). Valence refers to the attitude expressed (i.e., positive or negative) toward the technology. This is important because some scholars argue that technology is socially constructed (Fulk, 1993). That is, when individuals convey information and attitudes about a technology, their
communication helps shape the attitudes and behaviors of other members of their social network, so valence of information would influence outcomes.

The literature concerning computer-mediated communication, the social construction of technology, and the diffusion of innovations suggests that when a technological innovation is presented to a potential adopter, information the individual receives from structured implementation activities can help provide general knowledge about the innovation and is the first step in the individual adoption decision process. The individual also is likely to receive information about the innovation from members of his or her social network, information that is likely to be influential in shaping the individual’s attitudes and behaviors toward the technology. In order to investigate the relationships between these potential sources of influence and users’ adoption of a communication technology, the following research questions are posed:

**RQ1:** What is the relationship between adoption of the innovation and receiving information through informal channels of communication and through the following structured implementation activities: official channels of communication (university and department officials, and/or publications), informal channels, initial informative presentation to the users, on-site introduction to installation, or installation consultant?

**RQ2:** What is the relationship between adoption of the innovation and the perceived credibility of sources of information?

**RQ3:** To what extent does adoption of the innovation vary if information received through informal channels is perceived as being positive or negative toward the innovation?

2. **METHOD**

The population for this study was individuals in faculty, administration, and support staff positions at a large Midwestern university who had Microsoft Outlook installed on their computers by staff members of Academic Computing Services at the university. The survey was distributed to approximately 1500 persons. This figure represents all the individuals who had the installation completed at the time the survey was distributed; that is, the survey was made available to 100% of the individuals who met the population parameters. A total of 509 responses were submitted. Out of these, 7 were completely blank, yielding a valid set of 502 completed surveys.

Individuals were invited to participate in the research process via an e-mail from the Outlook Project Coordinator at the university. The e-mail
message from the Coordinator described the nature and purpose of the study to potential respondents, and provided a link that respondents could click on to access the on-line survey.

The invitational messages/survey were distributed on Thursday, May 11, 2000. A reminder message was sent on Thursday, May 18, 2000, and data collection ended on Monday May 22, 2000. The message/reminder format is consistent with recommendations in the existing literature on on-line survey methods (Comely, 1996).

Although 11 days may seem a short period for data collection, previous research in this area suggests that an abbreviated period of data collection is not only possible, but is one of the advantages of on-line survey research. Comely (1996) lists an average response time of 4 days for e-mail surveys versus 11 days for postal surveys, and Smith (1997) states that “a large if not majority of survey responses are submitted within 24-48 hours of exposure.” These ideas were supported by the current study, in which 243 out of the 509 total responses (48%) were received in the first 24 hours after exposure.

3. RESULTS

RQ1. Research Question One asks about the influence of utilization of sources of information on the decision to adopt. Sources of information included in the analyses were official sources (University or department officials or publications), informal channels, the initial informational presentation made to departments, the on-site presentation made to the departments at the time of installation, and information provided by the ACS consultant performing the installation on the individual’s computer.

A series of bivariate correlations was performed to answer RQ1 (see Table 1). There were no significant correlations (p < .05) between utilization of sources of information and adoption of Outlook.

When examining individual components of Outlook, information received through informal channels was correlated with the use of the Folders (document sharing) function (r (351) = .16, p = .03). Therefore, receiving more information about Outlook through informal channels was associated with more frequent use of the Folders function. No other significant correlations were found between sources of information and adoption of the individual components of Outlook.
Influences of Sources of Communication on Adoption

Table 1. Correlation of Utilization of Sources of Information with Adoption

<table>
<thead>
<tr>
<th>Independent variable</th>
<th>r</th>
<th>p</th>
<th>Df</th>
</tr>
</thead>
<tbody>
<tr>
<td>Official channels</td>
<td>.03</td>
<td>.50</td>
<td>495</td>
</tr>
<tr>
<td>Informal channels</td>
<td>.05</td>
<td>.38</td>
<td>349</td>
</tr>
<tr>
<td>Informational presentation</td>
<td>.06</td>
<td>.30</td>
<td>357</td>
</tr>
<tr>
<td>On-site introduction</td>
<td>.06</td>
<td>.25</td>
<td>372</td>
</tr>
<tr>
<td>Consultant</td>
<td>-.03</td>
<td>.59</td>
<td>426</td>
</tr>
</tbody>
</table>

Approximately 25% of respondents indicated that they received no information from informal sources, the initial informational meeting, or the on-site introduction conducted at the time of installation. A majority of respondents received information from at least one of these sources. Follow up t tests were conducted to determine if there were differences significant differences in adoption scores between individuals who received no information from these sources and those who utilized the sources. Results of the tests (see Table 2) indicated that for each of the three sources of information, respondents who were not exposed to the source of information were significantly less likely to adopt Outlook than those who were exposed.

Table 2. Differences in Adoption Based on Exposure to Sources of Information

<table>
<thead>
<tr>
<th>Source</th>
<th>Not Exposed</th>
<th>Exposed</th>
<th>t</th>
<th>df</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M (SD)</td>
<td>M (SD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Informal Channels</td>
<td>19.33 (7.50)</td>
<td>21.24 (7.16)</td>
<td>-2.70\textsuperscript{a}</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>n = 151</td>
<td>n = 351</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Informational</td>
<td>19.13 (8.57)</td>
<td>21.27 (6.66)</td>
<td>-2.98\textsuperscript{a}</td>
<td>500</td>
</tr>
<tr>
<td>Presentation</td>
<td>n = 143</td>
<td>n = 359</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-site Introduction</td>
<td>19.37 (8.82)</td>
<td>21.11 (6.67)</td>
<td>-2.34\textsuperscript{b}</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>n = 128</td>
<td>n = 374</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\textsuperscript{a} Significant at p < .01 level.
\textsuperscript{b} Significant at p < .05 level.

The results of the t tests are surprisingly different from the results of the correlations conducted to answer RQ1. These differing results are likely an effect of the different prompts used to assess the construct "utilization of sources of information." A comparison of the results of the two series of tests suggests that utilization of sources of information is important in terms
of respondents’ level of reported adoption, but perceptions of related constructs such as perceived amount of information or usefulness of the information seem less important.

**RQ2.** Research question two asks about the influence of perceived credibility of official and informal sources of information on the decision to adopt. Credibility of official sources of information was calculated by summing the scores for the items “official sources were well-informed” and “official sources were accurate.”

Bivariate correlations were performed to answer RQ2. Neither credibility of official channels of information nor credibility of informal sources of information was found to be significantly correlated with the adoption of Outlook (p < .05).

Perceived credibility of official sources of information was not significantly correlated with the adoption of Outlook (r (493) = .05, p = .25). Perceived accuracy of information received through informal channels was not significantly correlated with adoption of Outlook (r (350) = -.06, p = .28).

Therefore, the credibility of sources of information did not appear to be associated with respondents’ adoption of Outlook.

However, when considering the individual components of Outlook, the credibility of information received through informal channels was correlated with the use of the Tasks function (r = -.11 (350), p = .04). Therefore, respondents who perceived the information they received through informal channels as credible were less likely to utilize the Tasks function. No other significant correlations were found between sources of information and adoption of the individual components of Outlook.

**RQ3.** Research Question Three examined the influence of the valence of information received through informal channels on the decision to adopt. Valence refers to whether respondents perceived the information they received as being generally positive or negative toward Outlook. Respondents who indicated that they did not receive any information about Outlook through informal interactions were excluded from the analysis.

Results of a bivariate correlation revealed that no significant relationship existed between the valence of information received through informal channels and adoption (r (364) = .08, p = .14). Descriptive data suggest that respondents who did hear information about Outlook through informal sources generally heard positive things. However, the positive information respondents heard about Outlook apparently did not influence their adoption of the program.
4. DISCUSSION OF SURVEY DATA

This investigation examined the relationship between end users’ reported adoption of the technology and several potential influences on the decision to adopt: exposure to sources of information, perceived credibility of formal and informal channels of communication, and valence of information received through informal channels. Although respondents’ perceptions of information and attitudes communicated about the technology were not found to be statistically related to adoption, exposure to information through informal channels and through certain official channels was correlated with adoption of Outlook. Communication of information and attitudes had been anticipated to have strong positive relationships with adoption of the technology in the current study, based on the literature on social construction of technology (Fulk, 1993). The results are surprising, in that Marjahan and Peterson (1985) describe communication as being central to the process of diffusion. Indeed, Rogers (1995) describes diffusion as a particular type of communication—communication about an innovation. In the current study, neither perceptions of formal nor informal sources of information, perceived source credibility, nor reported valence of information from informal sources were correlated with adoption.

However, there were significant differences between respondents who reported receiving no information from informal sources and two of the structured implementation activities and those who reported exposure to these sources. These results suggest that respondents’ perceptions of the information they received from these sources did not matter as much as the fact that they exposed to the sources — and therefore the information - at all. That is, varying responses to sources of information such as perceived usefulness, helpfulness, or even amount of information were not important in this study; however, exposure to these sources was statistically significantly correlated with adoption of Outlook.

4.1 Official Sources

Lewis and Seibold (1993) state that organizations engage in a wide variety of structured implementation activities (formal sources of communication) in order to facilitate adoption of innovations.

The finding that exposure to structured implementation activities was associated with adoption contradicts Weenig (1999) and Rogers (1995), who indicate that formal channels are more useful for potential adopters to gather initial information about the innovation, rather than shaping attitudes and behaviors. In the current study, exposure to information from structured implementation activities was correlated with adoption.
Although results in this study indicate that structured implementation activities are useful in facilitating adoption, this finding should be interpreted in the context of the wide variety of other influences on adoption beyond exposure to information in briefings and presentations. Lewis and Seibold (1993) note that in addition to structured implementation activities, which provide official sources of information about an innovation, there was a wide variety of other influences on the decision to adopt. Informal sources of information, user characteristics, innovation characteristics, and organizational structure and hierarchy can all influence the adoption or rejection of an innovation. Although results indicate that organizations should continue to invest time, money and effort in structured implementation activities in order to facilitate the diffusion of innovations, focusing solely on such efforts seems myopic. Given the wide variety of other sources of influence on the adoption of an innovation, the prudent organization will also give credence to the other sources of influence on adoption, and attempt to make them part of their implementation strategy.

4.2 Informal Channels

Rogers (1995) argues that positive evaluations of an innovation from near-peers tend to motivate individuals who hear the evaluation to adopt the innovation. He argues that such peer evaluations are important, because individuals seek information from known colleagues (i.e., through informal channels) in order to reduce uncertainty about the innovation. Rogers’ assertion about individuals’ need to reduce uncertainty is supported by Lewis and Seibold (1996). Rogers (1995) explains:

All innovations carry some degree of uncertainty for the individual, who is typically unsure of the new idea’s results and thus feels a need for social reinforcement of the new idea. The individual wants to know that his or her thinking is on the right track, in comparison with the opinion of peers (p. 168).

In addition to the assertions of Rogers (1995) and Lewis and Seibold (1996), Weenig (1999) argues that information received through informal channels influences attitudes and behavior toward an innovation, Fulk, Schmitz, and Steinfield, (1990) argue that individuals’ perceptions of a technology are shaped through social interactions with peers. Results of the current support prior research, to the extent that exposure to information through informal channels was found to be correlated with adoption in the current study. Although exposure to information through informal channels was associated with adoption, descriptive data suggested a general lack of use of informal channels. In addition to the 28% of respondents to the survey who
indicated that they received no information about the Outlook conversion through informal channels, another 30% indicated that they disagreed with the statement “I received a great deal of information about Outlook through informal interactions with co-workers.” Based on these results, informal interactions were not an important source of information for this particular innovation in this particular organization. Still, as exposure to information from informal sources was found to be more important than respondents’ reports of the amount of information received, analysis of the 28% of respondents who reported receiving no information through informal channels is warranted.

A likely explanation for respondents who reported no use of informal channels is that they may have selected answers to be more socially desirable. That is, respondents may have indicated a lack of information received through informal channels because they perceived participation in “the grapevine” to be socially undesirable. This argument is supported by Moon (1998), who found that individuals completing computer-based surveys (like the one used in the present study) were indeed more likely to provide socially desirable responses than individuals who were responding to the survey orally. This pattern of behavior was reported, despite assurances of anonymity. Moon argues that this behavior may result from a belief that responses are in fact being tracked by the computer, and may ultimately be tied back to the respondent.

Despite the reported lack of utilization of informal channels of information, exposure to information received through informal channels was statistically significantly correlated with adoption. However, perceived credibility of informal sources of information and reported valence of information received through informal sources were not associated with adoption of Outlook. That is, although exposure was associated with adoption, there was no pattern in perceived credibility and reported valence; instead, exposure itself is the foundation of the relationship. These results contradict the findings of scholars who argue that adoption of an innovation is heavily influenced by individuals’ perceptions of the attitudes of other members of their social network (Fulk, et al., 1990; Rogers, 1995; Weenig, 1999).

This surprising result may be explained by Kimberly (1981) who argues that attitudes are less important in case of technological diffusion than in other innovation diffusion efforts. Indeed, Weenig (1999) and Lewis and Seibold (1993) studies are of policy and program innovations, which are likely to have been received by potential users very differently than a new technology that required few operating or philosophical changes. The current study provides support for Kimberly’s argument.
However, the results of the current investigation may also indicate that attitudes toward the innovation may not be predictive of actual adoption behaviors in the case of a technological innovation. That is, valence of information received through informal channels may shape attitudes, but adoption behaviors may be influenced by other sources that outweigh the attitudes toward the innovation. Fulk (1993) notes that behavioral compliance does not necessitate an internalization of attitudes. This means that pressure to conform and comply may influence an individual to adopt even in the absence of favorable attitudes toward the innovation.

5. DISCUSSION OF OBSERVATIONAL DATA

One of the opportunities inherent in the current investigation was the ability to gain insight into the social system that was the context for the diffusion under study. Insights gained through these observations of the social system and implementation process help explain some of the findings discussed earlier. Observations of the social network illuminate the results of the study by highlighting the role of social networks, including accuracy and valence of information received through informal channels. The researcher's own social network demonstrated that even though survey respondents reported receiving little significant information through informal channels, informal channels of information were nonetheless active. In many instances, the information distributed through the grapevine was inaccurate (e.g., "Oh yeah, that's part of PeopleSoft." or "So now, anybody can schedule a meeting on my calendar.").

Although there was no significant relationship between perceived credibility of information and adoption, accuracy of information (a component of credibility according to McCroskey, 1966) is an important consideration in organizational diffusion efforts. The absence of accurate information and the presence of inaccurate information may undermine efforts to diffuse an innovation, and may fuel negative attitudes toward an innovation that must be overcome. In the current study, inaccuracy of information appearing in informal channels of communication was the impetus for determining content in some formal channels. That is, inaccuracy of information regarding the diffusion of groupware in informal channels prompted the need to disseminate accurate information through formal channels. The Groupware Implementation Coordinator was often required to provide accurate information about Outlook and the implementation process to counter inaccurate information presented by organization members at the initial informational meetings held in the individual departments. In fact, part of the rationale for having such
informational meetings was to correct inaccurate information in an effort to reduce the build-up of negative attitudes toward the innovation. Results of the current study support providing information through such structured implementation activities as the initial informational meetings.

The valence of information also was exhibited through social networks. When the University was considering implementing the newest version of GroupWise across the campus, users of the older version referred to it as “GroupWorse,” clearly a comment with negative valence. In addition to the accuracy and valence, communication through social networks proved to be a useful source of horizontal and upward communication in the organization. Information about frequent GroupWise system crashes eventually worked its way up to the Groupware Implementation Team. For example, the researcher heard about frequent GroupWise crashes from colleagues at the University Medical Center and the main campus library, and the researcher shared these observations with the Groupware Implementation Coordinator. Additionally, conversations with systems administrators at other institutions that were using GroupWise revealed similar instability problems. Such information spread through intraorganizational and interorganizational networks, and ultimately prompted a review of the decision to implement GroupWise and rejection of GroupWise in favor of Outlook.

6. LIMITATIONS OF THE STUDY

During survey construction, the multiple (and sometimes competing) needs and goals of the researcher and ACS staff led to numerous compromises regarding survey length. This obviously affected the number of issues that could be addressed, and affected the ability to include multiple measures for constructs to improve reliability. Also, in the editing of the survey, compromises were made for the wording of many questions, imbedding inconsistency in item prompts, and this may have affected the validity of some items. Therefore, these potential threats to the reliability and validity of the survey must be viewed as limitations of the study.

The self-report nature of the data used in the study also must be considered. The data must be interpreted with appropriate consideration given to this fact. Although self-reports of technology-in-use may be more useful than speculative measures that ask respondents to gauge likelihood of future use, the potential threat to validity of survey results from inaccurate self-estimation or social desirability effects must be considered.

The greatest limitation to the study is that of potential response bias. Although the survey yielded a valid data set of 502 responses, the invitation to participate was sent to approximately 1500 potential users. The
invitational e-mail specifically requested the participation of people who did not or did not plan to use Outlook in an attempt to minimize the likelihood of a sample biased in favor of Outlook users. However, individuals who were not using Outlook at all may not have received the invitation and therefore would have been systematically excluded from the sample. In addition, some Outlook users who did receive the invitation chose not to respond. They may have made this decision for a range of reasons from distrust of electronic surveys to time constraints. This raises serious concerns about the ability to generalize from the data obtained in this study, which the descriptive data suggest are the attitudes and behaviors of adopters of Outlook. The potential exclusion of data from non-adopters and non-respondents inhibits the ability to draw meaningful conclusions about the influences on adoption for all members of the organization, not just adopters. The potential threat to validity of the results posed by over-representation of Outlook adopters in the survey sample must be considered when interpreting the results.

In addition to the threats to validity posed by data collection and statistical procedures, the absence of some key constructs from the data set must also be viewed as a limitation of the current investigation. Specifically, measures of respondents' attitudes about Outlook would help determine whether or not information received through informal channels was shaping attitudes, and might provide insight into the link between attitudes and adoption behaviors. In addition, a programming error resulted in the omission from the data set of one measure of source credibility of official channels of communication. Finally, an assessment of the perceived importance of the diffusion of Outlook would have been useful in interpreting end-users' perception of the context of the study. For example, the lack of perceived autonomy in adoption of Outlook may seem less significant if the end-users viewed adopting a new software package as something other than an innovation.

7. IMPLICATIONS FOR FUTURE RESEARCH

Future research in the intraorganizational diffusion of technological innovations should continue to test the relationship between the influences of near-peers and social networks and end-users' adoption of innovations. Also, the influence of structured implementation activities on adoption should continue to be tested. Although perceptions of information and attitudes were not related to adoption in the current study, they may be more influential in circumstances where the technology is less familiar and compatible, or in the case of non-technological innovations. Also, although
Influences of Sources of Communication on Adoption

an on-line survey was a valid and convenient way to collect data in this study, future research in technology diffusion should utilize other methods of data collection as well. Use of focus groups, interviews, paper-and-pencil surveys, and unobtrusive measures may improve the validity of the results obtained, particularly in the case of gathering data about non-adopters. Further investigations of the diffusion of technology may be useful in testing Lewis’ (2000) assertion that technology simply is no longer a big deal in organizations, so that the diffusion of a new technology is viewed as an ordinary occurrence in the organization’s life cycle.

Other conceptualizations of adoption also may be useful in future research. The technology exists to track time spent using technological innovations. This may provide a more useful measure than self-reports of use. Still, even if the actual use were being tracked, there is no guarantee that the innovation is being used as it was intended when it was implemented. Future investigations may help to further identify the factors that influence not just attitudes and beliefs, but operational adoption. It would be useful to develop a more clear understanding of how information, attitudes, and adoption are linked.

As King and Anderson (1995) indicate, a universal theory of diffusion may elude us. Perhaps what is needed is the ability to understand the multiplicity of factors that influence diffusion, and to be able to examine those factors in context.

REFERENCES


Knowledge Creation in Improving a Software Organisation

Pouya Pourkomeylian
Pouya.Pourkomeylian@astrazeneca.com

AstraZeneca R&D Mölndal, S-431 83 Mölndal, Sweden
Tel: +46 31 776 17 96, Fax: +46 31 776 37 30
Viktoria Institute, Box 620, 405 30 Göteborg, Sweden

Key words: Software Process Improvement (SPI), Organisational Knowledge Creation (OKC).

Abstract: Software Process Improvement (SPI) is a systematic approach for improving the capabilities of a software organisation. This study shows the results of a collaborative research initiative in which an SPI project was conducted and analysed as organisational knowledge creation. The study explains how knowledge is created through transformation between tacit and explicit knowledge and through interaction between different organisational levels of actors. On the basis of our findings it is suggested that two types of knowledge are created in an SPI project based on completely different knowledge creation behaviour.

1. INTRODUCTION

Software development has existed as a discipline for more than forty years but has not yet become a disciplined process. Software projects are almost always later than expected, the costs of developing software are
higher than planned, and the functionality and the quality of the final products (software and documentation) are lower than expected (Paulk, 1997). Software organizations have used different methods for improving software processes. The most recent approach for improving software processes is Software Process Improvement (SPI), which is a systematic approach toward changing software development practice.

The first step in improving software processes is to understand the current status of the software development process (Humphrey, 1989). One way of doing this is to make an assessment based on a model as a road map. During the past years software organisations have used different appraisal approaches to identify what should be improved in their software processes. The most popular assessment model is the Capability Maturity Model (CMM), which is a normative approach to software process improvement developed by the Software Engineering Institute (SEI) (Paulk, 1993). Other approaches include BOOTSTRAP (Kuvaaja, 1994), SPICE (Thomson & Mayhew, 1997), ami (ami, 1992), TickIT (TickIT, 1995), and TRILLIUM (Thomson & Mayhew, 1997). Common for all these approaches is that they apply Total Quality Management (TQM) principles to SPI. After making an assessment further improvement activities should be planned and performed to create new or modified software processes.

Different reports have pointed out difficulties in performing SPI projects in practice (Curtis, 1996), (Debou, 1997), and (Goldenson and Herbsleb, 1995). An SPI effort is successful when the new or modified software processes are created and used in the organisation's daily practice and have been proven to function in achieving their goals. Success with SPI seems to depend on a complex mix of highly interrelated factors acting in different phases in an SPI project. Different factors such as scaling the SPI initiative, setting realistic goals, the complexity of organisational changes, and the organisational culture have made it difficult to achieve success in SPI initiatives (Goldenson & Herbsleb, 1995), (Herbsleb et al., 1997), (Mashiko and Basili, 1997), and (Johansen and Mathiassen, 1998). But SPI has also been shown to be able to help organisations gain organisational benefits (Hayes and Zubrow 1995), (Larsen and Kautz 1997), and (Wohlwend and Rosenbaum 1994).

An organisation's software development practices are based on the existing knowledge of practitioners and managers about the software development practice (Arent and Norbjerg, 2000). To change software development practices the organisation should improve the practitioners' existing knowledge (both tacit and explicit) of the software practices. The created new or modified knowledge should then be transferred to all organisational levels to become part of the practitioners' daily work. Creating new or modified software processes in this way is a knowledge
creation process in which different actors at different organisational levels are involved in creating different types of knowledge.

Some recent reports have reflected on the importance of creating and managing knowledge and learning issues for SPI initiatives. Arent and Norbjerg (2000) analysed how organisational knowledge creation process and learning can support SPI initiatives. Stelzer, Mellis, and Herzwum (1998) studied how principles and technologies from organisational learning can apply to SPI initiatives and become enablers of SPI success. Halloran (1999) investigated the relationship between an SPI approach and organisational learning. These studies indicate that the concept of knowledge creation and learning can support SPI initiatives. We believe that the concept of organisational knowledge creation has much to offer the SPI community, especially in the following three questions: 1) What types of knowledge are created as the result of performing an SPI project? 2) Which actors are involved in the knowledge creation process? 3) How do they interact to create knowledge?

As a framework to support the analysis of the SPI project this study has chosen Nonaka and Takeuchi’s theory (see Nonaka and Takeuchi, 1995) because their theory explicitly deals with the fundamental process of knowledge creation and supports an understanding of the interaction between individuals, groups, and organisations in the knowledge creation process. This approach has demonstrated its usefulness in relation to SPI in a study by Arent and Norbjerg (Arent and Norbjerg, 2000) analysing the learning process in SPI.

1.1 The Research Approach

Using a collaborative practice research approach (see Mathiassen, 1998) this study has combined action research in combination with field experiment, and practice study aiming to change practice. Improving practice is the distinguishing feature of collaborative practice research and action research in general (Baskerville et al., 1996), (Mathiassen, 1998). The action research approach is chosen for this study because of its strong support in: 1) integrating research and practice, 2) involving practitioners in the problem being studied, 3) giving the possibility of introducing change at the same time the research is going on.

In this study an SPI project was done during the period of April 1999 to June 2000. Several practitioners (software engineers) were involved during both the evaluation of software projects and the improvement of software processes. The SPI-group including: software engineers, assisting consultants, and the author (leading the SPI project) working with improving software processes became a forum for evaluating the Software Engineering
(SE) and the SPI practices, for creating and experimenting with new or modified software processes, and for learning about SPI in practice. Action research in this study was set up to improve three main software processes using the IDEAL model (see section 2.1). Field experiments were set up as controlled research efforts in which the created software processes were tested in one selected software pilot project to show the effects of the created processes. Focused practice studies were initiated to learn about the current maturity level of the software organisation.

One focused SPI practice was to make a CMM assessment to establish the current maturity of software processes. To collect data about the current capability of software processes at the software organisation a modified CMM assessment based on a method called Questionnaire Based Assessment (QBA) (see Arent and Iversen, 1996) was made for three different software development projects chosen from two different software development groups. The following Key Practice Areas (KPAs) were included in the assessment to identify software process problems: 1) Software Project Planning, 2) Requirements Management, 3) Software Project Tracing and Oversight, 4) Software Quality Assurance, 5) Software Configuration Management.

Project managers and developers of three selected software development projects answered the CMM-based questionnaire. The collected data were statistically analysed and proposals were developed for improving software development projects on the basis of the results of analysed qualitative data collected from the assessment, the software process improvement literature, and other quality improvement findings from earlier quality activities in the software organisation. The SPI-group met at least eight times throughout the 14-month period for planning and organising SPI initiatives and discussing difficulties and problems. The new and modified software processes will be implemented in the whole organisation starting in August 2000. The results and lessons learned during the improvement phase have been documented. These lessons have been both interpretative, i.e. helped us to understand the practice, and normative, i.e. helped us to design new or modified software processes and improve the practice. Knowledge gained and experience from doing this research in practice have created new research activities for further studies. The author actively participated in the practical work with SPI in the organisation, such as conducting the CMM-based assessment, running and participating in workshops and seminars, performing interviews and analysing results.

The section below discusses the software process improvement and organisational knowledge creation concepts and presents the framework for the analysis. Section 3 presents the case. Section 4 presents a map of the knowledge creation process in the SPI project and discusses the findings.
according to the three questions mentioned above, and section 5 concludes the paper by presenting the lessons learned and pointing out areas for further research.

2. BACKGROUND

Software Process Improvement (SPI) was originally developed at the Software Engineering Institute (SEI) at the Carnegie Mellon University and was based on ideas presented by Humphrey (see Humphrey, 1989). According to Aaen et al. (2000) SPI is based on a number of ideas that offer answers to specific concerns. SPI has three fundamental concerns: the management of SPI activities, the approach taken to guide the SPI initiatives, and the perspective used to focus attention on the SPI goal(s).

The management of SPI initiatives is based on three ideas: 1) the SPI activities are organised in a dynamic fashion, 2) all improvement efforts are carefully planned and 3) feedback on effects on software engineering practices is ensured. The approach to SPI initiatives is guided by three additional ideas: 1) SPI is evolutionary in nature, 2) SPI is based on idealised, normative models of software engineering and 3) SPI is based on a careful creation and development of commitments between the involved actors. Finally the perspective forward the SPI goal is dominated by three ideas: 1) SPI is focused on software processes, 2) the practitioners' competencies are seen as the key resource and 3) SPI aims to change the context of the software operation to create sustainable support for the actors involved. The basic idea in SPI is to focus on software processes as social institutions with a complex interplay of people, methods, tools and products (Aaen et al. 2000).

SPI is focused on improving software processes based on practitioners' ideas and experiences. This involves capturing practitioners' tacit knowledge (know-how) and transferring it to explicit knowledge, which should then be combined with the organisation's other explicit knowledge prepared for use in practice by all practitioners in different organisational levels.

2.1 The IDEAL Model

A popular model in the field of SPI that is suitable for assistance in managing SPI initiatives for implementing organisational changes is the IDEAL model (see McFeeley, 1996). As shown in figure 1 the IDEAL model considers five phases (Initiating, Diagnosing, Establishing, Acting and Learning) of a software process improvement initiative, which provide a
continuous loop through the steps necessary for software process improvement (McFeely, 1996).

![Figure 1. The IDEAL Model (Gremba and Myres 1997)](image)

Once the first cycle of SPI has been completed there will be a need to regularly repeat the entire process. However, the ultimate goal in organisations should be to succeed in achieving process implementation in the whole organisation. This should lead to the creation of a process culture in which process discipline prevails. Our intention in using the IDEAL model was to establish successful improvement activities and infrastructures for SPI initiatives within the software organisation. This study includes the Initiating, Diagnosing, Establishing, and some parts of the Acting and Learning phases.

2.2 Organisational Knowledge Creation

Nonaka and Takeuchi (1995) use two dimensions of knowledge creation to explain the process of organisational knowledge creation: 1) the ontological and 2) the epistemological.

The ontological dimension focuses on individual knowledge creation. The organisation supports creative individuals or provides a context for them in which to create knowledge. Organisational knowledge creation is understood as a process that “organisationally” amplifies the knowledge created by individuals and crystallises it as a part of the knowledge network of the organisation. This process takes place within an expanding “community of interaction”, which crosses intra- and inter-organisational levels and boundaries.
For the epistemological dimension Nonaka and Takeuchi (1995) draw on Michael Polanyi’s (1966) distinction between explicit knowledge and tacit knowledge. Explicit knowledge refers to knowledge that is transmittable in formal, systematic languages. It can be articulated in formal languages including grammatical statements, mathematical expressions, specifications, manuals and so forth. It can be transmitted across individuals formally and easily. Tacit knowledge is personal, context-specific, and therefore difficult to formalise and communicate. It is personal knowledge embedded in individual experience and involves intangible factors such as personal belief, perspective, and the value system. Tacit knowledge is difficult to communicate and share in the organisation and must thus be converted into words or numbers that anyone can understand.

Nonaka and Konno (1998) describe two dimensions to tacit knowledge. The first dimension is the technical dimension, which encompasses the kind of informal personal skills or crafts often referred to as “know-how”. The second dimension is the cognitive dimension, which consists of beliefs, values, ideals and mental models that are deeply ingrained and which we often take for granted. They argue further that this cognitive dimension of tacit knowledge shapes the way we perceive the world. This kind of knowledge could also be defined as procedural knowledge used in problem solving and decision making (Nonaka, 1995), (Firebaugh, 1989). According to Nonaka and Takeuchi (1995) organisational knowledge is created during the time the “conversion” takes place, i.e. from tacit to explicit and back again into tacit. The interaction between these two forms of knowledge is the key dynamic of knowledge creation in the organisation.

Knowledge Conversion

Knowledge conversion is a “social” process between individuals and is not confined to one individual. Assuming that knowledge is created through the interaction between tacit and explicit knowledge four different modes of knowledge conversion are possible (Figure 2). The content of the knowledge created by each mode of knowledge conversion is naturally difficult, which creates different contents of knowledge (Nonaka and Takeuchi, 1995):

1. From tacit knowledge to tacit knowledge (socialisation that creates sympathised knowledge). The socialisation mode usually starts with building a “field” of interaction. This field facilitates the sharing of members’ experiences and mental models. Socialisation involves the sharing of tacit knowledge between individuals.

2. From tacit knowledge to explicit knowledge (externalisation that creates conceptual knowledge). The externalisation mode is triggered by meaningful “dialogue or collective reflection,” in which using
appropriate metaphor or analogy helps team members to articulate hidden tacit knowledge that is otherwise hard to communicate. Externalisation requires the expression of tacit knowledge and its translation into comprehensible forms that can be understood by others.

3. From explicit knowledge to explicit knowledge (combination that creates systematic knowledge). The combination mode is triggered by “networking” newly created knowledge and existing knowledge from other groups in the organisation, thereby crystallising them into a new product or service.

4. From explicit knowledge to tacit knowledge (internalisation that creates operational knowledge). “Learning by doing” triggers internalisation. The internalisation of newly created knowledge is the conversion of explicit knowledge into the organisation’s tacit knowledge. In practice, internalisation relies on two dimensions. First, explicit knowledge must be embodied in action and practice. Second, there is a process of embodying the explicit knowledge by using simulations or experiments to trigger learning by doing processes.

![Diagram of knowledge creation](image)

**Figure 2.** Contents of knowledge created via the four modes (Nonaka and Takeuchi, 1995)

This study is focused on the *process* issues of knowledge creation in SPI, the actors involved, and the conversion of new or modified knowledge between different organisational levels. We interpret changes in practitioners’ understanding of the software processes, and changes in practice as indicators of new *tacit knowledge*, and new guidelines, policies, and manuals as new *explicit knowledge*.
3. THE CASE

This study was performed at AstraZeneca one of the world’s leading pharmaceutical companies. AstraZeneca is a research-driven organisation with a formidable range of products designed to fight disease in important areas of medical need. The company was formed in April 1999 by the merger of Astra AB and Zeneca Group PLC. AstraZeneca has a strong research base and powerful product portfolio, designed in seven areas of true medical need – cancer, cardiovascular, central nervous system, gastrointestinal, infection, pain control and, and respiratory. AstraZeneca is globally number three (1999) in ethical pharmaceuticals and has more than 50,000 employees worldwide. It has research and development (R&D) centers of excellence in Sweden, UK and the USA and R&D headquarters in Södertalje, Sweden. The company has some 10,000 R&D personnel and a US $2 billion R&D investment in 1999, extensive global sales and marketing network, employing over 25,000 people, and 12,000 people employed in production in 20 countries.

3.1 The Software Organisation

AstraZeneca has four departments that supply global IT services to the whole company: one in the UK, one in the USA, one in Sweden and one to provide IT support for research and development for the whole organisation. In addition to this, there are five global supplier managers who have the responsibility of controlling the needs of IT services in the business functions in the company. Furthermore, there is a company staff with central IT departments for solving problems related to technology adoptions, infrastructure, security, integration, and strategies. Beyond all this, there are IT functions that support the local marketing company in the respective countries. There are in total 2,500 persons working with IT-related questions in AstraZeneca.

This research started before the merger between the two companies in an IS organisation called Clinical Research and Information Management (CRIM) at the former Astra Hässle in Sweden and continued later in the new IS organisation, which then changed its name to Development IS (DeVIIS). DeVIIS supports clinical and pharmaceutical projects, Regulatory Affairs and Product Strategy and Licenses at AstraZeneca R&D Mölndal. DeVIIS is also responsible for influencing the development of the global clinical research processes and IS/IT tools in AstraZeneca. DeVIIS comprises 90 people including contractors, most of whom have backgrounds in IS/IT.

Many regulatory authorities require that pharmaceutical companies and their software organisations comply with GXP (Good Manufacturing
Practice, Good Clinical Practice, and Good Laboratory Practice) rules. GXP rules are the authorities’ quality requirements to pharmaceutical companies for ensuring patient health, the quality of processes (e.g. clinical studies or software development) and the quality of products (e.g. tablets or software). As a software organisation in the pharmaceutical business, DevIS must address many quality requirements. One fundamental requirement is that DevIS must be able to show the authorities, by documented evidence, that software development activities (e.g. software change control, software validation, and data processing and storage) are being performed in compliance with quality requirements. Therefore every software project regulated by GXP requirements should carefully apply all quality rules and be able to show by documented evidence that the software is compliant with the related GXP requirements. The company long ago adopted standard operation procedures that explicitly describe the company’s software quality rules. These standard operation procedures should be applied in all information systems regulated by GXP requirements.

Employees of DevIS are basically engaged with software development, software maintenance and software operation activities. The software development activities occur in two forms: 1) development of totally new software products (software development) and 2) developing or changing existing software products (software maintenance). A typical software development project at DevIS is scheduled to take between six months and one year and includes analysis, design, construction, testing, and validation. Software maintenance activities can consist of changes in the code or developing a completely new application for existing software products. Software products in DevIS include the software and all related documentation (e.g. user requirement specification, test plan, validation plan, validation report, user manuals etc.).

3.2 The Problem Area

The results of a problem analysis performed in early 1999 in one software development group within DevIS showed a need for improving software project disciplines and providing guidelines to understand the standard operation procedures and GXP rules. The director of DevIS initiated an improvement project called Software Process Improvement at DevIS (SPID) to understand the existing problems and improve the organisation’s software processes. The following figure illustrates a rich picture of the SPID project.
The SPID project was initiated, organised, planned, and performed during the period of April 1999 to May 2000 and aimed to improve DevIS' software processes. A maturity assessment using a modified CMM-based (Capability Maturity Model) assessment method, QBA (see [1996]), showed that DevIS was by then a level one organisation and addressed improvement possibilities in all analysed KPAs (Key Practice Areas). An improvement report based on the assessment's findings and other findings from earlier improvement initiatives at DevIS addressed six improvement activities. The steering committee of SPID gave priority to the following improvement activities (improvement decision) from the improvement report:

1. To establish a minimum documentation level for documenting the results of software projects and create the software documentation process.
2. To improve processes for software validation, software change management, and document version control.
3. To create a template library including templates for documentation of software development activities, such as: user requirement specification, design specification, test plan, and validation plan.

The SPI group including (5 software engineers, 2 assisting consultants, and the author) started planning and performing improvement activities over a four-month period. This initial phase of SPID was scheduled to be finished in June 2000. The implementation activities for implementing the newly created software processes in the whole organisation start in August 2000.
This section discusses SPID on the basis of knowledge creation framework described earlier in this paper. We present a map illustrating the knowledge creation process in SPID including the software process improvement steps, the knowledge creation processes, the knowledge created, and the actors and the organisational levels involved.

During the initiating phase, we realised that the standard CMM maturity assessment developed by the SEI (see Zubrow et al. 1994) was too general for our situation. We adapted a simplified CMM-based assessment method, QBA (see Arent and Iversen, 1996), that focuses on level two and modified it to fit DevIS’s terminology and needs. The assessment indicated that we needed to focus more on software validation, change, and version control, and on creating templates rather than on processes related to project management (e.g. software project planning, software project tracking and oversight). The software organisation aimed to create only a few processes based on practitioners’ experiences and the organisation’s quality requirements. The SPI-group working with improvement activities spent a great deal of time studying and understanding the CMM, the organisation’s quality requirements, the organisation’s existing standards, and the software engineering literature to create its own understanding of what is needed to create the new processes.

However, we succeeded by following the IDEAL model in organising and planning for the performance of the improvement activities (the I, D, E and partly A, and L steps in IDEAL) in a systematic way. This allowed us to see very early in the project the starting point, the finishing point and all activities included in reaching our targets. We followed IDEAL’s main steps very naturally and learned how to systematically go about creating the new or modified software processes. However, the path from knowing what to do to doing it in practice was neither easy nor clear.

4.1 The knowledge creation process in SPID

According to the organisational knowledge creation framework presented earlier in this paper, the organisational knowledge creation process starts at the socialisation phase at the group level and continues to the externalisation phase, aiming to create new explicit knowledge based on the interaction between tacit and explicit knowledge. The created explicit knowledge will, then be combined with other already existing explicit knowledge in the organisation. The new or modified knowledge is subsequently put into practice through learning by doing to become tacit knowledge.
In our software process improvement project the first organisational knowledge creation process started in the socialisation phase creating sympathised knowledge about the SPI concept through interaction between tacit to tacit knowledge. The author arranged meetings with the management to introduce the concept of SPI (creating SPI-knowledge). These meetings were held on an individual level between the director of the software organisation, other software managers, and the author aiming to learn about the SPI and the possibilities for gaining benefits by doing an SPI project. Other meetings with the same contents were held with the SPI-group for similar purposes. The created tacit SPI-knowledge then became explicit SPI-knowledge through a dialogue in the externalisation phase in which conceptualised knowledge was created mostly by the author, the software managers, and the assisting consultants. The author arranged other meetings to identify the initial improvement infrastructure needed to carry out the project. The created SPI-knowledge in this phase was mostly in the form of project specifications including information about the SPI plan, resources needed, role descriptions, goals, and responsibilities. The created explicit SPI-knowledge (the project specification) was then combined with other existing knowledge and experience in improving software processes, and other improvement models at the software organisation to create systematic knowledge. In this phase (combination), knowledge was created mostly at the individual and group levels through a dialogue between all actors. The operational SPI-knowledge was then created through practising software process improvement activities. The author, the software engineers, the software managers, and the assisting consultants were all involved in making SPI happen in the organisation. In this phase the created explicit SPI-knowledge became tacit SPI-knowledge through practice. This means that, in our software process improvement project, all organisational knowledge creation phases involved all actors in individual and almost group levels in the creation of SPI-knowledge.

The other organisational knowledge creation process in our project deals with creating SE-knowledge (Software Engineering Knowledge). This process also started in the socialisation phase through externalisation, and combination to create SE-knowledge. A modified CMM assessment done in three software projects involving three software project managers and two software developers to identify the current level of software process problems. The results and the recommendations of the assessment were identified and documented. This action led to the creation of the first explicit SE-knowledge about the current maturity level of the organisation based on the results of the assessment. The author and the assisting consultants were involved in creating this knowledge. This knowledge was then combined and integrated with other organisational requirements, and findings from earlier
improvement activities. The author interpreted the authorities’ and organisation’s quality requirements on software development practice and created explicit knowledge about the quality requirements in an individual level. The created knowledge about the quality requirements, earlier improvement findings, and assessment findings were summarised on an improvement suggestion report created by the author and the assisting consultants and presented to management for a decision. The created explicit SE-knowledge was then transferred to group levels when the author held presentations to give results of the assessment in different meetings for different groups. This explicit SE-knowledge was then used by the SPI-group as input to create the minimum baseline level for documentation and other new software processes.

The next step involved the process of capturing practitioners’ ideas about software practices and forming them into explicit knowledge. The author arranged meetings in which the software engineers and project managers shared their experiences and ideas about the software processes (in the socialisation phase). We focused in this phase on creating a common understanding of the software process problems and get agreeing on the ideal software processes. In the next phase (externalisation) we focused on creating explicit knowledge about the new software processes based on the practitioners’ ideas. We created some drafts for the new processes and discussed and changed the contents of the processes several times until we could agree upon an acceptable level. This explicit SE-knowledge was created by the SPI-group at the individual and group levels. The software managers were not involved in creating SE-knowledge. This phase was one of the most important and also one of the most difficult phases in our SPI project. This suggests that such a project should not create new processes and standards detached from practices. New or modified processes should be created based on practitioners’ experiences from practice. This suggestion is supported in several studies that indicate that routines and standards are learned and developed in practice, not from formal explicit standards procedures (Brown and Duguid, 1991), (Arent and Norbjerg, 2000).
The following tables illustrate the knowledge creation process in our project: *: Total, -: Partly, -: Nothing.

**SPID: Software Process Improvement at Development IS (our SPI project)**

**SPIer: Software Process Improver, (the author)**

**SMs: Software Managers**

**SPICs: Software Process Improvement Consultants: (the assisting consultants)**

**SEs: Software Engineers**

**OKCL: Organisational Knowledge Creation Level**

**Ind.: Individual, Gro.: Group, Org.: Organisational**

**OKCP: Organisational Knowledge Creation Process**

**Soc: Socialisation, Ext.: Externalisation, Comb.: Combination, Intern.: Internalisation**

---

<table>
<thead>
<tr>
<th>OKCP Stage</th>
<th>OKCP Level</th>
<th>OKCP Activity</th>
<th>OKCL Ind.</th>
<th>OKCL Gro.</th>
<th>OKCL Org.</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socialising</td>
<td>*</td>
<td>Completing the SPI project</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Establishing the initial improvement objectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Defining the roles and responsibilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Organising the initial improvement activity for preparing the SPI project</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Choosing the SPI plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Defining the period of the SPI project</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Establishing the measures and SPIs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Externalising</td>
<td>*</td>
<td>Implementing the SPI project objectives and strategies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Choosing the SPI plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Developing the organisational knowledge to externalise</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Establishing the measures and SPIs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>Establishing the measures and SPIs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 4. The organisational knowledge creation process in SPID**
5. LESSONS LEARNED

We have analysed a software process improvement project from an organisational knowledge creation perspective using a framework based on Nonaka and Takeuchi (1995) of organisational knowledge creation process. From the perspective of this framework we believe that it is useful to view such a project as an organisational knowledge creation process. On the basis of our experiences from this study we believe that certain types of created knowledge within an SPI project deal with the fundamentals of Software Process Improvement (SPI-knowledge) such as: management of SPI initiatives, issues related to how such an initiative should be guided, and issues related to SPI’s focus on target(s) (see Aaen et al., 2000). Others deal with issues related to Software Engineering practice (SE-knowledge) such as: project planning, quality assurance, change control, and configuration management (see Pressman 1997).

To illustrate the knowledge creation process we have applied the Nonaka and Takeuchi’s framework to one SPI project and created a map illustrating: the software process improvement phases, the activities, the created knowledge, the actors involved, and the organisational levels on which knowledge is created. On the basis of our experiences from this study we suggest a number of lessons relevant for future software process improvement projects, the SPI practice, and the effect of organisational knowledge creation on SPI.

Lesson one: Two related knowledge domains (Software Process Improvement (SPI)-, and Software Engineering (SE)-knowledge) are involved in the knowledge creation process in an SPI project. Two types of knowledge, i.e. SPI-, and SE-knowledge, play an essential role in SPI activities and the knowledge creation practice involved in an SPI project. The knowledge creation behaviour is completely different in these two knowledge domains.

Lesson two: a: The project manager of the SPI project, software managers, and assisting consultants are the key actors involved in the creation of SPI-knowledge. The most involved actors in creating SPI-knowledge during the very initial phase of the project are the SPI project manager, and the software managers. The SPI-knowledge is created mostly during the initiating, establishing, and learning phases. The SPI project manager is most involved in creating SPI-knowledge. The assisting consultants and the software managers contribute to combining the SPI-knowledge with their ideas and experience. Later on in the project other actors will become involved in the knowledge creation process.

Lesson two: b: The software engineers and the SPI project manager are the key actors involved in the creation of SE-knowledge. The SE-knowledge
is created primarily during the diagnosing, acting, and learning phases. During the diagnosing phase the SPI project manager, the software engineers, and the assisting consultant are involved. Later on in the project only the SPI project manager and the software engineers are involved in the creation of SE-knowledge. SE-knowledge is created on the basis of software engineers’ experience and ideas about software processes. The software managers are not involved in any phases of the project for creating SE-knowledge.

**Lesson three: a:** *The knowledge creation process in the case of SPI-knowledge happens mostly on the individual level and sometimes on the group level.* The very first SPI-knowledge is created on an individual level through interaction between the SPI project manager, and the assisting consultants. This knowledge is then transferred to the group level in which the software managers are involved.

**Lesson three: b:** *The knowledge creation process in the case of SE-knowledge happens mostly on the group level and sometimes on the organisational level.* SE-knowledge about current software process problems and new software processes is created on the basis of software engineers’ ideas on a group level. This knowledge is then transferred to other organisational levels.

These lessons agree with Nonaka and Takeuchi’s suggestion of starting the organisational knowledge creation process at the socialisation phase on the team level. However, in our SPI project, creating SPI-knowledge started in the socialisation phase on the individual level and then passes through almost all other organisational knowledge creation phases to the group level. Creating SE-knowledge in our project also started in the socialisation phase mostly on the group level and then passes through almost all other organisational knowledge creation phases to the organisational level. If an SPI project ends before an implementation of the new software processes in the whole organisation (as it did for us) then the internalisation of the SE-knowledge might happen only during a test period in a pilot project (if any) in the improvement phase. The organisational knowledge creation process for SE-knowledge passes to the organisational level first when the created software processes are implemented in the whole organisation to become a part of all practitioners’ daily work.

The implementation of newly created software processes is an issue of implementing changes in practitioner’s daily work. These changes should be organised, planned, and implemented in the whole organisation to be a part of the organisation’s daily practice. As mentioned before, one great challenge for DevIS is to find a way to implement the new or modified software processes in the organisation. An important question for further research is how can theories such as organisational learning and change
management support, understand, and enhance implementation initiatives of new processes at DevIS?

6. ACKNOWLEDGEMENT

This research was sponsored by AstraZeneca R&D Mölndal in Sweden in collaboration with the Viktoria Institute and the Institute of Computer Science in Aalborg, Denmark.

7. REFERENCES


Ami: Aml, application of metrics in industry. A Qualitative Approach to Software Management, CSSE South Bank University, London. 1992


Bach James: The Imaturity of the CMM. American Programmer. 1994


Curtis, B.: Improving software organizations with the CMM. Seminar. 1996.


Knowledge Creation in Improving a Software Organisation 223

How To Live With Software Problems

Klaus Jul Jeppesen
The IT University of Copenhagen, Glentevej 67, DK-2400 Copenhagen NV, Denmark
+45 38 16 88 89
klausjj@it-c.dk

Abstract: In general, software systems are relatively error free and support us doing our work. We could not live without these systems if we want to maintain the level of service, to which our customers have become accustomed. Most users, however, from time to time experience problems. It seems that despite all improvement efforts in areas such as specification, usability, testing and so on, software in use will cause problems, and we must find ways to live with this fact.

From a user perspective a system is either in the phase of implementation or operation. This paper focuses on the problems encountered by users when a system is in the operational phase and the development organisation is not available for problem resolution. The goal must be to support the users to get the highest possible benefit from the system. We must help users find the best way to live with the system.

This paper describes how to record problems and the basic principles to follow when these problems are processed. Examples are used to show how the potential benefit can be estimated. This forms the base for a decision on whether it is worth to cure the problem or not. It is common sense to provide support in this way, but experience shows that many organisations do it in an unstructured way and without recording the costs and benefits. Many provide support only because the cost of task failure, due to system problems, is high.

The paper then presents various measures to be taken to help the users do their work efficiently and get the best out of the system. It will be demonstrated how many of the problems can be cured through better information to the users, better work procedures, and system tailoring - without modifying source code.
1. BACKGROUND

Computer systems have the potential to substantially improve work practices and allow users to develop organisation to which they belong. As Pressman [5] describes, software is a key technology and software-based systems are used in many places. Most companies could not operate without computer systems and maintain the level of service expected or assumed by customers.

The author has been working for 20 years within the field of development as well as that of operation of an installation with several hundred users. It is my experience as a manager of software development that the development process, by now, is fairly well described, and when the suggested practices are followed, the result can be a high-quality system. On the other hand, during my time as a manager of IT-support, I have frequently observed users encountering problems. This has caused them a lot of frustration, because small features of the system have given them a lot of inconvenience or annoyance. I noticed that often a small action by someone, who happened to be there and knew what to do, was most helpful to the user. Gradually I developed the idea that we must accept the fact that users encounter problems, and that we can improve the users’ situation, if we address the problems adequately.

The 70s and 80s brought many stories of errors in software systems. Some labelled it a “software crisis” (e.g. Pressman [5]). Many studies and much effort have been made to deliver software error free. Techniques for establishing specifications have been improved. Usability methods are used to ensure that systems will satisfy the users. Designs are reviewed, and modern programming languages are supposed to ensure a correct implementation. Rigorous testing verifies that the system operates as specified. Not that software, in general, is totally free of errors, but the situation has improved. As Wirth [6] stated: “There is an enormous amount of software in this world, here and everywhere, that works perfectly well and does its job”.

In spite of improved software engineering, resulting in higher quality software, the users complain because they encounter problems during operation. They often experience situations where the system makes work awkward or in some cases makes it difficult for them to complete their task successfully. These problems range from mild annoyance to outright faults causing task failures. It is time to study the operational phase in more detail and find better ways to support it.

Traditionally, usability is discussed in the context of software specification and development. The usability, however, can only be finally evaluated, when the system is in the operational phase. Then it may be too
late to change the software and we must seek other solutions to reduce the number and the impact of user-experienced problems. Most users can tell at least one story about a problem in a system they use daily. Practitioners in the field of systems operation confirm that users still experience problems. The software seldom malfunctions, as this kind of errors typically has been removed during development. However something annoys or bothers the user. An example is a user who has forgotten a password or forgotten to change it before it expired. The user suddenly cannot gain access to the system at all. Another example is a user, who consecutively enters records via a screen, and the screen is cleared between each record, so several record fields have to be re-entered although they contain the same data as the previous record. A further example is the user who computes a report manually, not knowing that the system can do it. These examples are typical, and together with a few more, they will be analysed in the following sections.

Often the system is based on commercial off the shelf (COTS) software. In this case, we have little or no influence on the details of the software specification. Furthermore McKinney [2] reports that a drawback of COTS software is that the availability of features and bug fixes is related to releases, and that a new version “often swells resource requirements significantly”, which may prohibit the use of a new release. The functionality of COTS systems may not fit perfectly, but we use it anyway and have to use it as is.

A rich literature describes the software development and the various aspects of the software engineering. The software life cycle process standard defines all the activities including resolution of problems. But the ‘user-support’ process is only briefly mentioned as part of the IEEE standard [4] and the general literature on the subject is limited. The standard is limited to error reports that are fed back into the development process. In this paper we focus on the case where this is not possible. During the operation phase there is often no communication path back to development. In general the operations phase deserves a detailed discussion.

Through better support, many of the user’s problems can be avoided or a solution provided, so that at least the user can live with the problem and achieve an acceptable level of productivity. Due to lack of knowledge, the users typically do not utilise the full benefit of the system capabilities. Again, better support could help.

Beizer [1] categorises errors according to the source. This relates the problem to activities in the development process and is valuable, when discussing how to improve development. Here, however, we focus on actions to be taken after the problem has been identified and after development is supposed to be complete and the system is in operation.
During this phase we are interested in the type of action that can solve the problem.

This paper suggests that the problem processing should comprise the following steps: First recognise the problem and make a proper recording, then make a rough estimate of the potential benefits if the problem is removed. Such an estimate may show that the benefit is small and that there is no reason for further action. It may on the other hand reveal an interesting potential benefit. If so, possible actions with associated costs must be explored and an optimal set of actions found.

2. PROBLEMS ENCOUNTERED BY USERS

The following examples illustrate the problem types that users encounter. We will later see how many of the problems can be resolved by the support organisation. The common aspect of the examples is that the user experiences a problem. The root of the problem can be either the user, the system, or indeterminable which is, however, not that significant. The point is that users feel they need assistance or system changes to get their work done efficiently.

Each problem has its own set of solutions. Some are temporary workarounds and others solve the problem. Optimal solutions may not be available or the cost to get them may be too high. But the user should be left with the feeling, that the best possible solution has been achieved.

2.1 Examples of Problems

The examples describe cases experienced by the author. Many more examples could be given, but these few are sufficient to demonstrate the scope of problems encountered by users. The benefit of various solutions is presented in the following section.

1. **Password.** The users forget their password or neglect changing it when required (often despite warnings given by the system). When the user arrives in the morning, the system is blocked, so the user cannot log on to the PC, and no work at all can be accomplished.

2. **Office change.** Another example is an organisation where the employees frequently move physically between offices. When they move, the IT-department has to change some of the set-up of the PC before the user can get access to the computer network. The average response time from the IT-department is 6 hours, and support cannot be initiated before the move. The user can not use the PC the first day in the new office.
3. **Mismatching part numbers.** A company uses one system for bookkeeping and another for production planning and storage management. Once every day the two systems exchange information to synchronise sales, production, purchase of parts, and shipping of items produced. One day the bookkeeping system was upgraded to a new version with a different part numbering data structure – and the two systems could not exchange data, because the part numbers did not match. The production came to a halt after some time.

4. **Missing report.** The user is a project leader in an engineering department. When controlling costs of an order, the project leader must enter the order and print out a page for each cost item on the order and then manually calculate the total and the difference from the forecast. The project leader spends quite some time, and occasionally makes a mistake in the manual calculation.

5. **Entering payment information.** The user is entering payments in an accounting system. For each payment the screen is cleared, and the user must re-enter all information, including the date, the user identification etc. This is additional work that the user feels is a waste of time.

The potential actions against the described problems are listed in section 6.

### 2.2 Benefit

The benefit of curing a problem depends on the organisational structure and the type of work. To get some idea of the approach, we use an organisation I have worked in. The users were:
- 10 project leaders
- 50 staff employed in general administration (bookkeeping)
- 60 administrative assistants
- 200 technical computer users
- 10 users in supply management etc.
- 40 staff employed in production of goods, storage handling, and shipping
- 30 sales persons.

This gives a total of 400 PC users. The examples below demonstrate how carefully you must calculate the benefit before deciding a solution, as the conclusions in some cases are surprising.

1. **Password.** Let us assume that the password must be changed once a month. 5% of the users (20 persons a month) forget to do this. They spend 10 minutes getting support staff to enable them to log on. The organisation could benefit 10 minutes times 20 per month, which is 10 minutes per working day. From a pure economic evaluation this is not worth spending any effort on. But if you are sensitive to employee
satisfaction, something should be done. You might simply mention the rationale of password changes to the users in the next IT information letter and encourage them to do as requested.

2. **Office change.** Let us assume that the project groups are moving every 6 months on average and that half of the technical staff is involved. If each person waste one day per office change, 100 days are lost every half-year—equivalent to one full-time person. It would obviously be a good investment to make a guide so that the technical staff can make the set-up themselves.

3. **Mismatching part numbers.** When a system supporting the company infrastructure fails, the damage may be substantial. The lost working hours is only part of the cost. The loss of business and the disruption of production may be fatal for the company. The upgrade of the bookkeeping system must be undone, or the production system must be upgraded so that the part numbers can be synchronised. We are facing here an example of a problem to which a solution must be found immediately, and which support staff ought to prevent from occurring at all.

4. **Missing report.** Assume that the project leaders must report monthly, that each leader has 10 projects, and that the manual calculation takes one hour. This corresponds to 100 hours per month. In the actual case, it turned out that a report used by bookkeeping provides the information, so the function can be provided without any change to the system. There is substantial benefit, 100 hours per month and reduced employee annoyance.

5. **Entering payment information.** Assume the organisation makes 60,000 purchases per year. The saving by tailoring the system is perhaps only 30 seconds per transaction. However this totals approximately 40 hours per month. Again, a small change can bring a substantial benefit.

3. **SOLUTION SPACE**

There is no universal key to problem resolution. The actions described in the following sections must be seen as a set of ideas based on lessons learned from solving problems as described in the examples. A support organisation should make its own list of actions tailored to the applications currently in use. The list should be enhanced while the support staff gains experience in connection with the type of problems typically encountered. The possible actions are described in four groups:

- The first group of actions addresses the problems that are basically caused by the user’s lack of knowledge due to ignorance, poor
How To Live With Software Problems

documentation, lack of training etc. Short-term resolution is normally possible by advising the user. In the long-term the general knowledge level must be increased or the system must be made easier to use in order to avoid other users experiencing the same problem.

- The second group solves problems through a change of procedures and rules.
- The third group solves problems through change of system parameters. Many applications are built on standard software tailored to the individual user or a group of users, which makes it possible to change the functionality without changing the program code. An important aspect of this is that it can often be done on the fly or with only minor interruption.
- Finally the support staff cannot resolve some problems. Resolution must be referred to someone with better knowledge and access to more specialised tools, or resolution may require special skills, e.g. programming. Typically, this type of action is more costly, and the cost/benefit of problem resolution becomes an important issue.

3.1 Inform Users

Many problems arise simply because the user does not know how to use the system or why the system requires operations to be performed in a certain way. The erroneous use of the system leads to wrong results or failures because the system is used in a way never intended or tested.

The supplied standard documentation is often related to the product and not to the specific context, in which it is used. The documentation often resembles a guide to a toolbox. The support staff may find such documents useful, but that type of information is not sufficient to support the user’s actual work practice. The support organisation must ensure the availability of user-relevant documentation, which can be in the form of short instructions or recommendations. For instance a checklist of tasks, which the user routinely must perform, could be provided.

The support staff must foster an environment, where users will admit problems and ask questions. The support organisation could establish and maintain a database with frequently asked questions. Support staff would be responsible for the maintenance, e.g. providing correct and accurate answers and removing irrelevant questions. Users should be allowed to enter information into the database, which will allow users to learn from each other. A user might have a problem to which support staff does not have a solution, but some other user does. It is the responsibility of support staff to have this information reviewed and to prevent bad habits from spreading. Another database could give information about known problems and the
intended resolution. The cost of maintaining such information systems must, however, be carefully considered.

Yet another approach could be to establish “ambassadors” of the system easily accessible to the users. They are often called “super-users”, as they are actually using the system. They ensure a close contact with users and immediate responses to problems. Super-users work in the user environment and are consequently close to the users and their problems and frustrations. One must, however, ensure that the super-users are well trained and kept up-to-date on problems and their solution. The recording becomes more important, but difficult to enforce, as the support staff is distributed in the organisation.

### 3.2 Operational Procedures

Support staff can assist the user in organising work practice. Often some functionality is available through several different operational procedures. By interviewing or observing some users, the optimal operation may be determined and then communicated as best practice to all users. Storyboards may be used to discuss better work practices with users. A forum of users could be created, where procedures and rules can be discussed and elaborated in order to find the best way to live with the system or to utilise a certain function. Such a forum can also be used to test ideas for new procedures. For some problems support staff can provide a workaround. This will typically consist in a change in the operation of a function or in a recommendation to use other features of the system to achieve the desired result.

### 3.3 Tailoring

Tailoring denotes the setting of parameters to make a standard product or a framework system function in a certain way. In some systems the set of parameters is huge, and the setting non-trivial, because some parameters are interrelated. Support staff should be capable of performing some tailoring and in the non-trivial cases possibly communicate with somebody with more competence. To control the tailoring, support staff is responsible for configuration management, which ensures that modifications are properly documented and tested, before they are put into operation.

Likewise support staff must be responsible for configuration management of the software components. Support staff must participate in the testing of new versions or modifications to the system. This has got three purposes: First, it can prevent cases, where the system fails totally. Second, performing the system test is a good training as regards the system functionality and
operation. Finally, support staff will have more confidence in the system, when they have witnessed the test and know that the system works, at least with the test set-up.

It should be noted that much of this tailoring and configuration management equals software development. The good practices employed during the development processes should therefore be applied during support as well. Tailoring must be properly specified, documented, tested, and the changes and test results carefully recorded.

3.4 Escalate resolution

When the user problem involves a program error or missing functionality, the tailoring may sometimes provide a solution. In other cases, however, it requires a modification of the source code. Programming and tailoring is often outside the capability of support staff. In these cases support staff have to pass the problem to a third party, with the required skills.

Some problems are difficult to resolve, and the process may require time and money. In such cases the actions often have to be decided by management after a more thorough evaluation of the benefits compared to the costs. It should be the responsibility of support staff that all information required to make this decision is obtained from appropriately qualified sources.

In many cases, however, support staff must provide a solution, at least in the form of an acceptable work-around. This could be the case, when a problem, which is critical for operation, has occurred, and an immediate solution is required, or when escalation is not possible. Some examples are:

- there is no access to the software development,
- the original supplier does not have a support function,
- the time constraints do not permit the optimal solution to be found.

The only possibility for support staff is then to establish a usable solution by combining actions from the previously mentioned groups, i.e. through user information, change of procedures, or tailoring.

4. STRUCTURING THE SUPPORT PROCESS

Two aspects of the support process are discussed in the following. The first concerns the activities of the resolution process and the second the registration of all reported problems and the actions taken. More work is required to make a more complete description of the support process.
4.1 Resolution process

The goal of the support organisation is to have problems resolved quickly and efficiently. A quick solution that creates a new problem may, however, be worse than no solution.

The problem resolution process consists of steps similar to the processing of anomalies, described in the IEEE standard [3]. The steps are as follows:

a) Detect and report the problem,
b) Analyse the problem,
c) Develop a solution,
d) Deliver the solution to the user.

The support process starts, when the user experiences a problem or somebody observes the user practice and detects that user performance could be improved. It is important that the user recognises the cases where something is a problem, and that the user prepares a proper report. The environment should help the user make reliable, accurate, descriptive, and consistent problem reports.

Users must see a benefit from reporting problems; otherwise they will feel it is a waste of time. To obtain such an approach requires the right attitude, the right tools, and the right training of support staff. The support process must be visible, and the results well communicated to the users.

The support staff must consist of well-educated and trained personnel. Support staff must know the systems at least as well as, and preferably better than, the users. They must be motivated to provide competent responses to user complaints.

Support staff must have appropriate tools to process the problems. The tools are used to analyse the problem and assist in providing solutions. This could, for instance, be a tool with remote access to the user’s PC, enabling support staff to see what the user is doing and provide guidance to the user via telephone. It could be a tool to track all transactions and detect bottlenecks, or it could be a separate “development system” to experiment with different tailoring or user procedures. The required tools will depend on the application, and support staff must participate in determining, what is appropriate.

Quality assurance must be applied to the support process. It has often been experienced that the user is asking for a quick solution, and that support staff comes up with a quick fix. This is tempting and rewarding when successful, and terrible when it creates new problems. To be successful in the long run, support staff must ensure a proper quality. Techniques from software development in terms of reviews and testing should be applied to ensure success.
When the problem has been resolved, the solution must be rolled out to the user. This must be done in a professional way, and the solution presented to the user without making the user feel incompetent. The solution must also be distributed to other users, so that they know what to do, if they encounter the same problem, or how to prevent it from occurring at all.

4.2 Registration and Analysis

All support requests should be recorded. The registration system or tool must be made very easy to use, so that it does not constitute a barrier to reporting. Registration must for each problem record a short description of the problem and later the resolution. Furthermore, the time of the event, the time of the resolution, and the time spent should be recorded.

The data collection serves several purposes. The registration will provide data for the evaluation of benefits, for the prevention of problems, for the quality assurance, and for the strategy for long-term evolution of the system. It may also be used to assess the quality of the support organisation.

A prime purpose of the registration is to drive the prevention of problems. Analysing the different types of problems and the frequency or repetition of problems can be used to identify the need for better user education, and for implementation of modifications to the system. The recording should be organised so that it is easy and fast for support staff to find out whether a new report describes a new problem or a problem already recorded, and potentially solved. Support staff could also provide a ‘first-aid’, which should describe the most frequently asked questions, and the answer to these. Support staff should propose additional user training, since support staff knows where the problems are.

The records should be used for an assessment of the application used. Applications with frequent problem reports are candidates for a closer inspection. The impact of the problems on the work should be evaluated, and the cause of the large number of problems should be analysed. The knowledge obtained can be used to make decisions on the short-term development and enhancement, and to make strategic decisions concerning the long-term evolution of the system, or a potential replacement.

Finally, the records can be used to evaluate the quality of the support function itself. These records can be used to examine whether responses cause new problems, were wrong, or did not provide optimal solutions. The records can also form the basis for statistics on the average response times, the lead times, the total resolution time etc. Special care must be exercised when these figures are being analysed, as they can easily reveal information about individuals. For instance missing efficiency by support staff, or disclosure of individual users, who report many problems caused by their
own bad performance. The statistics must be made public in a way that hides all personal details.

5. **THE BENEFIT**

As was illustrated above, user support can be a substantial benefit to the organisation. The benefit will depend on the application as well as on the organisation, and also on the knowledge level among the users and support staff. Furthermore, user support may give indirect benefits, which are difficult to calculate. Benefit will vary during the lifetime of the application. When users start using the system, they will have many questions, typically simple to answer. Later, when they have gained experience, and some tailoring has been done, problem frequency will decrease. If support is not reduced correspondingly, costs may outweigh benefits.

The most significant indirect benefit is increased user satisfaction. It is commonly agreed, that a high morale and good general well-being at work increases productivity. For some applications it also leads to better customer service, which provides several advantages to the organisation. The indirect benefits must be estimated individually on a case by case basis.

The first consequence of letting support staff solve the problems is that effort and cost are moved from the user group to the support group. We must, however, assume that the support staff, if better trained and having better tools, can solve the problems more efficiently than the users. Hence support staff should spend less time on resolving problems than the user would have to do, and the overall effort used on problems should be reduced.

The real benefit comes from the activities mentioned above. Establishment of better procedures, tailoring etc. are one-time efforts to be made by support staff, but this will potentially benefit all users repeatedly. Not only does the number of users multiply this saving. The saving is recurring every day over the lifetime of the system. This is clearly illustrated in the benefit calculation for the missing report and entering of payments.

The users benefit the most, when they conceive the system as a good and pleasant system, assisting them in accomplishing their work in a productive manner. So investment in improved user knowledge also benefits. To ensure a high level of knowledge, it should be made attractive to the users to spend some time on training, studying manuals, and understanding recommendations.
6. ANALYSIS OF THE EXAMPLES

This section compares the four types of actions against the five sample problems in section 2.1. In table 1 each column represents one type of action. A given problem resolution can be achieved by actions from one or more columns. Each row represents one of the examples, and the cells give a short description of possible actions. The cost of actions differs and the cost/benefit of the problem resolution must be considered when appropriate action(s) are to be chosen.

As can be seen from the table 1, a given problem may be addressed in several ways. The problem with entry of payment information, for instance, has two potential actions. If some users work best when the screen is cleared, the individual users should be trained to find the possibility of making the appropriate selection. However if no users profit from this selection it is better to change the functionality system-wide through tailoring. Support staff must select the appropriate action(s). One action may provide a short-term solution, and another action may be required for a better long-term solution.

7. CONCLUSION

Personal experience and discussions with other practitioners support the assumption that no matter how well we develop software, there will be problems to deal with during system operation. Users encounter problems ranging from small annoyances to functions that are cumbersome to utilise.

First of all, it is important that all problems are carefully registered. The recorded information can be used to estimate the benefit of solving the problem and used for a general assessment of the system in use.

The idea is that the elimination of time-wasting problems is the key to success. The examples in the paper demonstrate that in many cases the elimination of problems result in a substantial benefit to the users. The greatest benefit is achieved, when support staff plays an active role in preventing the problems in the first place.
<table>
<thead>
<tr>
<th>Example</th>
<th>Information</th>
<th>Procedure</th>
<th>Tailoring</th>
<th>Escalate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Make a task list for the users to remind them to check for password change.</td>
<td>Consider change of password period.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New office</td>
<td>Enable the user to make the changes themselves.</td>
<td>Write a procedure for the changes to be made, when staff changes office.</td>
<td></td>
<td>Consider automating the set-up of computers on the network.</td>
</tr>
<tr>
<td>Mismatch part numbers</td>
<td></td>
<td>Ensure proper change and test procedures.</td>
<td>Make configuration management, test before operate, and apply development tools.</td>
<td></td>
</tr>
<tr>
<td>Missing calculation</td>
<td>Ensure exchange of best practices between departments.</td>
<td>Establish procedure for project reporting.</td>
<td>Use tailoring to provide the required reports – if possible.</td>
<td>Get external assistance to provide the required reports.</td>
</tr>
<tr>
<td>Entering payment information</td>
<td>Inform users on the possibility of selection of functionality.</td>
<td></td>
<td>Change functionality through parameter setting.</td>
<td></td>
</tr>
</tbody>
</table>
How To Live With Software Problems

When a benefit has been identified, the next step is to find the appropriate actions. They must provide a useful solution, and the costs must be justifiable by the benefit. The paper describes four types of actions:

- Inform users. In some cases, benefit will be achieved by providing better information to the users and to ensure that users exchange information. The information should help users avoid problems and find the best way to use the system. Guides, written specifically for the actual application, are more relevant to the users than the standard guides typically provided for software products.

- Work procedures. In other cases work practice can be investigated and developed in co-operation with the users. Users should be guided to use the system in a way that best supports their work.

- Tailor the system. The system can be tailored to the users actual needs. It must be accepted that tailoring is a kind of programming, and that good practices from software development, such as configuration management and testing, must be applied.

- Escalate resolution. Support staff cannot solve some problems and the resolution must be escalated. Often a work-around must be provided and the users will benefit even from a non-optimal solution instead of none at all.

The study shows that data must be collected to gain a better understanding of the problem types experienced by users. Such data would allow us to better estimate the potential benefits and suggest a more systematic selection of problem-solving actions. The big challenge may then be to get practitioners to use these recommendations.

REFERENCES


Introducing Concurrent Functional Programming in the Telecommunications Industry

Bjarne Däcker
Computer Science Laboratory, Ericsson Utvecklings AB, Box 1505, 12525 Älvsjö, Sweden

Key words: Functional programming, Concurrent programming, Technology dissemination

Abstract: This paper gives an overview of the concurrent functional programming language and its development, dissemination, and use.

Erlang was developed at Ericsson and is used for several large and important telecom systems. It is also available externally, both supported and through open source.

Erlang provides a highly relevant case-study of technology diffusion since its development touches upon many relevant topics such as applied research in the industrial environment and spread of technology through open source.

1. INTRODUCTION

Telecommunications systems place very difficult demands on the underlying programming and computer technology, such as large number of concurrent activities, systems distributed over several computers, large and complex software systems, software maintenance (reconfiguration, etc.) while the system is in operation, fault tolerance to hardware failures and software errors, etc.

The concurrent functional programming language Erlang [3, 5] was developed as a software technology to meet these requirements and to facilitate the design of telecommunications systems. Section 2 gives a short
overview over Erlang and the subsequent sections describe how Erlang was developed and experiences from its spread both inside and outside Ericsson.

Concurrent functional programming language
Erlang

\[ \begin{array}{c}
\text{Concurrent systems programming languages like Ada, Modula or Chill} \\
\text{Functional programming languages like ML or Miranda}
\end{array} \]

*Figure 1. The ancestry of Erlang.*

2. **ERLANG**

Erlang is a single assignment, functional programming language with dynamic typing not unlike Scheme. Its syntax, however, is more like ML or Prolog. Erlang has data types like atoms, numbers, lists, and tuples and uses pattern matching to select between alternatives.

An Erlang program is built up of *modules* which are separately compiled and loaded. Only explicitly exported functions can be called from another module.

A function can be spawned to create a concurrent process (or "thread of control"). Concurrency is supported by the Erlang implementation without help from the operating system. Processes have no shared memory and communicate by sending and receiving messages asynchronously.

Erlang processes are extremely lightweight and their memory requirements can vary dynamically. Erlang implementations support applications with very large numbers of concurrent processes (typically in the region of 20,000-30,000).

Erlang supports programming "soft" real time systems, which require response times in the order of milliseconds. Long garbage collection delays in such systems are unacceptable, so Erlang is able to reclaim memory in small parts of the system every time the garbage collector is invoked.

Erlang permits transparent distribution. An Erlang program running on a computer is termed an Erlang node. A distributed Erlang system consists of several Erlang nodes spread over many computers (perhaps running different operating systems) connected over a network. Erlang processes in different
nodes communicate through message passing in the same way as processes within one node.

One Erlang process can crash (because of type error, division by zero etc.) but this will only bring down that process, not the entire node. Erlang processes, however, can monitor each other so that an error can be received as an error message. This enables the design of robust systems where supervisor processes can take action, reclaim resources, log errors, restart a transaction etc.

Erlang allows program code to be changed in a running system (*hot code loading*). When a new version of a module is loaded, newly spawned processes will run the new version while on-going processes continue and finish undisturbed. It is thus possible to install bug fixes and upgrades in a running system without disturbing its (currently running) operation.

Erlang processes communicate with other programs or the operating system using the same message passing mechanism as is used between Erlang processes. If required for reasons of efficiency, C programs can be directly linked into the Erlang run-time system.

Since Erlang is implemented in C it is essentially available on all systems that run C. Erlang is at present supported for the following operating systems: Solaris, Windows NT, VxWorks, and Linux.

Erlang allows the same rapid prototyping and interactive development as, for example, Lisp but extended into the world of concurrency and distribution. The error handling mechanisms and hot code loading allow the design of high availability, robust, non-stop systems.

### 3. 1982-86 TECHNOLOGY EVALUATIONS

The first development step were experiments at CSLab (*Computer Science Laboratory*) at programming telephony using different programming languages and technologies based on a PABX (*Private Branch Exchange*) controlled by a VAX/UNIX system. The main conclusions were:

- Telecommunications systems are so large and heterogeneous that it seems likely that different programming techniques would be used for different parts of the systems.
- The shortest, clearest, and most *beautiful* programs (also those that were closest to a formal specification) were those written using functional or logic programming languages.
4. **1986-89 PROTOTYPING**

The next development step started with Prolog to which concurrency was added. Soon backtracking had to be dropped (a ring signal once sent out cannot be retracted) and the budding programming language changed to a functional style [2]. It was named Erlang after the Danish mathematician Agner Krarup Erlang, creator of the *Erlang loss formula* widely used for traffic calculations.

During 1988-89 CSLab collaborated with a prototyping team at EBC (*Ericsson Business Communications AB*). They worked on system architectures and used Erlang for building prototypes and patiently endured sometimes radical changes to the new language. The collaborative project was reported in December 1989 and showed a striking improvement in design efficiency over current practices.

5. **1990-94 PRODUCTIFICATION**

The Erlang implementation so far was an interpreter written in Prolog which was acceptable for a prototype, but not for a real product. The Erlang design team then focused on implementation issues and developed a virtual machine in C which turned out to be 70 times faster than the original interpreter. This proved that concurrent functional programming could be
Introducing Concurrent Functional Programming

used for ”soft real time” system products, i.e. response times measured in milliseconds. The compiler and other ”tools” were written in Erlang itself.

Erlang was officially presented [1] at the ISS’90 (XIII International Switching Symposium) which took place in Stockholm in 1990. Erlang became recommended for prototyping purposes within Ericsson and was used to develop different demo systems, to control a photonic switch, to run cordless telephony etc.

In 1992 the decision was taken to develop Erlang into a product for use in production projects and a first project was started at EBC based on the above mentioned application prototype. The first production quality release of Erlang was delivered in October 1993.

In 1993 Erlang Systems AB was created as a an Ericsson subsidiary to market Erlang commercially and to offer training and consulting on a professional basis. Erlang Systems rapidly developed documentation and course material of professional quality and presented the following initial course program:

- Basic Erlang, 4 days,
- Interoperability, 4 days,
- Tools and libraries, 4 days,
- Advanced Erlang, 5 days.

Erlang Systems made serious attempts at marketing Erlang with tools commercially which included lecture tours in Sweden and the US and presentations at tools fairs, however, with little success. One difficulty was the lack of good reference systems at Ericsson. In 1995 Erlang Systems was made into a department of Ericsson Infocom AB.

Figure 3. Erlang Systems logotype.

The first edition of the standard Erlang textbook Concurrent Programming in Erlang [3] was published in 1993. This was a period of intensive technical developments. An ASN.1 compiler was the first telecommunications oriented ”tool” written in and for Erlang. Experiments with programming of distributed systems lead to the development of Distributed Erlang [17]. A translator [11] from SDL (Specification and
Description Language) to Erlang was developed and also a distributed real
time database [14] with transactions and query processing.

The first external delivery of Erlang was made already in 1989 (to Bellcore) and from then on Erlang has been delivered free-of-charge to universities for research, education, and prototyping. In 1995 this was made into a free distribution over the Internet.

Erlang dissemination pattern during this phase of development:
- Within Ericsson primarily for prototyping but for a few product projects.
- Outside Ericsson widely to academia but external marketing attempted.

Experiences:
- Prototyping of very different applications assured the generality of Erlang.
- External (primarily academic) contacts created a necessary reference.
- Publishing a book lifted Erlang out of a narrow Ericsson context.
- A professional unit outside the research laboratory was necessary to handle education and consulting.
- Marketing a new programming language is very difficult.

6. 1995-97 PLATFORM

In late 1995 Ericsson started a couple of important application projects which required an appropriate programming technology. The situation was very urgent since a large "platform" project recently had been closed down. CSLab made a proposal based upon:
- Commercial processors,
- Commercial operating systems,
- Erlang,
- Productified development tools (debugger, interpreter, etc.),
- A base system called SASL (System Architecture Support Libraries) inspired from the application prototype systems,
- Productification of various software including the distributed database,
- Interworking with device processors (usually programmed in C),
- Interworking with other software (protocol stacks, routing software, etc. usually written in C).

CSLab was given the go ahead with a tight schedule to produce a prototype system in six months. The system was named OTP (Open Telecom Platform) [15].

At this point external marketing of Erlang for product development was stopped since all efforts were to be concentrated on the OTP project. However, the free distribution for research, education, and prototyping (primarily to universities) continued.
A new unit was created for management, support, and further development of OTP. Technology transfer from CSLab to the OTP product unit was handled as follows:

- Already in the first prototype phase the product unit took over systems integration and release management.
- From the second development phase, the product unit took over project leadership and product management.
- Designers from the product unit joined the different design teams (for compiler, SASL, etc.) and CSLab personnel were phased out over a longer period.
- CSLab and the OTP product unit are still co-located.

By the end of 1998 the OTP product unit numbered about 20 people. With successive releases new functionality has been added such as an implementation of CORBA (Common Object Request Broker Architecture).

In 1998 there were about 14 projects ongoing based on Erlang and OTP as well as many projects just using Erlang. At the CeBit international trade fair at Hannover in April 1998 there were no less than nine Erlang based system products on display in the Ericsson stand.

![Figure 4. Number of Erlang related courses per year 1989-1999. There are about 12 pupils to each course. The first release of the Open Telecom Platform came in 1996.](image-url)
Perhaps the most significant application is Ericsson’s ATM (Asynchronous Transfer Mode) switching system AXD 301 which is scalable from 10 Gbit/s to 160 Gbit/s [9]. The basic system in one rack handles 10 Gbit/s and contains two general-purpose control processors which handle network-signaling termination, call control, and operation & maintenance (as well as smaller processors for device control).

During normal operation, one control processor handles calls while the other processor handles operation and maintenance. In addition, each processor acts as a standby for its counterpart. In the event that one of the processors should fail or be taken out of operation, the system automatically switches over to single-processor mode.

To date 250 AXD 301 systems have been delivered to 20 countries. The AXD 301 system is also an integral part of Ericsson’s ENGINE concept which has been ordered by operators such as British Telecom and Telefónica.

In 1997 the ban on external marketing was lifted and Erlang Systems recruited new marketing staff. The marketing goals were set high and Erlang/OTP was to have 10,000 users and be used in product development in 5 companies other than Ericsson by the end of the year 2001.

Figure 5. Erlang deliveries outside Ericsson per year prior to the open source. External marketing was stopped during 1994 and 1995.
However, during 1998, most possible major partners, including SUN and MicroSoft, had declined the Erlang/OTP technology. Erlang Systems then concentrated on embedded systems in a partnership with Wind River Systems with an implementation of Erlang/OTP on the VxWorks operating system and on high availability telephony in a partnership with Natural Micro Systems who have a leadership in compact PCI technology.

These partnerships would not get all the way to the goal of 10,000 users in three years, instead the strategy changed towards the open source initiative (see below).

Erlang dissemination pattern during this phase of development:

- Within Ericsson used for strategic product development.
- Outside Ericsson passively to academia. External marketing stopped but later resumed.

Experiences:

- Erlang is an excellent base for the creation of a platform for building distributed high-availability applications.
- The existence of a library of useful system (platform) components makes Erlang immensely more useful.
- A product unit outside the research laboratory was necessary to handle product maintenance, further developments, error reporting, new releases etc.
- Marketing a programming language even with platform components is still very difficult.

7. EXPERIENCES FROM ERLANG USAGE

Experiences from the use of Erlang in many sometimes very large projects indicate clearly the two different traditions within software engineering. The most successful projects are run by enthusiastic teams working hands-on producing rapid results. The prime example is the AXD 301 project which developed a small executable system very early and then continued by building successive increments, carefully adding new functionality, and all the time monitoring system performance.

Less successful has been the top-down methodological waterfall approach where several teams (perhaps spread over several countries) specify and code the whole system and then send their parts for integration test. Then there is much poorer feed-back to the designers and the whole idea of interactive programming (one of the strong points of functional programming) is lost.
8. 1998-2000 BACKLASH AND OPEN SOURCE

In February 1998, ERA (*Ericsson Radio AB*), a large and important part of Ericsson, decided not to start any further application projects based on Erlang. The primary reason for this was a fear that a proprietary programming language might lead to a dead-end and also a general effort to reduce the number of development platforms.

During the Autumn of 1998 a discussion was raised about releasing Erlang as open source in order to facilitate its spread externally and hopefully attract even Ericsson competitors to use it. A small group visited Red Hat Inc., a company that distributes Linux, and in December open source Erlang was released, web site [www.erlang.org](http://www.erlang.org).

![Number of requests per month to the open source Erlang web site. In addition there are four mirror sites in operation.](image-url)

Whereas earlier marketing efforts had tried to make a business from marketing Erlang as a programming language with implementation, the focus now changed to spreading the language and to eradicating the "proprietary" image. Compared with earlier distributions of Erlang, the open source distribution is a considerably more mature product in that it contains the full OTP implementation (SASL, database, libraries, etc.) as well.
During the first month there were 72,933 requests to the open source Erlang site. This dropped to about 40,000 requests for some months but has been climbing steadily since July 1999. The increased activity is also noticed on the mailing lists.

Erlang Systems and the OTP product unit also operate a commercial (professional) web site www.erlang.se which gives information about current courses etc. Most external users use the open source but Erlang Systems also offers support contracts. Academic licenses are also available which make all teaching material available at no charge.

![Downloads per month from the open source Erlang web site.](image)


Early in 1999 most of the original Erlang developers (including Joe Armstrong, Robert Virding and Claes Wikström) left Ericsson to set up Blutail, an external company based on venture capital. The business idea is to develop robust systems for ISPs (*Internet Service Providers*) using Erlang. Their first product was the Mail Robustifier followed by the Web Prioritizer. Blutail has delivered systems to operators like TeleNordia and are
marketing both in Europe and in the US. In the Spring of year 2000 Bluetail entered a partnership with Sendmail.

On August 28, 2000, Alteon WebSystems, a US company based in Silicon Valley, announced that they were in the process of acquiring Bluetail at a price of 152 million US$. This followed on some collaboration between Alteon and Bluetail. At the same time Alteon was being acquired by Nortel.

When 100's of units (web switches etc.) from a major US manufacturer are sold based on Erlang, this might herald a new era in the history of Erlang.

Some external users are Sendmail, the British ISP One2one, and CellPoint which specializes on building position based systems for mobile telephony. An important Swedish user is Telia Promotor which has designed a call center based on Erlang which they are marketing all through Europe.

Erlang has triggered much further technical research. A type system [13] and a program verification system [6, 7] have been developed for advanced users and experiments with capabilities [8] have been carried out to handle imported software in a secure manner. Two alternative implementation projects, ETOS (Erlang TO Scheme) at l'Université de Montréal and HiPE (High Performance Erlang) [12] at Uppsala University, are also under way.

Erlang has been much noted in the research world and in 1997 Joe Armstrong was invited speaker to the ACM SIGPLAN International Conference on Functional Programming which was held in Amsterdam. At the 12th International Workshop on the Implementation of Functional Languages in Aachen, Germany, in 2000 there was a special session on Erlang.

Every year there is an International Erlang/OTP User Conference in Stockholm. The proceedings can be found in www.erlang.se/euc.

Erlang dissemination pattern during this phase of development:
- Within Ericsson no new large projects but growing developments of the existing projects.
- Outside Ericsson external marketing was replaced by an open source release which took off after about one year. Growing use of Erlang for product development.

Experiences:
- Reorganizations in a large company can change the situation radically.
- Erlang and OTP are highly relevant also for rapid development of high-availability Internet related applications.
- Open source is a very effective method for spreading a new software technology.
- The economical basis for Erlang/OTP still are the in-house Ericsson projects.
Some external users came as an effect of encountering Erlang at courses at the university.

9. **ON THE SPREAD OF FUNCTIONAL PROGRAMMING**

Philip Wadler presents the following list of possible reasons for the resistance to functional programming languages [16]:

- Compatibility,
- Libraries,
- Portability,
- Availability,
- Packagability,
- Tools,
- Training,
- Popularity.

Most of these are self-defeating, because of the lack of X, no X will be created. All points except the last one are of a technical nature and can easily be remedied. The key point is the last one which is a bit like a *Catch 22*.

It might well be that it is difficult to introduce functional programming into an old and established company culture. This, on the other hand, leaves the field wide open for exploitation by new companies free from tradition.

10. **CONCLUSIONS**

For Erlang to be used inside Ericsson it needed to be used outside and for Erlang to spread outside Ericsson there had to be wide use of it inside. The only way to get around this was a steady spread of the language in both spheres. In fact, in the dynamic world of telecommunications, the history of Erlang has proceeded in a see-saw fashion with focus alternating between internal and external use.

The development and use of Erlang shows that for a new programming language to be reasonably successful there are, at least, the following prerequisites:

- There has to be a sizable and stable support organization.
- There has to be some niche that is sufficiently interesting and important for large sectors of industry. In Erlang's case high-availability, reliable, distributed systems and rapid design through high abstraction level and prototyping.
- The language must be reasonably simple to learn and to implement.
Table 1. Erlang history summarized.

<table>
<thead>
<tr>
<th>Year</th>
<th>Internal Usage</th>
<th>External Usage</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1984-6</td>
<td>-</td>
<td>-</td>
<td>Technology evaluations</td>
</tr>
<tr>
<td>1987-9</td>
<td>Use in prototypes</td>
<td>-</td>
<td>Experimental developments</td>
</tr>
<tr>
<td>1990-2</td>
<td>Limited use in products</td>
<td>Academic distribution</td>
<td>Presented at ISS’90</td>
</tr>
<tr>
<td>1993-5</td>
<td></td>
<td>External marketing</td>
<td>Erlang Systems established</td>
</tr>
<tr>
<td>1996</td>
<td>Use for strategic product development</td>
<td>External marketing halted</td>
<td>OTP development</td>
</tr>
<tr>
<td>1997</td>
<td></td>
<td>External marketing resorted</td>
<td>OTP development and deployment</td>
</tr>
<tr>
<td>1998</td>
<td>Nine products displayed at CeBit</td>
<td>Open source release</td>
<td>Erlang stopped at ERA</td>
</tr>
<tr>
<td>1999</td>
<td>AXD 301 and GPRS win important orders</td>
<td>Growing external use</td>
<td>Bluetail started</td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td>Open source takes off</td>
<td>Alteon buys Bluetail</td>
</tr>
</tbody>
</table>

11. COMMENTS ON TECHNOLOGY DIFFUSSION

Four important observations from the Erlang developments:

- Erlang and functional programming in general both enable and require a new way of working with much more interactivity. The top-down waterfall methodologies were designed to handle conventional programming languages. Technology and methodology both have to be changed.
- Marketing a new programming language and a new way of working requires a huge effort and investment. Twice Erlang Systems has tried marketing Erlang with limited resources and with meager results. Sun has probably spent a fortune on Java but that has paid back in the form of increased demand for computer equipment. Ericsson is a telecommunications company and selling Erlang would not sell more switches.
- Open source combined with a good support organization has meant the real break-through. Many more programmers can try Erlang and companies know that support and education are available if needed.
- It is much more difficult to gain acceptance for a new programming language than perhaps 10 or 20 years ago. C++ and Java are perceived as a standard. Perhaps it might have been easier if Erlang had been called "Concurrent Scheme" or "Concurrent Haskell".
REFERENCES


Change and Adaptive Behavior in Organizations
Innovation in Non-Competitive Environments as Typified by United States County and Local Governments

Andrew P. Jansma
Carlson School of Management, University of Minnesota, Minneapolis, Minnesota 55455
ajansma@csom.umn.edu

Abstract: This is an experience report that evolves into a survey of applicable theory that might be used for additional steps in the diffusion of artefacts into non-competitive environments. The non-competitive environment in this case is county and local government in the United States. The particular case is the office of the County Register of Deeds/Recorder, which has not seen a major change in the way business has been conducted since the 1950's. The background of the decision to develop a document image processing system being is provided, and experiences in introducing this technology are discussed. There is a discussion of how business is conducted today, the barriers to adopting new technology, and how change management methods such as Kotter, Rogers, and Tushman and Romanelli apply. There is also a discussion of different models for characterizing culture such as Meyerson and Schein, and how these apply; theories of development and change such as Van de Ven and Poole, Zipf, and Davis, Bagozzi, and Warshaw; and the need for additional research in this area.

1. INTRODUCTION

We offer this paper as an extended experience report and theory overview to call for additional research in a unique technology diffusion environment. We explore the interesting environment of non-competitive county and local governments. We do this by providing an example of a document image processing system being introduced into an environment, which has not seen a major change in the way business has been conducted since the 1950's. We describe the current environment and the barriers to adopting new technology in this environment. We overview various change
management methods such as Kotter (Kotter, 1996), Rogers (Rogers, 1995), and Tushman and Romanelli (Tushman, 1985); and look at their application in this environment. We also look at models for culture offered by Meyerson (Meyerson, 1987) and Schein (Schein, 1985), and how these apply. We examine theories of development and change of Van de Ven and Poole (Van de Ven, 1995), Zipf (Zipf, 1949), and Davis, Bagozzi, and Warshaw (Davis, 1989); looking for a theory base that provides additional steps to research in the diffusion of this technology. Finally, through the discussions we make the case for additional research in this area.

2. EXPERIENCE PROBLEM DOMAIN

We seek to analyse and define the diffusion of an advanced technological artifact (document image processing) into a non-competitive environment as typified by rural county governments. There is a need for this technology.

We have extensive experience in with this organizational environment:
1. We constructed and administered a nation-wide mail survey on this topic. (A final report of this grant-funded research is available, please contact the author)
2. While administering the survey, we conducted the onsite interviews and workflow analysis in rural counties in South Dakota, Minnesota, and Pennsylvania.
3. Using the focused results of this research, we applied for and received grants to develop image-processing technology for this environment. We were very successful in creating systems that are extremely cost effective, (they will pay for themselves in a year or less), very easy to use, and provide huge potential benefits to citizen stakeholders. (A final report of this grant-funded work is also available, please contact the author)
4. We seek to diffuse the technology rapidly on a nation-wide basis, but realize that organizational issues can prohibit or inhibit this diffusion in a cost-effective and orderly manner.
5. Although the diffusion might be phrased as a marketing question, of far more interest are the structure and moves of the non-competitive organization. What we have a chance to record has been fascinating, and we are looking forward to discovering more.

Using the information gained from the research and our experiences in this particular non-competitive environment, we seek to explore perspectives and theory that could be used to understand this interesting environment. These environments affect everyone; we are all stakeholders in some non-competitive environment enabled by government for our benefit. It could be
argued that everyone would gain if these organizations operate in a more effective and efficient manner. The eventual goal of this research is to develop a plan, based on theory, to diffuse cost-effective technology effectively and efficiently across a non-competitive domain.

3. IMPORTANCE OF RESEARCH

Local governments are facing tighter budgets, while at the same time; they are being required to comply with more state and federal regulations. County and other local government entities face difficult choices between raising taxes to increase revenue or reduce services. A third option that can provide a partial solution to these problems is the affective use of technology to become more efficient.

The use of document image processing technology presents a potential method to increase worker efficiency, in some cases reduce the number of employees, improve service, comply with federal regulations, consolidate or eliminate duplication of work by different government offices. The use of document image technology has special benefits in rural areas because of the distance between many communities and county seats.

The local governmental entities must maintain or have access to a variety of information. Some of the types of information such as marriage licenses, wills, mortgage encumbrances, deeds, veteran’s discharge information and much more must be maintained forever. Some of the information needs to be accessed frequently while other information may never be accessed again. These governmental entities spend large amount of money, space and time maintaining this information and providing the information upon request.

3.1 Document Image Processing

An electronic document image processing system is a computer-based system that converts the contents of paper documents to digitised images that can be viewed at a computer workstation. The digitised images can be held and manipulated in the memory of the computer, stored on magnetic or optical disk storage media, transmitted over networks and telephone lines and converted back to a paper image by a laser printer.

The electronic document image processing system, as illustrated in figure 1, includes a scanner to convert a paper image to a digitised image, a computer with sufficient memory and monitor graphics capability to hold and present digitised image, a magnetic or optical media drive to store images for processing and retrieval and a laser printer. Additional hardware
components may include network interfaces, erasable optical drives, optical jukeboxes, digitised cameras and other output devices.

In most settings, documents are entered into the system by a scanner. A clerical person operates the scanner much like a paper copier machine. The scanner converts the figures on paper to a sequence of binary codes representing the presence of absence of ink on the paper. The coded image is displayed on a computer workstation to allow the person scanning it to view the image and make adjustments or rescan the image. When the document is scanned, the image will be compressed to reduce storage requirements. All images are compressed for storage and decompressed for viewing.

Document image processing technology had been primarily limited to large corporations, governmental entities and other large organizations because of the cost and proprietary nature of each system. The development of more powerful desktop computers and the development of networks have lead to the design of smaller desktop-based document image systems that can be easily adapted for office use (Sanders, 1993). The hardware cost of a desktop image processing system has declined, and is expected to continue to decline. Today, most counties can purchase the image processing hardware for less that $5,000. Many counties already have personal computers that could be integrated in the system, further reducing the cost.
These costs could be recovered in labour savings and potentially generate revenue if a remote access to records was provided and a remote access fee imposed.

An effective document image processing system offers numerous benefits to rural county and municipal governments and its citizens. Some of the benefits include:
- reduced storage space needed to store large numbers of paper files.
- elimination of the need to refile microfilm and paper documents. This also eliminates the chance for misfiling or losing important documents.
- the user, either employee or citizen, can have immediate access to the document.
- worker productivity is increased by allowing users to share a document image with others and allows them to work on the same document at once.
- documents can be mailed or faxed electronically to remote locations.
- documents can be viewed or retrieved from remote locations. The retrieval from remote locations offers special benefits to rural areas when the courthouse could be 50 to 100 away. It also provides a means for state or federal offices to access county files without travelling to the site.

Information technology has progressed very rapidly over the last several decades. In the private sector, adoption of new technologies has progressed fairly rapidly. This adoption rate could be attributable to the need to remain competitive in a rapidly changing world, or that change is relatively easy to make in the private sector.

In the public sector, technology adoption and use is perceived to be far behind the curve of the private sector. The non-competitive environment allows exploration of new diffusion and adoption concepts.

In examining various theories, we are going to use as an example environment the Register of Deeds or County Recorder office in county government. The advantages of the adoption of document image processing are most evident here, and the other offices in county share most of the same attributes.

4. **THEORY BASE**

Unfortunately, there are several impediments to implementing computer technology into non-competitive environments besides the cost. From experience and research, the most common impediments to implementation appear to be inertia, political problems in the department, and training. Thomas Koulopoulos (Koulopoulos, 1995), a consultant who has many years of consulting experience in document imaging, sums up the major issue:
The largest single obstacle faced by organizations planning to implement workflow is that of existing organizational culture. Unfortunately, most information professionals do not easily accept the fact that the real issues are not technology-based but rather are mired in human factors and organizational issues. The reality is that culture accounts for 67% of the obstacles identified by evaluators and users of workflow. If these issues are not addressed, the success of workflow is compromised to a substantial degree.

Workflow is the way people do their jobs. A document imaging system, by definition, changes the way people work. Great care must be taken to specify products that will be the most effective, while trying to minimize the change required.

A government department is often sheltered from the day-to-day demands of business competition and the need to adapt technology for the business to survive. The county will be in business for a very long time and paychecks will be good. The county employee’s job has probably had a lot of commonality for decades and the employee resists large changes in this environment. Any major changes must be discussed well in advance of the actual implementation and the benefits identified. Cooperation of the employees is essential and the onus is on the department head. Most offices have more than enough work to do and some of the work is neglected because there are not enough resources to do the job.

The historical methods of introduction of technology into local government may likely be unsatisfactory, and the societal benefits of an open local government system are imperative. Our research seeks to find theory and methods to enhance the speed of diffusion and lower the cost of diffusion of technology by gaining economies of scale.

The popular press and almost every academic field have a change model associated with them. We did not find any specific model for introducing technological change into county and local governments. John P. Kotter provides an example of the type of change plan associated with business in his book, *Leading Change*, *(Kotter, 1996)* where he lists the following steps:
1. Establishing a Sense of Urgency
2. Creating a Guiding Coalition
3. Developing a Vision and Strategy
4. Communication of the Change Vision
5. Empowering Broad-Based Action
6. Generating Short-term Wins
7. Consolidating Gains and Producing More Change
8. Anchoring New Approaches in the Culture

Although we appreciate that this is a successful change cycle for management in a company, it is not a prescriptive strategy that would enable
the introduction of document imaging applications into government. Although we may have a sense of urgency and a vision, if we try to hammer these ideas into the heads of county officials, the result would probably be alienation - not change. The county officials may have gone for years and years without changing, why should they react now?

4.1 Perspective Lens of Rogers’ Diffusion of Innovations

Everett M. Rogers literally wrote the book on diffusion of innovations (Rogers, 1995). We assume that the reader is already well acquainted with Rogers’ work and there is no need to further discuss the theories here. We have experimented with Rogers’ theories and have come to conclusion that additional theory may be necessary to achieve rapid diffusion in the non-competitive environment.

4.2 Perspective Lens of Organizational Adaptation and Change

Tushman and Romanelli (Tushman, 1985), among many others, characterize organizational adaptation and change as “punctuated equilibrium”. From the punctuated equilibrium viewpoint, organizations have divergent periods of rapid and significant change, followed by longer convergent periods where the organization tries to maintain a more steady-state equilibrium.

Applying this lens to the non-competitive government environment that we are examining gives us a fascinating viewpoint – the last major shift in the core task that we are examining was in the 1950’s when paper copiers were integrated into Register of Deeds/ Recorder office. The office changed from hand copying records of transactions (real estate, mortgages, liens, etc.) into ledgers to making copies of the documents and recording only indexing information in ledgers. There have been smaller changes ordered by the state, such as forwarding a copy of the vital records such as birth and death records to a state office, but the environment has been in an long-term steady-state since then.

The case could be made that people in the office do not know how to make this kind of change, because they have never seen it. The recording process skill is generally passed from the recorder to their deputies in on-the-job training, the recorder likely has a long tenure because of the tendency of voters to re-elect incumbents, and a deputy recorder is the most likely person to replace a retiring recorder. Although most states have codified law prescribing the duties of the office, we found in almost every case that we
examined that there was some small practice that did not fit with the legal prescription. The practices did not interfere with the day-to-day practice of recording and retrieving copies of documents, and we attributed the deviations as an artefact of the transfer of process information.

Tushman and Romanelli (Tushman, 1985) also identify the need for legitimation in the political economy of the organization. Of particular interest to the author are the issues of external legitimation of the technology and how an inadvertent and incorrect perception can lead organizations who may be risk adverse to be even more so. Yet, there is information that is available that would lead to the dismissal of the incorrect perceptions by more astute viewers.

Tushman and Romanelli describe the infrastructure for external legitimation as recognition of society’s social and legal values and establishment of position with regulatory agencies. This introduces an interesting story in the legitimation process. Laws to make document imaging a legal form of recording documents were put into South Dakota law in 1986 and 1988 for the Register of Deeds and the Attorney General of the State of South Dakota. We are not aware why just these two entities are included, we assume that vendors sought to have the law changed for Register of Deeds and the Attorney General thought it would be a good idea for his office, too. Following is the applicable law for registrars:

7-9-1.1. Recording, filing, and indexing of records by microfilming or computerization. The functions of the register of deeds, including but not limited to, the recording of instruments, liens, satisfactions and releases and the filing of records, as well as the index to any such record, may be accomplished by means of microfilming or computerization, as provided in § 6-1-11.

6-1-11. Form of certain public records -- Duplicate -- Computerization. Whenever the creation, maintenance or storage of any public record is specified by state law for political subdivisions, such record may be in the form of punched cards, magnetic tapes, disks and other machine-sensible data media within a data processing system. Such records shall be backed up by a duplicate, be accessible to viewing members of the public, and be retained in accordance with all applicable requirements for the retention of manual records. To the extent an office is computerized, the office need not keep a hard, paper copy. If current public records are converted to a computerized format, the political subdivision may destroy those records which the state records destruction board has pursuant to § 1-27-19, declared to be of no further administrative, legal, fiscal, research or historical value.
When visiting with Registrars of Deeds in that state, copies of this codified law were provided as evidence that the process was indeed legitimised by the State of South Dakota. Last year, (1999) the state archivist spoke at the state convention of Registrars and made the statement that, “Document imaging is not archival”.

It seems that under the law that pertained to her office, document imaging had not been legitimated, although state agencies had been giving her documents in digital form for many years, and her office had been frantically trying to convert them all to microfilm form, hence her remark.

This statement had two effects. For many registrars, who did not want to change the way they did business, this was an affirmation of their decision. For a second group, with whom the author had been working to diffuse the technology, there was an outburst of frustration, because the statement could affect their plans because one of their rivals at home in the county courthouse could use the information, although it was incorrect, in a political battle for resources. The consensus attitude for these registrars was, “Why does the State of South Dakota have to be so backward!”

The author communicated with the archivist, and she really did not know that the registrars had enabling law that she did not have. And, the legislature quietly passed in the 2000 legislative session, and the Governor signed, a law to legitimise the technology for all the offices in the state. The archivist has not made public acknowledgement of her error, but we hope that she will soon.

This example brings forward a couple of issues. In many states, there are offices that are using document imaging that are not expressly legitimised by law, the officials are just going ahead because this is the way they have to go to be effective in their office. In the official’s mind the process is legitimised in itself, there is no need for additional state authorization. Minnesota happens to be one of these states. Yet, for the risk adverse official, which many are, the lack of explicit legitimisation by the state is one more reason not to adopt an innovative technology.

Greiner (Greiner, 1972) makes the case for five major cycles of punctuated equilibrium as the organization grows. In our experience, we found that rapid population growth in a county was a force for change; one would more likely see a document image or microfilm system in use in these counties. A possible scenario is that the rapid growth stresses the system, and when searching for solutions the county identifies the innovative technology as more effective and efficient than increasing space and/or employees.
4.3 Perspective Lens of Organizational Culture and Control

Culture is largely socially constructed as the individual interacts with his/her reference groups. In this case there are three cultures to examine: 1) the culture in the Office of the Registrar of Deeds, with the registrar and from one to five employees; 2) the culture in the courthouse, with the five to twenty different offices, depending on county size; 3) And, the culture of the registrars in any particular state. Each one of these cultures has a set of shared beliefs, values and norms and range from working at the will of the registrar, to competition for resources, to cooperation for a combined agenda.

The Office of the Register of Deeds is lead by the registrar and the tasks consist of primarily of routines (Cyert 1963), which are the behaviour in the organization. As we typified above, many of these routines have been practiced for decades, and changing them will be difficult, even if the registrar leads the change. It is almost impossible to imagine another organization environment with this kind of inertia. Another amazing point is that the routines are very similar across the country, we found the same tasks in South Dakota, Minnesota and Pennsylvania, it seemed like that there is more variation within a state than between states. Although routines are considered part of organizational learning, we use them here to help the reader understand the culture. The culture in the office is socially constructed around the routines and the deputy registrar(s) has some influence in the decision to adopt a new technology, but they do serve at the discretion of the registrar. If the registrar would decide to use new technology, such as a document imaging system, the staff size is small enough for her to change the culture without too much difficulty. In Meyerson and Martin's (Meyerson, 1987) taxonomy, this culture would be classified as the Integration Model: Culture is generally stable but leaders can successfully initiate and control organization-wide cultural changes.

The culture in the courthouse is a competitive one. Every courthouse we have visited has had some discord between two or more groups. The mainline offices such as Auditor, Treasurer, and Register/Recorder are elected positions (the Assessor, for political reasons, generally is not) and can be removed from office only by the voters. Yet, the offices compete for county resources distributed by the legislative body, the Board of Commissioners (Board). Each office presents an annual budget for legislative review, and if there are any requests for new technology, they are considered and allocated by the Board. In rural areas, it would be fair to characterize the Board as comprised of retired farmers and businessmen.
This is the culture in which the registrar operates, if the registrar wishes to introduce innovative technology into her office; she runs the gauntlet of competing with the other offices for the additional funds, getting the Board to understand the technology that she wishes to purchase, and hopefully the funds are readily available in the county budget. The author has had several registrars describe how they hate to run the gauntlet, even if they really need something.

They also dread being turned down by the board, both for the effects in the courthouse, but also because they have to be ever so much more careful when they bring the subject back to the Board again. This is different from culture described in the national survey we conducted. It appears that reports to the survey are different from interpersonal reports on a one-to-one basis. The author believes that the one-to-one reports are a more accurate representation of the culture we are studying. This is not a culture that encourages innovation. This culture would be classified as an ambiguity model in the taxonomy of Meyerson and Martin (Meyerson, 1987), where all the players are changing all the time.

Occasionally, we have seen a Board with a few enlightened leaders. Under this leadership, the culture represents the Meyerson and Meyer (Meyerson, 1987) differentiation model: Leaders’ efforts to manage change have localized impact - both intentional and unintentional - but predictable, organization-wide control is unlikely. The mainline officials are too independent to serve the Board in other ways, once they have passed the budget hurdles.

The final culture that affects the Registrars is the culture among his/her peers who hold the same office. We have seen a wide range of these types of relationships, depending on the state and the registrar’s involvement. In some states, the organizations are very active as a political body and have the respect of the Legislature and the Governor; in others, the opposite is true. In considering the adoption of technology, we would prefer the first culture, but this is something that is not easily changed from the outside.

How do we change these cultures? We can use Lewin’s (Schein 1985) unfreeze-change-freeze paradigm to hopefully make rational changes in the culture to create a new culture. The environment can also change culture in an evolutionary cycle of variation, selection, and retention in a socially constructed environment.

If the external environment can be changed in such a way that would reward resources to the county that could not be obtained any other way, then it might be possible, or even likely, that the cultures would respond. For example, if a document imaging system could be offered free, the registrar could use this information to change the culture of his/her office. The competitors in the courthouse may assist him/her because the funds would
not be coming from their possible budgets. The Board would encourage taking the opportunity because it would be easier for them; and finally, the peer group of registrars would be embarking on a journey together. The positive impact should be enhanced by establishing a collective culture with a shared sense of purpose.

Culture is also a major contributor to organizational learning, some thing that is important as the opportunity to use innovative technologies becomes more prevalent. As citizens become more acquainted with technologies such as the web, there will be more pressures to change. The organizational learning aspect of culture leads directly to adaptations of organizations, particularly to change. The interaction of culture and change leads us also to organizational structure. Culture is difficult to locate, identify and change, but it is the one aspect of organizations that appears to really make the difference.

### 4.4 Organizational Frameworks

The organization is one of the most complex artefacts of human existence. Andrew Van De Ven and M. Scott Poole (Van De Ven, 1995) analysed more than twenty different process theories of development and change in the social and biological sciences. They identified four different "motors" of change: Evolution, Dialectic (Hegelian), Life Cycle, and Teleology (Goal–Oriented). Figure 2 is a framework for the four motors.

![Figure 2. Process Theories of Organizational Development and Change (Arrows represent likely sequences of events, not causation of events)](attachment://image.png)
Van De Ven and Poole then combined the four motors to create a macro-framework that more accurately describes the behaviour that we see as an organization learns and changes. They make the case that there are more patterns available in their model to explain the various behaviours that we see.

Figure 3. Interplays of Process Theories of Organizational Development and Change (Arrows represent likely sequences of events, not causation of events)

Figure 3 demonstrates possible interplay of the various motors.

Although Van De Ven and Poole give us a more comprehensive explanatory theory, this does not help us derive an action plan for introducing technology into governments and other non-competitive environments.

4.5 Principle of Least Effort

Additional research uncovered an interesting theory by a Harvard professor, George K. Zipf, in his book entitled *Human Behavior and The Principle of Least Effort*, (Zipf, 1949). This work is most famous for Zipf’s
discussion of the Civil War. He made the case that, besides slavery, there were so many contentious issues dividing the North and the South that the easiest path to solve all the issues was to go to war.

Zipf’s theory in his own words:
We shall argue that an individual’s entire behaviour is subject to the minimizing of effort. Or, differently stated, every individual’s entire behaviour is governed by the Principle of Least Effort. (P 6)

With the plurality of definitions of behaviour, it is probably unfortunate that the Zipf used the word “entire”, for our discussions, we will assume that there are some exceptions. Zipf uses a simple discussion of travelling between two points to illustrate the start of his theory. If there are no intervening obstacles, the person will proceed by the most direct route between the two points. If there is an obstacle, for example a mountain, between the two points then the individual through trial and error will determine whether over the mountain or around the mountain is the easiest path and subsequently take that path. If a person is presented with an unknown path, the person will guess at the easiest path until they gain experience.

Of particular interest is Zipf’s discussion of tools. With reference to a craftsman working on a task, he identifies Principles of Economic Abbreviation, Versatility, Permutation, and Specialization of tools. The idea is that the craftsman would organize his tool bench to maximize the work accomplished while minimizing the effort in using the tools. The descriptive equation is \( w = f \times m \times d \), where \( f \) is the frequency of use of the tool, \( m \) or mass is the effort required to use the tool, and \( d \) or distance is the effort to access the tool. Zipf’s hypothesis is if we introduce a productive new tool, it will initially located far down the bench and as the craftsman gains experience with the effort of its use and versatility it will be moved closer on the bench to the craftsman, moving other tools down the bench. If a tool is moved down the bench far enough, it eventually may be discarded. There are interesting implications if we apply this analogy to non-competitive environments, if the tool is productive and they begin using it, we would see a shift over to the tool and the outdated methods would be replaced. The issue is then that of introducing the tool into the environment.

Zipf’s work seems obvious, but we have included it because people keep forgetting it. If the ideas are so obvious, why do we have to keep reminding ourselves?

Another take on Zipf’s principle of Least Effort predicts that most people, most of the time, are turned back by modest hurdles that they know could be overcome with effort. To be habitual, an action must be relatively effortless or carry a particularly large psychic reward. In addition, opinions
and motivations vary widely across individuals in what constitutes a “large reward.” These are ideas that the author would like to explore with this research.

Zipf’s book is 550 pages; there is not space to describe other principles, corollaries and examples here. The discussion of the theory and building of subsequent theory will be left to the further discussions.

Additional support for Zipf’s theory can be found in work by Davis, Bagozzi, and Warshaw (Davis, 1989). They review the theory of reasoned action (TRA), a popular theory used to predict and explain behaviour. TRA describes the internal beliefs and feelings of the individual as the person’s attitude; the attitude is combined with social norms (what people are supposed to think) to arrive at behavioural intention, which then is reflected in the actual behaviour.

Under TRA, we can try to change the individual’s attitude or we can try to change the social norms in the environment. In system adoption these manipulations, if successful, should reflect more acceptance of a system.

Davis had previously defined a modification of the TRA, named Technology Acceptance Model (TAM). TAM drops the normative influence on the intentional stance of the individual, and selects two internal perceived values of the technical systems as selectors of intentionality. He defines perceived usefulness (U) as the user’s subjective feeling that the system will increase his job performance within an organizational context. He defines perceived ease of use (EOU) as the degree that the user feels that the system will be free of effort. In reviewing the research, we get the feeling that much of EOU is the anticipated learning cost of effort, which drops in value as a predictor with experience with the system.

Davis constructed an experiment using adoption of a word processing tool by MBA students over a two-year period to test behavioural intention (BI) as a predictor of use and comparing TRA and TAM as predictors of BI. The results indicated that BI was indeed a good predictor of subsequent adoption, and that TAM was a better predictor of adoption than TRA. TAM predicted approximately 50 percent of the variance in BI, rising on subsequent tests.

Davis conclusions were that intentionality (BI) is a good predictor, usefulness (U) is a major determinate of the intentionality, and that perceived ease of use (EOU) is a major secondary determinate of the intentionality. This gives support to the theory of Principle of Least Effort as a determinate in the adoption of technology in the governmental sector.

Using the hypothesis that people will change if the effort to resist change is less than the effort to maintain status quo, we are going to proactively design a plan for more rapid diffusion of technology on a statewide basis. If
we are successful, we will establish a blueprint for other states to follow in implementing new technology into their local government infrastructure.

4.6 Using the Principle of Least Effort in Tool Design

We have constructed applications with a World Wide Web interface for the user that are available as prototypes for use in this research. A major part of the success of the Web has been because of its ease of use. One the goals in the design of the applications was the ability to train the typical user to use the application in an hour or less. When design issues arose, we applied the principle of least effort to try to achieve the easiest to use approach in the tool or application. The applications were also designed so that the information could be provided to other offices very easily with an inexpensive network connection. The other offices could start a browser, query the database, and look at documents without going to the originating county office. This should help diffusion throughout the courthouse. A policy decision by the commissioners/supervisors could easily be implemented to make the information available on the Internet for the consuming public.

4.7 PRIOR RESEARCH AND DEVELOPMENTS

There has been little research done in the organizational and technological environment of county and local governments, leading one to conclude that this research should be done. Particularly interesting is the diffusion of the technology.

4.7.1 National Science Foundation’s Digital Government Project

In response to congressional mandate and federal, state and local government needs, the National Science Foundation (NSF) in 1998 established a program for research in Digital Government. This program will be ongoing, providing funds for research in this most important of governmental issues. Recently, a digital government workshop, co-sponsored by the NSF and the Center for Technology in Government (CTG 1999) at the University at Albany, brought researchers and government practitioners together. There were eight issues identified at the workshop that the participants believed must be addressed in order for any digital government program to be successful. Those issues are:
Development of trusted and secure interoperable systems -- Research is needed to understand system integration in technological, organizational, and political terms.

Matching research resources to government needs -- Both theoretical research and fieldwork are needed to create practical ideas that government can use.

Better methods of information technology management -- This includes management of software development and upgrades, outsourced development, and operations and leadership.

Citizen participation -- How will the emergence of the Internet enable greater involvement of citizens in democratic processes and institutions of governance?

Electronic public service models and transactions -- The Internet's potential to offer new integrated services and self-services makes it necessary to develop new methods of authentication, record keeping, security and access.

New models for public-private partnerships -- Given the diverse players involved in delivering public services, developing effective information technology systems will require new partnerships across the public and private sectors.

Intuitive decision support tools for public officials -- With technologies and data standards that encourage information search, selection, analysis and sharing, how will leadership decisions be affected?

Archiving and electronic records management -- Now that most information is stored in electronic files, issues such as record definition and content, version control, and public access affect how government functions.

The NSF actions give the indication that there will be research funds available for research into innovation and diffusion of technology in the government environment. As citizens demand more of their county officials, the near future should be interesting.

5. CONCLUSION

We have described a unique environment for the diffusion of technology. We have also described a survey of theory that could be used as a basis for research efforts in diffusing a tool that we have constructed. We have found some merit in using "The Principle of Least Effort" (Zipf, 1949) as a basis for further steps in this environment (we have submitted grant proposals to provide free systems and training).
We invite any comments that might be appropriate in this context. We also encourage others to explore research in this area, and we will assist anyway we can.

REFERENCES

CTG. (1999, March), Some Assembly Required: Building a Digital Government, Center for Technology in Government, University of Albany, SUNY, Albany, NY. The report is also available at the following web address: http://www.ctg.albany.edu/.


In Search of an Efficient EDIcebreaker

Use of Electronic Marketplaces for the Diffusion of EDI Among SMEs

Helle Zinner Henriksen
Copenhagen Business School, Denmark

Abstract: Although EDI is considered to be an old technology, many firms have invested considerable resources in EDI. During the last decades, SMEs have implemented EDI in order to meet the demands of their business partners. However, the SMEs do not derive the full benefit of their investments because they use EDI with too few business partners. One problem in this context is that the EDI users are invisible and isolated to each other. With the common use of the Internet among businesses, electronic marketplaces become increasingly accessible. This paper suggests that a solution to the problem of the isolated EDI users can be found within electronic marketplaces. As for the bilateral EDI arrangements and the multilateral electronic marketplaces they have two characteristics in common: (1) both are based on electronic data transactions over telecommunication networks, and (2) both have so far proven to be most suitable for commodities and standardized products.

1. INTRODUCTION

Previous research has recognized that the use of IT in organizations can reduce coordination costs and reduce transaction risks (Clemons, et al., 1993). In a broad sense the use of IT for transaction coordination has been referred to as interorganizational Information systems (IOIS). One specific IOIS tool is EDI (Electronic Data Interchange). EDI allows business partners to make commercial transactions by sending and receiving digital documents over telecommunication networks (Raymond and Bergeron, 1996). Several studies have found that EDI gives the opportunity of short transaction time for messages, high data quality, and integration of data (Jones and Beatty, 1998; Cox and Gnoneim, 1996; Arunachalam, 1995). One aspect in regard to EDI that has to be stressed in this context is that as a natural consequence of
the perpetual development of IT better, smarter, and cheaper business-to-business e-commerce solutions have without doubt been developed. However, many companies have invested a large amount of resources in EDI and, therefore, they are interested in getting some payoff from these investments. This paper will concentrate on those investments made in the particular technologies supporting EDI transactions.

EDI is, similar to other interactive media, a medium from which benefit is derived only if there is broad access. A critical mass is essential (Jones and Beatty, 1998; Premkumar et al., 1997; Iacovou et al., 1995). Universal access is optimal (Markus, 1987) but not necessary; however, the more users the EDI community consists of the greater benefits (Mukhopadhyay et al., 1995). Literature has identified that especially small companies do not derive the benefits from EDI because they still have to maintain their traditional paper-based routines, since they only exchange EDI messages with a few customers (Ramamurthy et al., 1999).

Large companies, which have implemented EDI, often become known in the EDI landscape by different degrees of power. Power, in the context of interorganizational relationships, can be defined as "the capability of a firm to exert influence on another firm to act in a prescribed manner" (Hart and Saunders, 1997). Distinctions can be made between persuasive and coercive power (Hart and Saunders, 1998) and between competitive pressure and imposition by trading partners (Iacovou et al., 1995). Some industries such as the automotive industry have succeeded in forcing the majority of their suppliers to use EDI (Tuunainen, 1998; Mukhopadhyay et al., 1995).

However, the benefits of power are not available to small and medium sized companies (SMEs). Although they want to achieve gains from their EDI investments, they are not in a position to exert power toward their business partners and less so toward their customers, where an assertion of power would be essentially useless. In order to examine whether small companies are ready and willing to exchange EDI messages with other small companies, data was collected among the customers of one such small firm. The data shows that one out of four respondents is both ready and willing to start an EDI partnership. Consequently, we deduce from the data that there is a lack of knowledge among the firms regarding whether or not their business partners already use EDI. In this paper, we will address this phenomenon as the invisibility problem. Although research has shown that small companies gain from establishing EDI connections with EDI champions (Lee et al., 1999), it can be questioned whether the EDI investments could be utilized more efficiently in the small companies.

With the increased use of the Internet among businesses, electronic marketplaces might offer a solution. An electronic marketplace is an interorganizational information system that allows the participating buyers
and sellers to exchange information about process and product offerings (Bakos, 1991). With focus on the steel industry some existing electronic marketplaces are examined in order to find out what they offer to buyers and sellers. The purpose of the examination is to predict the suitability of the existing electronic marketplaces for SMEs that use EDI. Based on literature within information systems research on EDI, the paper will discuss the incentive for SMEs that have already invested in EDI to move to the electronic marketplace.

The organization of the paper is as follows: The next section presents data from a survey that aimed at investigating the willingness among SMEs to use EDI with each other. From data it is deduced that there is an invisibility problem. Along with the presentation of data, a short presentation of the company is given and a description of the data collection methodology. The following section gives an overview of basic concepts and theory of electronic marketplaces. A section follows which presents the link between EDI and electronic marketplaces. The next section describes elements of some existing electronic marketplaces for steel and machinery. Finally, a conclusion and directions for further research is outlined.

2. THE INVISIBILITY PROBLEM FOR SMEs IN THE EDI LANDSCAPE

Regardless of whether their core activity is manufacturing or wholesale, SMEs will typically have hundreds of suppliers and even more customers. Prior studies have shown that the adopters of EDI often have very few EDI links (Horluck, 1996) even when the EDI transactions are based on a global standard such as EDIFACT (Andersen et al., 2000). However, the implementation of EDI is a large investment both in direct purchases, such as hardware and software, and in terms of human resources and training of employees (Ramamurthy et al., 1999; Arunachalam, 1995). When the SME only uses EDI with a few business partners, the investment in technology and training is underutilized.

If a global standard for EDI (e.g. EDIFACT) is used in a company, the economic cost of including other business partners is relatively low. However, how is the company to know whether a particular business partner already uses EDI, and if it is the case whether the company is interested in exchanging EDI messages? As mentioned above, we refer to this issue as the invisibility problem because nobody really knows who is using EDI (even within a business sector that has a limited number of participants).
3. DATA COLLECTION AND RESEARCH METHODOLOGY

3.1 Background for the data collection

During the summer of 1999, an EDI connection was established between a small subsidiary company and its parent company. The subsidiary company is a wholesaler of products to the steel and machinery industry. Fifty-six people are employed in the company. To this firm the implementation of EDI with the supplier was motivated by the prospect of administrative savings, since 40% of all purchase orders and invoices are exchanged with the holding company. After overcoming minor initial obstacles from integrating the EDI messages with the SAP/R3 system, the EDI traffic runs without problems. So far the exchange of EDI messages includes orders, order confirmations, and invoices. The company does not yet exchange EDI messages with other business partners; however, the intention is to extend EDI to their customers as a next step.

3.2 Research methodology

During October 1999, a survey was performed among 139 of the company’s key customers which, the company for strategic, economic or other reasons would like to keep a business-relationship with. The selection of key-customers was based on two criteria: a) large-scale purchases during the last year; or b) a strong potential for large-scale purchases in the future. The aim was to find out if the preferred customers already used EDI and (if they did) whether they were interested in exchanging EDI based on the EDIFACT standard. We also asked whether those business partners who did not yet use EDI were interested in exchanging EDI messages.

One hundred thirty-nine questionnaires were mailed to the preferred customers. The receiving company’s name was printed on the questionnaire. Along with the questionnaire was a letter describing why and how the company would like to use EDI based on the EDIFACT standard with the customer. The only means for response suggested on the questionnaire was fax. The questionnaire, which had six questions, consisted of only one page. Before the questionnaire was distributed among the customers it was discussed among a few people internal in the organization.

Fifty-nine of the 139 questionnaires were returned, all by the means of fax; 58 of the returned questionnaires were valid. Since all the questionnaires could be identified from the name printed on the questionnaire itself, it was possible to add data from the SAP/R3 database regarding exact number of
In Search of an Efficient EDIcebreaker

order lines purchased during the last year and the total value of purchases between the company and the respondent. The variable on the number of order lines is in relation to EDI of greater interest than the value of purchases, because one of the benefits of EDI is to be found in the volume of exchanged messages. Volume, or number of order lines exchanged, is a measure of the tactical value of the improvements of an organizations business process (Massetti and Zmud, 1996). These data combined with data from questionnaires are the foundation for the further analysis.

3.3 Findings

Following is a summary analysis of some basic business transactions for the preferred customers. For all the preferred customers, the company supplied the yearly number of order lines and the total value of all orders from their SAP/R3 database. The average value per order line was also calculated.

<table>
<thead>
<tr>
<th>Responded questionnaire</th>
<th>#</th>
<th>Mean number of order lines</th>
<th>Median number of order lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>55</td>
<td>272,109</td>
<td>172,000</td>
</tr>
<tr>
<td>NO</td>
<td>80</td>
<td>114,388</td>
<td>35,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Responded questionnaire</th>
<th>#</th>
<th>Mean value of order</th>
<th>Median value of order</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>55</td>
<td>1,034,000</td>
<td>509,689</td>
</tr>
<tr>
<td>NO</td>
<td>79</td>
<td>302,521</td>
<td>132,421</td>
</tr>
</tbody>
</table>

There is a statistical significant difference between respondents and the non-respondents with respect to the number of order lines (Kruskal-Wallis H = 16.4 with a p-value of 0.000051) and also with respect to the value of purchases (Kruskal-Wallis H = 21.0 with a p-value of 0.000005).
Table 3. Average value per order line in DDK versus respondent

<table>
<thead>
<tr>
<th>Responded questionnaire</th>
<th>#</th>
<th>Mean average value per order line</th>
<th>Median average value per order line</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>55</td>
<td>5,915</td>
<td>3,183</td>
</tr>
<tr>
<td>NO</td>
<td>79</td>
<td>7,182</td>
<td>1,945</td>
</tr>
</tbody>
</table>

A test of difference between respondents and non-respondents with respect to the average value per order line resulted in a Kruskal-Wallis H = 4.0, giving a p-value of 0.05. The difference in average value per order line is DKK 1270.00 (about USD 160) and although this difference is statistically significant at the 5% level, it is not considered to be of any practical importance.

Approximately one out of four of the respondents replied in the survey that they were interested in exchanging EDI messages with the company.

Table 4. The interest among respondents to exchange EDI messages

<table>
<thead>
<tr>
<th>Interested in exchanging EDI</th>
<th>Frequency</th>
<th>Percentage</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>13</td>
<td>22.4</td>
<td>22.4</td>
</tr>
<tr>
<td>NO</td>
<td>45</td>
<td>77.6</td>
<td>100.0</td>
</tr>
<tr>
<td>Total</td>
<td>58</td>
<td>100.0</td>
<td></td>
</tr>
</tbody>
</table>

This survey reveals that presently at least 13 and maybe as many as 31 of the company’s preferred customers are interested in exchanging EDI messages with the company. It is our belief that although there is a statistically significant difference between respondents and non-respondents regarding the average value per order line, the difference has no practical importance. Therefore, it is plausible that there are as many non-respondents that are interested in exchanging EDI messages as there are among the respondents (0.224 * 139) ≈ 31.) These 13 to 31 companies have so far been invisible to this specific company.

Though the sample is small, it is our belief that these data reflect a general problem within SMEs. The SMEs have invested in EDI in order to fulfill demands from their business partners, but they do not use EDI in an optimal way because most of their customers and suppliers do not know that it is possible to exchange EDI messages with them. SMEs therefore often only exchange EDI messages with a few business partners. Some kind of transparency is needed to solve this problem of invisibility in the EDI landscape.
It is clearly not an efficient solution that all companies within a business sector distribute hundreds of questionnaires to their business partners in order to find out whether they use EDI, and if so are they then interested in exchanging EDI messages. A survey using a questionnaire is in most cases a very inefficient, impractical, and expensive way of obtaining that kind of information. Therefore, a more general solution to the problem of invisibility has to be found that holds the possibility of creating transparency for all potential EDI users. One solution to the problem of invisibility could be to move towards electronic marketplaces. Though the electronic marketplaces do not directly solve the invisibility problem they do hold the possibility of breaking the isolation and they do create an opportunity for SMEs to utilize their investments in EDI with a larger number of customers and suppliers.

4. ELECTRONIC MARKETPLACES

Through the 1990s, the improvements of the electronic communication tools such as the Internet have supported the idea of electronic marketplaces. This has led to a new era for businesses (Bakos, 1997) where it is affordable and relatively easy for SMEs to join the electronic marketplaces. The idea of electronic marketplaces was inspired by the concepts embodied in transaction cost theory. The terms electronic markets and electronic hierarchies were first introduced by Malone et al. in 1987 (Malone et al., 1987).

Building on the substantial work of Coase (1937) and Williamson (1975) on transaction cost theory, Malone et al. suggested that technological progress within information systems had made electronic interconnection, and thereby electronic markets, possible. They argued that the use of electronic interconnections was a result of three forces: (1) the electronic communication effect, which is facilitated by the technologies that have reduced both the time and cost of communicating information; (2) the electronic brokerage effect, where many different buyers and sellers are connected through a central database; and (3) the electronic integration effect, where the information technology is used to reuse data in different business processes, e.g. the use of EDI (Steinfield et al., 1995). Malone et al. found it likely that these three forces would lead to a decrease in the unit cost of coordination. They suggested that the benefits of electronic markets are most favorable when both asset specificity and the complexity of product descriptions are low and when there is, in principle, access for an unlimited number of sellers and buyers; that is, when the only restriction on the market is the law of supply and demand (Williamson, 1975).
5. ARE ELECTRONIC MARKETPLACES OF ANY RELEVANCE TO EDI USERS?

A distinction has been made between bilateral and multilateral interorganizational information systems (Choudhury, 1998). In bilateral systems, individual links are made between customer and supplier. An EDI link is an example of a bilateral system. In multilateral systems, firms get access to a large or unlimited number of trading partners. An electronic marketplace is an example of a multilateral system. Traditionally, when firms have established EDI connections, it has been to gain operational benefits such as fewer errors and shorter lead times (Arunachalam, 1995). However, the strategic benefits have also played a considerable role, including benefits such as the establishment of long-lasting business relations and closer ties with customers (Fearon and Philip, 1998; Dearing, 1990). In some cases, the implementation of EDI has locked in users, especially if a proprietary EDI standard has been used. These strategic benefits of EDI raise the question of whether SMEs will be interested in moving from a bilateral system to a multilateral system; that is, from close EDI connections with few well-known business partners to an open electronic marketplace.

Because of the considerably high cost of establishing EDI connections, e.g., investments in timely negotiations on EDI protocols (Hart and Saunders, 1998), long-term business relations characterize most EDI connections. In general, IT investments for the purpose of coordination will be made with long-term suppliers for at least three reasons: (1) time to recoup investments, (2) learning curve effects, and (3) incentives to support long-term contracts (Clemons et al., 1993). The cost of establishing the connection itself to an electronic marketplace is insignificant. (See next section).

We have however, not established a connection between EDI users and electronic marketplaces. Bilateral systems and close long-term business relations characterize EDI connections whereas multilateral systems and isolated transactions characterize electronic marketplaces. However, EDI and electronic marketplaces have two important characteristics in common: (1) both are based on electronic data transactions over telecommunication networks, and (2) both have so far proven to be most suitable for commodities and standardized products. Based on these two characteristics it is our claim that a link between EDI and electronic marketplaces can be established. And it is therefore our belief that electronic marketplaces are of relevance to those EDI users that are invisible and thereby isolated to other EDI users.

Open telecommunications networks, such as the Internet, are a prerequisite for electronic marketplaces (Steinfield et al., 1995). As shown in the section below, electronic marketplaces use the Internet as a means of
establishing and sustaining networks. EDI allows business partners to make commercial transactions by sending and receiving digital documents over telecommunication networks (Raymond and Bergeron, 1996). Until recently, one of the major problems for establishing networks based on EDI has been that the company in most cases had to subscribe to a costly Value Added Network (VAN). However, over the past few years the Internet has proved to be a less-costly and less-complicated alternative (Hart and Saunders, 1998; Muller, 1998). As a consequence it must be expected that in the future companies will choose to perform their EDI transactions via the Internet instead of via a VANs-operator or a direct connection.

There are a number of often cited examples in the literature of the kinds of goods traded on electronic marketplaces, including aircraft parts (Choudhury et al., 1998), airline tickets (SABRE) (Bakos, 1991), different types of procurement such as MROs (Berryman et al., 1998), and hospital supplies (Steinfeld et al., 1995). A common characteristic of these examples is that they are commodities or standardized products. Malone et al. (1989) have put it in a very straightforward way, "One way sellers can decide if electronic markets are likely to be useful in their industries is to consider whether customers can make purchase decisions based on information in a computerized database." As for EDI, messages have to be machine readable and data has to be unambiguous in relation to content, meaning, and format (Horluck, 1994). This requirement of highly structured protocols (Kalakota and Whinston, 1997) limits the beneficial area for EDI transactions to commodities and standardized products. Research has especially shown broad implementation of EDI in the automotive industry (Tuunainen, 1998; Mukhopadhyay et al., 1995), and in the grocery sector (Andersen et al., 2000). Commodities and standardized products characterize both these sectors. Even though EDI is useful for exchanging business information regardless of whether the item is a commodity or highly specified literature and practice have so far only concentrated on the commodities. Probably due to the above mentioned limitations on content, meaning and format.

In most companies the electronic connection is already there. A survey among Danish companies reveals that 87 percent of the companies had an Internet connection in 1999 (Ministry of Research and Technology, 2000). It also seems as if the target is the same for both the electronic marketplaces and EDI usage: commodities and standardized products. Thus the electronic marketplaces should be a good place for the invisible EDI users to go in order break the isolation and achieve a broader use of their EDI investments.
6. EXAMPLES OF ELECTRONIC MARKETPLACES WITHIN THE STEEL INDUSTRY

To get an overview of the support and services some of the electronic marketplaces offer to sellers and buyers, four electronic marketplaces for raw steel and steel products in USA and Denmark are presented in the following section.

The selection of the sites was based on the criterion that the site be a marketplace for raw steel and steel products. The list is by no means exhaustive. Nonetheless, the sample gives an idea of what activities the electronic marketplace for steel supports. All data are obtained from information available on the respective web sites and the links within the web sites. All data were collected during April 2000.

<table>
<thead>
<tr>
<th>Name of marketplace</th>
<th>URL</th>
<th>Established</th>
<th>Type of marketplace</th>
</tr>
</thead>
<tbody>
<tr>
<td>MetalSite</td>
<td>Metalsite.com</td>
<td>August 1998</td>
<td>Neutral third party</td>
</tr>
<tr>
<td>e-STEEL</td>
<td>Esteel.com</td>
<td>September 1999</td>
<td>Neutral third party</td>
</tr>
<tr>
<td>Metalexplorer</td>
<td>Metalexplorer.com</td>
<td>January 2000</td>
<td>Neutral third party</td>
</tr>
<tr>
<td>Industrilink</td>
<td>Industrilink.dk</td>
<td>1989</td>
<td>Controlled by sellers</td>
</tr>
</tbody>
</table>

In table 6 below, some features for the four electronic marketplaces are described. The information is broken into three segments: (1) on-line transactions, (2) requirements for entering the marketplace, and (3) price for performing a transaction on the electronic marketplace.

1. An electronic marketplace supports one or more of the following market-making functions: identification, selection, and execution (Choudhury et al., 1998). The information on degree of support is chosen to show the extent of information generated during the process on the electronic marketplace. If a site only serves as a broker, electronic data is necessarily not generated in the process; at least not in the regime of the electronic marketplace. If, on the other hand, the parties negotiate, generate a purchase order, and arrange shipping and payment valuable electronic data is generated that is then suitable for the IOISs of the seller and buyer.

2. The information on requirements for entering the electronic marketplace site is included in order to measure the actual cost of entering, which must be expected to be relevant, especially for SMEs.
3. The argument for including the cost of performing transactions in the marketplace is to document that using electronic marketplaces is not free. Although using the electronic marketplace reduces the transaction costs, the cost of operating at the marketplace has to be considered.

<table>
<thead>
<tr>
<th>Site</th>
<th>MetalSite</th>
<th>e-STEEL</th>
<th>Metalexplorer</th>
<th>Industrilink</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-line transactions</td>
<td>Find product</td>
<td>Initiate</td>
<td>Initiate</td>
<td>Send purchase</td>
</tr>
<tr>
<td></td>
<td>Find customer</td>
<td>Specify</td>
<td>Specify</td>
<td>order via</td>
</tr>
<tr>
<td></td>
<td>Negotiate</td>
<td>Target</td>
<td>Negotiate</td>
<td>WWW</td>
</tr>
<tr>
<td></td>
<td>Buy</td>
<td>Negotiate</td>
<td>Complete the transaction.</td>
<td>EDI</td>
</tr>
<tr>
<td></td>
<td>Send purchase order</td>
<td>Close</td>
<td>The electronic infrastructure</td>
<td>for EDI and e-commerce is provided.</td>
</tr>
<tr>
<td></td>
<td>Arrange shipping</td>
<td>Steeldirect:</td>
<td>Provides the ability to target specific groups for online business.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Track order</td>
<td>Provides the ability to target specific groups for online business.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Online purchase orders delivered by e-mail, EDI or fax.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td>Membership required. There are no membership or application fees.</td>
<td>Membership required. There are no membership or application fees.</td>
<td>Membership and use is free of charge if you subscribe as a Buyer member. Subscription as a Seller member gives unlimited access and no transaction fees. Membership fee is Euro 590/quarter.</td>
<td>Membership required. There are no membership or application fees for buyers. Sellers have to have EDI connections. Initially fee is Euro 270/quarter.*</td>
</tr>
<tr>
<td>Cost of Transaction</td>
<td>Free to buyers. Charges the seller from 1/4% up to 2%, for each sale that is completed online.</td>
<td>Free to buyers. Charges the seller a fixed transaction fee of 0.875% of the value of the transaction.</td>
<td>None.</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

* To become a seller of Industrilink.dk an e-commerce strategy has to be made within six months and implemented within twelve months. Until an e-commerce strategy is implemented the monthly fee for sellers is Euro 270.
Data from the web sites of four electronic marketplaces for raw steel and steel products shows that the sites to a large extent support business transactions. Two of the sites make it explicit that the data generated in the business process is available in EDI format. Regarding the requirements, none of the sites charge buyers anything to become members, but all the sites require that buyers go through an application procedure. The two Danish sites charge sellers a fee to join the marketplace. The Industrilink-site places considerable demands on the sellers. The sellers have to make up an e-commerce strategy that has to be implemented within one year. So far Industrilink.dk has 14 sellers connected to the site. The two U.S. sites charge sellers a percentage fee for each transaction. No data were available for Industrilink.dk regarding transaction fees. Metalexplorer.com does not charge a transaction fee.

7. CONCLUSION AND DIRECTIONS FOR FURTHER RESEARCH

The electronic marketplaces and EDI have common characteristics that should be utilized in the future: both have so far mostly been used for commodities and standardized products and they both base their transactions on electronic data via telecommunication networks. Data supports the argument that the EDI users are invisible and thereby isolated in the business environment. They do not, therefore, get an optimal return on investment in their EDI systems, because the number of business partners with whom they exchange EDI messages are limited to a few. The electronic marketplaces offer an open marketplace with the possibility of many players. The electronic marketplaces have a high level of business support regarding services of all kinds, including EDI support. The efficient EDI-breaker could therefore be found within the electronic marketplaces. Turning to the theory of diffusion of innovations it is likely to predict that the move is possible.

The diffusion of innovations is a process where “an innovation is communicated through certain channels, over time, among the members of a social system” (Rogers, 1995). According to Rogers the innovation can be an idea, practice or object, which is perceived as new among the group of users. To EDI users who are used to bilateral interorganizational information systems it would be a new practice if they included the multilateral electronic marketplaces in their business-performance. In determining whether it is likely that the change will take place the five perceived attributes of innovations; relative advantage, compatibility, complexity, trialability, and observability can be considered.
Relative advantage is the degree to which an innovation is perceived as being better than the idea it supersedes (Rogers, 1995). The relative advantage can be divided into tangible benefits and intangible benefits (Premkumar et al., 1994). The adoption of electronic marketplaces within organizations that use EDI has the possibility of gaining tangible benefits if business can be performed to a higher degree via electronic means. Compatibility is the degree to which an innovation is perceived as being consistent with the existing values, past experiences, and needs of the potential adopter (Rogers, 1995). If the use of electronic marketplaces is compatible with the sociocultural values, previous introduced ideas, and the company's need for innovation then the move is possible. Data can support the argument that an increased use of the EDI systems in the organizations is relevant, and that the company is willing to adopt the innovation. Complexity is the degree to which an innovation is perceived as relatively difficult to understand and use (Rogers, 1995). As mentioned previously do most companies have an Internet-connection already, and are therefore familiar with the media that host the electronic marketplaces. Those companies that have implemented EDI have without doubts already spend a substantial amount of effort in training employees. Trialibility is the degree to which an innovation may be experimented with on a limited basis (Rogers, 1995). In the case of electronic marketplaces there is a rich opportunity to try out the innovation on a limited basis. As shown in table 6 there is close to free access to the electronic marketplaces within the steel and machinery industry. Observability is the degree to which the results of an innovation are visible to others (Rogers, 1995). It is questionable if the move to an electronic marketplace is observable to others. In most cases the pool of actors within the electronic marketplace is hidden. Though the move to the electronic marketplace is not directly observable among the EDI users it is never the less hard to neglect the fact that the four previously mentioned determinants for adoption of an innovation are highly represented in the case of EDI users entering the electronic marketplaces.

However, the move from a pure EDI environment towards a mix of EDI and electronic marketplaces is not only a question of fit to existing organizational structures and willingness to innovation, common modes of transportation, and goods that are easy to categorize; strategic considerations also play a vital role. As pointed out earlier, the implementation of EDI among business partners can serve as a lock-in mechanism. When moving from a close EDI partnership towards open markets, the strategic advantage of closer ties to business partners disappears. An important task for future research is therefore to look at whether the operational advantages of EDI
are of greater importance than the strategic advantages, especially in the context of electronic marketplaces.

REFERENCES


Horluck, J. "EDI and Danish Companies." Dansk Industri, 1996.


In Search of an Efficient ED Icebreaker


Process Definition in Web-Time
*A Fast Start for a Fast-Moving Company*

Alan S. Koch
*Independent Consultant*

**Abstract:** In small entrepreneurial companies, the most critical things are speed and flexibility. We must be able to move quickly, react to both opportunities and threats, and make new product available to customers on a regular basis. In an environment like this, can we afford to spend the time to focus on developing and improving our software development processes? And can we afford the overhead involved in following well-defined processes?

A better question would be: "Can we afford not to address our software development processes?"

This paper will focus on the initial process definition work that I helped a young Internet company to get started on. I will chronicle the steps we took to get a quick start on the process definition they so desperately needed, and how we were able to achieve usable improvements in a relatively short time. We will discuss the challenges we faced and the things we did that helped the project along.

1. BACKGROUND

After 13 years at the Software Engineering Institute (SEI) and a year as Manager of Software Process & Quality Assurance at a small company, I had just begun to market my services as an independent consultant in Software Development Processes. With my recent experiences, I was especially well equipped to help small and growing companies. I had learned a lot about what not to do, and even had some familiarity with strategies that work.

The Subject Company was classified as an Internet company. It was a few years old and had just made its Initial Public Offering (IPO) late in
1999, after my first contact with them. They were experiencing the growing pains that are common to companies of this type. For example:
- Missed deadlines in spite of heroic effort
- Surprise features and missing features
- Last-minute requirements changes
- Code with serious problems delivered to test
- Growing animosity among departments

They engaged my services because they knew that their ad-hoc processes were causing them severe problems. And they were smart enough to know that the problems would only get worse as they continued to grow.

2. CHALLENGE #1: GETTING STARTED

Getting started should have been easy. The champion within the company for software process improvement was also the sponsoring executive; an enviable situation for any process improvement effort! And there was no significant resistance to the idea of process improvement from any level of the company. In fact, everyone seemed to be genuinely excited by the prospect of fixing the processes that were causing so much pain.

In spite of this surprising organizational eagerness, getting started turned out to be the most frustrating part of the entire project. My first contact with the company was on 15 October 1999, but we did not set the date for the kick-off activities until 14 January 2000 – a full three months later.

The problem wasn’t resistance or lack of interest; rather it was a matter of getting the right people together to discuss, agree and approve the proposed actions. Executives in this sort of company are routinely stretched so thin that redirecting their attention to an important but not urgent subject takes time and persistence. Even the sponsor of the effort had to be regularly reminded that she needed to make things happen.

In retrospect, I am not sure that I could have taken actions that would have significantly accelerated this part of the project. I believe it is just a cost of doing business with a fast-moving Level 1 organization.

The lesson from this challenge consists of patience and persistence. Begin the work with the expectation that some decisions will be slow in coming. But at the same time, be persistent. Once you have an executive sponsor, it is imperative that he or she keeps the subject on the table with the rest of the executive team until they agree on the appropriate actions.
3. THE KICK-OFF

The process definition effort was finally kicked off on 3 February 2000. The objectives of the kick-off were to:
- Bring everyone to a common understanding of the goals of the process improvement effort,
- Provide a very brief introduction to the CMM (just enough so that everyone could participate in the rest of the kick-off activities),
- Generate discussion of the problems that the various parties were experiencing in their current software development processes (so I could understand the biggest issues they were facing at the time), and
- Administer the SEI assessment questionnaire (which would provide a basis for the initial process improvement strategy).

The company was split into two parts for the kick-off event:
- In the morning, the engineering staff (including Tech Writers and QA) participated, and
- In the afternoon, the management and marketing folks went through it.

This was done for several reasons:
- To make the best use of each person’s time (this is a significant issue for a small fast-moving company),
- To allow the presentation and discussions to be focused differently for the two audiences (engineers were more interested in how the CMM would affect their work, and managers were more interested in the business case for the project), and
- To assure that everyone would feel free to speak openly about the problems they perceived in their current practices.

The kick-off event was successful in all respects. The only problem we experienced was that the managers and marketing folks required almost three weeks of badgering before they all had submitted their assessment questionnaires. Naturally, this is part of the problem discussed under “Challenge #1”, above.

The lesson from the kick-off is simply the value of doing it. Although there was wide agreement that a process improvement project was called for, there was such a diversity of understanding of what that meant that making any real progress toward the goals would have been very difficult. Bringing everyone together, learning a new vocabulary, and discussing the symptoms of their process problems established momentum for the harder work that would follow. The kick-off also highlighted senior management’s commitment to the process improvement work; committing everyone’s time to the event made it clear that process improvement is very important to the company’s future.
4. CHALLENGE #2: INTERPRETING THE MINI-ASSESSMENT

In a full CMM-Based Assessment for Internal Process Improvement (CBA-IPI), the assessment questionnaire is not the main source of information, rather it is used to focus the discussion on potential issues during interviews and other assessment activities. In a Mini-Assessment, the questionnaire (although it is supplemented with information gleaned from various discussions) becomes the main source of information about the company's process maturity and issues. The challenge in a Mini-Assessment is in making sense of the results of the questionnaire.

The SEI's questionnaire is organized by Key Process Area (KPA), and each question is roughly related to one of the goals for a KPA. For each question, the responder is asked to check:

- "Yes" (the goal is satisfied),
- "No" (the goal is not satisfied or only partly satisfied),
- "Does Not Apply" (the goal does not apply to this organization), or
- "I Don't Know"

and space is available for written comments.

For any particular question, some people answered "Yes", others answered "No" and others checked another answer or even left it blank. Given this variety of opinion and lack of options for verifying them, how could we come up with a coherent picture of the organization's process maturity from this data? I chose to use a two-dimensional view for each question:

- How highly they rated themselves – What percentage of the people who answered "Yes" or "No" answered "Yes"? This was computed as (#Yes / (#Yes + #No)). This rating gave us a sense of how likely it may have been that the organization actually satisfied the goal by indicating the percentage of the employees who thought it was satisfied.

- How strongly they hold that view – What percentage of all of the participants answered "Yes" or "No"? This was computed as ((#Yes + #No) / #Participants). This rating told us what portion of the organization felt they could answer the question, indicating how much weight we could give to the first rating.

For each KPA, I combined the results of the individual questions for a single composite view. Although teams working on a specific KPA would find the goal-by-goal scores useful, the objective of this exercise was to get a coherent view of the entire organization, so that level of detail was not necessary.

That analysis resulted in the chart in Figure 1:
Figure 1. Mini-Assessment Results

- Each black line with a square on it gives a composite view of the “How highly they rated themselves “scores for a KPA. It is a range-plot that shows the highest, lowest and mean ratings of the questions for that KPA.
- Each lightly-shaded bar gives a composite view of the “How strongly they hold that view” scores for a KPA. It shows the mean of the strengths for the questions for that KPA.
- The small circles highlight the three highest-rated KPA’s, and the small diamonds highlight the five lowest-rated KPA’s.
- For an example of reading this chart, refer to the first column, “Requirements Mgmt”:
  - The top of the black line is at almost 75%, meaning that on the best question for this KPA, nearly 75% of the people who answered “Yes” or “No” answered “Yes”.
  - The bottom of the black line is at about 12%, meaning that on the worst question for this KPA, only about 12% of the people who answered “Yes” or “No” answered “Yes”.
  - The black square on the line is at about 30%, meaning that on average, only about 30% of the people who answered “Yes” or “No” to questions for this KPA answered “Yes”.
  - The shaded bar goes up to about 80%, indicating a strong opinion with an average of 80% of the participants answering “Yes” or “No” to the questions for this KPA (as opposed to “I Don’t Know”, “Doesn’t Apply” or blank).
But the numerical score was only part of the result. The written comments also provided important information for interpreting the results. The most important of that information concerned people’s misunderstandings about the instructions for the questionnaire and the terminology used in the questions.

The net result of those misunderstandings was that most of the ratings should have been lower than shown on the chart. Here are some examples:

- "Yes – but..." – Quite often, people checked “Yes”, then explained in the comments how those things were not done regularly or under certain conditions. The questionnaire instructions clearly stated that those cases should receive a response of “No – but...”

- "Does the project follow a written policy/procedure/standard..." – Most of the KPA’s include a question of this type. These questions have three distinct foci:
  - Does the policy/procedure/standard exist,
  - Is it written down, and
  - Is it followed consistently?

- From my discussions with people during the kick-off and at other times, I knew that none of their policies was written down, so all of those questions should have been answered “No”.

- "Are measurements used to determine the status..." – Most of the KPA’s include a question of this type. The answers to these questions should all have been “No” because:
  - The company had no measurement program or metrics database, and I saw no evidence that any measurements were taken for anything, and
  - The comments indicated that most people interpreted the questions to be asking about measurements of work products, when they were actually referring to measurements of the processes themselves.

- "Are the activities for [the process] subjected to SQA [Software Quality Assurance] audit or review?" Again, the comments indicated that most people interpreted the questions to be asking about audits or reviews of work products, when they are actually referring to audits or reviews of the processes themselves.

The first lesson to be learned from this challenge is that misunderstandings should be forestalled by a thorough discussion of the instructions for answering the questions, and of the CMM-specific terminology used in the questions. The results of the questionnaire would be much more meaningful if you could be sure that the participants had a common understanding of what each question was asking and how to
respond to it. An extra half-hour focused on how to read and answer the questionnaire would have been time well spent.

The second lesson is that there will always be a diversity of opinion. Even in the same department, among people who interpreted the questions similarly, some people will answer “Yes” when others say “No”. The best option would be to probe the participants to understand the root of the differences. But in the case of a Mini-Assessment, when that option is not available, you must find a way to quantify such disparity and make it a part of the analysis.

5. MINI-ASSESSMENT RESULTS

I published the results of the Mini-Assessment on 2 March 2000, exactly a month after the Kick-off. (The delay was mainly due to waiting for surveys from the Marketing and management folks.) The results of the Mini-Assessment were not unexpected: the company was clearly Level 1. This in spite of the fact that most of the ratings should have been lower because of people answering “Yes” when they should have said “No” (see “Challenge #2”, above). The consensus was that no goal for any Key Process Area (KPA) was completely satisfied, and many goals were not even thought about.

One of the few bright spots from the Mini-Assessment was that they rated themselves surprisingly high on the Inter-Group Coordination KPA. They chose to capitalize on this perceived strength, making it a rallying cry: “Let’s continue to work together as we solve the process problems we have identified.”

Besides the “All Departments” analysis represented by the chart above, I also analyzed the data for each department to try to identify serious discrepancies in their views. I found that there was general agreement about most of the KPA’s across all of the departments. Only two KPA’s showed any significant disparity, and one of them was due to a consistent misinterpretation of the term “Defect Prevention” among the development staff. That left only one KPA with true disagreement; but since it was not a Level 2 KPA, this disagreement was not addressed in the initial planning.

Although the Mini-Assessment provided no surprises, and mostly confirmed what they already knew, there was none-the-less significant value in the exercise because it was a shared experience. It was not a matter of one person or one department (or even an outside consultant) pointing a finger at anyone else; rather the whole company was pointing a finger at itself. They were all in it together.
6. CHALLENGE #3: IDENTIFYING THE INITIAL PRIORITIES

We set the initial strategy using the results from the Mini-Assessment and these three principles:

- The CMM Principle: Focus on Level 2 first. The CMM is a progressive model, with the capabilities at the lower levels providing the foundation for those at the higher levels. According to the CMM, if you have problems with a process area that is part of Level 2, then you are unlikely to be successful at addressing process areas at any higher level.

- Attack things that everyone agrees are important problems. There will always be a diversity of opinion about what should be done first. Any time you can find agreement on an important problem, you should capitalize on it.

- Look for “low-hanging fruit” to generate some early wins. Some of the issues you will eventually deal with will be difficult and may take significant time to master. A record of achievement will provide the momentum you will need to complete the more difficult parts of the work, so focus first on the easier tasks.

The Software Subcontract Management KPA was a non-issue for this company because they had no subcontractors.

Software Configuration Management (SCM) was the lowest-rated of the other Level 2 KPA’s, suggesting that it should be an early priority. However it was not chosen for priority action because:

- We saw little opportunity for quick improvements (they already used automated code control, and most other SCM issues take significant time and effort to work out), and

- Many people did not see SCM as a problem area. (This would change, later!)

The remaining four Level 2 KPA’s were equally weak, generally recognized as problem areas, and each provided obvious opportunities for quick wins. So, three process teams were formed to address these four KPA’s.

6.1 Requirements Management

Everyone highlighted the Requirements Management KPA as a significant pain-point in the organization. No one was happy with the way it worked, and everyone could see opportunities for easy changes that would yield fast returns (though there was not agreement on exactly what those changes should be). Requirements Management was also recognized as an
important basis upon which all other work was dependent; so everyone agreed that it should be the top priority.

The Requirements Management team’s initial goals were to:
- Establish the requirements definition process and
- Refine the requirements template (agreeing on its contents and definitions of terms).

This team did not attempt to address the requirements change process, though everyone agrees that it would be a logical follow-on to the initial work.

6.2 Project Management

The Software Project Planning and Software Project Tracking & Oversight KPA’s were the other significant pain-points in the organization. There was significant confusion about how projects were initiated and planned, and problems with understanding the current status of projects was a recurring theme. There were clearly ample opportunities for quick improvements that would be quite beneficial. And like Requirements Management, Project Management was seen as a basic activity upon which all projects depend.

The Project Management team’s initial goals were to:
- Define the terms that are used in project planning and tracking (e.g. is a person-week 40 hours on task? Or does it include project overhead like team meetings? Or is it equivalent to a calendar week with all of the interruptions and wasted time that normally happen?),
- List all of the activities that must be planned and tracked for a project (including those done by QA, the technical writers and Marketing), and
- Define the process for creating a project plan.

Again, note that the initial goals do not include managing changes to plans.

6.3 Software Quality Assurance

SQA was not identified as a particular problem area, but we decided to include it in the initial priorities because it comprised significant opportunities. The SQA function was just being built from scratch, and everyone wanted to get it started off on the right foot.

The Quality Assurance Team’s initial goals were to:
- Define QA’s role in other departments’ activities (e.g. requirements definition, project planning, design reviews),
- Develop standards for test planning
- Define a standard testing process
Notice that the initial goals did not include any process assurance activities, only product assurance. This was done because product assurance had already begun to cause problems, and also because there were not yet any formal processes to assure. But with this in mind, the organizational structure and philosophies were being formed so that process assurance could be added as it became reasonable to do so.

The lesson from this challenge is simply that the guiding principles served us well. Any process improvement project must be based on sound principles that are expressly articulated. For any organization that is just starting a process improvement effort, the three principles identified above are a good starting point.

7. **CHALLENGE #4: ESTABLISHING A COMMON VIEW**

The three process improvement teams held their first meeting on 5 April 2000, a month after the Mini-Assessment findings were published. As with getting started, this delay was mainly due to the difficulty of getting senior management to discuss and agree on the actions to be taken.

In the very first set of meetings for the three teams, it became apparent that there was no common view of their software development process. Different people listed different sets of activities, and used the same words to mean different things. The disparity was most obvious between departments, though even within the development department there were significant differences among individuals.

By the end of the second meetings, a new highest-priority goal had been identified for the teams: Work with the other two teams to agree on a single description of the software development process. This included listing the steps in the process and defining the terms that were used, as well as identifying the parts of the over-all process on which each team would be focusing.

The project management team became the focal point for this effort, postponing work on their initial goals for the time being. The other two teams continued working toward their initial goals while participating in this work. The first one-to-two months of team meetings were spent agreeing on this common view of the development process. It took much more effort than anyone had anticipated, but it was a very educational and valuable exercise, and it provided the needed basis upon which the other work could be built.

In retrospect, it would have been good to collapse the three teams into one when we identified the need for a common view of the development
process. It likely would have facilitated the definition work and allowed all three teams to refocus back on their initial goals more quickly.

The lesson in this challenge is a cautionary one: Beware of the assumption that everyone knows how things are done today. In a level 1 organization, it is unlikely that the software process is that well understood. If the organization does not already have a high-level description of its process, then crafting one should be the first order of business. Without it, all other work will be most literally built on sand.

8. POSTMORTEM BEFORE STARTING

In late June 2000, as the three teams were working toward their initial goals, the company completed the development project that they had been working on. This provided a unique opportunity to do a postmortem analysis of the project to provide additional input to their process development teams.

This postmortem analysis yielded two important insights:

- Software Configuration Management (SCM) is a bigger problem than they realized. They discovered that they need to institute change control on all types of work products in order to bring some sanity to their work. This finding confirmed the results of the Mini-Assessment, and so SCM will be attacked next.

- Inter-Group Coordination is a root cause of many other ills. This contradicted their earlier opinion, showing that their ability to work together is not as good as they thought it was. They realized that although there are relatively good relationships among the groups, they need some formal mechanisms to insure that all coordination takes place as needed.

The lesson here is simple. Look for any source of information you can find. The process improvement work never goes on in a vacuum; people are always working on projects at the same time, and those projects can yield important insights.

A second lesson here is the value of project postmortems. In almost any situation, you can learn a tremendous amount about how your development process is working by investing a few hours in a postmortem workshop. Whether you hold a formal facilitated session, or just a "final" project team meeting, the insights you gain (if they are written down and acted upon) are worth their weight in gold. But beware: holding the postmortem meeting then ignoring the results can be devastating to the team members' morale!
9. INITIAL CHANGES IN PILOT TEST

With a new development project kicking off in July 2000, the three teams focused on identifying specific process changes that they could pilot test on the new project. Because of their early focus on "low-hanging fruit", they already had several process changes ready for pilot use.

These changes were pilot tested:

- Requirements Management:
  - The formal Requirements Definition process included steps for proposing, evaluating and prioritizing requirements, and for deciding on the actual content for the product version.
  - The Requirements Template included content guidelines for all sections and was based on commonly accepted definitions of terms.

- Project Management
  - The new Engineering templates provided an intermediate view of the system between the Requirements and Design specifications. This intermediate view was designed to facilitate the Engineering staff's evaluation of proposed Requirements, allow them to make more reasonable effort and schedule estimates, and provide a way to validate that the Design that is eventually specified accurately represents the intent of the Requirements Specification.
  - Added structure within the Engineering department was designed to allow them to more effectively carry out their wide variety of concurrent activities (e.g. requirements analysis for future versions, design & development for the current version, maintenance of the past version)

- Software Quality Assurance
  - Active early involvement of QA (and the technical writers) during Requirements Analysis and Design Review activities was done to improve the quality of the Requirements and Design Specifications, and at the same time, give the Quality Engineers a better understanding of the product that they would validate.
  - The new Test Planning process and standards would assure that reasonable, but complete tests had been specified and prepared while the software was in development.
  - The Testing process assured that both testing and problem tracking did not allow problems to "fall through the cracks".

The initial experience with these process changes was positive, and everyone was enthusiastically looking forward to the postmortem analysis of that project.
10. CONCLUSION

This company has done a commendable job of getting a fast start on their process definition work. From the date of their Mini-Assessment to the beginning of pilot testing some significant process changes was only 5 months. This is much faster than many process improvement efforts can move.

These process improvements should dramatically improve the stability of their projects, and demonstrate the value of process improvement. They should also provide the momentum that the company will need to continue with their process improvement work, especially the difficult job of establishing the change control mechanisms that they now recognize they need.

At the same time, it should be noted that these steps are only the beginning of a long process improvement effort. By themselves, these steps do not even bring the company close to achieving CMM Level2. Like any other company in any industry, process improvement for an Internet company is a long-term effort, even if it begins with a few simple steps.